



iMOCO4.E

Intelligent Motion Control under Industry 4.E

D3.7 Final design report on Perception & Instrumentation Layer

Due Date: M32 – 2024-04-30

Abstract:

This document is the final report of IMOCO4.E WP3, linked with the Perception & Instrumentation Layer. WP3 integrates five IMOCO4.E BBs (namely BB1, BB2, BB3, BB7, BB8) that have been developed and validated in the last two years. This document provides a comprehensive description of IMOCO4.E Perception and Instrumentation layer and details the functionalities developed in each WP3 BB, focusing on implementation aspects of relevant pilot applications, use cases and demonstrators.

Project Information

Grant Agreement Number	101007311
Project Acronym	IMOCO4.E
Project Full Title	Intelligent Motion Control under Industry 4.E
Starting Date	1 st September 2021
Duration	36 months
Call Identifier	H2020-ECSEL-2020-2-RIA-two-stage
Topic	ECSEL-2020-2-RIA
Project Website	https://www.imoco4e.eu/
Project Coordinator	Arend-Jan Beltman
Organisation	SIOUX TECHNOLOGIES BV (SIOUX)
Email	Arend-Jan.Beltman@sioux.eu

Document Information

Work Package	WP3 Perception and instrumentation Layer based on AI at the edge
Lead Beneficiary	EDI
Deliverable Title	Final design report on Perception & Instrumentation Layer
Version	1.0
Date of Submission	28/04/2024
Author(s)	Roberts Kadiķis (EDI) roberts.kadikis@edi.lv
Contributor(s)	<p>Álvaro Arco (SED) alvaro.arco@nav-timing.safrangroup.com</p> <p>Fabien Bouquillon (UNIMORE) fabien.bouquillon@unimore.it</p> <p>Diāna Dupļevska (EDI) diana.duplevska@edi.lv</p> <p>Anatolijs Zencovs (EDI) anatolijs.zencovs@edi.lv</p> <p>Artūrs Šimkūns (EDI) arturs.simkuns@edi.lv</p> <p>Aly Oraby (EDI) aly.oraby@edi.lv</p> <p>Lorenzo Diana (EVI) lorenzo.diana@huawei.com</p> <p>Tassos Kanellos (GNT) tasos.kanelos@gnt.gr</p> <p>Ilektra Simitsi (ITML) isimitsi@itml.gr</p> <p>Davide Plaia (HS) Davide.Plaia@Hahn-Schickard.de</p> <p>Zdeněk Havránek (BUT) zdenek.havranek@ceitec.vutbr.cz</p> <p>Martin Doseděl (BUT) martin.dosedel@ceitec.vutbr.cz</p> <p>Fabien Bouquillon (UNIMORE) fabien.bouquillon@unimore.it</p> <p>Lars Meyer (IMST) lars.meyer@imst.de</p> <p>Lorenzo Diana (EVI) lorenzo.diana@huawei.com</p> <p>Rosana Dias (INL) rosana.dias@inl.int</p> <p>Rob Snel (TNO) rob.snel@tno.nl</p> <p>Eric Zoeb (EDILASIO) eric.zoeb@edilasio.pt</p> <p>Gabriel Ribeiro (EDILASIO) gabriel.ribeiro@edilasio.pt</p> <p>Marvin Wiedemann (IML) marvin.wiedemann@iml.fraunhofer.de</p> <p>Nehal Amer (ADI) Nehal.Amer@analog.com</p> <p>Alan Dunne (EMDALO) alan.dunne@emdalo.com</p> <p>Alfie Keary (UCC) alfie.keary@tyndall.ie</p> <p>Sajid Mohamed (ITEC) sajid.mohamed@itecequipment.com</p> <p>Vibhor Jain (TUE) v.jain@tue.nl</p>

	Dip Goswami (TUE) d.goswami@tue.nl					
Internal Reviewer(s)	Rosana Dias (INL) rosana.dias@inl.int Marcel van der Kraan (TNO) marcel.vanderkraan@tno.nl					
Document Classification	Draft			Final		X
Deliverable Type	R	X	DEM		DEC	OTHER
Dissemination Level	PU	X	CO		CI	

History					
Version	Issue Date	Status	Distribution	Author	Comments
0.1	17/10/2023	Draft	WP3 partners	Roberts Kadikis	First draft with document structure
0.5	04/03/2024	Draft	WP3 partners	Roberts Kadikis	Draft with all partner inputs, ready for internal review
0.8	20/03/2024	Draft	WP3 partners	Roberts Kadikis	Updated draft with comments from internal reviewers
1.0	27/04/2024	Final	Public	Roberts Kadikis	Document is improved after comments from internal reviewers

Type of Contribution	
Partner	Description of Contribution to Contents
EDI	Deliverable owner. Integrated partner contributions. As a task leader, contributed to the overview Section 4.1. Contributed Section 4.2.
INL	As a task leader, contributed overview Section 2.1 on T3.2 solutions. Contributed section 2.4.
BUT	Contributed Section 2.2.
UWB	Contributed Section 2.3. As a task leader, contributed overview Section 5.1.
EDILASIO	Contributed Section 2.4.2.
ADI	Contributed Sections 2.5 and 3.4.
IMST	Contributed Sections 2.6 and 3.5.
REEXEN	Contributed Sections 2.7, 3.8, and 4.8.
EMC	As a task leader, contributed overview Section 3.1. Contributed Section 5.2.
SoC-e	Contributed Sections 3.2 and 4.11.
REX	Contributed Section 3.3.
UCC	Contributed Section 3.4.
EVI	Contributed Section 3.6.
EXE	Contributed Section 3.7.
SED	Contributed Section 3.9 and 4.10.
HS	Contributed Sections 3.10 and 4.6.
UNIMORE	Contributed Section 3.11.
SIOUX	Contributed Section 3.12.
GNT	Contributed Section 4.3.
ITML	Contributed Section 4.3.
TUE	Contributed Section 4.4.
ITEC	Contributed Section 4.4. and to the IMOCO4.E reference framework.
TNO	Contributed Section 4.5.

IML	Contributed Section 4.6.
STILL	Contributed Sections 4.6. and 4.7.
DTT	Contributed Section 4.7.
EMDALO	Contributed Section 4.9.
REX CONTROLS	Contributed Section 5.3.

Table of Contents

1	Introduction.....	19
1.1	Purpose of this Deliverable	19
1.2	Structure of this deliverable	19
1.3	Intended readership.....	19
1.4	Relation to other activities in the project	19
2	Solutions of Task 3.2: Novel sensors (new type of sensors, wireless communications, self-powered, low powered) (BB3)	21
2.1	Overview.....	21
2.2	Vibration diagnostic sensor (BUT)	22
2.3	Vibration and motion sensor (UWB).....	35
2.4	Overmolded temperature and pressure sensors (INL/ECS)	44
2.5	Low motion distortion 3D depth sensor (ADI)	57
2.6	Radar sensor (IMST).....	66
2.7	Dynamic vision sensor (Reexen)	73
3	Solutions of Task 3.3: SoC/FPGA platforms for smart control and signal processing (BB1).....	78
3.1	Overview.....	78
3.2	TSN platform based on FPGA (SoC-e)	80
3.3	Motion smart sensor (REX).....	86
3.4	Advanced wireless embedded processing platform (UCC, ADI)	91
3.5	Modular hardware platform based on a mini-PC (IMST).....	99
3.6	Modular platform for Smart Sensors with Advanced Data Processing (EVI)	105
3.7	Rugged embedded IO-controller platform (Exertus)	108
3.8	New technologies of energy-efficient AI processors (REEXEN).....	111
3.9	Platform for standard TSN bridging delivering time-aware scheduling (SED).....	120
3.10	AutoFlow: open-source software solution (HS)	125
3.11	Multiple sensors (UNIMORE).....	128
3.12	Vision-in-the-loop system (SIOUX).....	131
4	Solutions of Task 3.4: AI-based components (BB2, BB8)	133
4.1	Overview.....	133
4.2	Real-time AI-based processing component, using modern reinforcement learning techniques with the long-term aim of reducing the setup and deployment time of AI-based industrial machinery (EDI)	135

4.3 AI-based, real time data clustering and classification at the edge, including deployment of a data fusion mechanism able to aggregate, pre-process, and further analyze data from multiple different IoT sources in Industrial environments (ITML, GNT)..... 144

4.4 Energy-efficient and fast vision-processing system, based on AI and approximate computing techniques (incl. x-in- the-loop framework for development, testing and debugging (TUE)..... 155

4.5 Autonomous smart motion sensor based on a high speed camera (TNO) 161

4.6 Deep neural networks based system for deployment on robotic systems to enable recognition of the scene and understanding of complex situations in typical warehouses or production sites – load carrier pose estimation based on synthetic data. (IML, STILL). Optimization of these neural networks for FPGAs and MCUs with a focus on low-power (HS) 169

4.7 Deep neural networks based system for deployment on robotic systems to enable recognition of the scene and understanding of complex situations in typical warehouses or production sites - radar signal simulation for AI-driven scene recognition and complexity understanding (DTT, STILL)..... 176

4.8 Deep learning algorithm for Vision, audio, acceleration, pressure sensor data analysis; Ultra-low power ASIC/FPGA hardware implementation of neural networks (Reexen)..... 184

4.9 Real-time predictive models, accelerated for embedded processing accounting for latencies and deficiencies in gesture recognition (EMD) 189

4.10 Adaptable and ultra-reliable Time Sensitive Networking to support real-time AI-based acquisition, control and processing (SED)..... 196

4.11 General purpose AI operations speeding-up module, leveraging on FPGA fabric specific resources (SOC-e)..... 200

5 Solutions of Task 3.5: High Performance servo drives, variable speed drives (BB7) 212

5.1 Overview..... 212

5.2 EMI reduction methods (EMCMCC) 213

5.3 AMC - Advanced Motion Controller (REX CONTROLS) 219

6 Conclusions and next steps 240

List of Figures

Figure 1. Perception and Instrumentation Layer in IMOCO4.E framework architecture.....	20
Figure 2. Connection of the BB3 developments with the P/D/UC and other BBs.	21
Figure 3. Block topology of a Smart Wireless Sensor.	24
Figure 4. Photography of the designed wireless gateway.	26
Figure 5. Architecture of the vibration diagnostic sensor with emphasized main components.	27
Figure 6. Render of the PCBs (left), real prototype of the assembled sensor (middle) and sensor encapsulated in the 3D printed housing (right).....	27
Figure 7. Acceleration impulse signal and sinus waveform captured by the SWS.....	28
Figure 8. Photography inside the lift testing tower with the SWS attached to the lift cantilever, the wireless gateway and a PC running the application for data visualization.	29
Figure 9. Data from the lift right in the upward direction with no noise.	30
Figure 10. Data from the lift right in the upward direction with additional noise.	30
Figure 11. Comparison of the low frequency and high frequency sensing elements.....	31
Figure 12. Photography of the SWS, the gateway and LabVIEW visualization application in PC.....	31
Figure 13. Control loopback realized using wireless link. Links from the controller to the actuator at the controlled system as well as from the sensor installed at the controlled system to the controller are wireless. UWB has demonstrated, that sufficiently reliable radio link can be established to control even highly unstable and fast systems.	36
Figure 14. Imagination sensor prototype	37
Figure 15. RETIS principle of operation. The protocol can simultaneously operate on multiple RF channels or even bands. TFDMA (Time-Frequency Division Multiple Access) principle is used – base transmission cycle is divided into communication slots, which are assigned for certain nodes -> the communication is deterministic, with predictable delivery time and minimal jitter. RF nodes have right to transmit during the assigned slots only. Allowed is only one packet per slot, however multiple messages can be encapsulated in the packet. Master node distributes clock signal so that accurate global clock is available at all nodes.....	38
Figure 16. Imagination measurement set - sensor and DIN-rail mounted gateway with possibility to trigger measurement with digital signal.	40
Figure 17. Imagination sensor mounted on a lift cabin in test shaft in WEG. Our sensor (black) is mounted next to the commercial reference sensor (orange/black).....	40
Figure 18. SimpleAccelStreamViewer application used to visualize and log data from Imagination sensors	43
Figure 19. a) Layout of the NFC microantenna and b) main micromachining process steps.	47
Figure 20. a) photograph and microscope image (close-up) of the fabricated devices Temperature sensing tag before encapsulation, b) encapsulation schematic and c) picture of encapsulated tag.....	48
Figure 21. PCB-based tag layouts.....	49
Figure 22. Overmoulding trials with first samples (non-functional prototypes) to evaluate geometric constraints on the injection tool.	49
Figure 23. A microantenna run wafer before the final release step (left) and released devices (right).	50
Figure 24. Devices coil resistance values mapped in the wafer before release.	50
Figure 25. Climate chamber temperature characterization setup.....	51
Figure 26. Pressure characterization setup.....	52
Figure 27. Microfabricated tag temperature measurements. (Left) Data acquisition of the NFC tag and temperature reference sensor from -40 to 85 °C; (Right) Data interpolation of the sensors in the left.....	53

Figure 28. PCB based tag characterization against temperature. (Left) Data acquisition of NFC tag and temperature reference sensor from 25 to 65 °C; (Right) Data interpolation of the sensors in the left. (Negative values not possible with the firmware version of this particular tag)..... 53

Figure 29. Pressure measurements of the PCB-based tag using the vacuum line (90-160 mbar). (Left) Data acquisition of the NFC tag and pressure reference sensor; (Right) Data correlation of the sensors. 53

Figure 30. Pressure measurements of the PCB-based tag using the compressor (approximately 2.2 bar). Data acquisition of the NFC tag and pressure reference sensor; (Right) Data correlation of the sensors. . 54

Figure 31. Batch interrogation mode continuous operation. 54

Figure 32. Injection molded part with overmolded encapsulated sensor. 55

Figure 33. ADTF3175 3D ToF Module 57

Figure 34. ADTF3175 3D ToF Module Data Sheet. 59

Figure 35. Use case 3 Robot Tele-operations Overview..... 60

Figure 36. ADTF3175 Logical Processing Overview..... 61

Figure 37. ADTF3175 ToF Wrist Tracking in UC3. 61

Figure 38. Continuous Wave Time of Flight System Features 64

Figure 39. Time of Flight Sensor Components. 65

Figure 40. 77 GHz Radar Sensor with external USB camera. 68

Figure 41. Radar processing steps implemented into the firmware. 69

Figure 42. Radar implemented in Demo3 (left) and Pilot 4 (right)..... 70

Figure 43. Schematic of the operation of a DVS pixel, converting light into events. 73

Figure 44. Events in spacetime on the left side of the figure are color coded, from green (past) to red (present), The right side of the figure is the frame and overlaid events of a natural scene, the frames lag behind the low-latency events (colored according to polarity). 73

Figure 45. DVS chip. 74

Figure 46. DVS USB camera. 74

Figure 47. IMX636 camera. 75

Figure 48. RK3588 computing board. 75

Figure 49. Visualized results of human detection, numbers represent confidence values. 76

Figure 50. Mapping of the platforms with the project framework. 78

Figure 51. General blocks diagram. 81

Figure 52. Zynq Ultrascale+ MPSoC reconfigurable platform block diagram..... 81

Figure 53. Proposed communication interfaces. 82

Figure 54. 3D PCB..... 82

Figure 55. Time Aware Bridge error DUT as slave and multiple slaves connected to the DUT..... 84

Figure 56. Time Aware Bridge error DUT as slave and multiple slaves connected to the DUT-IXIA results. 84

Figure 57. Time Aware Bridge error DUT as slave and multiple slaves connected to the DUT-Calnex results. 84

Figure 58. Solution overview. 87

Figure 59. Configuration of the control system in REXYGEN Studio..... 88

Figure 60. Comparison of MOSS and standard calculation results. 89

Figure 61. Use Case 3 - Local Sensor End Architecture..... 91

Figure 62. Use Case 3 - Remote Robot End Architecture. 92

Figure 63. Use case 3 - Edge to Edge SW Architecture. 93

Figure 64. Use case 3 Sensor/Actuator Layer Local End Hardware..... 94

Figure 65. Use Case 3 – Sensor/Actuator Layer Remote End Hardware. 95

Figure 66. Application of Mini-PC on Demonstrator 3. 100

Figure 67. Assembly of Jetson-PC and connected sensors via Ethernet switch. 100

Figure 68. Connected hardware components. 101

Figure 69. Connector Ports of the Jetson.[1] 101

Figure 70. Radar, Camera are connected over the Jetson PC on a Philipps C-arc..... 102

Figure 71. Synchronization of sensor node data. 103

Figure 72. Camera data synchronized with radar data 104

Figure 73. Developer Kit Atlas 200 DK AI. 105

Figure 74. Atlas 300I Inference Card..... 105

Figure 75. High-level logical architecture of addressed scenario. 106

Figure 76. Setup employing the Huawei Atlas 200 DK..... 106

Figure 77. Setup employing the Huawei Atlas 300I inference card. 106

Figure 78. Example architecture of distributed mobile control system consisting of (from left to right) touch display, IoT connectivity unit, master controller, 2 CAN hubs and 5 IO controllers. 109

Figure 79. Rugged IO-controller software architecture diagram. 109

Figure 80. NPU Data Flow Direction. 112

Figure 81. ADA100 layout (Top) and ADA100 layout (Bottom). 112

Figure 82. ADA100 Smart sensor AI-processor. 113

Figure 83. ADA100 Structure. 113

Figure 84. System Block Diagram. 114

Figure 85. Unique selling point of car. 116

Figure 86. Test Record..... 116

Figure 87. ADA100 PPG sensor and signal conditioning circuit. 117

Figure 88. The difference of pose curve for the hand motion..... 117

Figure 89. Test environment and equipment connection diagram. 118

Figure 90. ADA100 super low-power Smart-mic..... 119

Figure 91. Diagram of the Programmable FPGA Logic. 120

Figure 92. a) Setup at SIOUX offices. b) Setup in operation. 123

Figure 93. Example results for traffic prioritization. 123

Figure 94. Solution overview..... 126

Figure 95. Xilinx ZCU102 Configuration for Pilot3. 129

Figure 96. NFC Reader setup..... 129

Figure 97. Data from NFC card printed on screen..... 130

Figure 98. Vision-in-the-loop system and deployment block. 131

Figure 99. Zynq UltraScale+ MPSoC platform diagram. 132

Figure 100. Mapping for functional parallelism. 132

Figure 101. Mapping for 2x data parallel pipeline..... 132

Figure 102. Function parallel pipeline. 132

Figure 103. 2x Data parallel pipeline..... 132

Figure 104. Object detection in RGB image and corresponding depth map..... 136

Figure 105. Motion planning using MoveIt2 in ROS2, visualised with RViz..... 137

Figure 106. Demonstrator 4 in the laboratory environment and its architecture. 137

Figure 107. Sim2Real translation example. Synthetic image on the left is made more lifelike and displayed on the right side..... 139

Figure 108. Block view diagram of SW-018 (Solution 2) operation, including interconnection with other IMOCO4.E BBs and positioning on edge and fog layers. 144

Figure 109. Vega Shrink-Wrapper new-worn blade data, six dataset, three new and three worn. 146

Figure 110. Implementation of SW-018 in Pilot 3 and interactions with other components..... 153

Figure 111. Set of semiconductor dies located on a silicon wafer. The encoder reads the position of the wafer and the camera captures an image of the (partial) die(s) in the ROI. The dies are positioned at the reference by moving the wafer..... 155

Figure 112. Block diagram of the system, with the linear encoder and camera as sensors, multi-sensor estimator, and PID controller. 155

Figure 113. Coordinates of dies for motion control with and without vision feedback for misalignment profiles m1 and m2 compared with the ideal scenario..... 156

Figure 114. Synchronous simulation framework for vision-in-the-loop system 160

Figure 115. Interrupt response latency histograms showing standard Linux (left) and improved (real time) kernel + adaptations by TNO (right)..... 161

Figure 116. XY motion platform and CoaXPress high speed camera. 162

Figure 117. Computer rack with host PC, PCI-extender, Main switch units and auxiliary electronics box 163

Figure 118. Bode diagram of tuned motion system 164

Figure 119. Overview of the technology chain of the used software components. 170

Figure 120. Implementation principle of the Synthetic Dataset Generator (SW-021)..... 171

Figure 121. Implementation of the Pose Estimation Model (SW-022)..... 172

Figure 122. Sample synthetic training images including random background, different textures for floor and background, flying distractors as well as a simulated warehouses. 173

Figure 123. Detected pallets including a red 2D bounding box from the Object Detection, the image crop marked in blue and the final green 3D bounding box of the Pose Estimation. The images are manually cropped to focus on the detected pallet, the FoV of the used camera stream is considerably larger and includes different environment objects. 173

Figure 124. Extraction of target parameters from Blender outputs [1]..... 177

Figure 125. a) Schematic overview of the simulation pipeline. (b) High-level schematic overview of the FMCW radar principle. The signal paths are replicated when more receiver or transmitter antennas are used [4]..... 178

Figure 126. Sample scene in Blender showing the virtual radar with two detectable objects. 179

Figure 127. a) Z-pass, b) combined-pass. 179

Figure 128. Number of targets detected for each frame..... 180

Figure 129. Sample scene in Unity 3D showing the virtual radar with two detectable objects..... 180

Figure 130. The range of the detected objects with radar and detected objects in stationary state..... 181

Figure 131. The range of the detected objects with radar and one of the detectable objects in stationary state and one of the detectable objects disappearing from the scene..... 181

Figure 132. ADA200 CIM TEST Chip..... 184

Figure 133. 3 data flow and CIM energy-efficient AI processors. 185

Figure 134. Chip clock logic..... 185

Figure 135. Test ADA200 Chip and YOLO Result..... 186

Figure 136. Use Case 3 - Local Sensor End Architecture..... 189

Figure 137. Use Case 3 - Remote Robot End Architecture. 190

Figure 138. UC3 AI and ML research on gesture recognition..... 191

Figure 139. Structure of the TSN monitoring tool..... 197

Figure 140. Configuration options of TSN. 198

Figure 141. Integration of DPU and Time-Sensitive Networking capabilities in the endpoint. 202

Figure 142. Fully ML Design Flow achieved for YOLO Model..... 203

Figure 143. Embedded software for latencies measurements development and for bounding boxes position and size identification. 204

Figure 144. Integración en el diseño de una tarjeta de ampliación FMC..... 205

Figure 145. TSN traffic in the concept-proof system. 206

Figure 146. AI complete design flow using Pre-trained models or "Full Flow" 209

Figure 147. Vitis AI Design Flow..... 209

Figure 148. Example of an intended VAJ profile, before local feedback. 214

Figure 149. Example of active mains harmonic compensation circuitry 215

Figure 150. Effect on outer screen current w.r.t. measures taken: 217

Figure 151. Effect on outer screen current w.r.t. measures taken: 218

Figure 152. AMC servo drive. 219

Figure 153: HW block diagram 221

Figure 154. Firmware main tasks..... 223

Figure 155. Typical control application. 226

Figure 156. Signal generator. 227

Figure 157. Signal monitor. 227

Figure 158. Signal monitor chart. 228

Figure 159. Process trace. 228

Figure 160. MicroREX application in REXYGEN studio..... 229

Figure 161: Configuration and control software..... 230

Figure 162: UI related to custom application..... 232

Figure 163. Test stand for smart control algorithm research. 233

Figure 164. Basic repetitive control schematics. 233

Figure 165. 7-axis robotic manipulator..... 234

List of Tables

Table 1- Parameters of SWS.....	26
Table 2- Requirements that apply to vibration diagnostic sensor development and its deployment in UC 1.	32
Table 3- Characteristics of the vibration and motion sensor.....	37
Table 4 - Requirements for the vibration and motion sensor.....	41
Table 5. List of types of sensors/tags developed.....	44
Table 6. List of the hardware, software and integration ID’s involved on this sensor and its deployment on demonstrator 2	45
Table 7. Theoretical (simulated) and experimental data of the microfabricated devices. Singular representation of each design.....	51
<i>Table 8. Verification of requirements for HW-021.</i>	55
<i>Table 9. Verification of requirements for low motion distortion 3D depth sensor</i>	62
Table 10: Resolution settings of measurement domains.....	69
Table 11: Requirements for the Radar sensor.....	71
Table 12: Requirements for the dynamic vision sensor.....	76
Table 13: Platform contribution to P/UC/D.....	78
Table 14: HW-016 verification results table.....	85
Table 15: Requirements for wireless embedded processing platform	96
Table 16: Requirements for Modular hardware platform based on a mini-PC.....	104
Table 17: Requirements for Modular platform for Smart Sensors with Advanced Data Processing	107
Table 18. Requirements for Rugged embedded IO-controller platform.....	110
Table 19: Requirements for energy-efficient AI processors	115
Table 20: HW-015 Requirements.....	124
Table 21: INT-020 Requirements.....	124
Table 22 Average latency and bandwidth for sensors in Pilot 3.....	130
Table 23. Results of the object detection, also described in paper [4].....	138
Table 24. Requirements related to solutions for Demo 4.....	139
Table 25. Comparison of clustering performance evaluation measurements for 3 algorithms (K-means, AP, Mean Shift) and for 3 Time Window sizes (50, 250, 500 samples). For each measurement type, green denotes the best performance and red the worst.	147
Table 26. Comparison of classification performance evaluation measurements of the model trained using the K-Means clustering algorithm in 3 cases of Time Window size (50, 250, 500 samples). For each measurement type, green denotes the best performance and red the worst.	148
Table 27. Architecture layer-specific requirements.....	149
Table 28. Overall (system-level) requirements.....	149
Table 29. BB8 (AI-based components) requirements.....	150
Table 30. Configuration of the multi-die scenario.....	156
Table 31 Performance of the closed loop system.....	157
Table 32 Architecture layer-specific requirements.....	157
Table 33 Overall (system-level) requirements.....	158
Table 34 BB2 (high-speed vision-in-the-loop) requirements	159
Table 35. Architecture layer-specific requirements.....	164
Table 36. Overall (system-level) requirements.....	165
Table 37. BB2 (high-speed vision-in-the-loop) requirements.....	166

Table 38. Software components of solution Error! Reference source not found. and their link to the IMOCO4.E architecture.....	169
Table 39. Requirements for Radar Signal Simulation for AI-driven Scene Recognition and Complexity Understanding.....	182
Table 40. Requirements for solution.....	186
Table 41. Requirements for solution.....	193
Table 42. SW-075 Requirements.....	199
Table 43. Logic resourced required for the system implementation.....	205
Table 44. Latency results for the Real-Time Video analytics application.....	206
Table 45. Requirements for General purpose AI operations speeding-up module.....	207
Table 46. Requirements that apply to High speed servo drives, variable speed drives.....	235

Abbreviations

Abbreviation	Explanation
3D CAD	3-dimensional Computer Aided Design
ABS	Acrylonitrile Butadiene Styrene
ADAS	Advanced Driver Assistance Systems
ADC	Analog-to-Digital Converter
AER	Address-Event Representation
AGV	Automated Guided Vehicles
AHF	Active Harmonic Filter
AI	Artificial Intelligence
AIC	Advanced Industrial Controller
AMC	Advanced Motion Controller
API	Application Programming Interface
AR	Augmented Reality
ASIC	Application-Specific Integrated Circuit
ASR	Audio Scene Recognition
BAR	Base Address Register
BGA	Ball Grid Array
BLDC	Brushless Direct Current motor
BLE	Bluetooth Low Energy
BNN	Binary Neural Network
CAN	Controller Area Network
CAPEX	Capital expenditures
CatE	Computing at the Edge
CDC	Current-to-Digital Converter
CFAR	Constant False Alarm Rate
CIM	Computing in Memory
CMOS	Complementary Metal–Oxide–Semiconductor
CNC	Central Network Controller
CNN	Convolutional Neural Network
CW	Continuous Wave
DAC	Digital-to-Analog Converter
DC	Direct Current
DDR4	Double Data Rate Fourth Generation
DMA	Direct Memory Access
DNN	Deep Neural Network
DoF	Degree of Freedom
DOPE	Deep Object Pose Estimation
DPU	Deep Learning Processor Unit
DS-CNN	Depth-wise Separable Convolutional Neural Network
DT	Digital Twin
DUT	Device Under Test
DVS	Dynamic Vision Sensor
ECG	Electrocardiogram
EDSA	Enhanced Distributed Scalable Architecture
EMAC	Ethernet Media Access Controller
EMC	Electromagnetic Compatibility

EMI	Electromagnetic Interference
eMMC	Embedded MultiMediaCard
FFT	Fast Fourier Transform
FID	Fréchet Inception Distance
FMC	FPGA Mezzanine Card
FMCW	Frequency Modulated Continuous Wave
FOI	Field of Illumination
FOV	Field of View
FPGA	Field-Programmable Gate Array
FPS	Frames Per Second
FPU	Floating Point Unit
FRER	Frame Replication and Elimination for Reliability
FSR	Full Scale Range
FW	Firmware
GEM	Gigabit Ethernet MAC
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HMI	Human Machine Interface
HPC	High-Performance Computing
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IGMP	Internet Group Management Protocol
IMU	Inertial Measurement Unit
IR	Infra-Red
ISR	Interrupt Service Routine
JTAG	Joint Test Action Group
KNN	K-Nearest Neighbours
KWS	Keyword Spotting
MAC	Medium Access Control
MAE	Mean Absolute Error
MCIC	Memory-Compute integrated Chip
MCU	Microcontroller
MDIO	Management Data Input/Output
MEMS	MicroElectroMechanical Systems
MIMO	Multiple Input Multiple Output
MIPI	Mobile Industry Processor Interface
ML	Machine Learning
MTSN	Multiport-TSN Switch
LF	Low Frequency
Li-Pol	Lithium polymer
LLDP	Link Layer Discovery Protocol
LSTM	Long Short-Term Memory
NFC	Near Field Communication
NGVA	NATO Generic Vehicle Architecture
NN	Neural Network
NPU	Neural Processing Unit
OEM	Original Equipment Manufacturer

OtA	Over-the-Air
PC	Polycarbonate
PCB	Printed Circuit Board
PHY	Physical
PID	Proportional–Integral–Derivative
PL	Programmable Logic
PLA	Polylactic Acid
PLC	Programmable Logic Controller
PNP	Perspective-n-Point
PPG	Photoplethysmogram
PS	Processor System
PTP	Precision time Protocol
PWM	Pulse Width Modulation
QSPI	Queue Direct Memory Access
R-CNN	Region-based Convolutional Neural Network
RDM	Range-Doppler Map
RETIS	Real-Time Sensitive Wireless Communication Solution for Industrial Control Applications
RGB	Red, Green, Blue
RF	Radio Frequency
RNN	Recurrent Neural Network
ROI	Region of Interest
ROS	Robot Operating System
RTDE	Real Time Data Exchange
RTOS	Real-Time Operating System
RTSP	Real Time Streaming Protocol
SDK	Software Development Kit
SIL	Software-in-the-Loop
SNMP	Simple Network Management Protocol
SoC	System on a Chip
SoM	System on Module
SPI	Serial Peripheral Interface
SSD	Single Shot MultiBox Detector
SSV	Steady State Value
STO	Safe Torque Off
SWS	Smart Wireless Sensor
TAS	Time-Aware traffic Shaper
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TFDMA	Time-Frequency Division Multiple Access
ToF	Time of Flight
TOPS	Tera Operations per Second
TSN	Time Sensitive Networking
TWS	True Wireless Stereo
UDP	User Datagram Protocol
ULP	Ultra-Low Power
VAD	Voice Active Detection
VAJ	Velocity, Acceleration and Jerk
VCSEL	Vertical Cavity Surface Emitting Laser

VCU	Video Codec Unit
VIL	Vision-in-the-Loop
VLAN	Virtual Local Area Network
VLC	VideoLAN Client
VR	Virtual Reality
WEEE	Waste from Electrical and Electronic Equipment
YOLO	You Only Look Once

Executive Summary

Deliverable 3.7 provides description of solutions and building blocks of IMOCO4.E framework's Layer 1 – Perception and instrumentation layer. The solutions are arranged by the WP3 tasks they were developed in:

- T3.2 Novel low-powered/self-powered real-time sensors - contributed to BB3.
- T3.3 New SoC, FPGA and multi-many core platforms for AI and smart data processing – contributed to BB1.
- T3.4 AI based perception systems and modules – contributed to BB2 and BB8.
- T3.5 High speed servo drives, variable speed drive – contributed to BB7.

Overall, the WP3 addressed ST1, ST2, ST4, SI1, SI2, SI3, SO1, SO2, SE1, SE2 objectives of IMOCO4.E project. Section on each task points out to precise technical objectives as well as targeted KPI's achieved by its solutions. Then follows a detailed description of all the solutions developed in the task. Solution's implementation details are given, along with results, compliance with requirements, customisation options, strengths and weaknesses, toolchains and lessons learned.

This is the final deliverable of WP3 and brings forth solutions that are ready for integration and use in project's demonstrators, use-cases, and pilots during activities of WP6 and WP7.

1 Introduction

1.1 Purpose of this Deliverable

This deliverable describes solutions developed in the WP3 of IMOCO4.E project and the status of those solutions by the end of WP3. The solutions contribute to five building blocks of IMOCO4.E framework – BB1, BB2, BB3, BB7, BB8. This document aims to not only give a technical overview of the solutions, but also disclose implementation details, strengths, customisation options, limitations, and solution's adherence to the requirements previously defined in the project.

1.2 Structure of this deliverable

Following this introduction, are the four main chapters of the deliverable. The chapters correspond to the tasks T3.2-T3.5 of the work package. Solutions of T3.2 directly correspond to BB3, solutions of T3.3 correspond to BB1, solutions of T3.5 correspond to BB7, but the T3.4 contributes to two building blocks – BB2 and BB8. This was the reason for structuring chapters by project's tasks and not by building blocks. Each section begins with an overview of its solutions, followed by project's technical objectives and KPIs these solutions are targeting.

Under each chapters, each solution has its own section. These sections follow a common structure, introducing the technology, describing implementation aspects, showing achieved results, and connecting the solution to previously designed requirements. Also, capabilities and limitations of the solution, its customisation an adaptation options, as well as necessary methodology and toolchains are discussed.

1.3 Intended readership

Deliverable D3.7 is disseminated as a public document. As the document collects wide range of technologies developed in WP3, it can be used by project reviewers to check the progress in WP3, by IMOCO4.E consortium partners to coordinate further developments in the project, and by external stakeholders and other interested parties to find out about developments of the IMOCO4.E WP3.

1.4 Relation to other activities in the project

WP3 dealt with the Perception and Instrumentation layer (Layer 1) technologies (Figure 1). Several SW and HW components of building blocks were developed and integrated into demonstrators, use-cases and pilots as solutions. The development of solutions was done in the tasks T3.2, T3.3, T3.4, and T3.5.

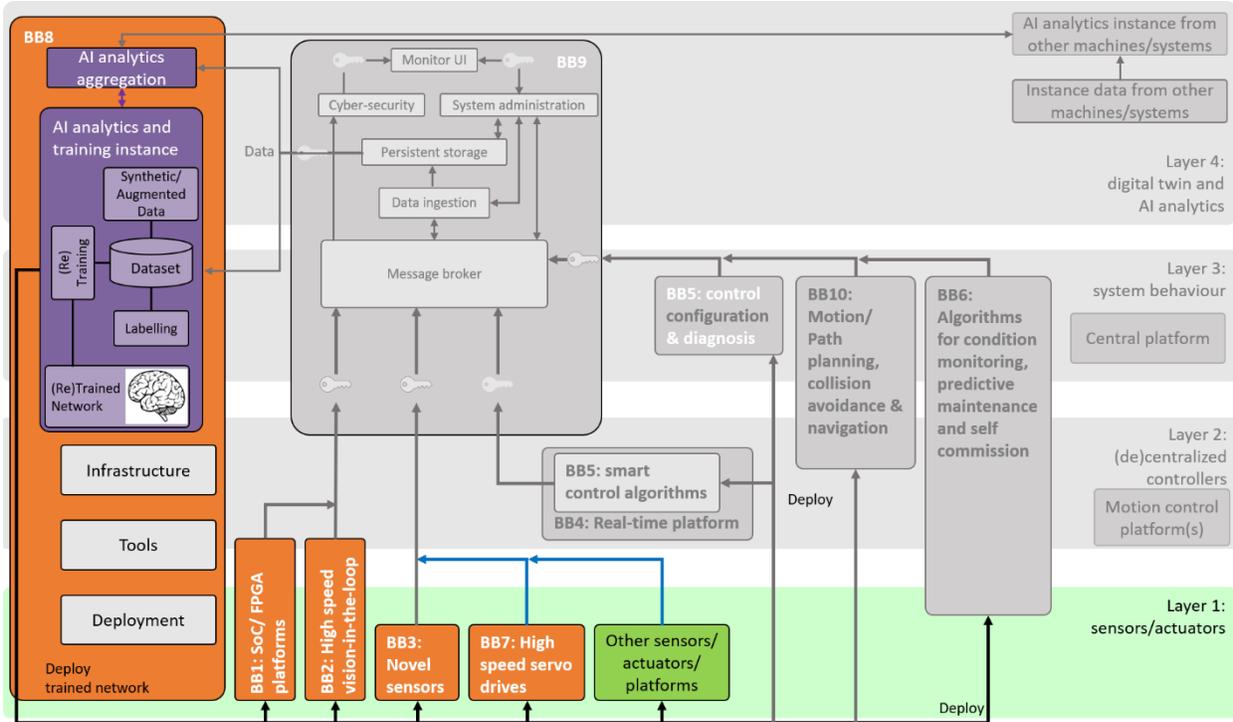


Figure 1. Perception and Instrumentation Layer in IMOCO4.E framework architecture.

Requirements and specifications for these solutions were defined in the task T3.1 of this work package and were described in deliverables D3.1 and D3.2. The task T3.1 ended prior other tasks of the WP3, and it is not further discussed in this deliverable. However, requirements defined in D3.1 and D3.2 are brought up for each solution in this document to measure the progress towards reaching the requirements. T3.1 requirements and specifications followed the system-level specification developed in WP2. As a result, the developed solutions also follow those specifications.

Timewise, WP3 was a parallel activity to WP4, WP5. While separate components were developed in those work packages, solutions from different WPs had to be integrated together into specific demos, use-cases, and pilots. Current deliverable mainly focuses on the WP3 solutions on their own.

Initial design of solutions was reported in deliverables D3.3, D3.4, D3.5, and D3.6. At that point project’s milestone MS4 was reached and developed solutions were brought to WP6 and WP7 for validation and trials. Feedback from those WPs was used to iteratively improve the solutions till the final stage described in this document. Now the work in WP3 closes, and the components and solutions are integrated into specific applications in WP6 and WP7.

Developments of WP3 have been a basis for scientific papers and other dissemination activities, thus contributing to WP8.

2 Solutions of Task 3.2: Novel sensors (new type of sensors, wireless communications, self-powered, low powered) (BB3)

2.1 Overview

2.1.1 Solutions of T3.2

The hardware solutions developed and tests reported in this task demonstrate tangible results of BB3 – Novel low/self powered real-time sensors. Each sensing solution has a section with technology description, main characteristics, functionalities and how they address requirements defined in WP3 (D3.2) and WP2 (D2.4). There are 6 different sensing solutions developed for different parameters/functionality: two for mechanical vibration targeting different specs/capabilities; pressure and temperature; 3D depth sensing, radar, and a dynamic vision sensor. These different BB3 sensors are deployed in different P/UC/D and connect to other building block within the IMOCO4.E framework (Figure 2):

- BB3.1: Vibration diagnostic → UC1
- BB3.2: 9-DoF Motion and vibration → UC1 and UC4
- BB3.3: Temp. and pressure sensing tags → D2
- BB3.4: 3D depth (3D time of flight) → UC3
- BB3.5: Radar → P4 and D3
- BB3.6: Dynamic vision sensor → D3

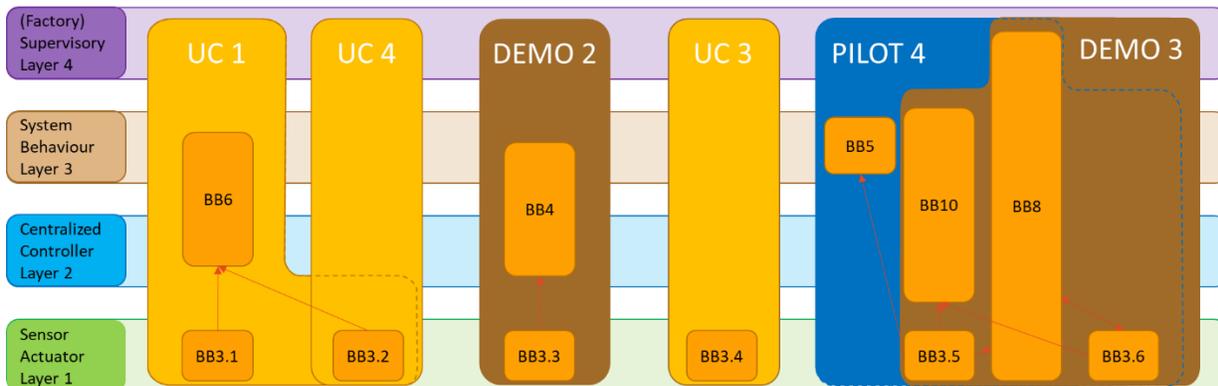


Figure 2. Connection of the BB3 developments with the P/D/UC and other BBs.

The deployment of these sensing solutions in the different P/UC/D targets diverse functions including: on-demand troubleshooting and long-term monitoring; active vibration suppression; part tracking and embedded sensing; object classification and recognition collision avoidance, human tracking, gesture recognition; and obstacle and human/pedestrian detection for path planning.

2.1.2 Addressed ST objectives and KPIs

BB3 contributes mainly to ST2, which concerns the development of the smart Instrumentation Layer gathering and processing visual and sensor information from supplementary instrumentation installed on

the moving parts of the different systems. BB3 developments concern and integrate Layer 1, the Smart Instrumentation Layer, of the IMOCO reference framework architecture, and specific solutions within BB3 provide data to BB4, BB5, BB6, BB9, BB8 and/or BB10, and receive requests from BB8 and the DT platform (contributing therefore to ST1). The information obtained is used to enhance the achievable performance and energy efficiency of the systems, as well as safety and reliability (further details in WP6 and WP7 final deliverables). The demonstration of the sensing solutions in the harsh environments particular of the P/D/UC where these are deployed are partially reported here and further in the respective P/D/UC deliverables (WP6 and WP7).

BB3 also contributes to ST3 (which concerns the development of Hardware and Software motion control building blocks with edge intelligence), i.e. smart Control Layer, by providing data that is used by BBs in this layer such as BB4 (Real-Time Smart-Control Platform), BB6 (Algorithms for condition monitoring and predictive maintenance – one of the KPIs that BB3 addresses) and BB10 (Motion / path planning, collision avoidance and navigation algorithms).

Additionally, in BB3, in order to achieve Adaptive motion drive EMI noise reduction and enable an optional power line communication (PLC) path (as an alternative for wireless) a set of guidelines (further reported in T3.5) were established for the wired components of the sensors.

2.2 Vibration diagnostic sensor (BUT)

2.2.1 Technology overview

Vibration diagnostics is an important method to evaluate the health state of the moving components of the mechatronic systems or rotating machines. It can help to predict the failures of the critical component in the system based on long term monitoring of vibration manifestations and their changes in time. Vibration based diagnostics plays an important role in analysis of bearings' damage, unbalance, loosening of connections in mechanical structures, electrical failures in rotating machinery, wear of moving parts, etc. Precise in-situ measurement of vibration signals represented as acceleration, velocity or displacement values is necessary to provide reliable and information rich data for these diagnostic tasks. It is beneficial to measure the vibrations in all three perpendicular axes in one measurement point on the machine, but usually the most important direction is only in the axis perpendicular to the mounting surface on the analysed component or in the radial direction on the rotating machine. Therefore, a significant amount of the devices is designed as a single-axis sensors.

In the typical industrial applications, a standard industrial-grade single axis accelerometer is mounted on the measured surface and a signal, expressing the input mechanical movement, and represented as an electrical voltage or current, is led through the metal wires to the measuring instrument. This approach needs a lot of effort, strong infrastructure, and engineering. A solution to overcome these difficulties is a wireless sensor with built-in intelligence capable to calculate basic condition indicators of a signal. This type does not need any wiring and infrastructure – it can be only mounted on the surface to be measured and connected to the wireless network and in a while can send an important health data of the machine to the system. The wireless sensors are a good option for short-term measurement (due to the limited power source represented by the rechargeable built-in batteries).

A wireless vibration diagnostic sensor developed within the IMOCO4.E project by Brno University of Technology (BUT) was focused on providing specialized energy consumption optimized wireless sensing node for monitoring and analysis of mechanical manifestations of mechatronic components and systems

through precise measurement of the vibration signals. The wireless sensor is intended to be used in the industrial applications, such as predictive maintenance of the machines, faults diagnostics, early detection of the anomalies and smooth and safe operating of the diagnosed technology. Benefits of using the wireless sensor, compared to the standard industrial one, are following:

- **easy installation** – developed sensor has a set of strong magnets in the bottom of the case – it can be attached to any magnetic surface, which leads to very short installation time. This type of mounting can be used only when no high frequencies measurement is required. Also, a thread hole with the standard M5 size for very tough mounting to the measured object is present at the bottom of the sensor.
- **no wiring** – thanks to the wireless technology and build-in rechargeable battery pack the sensor does not need any wires – neither for communication as well as for power supply. The sensor contains also a built-in wireless charging circuits, which emphasizes the wire-free operation of the unit.
- **small dimension and weight** – despite the number of benefits and built-in functionalities inside the sensor is has relatively small dimensions and weight – comparable to the standard industrial piezoelectric accelerometers.
- **calculation in the node** – the sensor has built-in a powerful microcontroller and contains robust digital signal processing methods. This allows a possibility of the signal preprocessing just in the sensor and not waste the transfer capacity of the communication channel. The sensor can be from this point of view considered as a Computing at the Edge (CatE) device.

Developed sensor helped to solve diagnostic tasks especially in the use case UC1, where novel or advanced systems for commissioning and operating the elevator cabin is to be developed, integrated and tested. In this case, the sensor can support several important activities performed during all phases of the life cycle of the elevator. During commissioning of the drive of the elevator, the sensor can help to fine tune the parameters of the motion control loop in the drive controller to minimize vibrations of the mechanical systems of the elevator (e.g. rope vibrations) to enhance the noise/vibration comfort in the cabin. For this phase, the sensor should operate in mode where current acceleration data (e.g. waveforms) is available for on-line analysis by the technician or for processing inside the control system of the elevator. During operation phase of the lift, the sensor can monitor the vibrations of the cabin and analyse the change in the value of the acceleration over time and with respect to specific frequency components which can indicate progressing of the damage of the critical parts of the elevator mechanical or electrical system components. In this mode, the sensor should analyse vibration data only in specific operating conditions allowing long-term operation with saving energy of the internal battery. The sensor can be used also for on-demand monitoring and diagnostic task when the sensor can be mounted and used in case of troubleshooting of the specific mechatronic component.

Smart Wireless Sensor (SWS) detailed description

Topology of the sensor can be seen in the following block schematic in the Figure 3.

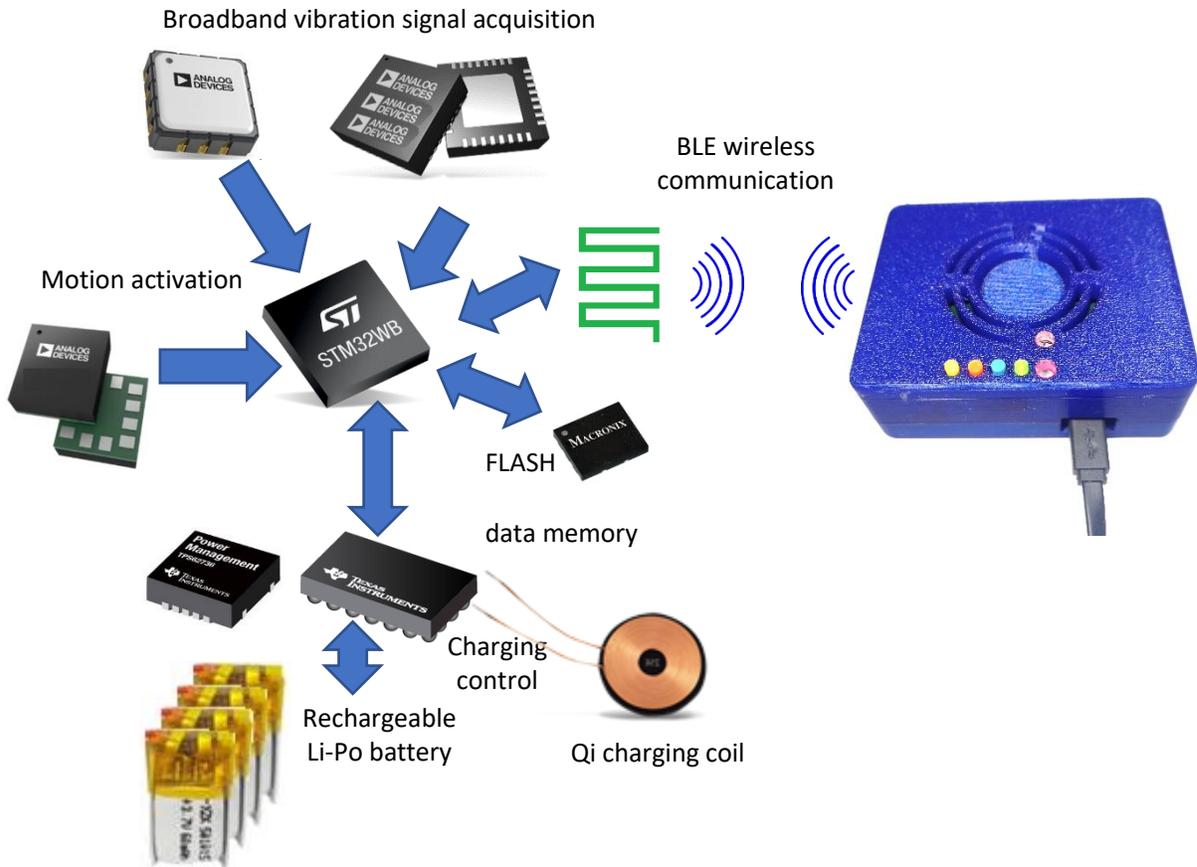


Figure 3. Block topology of a Smart Wireless Sensor.

A core of the sensor is ST Microelectronics 32-bit microcontroller from the STM32WB family. It is a powerful dual-core processor, where one core (ARM Cortex-M4+) serves as a main processor and second core (coprocessor with ARM Cortex-M0 core) ensures wireless communication supporting Bluetooth® Low Energy, Zigbee®, Thread®, and Matter connectivity. The processor ensures communication and data readout from the three specialized MEMS accelerometers manufactured by Analog Devices:

- HF - high frequency, high dynamic and single axis analogue acceleration sensor for precise measurement of the signals with the higher frequencies up to 10 kHz
- LF - low frequency, ultra-low noise, middle dynamic and three axis digital acceleration sensor for basic, multidirectional and high-resolution measurement for the standard signals representing typical mechanical faults of the machines.
- ULP - ultra-low power, low frequency, middle dynamic, three axis digital acceleration sensor for measurement control, confirmation of the correctness of the main measured values, but mainly for waking the system up in case of vibrations presence and sleep down the system in case of no mechanical movement of the sensor.

The output data from high performance accelerometers are used for calculation of basic condition indicators (ISO band vibration [1], effective acceleration value in specific band, specific frequency components effective value, peak values) and the number of indicators can be extended or optimized based on application requirements. Time waveforms from these sensing devices can be also transferred in the “raw” form for processing in any upper layer device requesting time waveform data. Ultra-low power consumption accelerometer is “always-on” for analysis the instantaneous acceleration values for adaptive wake-up of the whole sensing system of the diagnostic sensor. As mentioned in the overall architecture design description the motion activated sensor will command the main controller to perform measurement task depending on the adaptive acceleration threshold settings. This smart power management lead to decrease of the overall consumption of the sensor and allows to reduce the data produced with the sensor only to required or acceptable level for continuous monitoring of the analysed mechatronic component or system.

The SWS is for all the time in the sleep mode with very low power consumption (with the consumption of ca. 32 μ A) checking the input vibration and in case of exceeding the set limits (it means if the input vibrations are higher than set noise level) the SWS is woken up and starts the measurement of the input signal from the acceleration sensors and store them into the FLASH memory. Each record is marked by the real time stamp, so user has a detailed record of the machine behaviour in the time. If the vibration value falls down below uninteresting level, SWS finishes measurement, stores the data into the internal non-volatile memory and fall asleep. As soon as the user needs the stored data, simply requests the sensor and it will send the data through Bluetooth Low Energy through wireless gateway to the user.

SWS contains also sophisticated power circuits – a set of Li-Pol batteries with the total capacity of 200 mAh. Sensor can survive in the sleep mode for almost one year or ca. 14 hours of continuous measurement with this energy storage without being charged. Charging process is managed by the universal Qi charging circuitry and SWS can be charged by any commercial Qi charger useful for any mobile phone. Charging time is ca. 2 hrs.

Proprietary communication protocol is used for the wireless communication between the sensor and a gateway - base station. The communication gateway has also been developed as a part of the IMOCO4.E project. The purpose of the gateway is to:

- charge the sensor using Qi standard and a special charging pad,
- ensure the communication with the sensor using BLE,
- serve as a translation station between PC (through USB communication bus) and the SWS (through BLE).

A photography of the gateway can be seen in the Figure 4.



Figure 4. Photography of the designed wireless gateway.

All important technical parameters and characteristics of the SWS can be found in the following table:

Table 1- Parameters of SWS

MECHANICAL	
Mechanical dimension	ϕ25mm, height 30 mm
Weight	ca. 50g
Package material	3D ABS printed case (prototype stage) Brass case with plastic cap (production stage)
Mounting possibilities	Magnets for quick and reliable installation (indicative measurement) M5 thread hole (precise, long-term measurement)
ELECTRICAL	
Measuring range	DC – 10 kHz (in main axis of the sensor) / ±50 g / 15 bits resolution DC – 1 kHz (all three perpendicular axis) / ±(2-8) g / 20 bits resolution
Measuring quantity	Vibrations acceleration – raw data, condition indicators describing lift health status
Storage capacity	32 MB, built in non-volatile memory, approx. 10 minutes of complete raw data
Operating time	1 year (in sleep mode), 14 hours (continuous measurement)
Charging	Qi standard, approx. 2 hrs
Power source	Built-in LiPo battery pack 3,6 V/200 mAh
Communication	Wireless – BLE (2 Mbps), proprietary protocol

Device mechanical design

Mechanical design of the sensor was inspired with previous designs of the small wired diagnostic sensors developed at BUT, with typical conical shape of the upper part and hex nut shape of the bottom part for tightening the sensor in situation when the sensor is permanently mounted using M5 worm screw and the threaded blind hole at its base. For short-term attachment, there can be used four permanent magnets in case of magnetic mounting surface availability. The overall size of the sensor is 25 mm in diameter and 30 mm in height. The 3D model of the sensor housing in different views is shown on the Figure 5.

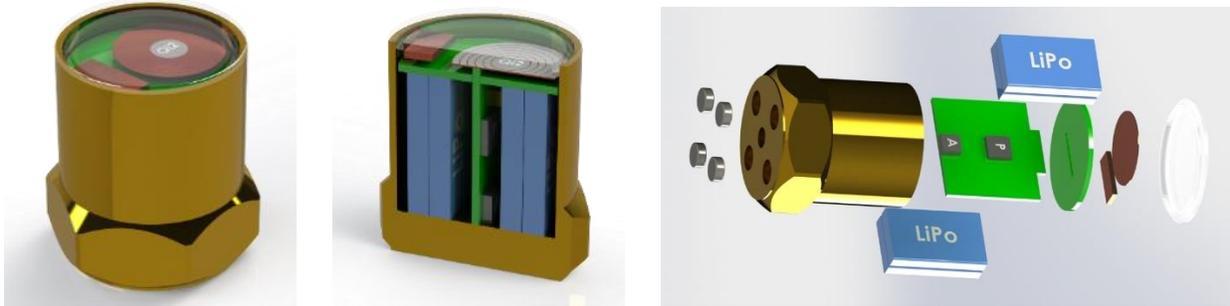


Figure 5. Architecture of the vibration diagnostic sensor with emphasized main components.

[Link to hardware and software catalogue and components integration](#)

The vibration diagnostic sensor development is linked to the hardware catalogue item no. HW-022 (Wireless vibration sensor with motion/vibration activation) and is also connected with software catalogue item no. SW-035 (Condition monitoring package for vibration monitoring of a lift cabin and diagnostics of a lift drive), where software components for condition monitoring of specific mechatronic components (e.g. lift cabin and lift drive) is defined and developed in WP5. These software components are expected to be available in different programming languages depending on the deployment target (sensing device in layer 1, or control/processing device in layer 2). In case of the vibration diagnostic sensor, some of the developed layer 1 software components programmed in C are deployed in the main controller of the sensor.

2.2.2 Implementation aspects

The electrical schematic as well as the printed circuit board have been developed during the project. PCB is designed as a 4-layers with small components due to fit the overall topology into the tiny housing. The sensor contains two PCBs:

- round top PCB with power management circuits, Qi charger, wireless microstrip antenna and service connector for firmware update and rechargeable LiPo batteries,
- square bottom PCB with all three acceleration sensors, microcontroller, and internal memory.

Rendered assembly set of the PCBs from design software can be seen in the Figure 6 on the left and prototype set with assembled component can be seen in the same figure in the middle.



Figure 6. Render of the PCBs (left), real prototype of the assembled sensor (middle) and sensor encapsulated in the 3D printed housing (right).

The housing for SWS has been designed and manufactured using 3D printing technology from ABS material. This technology is sufficient for prototyping manufacturing, but in case of the final production sensors, brass or aluminium housing will be manufactured. Sensor is intended to be in IP68 housing due to harsh industrial environment. The top cap of the sensor should still be from some plastic material even in the industrial-grade housing due to electromagnetic transfer path for wireless communication and also for Qi charging. Photo of the 3D printed housing can be seen in the Figure 6 on the right.

Since the PCB shown on the figure above is a prototype version of the sensor, there should be some small changes in the next revision – the bugs during design as well as the improvements, which have been found during the development and implementation of the advanced diagnostic features.

Complete functionality of the sensor is also done by the internal firmware, which was completely designed and coded in the C language.

2.2.3 Results

The sensor has been tested in the laboratory on the vibration exciter and using the artificial excitation. The data captured by the sensor during knocking on the table, stored into the internal memory, transferred through Bluetooth Low Energy, received by the gateway, and decoded and displayed by the LabVIEW application, can be seen in the Figure 7 on the left side. All three acceleration sensors (7 curves) can be seen in the figure. The most interesting are two of them – white curve expressing the high frequency accelerometer, while the blue one representing the low frequency sensing element. There can be seen a clear difference between frequency range of both accelerometers in the figure (in the size of the high frequency and high energy peaks). Sine wave with the frequency of 20 Hz and the amplitude of 200 mg generated by the vibration exciter and captured by the SWS can be seen in the same figure on the right-hand side.

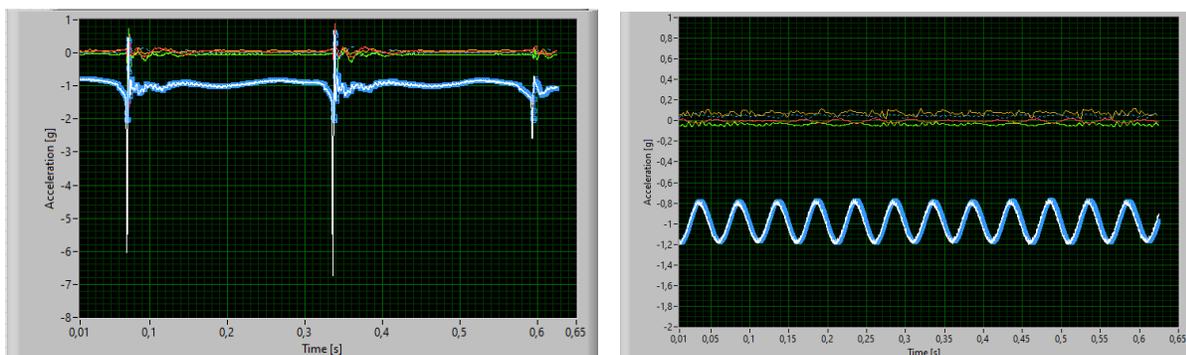


Figure 7. Acceleration impulse signal and sinus waveform captured by the SWS.

Testing of the SWS in the practical harsh industrial environment has been carried out beside the laboratory testing as well. Vibration measurement of the lift cabin has been done at WEG company within UC1 inside the lift testing tower. The sensor has been placed on the cantilever bellow the lift cabin and several test runs have been acquired. A photo of a lift propulsion, a cabin cantilever, the SWS and the wireless gateway during tests can be seen in the Figure 8.

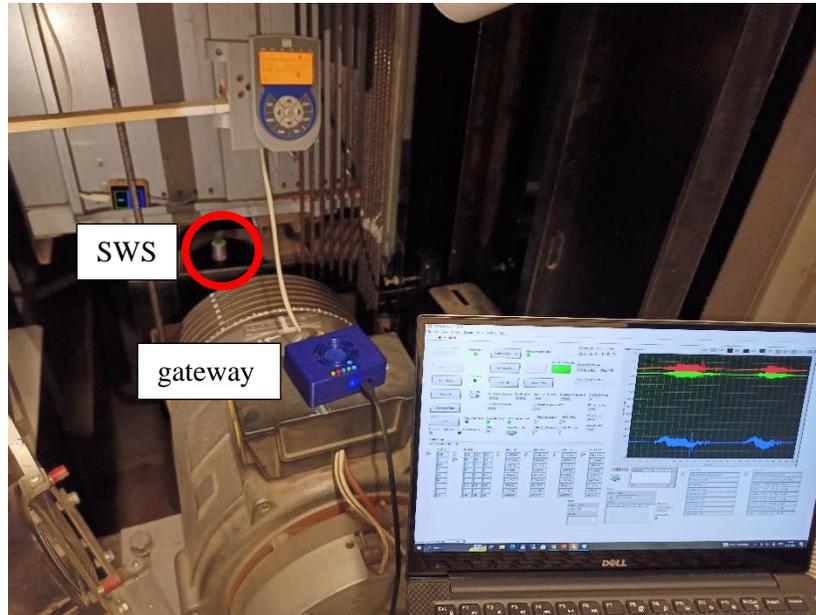


Figure 8. Photography inside the lift testing tower with the SWS attached to the lift cantilever, the wireless gateway and a PC running the application for data visualization.

Several test runs of the lift have been performed – with different lengths (from the ground floor to the 1st floor, from the 2nd floor to the 3rd floor), different ride directions (upward or downward direction) and with the different drive conditions (pure motor control algorithm, motor control with additional noise). Each test run has a duration of ca. 50 seconds. An example of the recorded data for downward direction from the 2nd floor to the ground floor with the stop at the 1st floor and with no additional noise can be seen in Figure 9. Please note, that only the signal from LF low frequency tri-axis accelerometer is shown (Y axis is parallel to the lift cabin ride direction). A ride with the same conditions, but with artificially injected noise to the motor control, can be seen in the following Figure 10.

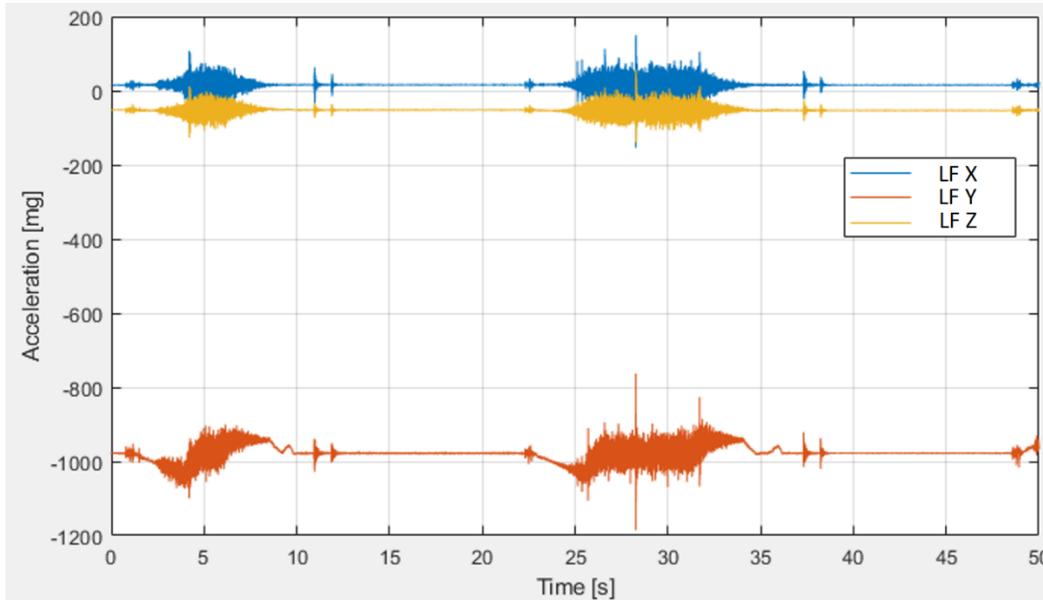


Figure 9. Data from the lift right in the upward direction with no noise.

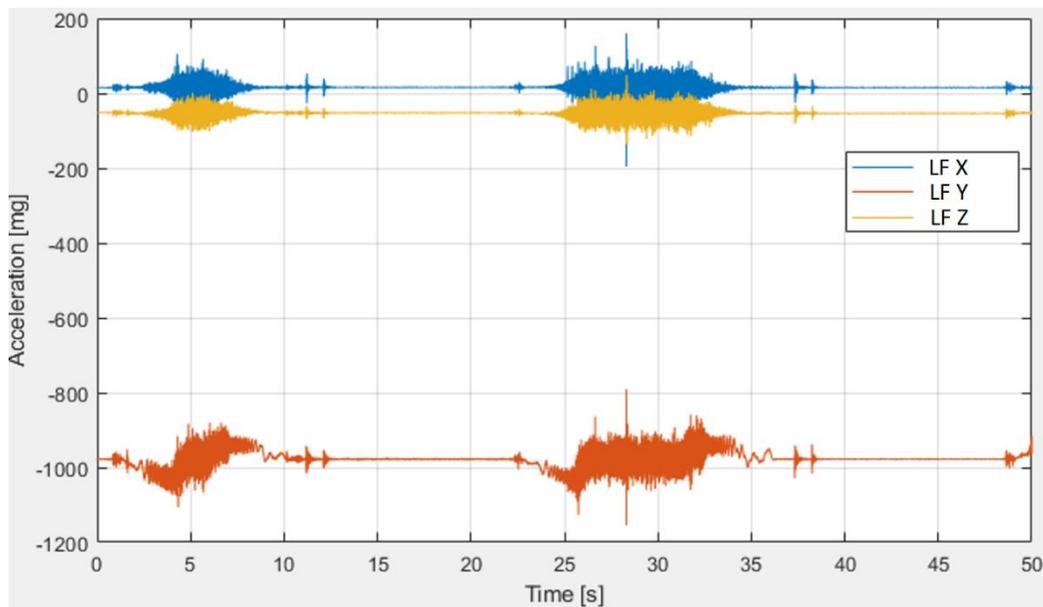


Figure 10. Data from the lift right in the upward direction with additional noise.

Since the acceleration value produced by the lift is quite small, the high frequency and high range accelerometer produces the signal with relatively low signal to noise ratio (as it can be seen in the Figure 11 compared to the low frequency sensor). But the advantage of this high frequency accelerometer will be seen in case of presence the high frequency and high energy peaks, where the low frequency sensing element will not be able to capture such a fast signal changes.

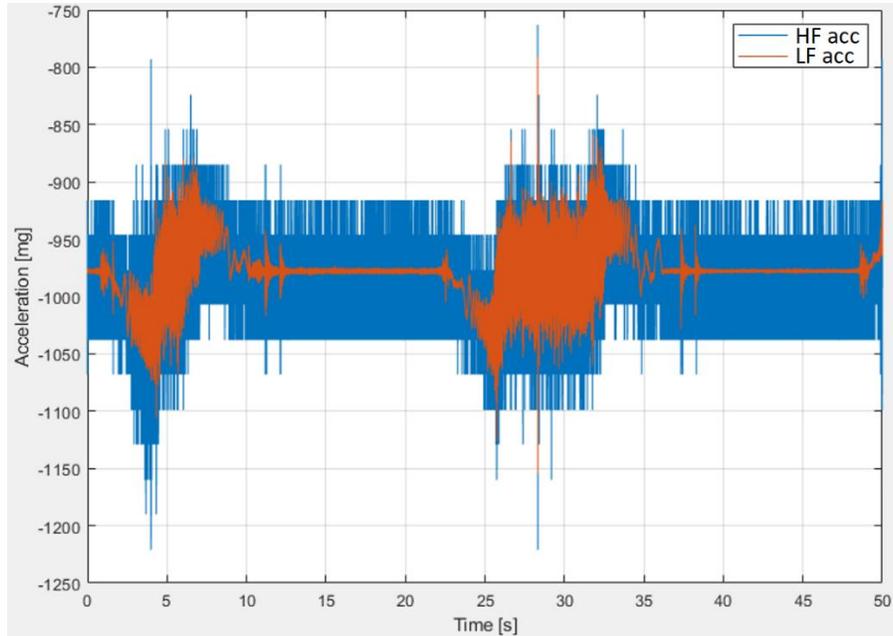


Figure 11. Comparison of the low frequency and high frequency sensing elements.

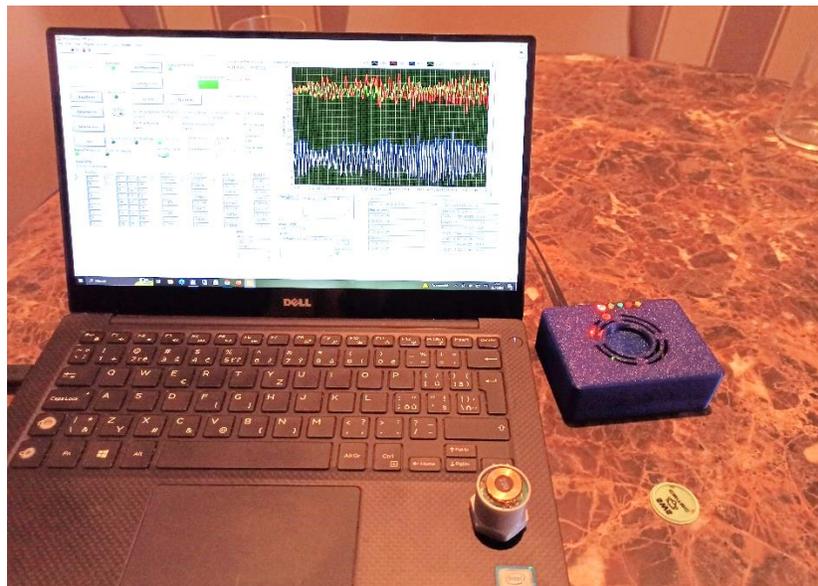


Figure 12. Photography of the SWS, the gateway and LabVIEW visualization application in PC.

Exploitation plan

BUT is not planning to exploit the sensor by itself. It is expected that industrial project partners will utilize the result of the development, e.g. WEG company within their electrical drive control system to support vibration diagnostics of the elevator cabin and help with commissioning of the drive. BUT is planning to valorise the knowledge gained during the development of the sensor in future research projects, for teaching

purposes and in research of doctoral students enrolled at the Faculty of Electrical Engineering and Communications of Brno University of Technology.

2.2.4 IMOCO4.E Requirements

Table 2- Requirements that apply to vibration diagnostic sensor development and its deployment in UC 1.

ID	Requirement	Result	Comments & Rationales
Interfaces and connectivity			
R051-D2.3-L1	Wireless connectivity of sensors for mechanical manifestations acquisition.	PASS	BLE communication implemented.
R106-D2.3-B3	Wireless communication interface for sensors not located in the closed vicinity of centralised controller.	PASS	BLE communication implemented.
Req-D3.1-L1-D2-hw	Sensors must have a reader/ controller connected to upper layers (through BB1 or BB4) by USB or Ethernet	PASS	Wireless gateway with USB connection developed.
Req-D3.2-L1-hw-01	Devices for vibration sensing should have wireless interface, e.g., BLE.	PASS	BLE communication implemented.
Req-D3.2-B1	Sensors could be connected to other devices using wireless interfaces.	PASS	BLE communication implemented.
R066-D2.3-L1	Implemented sensors communication interface/protocol allows simultaneous connection of multiple devices.	PASS	BLE communication implemented with different node names.
R107-D2.3-B3	Low energy wireless communication interface for sensors with at least 500 kbit/s burst data rate and operating range with at least a few tens of meters. Network star topology is preferred.	IN PROGRESS	Communication protocol needs small optimization to achieve desired speed.
Performance			
Req-D3.2-L1-hw-04	The vibration sensing device should have measurement resolution of at least 0.01 m/s ² , at least measurement range +/-4 g and maximal supported acceleration above 20 g.	PASS	Resolution of the SWS is 0.01 m/s ² based on LF sensor specification. Range is ±50 g for HF sensor, ±8 g for LF sensor.
Req-D3.2-L1-hw-05	Device for vibration sensing should be capable of achieving minimum sampling period 1 ms.	PASS	Sampling rate is 4 kHz for LF sensor.
Req-D3.2-L1-hw-06	Device for vibration sensing should be able to activate measurement by motion and/or vibrations.	PASS	Motion activation by ultra-low power MEMS device is implemented.
R060-D2.3-L1-hw-DAT	Sensors monitoring the condition of an asset should provide real-time measurements without disruptions	PASS	Real-time measurement implemented.
R139-D2.3-B6	Condition monitoring and predictive maintenance are capable of running on different layers depending on computational complexity and memory requirements.	PASS	Condition monitoring is done directly in the SWS, but can be performed outside the sensor from real-time data provided.

Req-D3.2-L1-hw-07	Device for vibration sensing could have simple controls (buttons) on the device for basic settings (e.g., on/off, reset).	PASS	SWS can be turned ON/OFF by a magnetic switch.
Compatibility (interoperability, co-existence)			
Req-D3.2-B1	Sensors and actuators used together must have low interference with each other. Wireless sensors require dedicated frequency bands or interoperable protocols.	PASS	BLE communication uses TDMA method.
Req-D3.2-L1-hw-09	Transmitting hardware in device for vibration sensing should be compatible with receiver based on ESP32 system.	PASS	SWS uses standardized BLE protocol.
Req-D3.2-L1-hw-10	Transmitting hardware in device for vibration sensing should be compatible BLE version 5.0 or newer.	PASS	SWS uses BLE 5.2 protocol stack.
Usability (operability)			
Req-D3.2-L1-hw-11	Device for vibration sensing should be able to record at least 1M of time data points for further data processing.	PASS	SWS has 32MB memory implemented allowing to store more than 2M data points.
R055-D2.3-L1	Sensors provide sufficient diagnostic information to evaluate their weaknesses or damages.	IN PROGRESS	Advanced CIs are still under the development under WP5.
R112-D2.3-B3	Power supply for expected lifetime operation integrated with the sensor, e.g. battery power, wireless power or energy harvesting, depending on the device's power requirements.	PASS	Li-Pol battery with the capacity of 200 mAh built-inside the SWS.
Req-D3.2-L1-hw-12	Device for vibration sensing could be operatable in industrial environment conditions, according to EN 50178.	NOT STARTED	Expected for the final sensor with brass case.
Req-D3.2-L1-hw-13	Device for vibration sensing would have ingress protection of level IP 54.	NOT STARTED	IP67 for final sensor with brass case is expected.
Reliability (fault tolerance, availability)			
Req-D3.2-B1	Sensors and actuators should be self-calibrated, or factory calibrated.	NOT STARTED	Factory calibration is expected for the final production.
Req-D3.2-L1-hw-17	Device for vibration sensing would be able to operate for at least 10 years.	PASS	Lifetime of the SWS is expected to be more than 10 years by design.
Req-D3.2-L1-hw-18	Device for vibration sensing could perform internal self-calibration and self-diagnostics to be able to detect internal fault of the device and report it to the upper layer device.	IN PROGRESS	Self-diagnostics is still under development.
Req-D3.2-B1	Sensors and actuators should be self-calibrated, or factory calibrated.	PASS	Factory calibration is expected for the final production.
Req-D3.2-B1	Include redundant sensors to ensure fault tolerance.	PASS	SWS contains three vibration sensors.

R113-D2.3-B3	Sensors analyse their internal performance parameters to evaluate the reliability of output data and report this information.	IN PROGRESS	SWS contains three vibration sensors enabling reliability analysis which is under development.
Maintainability (modularity, analyzability, testability)			
Req-D3.2-B1	The device’s firmware should be able to be updated using the defined interfaces.	PASS	FW update is possible through wired JTAG interface, or later Ota update.
Req-D3.2-B1	Continuous monitoring of the hardware to find faulty behaviours.	NOT STARTED	Functionality planned for future development.
Portability (adaptability, replaceability)			
Req-D3.2-L1-hw-19	Device for vibration sensing should be portable to be able to mount it freely on the lift cabin or use it during routine-based inspection, e.g., with weight of 100 grams or lower.	PASS	Weight is < 50 g, magnetic/screw mounting possibility.
Req-D3.2-L1-hw-20	Device for vibration sensing would be able to perform firmware updates through wireless connection or galvanic interface (e.g., USB).	PASS	FW update is possible through wired JTAG interface, or later Ota update.
Functional suitability			
R148-D2.3-B6-sw	Predictive maintenance software components should create real-time notifications about anticipated malfunctions of monitored assets.	IN PROGRESS	SWS calculates basic CIs to evaluate the state of the monitored assets. Advanced CIs calculation under development.

2.2.5 Capabilities and Limitations

Dimensions of the sensor are small enough to mount it on the standard place instead of the standard accelerometer. Wireless technology used for the data exchange is a benefit, which does not require cabling and other necessary and expensive infrastructure. Modularity of the complete system is also a benefit, which allows easy implementation of other customer-required functionalities.

Since the device is completely wireless and the charging is solved by the Qi wireless charging as well, this can be considered as a weakness. Sensor must be relatively close to the charging coil for efficient and successful charging – the distance between sensor’s surface and charging pad should be max. 3 mm. which can be in some cases hardly feasible.

2.2.6 Customizations and Adaptations

There is probably no need to upgrade a hardware solution thanks to the robust design and powerful hardware with an excess of performance. Any user requirement can be easily done only by implementation of the new functionality in the firmware.

2.2.7 Methodology and Toolchains

Development and design of the SWS have been done using following tools:

- Altium Designer – PCB design and PCB rendering
- SolidWorks – housing design and preliminary render of the instrument
- STM32Cube IDE – development of the internal firmware of the sensor as well as gateway
- LabVIEW – PC application for sensor setup, data acquisition, decoding the vibration information and exporting the results.

2.2.8 References

[1] ISO 20816-1:2016 - Mechanical vibration - Measurement and evaluation of machine vibration. ISO Standard, 2016. 34 pages.

2.3 Vibration and motion sensor (UWB)

2.3.1 Technology overview

Introduction and general Description of Sensor

Monitoring and analysing motion and vibrations is a key source of information in many application types across various branches. Having reliable motion data is a key for understanding processes, changes or malfunctions in the observed systems.

IMU (Inertial Measurement Unit) sensors typically consist of 3 independent sensors – accelerometer, gyroscope and magnetometer – and are a type of sensor technology used to directly measure the changes in motion (acceleration), angular rotation and orientation in external magnetic field. Further important derived values are tilt, position and velocity of an object in motion.

Wireless IMU sensors are a new development in IMU technology that allows for greater flexibility, portability and convenience in their use. Unlike traditional IMUs, which require a physical connection to the object they are measuring, wireless IMUs use radio waves to transmit data to a receiver. They can be easily attached to an object without the need for cables or wires, which makes them ideal for use in a variety of settings (e.g. rotating machine parts, inaccessible spots, places that are hard or too expensive to wire, etc.). Additionally, they can be used in environments where traditional IMUs would be impractical.

Wireless IMUs have a wide range of applications, including robotics, virtual reality, and motion capture. They are also commonly used in sports performance analysis:

1. **Industrial machinery:** Wireless accelerometers can be attached to machinery to measure vibration and motion. The data from the sensors can be used for failure detection or prediction. In more advanced systems they can be used by the control system to adjust the operating parameters of the machine, such as the speed or torque of a motor, to maintain optimal performance and prevent damage. And even better - in a closed loop the data can be used for active vibration suppression.
2. **Vehicle suspension:** Accelerometers can be used in a closed loop control system for vehicle suspension. By measuring the acceleration and tilt of the vehicle in real-time, the controller can adjust the suspension to maintain a smooth ride and prevent excessive bouncing or instability.
3. **Robotics:** Wireless accelerometers can be integrated into robotic systems to provide feedback on the motion and position of the robot. This data can be used to adjust the robot's movements and maintain precise control over its actions.
4. **Structural health monitoring:** Wireless accelerometers can be used to monitor the structural health of buildings, bridges, and other structures. By measuring the vibration and movement of

these structures, the sensors can provide early warning of potential damage or failure, allowing for preventative maintenance and repairs.

5. **Sports performance analysis and physiotherapy:** Wireless accelerometers can be used to measure the movements and forces exerted by athletes during training and competition (or patients under treatment). This data can be used to optimize training programs, identify areas for improvement or treatment, and help to bring the right decisions.

The section of Department of Cybernetics of the UWB that is participating in the IMOCO4.E project is strongly involved in process control, industrial automation and robotics topics. In many projects related to that topics, we have met with request for motion data. Applications requiring offline data (measure & process scenarios) could be served with existing industrial sensors. However, we were lacking wireless motion sensor with fast, low latency data delivery for **loopback control**. Therefore, we have decided to build on the results of the previous I-MECH project and develop rugged, industrial-grade sensor providing real-time motion data for the loopback control. Additionally, the sensor can work in so called batch mode, providing on-demand measurement with higher sample rate for offline data analysis. In this mode it works as a replacement of the existing wireless accelerometer sensors.

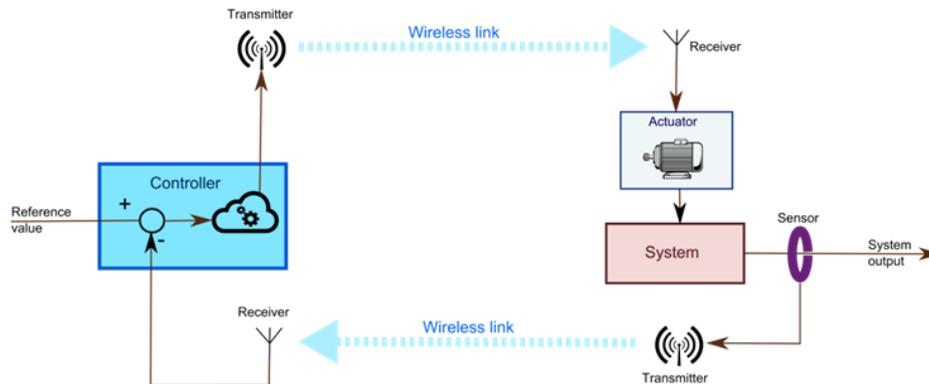


Figure 13. Control loopback realized using wireless link. Links from the controller to the actuator at the controlled system as well as from the sensor installed at the controlled system to the controller are wireless. UWB has demonstrated, that sufficiently reliable radio link can be established to control even highly unstable and fast systems.

Sensor features and characteristics

The developed sensor (a product name Imagination sensor has been given to it – originating from words Inertial, Measurement, Notation and a bit of Magic) is rugged MEMS based wireless IMU (Inertial Measurement Unit) vibration and motion sensor. It provides full 9 DoF measurements (3-axis accelerometer, gyroscope and magnetometer data) measured on demand in two performance modes:

- **synchronous mode:** sampling, measurement and data transfer is synchronous with control period of a remote controller. This mode is suitable for in the loop control – like vibration damping, etc.
- **batch mode:** the measured data are queued in a memory FIFO and transferred on the fly to the controller. This mode is targeted for off-line analysis and due to the batch transfer allows higher sampling rates.

To allow use in harsh industrial environments, the sensor is sealed in silicone housing with integrated LiPo battery and wireless charging capability. The RETIS [1] communication protocol operating in 433 MHz or 2.4 GHz band takes care of reliable low latency wireless data delivery.



Figure 14. Imaginotion sensor prototype

Sensor data can be accessed either directly using the RETIS wireless communication platform or via an intermediate gateway that exposes sensor data over USB virtual COM port or RS-485.

Table 3- Characteristics of the vibration and motion sensor

Dimensions:	55 x 55 x 35 mm
Power:	Integrated rechargeable LiPo cell 1000 mAh
Charging:	Wireless, Qi standard
Protection:	IP68
Connectivity (direct):	433 MHz / 2.4 GHz, RETIS
Connectivity (gateway):	USB, RS485
IMU unit:	TDK-Invensense ICM-20948
MCU & RF transceiver:	EFR32MG12P
Accelerometer:	3-axis Adjustable FSR (Full Scale Range) ± 2 g, ± 4 g, ± 8 g, ± 16 g, 16-bit resolution Max. sampling rate 1 kHz / 4.5 kHz (synchronous mode / batch mode)
Gyroscope:	3-axis Adjustable FSR ± 250 dps, ± 500 dps, ± 1000 dps, ± 2000 dps, 16-bit resolution Max. sampling rate 1 kHz / 9 kHz
Magnetometer:	3-axis ± 4900 μ T, 16-bit resolution Max. sampling rate 100 Hz / 100 Hz

[Link to Hardware- and Software-Catalogue and Components-Integration](#)

HW-Catalogue links:

- HW-019 – Imaginotion Wireless Motion Sensor

SW-Catalogue links:

- None

2.3.2 Implementation aspects

Motion sensing

The Imagination sensor is a MEMS based wireless IMU (Inertial Measurement Unit) vibration and motion sensor built on TDK/Invensense ICM-20948 chip [2]. It offers full 9 DoF data (3-axis accelerometer, gyroscope, and magnetometer) that can be measured on-demand or periodically. Range and performance is given by the chip used [3].

Data transfer

The wireless communication used to exchange data between the sensor and stationary gateway or controller uses and further extends proprietary communication platform called RETIS. The technology is especially suitable for critical loopback control applications and allows to reliably transmit real-time data with superb latency and jitter. On the other hand, during inactivity periods, it allows to put the system into deep sleep mode and wake up only occasionally to synchronize with the controller. This approach allows to save energy and stay in operation for a very long time.

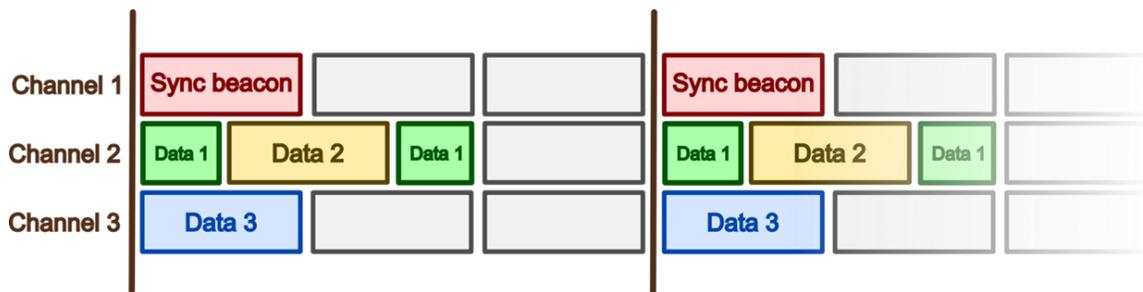


Figure 15. RETIS principle of operation. The protocol can simultaneously operate on multiple RF channels or even bands. TFDMA (Time-Frequency Division Multiple Access) principle is used – base transmission cycle is divided into communication slots, which are assigned for certain nodes -> the communication is deterministic, with predictable delivery time and minimal jitter. RF nodes have right to transmit during the assigned slots only. Allowed is only one packet per slot, however multiple messages can be encapsulated in the packet. Master node distributes clock signal so that accurate global clock is available at all nodes.

The communication runs in free 433 MHz band. Using sub-GHz band is suitable for complex environments as longer waves can better handle obstacles in the communication path. Also the range is longer in comparison to commonly used communication protocols at 2.4 GHz (like BLE, WiFi, IEEE 802.15.4, ...). To implement the RF functionality a RF MCU EFR32MG12P from Silicon Labs was used [4], [5]. The same MCU is used to process commands from the controller and preprocess data from the IMU unit.

Housing

Housing for the sensor was designed using 3D CAD design tools (SolidWorks). For the final product it is planned to seal the complete electronics (PCB, antenna, battery, wireless charging coil) into a capsule made of additive silicone – a two-component material that cures after pouring into the mold. Using this type of housing will create the best water and dust protection and is suitable even for the most demanding applications.

For development and prototyping, 3D printed capsules of the same shape were used.

Power & wireless charging

When considering power source for the sensor we have opted for single Lion cell accumulator. Their nominal voltage (3.7V) suits well the ICs used. Nowadays there are lots of possible variants of these cells. As we wanted to keep the sensor capsule reasonably small, we have to limit the cell size. Cells of the suitable dimensions are about 1000 mAh in capacity.

Another question was how to recharge the depleted batteries. A charging connector could be built in. However, this solution would decrease the sensor's IP protection level. Therefore, we have decided for wireless charging. In this case, the sensor electronics can be completely sealed in a water and dust proof capsule. The existing wireless charging standard Qi was selected [6]. It fits well the sensor needs and wireless chargers of this standard are readily available. To control this functionality a BQ51050 IC from Texas Instruments was used [7].

2.3.3 Results

The sensors have been tested in laboratory environment as well as in real-world applications.

Laboratory tests

The laboratory tests validated sensor acquisition performance on a vibration bed scanning through wide spectrum of vibration frequencies. The sensor performed according to the values specified in the datasheet.

Further tests have targeted RF link performance. They have brought surprisingly good results. The sensor range in a free space (direct sight) was around 100 m and in obstructed space (production halls, office buildings, lift shaft) ranged around 40 m even behind corners or closed doors. The link showed minimal disruption even when RF noise was injected in nearby RF channels. Few packets were lost when wide band noise was present (power motor switch on/off), but the communication was always recovered, and the PLR (packet loss ratio) was very low.

Wireless charging of the sensors works reliably with available phone Qi chargers. Tested were types from Samsung, Apple and several no-named products.

Real-world applications

Recently the sensors were tested in 2 controlled real-world environments:

- Company manufacturing gears was testing the sensors in a bus carrying their gear (the name of the company as well as test details are confidential). The sensor operated continuously for several hours providing real-time stream of data.
- The sensor was used at WEG facilities to monitor vibrations of lift cabin according to UC1. Vibrations were measured during normal lift operation as well as with intentional harmonic noise introduced by the motor inverter. Data in batch mode were correctly acquired and their analysis will be presented in scope of UC1.



Figure 16. Imagination measurement set - sensor and DIN-rail mounted gateway with possibility to trigger measurement with digital signal.



Figure 17. Imagination sensor mounted on a lift cabin in test shaft in WEG. Our sensor (black) is mounted next to the commercial reference sensor (orange/black).

2.3.4 IMOCO4.E Requirements

Table 4 - Requirements for the vibration and motion sensor

Requirement	Description	Status
R041-D2.3	Electromagnetic compatibility: emission and immunity requirements. All electric and electronic equipment must satisfy the EMC directives, as applicable to the products considered: 2014/30/E.U. (EMC directive): IEC/EN 55011/61326	Certification not started yet
R042-D2.3	Radio equipment, Radio frequency emissions All products that incorporate wireless and/or radio-related functions must satisfy the Radio Equipment Directive (2014/53/E.U.).	Certification not started yet
R069-D2.3-L1	Implemented sensors communication interface/protocol allows simultaneous connection of multiple devices.	PASSED
R109-D2.3-B3	Wireless communication interface for sensors not located in the closed vicinity of centralised controller.	PASSED
R110-D2.3-B3	Low energy wireless communication interface for sensors with at least 500 kbit/s burst data rate and operating range with at least a few tens of meters. Network star topology is preferred.	PASSED
R117-D2.3-B3	Power supply for expected lifetime operation integrated with the sensor, e.g. battery power, wireless power or energy harvesting, depending on the device’s power requirements.	PASSED
Req-D3.1-L1-D2-hw	Sensors must have a reader/ controller connected to upper layers (through BB1 or BB4) by USB or Ethernet	PASSED
Req-D3.2-L1-hw-01	Devices for vibration sensing should have wireless interface, e.g., BLE.	PASSED
Req-D3.2-L1-hw-04	The vibration sensing device should have measurement resolution of at least 0.01 m/s ² , at least measurement range +/-4 g and maximal supported acceleration above 20 g.	PASSED
Req-D3.2-L1-hw-05	Device for vibration sensing should be capable of achieving minimum sampling period 1 ms.	PASSED
Req-D3.2-L1-hw-08	Device for motion sensing must provide data with lowest possible latency (below 500us) to allow feedback control.	PASSED
Req-D3.2-L1-hw-xx	Device for vibration sensing could be operable in industrial environment conditions, according to EN 50178.	PASSED
Req-D3.2-L1-hw-xx	Device for motion sensing should have ingress protection of level IP 68.	PASSED
Req-D3.2-L1-hw-19	Device for vibration sensing should be portable to be able to mount it freely on the lift cabin or use it during routine-based inspection, e.g., with weight of 100 grams or lower.	PASSED
Req-D3.2-L1-hw-21	Device for motion sensing must support wireless charging (to allow molding with IP68)	PASSED
Req-D3.2-B1	Sensors and actuators used together must have low interference with each other. Wireless sensors require dedicated frequency bands or interoperable protocols.	PASSED
Req-D3.2-B1	Sensors and actuators should be self-calibrated, or factory calibrated.	PASSED

2.3.5 Capabilities and Limitations

The Imagination IMU sensors work according to the description in above paragraphs. Their highlight is the possibility to provide real-time IMU data with extremely low latency and jitter and therefore allow closed loopback control over wireless link. Currently there is no comparable solution in the market.

Additionally, they allow standard buffered measurement and recording of IMU and vibration data at higher frequencies for later analysis and processing. In this case they can be used as an alternative of an existing solution.

As a possible weakness we see the limited life of the integrated battery which can supply the device with energy only for a limited time. The suitability of the wireless solution is always a matter of the specific application and various aspects need to be considered. In stand-by mode the sensor is consuming only a few microamps and is able wait for commands for several years. However, in the full performance mode when the IMU unit is measuring with high sampling rate and the RF transceiver is simultaneously transmitting data, the power consumption can rise up to 27 mA, which allows life time of hours or maximally days. Afterwards, the device must be re-charged.

2.3.6 Customizations and Adaptations

Customization and modifications of the provided solution are possible. The sensor consists of relatively easily separable modules that can be replaced or modified:

- For the first sensor implementation we have selected the golden middle path between price and quality of the sensing element (ICM-20948). If more accurate sensing is required, the sensor IC can be replaced with higher performance ICs leading to increased price of the sensor.
- The wireless communication platform RETIS provides many possibilities of customization. Eg. transmission in another RF band (2.4 GHz). Or redundant simultaneous transmission in multiple RF channels to increase the reliability of the transmission, adjusting parameters of the RF PHY layer (modulation, bitrate, whitening).
- Modifications to the sensor capsule - various shapes and sizes can be utilized. Eg. in case when the size of the sensor does not matter, but the lifetime of the sensor is crucial, bigger capsule integrating high-capacity battery can be designed.

2.3.7 Methodology and Toolchains

During the design phase the following tools have been used:

- SolidWorks – to design the mechanical dimensions and shape of the sensor capsule
- Altium Designer – to design the sensor electrical schematics and PCB layout
- Silicon Labs Simplicity Studio – sensor firmware development
- BQ51050 Development Kit – to analyse and design wireless charging circuitry

In deployment phase, the solution integration is quite straightforward and no specific third-party tools are required. The RETIS gateway establishes connection to the sensors and provides data to its client over USB virtual COM port or over RS-485. The data messages are simply structured and can be parsed by custom application or by provided Windows application allowing data logging and real-time visualization.

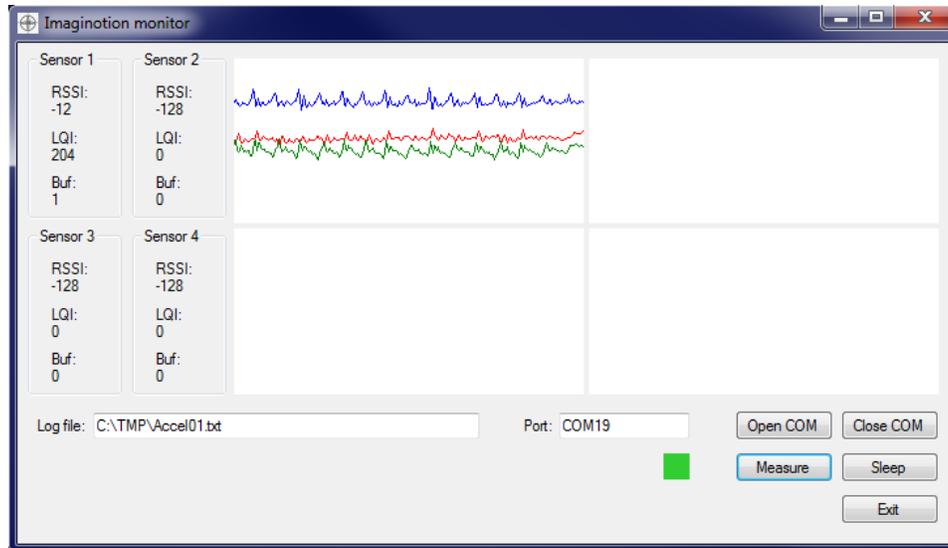


Figure 18. SimpleAccelStreamViewer application used to visualize and log data from Imagination sensors

2.3.8 References

- [1] R. Čečil, V. Šetka, D. Tolar, A. Sikora, "RETIS – Real-Time Sensitive Wireless Communication Solution for Industrial Control Applications", 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), September 2020
- [2] ICM-20948 - World's Lowest Power 9-Axis MEMS MotionTracking™ Device datasheet, revision 1.3, TDK Corporation, 6.2.2017
- [3] STM32WB55xx, STM32WB35xx Multiprotocol wireless 32-bit MCU Arm®-based Cortex®-M4 with FPU, Bluetooth® 5.3 and 802.15.4 radio solution [online]. Available: <https://www.st.com/resource/en/datasheet/stm32wb55cc.pdf>
- [4] EFR32MG12 Gecko Multi-Protocol Wireless SoC Family Data Sheet, revision 1.8, Silicon Labs, June 2022, <https://www.silabs.com/documents/public/data-sheets/efr32mg12-datasheet.pdf>
- [5] EFR32xG12 Wireless Gecko Reference Manual, revision 1.3, Silicon Labs, July 2022, <https://www.silabs.com/documents/public/reference-manuals/efr32xg12-rm.pdf>
- [6] Qi v1.3 specification, Wireless Power Consortium, January 2021, <https://www.wirelesspowerconsortium.com/data/downloadables/3/3/2/3/qi-v13-public.zip>
- [7] BQ5105xB High-Efficiency Qi v1.2-Compliant Wireless Power Receiver and Battery Charger, Texas Instruments, June 2017, <https://www.ti.com/lit/ds/symlink/bq51050b.pdf?ts=1707989694373>

2.4 Overmolded temperature and pressure sensors (INL/ECS)

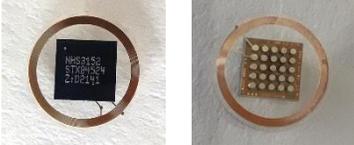
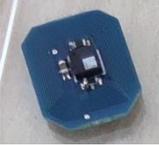
2.4.1 Technology overview

Sensor description – Introduction

This solution focuses on smart sensing tags that can be overmolded (in Demonstrator 2) and incorporated into injection molded parts. The NFC (Near Field Communication) tag/sensor is therefore a wireless sensor which is self-powered: the sensor is powered when in proximity of an NFC reader (wireless interface), delivering an NFC message to it. The NFC tags are meant to provide temperature and ambient pressure sensing capabilities, and traceability to the part. Therefore, the parameters that can be read with the smart NFC sensor are, besides the part ID, the temperature and the ambient atmospheric pressure (air pressure). The tags do not feature parameterization/adjustment or calibration capabilities. The small form-factor enables integration into injection-molded parts, making them a versatile option, highly convenient and non-intrusive solution for precise monitoring of temperature and pressure levels. The main challenges are the design – the trade-off between miniaturization and performance (namely readout distance).

Two parallel approaches were pursued (Table 5): a tag comprising a microfabricated antenna – microfabricated tag, and a PCB-based tag. The microfabricated tags comprise an NFC chip with integrated temperature sensor assembled with a thin-film microfabricated antenna (5.1mm Ø and approx. 0.5mm-thick before encapsulation; and minimum dimensions of 6.2mm Ø and 1.5mm-thick after encapsulation). The goal of this approach was to integrate a thin-film pressure sensor (either capacitive or resistive) in the same microfabrication process as the antenna. The PCB-based approach uses the same NFC IC, but on PCB-based antennas with the commercial pressure sensor IC (either on the same or opposite side of the NFC chip).

Table 5. List of types of sensors/tags developed.

Tag type	Pre-encapsulation: picture, dimensions	Post-encapsulation: picture, dimensions
Microfabricated tag	 <p>5.1mm Ø × 500µm</p>	 <p>6.2mm Ø × 1.5mm 9.2mm Ø × 3.1mm</p>
Single side PCB tag	 <p>8.7x8.8x1.5mm³</p>	 <p>10.2x10.2x4.3mm³</p>
Double side PCB tag	 <p>7.75x9.05x2.03mm³</p>	 <p>10.2x8.8x4.0mm³</p>

After assembly and prior to injection overmolding, the tags are encapsulated using 3D filament printing. Three materials were tested: PLA – polylactic acid, ABS – acrylonitrile butadiene styrene and PC (polycarbonate). As can be seen in Table 5, the PCB-based tags, since these include the commercial pressure

IC, were encapsulated with a opened channel/window corresponding to the location of the sensing membrane on the chip. It serves two purposes: provide a connection to the outside air pressure and to prevent the 3D printing process to damage or block the sensor.

The NFC chip selected was the NHS3152 from NXP. It is in fact an IC optimized for therapy adherence monitoring and logging, with an embedded NFC interface, a resistive network sensing-interface and an internal temperature sensor. It is supplied in wafer level chip-scale package (25 balls; $2.51 \times 2.51 \times 0.5$ mm, SOT1401-1). The internal temperature sensor has a rated absolute accuracy of ± 0.3 °C between 0 °C and +45 °C and ± 0.5 °C in the remaining range between -40 °C and +85 °C. It has an Analog-to-Digital Converter (ADC), a Digital-to-Analog Converter (DAC), a Current-to-Digital Converter (CDC) and 6 analog I/O pins. Most importantly, the NHS3152 can be powered from the NFC field.

The pressure sensor selected was the LPS22DF from ST, which is a low-power and high-precision MEMS pressure sensor for the absolute pressure range 260-1260 hPa. It has an accuracy of 0.5 hPa, a temperature coefficient of 0.45 Pa/°C and supports an output data rate from 1 Hz to 200 Hz. It also features embedded temperature compensation, since it has an internal temperature sensor with 1.5°C accuracy.

The sensor does not feature parameterization/adjustment or calibration capabilities. The sensor will be powered (self-powered) and read by an NFC reader (wireless interface). The reading distance of the microfabricated tag has been demonstrated to be at least 7mm.

Deployment in Demonstrator 2 – Injection molding

This sensors target being overmolded in injected parts, within Demo 2, which can have medical applications, automotive or others. Besides the injection overmolding compatibility, which is in itself a challenge, the novelty will also be the reduced size of the smart sensing tag, which, depending on the particular injected part it will integrate, will enable new applications not yet available. A straightforward example would be the automatic traceability upon handling small injected parts, provided that an NFC reader is integrated in the gripper of the handling robotic tool.

In Demo 2 the injection molding of sensorized parts and its reading at the tool output will allow the monitoring of the yield of the overmolding process, the identification and tracking of each piece, and enable different sensing applications depending on the nature of the injected part. The expected outcome for the sensor and its integration at the end of the IMOCO4.E project is therefore this demonstration beyond the current state-of-the-art and finding partners for new applications of sensorized parts.

Hardware, software and integration components

INL in coordination with ECS (Demo2 owner where the sensor is being deployed) defined hardware and software components as well as components integration within WP6 “Implementation and integration of IMOCO4.E platform” and the documents “HW_components_catalogue.xlsx”, “SW_components_catalogue.xlsx” and “Components_integration.xlsx”. Besides those defined by INL and ECS, there are also components defined by UNIMORE that will have connection to this sensor (in particular in its deployment on Demo 2), as compiled in Table 6.

Table 6. List of the hardware, software and integration ID’s involved on this sensor and its deployment on demonstrator 2

HW ID	Building Block(s)	Task	Partner	Layer	Description
HW-021	3	T3.2	INL,ECS	1,2	NXP NHS3152 microcontroller (HLGA-10 footprint); NFC reader with USB, Wi-Fi and ethernet outputs.

SW ID	Building Block(s)	Task	Partner	Layer	Description
SW-073	3,4	T4.6	UNIMORE	2	Software to read data from NFC reader and store on internal storage disk.
INT ID	Comp. A	Comp. B	Partner		Description
INT-004	SW-073	HW-003	UNIMORE / INL		SW-073 reads the data from the NFC reader and stores them on internal storage disk
INT-047	HW-021	HW-003	UNIMORE / INL/ECS		HW-003 receives data via USB interface from HW-021

2.4.2 Implementation aspects

Sensor description - Microfabricated tags

In summary, the development of the microfabricated tags required:

- 1- The design of the microfabrication process flow for the thin-film microantenna comprising two electroplated metal layers and an approximately 50µm target thickness. This process was designed to be compatible with the microfabrication of thin-film pressure sensor, either resistive or capacitive (prior to the patterning of the microantenna).
- 2- Procurement of bare die NFC chip versions and suppliers, as well as NFC readers with ethernet and USB interface.
- 3- Simulation of microantenna parameters from geometrical constraints and material properties (considering microfabrication constraints).
- 4- The design of layouts and fabrication of hard masks with microantennas of 5.1mm diameter (different designs with trace width and spacing ranging from 15 to 25 microns, and number of windings either 2×8 or 2×9).
- 5- Microfabrication activities including several micromachining runs on 8-inch wafers (main process and calibration runs such as for photolithography optimization), inspection and metrology.
- 6- Experimental evaluation and characterization of the microcoils.
- 7- Firmware development for the NFC chips, temporary soldering of the NFC chips (onto a developed Programming board) for firmware programming.
- 8- Assembly of the NFC chips on the microantennas and encapsulation procedures.
- 9- Development of experimental setups (including reference sensors procurement and readout development) and performance characterization of the sensors (as a function of temperature).

The small form-factor (5.1 mm diameter) of the microfabricated tags enables integration into injection-molded parts, making them a versatile option, highly convenient and non-intrusive solution for precise monitoring of temperature levels. The main challenges are the design – the trade-off between miniaturization and performance, and the microfabrication – the difficulty of patterning high aspect ratio features and vias on 10-25µm thick polymeric layers (photoresist and SU-8). From the simulations of a multilayer planar coil (magnetic quasistatic simulations), a number of windings of 18, trace width and spacing 20 µm, a height of 15 µm, and an internal diameter of 4mm (to easily fit the NFC chip without major interference) results in an inductance of 2.8 µH and a quality factor of 16.

The thin-film (50 µm thick) microfabricated antenna consists of a 2-layer electroplated copper coil (8 or 9 turns each, depending on the design) embedded in SU-8, with 5.1mm diameter and BGA compatible pads.

The layout and the fabrication process are depicted in **Error! Reference source not found.** The microfabrication process (**Error! Reference source not found.**) starts with deposition and patterning of SiO₂ (sacrificial release layer), followed by a 5 μm SU-8 layer which acts as the device bottom flexible base. The active features (coil) are 2 layers of 15 μm electroplated copper, separated by 10 μm of SU-8 (with patterned vias for copper connections). The last material added is a 20 μm-thick layer of SU-8 which is coated and patterned leaving the BGA copper contacts exposed. The devices are finally released using hydrogen fluoride (HF) vapor etch. The runs following this process resulted in significant copper pads oxidation, impairing the use of several microantennas and decreasing yield. Measures to minimize oxidation during the process were effective, but not during the last release step. In a different set of microfabrication runs the process was inverted, that is, started by the BGA copper pads, however, the residual stress of the SU-8 layers upon release damaged several structures upon release (the pads remained attach to the support wafer with the remaining SU-8 area was released faster and buckled). The process was therefore reverted to its original flow order and in order to protect the copper pads from oxidation and enable soldering of the IC, a nickel/gold bilayer was sputtered and patterned on the BGA pads.

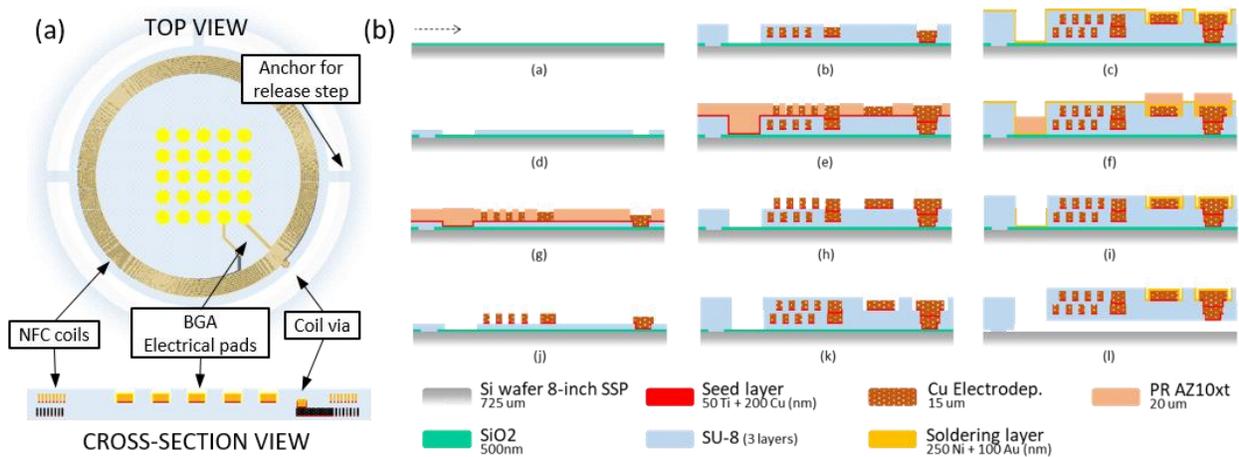


Figure 19. a) Layout of the NFC microantenna and b) main micromachining process steps.

The NFC ICs with integrated temperature sensor (and a resistive network sensing-interface) are then soldered to the flexible microantennas and enclosed in 3D printed shells (**Error! Reference source not found.**).

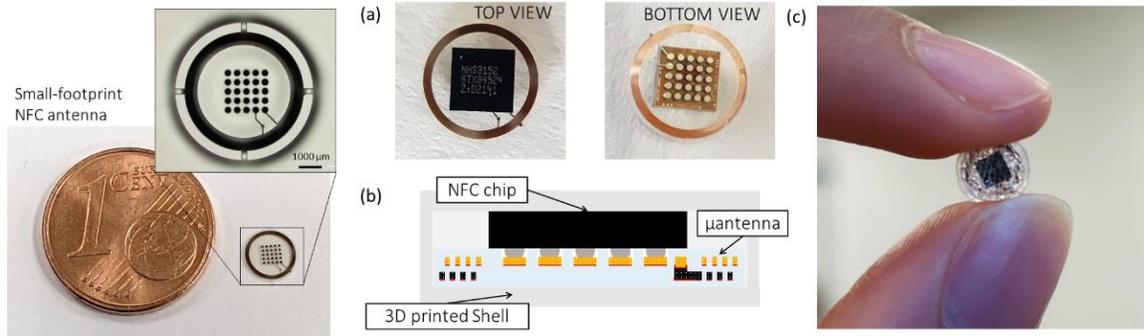


Figure 20. a) photograph and microscope image (close-up) of the fabricated devices Temperature sensing tag before encapsulation, b) encapsulation schematic and c) picture of encapsulated tag.

Regarding the integration of a thin-film pressure sensor with microfabricated antenna, the activities developed were firstly the conceptualization of its layout and respective microfabrication process, and then analytical and Finite Element Modelling simulations were performed to evaluate the achievable electrical capacitance and resistance output as a function of pressure. In parallel, the readout noise limits of the different readout approaches were accessed, including experimentally evaluating the noise of the analog inputs of the NFC chip. The fabricated hardmasks already included layouts for the (electroplated) capacitor plates required for the pressure sensor. The simulations of the thin-film cavity-based pressure sensor considered a membrane diameter range from 0.5 to 1mm (in arrays of 3 by 3), an air cavity thickness between 0.5 to 1 microns, and a dielectric layer (Polyimide) between 2 to 5 microns. The electrode parallel plates would be realized in aluminum on the bottom and copper on top. These simulations indicated that nominal capacitances between 4pF and 80pF would be achievable, corresponding to time constants (using 1M Ω resistor) of 40 to 370 μ s. One readout option would be to monitor the charge time of the capacitor in order to detect a capacitance shift due to external air pressure. The capacitive variations due to the target measurement range (0.8 to 1.2bar) are however too small to be measured considering the ADC maximum resolution of 12.5 μ s. Another set of simulations considered an amorphous silicon piezoresistor patterned at the membrane anchors, but the resistance variations were lower than 0.4% R/R0 was insufficient to be read with the NFC internal resistive readout (which has a maximum rated resolution of 0.2%). The solution of a thin-film based pressure sensor integrated with the thin-film microfabricated antenna can therefore only be pursued with a higher gauge factor piezoresistive material, or alternatively with an NFC chip with higher readout capabilities (at the moment not available). This approach was therefore no longer pursued after these findings.

Sensor description - PCB-based tags

Regarding the PCB-based tags the activities included:

- 1- Procurement of bare die or low profile packaged pressure sensors, technical specifications and suppliers.
- 2- Simulation of microantenna parameters from geometrical constraints and material properties and layout design of the PCBs with microantennas (**Error! Reference source not found.**), and footprint for the selected pressure sensor (the PCBs were fabricated elsewhere).
- 3- Firmware development (progressively improved for higher operation frequencies) for the NFC chips (now including the pressure sensor readout), temporary soldering of the NFC chips (onto a developed Programming board) for firmware programming.
- 4- Assembly of the NFC chips on the microantennas and 3D filament packaging.
- 5- Development of setups for pressure characterization and experimental characterization of the sensors.

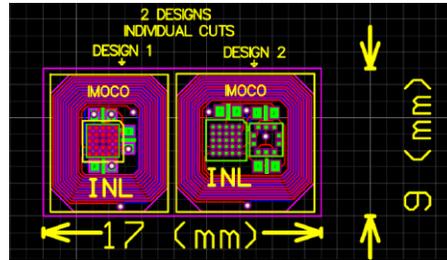


Figure 21. PCB-based tag layouts.

These tags were prepared with firmware updates that allow for batch mode readout at 10Hz.

Deployment in Demonstrator 2 – Injection molding

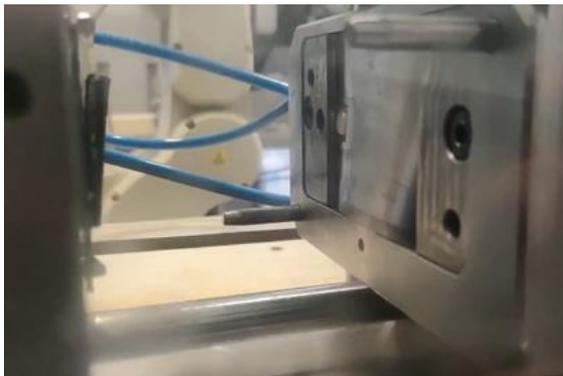
Upon the first integration tests in the injection tool (Figure 22), some constraints were found in terms of the 3D printed capsule shape and the filament used that needed improvements to provide better thermal resistance and cohesion with the injected polymer. The first samples (PLA capsule, 200°C melting temperature) were overmolded but there was deformation of the capsule due to low melting temperature, and the capsule shape was not ideal for handling and fitting. This required changes in the capsule filament material and geometry of the capsule. The materials tested next were PC (270°C melting temperature) and ABS (220°C melting temperature), requiring some tuning of the extrusion process and resulted in slight final capsule dimensions (up to 300 micron difference depending on the material).



a) Test environment.



b) Position of the sample in the tool.



c) Inserts and tool mounted in the injection unit.



d) Plastic component with sample overmoulded after the ejection.

Figure 22. Overmoulding trials with first samples (non-functional prototypes) to evaluate geometric constraints on the injection tool.

2.4.3 Results

Microfabrication results

Prior to the final release step (Figure 23, left side) of the microfabricated antennas, the electrical resistance mapping was performed (Figure 24). This was repeated after release (Figure 23, right side), although upon release some antennas were detached from the polymer frame, so their location could not be specified.

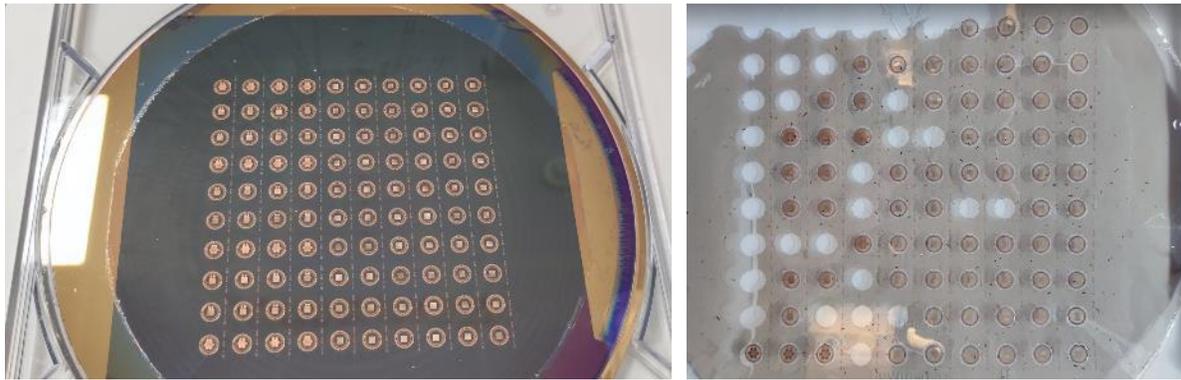


Figure 23. A microantenna run wafer before the final release step (left) and released devices (right).

W20S20N18	W20S20N16								
15.98	16.84	18.81	17.92	18.25	13.37	16.62	14.97	5.48	7.21
W25S15N18	W25S15N16								
12.17	14.2	16.46	15.83	17.26	14.58	15.59	12.93	13.71	9.41
W25S20N18	W25S20N16								
16.47	15.01	17.46	2.1 M	17.32	2.2 M	17.18	11.58	16.08	13.98
W20S15N18	W20S15N16								
15.21	15.26	13.78	12.5	42.5 M	12.62	13.99	14.81	15.09	8.42
W20S20N18	W20S20N16								
18.16	15.92	1.2 M	5.4	15.22	1.5M	9.66	16.28	18.74	15.08
W25S15N18	W25S15N16								
13.48	11.3	11.93	11.18	7.07	0.825M	11.88	6.18	14.26	4.48 M
W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3
18.28	18.15	18.68	18.74	0.21M	17.41	18.41	18.75	17.86	18.23
W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3
18.74	19	14.86	19.11	18.87	17.53	19.19	18.17	17.95	17.65
W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3
16.46	4.2M	17.69	18.56	18.52	17.81	19.22	18.59	17.66	15.73
W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3	W20S20N18C2	W20S20N18C1	W20S20N18C3
13.85	7.8	15.72	18.68	18.06	17.93	17.02	16.86	15.65	3.04

Working 81% Not working 9% Low Res. 8% Damaged 2%

Figure 24. Devices coil resistance values mapped in the wafer before release.

The characterization the microfabricated coils after fabrication and release targeted two performance factors: resonance frequency and quality factor (Table 7). The resonance frequency of the device should be as close as possible to 13.56 MHz for communication with commercial readers, while higher quality factors

indicate larger reading distances (provided that there are no frequency shifts, in which case a lower quality factor is more advantageous).

Table 7. Theoretical (simulated) and experimental data of the microfabricated devices. Singular representation of each design.

Coil device	Resistance [ohm]		Inductance [μ H]		Capacitance [pF]		Resonance frequency [MHz]		Quality factor	
	Theoretical	Real	Theoretical	Real	Theoretical	Real	Theoretical	Real	Theoretical	Real
W20S20N18	14.43	13.08	2.83	2.70	48.6	54.49	13.56	13.11	16.730	17.03
W25S15N18	11.86	11.35	2.83	2.98	48.7	52.16		12.77	20.337	21.06
W25S20N18	11.87	11.59	2.78	2.88	49.5	53.17		12.86	19.973	20.08
W20S15N18	14.37	13.32	2.89	2.97	47.7	51.81		12.83	14.374	17.97
W20S20N16	12.70	11.81	2.28	2.36	60.4	52.86		14.25	15.291	17.89
W25S15N16	10.42	10.14	2.28	2.33	60.4	52.03		14.45	18.652	20.87
W25S20N16	10.43	10.32	2.24	2.27	61.6	53.90		14.39	18.279	19.89
W20S15N16	12.65	12.02	2.33	2.37	59.2	54.23		14.04	15.667	17.39

A device is therefore deemed successful when first and foremost the measured resonance frequency is approximate or equal to 13.56 MHz and the quality factor is of approximate or higher value than the simulated values.

The characterized devices in general present a higher quality factor than expected from the simulated value, and for the resonance frequency only a deviation of 6.5% from the 13.56 MHz is shown (Table 7), demonstrating a very close approximation to the designs simulation.

Sensors characterization setups

After encapsulation with 3D filament printing (and before injection molding), the temperature data and tag ID were successfully read with a smartphone (model Xiaomi Mi 10T) NFC reader at distances of up to 7 mm in the case of the microfabricated tags, and up to 14mm in the case of the PCB single-side tags.

An NFC reader from Advanced Card Systems Ltd was used together with the software NTAG SmartSensor Automator (from NXP Semiconductors) for readout inside the test equipment.

Both microfabricated tags and NFC tags were characterized for temperature response inside a Weiss WKL 34 climate chamber (allowing temperature tests in an enclosed environment with electrical connections to the outside) (Figure 25). The reference sensor used was a Semitec TX04F103F3380ER thick film thermistor (operation range from -40°C to 125°C).

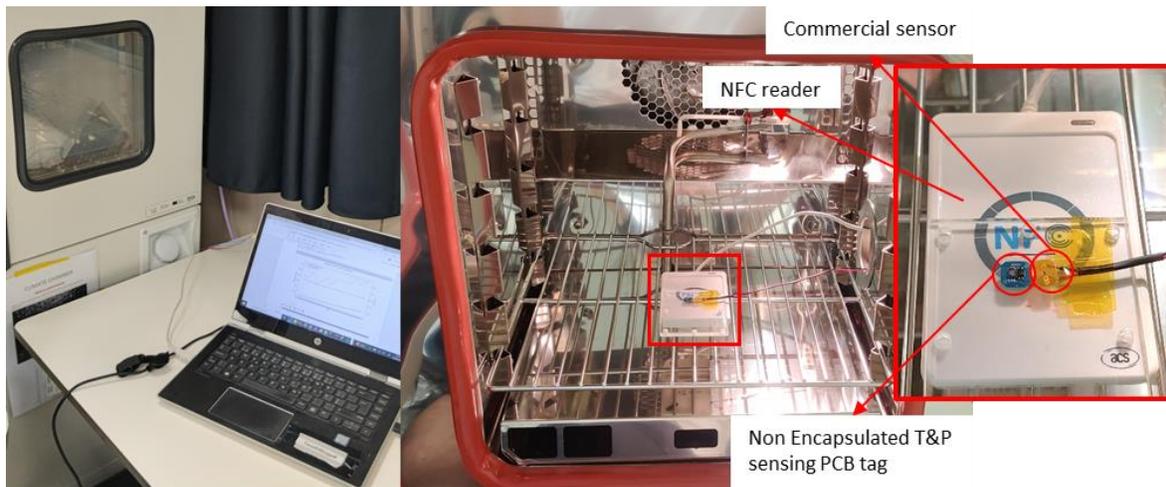


Figure 25. Climate chamber temperature characterization setup.

The PCB-based tags were further characterized for pressure response. A sealed acrylic chamber was fabricated to accommodate the tags and the reference sensors, and connected to either a vacuum line (-0.9 bar at the exit of the vacuum pumps) or a compressor (set at roughly 2.2 bar).

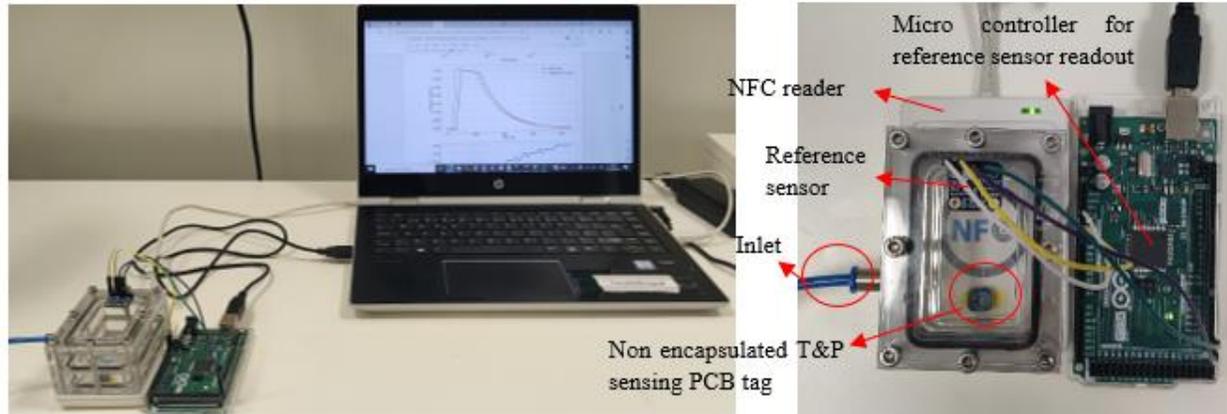


Figure 26. Pressure characterization setup.

Inside the airtight chamber were contained the non-encapsulated temperature and pressure sensing PCB tag and for pressure reference the TE Connectivity MS8607 PHT sensor (10 to 2000 mbar, with 0.01mbar resolution) (Figure 26). Underneath the chamber the NFC reader was placed and then the inlet was connected to the vacuum line or the compressor and completely opened, letting it stabilize for 50 s and then closing the line. The airtight chamber then increased or decreased pressure gradually until it reached atmospheric pressure.

For these measurements, a reading is requested every 500ms, however the software of the NFC reader as an intrinsic latency in which the values measured are only received roughly 700ms after. With this, the real measurements time period is 1.2s (for tests with microfabricated and PCB-based tags). Decreasing the measurement interval to 300 ms results in a measurement time period of 1s, but since the reader software only attributes a 1s resolution time stamp, it can happen that two consecutive values are stored with the same time stamp.

The minimum measurement times achieved with this version of the firmware is between 666 ms and 625 ms (confirmed by reducing the request period to 1 ms, and counting the number of values acquired within 10s with a PCB-based tag: 15-16 measurements), which corresponds to an operation/readout frequency of more than 1.5Hz.

The measurements taken by the commercial sensors were done at an acquisition rate of 0.25 s (4Hz) and the NFC tags seemingly kept up with the measurements of the both the temperature and pressure commercial sensor.

Sensors temperature characterization results

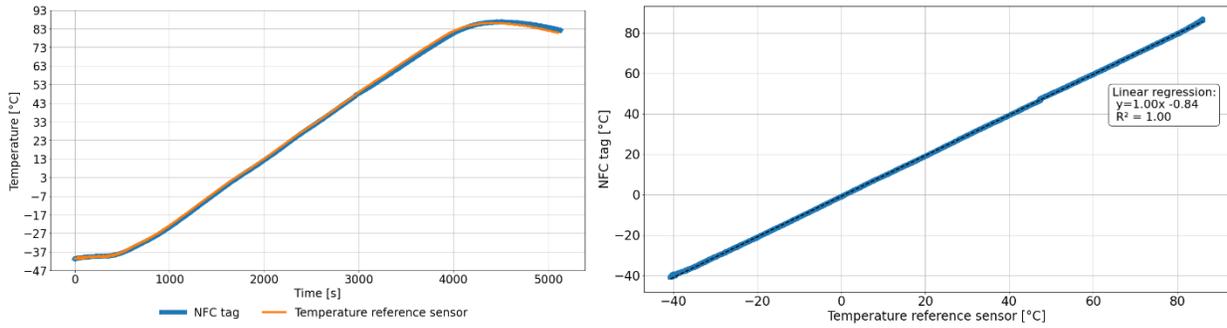


Figure 27. Microfabricated tag temperature measurements. (Left) Data acquisition of the NFC tag and temperature reference sensor from -40 to 85 °C; (Right) Data interpolation of the sensors in the left.

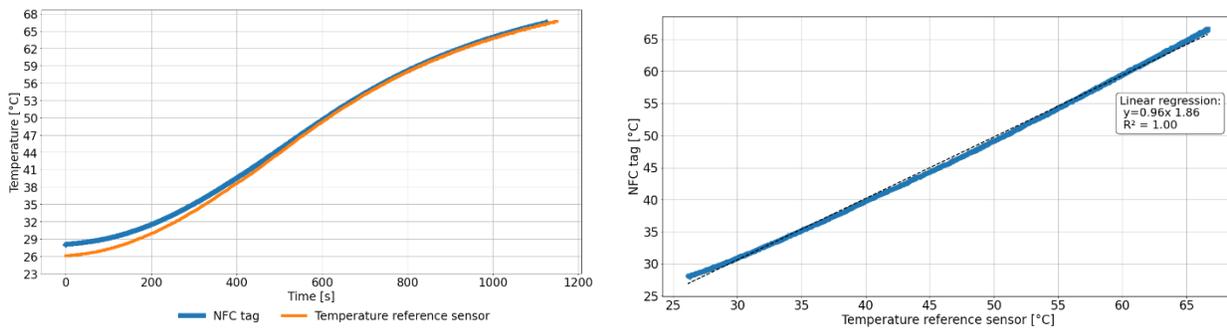


Figure 28. PCB based tag characterization against temperature. (Left) Data acquisition of NFC tag and temperature reference sensor from 25 to 65 °C; (Right) Data interpolation of the sensors in the left. (Negative values not possible with the firmware version of this particular tag).

The NFC tag and the temperature reference sensor present very similar values both for the microfabricated (Figure 27) and PCB based tag (Figure 28). On the PCB-based tag however, there is a discrepancy at the beginning of the measurements (lower temperatures) of roughly 2 °C. This may be due to the fact that the reader is always on and imposing a magnetic field in the sensor, and therefore inducing current on the coil and producing self-heating.

Sensors pressure characterization results

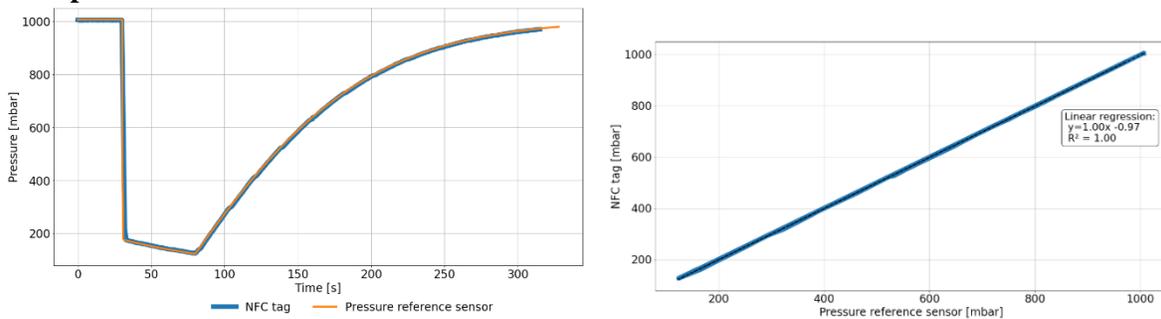


Figure 29. Pressure measurements of the PCB-based tag using the vacuum line (90-160 mbar). (Left) Data acquisition of the NFC tag and pressure reference sensor; (Right) Data correlation of the sensors.

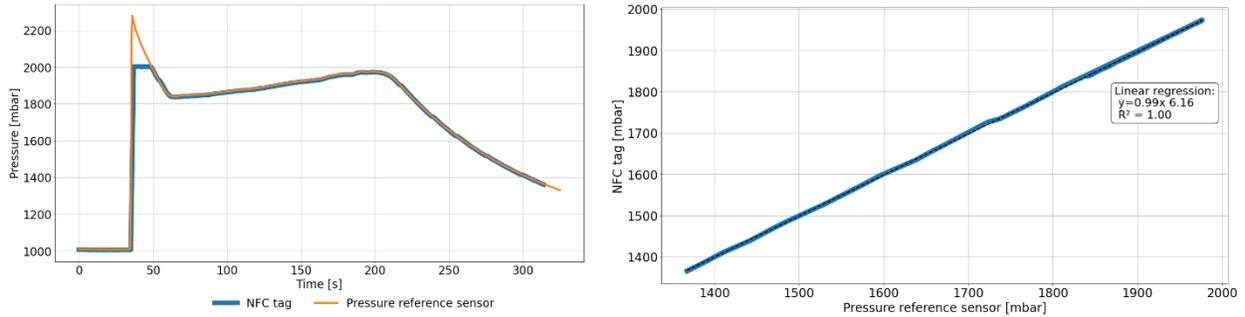


Figure 30. Pressure measurements of the PCB-based tag using the compressor (approximately 2.2 bar). Data acquisition of the NFC tag and pressure reference sensor; (Right) Data correlation of the sensors.

On both cases, respectively the vacuum line (90-160 mbar, Figure 29) and the compressor (2.2 bar, Figure 30), the pressure on the NFC tag accompanied the pressure measured by the commercial sensor, either on the initial fast change and on the slow recover to atmospheric pressure. NFC tag pressure also showed a plateau at 2 bar, as expected, being this the limit of detection of the pressure sensor on the PCB NFC tag.

Readout frequency test (single and batch measurement interrogation modes)

As mentioned in the previous section, the minimum measurement times achieved with this version of the firmware in single-measurement mode is between 666 ms and 625 ms which corresponds to a single measurement operation/readout frequency of more than 1.5Hz.

The firmware was updated to allow for batch mode measurement request, at 10Hz. In this mode the tag receives a request for a package of 10 measurements spaced by 100 ms. Consecutively receiving and requesting data with the reader allows for 10 batch mode measurements in 25s, meaning that the latency increases to 1.5s (blind interval; Figure 31). This is due to message preparation time, NFC protocol timeout and memory limitations. Batch interrogation mode therefore allows for 10Hz measurements over periods of 1 s, at a 0.4Hz frequency.

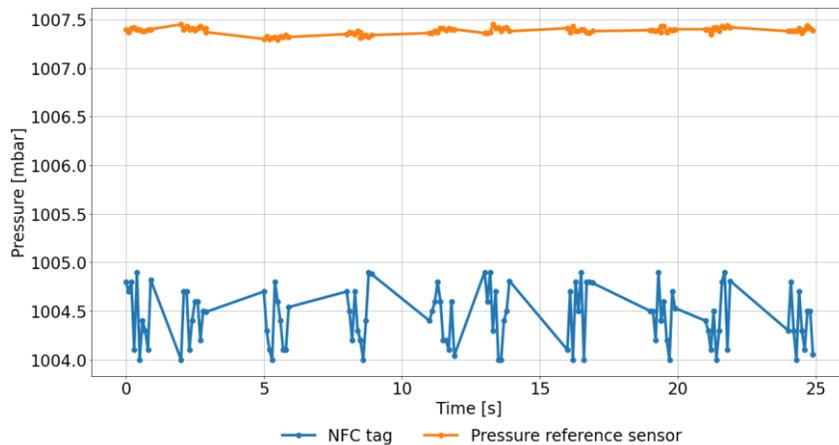


Figure 31. Batch interrogation mode continuous operation.

As the 1s timestamp limitation of the reader software remains, the latency seems to vary between 1 and 2 seconds, but this is an artifact from the 1s resolution.

Preliminary overmolding results

Adjustments were made in order to enhance the fitting of the sensors into the molding area (Figure 32). It was confirmed that the sensors are readable after overmolding. Further functional tests will be reported under the activity 7.2.2 (referring to Demo 2).



Figure 32. Injection molded part with overmolded encapsulated sensor.

2.4.4 IMOCO4.E Requirements

Table 8 shows the results of the tests of requirements verification regarding these sensors. Limitations were encountered in terms of the temperature test range and accuracy. The accuracy limitation relates to the fact that the temperature sensor is integrated in the NFC chip, heat is generated when the NFC tag is being interrogated, and one cannot probe the exact temperature at the temperature sensor location/site (which will therefore be different than the surrounding environment). Additionally, measurements were gathered by the tag at 10Hz, the message with the 10 readings (batch mode) could not be communicated/received at 1Hz frequency due to intrinsic limitations of message preparation communication time.

Table 8. Verification of requirements for HW-021.

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
Req-D3.1-D2	Sensors must be able to read temperatures within the range of -40°C to 85°C, with at least 0.5°C accuracy and in the range of 0°C to 45°C with at least 0.3°C accuracy.	T	TEST-052	PASS	All hardware support an operating temperature range of -40 to +85°C. The internal temperature sensor specifications meet the accuracies required, but chip self-heating in certain conditions (dependent on the reading conditions such as frequency and duration) creates a mismatch between the ambient temperature and the temperature read.
Req-D3.1-D2	Sensors must be able to read variations of pressure and temperature, at least 10Hz.	T	TEST-052	PASS	Limitations known in terms of batch measurement mode duration (due to message preparation time, NFC protocol timeout and memory limitations), but 10Hz batch measurement demonstrated (with blind 1.1 s interval between packages).

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
Req-D3.1-D2	Sensors must be fitted on the tool molding area.	I	-	PASS	
Req-D3.1-D2	Sensors must stand the injection molding pressure and temperature.	T	TEST-052	PASS	The sensors are functional/readable after injection molding

2.4.5 Capabilities and Limitations

The unique selling point of the technology developed is the small footprint of tags, which enables its integration even into injection molded parts. In fact, these are the smallest tags ever reported, to the best of our knowledge, the next best being the 25 × 25 mm² humidity, pressure and temperature sensing tag reported in [1]. The main weakness is the limited readout distance: the reader must be brought to close proximity of the tag. A possible mitigation would be to design a custom antenna for the reader to maximize the readout distance (which will hardly be larger than several millimeters). The encapsulation geometry of the tag shall also be modified so that the antenna is as close to the injected part surface as possible.

The operation frequency being limited to just above 10Hz may also be a shortcoming for some given applications. The minimum measurement time period achieved with this version of the firmware is approximately 650 ms, which corresponds to a tag/reader communication frequency above 1.5Hz. In the developed setup, it was in fact not possible to distinguish how much of this latency comes from the tag message preparation time or the pressure sensor access time. The message preparation time is a limitation inherent to the NFC protocol. The sensor access time is also worsened by the usage of the NFC in energy harvesting mode (the sensor could be interrogated at higher frequencies if the NFC chip would provide higher power to the pressure sensor chip; the power available is dependent on the antenna layout). Therefore, improvement of the frequency performance would require sacrificing the miniaturization level achieved and possibly shift to a battery-based concept. These frequency limitations should not however be impactful for most applications since temperature and air pressure are parameters that typically do not change rapidly, as is the case of Demo 2 (Demo 2 specific frequency requirements have been achieved). Another limitation of the used commercial reader, in particular the reader software, is the timestamp which is given to the received samples with 1s resolution. In the deployment of this solution in Demonstrator 2, however, a different reader (µFR Nano Online Network Reader by Digital Logic) is used, and more importantly, the communication and data storage is managed by a software component (developed by UNIMORE) running on the NVIDIA Jetson AGX Xavier, so more flexibility exist on labelling the data gathered.

2.4.6 Customizations and Adaptations

In order to reduce the size, the sensors do not provide a programming connection for the tags. The delivered sensors do not therefore feature parameterization/adjustment or calibration capabilities (beyond the internal temperature compensation of the pressure sensor). The only customizations that are easily implemented refer to the encapsulation options (material, shape). New tags may be implemented with different firmware versions: the sensor data rate may be adjusted (right now it only supports single reading enquiry or 10Hz

batch mode enquiry) and the temperature sensor used may be either the NFC internal one (high accuracy but susceptible to self-heating upon interrogation) or the one from the pressure sensor (lower accuracy but less affected by self-heating).

2.4.7 Methodology and Toolchains

The tools used in the sensor development phase were:

- Octave, FEMM 4.2, and Coil 32 (for Coil/Antenna simulation)
- KLayout (for microfabrication design)
- Altium (for PCB design)
- NXP MCUXpresso IDE (for tag firmware programming)
- NTAG SmartSensor Automator (NFC reader app for characterization and data acquisition)
- Anaconda.Navigato, jupyter Notebook (for python data processing) and Matlab

Lessons learnt: In order to read the injected part at the injection molding tool output, the positioning of the reader is critical, since the readout distance is limited. In the future, the NFC reader must be provided with a custom-made antenna to increase the readout distance, and an injected part (or reader) handling tool may also need to be developed in order to place and hold it in the reading zone for the minimum amount of time (the tag and the reader take more than 600 ms to order and create and send back the message).

2.5 Low motion distortion 3D depth sensor (ADI)

2.5.1 Technology overview

This solution uses the Analog Devices 3D ADTF3175 Time-of Flight (ToF) module based on CMOS sensor technology, enabling advanced machine vision applications, and allowing people and devices to sense, capture and interact with their spatial environments. The sensor is integrated into Use Case 3 (UC3) - Tactile Robot Teleoperations discussed below.

The module is used in many applications such as machine vision systems, robotics, building automation and augmented reality systems.

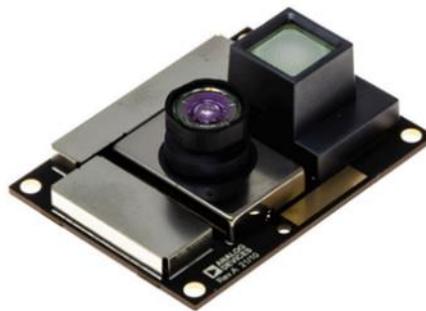


Figure 33. ADTF3175 3D ToF Module

The following list a number of key features of the ADTF3175:

- 1024 × 1024 ToF imager with 3.5 μm × 3.5 μm pixels

- $75^\circ \times 75^\circ$ field of view
- Imager lens subassembly with 940 nm band-pass filter
- Illumination subassembly with eye safety support
- 4-lane MIPI CSI-2 transmitter interface, 1.5 Gbps per lane
- 4-wire SPI and 2-wire I2C serial interfaces
- NVM (Flash) for module boot-up sequence
- Power regulators for local imager and illumination rails
- Calibrated modes at 1024×1024 and 512×512 resolutions
- Depth range: 0.4 m to 4 m (depth noise (1σ) 15 mm maximum)
- 19% minimum target reflectance, 3 klux equivalent sunlight
- Depth accuracy: ± 3 mm (across full depth range)
- Available in 50-lead, MODULE, ML-50-1 package

Data Sheet of ADTF3175:

iMOCO4.E	Novel Sensors	3DToF Sensor	April 2023	
Short Description	BB3		V 0.5	Analog Devices
<p>3D Time of Flight (ToF) Depth Sensor based on CMOS sensor technology. 1MPix depth module with 75°x 75° field of view (FOV) with approx 0.4m-4m operating range, @15% reflectance. Camera module incorporating ToF imager, laser driver, temperature sensor, 940nm VCSEL illumination and optics. High accuracy and precision - +/-3mm with <15mm depth noise. 1Mpix operates at 30fps, 512x512 pixel.</p>				
Drawings	Module: 66mmx58.6mmx67.9mm (approx)			
				
Specifications				
<p>Power supply: USB 3.0 Type C, 2.0A Min, USB3.1, 3.0 A recommended Range resolution/accuracy: +/- 3mm depth error in 0.4m-4m range @15% reflectance Frame rate: 30FPS@512x512 Camera Field of View (FoV): 75°x 75° horizontal</p>				
Interfaces, Data Formats				
<p>Data interface(s), internal: MIPI Data interface(s), external: USB 3.0/3.1 Measurement Data: Depth map, Active brightness map, point cloud Sensor configuration parameters: Resolution mode</p>				
<p>PC-Software: ADIToF GUI displaying point cloud, depth map, and active brightness. Command line scripts to save frame data</p>				
<p>IMOCO4.E application(s): UC3</p>				
<p>AI Contribution: Depth data for object classification, collision avoidance, gesture recognition</p>				
<p>DT Contribution: upon request, sensor information and data can be made available</p>				
© ADI, 2022				

Figure 34. ADTF3175 3D ToF Module Data Sheet.

2.5.2 Implementation aspects

The Analog Devices 3D ToF sensor is implemented in the UC3 architecture (primarily across Layers 1 and 2) to create a safe remote teleoperation via a tactile robot represented in the below figure. Humans in the

loop will be considered through complex Human Machine Interfaces (HMI) coupled with a digital twin representation of the process implemented in virtual reality. The teleoperations platform will be enabled with high performance AI embedded close to the edge. ToF will be used at both the local and the remote ends in UC3 to provide depth data for object classification, object recognition, collision avoidance, human tracking, and gesture recognition.

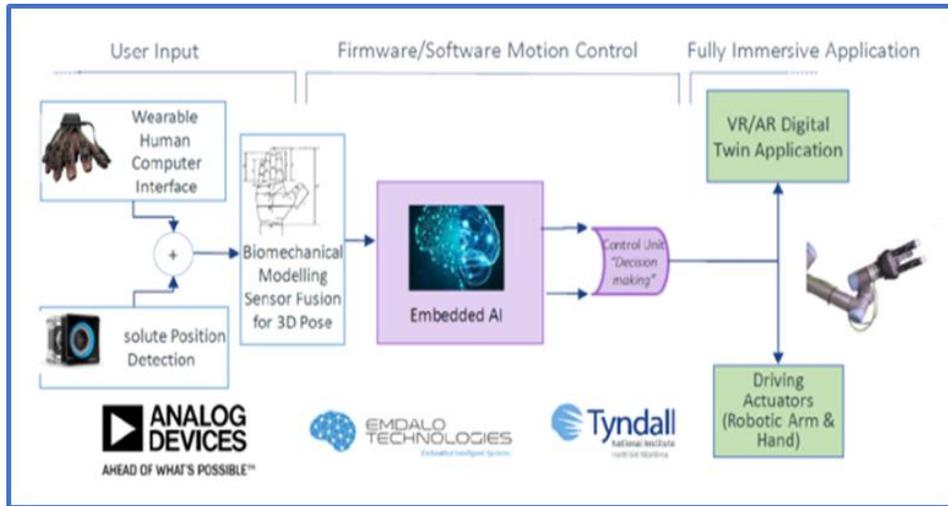


Figure 35. Use case 3 Robot Tele-operations Overview.

The ADTF3175 ToF module is used to generate Infra-Red (IR) and depth images. Using machine vision an algorithm is trained to detect a person's hand and follow its movements. This training can be done by feeding the algorithm with thousands of images of humans, each with the key points of a hand labelled, so that the algorithm can learn to detect when a hand is in an image. There are many pretrained models already developed for use with RGB images. However, the number of pretrained models available for either IR or depth images is extremely limited. This makes the use of IR/depth images for hand tracking challenging. However, a key advantage of IR/depth is that it requires less computational power compared to RGB, which makes it more suitable for use at the intelligent edge.

As part of the UC3 research conducted, multiple hand tracking algorithms were developed and tested to find which created the best model to suit the tele-operations platform. These included looking at pretrained models, as well as developing customised models, trained internally using available datasets. The research also used a combination of RGB and IR datasets to train a new model. The most successful approach tested to date has been using IR images with a pretrained RGB hand-tracking model. This provided a successful detection rate of 89%. Additional work in this area will be to train a new model with IR images and compare the level of accuracy and processing times achieved with current performance. The below figure provides an overview of the ToF logical processing related to UC3.

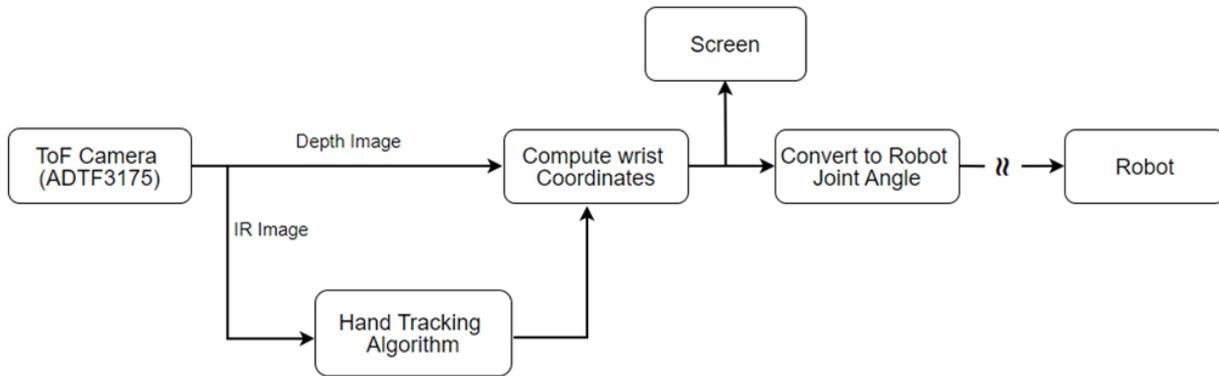


Figure 36. ADF3175 Logical Processing Overview.

The IR image together with the depth image are used to find the position of the wrist in space with X, Y and Z coordinates, where X and Y are obtained from the pixel location of the wrist in the IR image and Z is the wrist depth, obtained from the depth image as shown in the below figure. To begin the remote control of the robot (teleoperation) in UC3, the user would position their hand in front of the camera (ADTF3175) with their wrist centered in the middle of the screen. The system will then detect the position of the wrist and save the X, Y, Z coordinates. Next, it will start to calculate the tele-operators wrist movements across the frames and convert it to robot joint angles, which will be used to move the robot at the remote end of the UC3 platform.

In the UC3 configuration, the robot is pre-programmed with a pre-defined initial starting position. When the robot receives the computed joint angle co-ordinates it will navigate to the requested position. This will allow the robot to continuously track the movements of the operator’s wrist. As part of the research and development work conducted, several options for real time robot control were tested. The optimum in terms of latency for the UC3 project was found to be when using the Real Time Data Exchange (RTDE) Robot interface.



Figure 37. ADF3175 ToF Wrist Tracking in UC3.

2.5.3 Results

This ToF focused phase of the UC3 research has been conducted at the ADI Catalyst centre using the Intel® NUC 10 performance kit (NUC10i7FNH) edge computing device. In overall testing the system shows reliable results for real-time teleoperation with minimum latency. To put this into context, a latency of 0.3 seconds is considered safe for teleoperation for precision surgery.

For our current test system at ADI Catalyst we are currently achieving:

- **Overall system latency:** Achieved 0.3 secs for the total latency measured from the operator movement to the equivalent robot movement.
- **Frame rate:** 30 fps for the hand tracking algorithm.
- **Detection rate:** 89% detection of the wrist from images frames.

At this stage, the testing conducted at the Catalyst centre confirms that the robotic teleoperations system works in real time such that the robot moves almost instantaneously with the user hand movements.

Future Research and Directions:

- New IR hand tracking model is under development and should increase the detection rate further.
- Work is to start on developing a gesture recognition model that will be used to open and close a robotic gripper device.

Readers note: Full HW and SW testing for UC3 across all technologies is more formally reported into WP6 deliverables.

2.5.4 IMOCO4.E Requirements

The below presents a set of requirements that are being addressed as part of the ADI ToF research and development engineering for UC3. A status update is also provided in relation to the progress of the requirements. The reader is referred to WP3 D3.3, for detailed discussions on the key requirements presented below.

Table 9. Verification of requirements for low motion distortion 3D depth sensor

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
Interfaces and connectivity					
Req-D3.2-U3-4-com	Optimization of the Frames Per Second (FPS) connectivity between 3D ToF camera and PolarFire edge components.	I	TEST-046	IN PROGRESS	We are working on achieving this requirement.
Performance					

Req-D3.2-U3-1-hw-sw	Frame rate: 5 frames per second at 640 x 480 sized frames.	I-D	TEST-046	IN PROGRESS	We are aiming to reach better fps than the requirement specified.
Req-D3.2-U3-4-hw-sw	Application and bespoke development of position tracking algorithms using 3D ToF data, capable of detecting wrist movement of a standing human at a fixed distance from the camera to an accuracy of 98%, at the local end, in normal lighting conditions with a latency of < 200 ms.	I-D	TEST-046	IN PROGRESS	We are working to achieve this requirement.
Req-D3.2-U3-5-hw-sw	Application and bespoke development of object tracking algorithms using 3D ToF data, capable of detecting objects in the remote scene (at the robot/gripper end) in 3D space; objects to be sized at 5 cm or larger.	I-D	TEST-046	IN PROGRESS	We are working to achieve this requirement.
Tools/toolchains					
Req-D3.2-U3-9-sw	Produce a number of use case related data sets, incorporating HMI-IMU (tactile glove), motion trackers and ToF (depth camera) sensor data streams to be used in both cloud and PolarFire embedded AI\ML related research.	I-D	TEST-046	IN PROGRESS	We are working to achieve this requirement.
Req-D3.2-U3-10-hw	Research and development using generally available open-source APIs and SDKs to develop, test and re-engineer (as applicable) detection algorithms for the various 3D ToF camera data used in the use case sensor architecture.	I-D	TEST-046	IN PROGRESS	We are working to achieve this requirement.

2.5.5 Capabilities & Limitations

The 3D ToF sensor along with embedded AI vision algorithms is used to capture a user’s wrist, hand gesture movements and 3D positioning to move a remote robot in the UC3 platform.

Unique selling points (USP): Its small size and high accuracy are the unique selling points for the ADI ToF sensor. For applications that benefit from higher resolution 3D depth information, CW CMOS ToF cameras provide one of the highest performance solutions on the market.

Strengths: Some of the system features enabled by high resolution CW ToF sensor technology are described in more detail in the below figure. These system features also translate to consumer use cases such as video bokeh, facial authentication, and measurement applications, as well as automotive use cases such as driver alertness monitoring and automated in-cabin configuration.

Continuous Wave Time of Flight System Features	
System Feature	Enablers
Depth precision and accuracy	Modulation frequency Modulation schemes and depth processing
Dynamic range	Readout noise Raw frame rate
Ease of use	Calibration procedure Temperature compensation Eye-safety monitoring
Outdoor operation	Sensitivity at 940 nm Illumination power and efficiency
2D/3D fusion	Pixel size Depth and 2D IR images
Multisystem operation	In-pixel cancellation of interfering light Camera synchronization

Figure 38. Continuous Wave Time of Flight System Features

Weaknesses: ADI ToF 1 Megapixel depth module provides only depth and IR data. It does not provide RGB data. ADI are planning to integrate RGB image functionality to the sensor in the future.

Limitations: ADI ToF 1 Megapixel depth module has a license agreement for industry-best ToF technology with the following features:

- Resolution: 1024×1024.
- Field of view (FOV): 75°x75°.
- Depth range: 0.4 m to 4 m @ 15% reflectance.
- Depth Noise: less than 15mm.
- Accuracy: ± 3mm depth error.

2.5.6 Customizations & Adaptations

The ToF sensor and related algorithms can be used for multiple forms of object tracking and with specific customisation ToF may be applied across selected IMOCO4.E projects. ADI are constantly innovating and the research conducted for UC3 will have major potential in industry, healthcare, sustainability and AGV sectors in the future.

Modifications: Modifications will be made to the ToF SDK to be compatible with more advanced ToF sensors in the future and also with a more intuitive installation interface.

Extensions: Planned RGB sensor to be integrate to the ToF sensor in the future. ROS 2 driver will also be added to the SDK. Additional AI models are expected to be integrated into the SDK in the future.

2.5.7 Methodology & Toolchains

Methodology & Toolchain: A depth camera is a camera where each pixel outputs the distance between the camera and the scene. One technique to measure depth is to calculate the time it takes for the light to travel from a light source on the camera to a reflective surface and back to the camera. This travel time is commonly referred to as the time of flight (ToF).

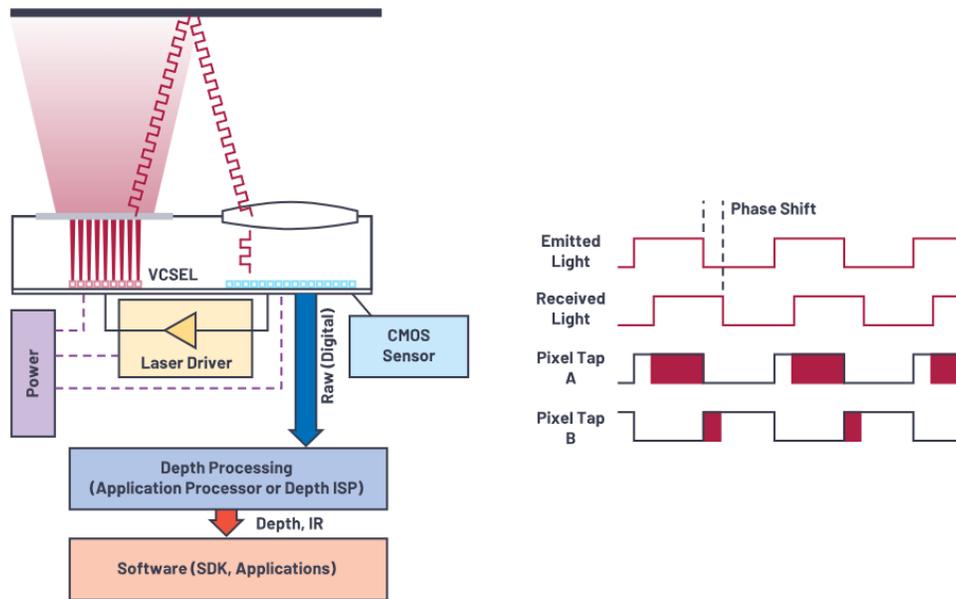


Figure 39. Time of Flight Sensor Components.

A ToF camera is comprised of several elements (see figure above) including:

- **Light source:** Such as a vertical cavity surface emitting laser (VCSEL) or edge-emitting laser diode—that emits light in the near infrared domain. The most commonly used wavelengths are 850 nm and 940 nm. The light source is usually a diffuse source (flood illumination) that emits a beam of light with a certain divergence (aka, field of illumination or FOI) to illuminate the scene in front of the camera.
- **Laser driver:** Modulates the intensity of the light emitted by the light source.

- **Sensor with a pixel array:** Collects the returning light from the scene and outputs values for each pixel.
- **Lens:** Focuses the returning light on the sensor array.
- **Band-pass filter co-located with the lens:** Filters out light outside of a narrow bandwidth around the light source wavelength.
- **Processing algorithm:** Converts raw frames from the sensor into depth images or point clouds.

One can use multiple approaches to modulate the light in a ToF camera. A simple approach is to use a continuous wave modulation—for example, a square wave modulation with 50% duty cycle. In practice, the laser waveform is rarely a perfect square wave and may look closer to a sine wave. A square laser waveform yields better signal-to-noise ratio for a given optical power, but also introduces depth nonlinearity errors due to the presence of high frequency harmonics.

Integration aspects: ToF is an emerging 3D sensing and imaging technology that has found numerous applications in areas such as autonomous vehicles, virtual and augmented reality, feature identification, and object dimensioning. ToF cameras acquire depth images by measuring the time it takes the light to travel from a light source to objects in the scene and back to the pixel array. The specific type of technology that Analog Devices uses in its sensor implementation is called continuous wave (CW) modulation, which is an indirect ToF sensing method.

ToF limitation: At the moment, the ToF sensor can be integrated with ROS version 1.0. ADI are working on adding integration for ROS 2.0.

Lessons learnt: Considerable learning, time and engineering effort is required for the deployment of ToF technologies to specific use cases. In particular, this may involve customised adjustments to the hardware or physical environment of the use case. The development of the ToF software stack will also involve various bespoke development and algorithm engineering in relation to the use case in question.

2.6 Radar sensor (IMST)

2.6.1 Technology overview (From D3.3)

In this project a 79 GHz frequency modulated continuous wave (FMCW) radar is being developed. One novelty for us was the implementation of a MIMO (Multiple Input Multiple Output) system. Hence the use of multiple antennas is necessary. Three transmit (Tx) and four receive (Rx) antennas are used. This system allows an angular resolution for detected objects or participants, which improves the radar's possibilities to distinguish objects with different angles to the radar in addition to range and speed domain.

Before IMOCO, the radar was used solely for stationary applications. Hence another new challenge is to mount the radar on a moving vehicle within Demonstrator 3. Furthermore, the implementation of AI into the sensor will be supported by hardware pre-settings. This will allow the radar to classify moving objects based on their characteristic signal spectrum into categories e.g. pedestrian, forklift or bicycle. This characteristic data is called Micro-Doppler data, since different objects exhibit additional velocity (Doppler) information e.g. through the rotation movements of wheels. A neural network can be fed with such data and classify targets. Neural networks will be provided by other project partners.

The radar sensor is operating in FMCW mode. Therefore, it measures the distance and velocity of reflective objects with respect to the radar. Also, the angle of an object or person can be measured. Often a detected radar object has multiple targets distributed over range and angle and velocity. Therefore, the radar can

combine in these dimensions nearly overlapping targets to tracks. This combines multiple targets into one track. The movement of these tracks can be extrapolated. This feature is very useful when it comes to detecting a human which has multiple reflections due to corpus, arms and legs. When a track is detected the Micro-Doppler data can be saved as well.

An innovation which was developed within this project is the estimate of the vehicle's own travelling speed. A moving radar recognizes all static targets as moving targets relative to its self-velocity. Since the classification can only be done on moving targets, the radar has to distinguish between static and dynamic ones.

At the end of the IMOCO project IMST aims to have a fully functional MIMO radar which detects targets in the above-mentioned domains. Additionally, it can distinguish between static and dynamic targets. Micro-Doppler spectra of targets identified as dynamic objects can be saved separately. The radar can classify moving targets as soon trained Neural Network is available.

The radar sensor will be developed for obstacle detection, collision avoidance, path finding, navigation and target classification for applications in industrial (or medical) environments. The sensor will be optimized for forklift navigation in warehouses. The activities are related to Demo 3 "Warehouse logistics" led by partner STILL within IMOCO4.E. Furthermore, the radar sensor appears to be applicable in Pilot 4 "Healthcare robotics" led by partner PMS. First radar measurements have been made within the frame of Demo 3. Tests within Pilot 4 have been conducted as well.

The **radar hardware** (HW-001) is ready developed and fully manufactured within an IP65 housing as shown in Figure 40. The following figure shows a photo of the sensor with a transparent cover, which allows a view onto the two radar boards. On the upper one the antennas and the black radar chip can be seen. There are 4 receive (left side) and 3 transmit (right side) patch arrays in a MIMO configuration visible. The 77 GHz radar chip is similar to those, which are applied for automotive ACC applications and can be either used in the 76 to 77 GHz or in the 77 to 81 GHz frequency band. The sensor allows detection and tracking of targets in azimuth direction and estimation of obstacles in elevation (e.g. the height of a subway). Targets will be measured in distance with a range resolution of 15cm @ 1 GHz bandwidth, in speed, in direction (azimuth and elevation angle), signal strength (magnitude of radar reflection), lifetime parameter and micro-Doppler spectra. With the help of μ -Doppler data and the execution of trained Neural Networks (NN) radar targets can be classified. The lower board within the radar sensor combines these functions: DC-supply, digital data interfaces, Radar Micro-Controller Unit (RMCU) and Digital Signal Processor (DSP).

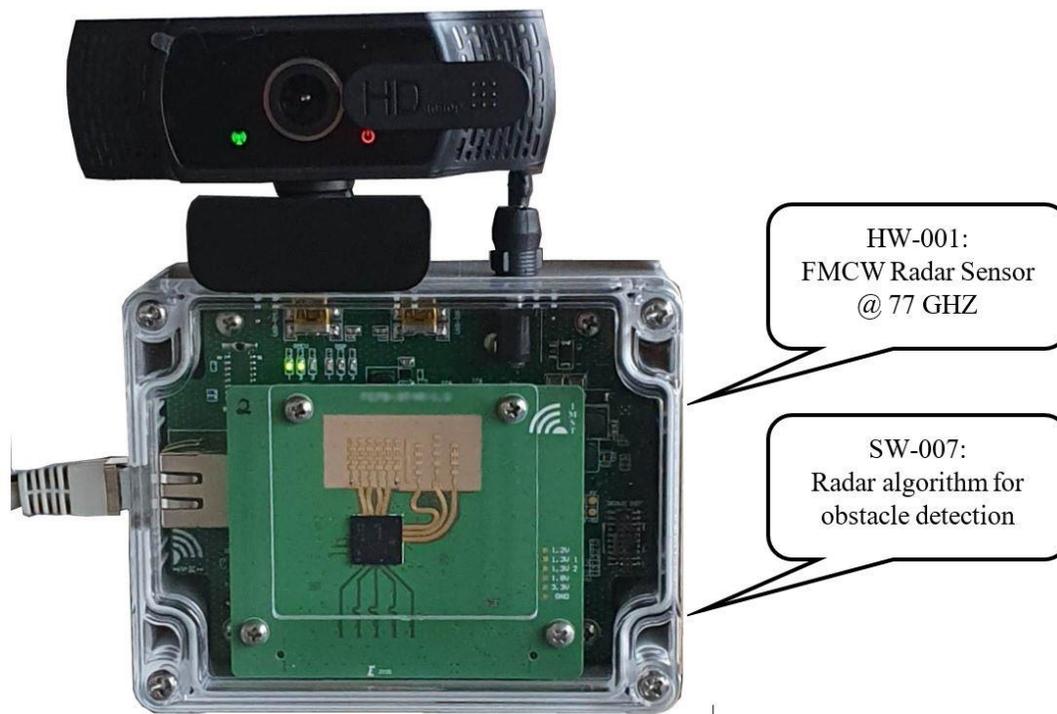


Figure 40. 77 GHz Radar Sensor with external USB camera.

Radar firmware (SW-007) was optimized and adapted to specific measurement tasks during the ongoing project. IMST's radar sensor uses FMCW method (Frequency Modulated Continuous Wave) to measure distances to objects/targets with single frequency ramps around 77 GHz. If not, only a single frequency ramp is transmitted, received and evaluated, but a sequence of e.g. 64 or 128 ramps/pulses are processed one after each other, the time signal can be processed (after A/D conversion) by a 2-dimensional FFT. The Range FFT will determine distances of targets, while the Doppler FFT will evaluate the corresponding velocities. The result of this can be written into a range-Doppler map (RDM). Since multiple antennas are used, each combination results in one RDM, whereby 12 is the maximum RDM number. In the step "Antenna Combining" these results are merged into one RDM.

In [1] we show a method how a moving radar can estimate its own velocity. This is useful since the micro-Doppler signals of moving objects/participants are desired for classification. A moving radar, as it would be moving on Demonstrator 3, recognizes all static targets as dynamic ones. Therefore, a new step in the Firmware is introduced which is called "Velocity Estimation" in order to separate static objects from dynamic targets. It is performed on the resulting RDM of the previous step. With this additional processing step the radar recognizes its own moving speed, hence dynamic targets can be distinguished from static ones. This step only calculates the own moving velocity and doesn't alter the RDM data.

"Peak Detection" and a CFAR (Constant False Alarm Rate) algorithm are the next logical step. After these two steps only the areas in the RDM with comparatively high magnitude will be considered as detections. The resulting data will be used to "Detect" and "Track" targets within the entire Radar observation area. By using a MIMO method (Multiple-Input-Multiple-Output), it is also possible to determine the directional angles in azimuth and elevation. This is done in the processing step "Target Detection". A target list will be

created and made available via an Ethernet interface. Data from each processing step can be obtained from the sensor. The following drawing illustrates the Radar processing steps, which are implemented into the firmware. Radar parameters can also be set, and radar data can be read out via a Windows GUI (**SenTool™**) or with the help of a comprehensive set of commands. Moreover, SenTool™ offers various display options for plotting radar data.

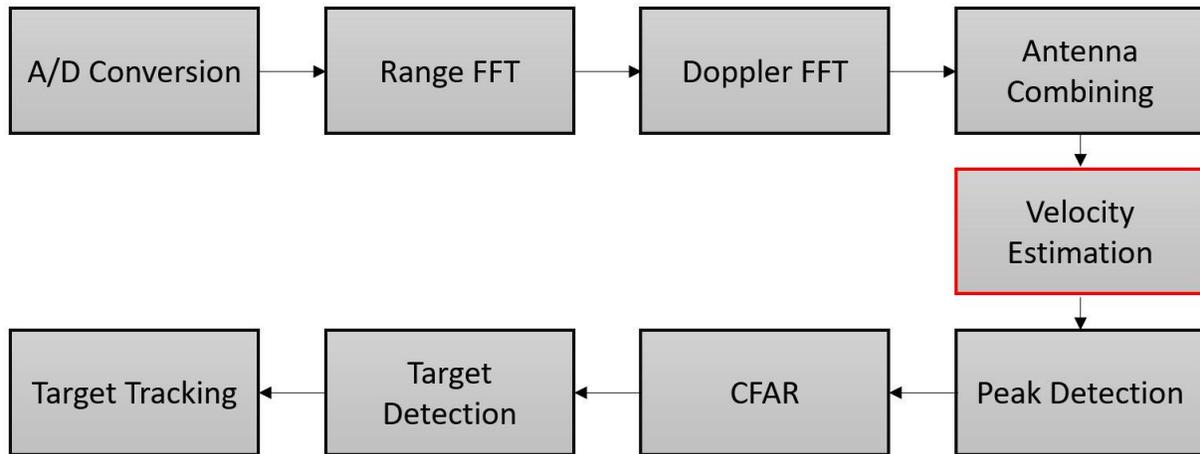


Figure 41. Radar processing steps implemented into the firmware.

If a radar sensor is connected to the Ethernet, it can be accessed with its IP address within a standard Internet browser. The graphical user interface (GUI) of the web interface is part of the radar firmware, which is entitled "IMST **Sentire™** Radar". "Sentire™ Radar" is a trade name registered by IMST GmbH and covers all IMST radar products and radar demonstrators. The GUI consists of 5 different windows: Information, Processing, Frontend, Ethernet and Errors, where information and parameters of the radar can be displayed or modified. This web interface is the easiest way for a user to read, change and save radar parameters.

Table 10: Resolution settings of measurement domains.

Parameter	Resolution	Unit
Distance	0.075 m @ 2 GHz Bandwidth	m
Velocity	0.22 @ 44.3 μs per Chirp (64 Chirps)	m/s
Angle	16°	°

The parameter resolutions used in Table 10 are an example which fits our application. The range and velocity resolution also can be fit for other applications e.g. if slower objects or longer ranges are needed. The angular resolution can only be downgraded if necessary.

2.6.2 Implementation aspects

Originally, the radar was to be used on a forklift in demonstrator 3. As can be seen in the following illustration, it was mounted on the front of the vehicle. This allowed it to take measurements from this side. The data was recorded directly from a GUI running on the laptop. At the same time, a camera was in operation to validate the radar data. Several scenarios have been conducted. The scenario complexity starts

simple for first scenarios and increases with later scenarios. The definitions can be found in D6.3 in section 3.8.3.3 (Demo3) and 3.4.3.3 (Pilot 4).

For the testing campaign in Pilot 4 two radars have been used. This can be also seen in Figure 42. One radar was used to gather data in the azimuthal direction while the other radar, rotated by 90°, was able to acquire data in elevational direction. The radar, which was aligned for elevation measurements should observe the detector of the C-arc robot. The azimuthally aligned radar should record obstacles in the pathway of the moving C-arc robot. In this case a camera has been used as well to identify radar targets for evaluation.

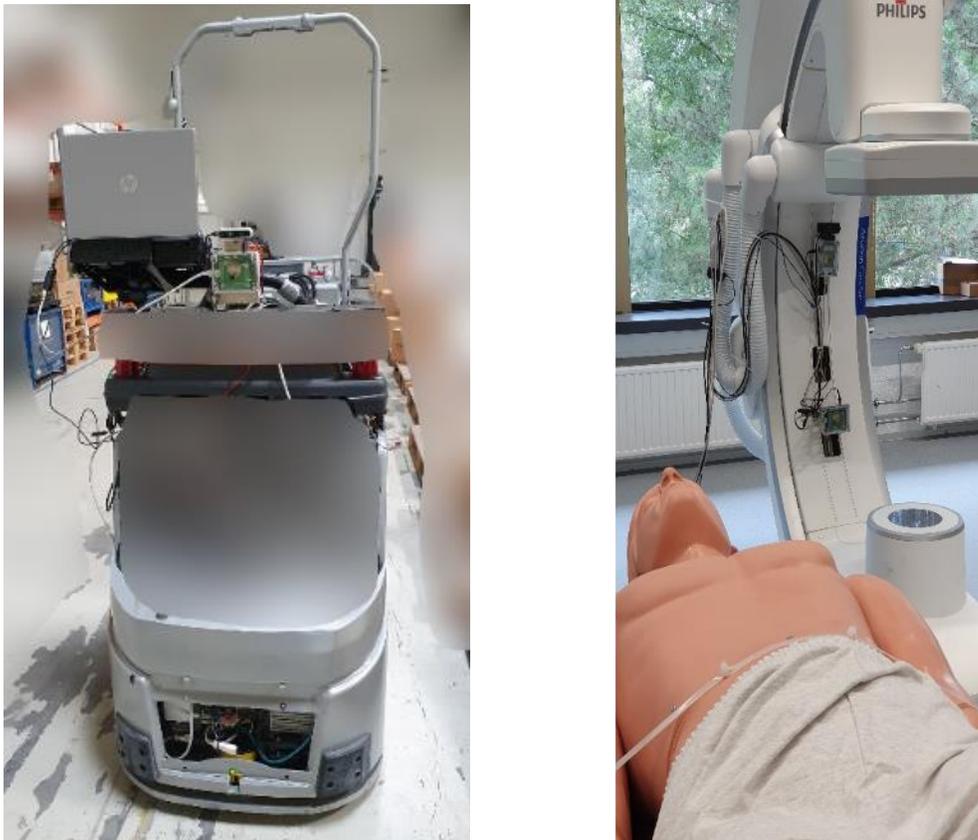


Figure 42. Radar implemented in Demo3 (left) and Pilot 4 (right).

2.6.3 Results

The radar sensor is intended for the integration into Demo 3 to support autonomous driving of a forklift vehicle in an industrial area. In Pilot 4 it is intended to use the radar to increase safety in terms of a medical robot (C-Arc) operating as a high-tech tool for doctors. Requirements were derived from this application: Detect static obstacles within warehouse environments like shelves, beams/columns/pillars, walls, passages, larger objects such as pieces of goods as well as moving targets like other industrial vehicles and people.

After analyzing data from the first measurement campaign in Hamburg, it is evident that the radar provides the detection of most of named obstacles. Especially humans, shelves and large to medium sized metallic objects were detected. However, the radar is not able to detect non-metallic objects like card boxes due to the very low reflectivity of these items.

The results in Pilot 4 are different to the ones obtained from the results of Demo3. The operation of the radar in a higher reflective environment shows, that the results became inferior compared to the data in Demo 3. Highly reflective objects signal's, like the robot's detector arm, can interfere as their signal is multiple times larger compared to other desired detections. Static obstacles were detectable. However, in some cases humans were only detected as they were moving due to above mentioned reason.

2.6.4 IMOCO4.E Requirements

The validation of the following requirements was conducted and recorded in Deliverable 6.5.

Table 11: Requirements for the Radar sensor.

Req ID	Requirement Description	Verify	Comments & Rationales
Req-D3.2-L1-hw-02	Radar sensor has a data interface via Ethernet or USB. Ethernet is preferred.	T	
Req-D3.2-P4-com	Sensors shall provide industry standard wired interface for data exchange (e.g., EtherCAT, Ethernet or USB).	I	
Req-D3.2-L1-fw-03	Firmware of sensor can be updated via Ethernet or USB interface.	D	
Req-D3.2-D3-hw	Sensors can detect static and moving targets at a maximum distance of at least 8 meters in an indoor environment.	T	Demo 3 D6.5 results for reference.
Req-D3.2-P4-hw	Within 1.5m distance the sensor will detect obstacles of at least 15 cm in size with an accuracy of at least 5cm.	T	Changed object size from 10 cm to 15 cm.
Req-D3.2-P4-hw	The sensor shall not require a power supply voltage greater than 24Vdc	I	Minimum 12V are required.
Req-D3.2-P4-hw	The sensor shall update obstacle data with at least 5Hz.	I/D	Minimum rate: 6.7 Hz
Req-D3.2-P4-hw/sw	The sensor shall fault safe. This means that by output of the sensor the receiver of the data can determine that the sensor is in a fault state (e.g., a watchdog signal or checksum mismatch)	I	Error messages can be sent by GUI or read from device via Web interface.
R69-swD4.2	Radar data must be saved as .bin or .csv data	T	.bin is always preferred
R70-swD4.2	The user can determine important thresholds and settings from a GUI interface	T	

R71- swD4.2	Constant False Alarm Rate (CFAR) can be done in range and Doppler domain	T	
Req-D3.1- D3-sw	Use of the 77-81 GHz band: 2 GHz bandwidth in the first measurements.	D	
Req-D3.2- P4-sw	If the sensor has some form of onboard “AI” processing that determines its output, the “raw” data of the sensor shall be traceable for analysis.	I/D	

2.6.5 Capabilities and Limitations

Strengths and weaknesses

The strength of the radar lies in the detection of objects through distance and speed measurements as well as in the fast processing of the raw data. The radar can also detect objects in situations where there is no light or only poor visibility. On the other hand, objects or participants with low reflectivity cannot be detected reliably.

2.6.6 Customizations and Adaptations

- **Component name:** 77 GHz Radar
- **Component HW/SW/INT catalogues reference:** HW-001 / SW-007
- **Use of component in the P/D/UC (*out-of-the-box functionality*):** The radar device detects objects and humans in range, velocity and angle respective to the radar

2.6.7 Methodology and Toolchains

Lessons learnt

- The result of Pilot 4 shows that environments with many objects with high reflectivity can lead to a deterioration in the radar's ability to detect participants in certain cases. This shows that the radar must always be evaluated in a situation such as the environment in which it is to be used.
- Successfully estimating the speed of a moving radar becomes challenging while the radar is driving in curves.

2.6.8 References

[1] L. Meyer, R. Jetten, R. Kulke: „Radar for Industrial Vehicular Application“, International Radar Symposium IRS 2023, Berlin, 26. May 2023”

2.7 Dynamic vision sensor (Reexen)

2.7.1 Technology overview

In contrast to standard cameras, which acquire full images at a rate specified by an external clock (e.g., 30 fps) the Dynamic Vision Sensor (DVS), respond to brightness changes in the scene asynchronously and independently for every pixel (Figure 43). Thus, the output of an event camera is a variable datarate sequence of digital “events” or “spikes”, with each event representing a change of brightness (log intensity) of predefined magnitude at a pixel at a particular time. This encoding is inspired by the spiking nature of biological visual pathways.

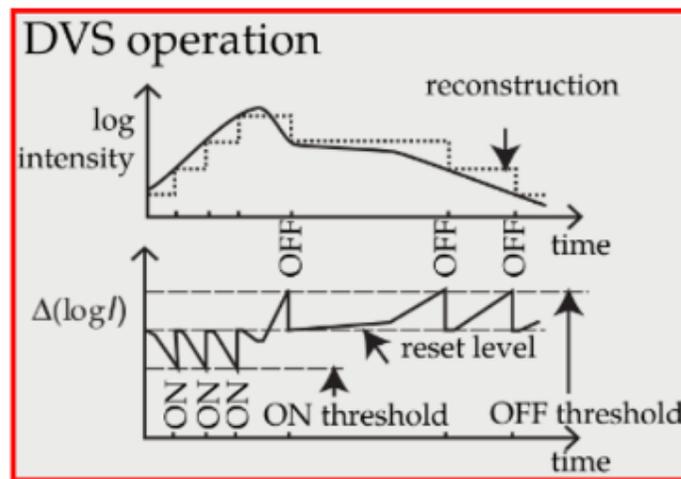


Figure 43. Schematic of the operation of a DVS pixel, converting light into events.

Each pixel memorizes the log intensity each time it sends an event, and continuously monitors for a change of sufficient magnitude from this memorized value. When the change exceeds a threshold, the camera sends an event, which is transmitted from the chip with the x, y location, the time t, and the 1-bit polarity p of the change (i.e., brightness increase (“ON”) or decrease (“OFF”)). (Figure 44)

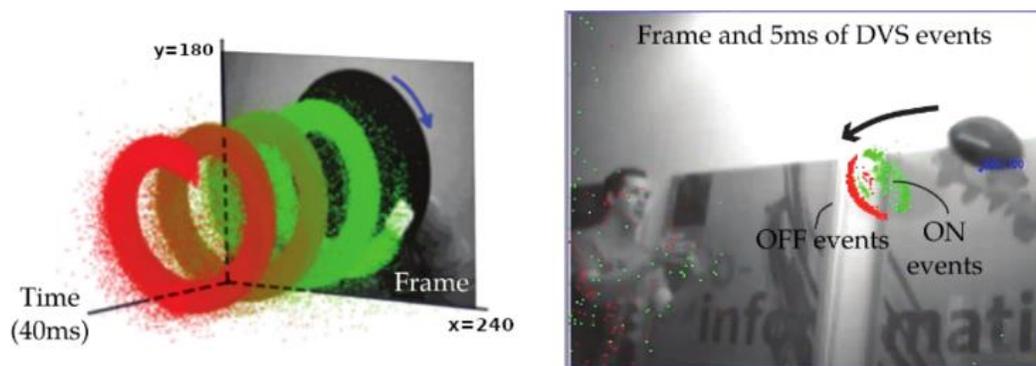


Figure 44. Events in spacetime on the left side of the figure are color coded, from green (past) to red (present). The right side of the figure is the frame and overlaid events of a natural scene, the frames lag behind the low-latency events (colored according to polarity).

The events are transmitted from the pixel array to periphery and then out of the camera using a shared digital output bus, typically by using address-event representation (AER) readout (*Figure 45*). This bus can become saturated, which perturbs the times that events are sent. Event camera shave readout rates ranging from 2 MHz to 1200 MHz, depending on the chip and type of hardware interface. Event cameras are data-driven sensors: their output depends on the amount of motion or brightness change in the scene. The faster the motion, the more events per second are generated, since each pixel adapts its delta modulator sampling rate to the rate of change of the log intensity signal that it monitors. Events are timestamped with microsecond resolution and are transmitted with sub-millisecond latency, which makes these sensors react quickly to visual stimuli.

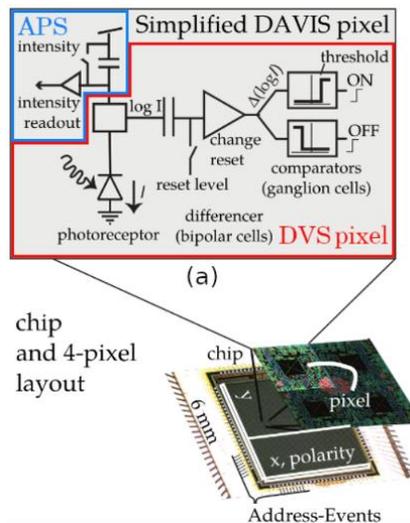


Figure 45. DVS chip.



Figure 46. DVS USB camera.

2.7.2 Implementation aspects

Each pixel of Dynamic Vision Sensor (DVS) works independently, and there is no need to wait for a global exposure time of the frame, As soon as the change is detected, it is transmitted. Hence, DVS have minimal latency, making it suitable for recognizing rapidly moving objects. We use DVS for human detection.

Specifically, the camera model we use is imx636 (Figure 47), with a resolution of 720p. It is connected to a laptop or development board (Figure 48) via USB. The event stream data is transmitted to the processor in real time, and the human detection task is implemented through AI algorithms.



Figure 47. IMX636 camera.



Figure 48. RK3588 computing board.

2.7.3 Results

To better display and implement AI algorithms, the event stream within a fixed time period is usually compressed into a frame. Through the cooperation of DVS and AI algorithms, we have implemented a real-time anti-blur human detection application.



Figure 49. Visualized results of human detection, numbers represent confidence values.

2.7.4 IMOCO4.E Requirements

REEXEN’s DVS sensor does not have officially-listed requirements. However, according to D3.3, following parameters and constraints will be of high importance in its construction in the scope of its potential application to Demonstrator 3:

Table 12: Requirements for the dynamic vision sensor.

Item	Requirement	Reality	Status
Resolution	1280x720 (~0.9Mpx)	1280x720	achieved
Field of view (FOV)	120°per image sensor		achieved
Pixel latency	< 1 ms @ 1000 lux < 10 ms @ 100 lux	0.1ms average	achieved
Low light scene	< 100 lux	5 lux	achieved
Backlight scene	> 100 dB	110dB	achieved
Forklift moving up to	50 km/h	50 km/h	achieved
NN processor efficiency	10-20 TOPS/W with int8 computations for both inputs and weights	20 TOPS/W	achieved
Interface (data output)	Not mentioned	USB 3.0	achieved
Camera Max.Bandwidth	Not mentioned	1.6Gbps	achieved
Camera power	Not mentioned	0.5W	achieved

2.7.5 Capabilities and Limitations

Dynamic vision sensor offers attractive properties compared to traditional cameras: high temporal resolution (in the order of μs), very high dynamic range (140 dB vs. 60 dB), low power consumption, and high pixel bandwidth (on the order of kHz).

Dynamic vision sensors cannot provide rich visual texture and color information like traditional cameras, and the price is more expensive.

2.7.6 Customizations and Adaptations

DVS can not only be applied to human detection tasks, but also can be used to detect more other objects using similar methods. This type of application is suitable for installation in robots, cars, and AR/VR products.

2.7.7 Methodology and Toolchains

In terms of tool integration, it is necessary to consider whether the computing power can meet the algorithm requirements and select computing equipment suitable for connection with DVS. We use Prophesee's software toolkit to complete data processing and denoising, which facilitates algorithm development.

3 Solutions of Task 3.3: SoC/FPGA platforms for smart control and signal processing (BB1)

3.1 Overview

3.1.1 Solutions of T3.3

BB1 (and some BB4) demonstrators of the SoC+FPGA and multi-many core platforms developed. These demonstrators or battery of tests will demonstrate the first tangible results of BB1 and BB4. In addition to that, each platform will have some sections with technology description (with pictures), main characteristics, functionalities and how they address requirements defined in WP3 (D3.2) and WP2 (D2.4). All these descriptions will be collected and included in this D3.4 report document.

These different BB1 and BB4 platforms are deployed in different P/UC/D and connect to other building block within the IMOCO4.E framework:

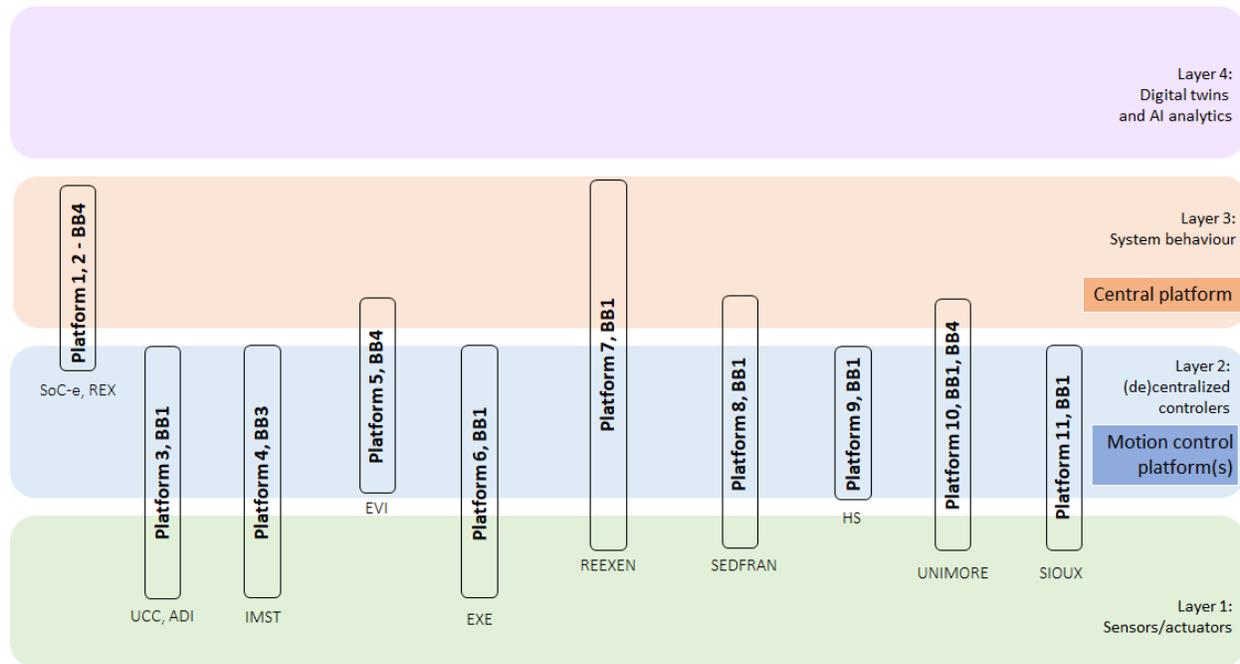


Figure 50. Mapping of the platforms with the project framework.

The table below shows the contribution of the platforms found in this document to the Pilots, Use Cases and Demonstrators of the project, in addition to indicating the main Building Blocks to which they belong, although there may be others that are left out of this task.

Table 13: Platform contribution to P/UC/D.

Platform ID	Involved Partners	BBs	Pilot	Demonstrator	Use Case
Platform 1,2	SOC-e, REX	4	3	-	-

Platform 3	UCC, ADI	1	-	-	3
Platform 4	IMST	3,10	-	3	-
Platform 5	EVI	4	2	-	-
Platform 6	EXE	1	5	-	-
Platform 7	REEXEN	1,8,10	4	3	-
Platform 8	SEDFRAN	1	1, 2	-	-
Platform 9	HS	1	-	-	-
Platform 10	UNIMORE	1,4	3	2	-
Platform 11	SIOUX	1	2	-	-

3.1.2 Addressed ST objectives and KPIs

BB1 contributes mainly to ST2, which concerns the development of the smart Instrumentation Layer gathering and processing visual and sensor information from supplementary instrumentation installed on the moving parts of the different systems.

BB1 KPIs:

Usability level.

- SoC-e and SEDFRAN platforms includes SW developments to facilitate a configuration interface for engineers not familiar with e.g.FPGA development.
- HS allows AutoFlow users to use AutoKeras for model generation, improving the overall workflow.

Communication interface performance.

- REX algorithm is capable of providing valid real-time results.
- SoC-e platform provides communication interfaces on copper and fibre optics, supporting combined line speeds of 1G and 10G.

Utilization of asymmetric multiprocessing.

- UNIMORE solution presents multiple sensors with data flows of different natures and priorities managed.

Portability across multiple FPGA vendors

- REX developments allowed to transfer their algorithms to more general FPGA boards for standard PCs (or Industrial PCs).

Platform for centralized TSN data management

- SoC-e and SEDFRAN proposals supports TSN IP implementations with a high number of communication ports.

HW/SW modules for bounded latency communications

- Tests carried out by UCC and ADI showed the latency of the program when it is run on an STM ARM Cortex-M7 core is about 0.5ms, meaning the program module decodes the received message in less than 0.5ms.
- The PCI card developed by EVI has been designed to achieve low-latency NN inference for edge scenarios.
- SoC-e and SEDFRAN platforms have been developed following the 802.1Qbv standard which is fundamental for maintaining predictability, low latency, and determinism in environments where timing is critical.
- The powerful and rugged embedded IO-controller platform developed by EXERTUS is targeted to accomplish the requirements of applications working in harsh conditions and requiring high performance, versatility, reliability, and safety.

3.2 TSN platform based on FPGA (SoC-e)

3.2.1 Technology overview

Inside Pilot 3, SoC-e has developed a modular platform with cutting-edge (real-time) communications and processing capabilities, with sensor acquisition capabilities as well, supporting highly customizable functionalities in all those aspects.

This platform will accommodate many of the requirements derived from the “BB4 Real-Time smart Control Platform”: compute-intensive workload, real-time requirements and advanced communications and will be developed on the basis of the following specifications: the proposal supports TSN IP implementations with a high number of communication ports, offers the capability to use copper and fibre optic physical communication interfaces, supports combined 1G and 10G line rates, facilitates the implementation of software and hardware computing units for the execution of AI algorithms, and supports conduction heat dissipation to avoid the use of fans and ensure operation over a wide temperature range.

FPGA based multi-protocol and multi-port platform (HW-016) is going to be used as a central processing and bridging unit into the Pilot3. There are going to be two additional HW platforms (HW-005 and HW-003) gathering info from sensors and cameras and performing some initial processing algorithms over that data. FPGA based platform (HW-016) is going to bridge the pre-processed data, as an Edge element, to the Fog, for an ulterior processing based on a Kafka system. That Edge-to-Fog communication is going to leverage TSN in order to assure a disciplined traffic ordering and timing closure. Additionally, and taking benefit of the location of this FPGA based platform on the data value chain (access to the camera images), several FPGA aided artifacts are going to be tested with the aim of accelerating standard image processing AI algorithms.

3.2.2 Implementation aspects

A latest generation Zynq Ultrascale+ MPSoC SoC reconfigurable platform has been designed that incorporates a high capacity FPGA section. In this section the TSN IP would be implemented up to a maximum of 32 ports, which can be distributed to external communication ports or internally to the processor section (CPUs and GPUs) internal to the reconfigurable platform.

In order to offer the ability to use physical copper and fibre optic communication interfaces and to support combined 1G and 10G line speeds, two systems have been incorporated into the electronics. On the one hand, a block of high density RJ45 copper ports capable of providing 8x (+ 8x9 external 1G communication

ports (in figure "B2"). On the other hand, a set of 4x SFP+ device *cages* capable of accommodating 10G Fibre Optic communication transceivers (in figure "B3"). Additionally, to provide Fibre Optic communication capacity in particularly demanding environmental and mechanical environments, 4x ruggedised 10G fibre transceivers have also been incorporated.

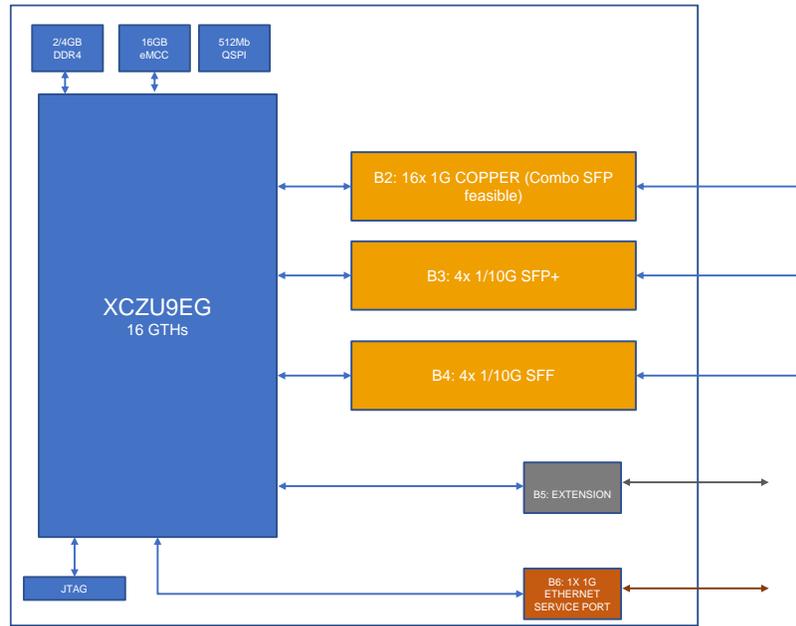


Figure 51. General blocks diagram.

The implementation of software and hardware computing units for the execution of user-specific services (e.g. AI algorithms, cybersecurity services, etc.) can be offered thanks to the internal composition of the reconfigurable Zynq Ultrascale+ MPSoC SoC platform. As shown in the figure below, the non-reconfigurable section integrates 4x ARM Cortex-A53 processors and 2x Dual-core ARM Cortex-R5 CPUs, 1x Mali™-400 MP2 GPU. The proposed memories are a high-capacity DDR4 oriented to support high-end applications and operating systems and non-volatile QSPI and eMMC memories that would complete the needs of the embedded systems. In the FPGA section, the communications IPs have been implemented as indicated in the previous section.

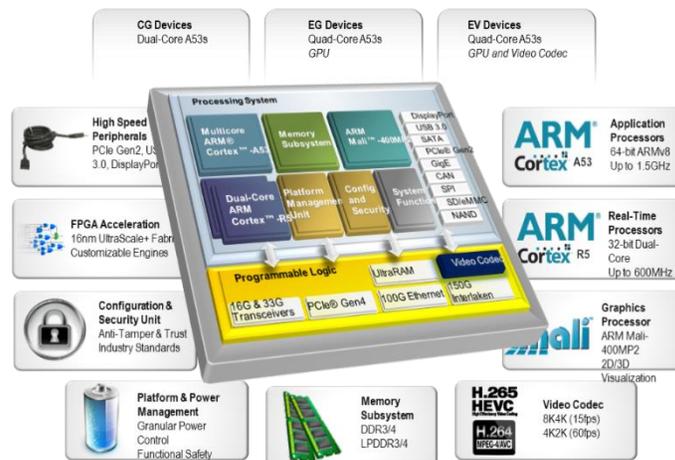


Figure 52. Zynq Ultrascale+ MPSoC reconfigurable platform block diagram.

The block diagram depicted in the figure below details the proposed incorporation of 4 circuits for demultiplexing the high-speed interfaces coming from the reconfigurable platform in order to satisfy the 8x direct RJ45 copper ports with a possible expansion of 8x ports to another auxiliary board.

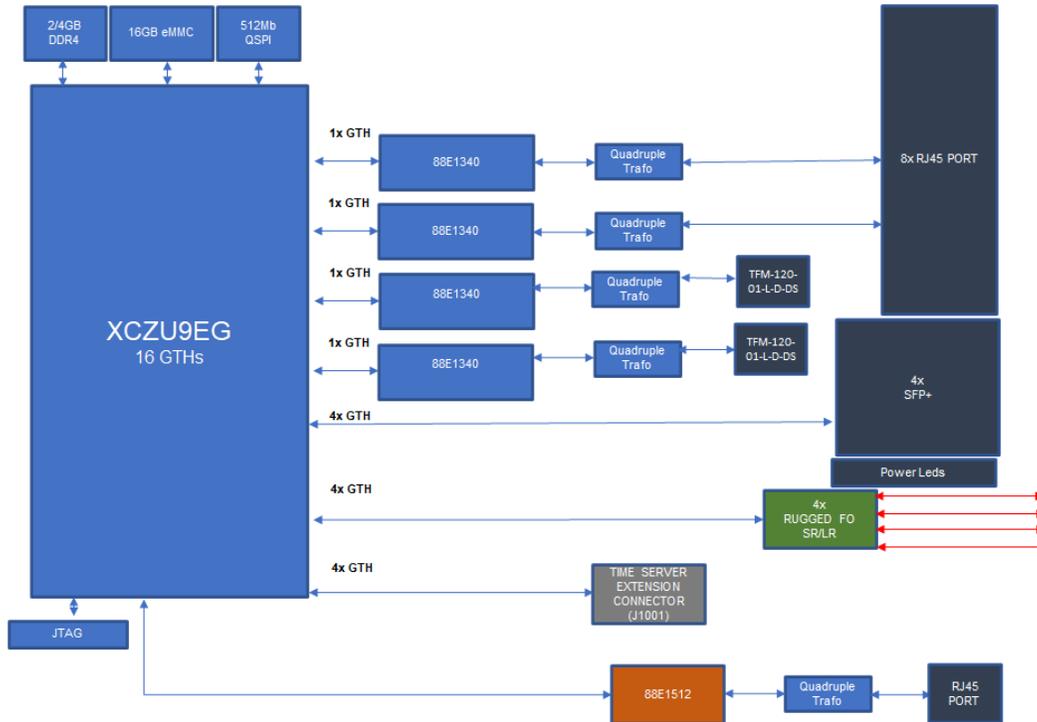


Figure 53. Proposed communication interfaces.

Below we show the 3D of the PCB:

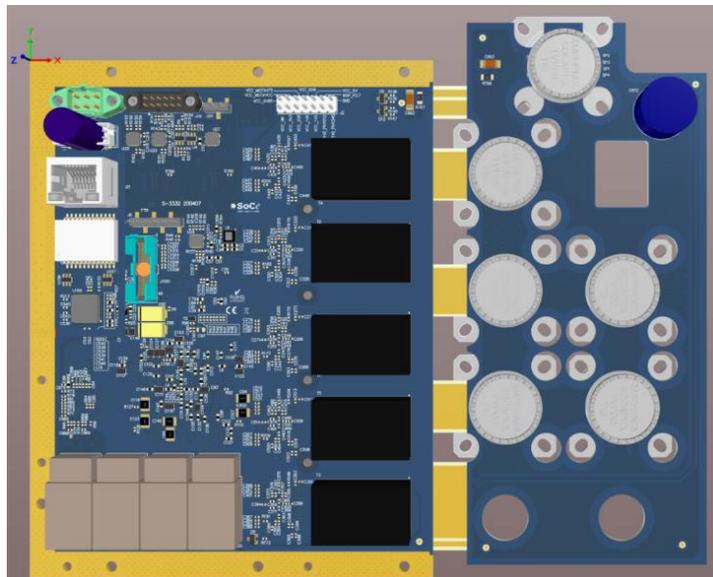


Figure 54. 3D PCB.

3.2.3 Results

We have developed a HW platform capable of supporting TSN IP implementations with a high number of communication ports, offering the ability to use physical communication interfaces on copper and fibre optics, supporting combined line speeds of 1G and 10G, facilitating the implementation of software and hardware computing units for the execution of AI algorithms and supporting conduction heat dissipation to avoid the use of fans ensuring operation over a wide temperature range.

A comprehensive set of tests has been designed to test the integration of IP Core TSN with the developed HW platform. The tests can be grouped into the following large blocks, each of which will in turn be broken down into a set of mini setups to test specific functionalities and features. All the tests carried out have been satisfactory, so by way of example we show a set up relating to the 802.1AS protocol (relating to one of the main requirements to be met by the developed platform).

- DEVICE BASIC ACCESS
- FPGA CONFIG COMMAND LINE INTERFACE
- WEB INTERFACE
- ETHERNET SWITCHING
- VLAN MANAGEMENT
- DYNAMIC FILTERING DATABASE
- STATIC FILTERING DATABASE
- FILTERING DATABASE QUERY
- QUALITY OF SERVICE (PRIORITIES)
- EDSA TAGGING
- IGMP SNOOPING
- REDUNDANCY RSTP
- REDUNDANCY MSTP
- IEEE 802.1AS
- IEEE 802.1Qbv
- IEEE 802.1Qav
- Netconf
- SYNCHRONIZATION NTP
- SECURITY
- LLDP
- SYSLOG
- RECOVERY & FIRMWARE UPDATE

Example: IEEE 802.1AS - Time Aware Bridge DUT as slave and multiple slaves connected to the DUT

Purpose:

To verify that the device under test (DUT) implements correctly Time Aware Bridge functionality defined in IEEE 802.1AS. The goal of the test is determine the correct functionality of the DUT as slave connected to a Master while in the meantime different slaves are connected to the DUT .

References:

- [1] IEEE Std.802.1AS-2011

Discussion:

None.

Test Setup:

Connect the Testing Stations to the DUT as shown below.

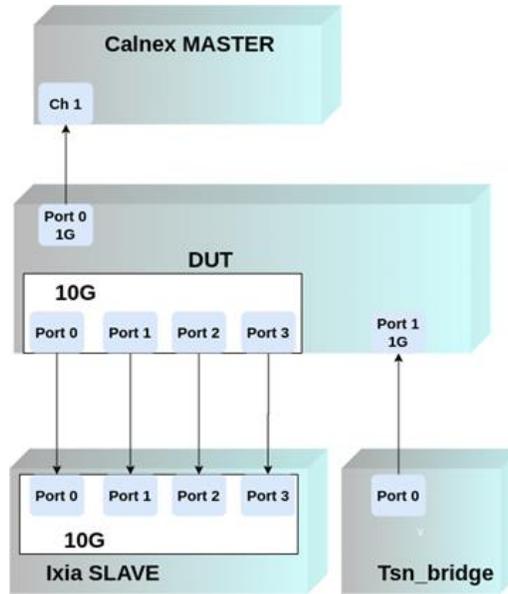


Figure 55. Time Aware Bridge error DUT as slave and multiple slaves connected to the DUT.

Observable Results:

1. On Ixia → check that all PTP profiles are green and that the status is “slave”

Grouping	Topology	Device Group	Device#	Status	Session Info	PTP State	Master Clock ID	Clock Role	Custom Clock ID
PTP 1: 4 ports	Topology 1	Device Group 1	4	4 of 4 Up				Slave	
Ethernet - 001	Topology 1	Device Group 1	# 1	Up	Multiple Pee...	Slave	0x00:00:00:00:00:00:01	Slave	
Ethernet - 002	Topology 1	Device Group 1	# 2	Up	Multiple Pee...	Slave	0x00:00:00:00:00:00:01	Slave	
Ethernet - 003	Topology 1	Device Group 1	# 3	Up	Multiple Pee...	Slave	0x00:00:00:00:00:00:01	Slave	
Ethernet - 004	Topology 1	Device Group 1	# 4	Up	Multiple Pee...	Slave	0x00:00:00:00:00:00:01	Slave	

Figure 56. Time Aware Bridge error DUT as slave and multiple slaves connected to the DUT-IXIA results.

2. On Calnex, all slave should be listed (in the image below only Ixia is shown, also DUT and TSN_bridge should be listed).

Multicast / Peer / Slave Address	pdReq Rate	Sync Rate	Ann Rate
01 80 c2 00 00 0e	1	8	1
00 11 01 00 00 01	M		
00 11 04 00 00 01	M		
00 11 03 00 00 01	M		
00 11 02 00 00 01	M		

Figure 57. Time Aware Bridge error DUT as slave and multiple slaves connected to the DUT-Calnex results.

3. On both DUT and TSN_Bridge on the synchronization PTP screen, the PTP pile should be running and the Calnex should be reported as grandmaster (gm.identity: 00:00.....:01)

The platform has also tested within Pilot 3. It has been connected to the others devices as the Atlas platform (HW-004), Nvidia Xavier (HW-003), Xilinx platform (HW-004) and Fog Server. In this configuration the following tests has been done:

- The connectivity between all the HW devices through SoCe’s TSN switch has been successfully tested. Bidirectional communication between all the devices have been proven to work correctly.
- For the SoC-e’s TSN switch, the configuration of the Credit Based Shaper has been successfully tested. Different bandwidth reservation has been applied to each of the data flows and the results have been successful.

3.2.4 IMOCO4.E Requirements

Table 14: HW-016 verification results table

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
R1-D4.2-P3	BB4 should provide TSN support	T	TEST-056	PASS	Connection’s bandwidth has been measured among the devices.
R073-D2.3	IEEE 802.1Q Mixed-critical communication support.	T	TEST-056	PASS	Several data flows with different priorities haven been sent among devices.
R076-D2.3	Time Sensitive Network data plane will expose telemetry monitoring and configuration to a centralised TSN data management.	D	TEST-056	PASS	The configuration and monitoring of SoC-e's TSN switch has been tested using a remote management protocol.
R078-D2.3	Sub-µs time synchronisation based on IEEE 1588 and IEEE 802.1AS.	T	TEST-056	PASS	The synchronization among devices has been correctly tested using 802.1AS protocol.
R079-D2.3	End to End deterministic latency for time-constrained TSN data streams.	T	TEST-056	PASS	Connection’s latency has been measured among devices.

3.2.5 Capabilities and Limitations

There are many high-performance FPGA-SoPC evaluation platforms on the market, as well as SoM modules based on high-performance FPGA-SoPCs, but there is no platform that supports such a high number of Ethernet ports, nor such a high heterogeneity in terms of physical medium (Cu or FI), and/or line speed (100Mbps/1Gbps/10Gbps).

3.2.6 Customizations and Adaptations

This platform has also been validated within Pilot 3 and during the integration sessions we have performed, the connectivity between all the HW devices through SoCe’s TSN switch has been successfully tested

The developed platform can transcend to the end-equipment category with a certified high-performance mechanical enclosure. This would allow to:

- Transcend beyond what is an FPGA-based evaluation platform in use, and offer a 'final equipment', with custom-designed mechanics (other times standard mechanics have been used, for which the electronics have been adapted, with the penalty that this entails in terms of mechanical dissipation, connector layout, etc.).
- Thermal performance appropriate to the level of consumption required by the FPGA in applications with this number of ports, line speeds and performance.

3.2.7 Methodology and Toolchains

During the design and development phases the following tools have been used:

For the implementation of code within the TSN nodes, Vivado tool has been used for low-level FPGA programming using VHDL language.

During the integration of SW + IP Core (FPGA), the programmable logic part of the SoPC has been configured with a bitstream obtained by making a project with the Xilinx Vivado tool.

During the testing phase, the following tools has been used:

- IXIA and Albedo equipment for Ethernet tests.
- Calnex equipment for PTP-1588 synchronisation tests.
- RELYUM Lab-Tool equipment for capturing traffic and timestamps.
- Wireshark application on high-performance PCs for traffic capture and analysis.

3.3 Motion smart sensor (REX)

3.3.1 Technology overview

This solution focuses on smart sensing of motion, acceleration, velocity and using standard (imperfect) incremental encoders. The sensor data processing is based on real-time algorithms which corresponds to "BB4 Real-Time smart Control Platform".

It consists of two parts. The first part of algorithms is implemented in FPGA, which provides communication with input sensors and gather set of information required for output processing. The second part of the subsequent algorithms are operated in the REXYGEN system under Linux operating system. In REXYGEN system tool, running on SoC processor, enables presentation of processed data and their usage in control tasks of the system.

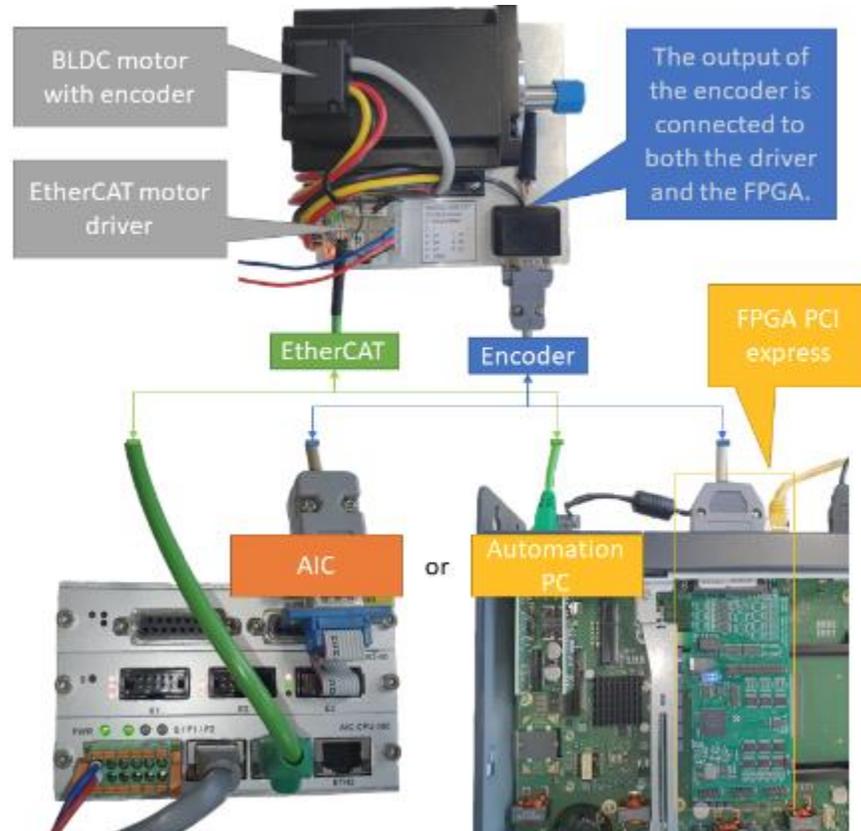


Figure 58. Solution overview.

3.3.2 Implementation aspects

FPGA algorithms have been developed on the AIC (Advanced Industrial Controller) based on Intel Cyclone V SoC. It is an Industrial-grade modular hardware developed by REX Controls. This controller was earlier used for different tasks in the I-MECH project. Development in the IMOCO4.E project allowed to transfer these algorithms to more general FPGA boards for standard PCs (or Industrial PCs). AIC system uses a digital input I/O card. This card is supported in the FPGA by a MOSS IP core that processes decoding the state of the incremental sensor, and also accurate timestamps with 1ns resolution to support the MOSS processing. The algorithm is processed in a functional block of the REXYGEN system. This block provides estimation of acceleration, velocity and position on an equidistant time based on augmented Kalman filter.

The system is designed to support general logic array and processor architecture, specific SoC is not necessary. The functionality was also tested on REXYGEN system running on an industrial PC.

Postprocessing of the FPGA data is provided by specially developed function block MOSS (MOtion Smart Sensor) in REXYGEN. The configuration of the control system in REXYGEN Studio is shown in Figure 59. Configuration of the control system in REXYGEN Studio.. The FPGA_Read_ENCODER block ensures loading values from the MOSS IP Core of the FPGA via system bus. The MOSS block performs software calculations and provides results for real-time control and visualization.

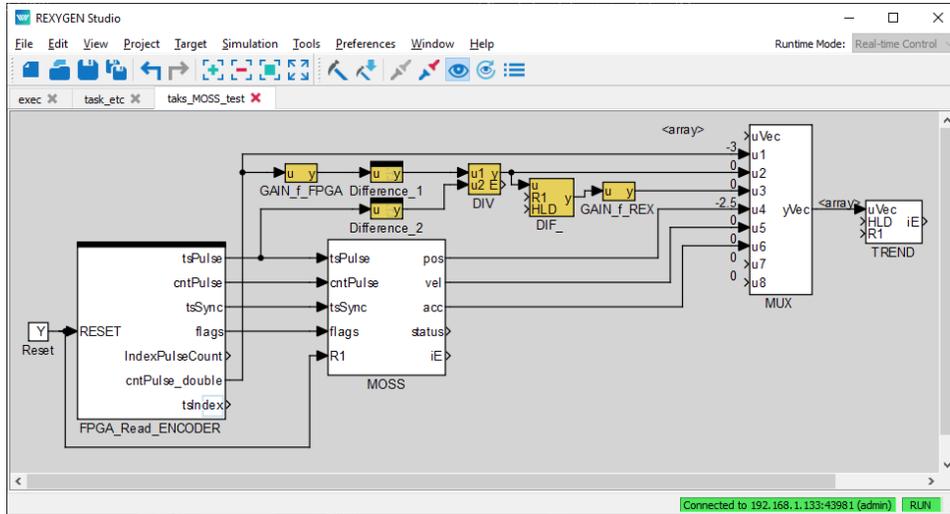


Figure 59. Configuration of the control system in REXYGEN Studio.

3.3.3 Results

Measurements on a demonstration device confirmed that the algorithm is capable of providing valid real-time results. The velocity and acceleration values contain lower noise than the values calculated in the standard way from the position difference. The results are applicable for system analysis and control.

This method of calculation is mainly used in:

- Rotation velocity uniformity monitoring - vibration detection from rotational acceleration values
- Shaft torsion and torsional vibration monitoring - torsion = difference between instantaneous position at two shaft points
- Shaft axial movement measurement -by multiple (2-3) sensors around a single toothed gear
- Repetitive control

A comparison of the results of position, speed and acceleration on BLDC motor for 10 rot./sec. calculated using the standard method and the MOSS algorithm is shown in the following figure.

The measurements were performed on an AIC system containing an ARM A7 at 800MHz. This allowed a processing period of 1 ms.

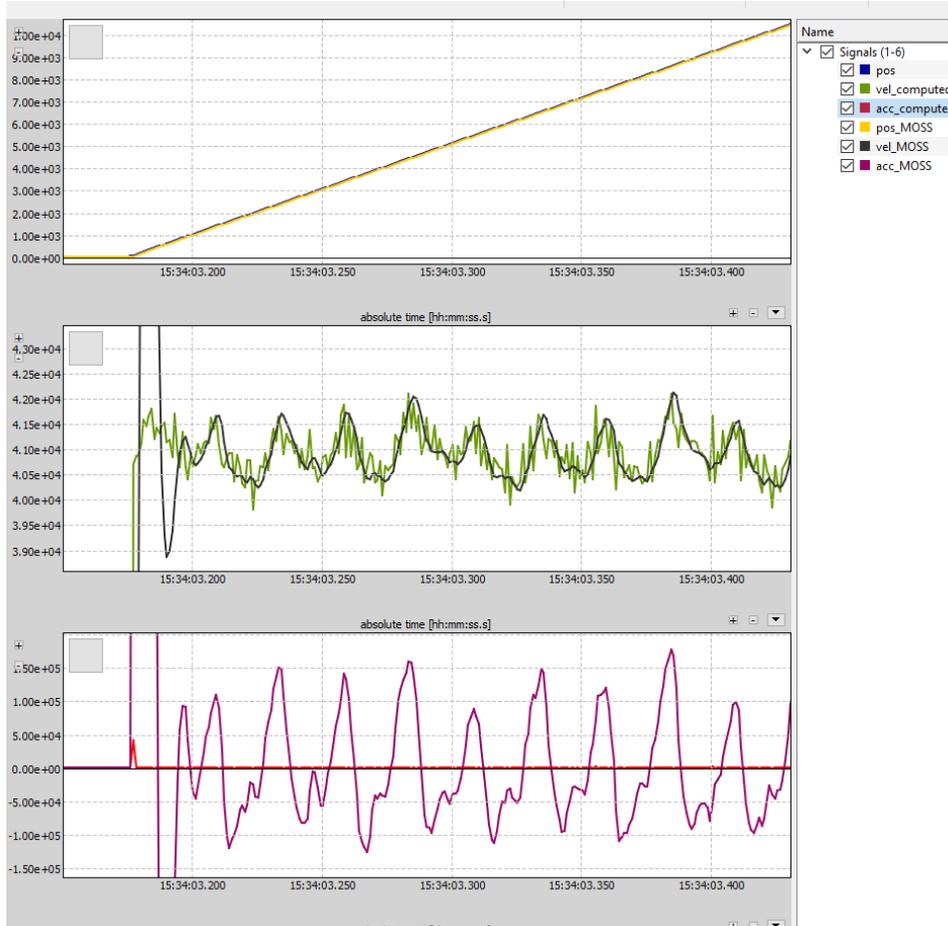


Figure 60. Comparison of MOSS and standard calculation results.

3.3.4 IMOCO4.E Requirements

R071-D2.3 BB1 shall support standard and vendor-neutral Wired 1G Ethernet. Target to User interface connected by GBE EMAC of SoC. This interface supports continuous system monitoring

R081 SoC Intel Cyclone V SE

3.3.5 Capabilities and Limitations

Our solution brings significant accuracy to position, velocity and acceleration measurements by eliminating the inaccuracies of standard rotation sensors using advanced algorithms. This is beneficial in the analysis of rotating machines as well as in control tasks.

The use of advanced algorithms introduces the need to adapt Kalman filters to the dynamics of the system on which the measurements are made. This adjustment of the system configuration is necessary to obtain appropriate results.

In terms of hardware, the described solution is a combination of an FPGA part containing the necessary data capture and its processing by an algorithm in the processor part of the system. For this configuration, it seems optimal to use an SoC that integrates both of these components. However, a separate solution has also been tested, using a PCIe bus for data transfer between the FPGA and the processor. This solution

offers the possibility of a wider choice of usable hardware, especially if higher computational demands are required for the processor part.

3.3.6 Customizations and Adaptations

As mentioned above, the solution introduces the necessity of parameterizing the Kalman filter to account for the dynamics of the system. This allows the system to be adapted for different applications with a wide range of processed values.

REXYGEN is provided for various processor architectures. The processing period has to be chosen taking into account the performance of the platform used.

MOSS IP in FPGA occupies approx. 1000 ALMs, which limits the choice of possible FPGA chip alternatives.

3.3.7 Methodology and Toolchains

For development of the FPGA part, Intel Quartus Prime Design Software was used.

3.4 Advanced wireless embedded processing platform (UCC, ADI)

3.4.1 Technology overview

The Use Case 3 (UC3) Tactile Robot Teleoperation Platform has been discussed previously as part of the ADI ToF in Section 2.5. In order to understand the embedded processing aspects of UC3 the below presents a technical overview of the UC3 IMOCO4.E architecture. UC3 has both a local sensor end architecture (where the user will operate) and remote robotic systems end architecture.

Use Case 3 Architecture - Local Sensor End Overview:

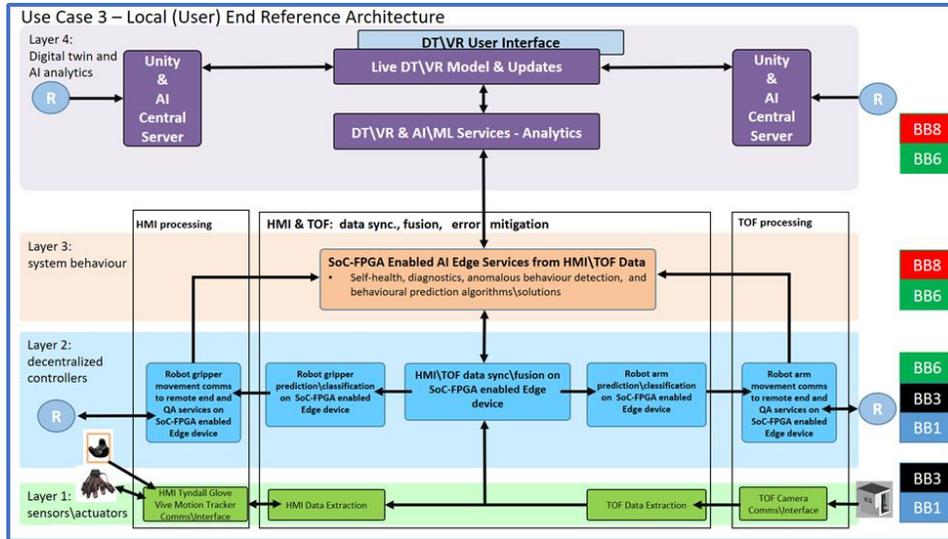


Figure 61. Use Case 3 - Local Sensor End Architecture.

With main focus on the sensor layer, Layer 1 sensors/actuators (L1) provides the interfaces to the HMI (Tyndall Glove, Vive Motion Trackers) and the Time of Flight (ToF) sensor devices deployed at L1. Data is extracted from all sensors and is passed on via the L1 – L2 interface for applied processing. The above figure also provides an overview of four Building Blocks (BBs) that apply across the four layers of the UC3 IMOCO4.E architecture.

Use Case 3 Architecture - Remote End Overview:

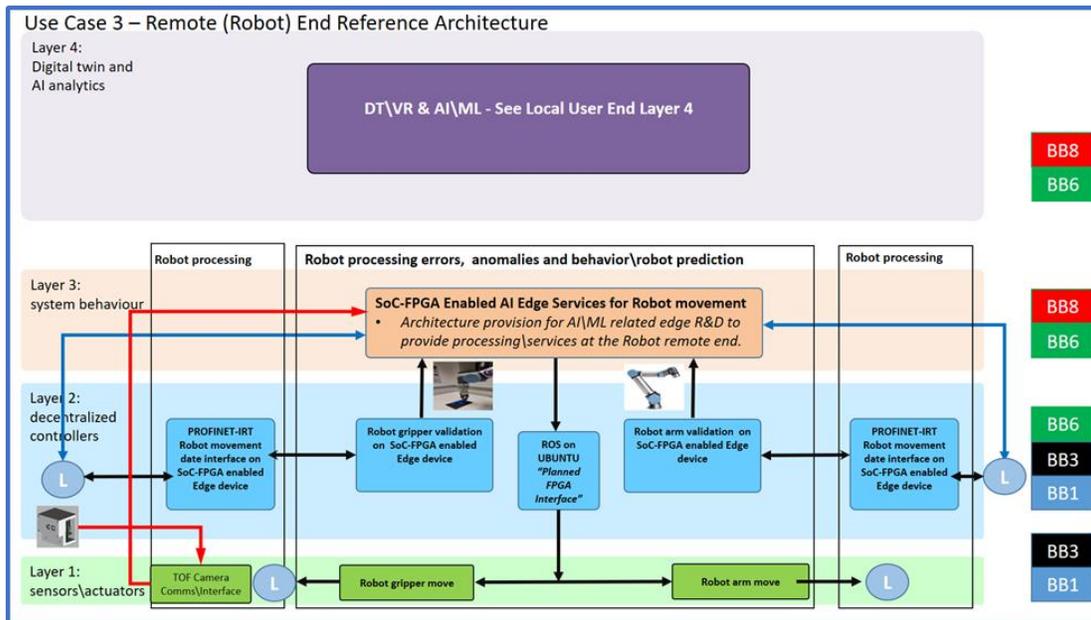


Figure 62. Use Case 3 - Remote Robot End Architecture.

Robot movement coordinates are received at the remote edge device at Layer 2 (L2) of the architecture. L2 is responsible for receiving and sending data with the L2 at the local end of the UC3 platform. L2 has two validation modules for the arm and gripper coordinates which are interfaced with Layer 3 (L3). Robot movement coordinates received at L3 are the inputs to a set of AI and ML services that can be engineered and deployed at the remote end. In particular, the use of ToF technologies at the remote end of the platform will provide future potential to embed vision based detection capabilities to the platform and will offer AI powered object and robotic movement recognition functionality. L3 interfaces with L1 which actually forms the interface with the robot arm and gripper. Here the movement coordinates are then used to physically drive the robot for the next step in the task the user is driving remotely from the local end of the UC3 platform.

3.4.2 Implementation aspects

Use Case 3 BB1 Edge to Edge Software Architecture

This section presents the edge-to-edge architecture for UC3 from a software perspective.

Sensor Local End – BB1 Edge Software Architecture:

The ToF camera is currently connected to the PolarFire through a PC, which is also used for the Digital Twin (DT) engine in addition to the interface for ToF. Unity is used as the DT engine and both the robot and the finger gripper have been modelled as part of WP5 Digital Twin research and development (Please see deliverable D5.8). The PC, PolarFire, and the Nano router in the below figure are connected over an Ethernet network through a simple unmanaged network switch. In order to connect the local-end edge devices to the remote-end edge devices, a multi-protocol industrial Ethernet module (Rapid) is connected to the STM32/PolarFire as the network interface.

Robot Remote End – BB1 Edge Software Architecture:

On the remote end, there are several components making up the teleoperation system which should be controlled, including the 3-fingers gripper and the UR16e robot by the edge devices (STM32/PolarFire SoC-FPGA). There is also provision for a second ToF camera, which can be used for object recognition and a PC as the interface for the PolarFire to be connected to the ToF. All of these components are connected to an Ethernet network through a simple managed network switch. To connect the remote-end edge device to the local-end edge device, a multi-protocol industrial Ethernet module (Rapid) is connected to the STM32/PolarFire as a network interface.

UC3 BB1 Edge to Edge Software Architecture Futures:

As UC3 developed throughout the life of the IMOCO4.E project the edge research and development incorporated various components. In particular the STM32 has become an additional component to compliment the MicroChip PolarFire SoC FPGA. Future research will incorporate the integration and testing of the ToF device and algorithms on the PolarFire. This work will also involve the testing and optimisation of various embedded AI algorithms on the PolarFire. With the addition of the STM32 research will also be conducted in relation to integration with the PolarFire at both the local and remote ends of the UC3 platform.

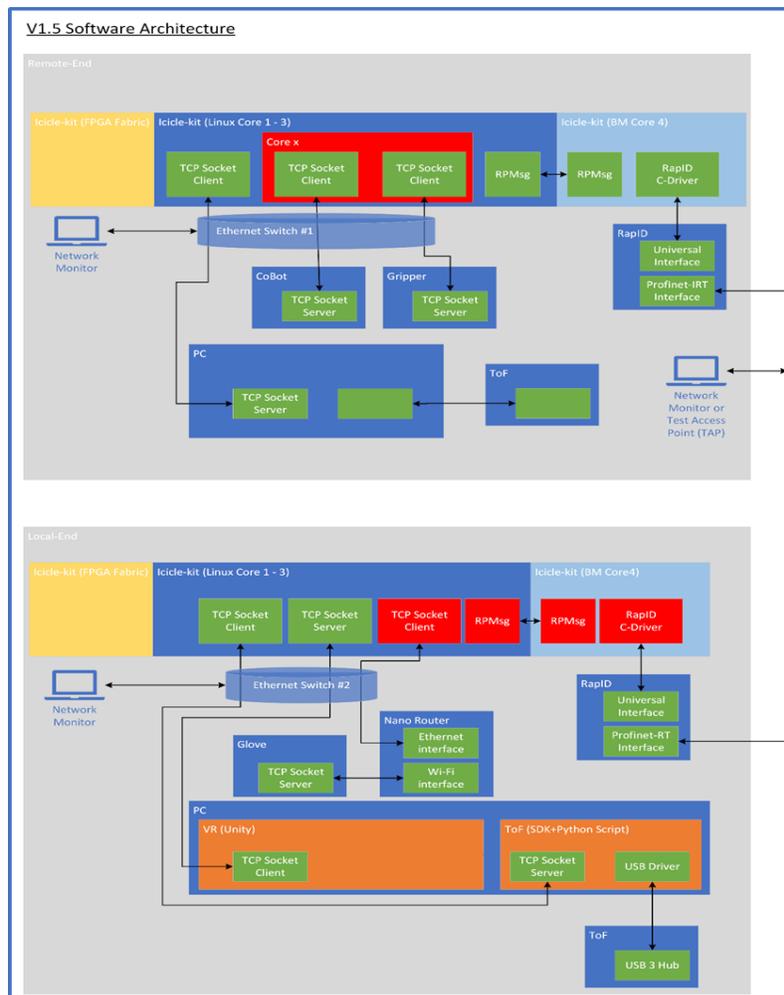


Figure 63. Use case 3 - Edge to Edge SW Architecture.

Setup of UC3 BB1 Related Hardware

The UC3 BB1 Sensor/Actuator layer hardware (HW) has been described in a number of WP3 and WP6 deliverables. The below provides an update and general overview of the HW at both the local and remote end of the platform.

As per the below figure the local end of the platform primarily connects the sensor/actuator layer with the STM32 and Polarfire edge devices represented in the centre of the figure. The current iteration of the local end hardware provides for the connectivity of the Tyndall tactile glove, Vive motion trackers and ToF depth camera technologies. The sensor devices are connected with the edge devices via a network switch. The Profinet Industrial Ethernet is connected to the edge device using the Analog Devices (ADI) RapID module. The PLC: Siemens S7-1500 CPU acts as the network manager for the connectivity across the local and remote ends of the platform.

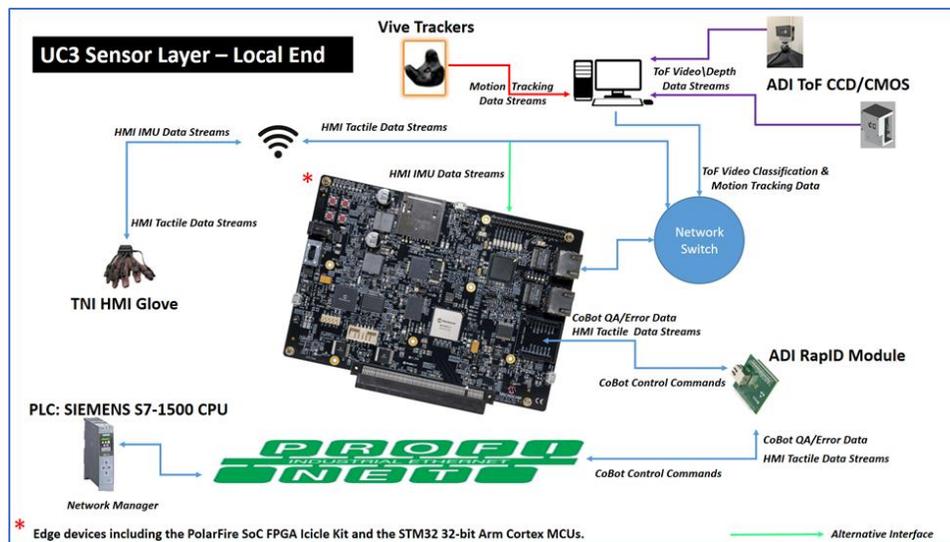


Figure 64. Use case 3 Sensor/Actuator Layer Local End Hardware.

Use Case 3 BB1 Sensor/Actuator Layer Remote End Hardware:

The robot remote end hardware is represented in the below figure. The edge device at the remote robot end will receive data streams for embedded processing in order to move the robot arm and gripper. The network switch is used to connect the UR16e robot components along with the connectivity to the additional remote ToF camera. As in the local end, the Profinet Industrial Ethernet is connected to the edge devices using the Analog Devices (ADI) RapID module. The PLC: Siemens S7-1500 CPU acts as the network manager for the connectivity across the local and remote ends of the platform.

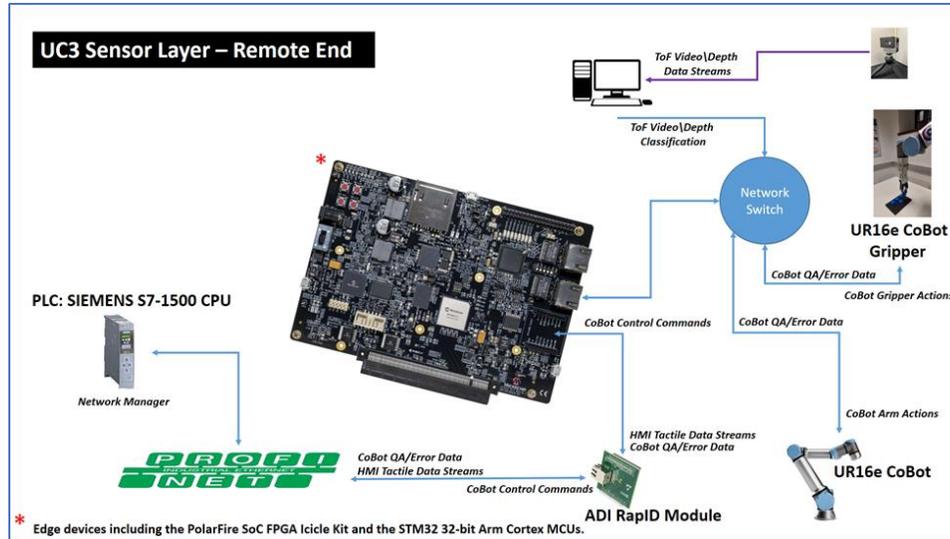


Figure 65. Use Case 3 – Sensor/Actuator Layer Remote End Hardware.

Note: Due to the evolving nature of this research, the UC3 technical partners reserve the right to alter and modify various aspects, configurations, components of the described platform throughout the lifetime of the IMOCO4.E project.

3.4.3 Results

The following section primarily presents an overview of the BB1 related UC3 testing that has been conducted in relation to the edge-to-edge processing.

BB1 - Proof of concept sensor/actuator layers at the local and remote ends: This test was developed at the Tyndall laboratories in conjunction with inputs from all UC3 partners. The results have been a proof-of-concept make-up of both the local sensor end and the remote robot end of the platform. This work incorporated both sensor and robot integration activities. As part of the test activities, the concept of developing a DT of the robotic teleoperation process was also investigated and developed.

Results: The UC3 infrastructure was tested in-house by a number of researchers in an in-formal capacity in order to evaluate the ease of use and functionality of the various sensors used and the accuracy of the robot movements. A video relating to the test results is available and is part of the IMOCO4.E WP8 education modules. Please refer to Task 8.3 module on the “Internet of Skills” for further details.

BB1 - Edge to edge connectivity and latency: This test involved the characterisation of program latency for the RapID C-Driver program module (SW-066) on the Edge device. The RapID (HW-036) is the multi-protocol real-time industrial Ethernet interface module for the IMOCO4.E UC3. The main driver for this module must be very light to be run on micro-controllers cores. Otherwise, the edge device is not able to satisfy the real-time requirements of a real-time industrial communication protocol. Therefore, the latency of the RapID C-Driver program module on the edge device is characterised to ensure the developed program module meets the real-time requirements.

Results: The result of the test showed the latency of the program when it is run on an STM ARM Cortex-M7 core is about 0.5ms, meaning the program module decodes the received message in less than 0.5ms. Therefore, this program module satisfies the real-time requirements of Profinet-RT (HW-036), the real-time industrial Ethernet communication protocols used for UC3.

BB1 - Characterisation of Edge-to-Edge communication latency: Edge-to-Edge communication is one of the main aspects of UC3 and according to system requirements, the latency in this communication should be very small and it should also be deterministic. To validate this communication open-source software will be used to monitor, record, and characterise the traffic on this real-time Ethernet-based industrial network. A set of specific tests will be developed and reported on during the lifetime of the UC3 project.

Results: To be reported in WP6.

BB1 - Characterisation of program latency for the inverse kinematic calculation program module (SW-078) on the Edge device: The data collected on the sensor layer at the local end from the tele-operator (user) are the 3D positioning of the operator’s wrist in the ToF (HW-030) camera’s coordination frame. This 3D position should be mapped to the robot’s (HW-032) coordination frame. Therefore, the latency that running this processing module introduces to the system needs to be characterised to ensure this latency meets the real-time requirements of both UC3 and the robot.

Results: To be reported in WP6.

3.4.4 IMOCO4.E Requirements

The below provides a BB1 requirements update in the context of UC3.

Table 15: Requirements for wireless embedded processing platform

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
Interfaces and connectivity					
R070-D2.3	The interfaces to BB1 shall be an industry standard.	I	TEST-045	IN PROGRESS	Such standards are applied where possible in UC3.
R072-D2.3	BB1 shall support standard API between application controller and distributed edge instance, exposing local configuration and telemetry.	T	TEST-045	IN PROGRESS	Industry standards such as Profinet are applied in UC3.
R074-D2.3	The BB1 should have interface with camera sensors.	T	TEST-046	IN PROGRESS	Part of UC3 layer #1 and local and remote ends.

Req-D3.2-U3-1-com	Connectivity of the communications interface for the PolarFire SoC-FPGA (MPFS250T-FCVG484EES) placed at the local user-end with the PolarFire SoC-FPGA (MPFS250T-FCVG484EES) placed at the remote CoBot-end.	I-D	TEST-045	IN PROGRESS	STM32 are currently in use for edge-to-edge connectivity with the plan to interface the STM32 edge devices with the PolarFire edge devices.
Req-D3.2-U3-2-com	Driver engineering for the RapID-NI-V2007 to interface the PolarFire SoC-FPGA (MPFS250T-FCVG484EES) with the PROFINET-IRT.	I-D	TEST-045	IN PROGRESS	As advised above this interfacing has initially taken place using the STM32 edge devices.
Req-D3.2-U3-3-com	SIEMENS S7-1550 for the integration of new devices, time synchronization between devices and reading the messages (data packets) from devices at pre-defined cycle times.	I-D	TEST-045	IN PROGRESS	The SIEMENS PLC has been integrated into the UC3 edge to edge architecture.
Reliability (fault tolerance, availability)					
Req-D3.2-U3-1-hw-sw-com	The platform should have in-built fault tolerant capabilities, be able to recover from user or robot errors and fail gracefully as relevant.	I	TEST-045	IN PROGRESS	This is a core aspect of UC3 engineering and is continually being addressed.
Portability (adaptability, replaceability)					
R083-D2.3	BB1 shall offer ways of working and toolchains that extend across multiple (generations of) platforms.	D	TEST-045 TEST-046	IN PROGRESS	Applied as UC3 work with two edge devices now (STM32 and PolarFire)
Req-D3.2-U3-1-hw-sw-com	General requirement(s) that the use case is engineered with the potential for portability and adaptability to varying form of industrial tele-operation tasks for the enablement of remote factory/production working in the future.	I	TEST-045	IN PROGRESS	Accepted and applied throughout. Also see next section.
Scalability					

R085-D2.3	B1 shall offer a scalable amount of computational resources, e.g. by means of the firmware implementation or by offering a family of processing units with different capacities.	D	TEST-045	IN PROGRESS	UC3 currently using the STM32 and PolarFire edge devices in terms of scalable resources.
Req-D3.2-U3-1-hw-sw-com	The use case three engineered solution could be able to scale to different users being able to use the platform and also to enable the users to work across different robots.	I	TEST-046	IN PROGRESS	This is a long-term goal of UC3 and a core focus as the platform will develop in the future.
Tools/toolchains					
Req-D3.2-U3-8-sw-com	Generally available APIs and related SDKs (C/C++, Python) to interface the remote PolarFire SoC-FPGA (MPFS250T-FCVG484EES) edge device with the UR16e CoBot and the UR16e finger gripper device.	I-D	TEST-045	IN PROGRESS	Selected APIs and SDK are used throughout the UC3 engineering activities.
Safety					
Req-D3.2-U3-1-hw-sw	Continually addressing the health and safety aspects of the functionality to be implemented at both the user local end and the remote robot end of the tele-operation platform.	I-D	TEST-046	IN PROGRESS	Safety of the user/operator is critical to the design and engineering of the UC3 platform.

3.4.5 Capabilities & Limitations

Unique Selling Point: Research conducted is specifically addressing latency aspects at the edge with reference to sensors, edge to edge communications and remote robotic teleoperations. Built on the IMOCO4.E architecture and foundation BBs, UC3 has the potential to develop into a highly-engineered robotics teleoperations platform, operated through a sophisticated DT/VR/XR environment for the Internet of Skills and the Internet of Robotic Things.

Strengths & weaknesses: UC3 edge to edge infrastructure is originally built on industrial protocol Profinet/RapID and availability of related expertise is limited. This is one of the main weaknesses but with future engineering a move to TSN may be beneficial for the overall advancement of UC3. One of the major

strengths of the UC3 research has been the focus on embedded AI in terms of addressing latency in the sensors and at the edge-to-edge communications, local and remote ends.

3.4.6 Customizations & Adaptations

The below presents UC3 partners thoughts on how selected component(s) may have customisation potential across other IMOCO4.E projects (P/D/UC) or in other industrial applications outside the project.

Use in the P/D/UC: The edge-to-edge devices are connected using the RapID (ADI) and Profinet (Siemens) industrial communications infrastructure to communicate sensor generated signal data across the UC3 platform such that a robot can carry out a user guided task at the remote end. Such typical UC3 edge to edge infrastructure aims to provide user guided precision capabilities to conduct complex challenging tasks using teleoperations.

Customisation potential/options: The edge-to-edge devices used for UC3 are quite specific and known to the UC3 partners. This means that the HW and SW are not easily customizable for other IMOCO4.E projects. Edge advancement will certainly open in the future for increased portability and customisation with standards advancements but for now the UC3 BB1 research is not easily adapted or customized for non-teleoperations type projects. Various other components developed for UC3 have all got customisation and integration potential in advancing robotic platforms.

Possible modifications and extensions: UC3 needs a full dedicated DT extension, and the platform should be easy to modify for the additions of new sensors and robotic components in the future.

3.4.7 Methodology & Toolchains

Toolchain aspects: Multiple forms of toolchains have been used in the UC3 research and development activities. Once a full platform is in place with both modification and extension capabilities then a specific toolchain for the UC3 teleoperations task building would need to be developed. This would enable industry to program/configure the UC3 platform for specific remote teleoperations tasks for specific domains.

Lessons learnt during IMOCO4.E:

- Architecture is vital for the development vision of your project.
- Think at a component and building block integration level.
- Plan for resources issues in terms of HW supply chains and staff availability.
- Embedded AI is difficult for user behaviour prediction to address latency.

3.5 Modular hardware platform based on a mini-PC (IMST)

3.5.1 Technology overview (from D3.4)

The aim is to support demonstrator 3 with a sensor node that connects to the overall system and performs obstacle detection. The sensors are connected to a mini-PC platform, which will be an Nvidia Jetson AGX Orin. The Jetson is a device designed for the development of AI applications, but can also be used to synchronize sensors such as a camera and a radar. The sensors will be directly mounted on the forklift system.

In this task IMST focuses on synchronized data acquisition of camera and radar data. The radar implementation has been described in chapter 2.6.2. The Jetson PC will control connected sensors in order of synchronized data acquisition.

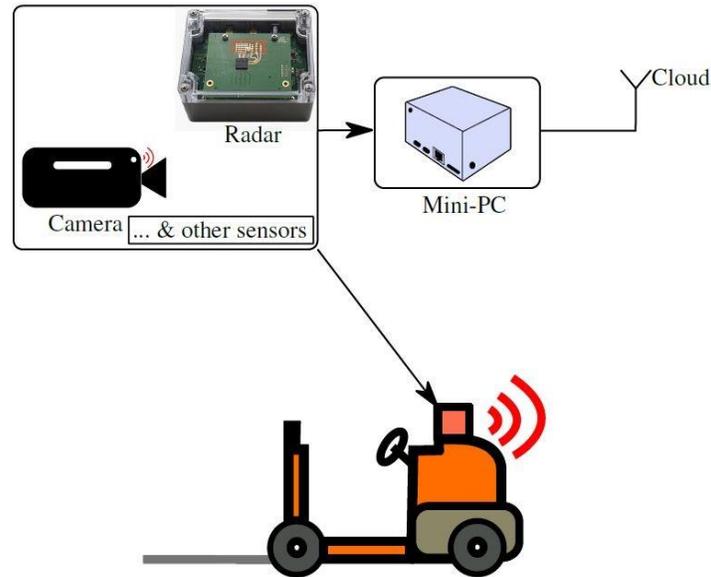


Figure 66. Application of Mini-PC on Demonstrator 3.

The following Figure 67 shows the setup of the Jetson PC. A camera and a radar are connected to an Ethernet hub each. The Ethernet hub connects the sensors to the Jetson PC. The Jetson PC can be controlled via USB connection.



Figure 67. Assembly of Jetson-PC and connected sensors via Ethernet switch.

System architecture

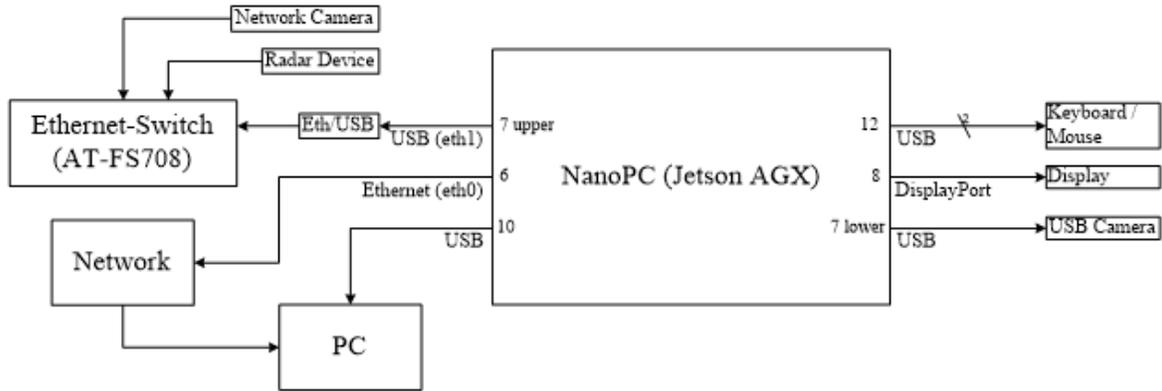


Figure 68. Connected hardware components.

Network Configuration

Figure 69 shows the overall network configuration of the system. The Ethernet port is configured as "eth0" in the operating system. "eth1" is via an adapter connected to the Ethernet switch, which allows simultaneous usage of the radar and a network camera. If the network camera is not in use the radar can directly be connected to the Ethernet/USB adapter. However, the direct Ethernet connection between Jetson PC and radar is a source of error, which will be described in section 3.5.3. All connected network devices must be in the same subnet. Therefore, all connections are configured accordingly.

Hardware Overview

The Mini-PC device is a Jetson AGX Orin Developer Kit with Ubuntu 20.4. The connector ports of the Jetson are depicted in Figure 69 below.

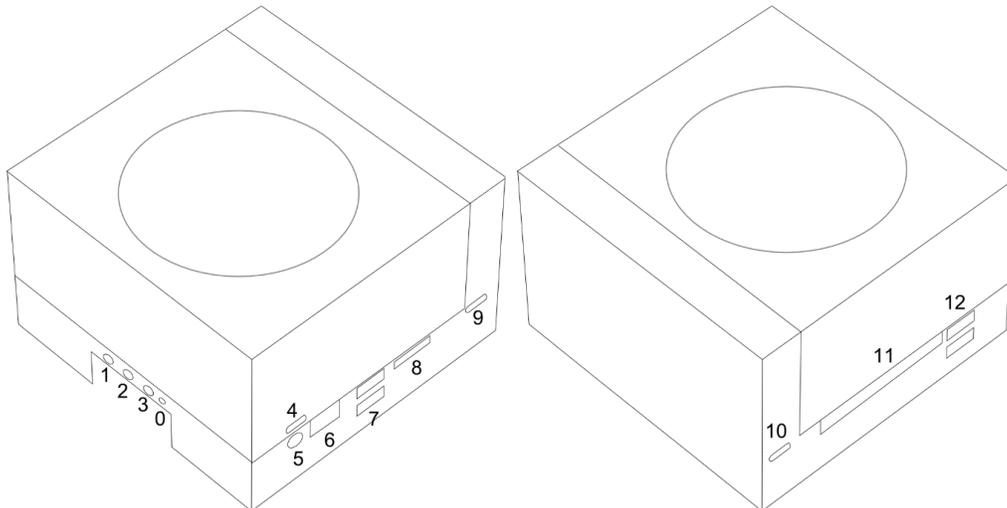


Figure 69. Connector Ports of the Jetson.[1]

The connectors are:

- 0 LED (white)

- 1 Power on/off button
- 2 Force Recovery button
- 3 Reset button
- 4 USB-C
- 5 DC power supply
- 6 Ethernet port
- 7 2 x USB-A
- 8 DisplayPort output
- 9 USB-B
- 10 USB-C
- 11 40-pin connector
- 12 2 x USB-A

Connected Devices

The display is connected to connector 8. The upper USB port 7 is connected via an USB to Ethernet adapter to a switch. The lower port can be connected to a USB camera. The switch is also connected to the radar device. A network camera can optionally be connected. The Ethernet port is connected to the local network where the PC has access. A direct connection to the PC is via USB by connector 10. This connection is for file exchange and.

3.5.2 Implementation aspects

The scope of this implementation was to show, that a radar can be connected via a sensor node as already described in section 3.5.1. Two radar devices have been mounted on the Pilot 4 C-arc robot as depicted in Figure 70. For the testing of the Jetson PC, one radar and a USB-camera were connected to it. The data acquisition on the Jetson PC had to be manually started through the connection from an external computer with a screen. This configuration can be seen in the following figure. The radar devices are mounted on the detector of the C-arc and directed to the ground. In this scenario the C-arc moves towards a patient bed and the height of the radar device is adjusted by using altering the position of the detector.



Figure 70. Radar, Camera are connected over the Jetson PC on a Philipps C-arc.

Data Player

The following Figure 71 provides an overview of the software components. The radar and video data are recorded with the Jetson PC. They must be transferred to an external PC via Ethernet or USB using special software such as WinSCP. The evaluation software, e.g. Data Player, displays synchronized video and radar data. The time stamp information allows radar frames to be paired with camera frames. The graphical overview and the result of the Data Player can be found in the following chapter.

It has been decided that a ROS-2 system will not be implemented on this system.

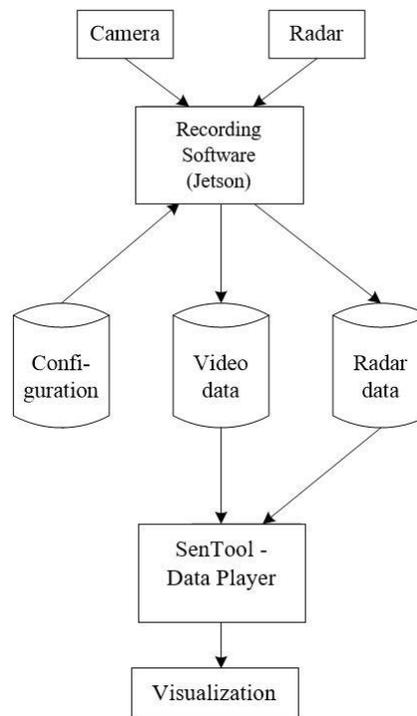


Figure 71. Synchronization of sensor node data.

3.5.3 Results

In the second measurement campaign in Eindhoven this system was applied. The occurrence of an error in the Ethernet connection between Jetson and radar led to a stop in the radar measurement. Since this system is autonomous, a real time observation of the measurement was not possible. However, from later investigation it was found out, that using an Ethernet switch between radar and Jetson PC, led to a more stable connection such that data could be collected. Further inhouse tests were conducted in which radar and camera data were successfully captured at the same time.

In the following Figure 72 a newly developed software is shown. The **SenTool™** Data Player can display recorded radar and camera data for example in .jpg format. Both kind of data are displayed synchronously based on their respective timestamp. The tool works like a common video player. It can also cut the radar data stream if it is needed.

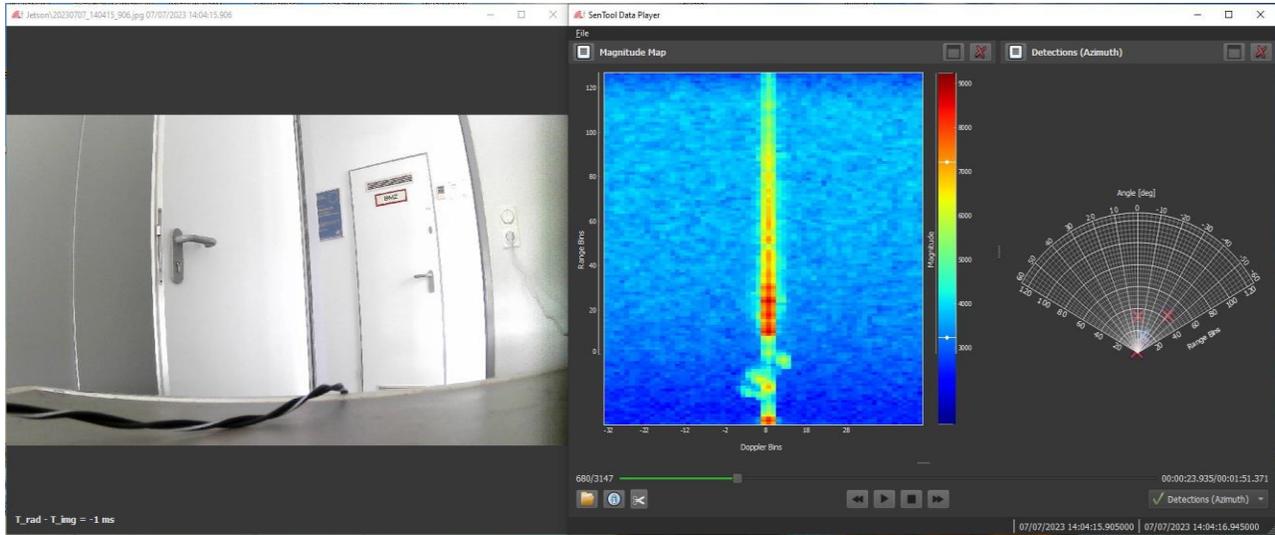


Figure 72. Camera data synchronized with radar data.

3.5.4 IMOCO4.E Requirements

Table 16: Requirements for Modular hardware platform based on a mini-PC.

ID	Requirement	Verify	
R69-swD4.2	Radar data must be saved as .bin or .csv data	T	.bin is always preferred
Req-D3.2-L1-fw-03	Firmware of sensor can be updated via Ethernet or USB interface.	D	Ethernet compliancy verified
R65-swD4.2	Interconnection of Radar and Mini-Pc to other Demonstrator components via Ethernet or USB interface	T	This requirement is to be fulfilled in the last step
R221-D2.3	Enough processing power is needed to work with real-time sensor data (for localization and navigation calculation).	T	The Jetson PC possesses the required amount of processing power,

3.5.5 Customizations and Adaptations

- **Component name:** NVIDIA Jetson AGX Orin
- **Component HW/SW/INT catalogues reference:** HW-002 / SW-008
- **Use of component in the P/D/UC (out-of-the-box functionality):** The Jetson PC operates as a sensor nodes. Multiple radar devices and a USB camera can be controlled.
- **Possible modifications and extensions:** ROS-2 system can be implemented on the Jetson
- **Lessons learnt during IMOCO:** Post Processing of radar data for better visibility necessary

3.5.6 References

[1] Jetson Hardware Layout, 2023. NVIDIA, https://developer.nvidia.com/embedded/learn/jetson-agx-orin-devkit-user-guide/developer_kit_layout.html

3.6 Modular platform for Smart Sensors with Advanced Data Processing (EVI)

3.6.1 Technology overview

Inside Pilot 2, Evidence proposed to leverage Huawei’s platforms to reduce the execution time of Artificial intelligence (AI) based tasks. To this aim, Huawei’s platforms featuring Huawei Ascend 310 chip will be used. The Ascend 310 chip is a highly efficient, flexible, and programmable AI processor. It can achieve up to 16 TOPS (for INT8) and 8 TOPS (for FP16) with a power consumption of 8 W.

Two different platforms were used to replicate the use case of Pilot 2. The first platform is the Huawei Atlas 200 DK. It features the Ascend chip to run Neural Network (NN) inference as well as a Ethernet port and other ports to enable I/O. Figure 73 shows the Huawei Atlas 200 DK.



Figure 73. Developer Kit Atlas 200 DK AI.

The second platform is the Huawei Atlas 300I inference card. It is a PCIe card used to accelerate NN inference on pre-existing host systems. This PCIe card is available for both PC and ARM-based hosts. It is designed to achieve low-latency NN inference for edge scenarios. Figure 74 shows the Huawei Atlas 300I inference card.



Figure 74. Atlas 300I Inference Card.

MindSpore is the open-source deep learning training/inference framework developed by Huawei. It makes full use of the computing power of the Ascend chips and provides unified APIs and end-to-end AI capabilities for model development, execution, and deployment both on cloud and edge devices. MindSpore can also be used to import NNs from other frameworks (e.g., TensorFlow, etc) and run it on platforms featuring the Ascend chip.

3.6.2 Implementation aspects

In the real scenario of the Pilot 2, a PC equipped with a Huawei Atlas 300I inference card will be used as a host where data is collected and analysed. Figure 75 shows the high-level logical architecture of this scenario.



Figure 75. High-level logical architecture of addressed scenario.

Evidence replicated two different scenarios using both the Atlas 200DK and the Atlas 300I inference card to provide different performance/cost trade-offs. In these scenarios, the dataset is stored directly on disk since the main goal of the activity is to reduce the execution time of the AI-based algorithm.

The first setup employs the Huawei Atlas 200DK board. This board feature MIPI connectors that can be used to connect external camera to acquire images; while the output of the AI-based algorithm can be sent out via the Ethernet interface. This board can be used as a single board solution for this Pilot’s scenario. Figure 76 shows a schema representing this setup.



Figure 76. Setup employing the Huawei Atlas 200 DK.

The second setup employs a PC equipped with the Huawei Atlas 300i inference card. This card is meant to be included in an already working scenario to enhance NN inference. It features a standard PCIe connection, and it is available in two different versions: one for standard PC hosts and one for ARM-based hosts. Figure 77 shows a schema representing this setup.

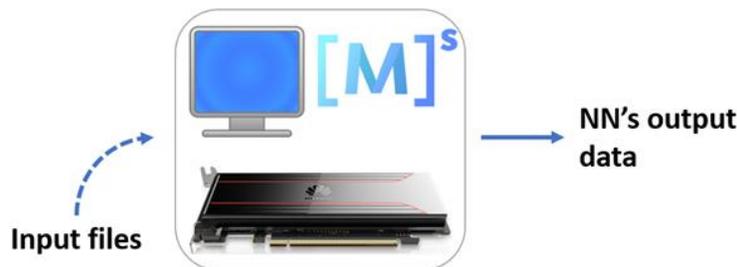


Figure 77. Setup employing the Huawei Atlas 300I inference card.

3.6.3 Results

Preliminary results carried out in the scenarios replicated by Evidence shows that Huawei’s board can effectively reduce the AI-based algorithms execution time. This result confirms the expectation since a hardware accelerator usually enables shorter execution time compared to general-purpose hardware. Validation of these results are expected to be carried out in the future by ITEC in its facilities where a complete real-world scenario can be realised.

In these months Evidence focused on the Pilot 3 and the research activity, the results of which are reported in deliverable 4.8. In next months, Evidence will focus on completing the activity of Pilot 2.

3.6.4 IMOCO4.E Requirements

This activity meets the following requirements.

Table 17: Requirements for Modular platform for Smart Sensors with Advanced Data Processing

ID	Requirement	Priority	Verify	Comments	Tasks	Compliant
R063-D2.3-L2	The optimised Neural Networks must be able to run on the available (brownfield) hardware platforms.	M	D	The NN provided by DTT has been successfully imported and executed on the proposed platform.	T3.3, T3.4	Yes. Activity finished.
R123-D2.3	BB4 shall have a target cost of goods of 10k€ for a basic version	M	I	The proposed solution met this requirement.	T4.6	Yes. Activity finished.
R115-D2.3	BB4 must be able to run on an ARM based platform.	M	I	The proposed platform is an ARM-based platform.	T4.6	Yes. Activity finished.

R063-D2.3-L2: NNs must be ported using software tools from the MindSpore ecosystem before running them.

R115-D2.3: The Huawei Atlas 300I inference card is available both for x86 and ARM servers with PCIe connections.

3.6.5 Capabilities and Limitations

Our solution focus on the acceleration of AI-based algorithms, particularly NNs, by leveraging the Atlas 300I. It can be adapted to different scenarios requiring NNs. The Atlas device family includes stand-alone devices (i.e., the Atlas 200DK, Atlas 500, Atlas 500 Pro, etc.) and PCIe based solutions (i.e. the Atlas 300 cards). We integrated the Atlas 300I PCIe card in the Pilot 2.

The Atlas 300I inference card unlocks superior AI inference performance. It is powered by the Ascend processor and provides high computing power making it an ideal option for different intelligent application scenarios.

3.6.6 Customizations and Adaptations

The Atlas 300I can be used to accelerate AI-based algorithms, in particular NN-based algorithms. This device can be used in any scenario where low-latency and high throughput AI processing are needed.

For scenarios with very high throughput requirement, the Atlas 300I can be used inside the Atlas 50 Pro. The Atlas 500 Pro is an ARM-based system that features a Linux based OS and up to four Atlas 300I cards.

3.6.7 Methodology and Toolchains

NNs able to run on the Atlas 300I can be developed using the MindSpore tool or can be imported from other third-party frameworks (e.g., TensorFlow).

3.7 Rugged embedded IO-controller platform (Exertus)

3.7.1 Technology overview

The powerful and rugged embedded IO-controller platform developed by Exertus is targeted to accomplish the requirements of modern mobile machinery e.g. a mining machine. The technology used is selected based on the requirements of the typical applications working in harsh conditions and requiring high performance, versatility, reliability, and safety.

The architecture is based on ST Microelectronics STM32H7 high-end microcontroller which is running ARM Cortex M7 and M4 in parallel utilizing 480MHz clock frequency. The peripherals cover the needs of connectivity in the machine level including several CAN-buses and Ethernet. The platform can drive a high number of parallel current-controlled outputs controlling for example hydraulic valves and other auxiliary devices as well as reading a high number of analogue and digital sensor inputs.

As safety is the number one requirement in mobile machinery, special attention has been paid to safety requirements covering for example duplicated signalling consisting of the high-side and low-side controls for output activation as well as duplicated reading of input signals. The firmware will also include a feature rich set of different verification and validation methods for system level functions.

3.7.2 Implementation aspects

The typical architecture of a distributed control system of a mobile machine is spreading around the machine platform where inputs and outputs are being handled and control loops are running in the specified IO-controllers. As the requirements for performance and complexity of the systems are increasing due to an autonomic function, more powerful IO-controllers will be needed to fulfil the requirements.

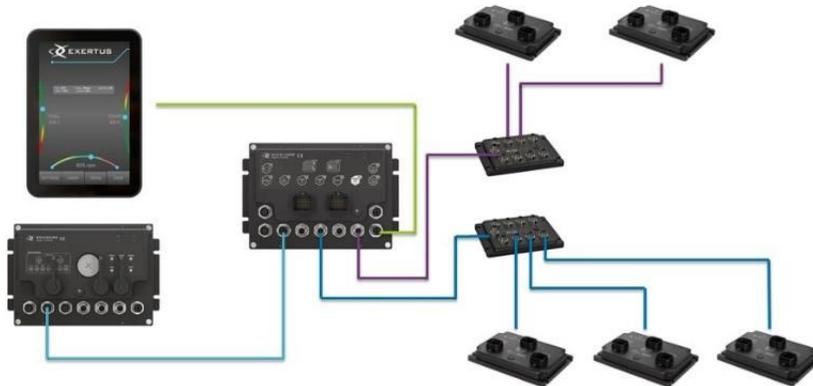


Figure 78. Example architecture of distributed mobile control system consisting of (from left to right) touch display, IoT connectivity unit, master controller, 2 CAN hubs and 5 IO controllers.

The overall software architecture for the platform is presented below in the next picture. The Hardware Abstraction Layer is comprised of MCU manufacturer provided low-level driver parts integrated with higher-level drivers implemented by Exertus. Firmware layer can include the RTOS environment, function library and communication interface stacks, and the safety monitoring implementation of the platform.

On the application layer there are two distinct APIs: direct safety critical C API, and the Runtime API which is much more featureful including additional functionality for e.g. graphics rendering and connectivity.

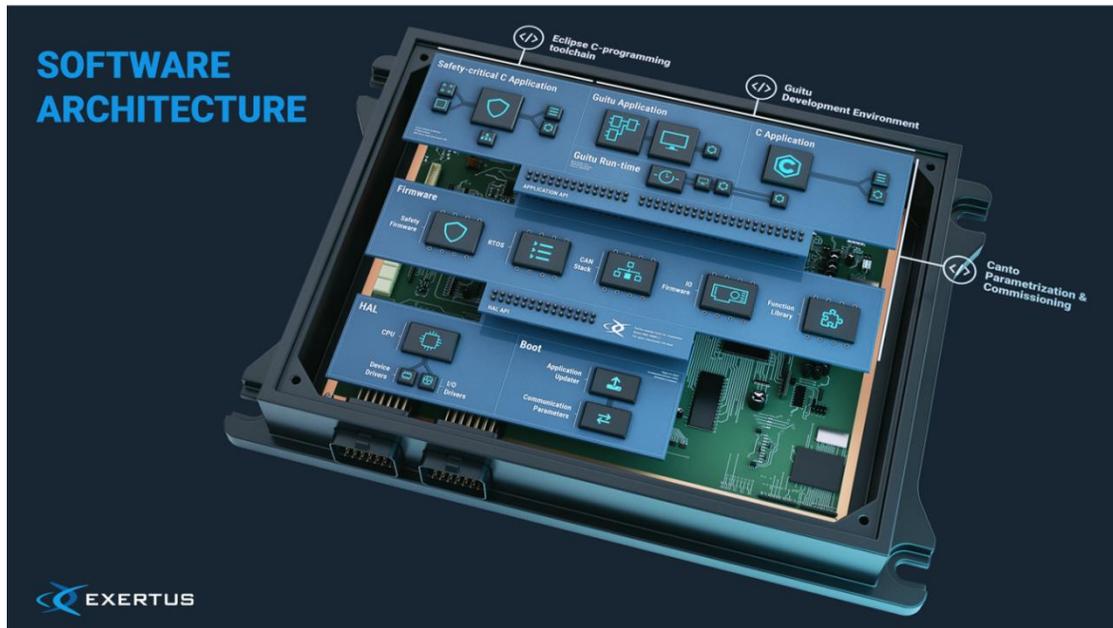


Figure 79. Rugged IO-controller software architecture diagram.

3.7.3 Results

The controller prototype has been successfully implemented including electronics and mechanics. The software implementation includes all the necessary part of hardware abstraction layer for the current hardware setup as well as firmware including the elements according to the architecture picture above. IMOCO4.E requirements are fulfilled by following the industry standard CANopen interface which can be used also for configuring the parameters without firmware changes.

Implementation of safety features was requiring lots of attention both in hardware and firmware implementation. Based on the concept evaluation made by the official party, the implementation seems to reach the required level. Fault monitoring and other safety related matters in the platform have been evaluated by the official party.

3.7.4 IMOCO4.E Requirements

This platform fulfils the following requirements. The standard followed in interfacing and configuration is CANopen which is widely used in the mobile machinery control systems.

Table 18. Requirements for Rugged embedded IO-controller platform.

ID	Requirement	Priority	Verify	Comments	Tasks
R070-D2.3	The interfaces to BB1 shall be an industry standard (such as USB, Ethernet, expansion port or optical fiber connectors).	M	I	Completed	T3.1
R082-D2.3	BB1 shall have a configuration interface to modify all (pre-defined) configuration parameters without requiring firmware changes	M	D	Completed	T3.1
R083-D2.3	BB1 shall offer ways of working and toolchains that extend across multiple (generations of) platforms	M	D	Completed	T3.1
Req-D3.2-B1	Continuous monitoring of the hardware to find faulty behaviours.	C	T	Completed	T3.1, T3.2
Req-D3.2-B1	Sensors and actuators used together must have low interference with each other. Wireless sensors require dedicated frequency bands or interoperable protocols.	M	D	Completed	T3.2, T3.5

3.7.5 Capabilities and Limitations

The capabilities of the new platform are supporting forthcoming needs for advanced automation in mobile machinery. Computation power enables e.g. crane tip control or automatic steering applications to be implemented in the platform. As the same platform is capable of driving proportional hydraulic valves, it

makes it compact and effective solution for this kind of solutions on the way towards autonomy of the machines.

The strength of the platform is very compact design taking into account all the features and number of inputs/outputs. The functional safety architecture supports also implementation of safety critical applications for the controller.

3.7.6 Customizations and Adaptations

The platform will be adapted to and tested with a real mining machine in the context of Pilot 5 Mining Machine. Due to delays these tests have not yet been carried out as of the release of D3.7 but are still scheduled to be finalized before the official closing of IMOCO4.E.

Intended for utilization in mobile work machinery control systems, the platform can be easily adapted to use in any CAN bus-based environment requiring high I/O performance and functional safety. In addition, it supports OEM customization with C code through different APIs (either for functional safety function implementations or more traditional control logic).

3.7.7 Methodology and Toolchains

One of the key aspects of the project was utilizing STMicroelectronics' STM32 series reference safety architecture for a custom-designed control system platform. So far, the reference architecture has proven to be well-adaptable to concrete use cases and well-define enough to be of significant help during design and implementation phases.

3.8 New technologies of energy-efficient AI processors (REEXEN)

3.8.1 Technology overview

ADA100 is a mixed-signal near-sensor computing intelligent processor of ultra-low power consumption, which can operate in always-on mode. It can deal with signal processing and neural network inference for various time series sensors. At the same time, it features for extremely small size and ultra-low power consumption, which is suitable for deployment on battery-powered end-point AI products, such as AR/VR devices, TWS headphones, smart watches, smart home devices, etc. It is to equip these products with intelligent and high-precision human-computer interaction.

ADA100 has a built-in deep learning network. On the embedded device side, deep learning is mainly used when designing the network due to the difference in computing power, so as to enhance the perception ability as much as possible while ensuring the perception ability (The article for details Xception: Deep Learning with Depthwise Separable Convolutions, François Chollet). Therefore, deep learning is used internally in ADA100 to realize the classification task.

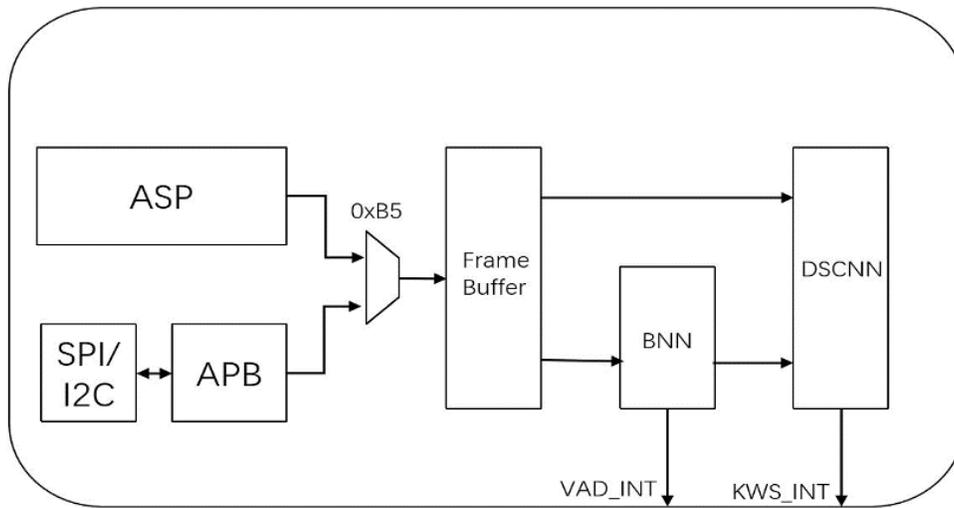


Figure 80. NPU Data Flow Direction.

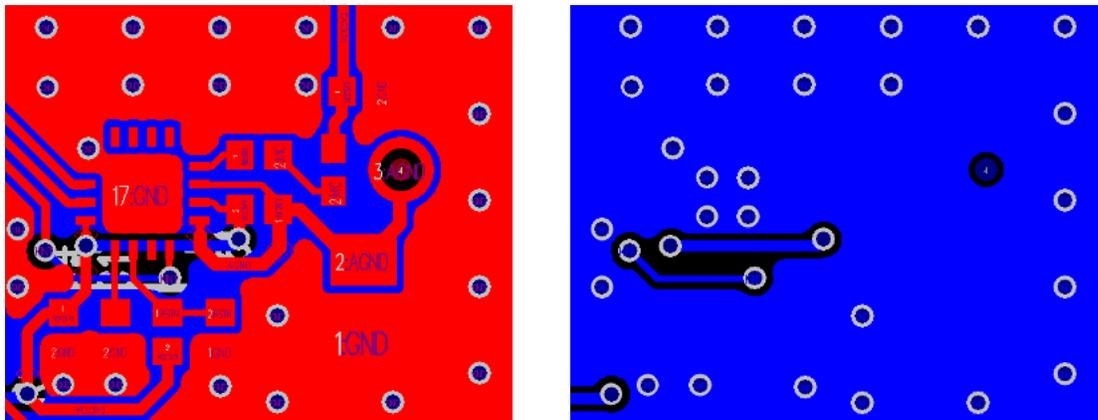


Figure 81. ADA100 layout (Top) and ADA100 layout (Bottom).

As shown in the figure, the ADA100 NPU has two operating modes: continuous mode and bypass mode. When the 0xFF register is set to 0x00, ADA100 works in continuous mode, and the audio input signals are collected by the ASP module and the amplitude frequency characteristics are extracted, which are input to BNN or DS-CNN network. So as to realize VAD(Voice Active Detection), KWS(Keyword Spotting), ASR(Audio Scene Recognition) and other applications. When the 0xFF register is set to 0x02, the ADA100 operates in bypass mode, in which the ASP analog feature extraction module does not work and therefore, non-audio digital signals can be manually entered.

3.8.2 Implementation aspects

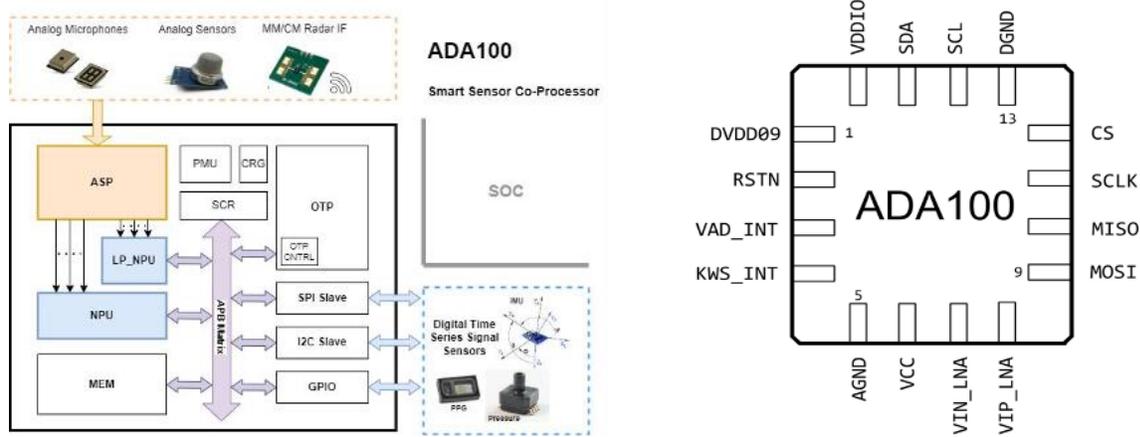


Figure 82. ADA100 Smart sensor AI-processor.

The circuit is suitable for I2C or SPI communication scheme, the maximum support of 30 keywords recognition, when there is speech, VADINT(PIN3) output high pulse indicator to detect the human voice; When the keywords are identified, KWSINT(PIN4) outputs a nanosecond pulse, indicating that it is time to read the ADA100 register and extract the identified keywords.

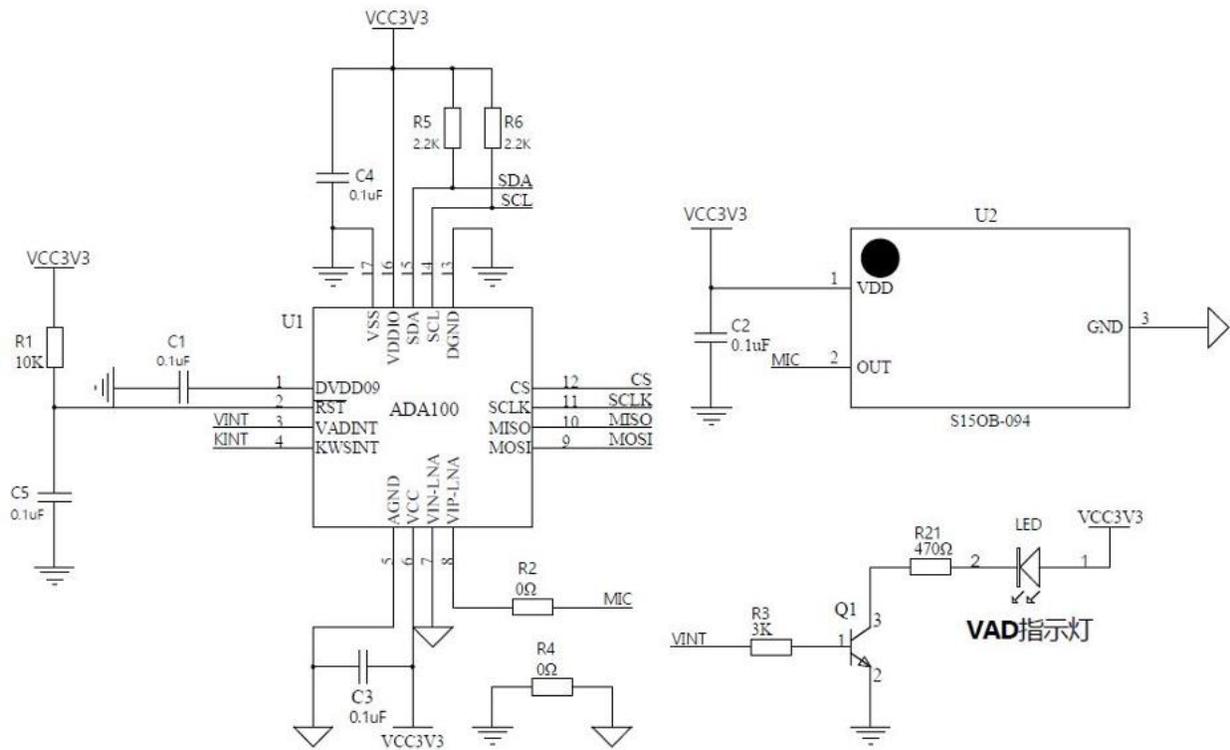


Figure 83. ADA100 Structure.

- Timing signal multisensor processing chip.
- Similar chip 1/3 cost
1/10 power consumption.
- Application scenarios:
Smart Voice cockpit, In-car baby cry, awake equipment inside and outside the vehicle

3.8.3 Results

Wearable products have strict requirements on power saving to maintain the battery life of products. Traditionally, the key word identification (KWS) is processed by the main system (main control chip), which requires large power consumption. It is a very practical function to set the voice wake-up word (VAD) function to make the main system change from sleep state to work state, which can greatly save power. The following is the system design of VAD+KWS scenario using ADA100 on TWS headset.

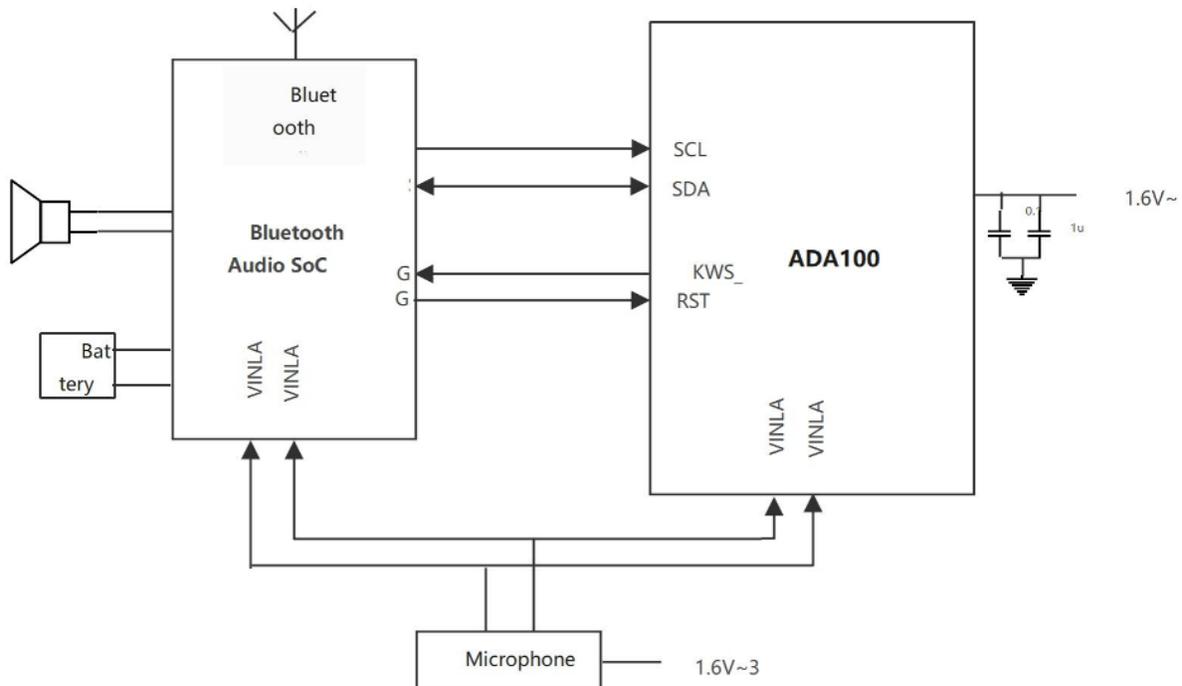


Figure 84. System Block Diagram.

Module Description:

- Bluetooth Audio SoC : System-on-chip with on-chip Bluetooth, audio, and programmable application processors
- ADA100 : Super-low power KWS chip

This approach works well in VAD-triggered KWS mode. First, human voice detection is performed in very low power mode, with a power consumption of about 70uA. Second, when the human voice is recognized, it will run the KWS mode to match the preset command words. Finally, after the correct judgment, the SoC will be awakened by the interrupt wake-up, and the SoC will open the normal working mode and actively read the command word data in the register through the IIC or SPI data port to carry out relevant actions.

3.8.4 IMOCO4.E Requirements

Table 19: Requirements for energy-efficient AI processors

ID	Requirement	Priority	Verify	Task	Result
Req-D2.3-B8	Real time inference (limited and deterministic).	S	D		Achieved
Req-D3.2-B1	Continuous monitoring of the hardware to find faulty behaviors.	C	T	T3.1 T3.2	Achieved
RO83-D2.3	BB1 shall offer ways of working and toolchains that extend across multiple (generations of) platforms	M	D	T3.1	In progress

3.8.5 Capabilities and Limitations

Performance Evaluation:

1 Support for up to 30 KWS keywords

1 Support single VAD, single KWS and VAD+KWS modes

1 Support VAD+user_KWS mode (Support users to extract feature values and calculate KWS results by themselves)

1 VAD and KWS algorithms support changing algorithm parameters, It can be loaded via I2C or SPI.

1 For power consumption in KWS+VAD mode, see "Active Mode" in "Performance Specifications - Typical Power Consumption"

Use Case:

With the continuous development of smart technologies, vehicle control system constantly innovating and improving to provide a more convenient and intelligent user experience. This project aims to develop a module based on the ADA100 voice chip for controlling the vehicle's trunk.

This module will allow car owners to easily and quickly open or close the vehicle's trunk using voice commands (in the actual demo, as BMW did not provide a vehicle, a light bulb switch is used to simulate the opening and closing of the vehicle's trunk), enhancing the convenience and technological aspects of vehicle usage.



Figure 85. Unique selling point of car.

3.8.6 Customizations and Adaptations

In this scenario, ADA100 can be configured to work in NPU active mode. Firstly, heart rate and blood oxygen data can be collected through PPG sensors, and NPU can be configured

- 2) After obtaining the required registers for active mode, input the preprocessed attitude data through SPI/I2C into ADA100 NPU and wait for ADA100 NPU.
- 3) After the calculation is completed, an interruption is generated to notify the SoC to read the current user's health status (abnormal, normal). When the PPG dataset is sufficiently rich
- 4) Afterwards, the initial screening and classification function of specific lesions can also be achieved.

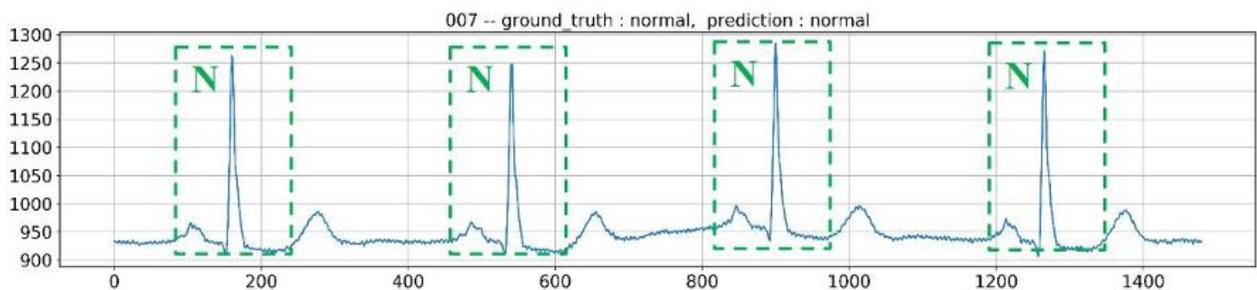


Figure 86. Test Record.



Figure 87. ADA100 PPG sensor and signal conditioning circuit.

The revised voice signal processing chip is capable of capturing and analyzing the weak biological electrical signals produced by the human body, known as electrocardiogram (ECG) signals. Leveraging its built-in high-precision analog-to-digital converter (ADC) and advanced digital signal processing algorithms, the chip accurately extracts heart rate information from the ECG signals, enabling non-contact and real-time heart rate monitoring. This functionality holds vast potential for applications in health monitoring, sports and fitness, medical diagnosis, and beyond.

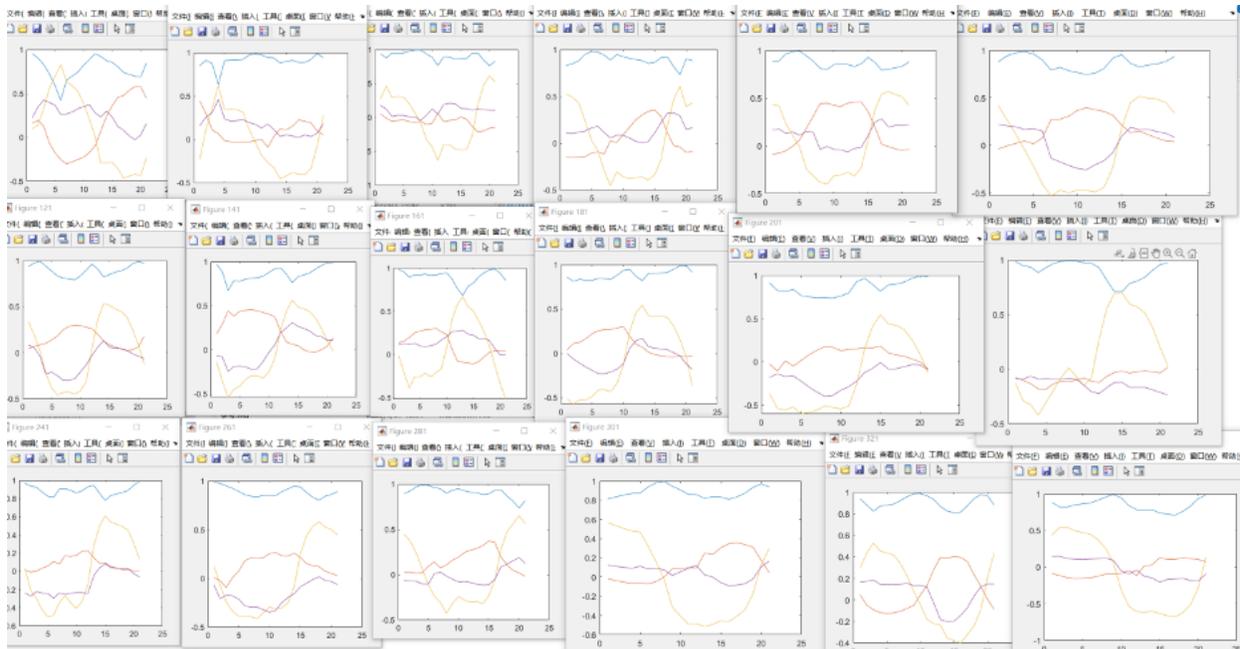


Figure 88. The difference of pose curve for the hand motion.

In addition to heart rate monitoring, the revised voice signal processing chip also integrates IMU (Inertial Measurement Unit) capabilities. The IMU is a device that measures the angular velocity and acceleration of an object in three-dimensional space, finding widespread use in robots, drones, virtual reality, and other fields. Through its integrated gyroscope and accelerometer, the revised chip can capture real-time motion data of objects, providing accurate data support for subsequent attitude estimation, navigation, and positioning.

- **High Integration:** The revised voice signal processing chip integrates voice processing, heart rate monitoring, and IMU capabilities into a single unit, significantly simplifying hardware design and reducing system costs.
- **Low Power Consumption:** The chip employs advanced low-power design techniques, ensuring stable performance even under prolonged operation.
- **High-Precision Measurement:** Through optimized algorithms and hardware design, the revised chip demonstrates exceptional accuracy and stability in both heart rate monitoring and IMU measurements.
- **Ease of Expansion and Integration:** The chip features standard communication interfaces and protocols, facilitating seamless connection and integration with other devices or systems.

In conclusion, the revised voice signal processing chip, with its expanded capabilities in heart rate monitoring and IMU functionality, represents a significant technological breakthrough. Its integration of multiple features, coupled with its high precision, low power consumption, and ease of integration, positions it as a valuable addition to a wide range of applications in diverse fields.

3.8.7 Methodology and Toolchains

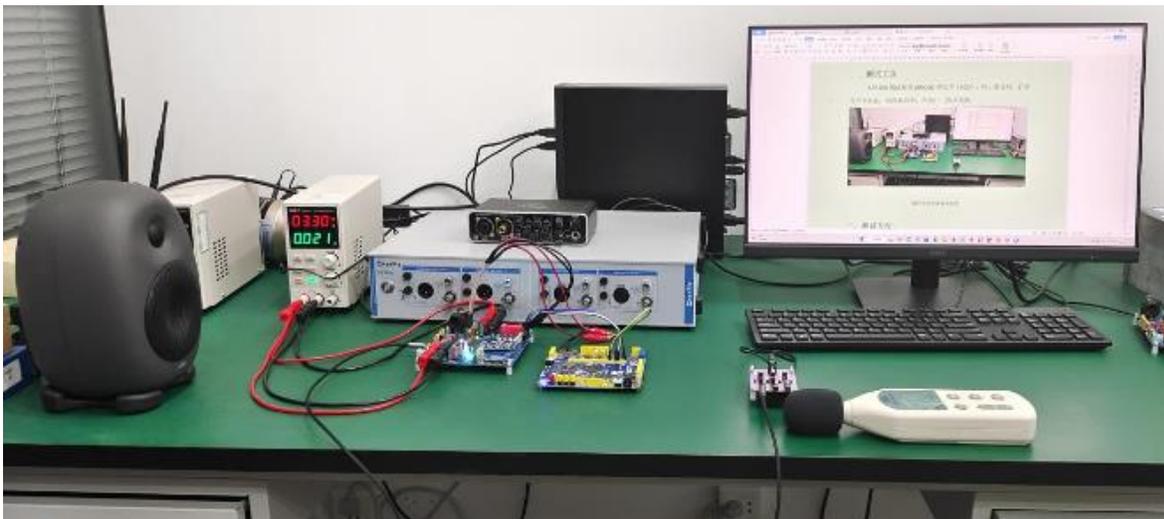


Figure 89. Test environment and equipment connection diagram.

Limitation in Interface Design: Many offline speech chips have relatively monotonous interface designs, lacking diversity and flexibility. This limitation may restrict the use of these chips in specific application scenarios, as they may not be able to seamlessly integrate with certain devices or systems.

Existing mature customers have already been integrated into the product, with good market feedback and application foundation.

ADA100 adopts an in-memory computing architecture, which has 5-10 times lower power consumption compared to ordinary architectures.

The diversity in model development allows for flexible development.

We have the best team with strong backgrounds, including colleagues specializing in software algorithms and hardware development. We are also working on developing high-performance AI chips within the allotted time.

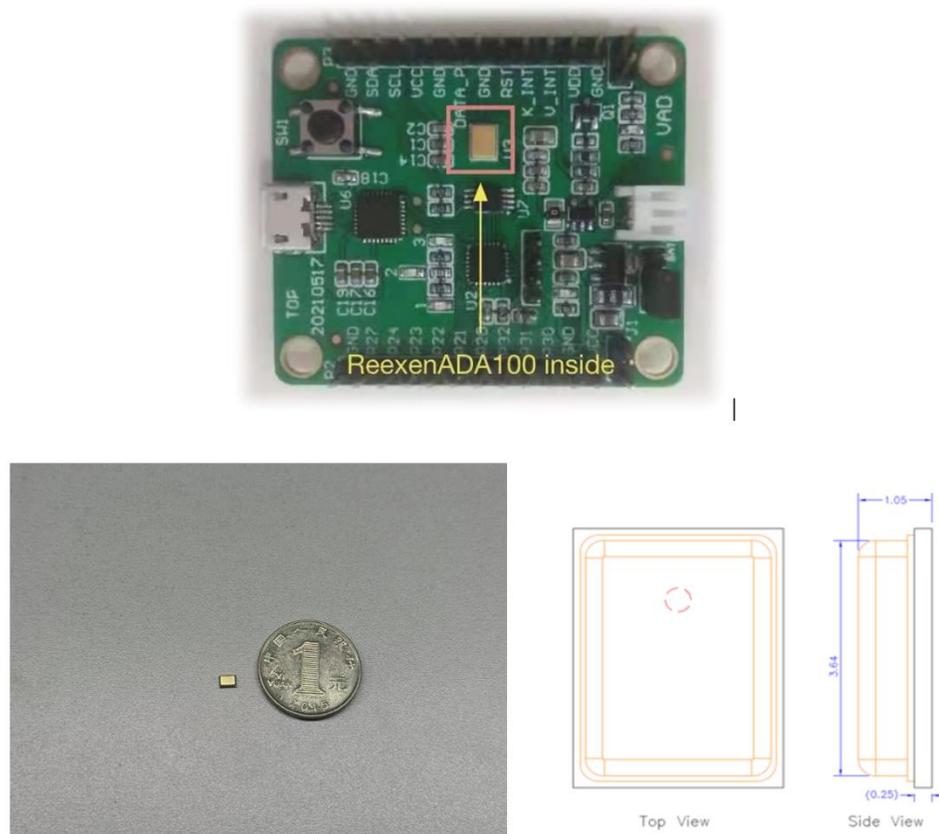


Figure 90. ADA100 super low-power Smart-mic

Size: $4\text{mm} \times 3\text{mm} \times 1.05\text{mm}$

Power consumption: Active 170uA.

Our voice chip comes with a complete SDK and toolchain, which includes circuit design tools, simulation tools, and more. These tools assist engineers in circuit design, performance simulation, and verification during the chip development stage. Development tools are used to write and debug the software and firmware for offline voice chips. This may include compilers, debuggers, integrated development environments (IDEs), and so on. These tools help developers write and optimize code to ensure that the chip operates correctly and fulfills the expected functions.

Testing tools are used to test and validate the offline voice chip to ensure its performance and reliability. This may include automated test equipment, test frameworks, test cases, and more. These tools assist test personnel in discovering and fixing issues in the chip, ensuring product quality.

Production tools are used to manufacture and package offline voice chips. This may include production line equipment, testing equipment, packaging equipment, and more. These tools help manufacturers transform designs into actual products, ensuring the efficiency and reliability of the production process.

3.9 Platform for standard TSN bridging delivering time-aware scheduling (SED)

3.9.1 Technology overview (HW-015)

The main goal of SED in this task was to achieve a platform that would enable communication networks with time-aware scheduling. This was accomplished through the implementation of TSN (Time Sensitive Networking), an extension of the Ethernet protocol that allows for deterministic communications and prioritization of various traffic within the network. In the IMOCO4.E project, the development has been continued to achieve a complete TSN system, including a software monitoring tool (SW-075) that will be explored further in T4.4 and an FPGA-based hardware platform (HW-015), which we will focus on now.

The hardware component (HW-015) or TSN platform in this task falls under BB1. Given the complexity of TSN as a technology, it can be divided into several subsystems that are essential for the complete operation of the platform. In Figure 91, a simplified diagram illustrates the programmable logic of the FPGA, showcasing parts of these subsystems. Components of the Ethernet system (providing high-capacity data transport for the platform), Switching (data frames forwarding capability), Operating System (main execution framework), and PTP synchronization (system-wide network time reference) are visible. However, the primary advancements have been made in the TSN subsystem.

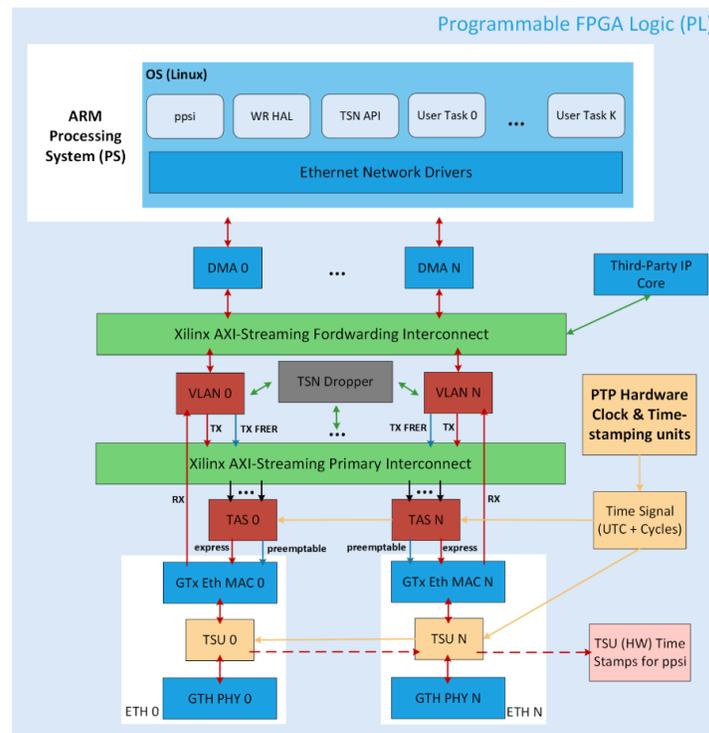


Figure 91. Diagram of the Programmable FPGA Logic.

The main components of a TSN system, and the ones we have focused on in this project, are the traffic identification (VLAN), traffic shaping (TAS), frame preemption, and redundancy (FRER) modules. In this task and building block, the first three modules have been included, leaving the redundancy module, which is more related to security, for BB9, as explained in D5.8.

Traffic identification module (VLAN - 802.1Q)

Traffic identification in the network is carried out through the VLAN (Virtual Local Area Network) core following the 802.1Q standard, and it is one of the main components within the TSN system. This module provides a mechanism to organize and prioritize network traffic by logically segmenting it into different VLANs. Each VLAN represents a virtual LAN, allowing for the isolation and differentiation of traffic flows based on the tags embedded within Ethernet frames. By leveraging tagging, the module enables efficient management of network resources, ensuring that traffic flows are categorized and treated appropriately according to the assigned VLAN.

The core operates at the data link layer, where it intercepts incoming Ethernet frames and examines their VLAN tags. After extracting the VLAN identifiers, routing and transmission of frames within the TSN network occur. This segmentation allows for the differentiation of each of the defined traffics within the network and their corresponding prioritization since, in addition to the VLAN ID, a priority level can be indicated. Using these fields, the criticality of each traffic can be defined to assign different slots within the TSN network. The number of traffics supported by this module will largely determine the scalability and flexibility of the TSN system. Furthermore, the use of VLAN tagging facilitates integration with existing network infrastructures, as simply configuring tagging on network interfaces allows to directly connect it to the TSN network.

Time-aware traffic shaper (TAS - 802.1Qbv)

The Time-Aware Traffic Shaper (TAS) is another essential component in TSN systems, developed following the 802.1Qbv standard. This core plays a crucial role in traffic management within the network, ensuring timely delivery of critical data. Following the standard, TAS assigns specific time slots for different traffic flows, ensuring that data packets are transmitted within specified time intervals. This capability is fundamental for maintaining predictability, low latency, and determinism in environments where timing is critical, such as industrial applications or real-time communication systems.

In implementation, TAS operates with advanced time scheduling and Medium Access Control (MAC) mechanisms, coordinating the transmission of data packets synchronously and preventing packet collisions. Depending on the configuration, priorities are assigned to traffic flows based on their time requirements, and TDMA-like processing is performed for TSN flows, ensuring that data is transmitted within specified limits.

Enhancements for preemption (802.1Qbu & 802.3br)

In the cores that have been further developed during the project, significant enhancements related to preemption have also been made, following the standards 802.1Qbu and 802.3br. The aim of these implementations is to provide advanced mechanisms to ensure that critical data can interrupt lower priority transmissions, enabling a faster response to important events in the network. The 802.1Qbu standard focuses on frame preemption in Ethernet, that works in conjunction with the previous TAS module, while 802.3br addresses preemption at the physical layer, more related with the MAC, providing a comprehensive solution to improve network efficiency and reliability.

In practice, these preemption enhancements allow network devices to dynamically respond to critical events, such as medium access control and congestion management, by temporarily suspending low priority transmissions without losing bandwidth. This can be done in conjunction with the VLAN core explained earlier, as it is necessary to identify each of the traffic types and their respective priorities to enable message interruption. Furthermore, by incorporating these preemption mechanisms into the network, the ability to meet strict latency requirements is improved, especially in applications where real-time responsiveness is crucial.

3.9.2 Implementation aspects

The implementation of all the functionalities and cores mentioned previously has been carried out on the Xilinx Zynq 7000 family SoC. Before this project, basic developments of a TSN architecture had been conducted on other simpler and less heavy platforms. To adapt to the new platform, developments were made across the entire system, working on and adjusting the Ethernet subsystem, PTP timing or switching subsystems, as well as all the TSN parts and new features. This has resulted in a new platform called TSN Z16.

Regarding the integration of this platform with the project's pilots, it was done in both P1 and P2, but we will focus on P1 as an example. Once we had the essential development of the platform, we requested some sample traffic similar to what we would find during integration to look for a basic and suitable configuration for the pilot scenario. The initial parameters for configuring a TSN network with our platform were as follows.

- **Transaction rules:** Parameters that define the traffic to be encapsulated. These include parameters such as the destination MAC address, VLAN ID, or priority.
- **Configuration values:** Set of parameters that allow the identification of encapsulated traffic. These can be MAC addresses, IP addresses, ports, or others.
- **Redirection values:** Parameters indicating the path that traffic flows should follow, whether redirection should occur through another port of the switch or if it should be redirected externally.
- **Redundancy:** Parameters indicating whether traffic should be redundant and how to achieve it. This is explained in more detail in D5.8.
- **TAS rules:** Set of parameters that define the time intervals that a port has assigned to each traffic flow. This is linked to the determinism and latency that the packets will have.

To integrate with the pilot, we traveled to the SIOUX offices. Upon arrival, we were provided with a setup that simulated the pilot scenario (Figure 92a)), with one PC simulating all the Tissector traffic, another simulating the digital twin generating its corresponding traffic, and a third acting as the Host PC, where machine control would take place. Based on the previous configuration, a setup was reached that communicated all necessary flows (Figure 92b) shows the scenario in operation), with different priorities depending on the type of traffic.

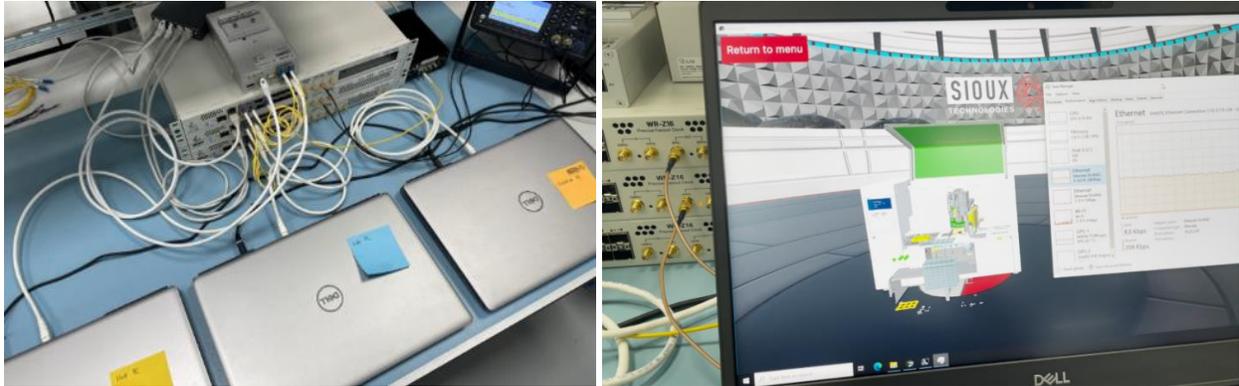


Figure 92. a) Setup at SIOUX offices. b) Setup in operation.

Once a real configuration was established, optimization tests began to achieve maximum determinism within the network with minimal latency. This led to several configurations and multiple rounds of results taken with each of them. Therefore, after achieving the integration of our TSN network with the pilot scenario and obtaining multiple measurements, the integration was considered completed.

3.9.3 Results

Each feature of the TSN platform has been tested during the project to ensure its proper functioning. However, in the second year of the project, a couple of specific tests were defined, as included in D6.3, and the results for this HW-015 component can be found in D6.4. Here, the results of two laboratory tests are presented: "Check the arrival of traffic of different criticality levels" and "Test packet delivery with different bandwidth saturation levels." Both tests were successfully passed, with the first (Figure 93) verifying the modelling and prioritization of traffic with different levels of criticality, and the second aggregating several network flows over the same bandwidth, demonstrating the ability to maintain the integrity of different types of traffic transmitted when applying various configurations.

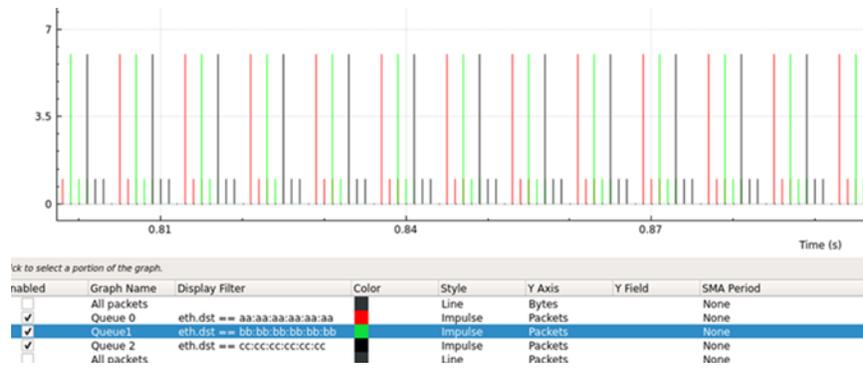


Figure 93. Example results for traffic prioritization.

Nevertheless, the most effective way to test the proper functioning of the platform occurred with the successful integration in both P1 and P2. In these integrations, the various capabilities of the TSN network could be utilized with traffic from the pilots. Several measurements were obtained from these integrations, which will be used to obtain the results of the project and will be included in the final deliverable of WP6. The following tests are planned:

- Check synchronization level of the equipment. Results: To be reported in WP6.
- Latency between network ends. Results: To be reported in WP6.
- Measuring network determinism by changing GCL levels. Results: To be reported in WP6.
- Check network redundancy operation. Results: To be reported in WP6.

3.9.4 IMOCO4.E Requirements

The requirements directly related to the HW-015 of the TSN platform are found in the first table. The remaining, more complex requirements are associated with the integrated component INT-020, which encapsulates the HW-015 component and thus verifies some of its characteristics. These requirements are listed in the second table.

Table 20: HW-015 Requirements.

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
R071-D2.3	BB1 shall support standard and vendor-neutral Wired 1G Ethernet.	I	-	PASS	The HW has interfaces with 1G Ethernet standard support.
R073-D2.3	IEEE 802.1Q Mixed-critical communication support.	T	TEST-006	PASS	Validated in TEST- 006A in D6.4.
R080-D2.3	Delivery guarantee for rate constrained TSN data streams. (Bandwidth)	T	TEST-006	PASS	Validated in TEST- 006B in D6.4.

Table 21: INT-020 Requirements.

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
R078-D2.3	Sub-µs time synchronization based on IEEE 1588 and IEEE 802.1AS.	T	TEST-021	IN PROGRESS	Results to be reported in WP6.
R079-D2.3	End to End deterministic latency for time-constrained TSN data streams.	T	TEST-021	IN PROGRESS	Results to be reported in WP6.
Req-D3.2-B1	Deterministic communication should be maintained to transmit the signals from the sensors and actuators.	D -> T	TEST-021	IN PROGRESS	Results to be reported in WP6.
R023-D5.1-B1	Frame Replication and Elimination Reliability (IEEE 802.1CB) available for user designated data streams.	I -> T	TEST-021	IN PROGRESS	Results to be reported in WP6.

3.9.5 Capabilities and Limitations

The TSN component we've focused on in this section boasts a wide array of features that set it apart from other market options. The platform features 16 ports, facilitating the creation of large and complex networks with a reduced number of devices. It incorporates all essential TSN functionalities, enabling network configuration and leveraging the advantages of this technology. Additionally, it includes monitoring tool (SW-075), allowing for remote visual monitoring and control of the entire network in a simple manner. However, the main strength of this component lies in its high timing and synchronization capacity, which allows for better adjustment of windows and granularity in TSN operations, enhancing bandwidth optimization and latency capabilities in the network.

Some potential limitations of this component may include its size, as its numerous capabilities make it less portable, requiring adjustments to integrate it into confined environments. Additionally, the number of different traffics that can exist in the network is limited, as currently, it supports up to 12 distinct priorities, and in some complex cases, expansion may be necessary.

3.9.6 Customizations and Adaptations

This component has been slightly customized for each integration in this project, both in P1 and P2. Primarily, the changes made have been in the configuration of the networks, requiring alterations to flows, topology, and various parameters for each case. This is common as two scenarios will never be the same where something previously done can be reused, making customization and configuration fundamental in communication networks, especially with TSN.

3.9.7 Methodology and Toolchains

The toolchain used within this component spans a broad spectrum. For the implementation of code within the TSN nodes, the Vivado tool and VHDL language has been used for low-level FPGA programming. Additionally, C code has been utilized for testing and programming the device software, while high-level Python code has been employed for automations and function dispatching. For the user interface part, web languages such as HTTP, CSS, and JavaScript have been utilized, and databases like VictoriaMetrics have been programmed. Furthermore, network tools like Iperf for traffic injection and monitoring tools like Pyshark have been employed.

In the development of this component, we had to leverage this diverse set of tools, learn how to integrate them, and ensure that the whole system becomes intuitive and useful for the end users. We have learned that configuring and adapting each network is crucial for achieving optimal results.

3.10 AutoFlow: open-source software solution (HS)

3.10.1 Technology overview

The recent advancements in the AutoFlow framework, primarily through the integration of AutoKeras, represent a significant leap in machine learning (ML) application development, especially for embedded devices. This overview encapsulates the objectives, implementation, results, and future directions of this initiative.

Objectives and Implementation:

AutoKeras Integration: The core objective was to incorporate AutoKeras into the AutoFlow environment. This integration allows users to harness the power of AutoKeras directly from the AutoFlow graphical user interface (GUI), thus streamlining the ML development process.

Customization for Embedded Devices: A pivotal development was the customization of AutoKeras to automatically generate neural network architectures tailored specifically for embedded devices. This customization takes into account the unique constraints of such devices, including limited memory, processing power, and energy efficiency requirements.

3.10.2 Implementation aspects

Implementation:

- AutoKeras integration into AutoFlow simplifies the machine learning development process.
- Customization of AutoKeras ensures that neural networks are optimized for embedded hardware, considering memory, processing power, and energy efficiency.

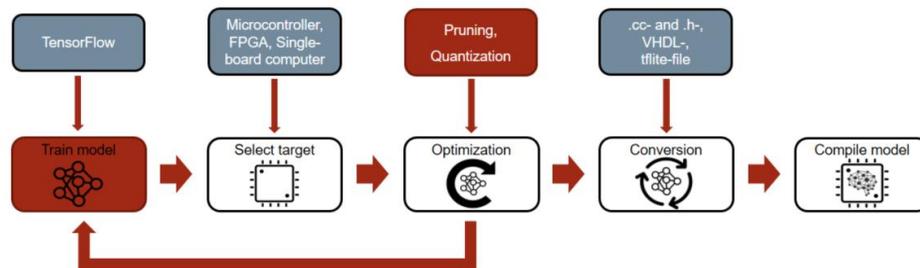


Figure 94. Solution overview.

3.10.3 Results

Results:

1. Seamless Workflow: AutoFlow users can now utilize AutoKeras for model generation, improving the overall workflow.
2. Optimized Neural Networks: Customized AutoKeras produces neural network architectures optimized for embedded devices, enhancing resource utilization and performance.
3. Resource Awareness: AutoKeras considers hardware constraints when generating neural networks, enhancing suitability for embedded platforms.

Future Developments:

1. Fine-Tuning and Benchmarking: Refinement of the embedded device optimization module and extensive benchmarking on various embedded hardware platforms.
2. Platform Expansion: Expanding supported embedded platforms to accommodate diverse hardware constraints.
3. Community Contributions: Encouraging contributions from the AutoFlow community to enhance embedded device optimization.

Conclusion:

The integration of AutoKeras into AutoFlow and its customization for embedded devices marks a significant milestone in simplifying machine learning deployment on resource-constrained platforms. This enhancement benefits developers and researchers working on machine learning applications for IoT and edge computing.

3.10.4 IMOCO4.E Requirements

- R236-D2.3: All methods and algorithms were tested on a Nvidia Jetson Orin.
- R237-D2.3: FPGA hardware was ordered successfully.

3.10.5 Capabilities and Limitations

The unique selling point of the AutoFlow framework, enhanced by AutoKeras integration, lies in its pioneering approach to simplifying machine learning (ML) application development for embedded devices. By offering a user-friendly graphical interface that seamlessly incorporates AutoKeras, AutoFlow sets itself apart as a groundbreaking tool that democratizes the creation of optimized neural network architectures. This integration is specifically tailored to overcome the challenges of limited memory, processing power, and energy efficiency inherent in embedded systems, making it an invaluable asset for developers and researchers in the IoT and edge computing sectors looking to streamline their ML workflows.

Strengths:

- **Simplified ML Workflow:** AutoFlow with AutoKeras integration provides a streamlined, user-friendly GUI for ML development, significantly reducing the complexity and technical barriers for developers, especially those focusing on embedded systems.
- **Optimized for Embedded Devices:** The framework's ability to customize neural networks for embedded devices ensures optimized performance and resource utilization, addressing the critical constraints of memory, processing power, and energy efficiency.
- **Community and Future Growth:** The focus on future developments, including platform expansion and encouraging community contributions, positions AutoFlow for continuous improvement and adaptation to emerging needs in embedded device optimization.

Weaknesses:

- **Limited Initial Platform Support:** While the framework aims to expand, the current supported embedded platforms might be limited, potentially restricting initial adoption among developers with diverse hardware requirements.
- **Dependence on Community Contributions:** The emphasis on community contributions for future enhancements could lead to variable progress and innovation rates, depending on the engagement and expertise of the community.
- **Learning Curve and Integration Challenges:** Despite the user-friendly GUI, there might still be a learning curve associated with mastering AutoFlow and AutoKeras integration, especially for those new to ML or specific embedded platforms. Additionally, integrating with existing development pipelines could present challenges without robust documentation and support.

3.10.6 Customizations and Adaptations

Use in the Product/Design/Use Case (P/D/UC):

AutoFlow, with its AutoKeras integration, is particularly beneficial in the product, design, or use case (P/D/UC) scenarios that involve the development of intelligent embedded systems, such as IoT devices and edge computing solutions. It significantly streamlines the design process by enabling automatic, optimized machine learning model creation tailored to the specific constraints and requirements of embedded devices.

Customization Potential/Options:

The framework offers substantial customization potential, allowing developers to specify constraints and requirements for their embedded systems, such as memory limitations, processing capabilities, and energy consumption. This ensures that the generated neural network models are not only optimized for performance but also conform precisely to the device specifications.

Possible Modifications and Extensions:

AutoFlow and AutoKeras's architecture allows for various modifications and extensions, such as the addition of support for newer machine learning algorithms, incorporation of more advanced optimization techniques for embedded systems, and expansion to cover a broader range of embedded hardware platforms. Additionally, there's potential for integrating domain-specific model evaluation metrics to further refine the optimization process for targeted applications.

3.10.7 Methodology and Toolchains

The AutoFlow framework, enhanced by AutoKeras integration, fits into modern machine learning toolchains by offering a bridge between high-level ML model design and low-level optimization for embedded devices. It complements existing development environments and tools, such as TensorFlow for model training and deployment, and platform-specific SDKs for embedded systems, creating a cohesive pipeline from model inception to deployment.

The development and implementation of AutoFlow with AutoKeras integration have underscored the importance of user-friendly interfaces in democratizing access to complex machine learning workflows, especially for embedded systems. This approach has proven to significantly lower the barrier to entry for developers and researchers, allowing for more innovative applications of ML in constrained environments. A key lesson learned is the critical need for balance between automation and customization; while automation speeds up the development process, ample customization options are essential for addressing the diverse needs of embedded devices. Furthermore, the initiative highlighted the value of community feedback and contributions in evolving the platform to meet emerging needs. Challenges such as tool integration with existing workflows, addressing the unique constraints of various embedded platforms, and ensuring scalability of the solution were identified as areas requiring ongoing attention and refinement. These insights emphasize the iterative nature of tool development in the fast-evolving field of machine learning and embedded systems, pointing to the necessity of flexible, adaptable solutions that can evolve in response to new challenges and opportunities.

3.11 Multiple sensors (UNIMORE)

3.11.1 Technology overview

Hypervisor based solution

The presence of multiple sensors means that data flows of different natures and priorities need to be managed. To oversee these sensors data flows on a platform, we can use a virtual machine for each sensor. A VM based solution offers isolation between sensors and can be tuned to allow more resources to the high priority sensors VM.

Resources as I/O ports need to be shared between sensors, and a hypervisor-based solution allows the sharing of such resources according to the priority of the different VMs. These features are available natively in hypervisors like Xen or must be developed, as we did with our Virtual TSN solution for Jailhouse Hypervisor.

NFC reader communication

UNIMORE provided a software component that can read and store data from NFC readers internal storage in the context of Demo 2. This component enables the communication between the NFC reader and the rest of the system, as well as the processing and storage of the data collected by the reader.

3.11.2 Implementation aspects

Hypervisor based solution

In pilot 3 we implemented 2 sensors, a sensor for health of the cutting blade and a camera for the bottles. Both were emulated on a Xilinx ZCU102 as virtual machines (VMs) under Jailhouse Hypervisor supervision, the layout is presented in Figure 95. Each VM used GStreamer to generate the flow of data of each sensor. Two other VMs were used, one for the actuator and the last one for our Virtual TSN (VTSN) solution that will be presented in D4.8. VTSN allows the sharing of the Ethernet port of the Xilinx ZCU102 between the VMs of the sensor blade and the camera.

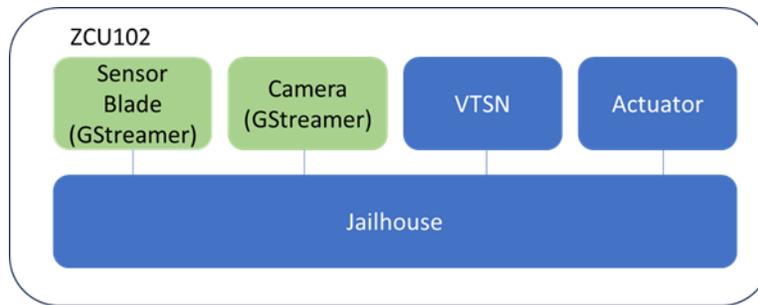


Figure 95. Xilinx ZCU102 Configuration for Pilot3.

NFC reader communication

In the context of Demo 2, the software runs on the NVIDIA Jetson AGX Xavier, which is connected to an NFC Reader (μ FR Nano Online Network Reader by Digital Logic) as illustrated by Figure 96.



Figure 96. NFC Reader setup.

3.11.3 Results

Hypervisor based solution

The latency and the bandwidths (UDP and TCP) of the data flows generated as the output of the Xilinx ZCU102 for the two sensors used in Pilot 3 have been evaluated during the integration days of Pilot 3. The values for the latency and the bandwidth are presented in Table 22.

Table 22 Average latency and bandwidth for sensors in Pilot 3

Sensor Name	Average Latency	Average Bandwidth (TCP)	Average Bandwidth (UDP)
Sensor Blade	0.079ms	934Mbps	387Mbps
Camera	0.089ms	856Mbps	313Mbps

NFC reader communication

UNIMORE has successfully tested the software for Demo 2 and verified that it can read and store data from NFC cards on the hard drive attached to the NVIDIA Jetson AGX Xavier as shown by Figure 97.

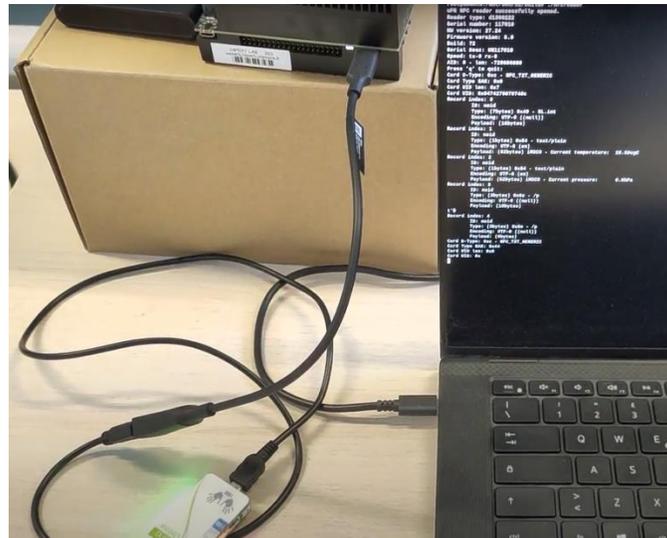


Figure 97. Data from NFC card printed on screen

3.11.4 IMOCO4.E Requirements

- Req-D3.1- L1-hw Used vision sensors are easy to connect to a PC-based processing unit (USB2, USB3) : ZCU102 can support USB using USB3 micro-B to USB3-USB2a adapter.
- Req-D3.1- L1-D2-hw Sensors must have a reader/ controller connected to upper layers (through BB1 or BB4) by USB or Ethernet: ZCU102 can support USB using USB3 micro-B to USB3-USB2a adapter and has one ethernet port.
- Req-D2.3 BB1 shall support standard and vendor-neutral Wired 1G Ethernet: ZCU102 cans support 1G ethernet connection.

3.11.5 Capabilities and Limitations

The number of sensors managed by our components is limited by the hardware used. This limitation is due to the utilization of a hypervisor that does not fully virtualize the resources of the hardware platform. Each sensor must be assigned a core that is not shared with the other sensors. This limitation is mandatory to guarantee the time constraints, as it isolates the sensors of the system from each other and limit the resource contention between them.

3.11.6 Customizations and Adaptations

The number of sensors managed using Jailhouse hypervisor is limited by the number of cores of the platform chosen, in Pilot 3 the Xilinx ZCU102. Indeed, each sensor must have a reserved core for themselves that cannot be shared with another sensor. The number of I/O ports, however, can be increased using FPGA Mezzanine Card (FMC).

3.11.7 Methodology and Toolchains

The Jailhouse hypervisor has been chosen for its low impact on the real time performance of the systems and for its low code base which make easier its customization. In Pilot 3, sensors have been emulated using GStreamer, a library that allows the production of stream of data such as video.

3.12 Vision-in-the-loop system (SIOUX)

3.12.1 Technology overview

Figure 98 shows the block diagram of the vision-in-the-loop for motion control for Pilot 2. The motion control system consists of a controller which calculates the required force based on the current position of the die. This position is estimated by a Kalman Filter which receives two position measurements, one from the encoders, and one from vision, at different frequencies. The vision position is determined by a vision process that receives an image taken by a camera on the host platform, and extracts the die position from this image. The ADAT host platform, either the physical setup or a simulation, closes the loop. The vision process uses Deep Neural Network (DNN) to detect the position of the die. Specifically MobileNet with modified prediction layers was used to obtain the position of the die.

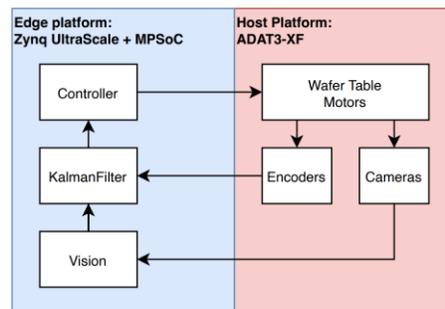


Figure 98. Vision-in-the-loop system and deployment block.

3.12.2 Implementation aspects

To achieve the required latency for the vision-in-the-loop system, a part of the processing is implemented on MPSoC edge platform (shown in Figure 99). The architecture of Zynq UltraScale + MPSoC is divided in Processing System (PS), containing multiple microprocessors and a Programmable Logic (PL). These components are used for the implementation of the DNN, image pre-processing and post-processing stages,

and control logic. Several implementation of the vision-in-the-loop controller were explored while considering the baseline case as running the DNN on ARM cortex A53 block.

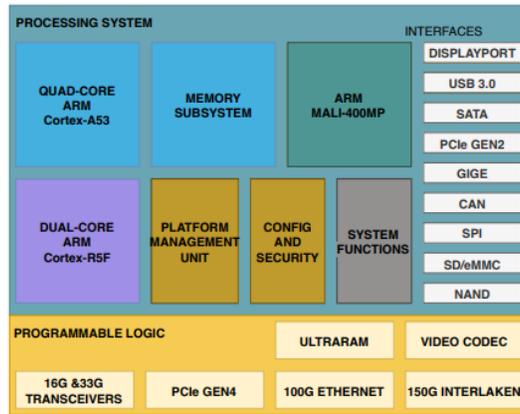


Figure 99. Zynq UltraScale+ MPSoC platform diagram.

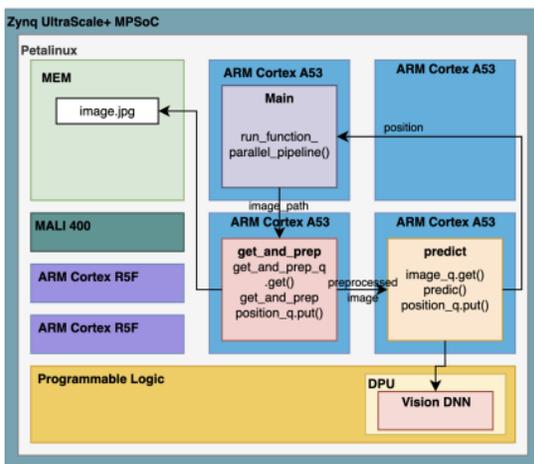


Figure 100. Mapping for functional parallelism.

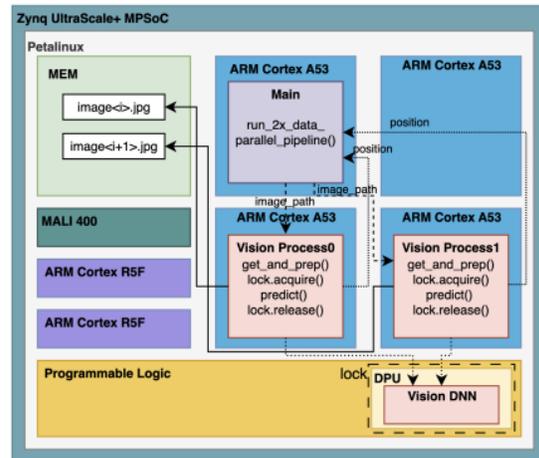


Figure 101. Mapping for 2x data parallel pipeline.

3.12.3 Results

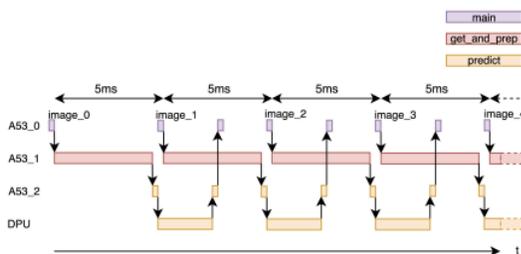


Figure 102. Function parallel pipeline.

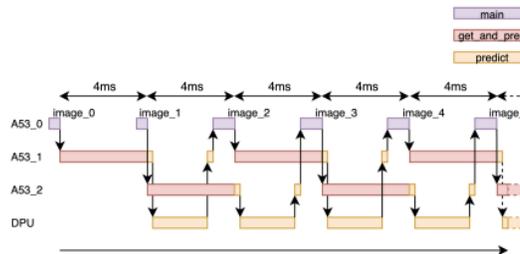


Figure 103. 2x Data parallel pipeline.

4 Solutions of Task 3.4: AI-based components (BB2, BB8)

4.1 Overview

4.1.1 Solutions of T3.4

Partners of Task 3.4 worked on 9 AI-based solutions contributing to BB2 and BB8 – developing AI-based perception and control solutions as well as integrating such solutions in high-speed Vision in the Loop systems. The developments span from enhancing the adaptability of smart industrial robots to improving precision in semiconductor manufacturing and automating logistical operations in warehouses. Key themes include the integration of AI for enhanced object recognition and environment sensing, the fusion of deep neural networks (DNNs) with traditional feedback systems for superior motion control, and the deployment of high-speed vision systems for micro-scale accuracy in manufacturing processes.

These solutions leverage cutting-edge technologies such as synthetic data generation, Sim2Real translation techniques, and advanced networking through Time Sensitive Networking (TSN) to bridge the gap between virtual simulations and real-world applications both in the term of realism and in the terms of latency. The focus on developing realistic simulation frameworks, particularly in radar technology for scene recognition, underscores the push towards creating more accurate models that reflect complex real-world conditions.

Embedded AI emerges as a critical component in real-time predictive models, especially in gesture recognition and robotic teleoperations, indicating a move towards more intuitive and responsive systems. The integration of AI at both the sensor level and in remote systems facilitates immediate data processing and decision-making, enhancing the efficiency and effectiveness of automated systems.

The use of Field-Programmable Gate Arrays (FPGAs) for AI model inference and video compression further illustrates the shift towards edge computing, enabling faster and more reliable video analytics applications critical for security, surveillance, and process monitoring in real-time.

Collectively, these initiatives represent a significant stride towards incorporating AI into the fabric of industrial and technological operations, aiming for greater adaptability, precision, and efficiency. The emphasis on real-world application and validation of these technologies ensures their relevance and potential impact on future developments within these sectors.

4.1.2 Addressed ST objectives and KPIs

Solutions of the task addressed three scientific and technological development objectives of IMOCO4.E - ST1, ST2, ST3.

ST1 aims to develop advanced model-based and knowledge-based methods for building digital twins for design, optimisation, customisation, virtual commissioning and predictive maintenance of machines and robots using existing and novel datasets. While WP5 was dedicated specifically to digital twins, the WP3 developed enabling components. Solution by EDI (section 4.2) was dedicated on connecting learning on a virtual robot, with actions on the real counterpart. The same solution, as well as one by IML (section 4.6) provided synthetic data generation to efficiently acquire new datasets. TUE (section 4.4) used digital twin framework to test their solution of AI-based object detection and transferred the solution to a physical system.

ST2 aims at developing a smart Instrumentation Layer with data gathering and processing at the edge to enhance the achievable performance and energy efficiency of systems. BB2 components developed in Task 3.4 directly contributed to this goal by providing ultra-high speed vision with AI-based scene recognition. Solution by TNO (section 4.5) provided a smart motion sensor based on a high-speed camera for positioning tasks requiring micrometer precision. Solution by GNT and ITML (section 4.3) brought pre-processing and clustering algorithms to the edge, while solution by IML does pose estimation and solution by EMD (section 4.9) does gesture recognition on the edge devices. Most of the solutions of the Task 3.4 addressed the issues of performance and efficiency of sensor data processing and contribute towards ST2, even if not all are implemented on the edge. For example, Time Sensitive Networking solution by SED (section 4.10) targets low-latency data delivery, crucial for real-time systems.

ST3 is dedicated to a smart Control Layer which should be modular and include edge intelligence. Even though most of the T3.4 solutions do not directly target control, the high-speed vision in the loop provided by some of the solutions is a crucial enabling component for the ST3. Also, modularity of provided perception solutions contributes to easier integration of components from different layers.

BB2 KPIs:

- High rate ($1 > \text{KHz}$) image processing unit for megapixel sensor for industrial 'in the loop' control.
 - TNO solution (section 4.4) demonstrates parallel and deterministic execution for high-speed vision tasks and provides possibility to have $> 5\text{kHz}$ frame rate of > 0.25 Mpixel frames with latency $< 300 \mu\text{sec}$.
- Real time image processing and machine learning on embedded calculation hardware (FPGA, multi many core etc) integrated to above image acquisition.
 - IML solution (section 4.6) provides GPU edge-based real-time video processing, which is embedded into a single system together with cameras, processing units and AI models.
 - EMD solution (section 4.9) provides real-time (30 FPS) machine-learning based recognition, embedded on an FPGA-based edge device.
 - SOC-e solution (section 4.11) provides an FPGA-based hardware component for acceleration of AI models, enabling real-time video processing on the edge.
 - Reexen solution (section 4.8) contributes towards the embedding of high-performance computing and image processing operating in a single embedded system by providing CIM (computing in memory) technology.

BB8 KPIs:

- The current performance of AI-based vision-in-the-loop functionality driven by such edge computing technologies as embedded GPUs, FPGAs, MPUs, ASICs.
 - Reexen solution (section 4.8) leverages in-memory computing as a HW acceleration technique.
 - SOC-e solution (section 4.11) demonstrates DPU usage for accelerating neural network execution as well as uses Time-Sensitive Networking to ensure deterministic transmission of critical information.
- The possibility and correctness of applying the reinforcement learning techniques for realizing control algorithms and transferring learned models to physical setups.
 - EDI solution (section 4.2) provides a pick & place system with the ability to transfer movements learned by RL in simulation to a real robot.
- Hardware (ASIC) realization of Deep Learning Techniques and Neural Networks, and the length of achievable real-time control loop.

- Reexen solution (section 4.8) explores CIM technology as a hardware technique for accelerating Deep Learning.
- SOC-e solution (section 4.11) for HW accelerating Deep Learning models is prototyped on an FPGA.

4.2 Real-time AI-based processing component, using modern reinforcement learning techniques with the long-term aim of reducing the setup and deployment time of AI-based industrial machinery (EDI)

4.2.1 Technology overview

EDI worked on a solution for smart industrial robots that would allow easier adaptation of such robots to work with new objects on the production line. The solution includes AI-based perception of the objects and the environment as well as robot planning and control systems. In current implementation, the solution is adapted for a bin-picking task and bottle-shaped objects. The focus of the solutions was on making the perception as well as planning and control to be adaptable by personnel who is not machine learning and robotics experts by incorporating use of synthetic data, robot simulators, reinforcement learning, and Sim2Real translation approaches. Several modules were developed and integrated into a single system, however three BB8 components were at the core of our developments:

- Perception models trained to specific tasks (SW-006).
- Sim2Real image translation module (SW-004).
- Sim2Real Reinforcement learning (SW-005).

Perception models, specifically object detectors, are a well-established technology and widely used product of machine learning. Specific demonstrator requested a fine-tuning of foundational model, which requires original labelled training data. In IMOCO4.E we used 3D modelling and simulation in Blender software to create synthetic dataset. Such approach accelerates training process, however, Sim2Real gap leads to lower accuracy scores of perception models than the ones trained on large datasets of real images.

Aim of **Sim2Real** image translation module is to reduce this gap between synthetic and real images, by making synthetic images look more like real ones. In several iterations, a Generative Adversarial Network-based approach was improved until a satisfactory Sim2Real translator was trained and tested.

Reinforcement Learning in simulation was introduced to enable the training of robots' motion planning and control. The most challenging and experimental part of this development was to transfer motions learned in the simulator to a real robot. An integrated system was developed that combines virtual robot, real robot, as well as Sim2Real translation modules. Next steps are to validate if such a system is usable for a real-world task.

4.2.2 Implementation aspects

Perception models

First task of the deployed and fine-tuned system is to detect a pickable object in a pile of similar objects (Figure 104). Also, the direction of objects is determined so that robotic arm can orient and use a two-finger gripper for the picking. The bounding boxes of pickable objects are detected by trained convolutional neural network. Usually, it is trained with labelled real images. In order to reduce the necessity of large amounts

of real labelled data, our solution incorporated significant amount of synthetic data that was pre-processed by Sim2Real module described below. Direction of detected bottles is detected by a custom image processing approach. This approach also uses a depth map of the scene to better delineate among similar objects. The depth map is also required to get 3D positions of the objects necessary for planning robots' movements.

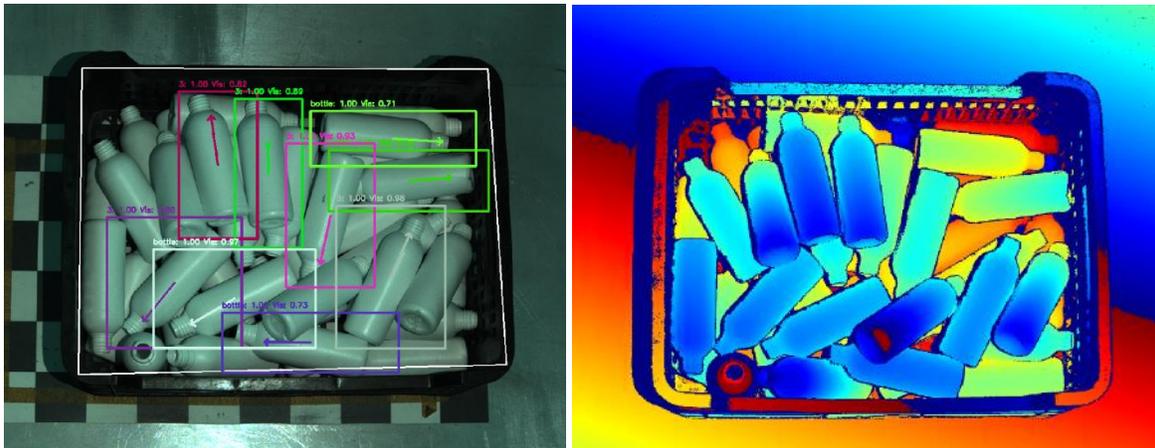


Figure 104. Object detection in RGB image and corresponding depth map.

Sim2Real

Developed AI-based Sim2Real module is capable of making images from simulations look more like real images and has several applications in the developed robotic system. It can be used to make synthetic images acquired in 3D modelling software be more like real-world frames of an RGB camera. This approach was used to augment training datasets of the perception models. Additional module Real2Sim, which is trained simultaneously with Sim2Real, can be used in real time to translate real images acquired by robots' cameras into domain of synthetic images. In this case, object detectors trained on the synthetic images can be used for the perception after such pre-processing step. These modules also work as a connecting adapters between the real robot and it's simulated twin and can be applied either to the real camera or the virtual camera in the simulation.

Translation models are based on CycleGAN [1] architecture, leveraging the foundational architecture of the Residual Network (Res-Net) generator. Additionally, a series of adaptations were implemented to enhance the stability of training procedures and augment the fidelity of resultant image outputs:

- Several image augmentations like random crop\center crop\image shift.
- Label smoothing.
- Addition of Gaussian noise to discriminator.
- Weight demodulation instead of common Instance Normalisation [2].
- Spectral Normalization in discriminator [3].

Trained models are saved in Keras .h5 format and take only 9.4 MB. When running alongside the object detection module, both models together use 1.7 GB VRAM, making them suitable for deployment even on portable devices.

RL-based control system

Object detector sends information to MoveIt2 platform which performs robots’ motion planning (Figure 105). Through ROS2 “Humble” the planned motions are executed on the real robot. During the project, the MoveIt planner was also bridged to Ignition Gazebo simulator. In this simulator, an environment with the virtual robotic arm UR5e was prepared. In this environment, Stable Baselines3 Reinforcement Learning algorithms were used to learn robot movements. Although the final ambition is to learn full task of picking an object and putting it into a movable socket on the conveyor belt, during IMOCO4.E project the task was simplified. RL was used so that simulated robot would learn to get the gripper to a position indicated by the object detection module.

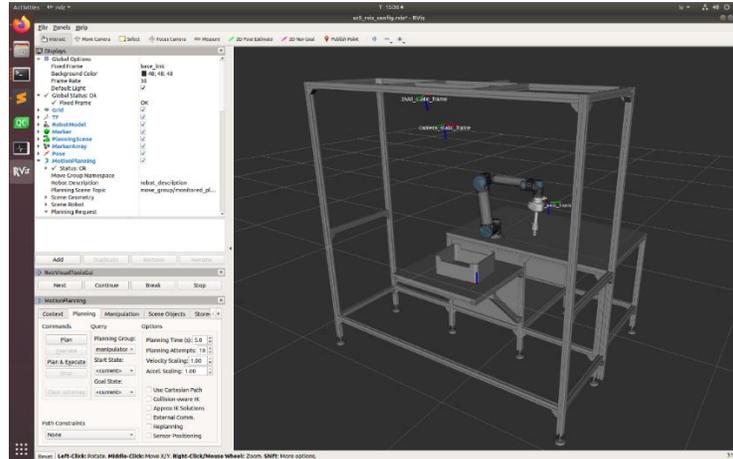


Figure 105. Motion planning using MoveIt2 in ROS2, visualised with RViz.

Integrated system – Demonstrator 4

Developed software components were integrated into a demonstrator depicted in Figure 106 along with the full architecture. As quite a few software tools had to be integrated and versions aligned, for the future portability and deployment, the final solution was containerized using Docker. Main hardware components were the robotic arm UR5e, a gripper Robotiq 2f-140, a Realsense camera, and a GPU-equipped PC.

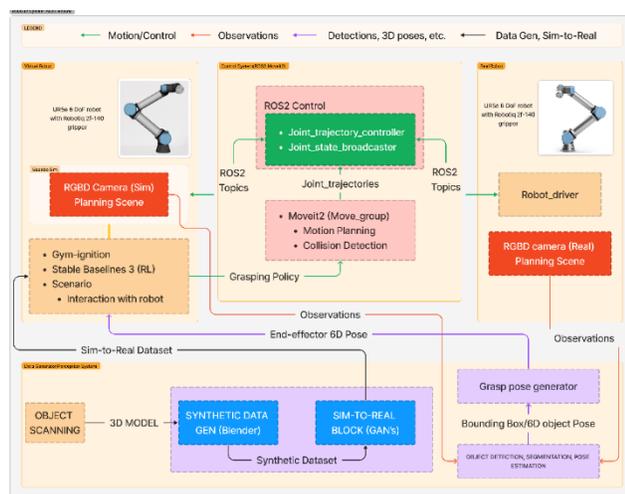


Figure 106. Demonstrator 4 in the laboratory environment and its architecture.

4.2.3 Results

Perception models of different size were tested on prepared test sets, measuring precision, recall, and mAP values. Table 23 Table 23. Results of the object detection, also described in paper [4], depicts that each model was trained on three different datasets – one contained original synthetic images, while the other two contained images augmented by Sim2Real translation. The test set consisted of real images.

Table 23. Results of the object detection, also described in paper [4].

Model	Dataset	Precision	Recall	mAP (threshold 0.5)	mAP (avg for IoU $\in [0.5 : 0.05$ $: 0.95]$)
YOLOv5 Small	Original synthetic	0.689	0.894	0.742	0.447
	Resized	0.615	0.838	0.632	0.245
	transpose Augmented noise	0.731	0.905	0.777	0.48
YOLOv5 Medium	Original synthetic	0.693	0.802	0.72	0.42
	Resized	0.581	0.782	0.589	0.204
	transpose Augmented noise	0.712	0.91	0.753	0.455
YOLOv5 XLarge	Original synthetic	0.718	0.785	0.75	0.413
	Resized	0.683	0.867	0.715	0.339
	transpose Augmented noise	0.721	0.872	0.761	0.461

Sim2Real models were tested in stages. First, subjective visual inspection of translated images indicated if those images are like the real images, and if there are some obvious artefacts. The combination of proposed modifications to the vanilla CycleGAN made it possible to achieve good image quality at a resolution of 512x512 pixels, without severe blur or additionally drawn non-existent objects, or other various artifacts (Figure 107. Sim2Real translation example. Synthetic image on the left is made more lifelike and displayed on the right side). To evaluate the quality of generated images in a more objective manner and choose the most promising modifications, we used the FID (Frechet inception distance) score [5]. This score calculates a distance between reference real images and the images resulting from Sim2Real translation. The lower the FID score, the more similar are the two datasets. With the most promising combination of improvements, we achieved FID score of 52, compared to the initial baseline of 230 achieved by the vanilla CycleGAN.



Figure 107. Sim2Real translation example. Synthetic image on the left is made more lifelike and displayed on the right side.

The next stage to test performance of Sim2Real was to use translated images as training data for object detection models and measure the detection performance on a test set of real images. Sim2Real improved detection results compared to models trained on synthetic data without application of Sim2Real. However, the results didn't show a clear correlation between better FID scores and better performance of the detectors. An important conclusion we gained is that FID score is not a quick performance measurement that could replace much more complex process of training and testing perception models with data processed by different Sim2Real models.

In addition to a GPU-equipped PC, perception and Sim2Real models were also implemented on a NVIDIA AGX Jetson device. At the project's second review meeting, the embedded system was demonstrated to reviewers and partners, and it was shown that both models can do simultaneous inference (image translation and object detection) in real time on an edge device.

RL-based motion control system at this stage of the project is not yet validated. Preparation of this system for validation is the main focus of Demonstrator 4 in the coming months.

4.2.4 IMOCO4.E Requirements

Table 24. Requirements related to solutions for Demo 4.

Req ID	Requirement Description	Verify	Result	Comments & Rationales
R004-D2.3-D4	Internet connection for the possibility of remotely inspecting the behaviour of perception and control modules.	D	PASS	
Req-D4-002	Needs to satisfy formal regulations: RoHs.	I	PASS	No component of the solution violates RoHs.

Req-D4-005	Demonstrator should have input HW and output monitors/screens so it is possible to manually inspect and correct perception and control modules on site.	D	PASS	
Req-D4-011	Overall cost of deploying the demonstrator (without R&D) < 200,000 EUR.	I	PASS	All the developed components of the solution together with off-the-shelf hardware (including robots) costs less than 200,000 ER
Req-D4-014	Compliance with safety standards. People should be safe to operate in the same room the demonstrator is operating.	I	PASS	Developed solution is used with a collaborative robot UR5.
Req-D4-015	Compliance with good manufacturing practice. Parts must not disconnect and contaminate product or packaging.	D	PASS	
R185-D2.3-B8-D4	Sim2Real transfer provides synthetically trained object detection algorithms that detect objects of interest in 80% of images with said objects.	D	PASS	Sim2Real transfer have been demonstrated in second review meeting of the project.
R057-D2.3-L3-D4	Systems perception models should be retrainable by a non Machine Learning expert.	D	IN PROGRESS	Tests on training perception models with synthetic and Sim2Real translated data has demonstrated satisfactory results. The next step in demonstrating such capability is a simple user interface integrated into a full demo.
R003-D2.3	Demonstrator can be adjusted to several conveyors/production lines.	D	IN PROGRESS	Although different modules for easier adjustment of the demonstrator were developed and tested, all the tests were still done with the same type of object.
Req-D4-008	Number of successful picks from a random pile in a minute up to 10.	T	IN PROGRESS	Suction cup-based picking of bottles already can satisfy the requirement; however, the demonstrator targets picking with a gripper, so that it can be

				used and adjusted to broader range of object types.
Req-D4-009	95% successful placements of the bottle into a socket on first try.	T	IN PROGRESS	Tests of this requirement are not yet started as the control module for placing bottles into a socket is still in development.
R094-D2.3-B2-D4	Object detection and pose estimation model processes > 5 frames per second.	D	PASS	Pose estimation allows necessary inference speed without the need for high-end hardware.

4.2.5 Capabilities and Limitations

The system's unique selling point is its ability to simplify product line adjustments for new products through an automated process, thereby reducing the need for machine learning expertise for every product change. At current stage, training of perception models with transformed synthetic data is the most ready and tested part of the solution.

Strengths:

- Customization options of the solution (more in the following subsection).
- System integrates virtual and physical robots so that both are controlled by same software tools. This makes the transfer of motion between one domain and the other simpler in terms of interfacing the robots.
- A modular architecture and containerization using Docker. This allows further modification and improvement of the system, which should be faster and easier than creating this system in the first place.
- Efficient inference. AI-based perception includes not only common real-time object detection module, but also Sim2Real image translator. Both modules have been demonstrated to run in parallel in real time on the same edge device.
- Generation and use of synthetic data for training perception models. This ability reduces the amount of necessary real labelled training data.

Weaknesses:

- Complexity. Setting up and configuring the whole system initially may require significant effort and troubleshooting. Because of the many interconnected tools, some of which are new and still in active development, dependency management is challenging. If some of the used tools have significant upgrades, for example, Ignition Gazebo increases the realism of the simulations, integrating these updates into our system would be time consuming and would require specialists in both – machine learning and robotics.
- Simulation realism. Currently used simulation environment of Gazebo Ignition is not the most realistic simulator when it comes to graphics. This limits the performance of the real robot, as our Sim2Real modules aren't removing the Sim2Real gap completely. For example, better visual

realism might be provided by Isaac Sim, but a change of simulator would lead to compatibility problems with the whole Demonstrator 4 system.

- Required computational resources. Even though inference of used models is efficient, the possibility to retrain them requires greater GPU resources and takes time, which have to be planned in the factory setting. For Sim2Real models - larger training image sizes require more vRAM, hence it is not recommended to use images larger than 512x512 pixels. However, to maintain the quality of translated images, it is advised not to use images smaller than 256x256 pixels.
- Perception and Sim2Real models have not been tested on transparent objects. Semi-transparent objects may also be misinterpreted by the neural network, and fully transparent objects pose a greater challenge.

4.2.6 Customizations and Adaptations

Customisation is at the core of the developed solution, as it aims to be easy retrainable to work with different kind of objects and to make different kinds of movements. At current stage, the readiness of customizable perception system is high, while the planning and movement parts are still experimental.

As for separate components, object detector is retrainable for other objects, while the data generation pipeline makes it easy to acquire necessary training data (synthetic images) given a 3D scan of the object of interest.

Sim2Real module can be used not only to translate synthetic images of bottles to more realistic ones, but to translate any kind of synthetic images into real looking counterparts. However, this requires acquisition of relevant datasets. It requires a dataset with more than 1000 realistic and synthetic images. The images can be unpaired, meaning they do not exactly match each other, but should be similar in terms of object size, lighting, and quantity.

The process of adjusting settings for training is fast and simple by manually entering parameters like the size of the dataset and image resolution. Res-Net models have no limitations on the size of images for training and processing, given the conditions of training for full image style transfer and a dataset size of 1000 images; no adjustments are necessary, and it is better to empirically evaluate the image translation results and stop training if the image quality is satisfactory.

When transitioning to a new data type, the training process must start from scratch, as using transfer learning may recreate features that do not exist in the new data type. It is estimated to take approximately 1-3 hours using an A100 GPU and 20 epochs. Checkpoints of the model are created every 5 epochs during training, facilitating the ability to interrupt training if needed and resume later.

With more modifications, this image translation approach is usable even more broadly than changing synthetic images into realistic images. EDI has already extended this method and used it for virtual staining of veterinary histology images [6].

While retraining of object detector and Sim2Real image translator can cover customisation towards new products for robot to handle with different shapes, sizes, materials, the RL-based motion planning could potentially be applied to a different robot and different task altogether. This would require an availability or development of the description of the specific robot for the specific simulated environment. However, in the scope of IMOCO4.E project, it remains untested if used RL approach would perform well with different kinds of robots. Also, depending on the support of the specific robot by ROS2 and other used modules, in the full pipeline which integrates the real robot with its virtual counterpart would require small or extensive modifications.

4.2.7 Methodology and Toolchains

The development toolchain of the Demonstrator 4 is extensive. All components are running on an Ubuntu system.

Object detection and Sim2Real translation modules require a Python environment with TensorFlow framework for AI and OpenCV library for additional processing of images and reading data from camera. Trained AI models for detecting specific objects and Sim2Real model for a specific environment can be already pre-stored on the system for use. However, to adapt system to new objects, fine-tuning of the models takes place on the same system in the same virtual environment. This requires CUDA-equipped GPU resources on the system.

Robot motion planning and control involves Ignition Gazebo, ROS2 Humble, MoveIt2, reinforcement learning frameworks like Stable-Baselines3 and PyTorch, as well as code for communication between components, such as `ros_gz_bridge` and `gz-transport`, and custom code.

Some components can be used separately from the whole system, especially Sim2Real translation module is well tested on its own.

Integrating all components together was the most time consuming and challenging part of developing the Demo 4 system. Main contributing factors to this challenge were compatibility of tools and their versions as well as lacking documentation of some of the tools. To operate with the state-of-the-art tools, EDI chose newer versions of ROS and Gazebo in the beginning of the project (ROS2 and Ignition Gazebo). This led to longer development of the demonstrator system; however, the resulting system should be great up to date basis for further developments and projects.

4.2.8 References

- [1] J.Y. Zhu, T. Park, P. Isola, and A. A. Efros, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. 2020.
- [2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, Analyzing and Improving the Image Quality of StyleGAN. 2020.
- [3] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, Spectral Normalization for Generative Adversarial Networks. 2018.
- [4] D. Duplevska, M. Ivanovs, J. Arents and R. Kadikis, "Sim2Real image translation to improve a synthetic dataset for a bin picking task," 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, Germany, 2022, pp. 1-7, doi: 10.1109/ETFA52439.2022.9921431.
- [5] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," in Advances in Neural Information Processing Systems, 2017, vol. 30.
- [6] D. Duplevska, R. Maliks, M. Tamošiunas, M. Melderis, D. Viškere, B. Cugmas, R. Kadikis, I. M. Houtana, "Interpreting microscopic structures in virtually stained histological sections for veterinary oncology applications", Photonics West 2024, Advanced Biomedical and Clinical Diagnostic and Surgical Guidance Systems XXII.

4.3 AI-based, real time data clustering and classification at the edge, including deployment of a data fusion mechanism able to aggregate, pre-process, and further analyze data from multiple different IoT sources in Industrial environments (ITML, GNT)

4.3.1 Technology overview

The work reported here corresponds to **solution #2 of T3.4** (referenced as Module #2 in the DoA): “*AI-based, real time data clustering and classification at the edge, including deployment of a data fusion mechanism able to aggregate, pre-process, and further analyze data from multiple different IoT sources in Industrial environments*”. It is developed jointly by GNT and ITML and is registered as the IMOCO4.E component **SW-018 of BB8** in the WP6 Software catalogue. GNT focuses on the design and implementation of the ML approach, while ITML focuses on data pre-processing.

Solution 2 (SW-018) implements **clustering analysis of unlabelled datasets for determining classification categories and training a corresponding classifier ML model, which is then deployed at the edge and tasked with the prediction of the category of each incoming data point in real time**. The approach is complemented with a data fusion mechanism that is delivered by BB9 in IMOCO4.E and is used for transferring pre-processed data to the cloud for training purposes and for transferring the classification results for storage and presentation to the user under operational conditions. The following diagram in Figure 108 illustrates the envisaged operation of SW-018.

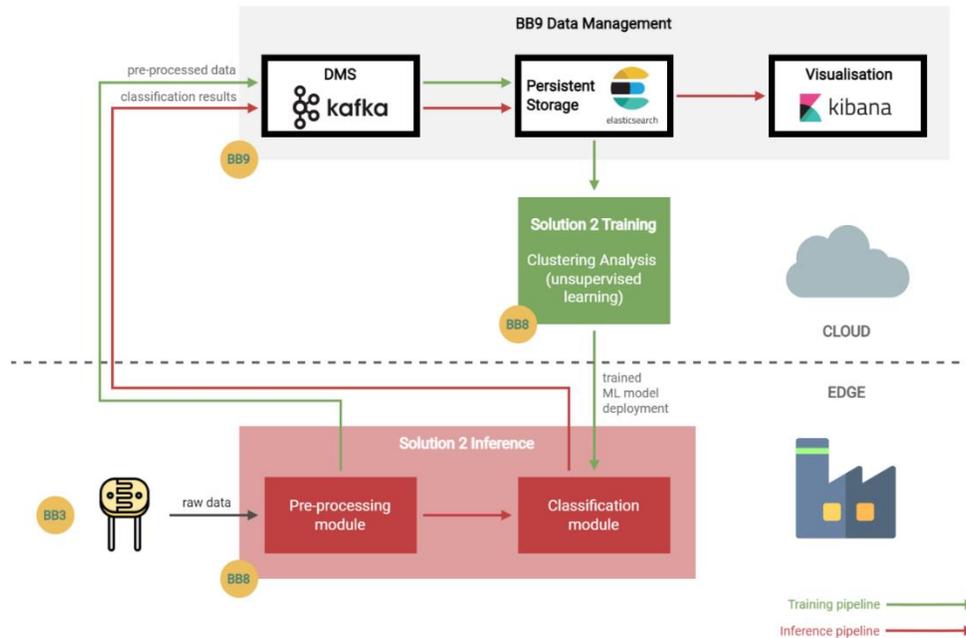


Figure 108. Block view diagram of SW-018 (Solution 2) operation, including interconnection with other IMOCO4.E BBs and positioning on edge and fog layers.

Edge devices that have direct access to the input sensor data streams are foreseen to host the classifier module, which is coupled with a pre-processing module that performs any necessary data transformation

tasks to bring the data to a suitable format. The pre-processing module feeds the transformed data to the classifier module, which in turn processes them in real time and generates a prediction of the category of each data point as an output (classification results). Both the pre-processed data and the classification results are relayed to the cloud by implementing a Kafka producer client that publishes these data as messages to a Kafka broker provided by BB9 and hosted on cloud infrastructure. The data received by the BB9 Kafka broker are also persisted to an Elastic Search repository. The collected pre-processed data at the Elastic Search can be used for follow-up clustering analysis and re-training of the classifier model. Resulting updated ML models can be re-deployed at the edge for achieving improved real-time classification performance. The clustering analysis, classifier model training and deployment of updated models can be configured to take place according to a schedule or rule-based conditions.

4.3.2 Implementation aspects

Application in Pilot 3. SW-018 is applied in Pilot 3, which refers to the High Speed Packaging industry and is focused on improving current machine monitoring activities and quality control processes. It was decided to apply SW-018 in Pilot 3 for the purpose of predictive maintenance of industrial equipment. In this context, SW-018 is meant to analyse input sensor data streams at the edge and detect anomalies that are associated with early stages of industrial equipment performance loss and indicate the need to perform equipment maintenance.

Dataset. The “Vega shrink-wrapper component degradation” dataset published on Kaggle (<https://www.kaggle.com/datasets/inIT-OWL/vega-shrinkwrapper-runtofailure-data>) was provided in the context of Pilot 3. The dataset focuses on the degradation of the cutting blade in the Vega shrink-wrapper machine, specifically comparing the performance of a new blade to a worn-out blade. The dataset includes six (6) data snippets as separate CSV files, each of which is 8 seconds long and consists of 2048 data points. The dataset compares a new blade to a completely worn-out one, with a focus on deviations in the cutting process. The dataset is labeled, as three (3) of these files clearly indicate that they contain data obtained from a new cutting blade, while the other three (3) contain data obtained from a worn blade. Each data point in each file includes values for eight (8) parameters, which correspond to columns in the tabular dataset format. A visualization of the each of these time-series parameters in each dataset was produced for the needs of analysis in IMOCO4.E, which is presented in Figure 109.

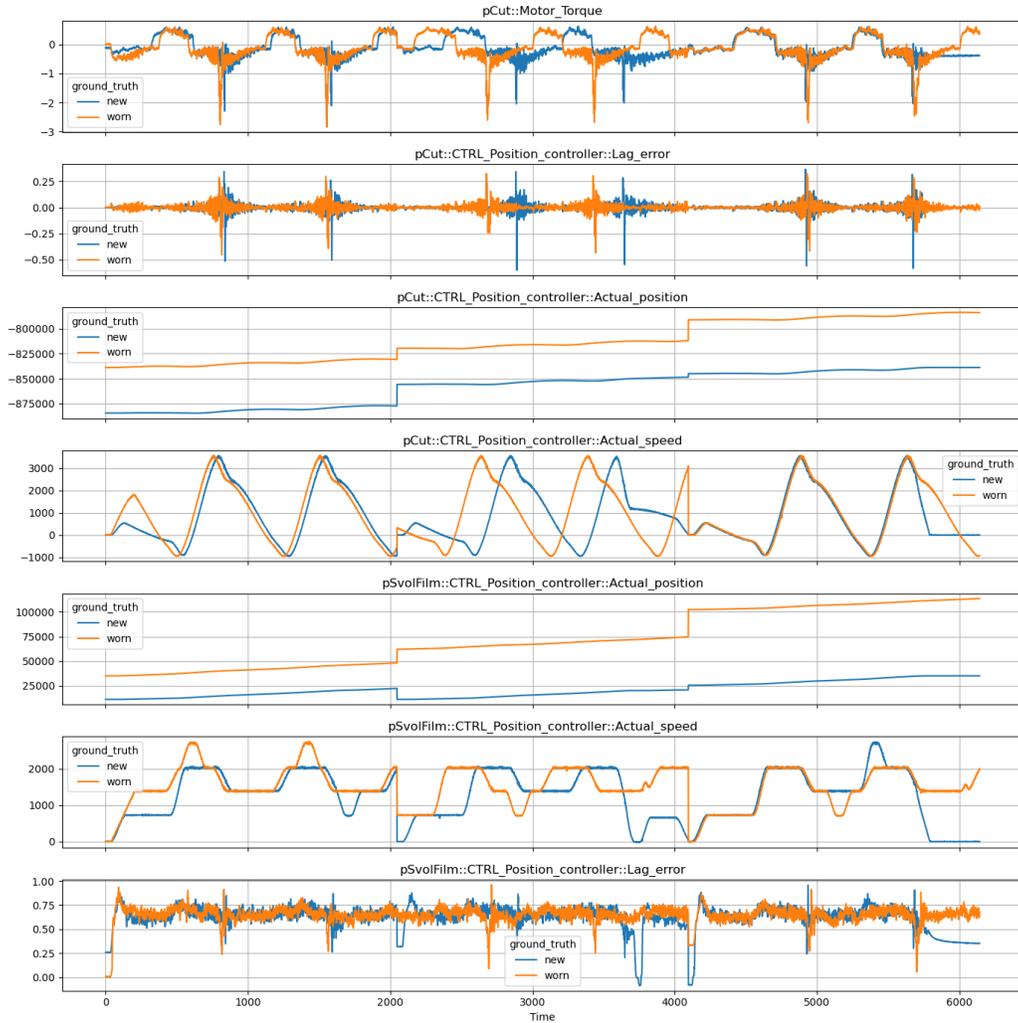


Figure 109. Vega Shrink-Wrapper new-worn blade data, six dataset, three new and three worn.

ML Model. The ML model to be implemented relies on clustering techniques. Clustering is the process of analysing data in order to group data points that have similar traits together. The following clustering unsupervised learning methods have been identified as relevant to the needs of SW-018 Solution 2 in the context of Pilot 3: *k-means clustering* [1], *Meanshift clustering* [2] and *Affinity Propagation (AP)* [3]. Classification is a byproduct of the fitted clustering model, i.e., by defining two clusters and assigning datapoints to each of them, the model acts implicitly as a binary classifier. Another direction to investigate is cluster-based feature extraction followed by classification. In this setting, the cluster assignment for each data point is obtained and used as a new feature, which is combined with the original features. The new feature set is used as input to a binary classification model, such as logistic regression, or support vector machines (SVM).

Training and Inference pipeline operation. The implementation of the component includes two pipelines, one for training and one for inference. Each pipeline consists of multiple steps which are essential for the final goal. The **training pipeline (clustering)** is initiated by reading an unlabeled dataset from CSV files. The next step refers to data **pre-processing**, where data is concatenated and transformed to suitable formats

and value ranges. This step also includes the application of Time Window Processing, that divides the incoming time series data into segments (or windows) of a specific sample size and extracts relevant features from each window. Next, the selected clustering algorithm is used to cluster the data into two groups, new and worn. The **inference pipeline (classification)** starts by reading new test data, which are subjected to the same pre-processing as in the training pipeline. Subsequently, the label of each data point is predicted using the model that has resulted from fitting the training data in the training pipeline. In the last step, the generated prediction is logged.

4.3.3 Results

Out of the 6 available datasets/snippets, 4 were used for training / clustering (2 sets from a new blade and 2 sets from a worn blade) and 2 were reserved for validation of the inference / classification prediction (1 set from a new blade and 1 set from a worn blade). The 4 csv files of the datasets for training were consolidated into a unified dataset and any reference that could reveal the origin of each data point was removed. The resulting dataset was therefore considered to be unlabeled and was used to perform clustering analysis. The purpose of the analysis was to detect distinguishable data clusters that should correspond to the original dataset distinction, which referred to either a new or a worn blade.

Using the unlabeled dataset for training described above, SW-018 was tested with three different clustering algorithms to evaluate their performance in terms of cluster formation: (i) **K-Means**, (ii) **Affinity Propagation (AP)** and (iii) **Mean Shift**.

In each case, the clustering performance was also evaluated by using 3 different options in terms of number of samples for the size of the Time Window used in the pre-processing step. The options were (a) N=50, (b) N=250 and (c) N=500.

Three measurement types were calculated to evaluate clustering performance:

- **Silhouette Value:** The Silhouette value is a measure of how well the data points within a cluster are separated from each other, and how well the clusters are separated from each other. The score ranges from -1 to 1, with higher values indicating better clustering performance.
- **Calinski-Harabasz Index:** The Calinski-Harabasz index is a measure of how well-separated the clusters are. It compares the ratio of the between-cluster dispersion and the within-cluster dispersion for each cluster. Higher values of Calinski-Harabasz index indicate better clustering performance.
- **Davies-Bouldin Index:** The Davies-Bouldin index is a measure of how well-separated the clusters are and how compact the clusters are. It compares the average distance between each cluster to the average intra-cluster distance. Lower values of Davies-Bouldin index indicate better clustering performance.

The following Table 25 presents the obtained results.

Table 25. Comparison of clustering performance evaluation measurements for 3 algorithms (K-means, AP, Mean Shift) and for 3 Time Window sizes (50, 250, 500 samples). For each measurement type, green denotes the best performance and red the worst.

Clustering algorithm	Clustering performance measurements		
	Silhouette Value	Calinski-Harabasz Index	Davies-Bouldin Index
Time Window Size: 50 samples			
K-means	0.6621	25135.6	0.548
AP	0.0867	677.155	0.0681

Mean shift	0.478	20294.9	0.7219
Time Window Size: 250 samples			
K-means	0.6679	22968.5	0.5461
AP	0.0392	245.681	0.0845
Mean shift	0.4615	17952.9	0.7329
Time Window Size: 500 samples			
K-means	0.6771	20188.2	0.5437
AP	0.0255	3601.5	0.0531
Mean shift	0.4554	15112.4	0.7467

The results indicated that the K-Means algorithm seems to perform better than the AP and Mean Shift algorithms in two of the three clustering performance evaluation measurements (Silhouette Value and Calinski-Harabasz Index). The AP algorithm seemed to outperform the other two as far as the Davies-Bouldin Index is concerned, indicating good cluster compactness and separation. Furthermore, the AP algorithm does not require the number of clusters to be specified in advance. It is influenced by the data's structure and can identify exemplars that represent clusters. However, it might generate more clusters than intended, leading to lower performance on this specific clustering task.

The Time Window size does not seem to have a significant impact on the Silhouette Value and the Davies-Bouldin Index, but it has a noticeable impact on the Calinski-Harabasz Index, as higher sample sizes seem to have a negative effect on this score.

Based on these results, the K-Means algorithm was promoted for further testing as it was found to produce well-separated, cohesive and distinct clusters and offer a better all-around option for the specific case.

Subsequent tests of SW-018 included the use of the K-Means algorithm as the clustering method in the training stage to evaluate the inference performance of the resulting model in classification prediction. For this purpose, the 2 datasets that had been reserved for the validation of inference / classification were used and the SW-018 component predicted the classification of all associated data points. The same three cases of Time Window size were also examined (50, 250 and 500 samples). In each case, four measurement types were calculated: (1) Accuracy, (2) Precision, (3) Recall, (4) F1-score. The following Table 26 presents the obtained results.

Table 26. Comparison of classification performance evaluation measurements of the model trained using the K-Means clustering algorithm in 3 cases of Time Window size (50, 250, 500 samples). For each measurement type, green denotes the best performance and red the worst.

Classification performance measurements			
Accuracy	Precision	Recall	F1-score
Time Window Size: 50 samples			
73.90%	82.80%	73.90%	72%
Time Window Size: 250 samples			
74%	83%	74%	72%
Time Window Size: 500 samples			
25%	17%	24.50%	20%

According to the results, the classification prediction performance is almost identical between the cases of 50 and 250 samples, with only a marginal advantage in the case of the 250 sample size. The performance seems to significantly deteriorate past the 250 samples, as can be seen in the case of 500 samples. In the case of 250 samples:

- the accuracy of 74% indicates that the model is correct in its predictions for 74% of the instances,
- the precision of 83% suggests that when the model predicts a positive instance, it is correct about 83% of the time.
- The recall of 73% indicates that the model captures 73% of the actual positive instances.
- The F1-score of 72% signifies a balanced performance considering both precision and recall.
- In summary, the model seems to have a decent overall performance, balancing precision and recall.

As a next test, we compared the cases of 50 samples and 250 samples of Time Window size in terms of the latency introduced by the calculations for the classification of a single data point (i.e., inference time). The tests indicated that the latency for a data point of 50 samples was **52ms** on average, while the latency for a data point of 250 samples was **57ms** on average. This latency included the time to pre-process the aggregated samples and to perform the classification prediction of the resulting data point using the trained model. It is noted that the tests were carried out using a version of the component in a docker container, deployed on a Jetson Orin Nano 8GB.

Based on the aforementioned latency results, **the combination of the K-means model with a Time Window size of 50 samples was nominated as the most efficient** solution for the overall performance of the SW-018 component in terms of **clustering performance, classification performance and inference latency**.

4.3.4 IMOCO4.E Requirements

Architecture layer requirements

Table 27. Architecture layer-specific requirements

ID	Requirement	Priority	Verify	Tasks	Compliance
R057-D2.3-L3-D4	Systems perception models should be retrainable by a non Machine Learning expert.	S	D	T3.1	Yes, process can be configured to run automatically

Priority: Severe, Medium, Low. Verify: Test, Design, Inspection.

Overall system-level requirements

Table 28. Overall (system-level) requirements

ID	Requirement	Priority	Verify	Comments	Tasks	Compliance
Interfaces and connectivity						
R001-D2.3	IMOCO4.E reference framework shall only implement interfaces and protocols with an open standard	M	I		T3.1, T4.1, T5.1, T7.1, T7.2	Yes, utilizes Linux and open source tooling

R002-D2.3	All software interfaces in the IMOCO4.E reference framework shall comply with a documented standard	M	I		T3.1, T4.1, T5.1, T7.1, T7.2	Yes
R004-D2.3-D4	Internet connection for the possibility of remotely inspecting the behaviour of perception and control modules	S	I		T3.1, T7.2	Yes
Maintainability (modularity, analysability, testability)						
R009-D2.3	The developed system is easily extendable	S	D	E.g. EtherCAT device can be added	T3.1, T4.1	Yes
R016-D2.3-U3-hw-sw	Co-existence of Information Technology (IT) and Operation Technology (OT) on the same network infrastructure will be supported.	M	I		T3.3, T3.4	Yes
Usability (operability)						
R020-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	Algorithms, AI-components and digital twin models that are intended for real-time deployment shall not adversely affect the responsiveness of the system to user requests.	M	T		T3.1, T4.1, T5.1	Yes
Tools/toolchains						
R031-D2.3	IMOCO4.E reference framework shall provide support for model-based design. A set of tools and toolchain(s) are selected and configured to support a model-driven engineering approach.	M	D		T2.3, T3.1, T4.1, T5.1, T6.1	Yes
Safety						
R047-D2.3-L2-L3-L4-B5-B6-B8-B10	Any smart control algorithms, AI-components and digital twin models shall not adversely affect the safety of the system.	M	T		T3.1, T4.1, T5.1, T6.1	Yes

Building block specific requirements

Table 29. BB8 (AI-based components) requirements

ID	Requirement	Priority	Verify	Tasks	Compliance
R182-D2.3	To support integration across all layers, BB8 shall offer industry-standard interfaces to each of the IMOCO4.E layers to exchange data	M	I	T3.1, T4.1, T5.1	Yes
R183-D2.3	Interfaces to deploy learned networks are present	M	I	T3.1, T4.1, T5.1	Yes

	Note: The main targets are BB1, BB2, BB5, and BB6				
R184-D2.3	On-site update in-the-field	S	I	T5.1	Yes
R186-D2.3	BB8 shall support real-time inference (limited and deterministic)	S	D	T3.1, T4.1, T5.1	Yes
R187-D2.3	Support and be operational in multiple Pilots/Demos/Use-cases	S	I	T3.1, T4.1, T5.1	Yes
R188-D2.3	Any user could operate (without expert knowledge)	S	D	T5.1	Yes
R189-D2.3	Minimize downtime	S	D	T5.1	Yes
R190-D2.3	BB8 shall offer AI components including one or more forms of verifiability, for example: <ul style="list-style-type: none"> - Providing a human-interpretable view of the algorithm - Providing a framework to assess reliability in a simulation/digital twin environment 	M	D	T3.1, T4.1, T5.1	Yes
R191-D2.3	Only authorised users have access to systems and data	M	I	T5.1	Yes
R192-D2.3	BB8 shall offer customizability such that non-standard tasks (i.e., tasks which are typically performed in research) can be performed. Examples include flexibility in allowed controller structures and reference/feedforward signals.	S	T	T3.1, T4.1, T5.1	Yes
R193-D2.3	BB8 shall support a computing continuum in the sense that BB8 can operate in all layers, i.e. from the instrumentation layer up to the cloud layer.	M	D	T3.1, T4.1, T5.1	Yes
R194-D2.3	EU-first tools	S	I	T3.1, T4.1, T5.1	Yes
R197-D2.3	All (generated) software for hardware targets should also run on digital twin (unmodified).	S	D	T3.1, T4.1	Yes

4.3.5 Capabilities and Limitations

Clustering and classification are two common ML techniques that have a wide range of applications. Clustering involves grouping similar data points together based on their characteristics. This can be useful in identifying patterns and trends in the data. Classification, on the other hand, involves assigning data points to predefined categories or classes based on their features or characteristics.

In the context of Industry 4.0, **classification** can be used to provide useful insight by analysing data produced by sensors on industrial equipment and predicting the category where each data sample belongs.

This technique can be used to classify data in real-time at the edge, which can be useful for monitoring and controlling industrial processes for diverse purposes, such as predictive maintenance [4], quality control [5], and optimisation of processes, resources and productivity ([6], [7], [8]). Performing classification at the edge is also aligned with current trends in IoT systems and applications, where computation is shifted from centralised computational clusters at the cloud towards distributed nodes near the edge [9] in order to reduce energy consumed for computation, conserve bandwidth (less data transferred), reduce latency in time-critical applications, while increasing security and reliability and decreasing running costs.

In order to train an effective ML classification model, large volumes of labelled datasets are required, in which each data point needs to be already assigned to one of the categories that the classifier will eventually be required to predict for new data points during system operation. However, the raw data that are obtained from sensors are typically unlabelled and data labelling is a particularly challenging task in case a human operator needs to perform the annotation. Therefore, an automation of this process is considered to be critical. To that end, **clustering techniques** can serve as an effective unsupervised learning method, being able to yield such necessary labelled datasets.

One major limitation of ML clustering and classification algorithms when dealing with time series data is their inability to explicitly capture **temporal dependencies** within the data. However, there are approaches to address this challenge by transforming the data to incorporate temporal information. One such approach is **Time Window Processing**, where the time series data is divided into segments or windows, and relevant features are extracted from each window. These features can include statistical measures such as mean, standard deviation, min or max. Another feature set that can be extracted is based on higher-order derivatives, such as the first and second derivatives of the time series. These derivatives capture the rate of change and curvature of the data, providing additional information about temporal dynamics. By incorporating these features into the process, the algorithm can better capture temporal dependencies and patterns within the time series data.

SW-018 has explored the implementation of clustering as an unsupervised learning method to train a model that can be applied for classification prediction in the field of predictive maintenance. A Time Window Processing approach has also been explored to overcome limitations of clustering associated with the inability to capture temporal dependencies in time series data. The achieved results indicate that this is a valid approach for the particular use case.

4.3.6 Customizations and Adaptations

In the context of application in Pilot 3, SW-18 was deployed on a Jetson device at the edge using a Docker container. It was configured to receive its input time series data over a TCP connection from SW-076. It was also configured to implement an Apache Kafka client so as to send its inference results (classification predictions) as messages to a Kafka broker at the fog (SW-040). The inference results were also permanently stored in an Elastic Search repository (SW-041) and were visualised using a Kibana dashboard. The interactions of SW-018 with other components in the Pilot 3 environment are illustrated in [Figure 110](#).

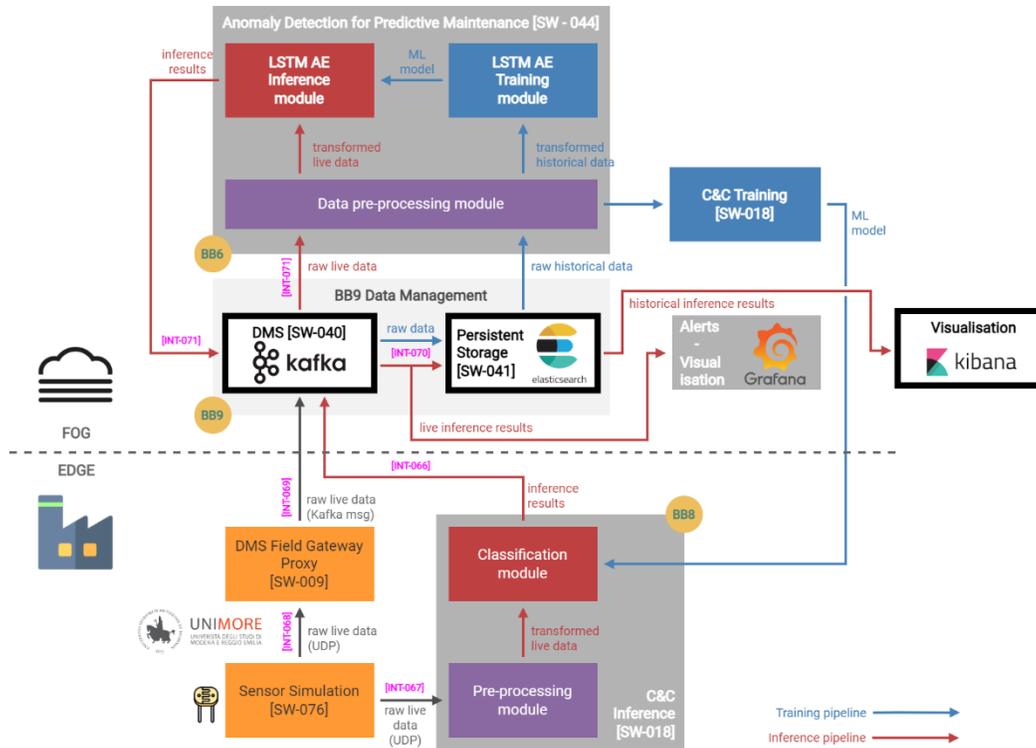


Figure 110. Implementation of SW-018 in Pilot 3 and interactions with other components.

The containerisation of SW-018 allows it to be easily **deployed and maintained on different host environments**. In future implementations, SW-018 can adopt **different protocols for receiving input data and sending its inference results**, as a modular design of the component has been adopted internally. Furthermore, depending on the use case and available time series dataset, SW-018 can be re-trained and customised by implementing other **clustering algorithms** and **Time Window sizes** that may prove to be more suitable; also in this case, the modular internal design and exposure of critical re-configurable variables allows for an easy adaptation of such aspects.

4.3.7 Methodology and Toolchains

The following tools and toolchains have been used for the development and evaluation of SW-018:

- **Jira** [10]: Used for Issue Tracking
- **Bitbucket GIT Repository** [11]: Used for source code version control.
- **Docker** [12]: Used for containerisation and deployment.
- **MLFlow** [13]: Used for ML automations, ML model management and experiment tracking
- **ZenML** [14]: Used for ML model development
- **Tensorflow** [15]: Used for ML model development
- **PyCaret** [16]: Used for ML model development and ML automations
- **Quix** [17]: Used for handling data streams in ML components

4.3.8 References

- [1] Y. Wei, X. Zhang, Y. Shi, L. Xia, S. Pan, J. Wu, M. Han and X. Zhao, “A review of data-driven approaches for prediction and classification of building energy consumption,” in *Renewable and Sustainable Energy Reviews*, <https://doi.org/10.1016/j.rser.2017.09.108>, 2018, pp. 1027-1047.
- [2] J. D. de Guia, R. S. Concepcion, H. A. Calinao, S. C. Lauguico, E. P. Dadios and R. R. P. Vicerra, “Application of Ensemble Learning with Mean Shift Clustering for Output Profile Classification and Anomaly Detection in Energy Production of Grid-Tied Photovoltaic System,” in *12th International Conference on Information Technology and Electrical Engineering (ICITEE)*, doi: 10.1109/ICITEE49829.2020.9271699, 2020.
- [3] Frey, Brendan & Dueck, Delbert. (2007). Clustering by Passing Messages Between Data Points. *Science* (New York, N.Y.). 315. 972-6. 10.1126/science.1136800
- [4] Achouch, M., Dimitrova, M., Ziane, K., Sattarpanah Karganroudi, S., Dhouib, R., Ibrahim, H., & Adda, M. (2022). On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. *Applied Sciences*, 12(16), 8081. <https://doi.org/10.3390/app12168081>
- [5] Escobar, C.A., McGovern, M.E. & Morales-Menendez, R. Quality 4.0: a review of big data challenges in manufacturing. *J Intell Manuf* 32, 2319–2334 (2021). <https://doi.org/10.1007/s10845-021-01765-4>
- [6] Manerba, Daniele & Li, Yuanyuan & Carabelli, Stefano & Tadei, Roberto & Terzo, Olivier & Fadda, Edoardo. (2020). Machine Learning and Optimization for Production Rescheduling in Industry 4.0. *International Journal of Advanced Manufacturing Technology*. 10.1007/s00170-020-05850-5.
- [7] Temich, S., Pollak, A., Kucharczyk, J., Ptasinski, W., Mężyk, A., Gąsiorek, D. (2021). Prediction of energy consumption in the Industry 4.0 platform- solutions overview. *Journal of Theoretical and Applied Mechanics*, 59(3), 455-468. <https://doi.org/10.15632/jtam-pl/140203>
- [8] Chen, C.-W., Li, C.-C., & Lin, C.-Y. (2020). Combine Clustering and Machine Learning for Enhancing the Efficiency of Energy Baseline of Chiller System. *Energies*, 13(17), 4368. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/en13174368>
- [9] Bajic, Bojana & Cosic, Ilija & Katalinic, Branko & Morača, Slobodan & Lazarevic, Milovan & Rikalovic, Aleksandar. (2019). Edge Computing vs. Cloud Computing: Challenges and Opportunities in Industry 4.0. 10.2507/30th.daaam.proceedings.120.
- [10] Atlassian Jira website [Online]. Available: <https://www.atlassian.com/software/jira> [Accessed 02 2024].
- [11] Atlassian Bitbucket website [Online]. Available: <https://bitbucket.org/product> [Accessed 02 2024].
- [12] Docker website [Online]. Available: <https://www.docker.com/> [Accessed 02 2024].
- [13] MLflow website [Online]. Available: <https://mlflow.org/> [Accessed 02 2024].
- [14] ZenML website [Online]. Available: <https://zenml.io/> [Accessed 02 2024].
- [15] TensorFlow website [Online]. Available: <https://www.tensorflow.org/> [Accessed 02 2024].
- [16] PyCaret website [Online]. Available: <https://pycaret.org/> [Accessed 02 2024].
- [17] Quix website [Online]. Available: <https://quix.io/> [Accessed 02 2024].

4.4 Energy-efficient and fast vision-processing system, based on AI and approximate computing techniques (incl. x-in- the-loop framework for development, testing and debugging (TUE))

4.4.1 Technology overview

This work introduces an approach for closed-loop motion control in semiconductor manufacturing (Pilot 2), leveraging Deep Neural Networks (DNNs) for improved perception. By combining DNN-based perception with traditional encoder-based position measurements, the proposed method improves the ability to detect and compensate for misalignments during high-speed, micro-scale positioning tasks. Key contributions include the utilization of DNN-based perception for vision-in-the-loop (VIL) systems in high-precision motion control and the introduction of a fusion mechanism to combine vision feedback with encoder data. Additionally, a software-in-the-loop (SIL) framework utilising digital twin, is employed to validate the proposed method's efficiency and effectiveness in eliminating misalignments.

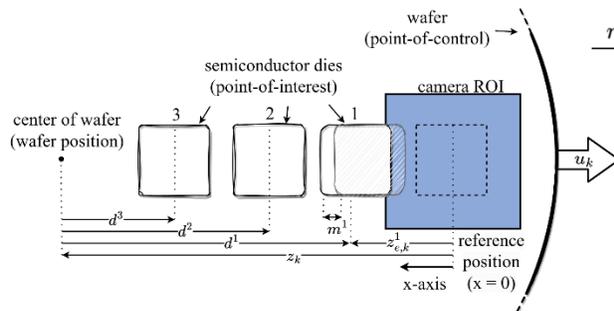


Figure 111. Set of semiconductor dies located on a silicon wafer. The encoder reads the position of the wafer and the camera captures an image of the (partial) die(s) in the ROI. The dies are positioned at the reference by moving the wafer.

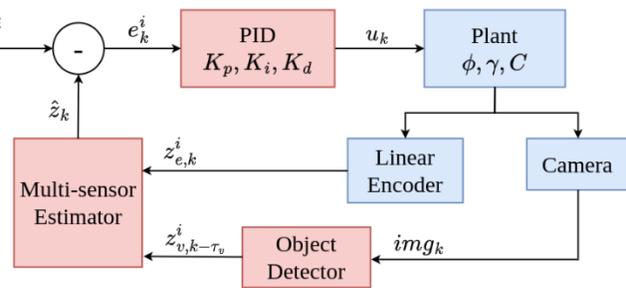


Figure 112. Block diagram of the system, with the linear encoder and camera as sensors, multi-sensor estimator, and PID controller.

4.4.2 Implementation aspects

The system under consideration is composed of a linear motor which is used to control the position of the platform which has the silicon wafer with thousands of dies systematically placed onto it. This pattern is defined by the coordinates representing the ideal position of the dies d^i with respect to the center of the wafer. For example, in Figure 111 since dies 1, 2, and 3 are on the right side of the centre of the wafer d^1 , d^2 and d^3 are negative constants. In practice, as the dies are distributed uniformly on the wafer, these values can be defined by just two parameters, the die size and the centre-to-centre distance between the neighbouring dies. The primary goal of the motion control system is to accurately position these semiconductor dies one after the other at the *reference position* where they can be picked up. To achieve this task the output of linear motors is sensed by encoders, which accurately read the movements of the motor and therefore the wafer (point-of-control). The positioning task is achieved by defining a set of control references with the help of the ideal positions of the semiconductor dies with respect to the wafer i.e. d^i . For example, to maintain the throughput of more than 70,000 units per hour a die is picked up every 50 ms. During pickup, the neighbouring dies get misaligned from their initial positions. This misalignment is denoted by m_k^i with respect to the ideal position of the die. To sense this misalignment vision feedback

is used by capturing the images of the die in the camera's Region of Interest (ROI) and using an object detector to get the true position of the die and thus the misalignment of the die from its ideal position. The Figure 112 shows the block diagram of the various components of the system. Along with the linear encoders, A camera is added to the feedback loop for the controller, which captures the images and sends it to the Faster R-CNN object detector.

Table 30. Configuration of the multi-die scenario.

Die (<i>i</i>)	Initial die Position ($x_{e,0}^i$)	Control Reference (r^i)	<i>m1</i>	<i>m2</i>
1	300	900	-20	20
2	600	600	-25	25
3	900	300	-30	30

4.4.3 Results

To analyse the behaviour of the multi-die positioning system we use the above-mentioned SIL framework and the die configuration with the two misalignment profiles *m1* and *m2* described in Table 30. Along with the two misalignment profiles, an ideal scenario is considered with no misalignment in die positions.

To study the performance of the motion control system with vision feedback we consider two key metrics Steady State Value (SSV) i.e. the position of the dies at every 50 ms mark when they must be picked up and Mean Absolute Error (MAE) of the estimator during every 50 ms run.

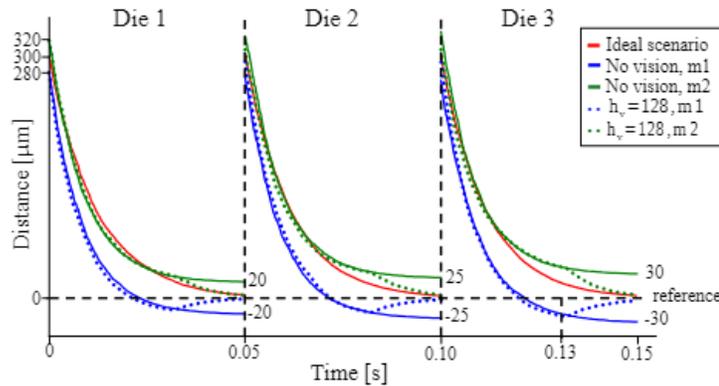


Figure 113. Coordinates of dies for motion control with and without vision feedback for misalignment profiles *m1* and *m2* compared with the ideal scenario.

Figure 113 shows the controller performance for both misalignment profiles and the ideal case (no misalignment, shown in red). In case of ideal scenario, the controller can move the die to the reference for the pickup, though it fails to do so without any vision feedback (solid blue and green lines). With vision feedback it is possible to detect the misalignment is the die position, and controller corrects it before the periodic pick-up needs to happen i.e. within 50 ms.

Table 31 shows the performance (SSV and MAE) of the closed-loop system with different vision sampling rates. In the cases with an SSV in the range of $[-20 \mu\text{m}, 20 \mu\text{m}]$ the dies are within the acceptable misalignment range, and can be picked up. So in case of *m2*, for die 1 with a misalignment of $20 \mu\text{m}$ and the controller with no vision feedback, it reaches the reference with an SSV of $20.30 \mu\text{m}$, which is out of the acceptable range of SSV.

Table 31 Performance of the closed loop system

h_v	Die	m1			m2		
		SSV μm	MAE μm	Pick-up	SSV μm	MAE μm	Pick-up
Without vision feedback	1	-19.61	20.0	Yes	20.30	20.0	No
	2	-24.70	25.0	No	25.47	25.0	No
	3	-29.34	30.0	No	30.32	30.0	No
8	1	0.31	2.61	Yes	0.36	3.03	Yes
	2	0.25	2.51	Yes	0.13	3.00	Yes
	3	0.23	2.89	Yes	0.18	4.14	Yes
64	1	0.53	6.68	Yes	1.02	6.90	Yes
	2	-0.29	8.2	Yes	1.02	8.47	Yes
	3	-0.20	9.91	Yes	0.99	9.93	Yes
128	1	-1.29	13.16	Yes	3.59	12.89	Yes
	2	-1.87	16.27	Yes	3.80	16.09	Yes
	3	-2.97	19.36	Yes	4.08	19.29	Yes

4.4.4 IMOCO4.E Requirements

Architecture layer requirements

Table 32 Architecture layer-specific requirements

ID	Requirement	Priority	Verify	Tasks	Compliance
R049-D2.3-L1	The interface between Layer 1 and Layer 2 shall be time-critical and support sample rates of at least 20 kHz	S	T	T3.1, T4.1, T6.1	Not applicable
R057-D2.3-L3-D4	Systems perception models should be retrainable by a non Machine Learning expert.	S	D	T3.1	Yes
R060-D2.3-L1-hw-DAT	Sensors monitoring the condition of an asset should provide real-time measurements without disruptions	S	T	T3.1	Yes
R062-D2.3-L1	Hardware components must be interchangeable; boundary constraints must be set: volume, temperature, power supply, etc.	S	I	T3.1	Yes
R063-D2.3-L2	The optimised Neural Networks must be able to run on the available (brownfield) hardware platforms.	M	D	T3.3, T3.4	Yes
R065-D2.3-L1-hw	The hardware needed to compensate for EMI noise needs to be more compact, lighter, flexible, cheaper than conventional shielding and filtering	S	I	T3.1, T6.5	Not applicable

Priority: Severe, Medium, Low. Verify: Test, Design, Inspection.

Overall system-level requirements

Table 33 Overall (system-level) requirements

ID	Requirement	Priority	Verify	Tasks	Compliance
R001-D2.3	IMOCO4.E reference framework shall only implement interfaces and protocols with an open standard	M	I	T3.1, T4.1, T5.1, T7.1, T7.2	Yes
R002-D2.3	All software interfaces in the IMOCO4.E reference framework shall comply with a documented standard	M	I	T3.1, T4.1, T5.1, T7.1, T7.2	Yes
R004-D2.3-D4	Internet connection for the possibility of remotely inspecting the behaviour of perception and control modules	S	I	T3.1, T7.2	Yes
R005-D2.3-D4	Screen and input hardware for inspection and correction of perception and control modules	S	I	T3.1, T7.2	Yes
R006-D2.3-L1-hw-sw	The IMOCO4.E reference framework shall support EtherCAT connectivity	S	I	T3.1, T4.1, T5.1	Yes
R009-D2.3	The developed system is easily extendable	S	D	T3.1, T4.1	Yes
R016-D2.3-U3-hw-sw	Co-existence of Information Technology (IT) and Operation Technology (OT) on the same network infrastructure will be supported.	M	I	T3.3, T3.4	Not applicable
R020-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	Algorithms, AI-components and digital twin models that are intended for real-time deployment shall not adversely affect the responsiveness of the system to user requests.	M	T	T3.1, T4.1, T5.1	Yes
R031-D2.3	IMOCO4.E reference framework shall provide support for model-based design. A set of tools and toolchain(s) are selected and configured to support a model-driven engineering approach.	M	D	T2.3, T3.1, T4.1, T5.1, T6.1	Yes
R047-D2.3-L2-L3-L4-B5-B6-B8-B10	Any smart control algorithms, AI-components and digital twin models shall not adversely affect the safety of the system.	M	T	T3.1, T4.1, T5.1, T6.1	Yes

Building block specific requirements

Table 34 BB2 (high-speed vision-in-the-loop) requirements

ID	Requirement	Priority	Verify	Tasks	Compliance
R088-D2.3-B2-P2	BB2 shall use standardised interfaces to the various involved layers (Layer 1 and Layer 2 specifically)	M	I	T3.1, T4.1	Yes
R089-D2.3-B2-P2	Must have physical mounting plate and volume claim	M	I	T3.1	Not applicable
R090-D2.3-B2-P2	BB2 must offer interfacing to industrial cameras via an industry-standard interface	M	D	T3.1	Yes
R091-D2.3-B2-P2	BB2 can be connected to Matlab/Simulink	C	A	T3.1, T4.1	Yes
R092-D2.3-B2-P2	BB2 shall be modular with testable interfaces	S	I	T3.1	Yes
R093-D2.3-B2-P2	The low-cost BB2 implementation shall offer a vision position update rate of at least 75 Hz	M	D	T3.1	Yes
R094-D2.3-B2-D4	Object detection and pose estimation model processes > 5 frames per second	S	D	T3.1	Yes
R095-D2.3-B2-P2	The high performance BB2 implementation must have frame rate of 5 kHz	M	I	T3.1	Yes
R096-D2.3-B2-P2	Must have Latency <300 µsec	M	I	T3.1	Not applicable
R097-D2.3-B2-P2	Must have Image frame size > 0.25 Mpixel	M	I	T3.1	Not applicable
R098-D2.3-B2-P2	BB2 shall have a configuration interface to modify all (pre-defined) configuration parameters without requiring firmware changes	M	D	T3.1	Yes
R099-D2.3-B2-P2	BB2 must be able to operate disconnected from connections to the internet	M	I	T3.1	Yes
R100-D2.3-B2-P2	BB2 shall offer customizability such that non-standard algorithms can be implemented	M	D	T3.1	Yes
R101-D2.3-B2-P2	BB2 shall have a cost target of <1000€, excluding camera, optics and motion drives (for lower frame rates)	M	I	T3.1, T7.1	Yes
R102-D2.3-B2-P2	BB2 shall have a cost target of <25000€, including camera; excl optics and motion drives. (for higher frame rates)	C	I	T3.1, T7.1	Yes
R103-D2.3-B2-P2	BB2 shall offer a scalable processing platform, to enable higher update rates	M	I	T3.1, T7.1	Yes

	or more complicated image processing pipelines if required				
R104-D2.3-B2-P2	BB2 would preferably be based on open software tooling	M	I	T3.1	Yes
R105-D2.3-B2-P2	Must be safe for humans, products, machine/system and environment	M	D	T3.1, T4.1, T5.1	Yes

4.4.5 Capabilities and Limitations

This work proposes the use of a deep neural network for object detection tasks in a closed-loop positioning application, which improves the positioning accuracy, and adds robustness to the detection task. The proposed methodology developed in the digital twin framework was also demonstrated on th physical system during the IMOCO4.E M24 meeting.

Training the model requires a lot of data, which is sometimes an issue in these domains. Using the digital twin helps in collect initial set of data. The model inference has a significant timing cost, but new computation paradigms are being developed to mitigate this limitation.

4.4.6 Customizations and Adaptations

The framework can be customized to suit specific application requirements by modifying the DNN architecture, training data, and controller algorithms. The framework is designed to be easily adaptable, allowing for independent development and testing of various components like the DNN and controller. All the existing components have network endpoints, so extending the framework to add new components is a possibility.

4.4.7 Methodology and Toolchains

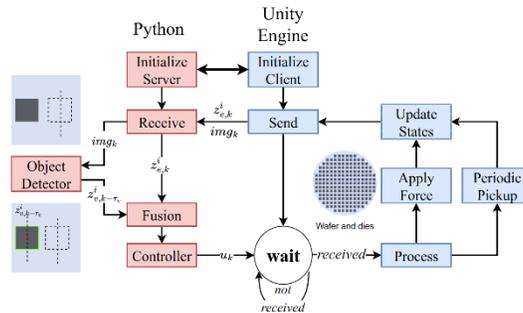


Figure 114. Synchronous simulation framework for vision-in-the-loop system

To validate the proposed method for correcting misalignment in the semiconductor die packaging, we create a model of discrete-time dynamics of the wafer for multi-die positioning scenarios within the Unity engine. The object detector, controller, and estimator are developed in Python. DNN training and inference was done with the help of PyTorch library. The Python script acts as a server and the Unity model as a client, both operating in synchronous simulation for software-in-the-loop (SIL) verification as shown in Figure 114. The base sampling period $h_{encoder}$ is 0.125ms

The framework allows image captures at a rate of $1/h_{encoder}$, which is not possible to achieve in the real system but allows us to gain important insights on controller performance by varying vision feedback rates.

4.5 Autonomous smart motion sensor based on a high speed camera (TNO)

4.5.1 Technology overview

The work by TNO contributes to Scientific and Technological development Objectives as stated in the DOA: Ultra-High speed vision with AI-based scene recognition (ST2). Analysis of requirements indicates that the bandwidth requirements are a superposition of both motion trajectory setpoint requirements and suppression of disturbances on the plant. The most challenging requirements for visual servoing are in Pilot 2. For realizing positioning accuracies well below 1micrometer, disturbances as introduced by vibrating coolers and other peripheral machinery need to be suppressed. Analysis shows error suppression until at least 150 Hz is required. In order to achieve this performance figure, as well as to comply with stability criteria, some 300 μ sec of signal latency (defined as: camera exposure to the exercise of force by the actuator) is allowed (see requirement R096-D2.3-B2- P2).

TNO is contributing to the WP3 objective on development of a visual servoing system based on a highspeed camera and a high-performing data processing pipeline. The result from this work is an embedded computing platform that combines time determinism (microseconds), High-Performance Computing, open IO architecture suited high data rates.

Image processing with latencies down to 50 μ sec latencies strains any processing unit. CPUs are more flexible, but fall short by orders of magnitude, whereas GPUs excel in massive parallel number crunching and peak as high as 7 TFLOPS due to 5000 free programmable cores in combination with HBM2 memory reaching unrivalled throughputs of 900 GByte/second. The high-speed vision control system by TNO utilizes a heterogeneous computing platform; mixing both GPUs and CPUs.

In order to reduce CAPEX, the IMOCO4E-systems requires the utilization of open standards and COTS components where possible. System design is based on the PC industry's adoption of the PCI-Express standard. PCI-Express allows for rapid interaction between the 'host' (=CPU + main-memory + IO control) and a wide range of additional devices (GPUs, frame grabbers, analogue interfaces, networks controllers etc.).

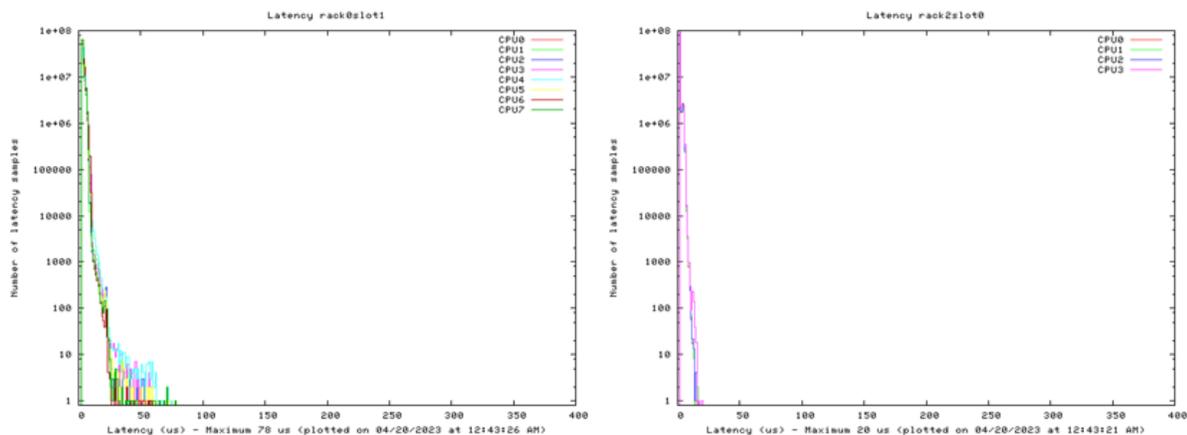


Figure 115. Interrupt response latency histograms showing standard Linux (left) and improved (real time) kernel + adaptations by TNO (right)

Multitasking computers struggle as per definition with deterministic timing. Linux is no exception in this regard. The high-speed vision system by TNO greatly enhances deterministic behavior by utilizing

- Double Direct Memory Access (DMA) to contiguously stream camera data over the PCIe-Bus into GPU-memory, without involvement of the host.
- A real-time Linux kernel on the host, complemented with a tuned resource locking strategy

Towards application development and demonstration in Pilot2, performance development of the system has been tracked.

The system's overall latency could be improved by adding a customized field bus and Linux device driver. See Figure 115 Latency variation has been reduced from 25% to less than 5%.

4.5.2 Implementation aspects

Implementation of the High-speed vision in the loop has been focused towards commercial of the shelf components as to improve the price-performance ratio.

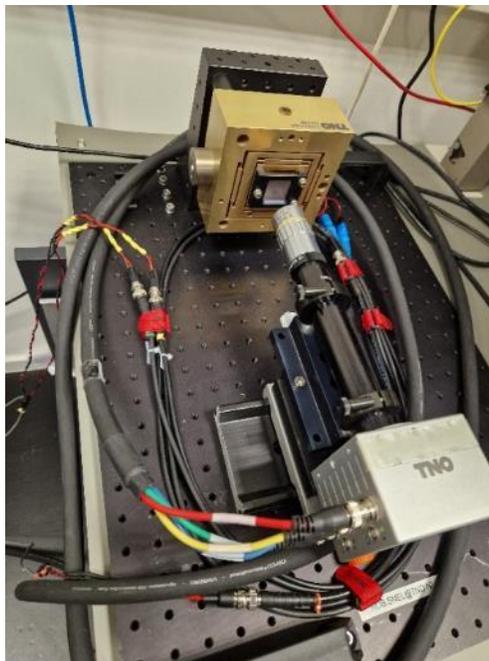


Figure 116. XY motion platform and CoaXPress high speed camera.

Figure 116 shows the motion platform with a cantilever guiding mechanism actuated by 2 voice-coil (Bei Kimco). The Optronis OPTRONIS CP80-3-M-540 camera is connected via CoaXPress to the host computer See Figure 117.



Figure 117. Computer rack with host PC, PCI-extender, Main switch units and auxiliary electronics box

The host computer is a rack-mounted Asus Maximus Hero Z390 equipped with an Intel I7-8700K processor and a PCI-extender to host the framegrabber, GPU and PWM-output interface. The GPU and framegrabber have been selected to cooperate with each other in a PCIe peer-to-peer communication mode and map each other's memory via the Base Address Register of the host computer.

4.5.3 Results

The control performance of demonstration system has been optimized for reliability, robustness and bandwidth via loop shaping with SHAPE-IT tool. Performance as realized so far has exceeded disturbance rejection up to 120Hz @ 5KHz frame rate as limit for closed loop disturbance rejection. In the figure below a scree shot is given of the performance when scaled at 1KHz frame rate.

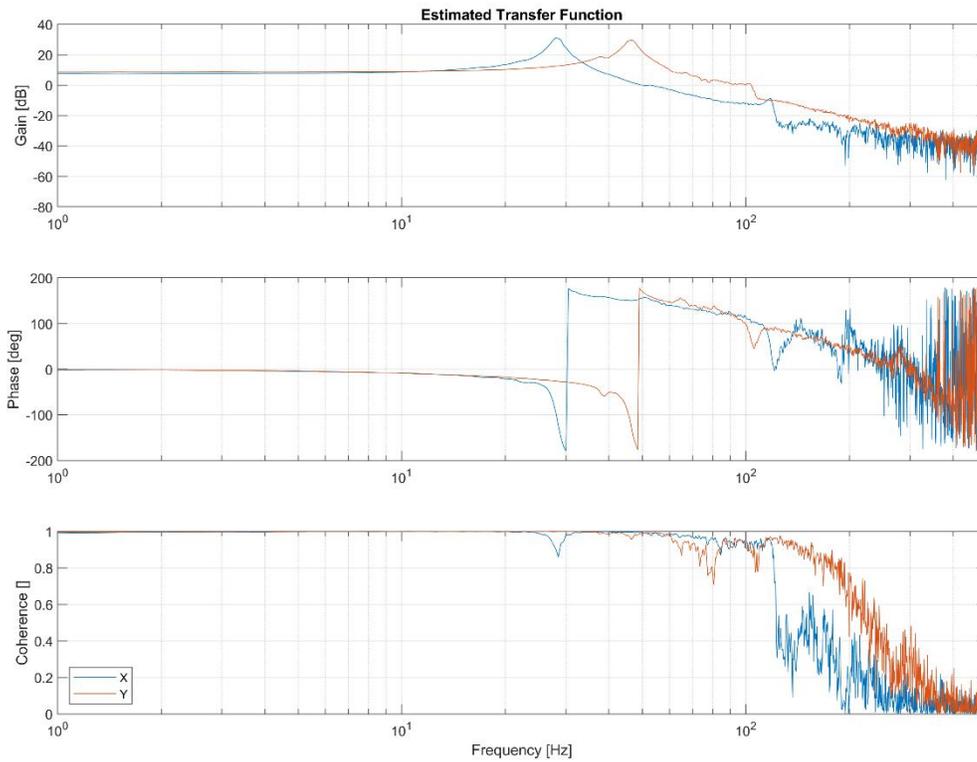


Figure 118. Bode diagram of tuned motion system

4.5.4 IMOCO4.E Requirements

Specifications have been verified against their respective requirements as specified is D3.2.

Table 35. Architecture layer-specific requirements

ID	Requirement	Priority	Verify	Tasks	Compliance
R049-D2.3-L1	The interface between Layer 1 and Layer 2 shall be time-critical and support sample rates of at least 20 kHz	S	T	T3.1, T4.1, T6.1	Yes, software API supports this
R057-D2.3-L3-D4	Systems perception models should be retrainable by a non Machine Learning expert.	S	D	T3.1	Yes
R060-D2.3-L1-hw-DAT	Sensors monitoring the condition of an asset should provide real-time measurements without disruptions	S	T	T3.1	Not applicable
R062-D2.3-L1	Hardware components must be interchangeable; boundary constraints must be set: volume, temperature, power supply, etc.	S	I	T3.1	Yes, utilizing COTS components as much as possible. Software API.

R063-D2.3-L2	The optimised Neural Networks must be able to run on the available (brownfield) hardware platforms.	M	D	T3.3, T3.4	Yes, BB2-platform supports heterogeneous GPGPU for powerful NN
R065-D2.3-L1-hw	The hardware needed to compensate for EMI noise needs to be more compact, lighter, flexible, cheaper than conventional shielding and filtering	S	I	T3.1, T6.5	Not applicable

Priority: Severe, Medium, Low. Verify: Test, Design, Inspection.

Overall system-level requirements

Table 36. Overall (system-level) requirements

ID	Requirement	Priority	Verify	Comments	Tasks	Compliance
Interfaces and connectivity						
R001-D2.3	IMOCO4.E reference framework shall only implement interfaces and protocols with an open standard	M	I		T3.1, T4.1, T5.1, T7.1, T7.2	Yes, utilizes Linux and open source tooling where possible
R002-D2.3	All software interfaces in the IMOCO4.E reference framework shall comply with a documented standard	M	I		T3.1, T4.1, T5.1, T7.1, T7.2	Yes
R004-D2.3-D4	Internet connection for the possibility of remotely inspecting the behaviour of perception and control modules	S	I		T3.1, T7.2	Yes, socket interface
R005-D2.3-D4	Screen and input hardware for inspection and correction of perception and control modules	S	I		T3.1, T7.2	Yes
R006-D2.3-L1-hw-sw	The IMOCO4.E reference framework shall support EtherCAT connectivity	S	I		T3.1, T4.1, T5.1	Yes
Maintainability (modularity, analysability, testability)						
R009-D2.3	The developed system is easily extendable	S	D	E.g. EtherCAT device can be added	T3.1, T4.1	Yes, modular design
R016-D2.3-	Co-existence of Information Technology (IT) and Operation Technology (OT) on the	M	I		T3.3, T3.4	Yes

U3-hw-sw	same network infrastructure will be supported.					
Usability (operability)						
R020-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	Algorithms, AI-components and digital twin models that are intended for real-time deployment shall not adversely affect the responsiveness of the system to user requests.	M	T		T3.1, T4.1, T5.1	Yes, multi core processing and strict separation of functionality
Tools/toolchains						
R031-D2.3	IMOCO4.E reference framework shall provide support for model-based design. A set of tools and toolchain(s) are selected and configured to support a model-driven engineering approach.	M	D		T2.3, T3.1, T4.1, T5.1, T6.1	Yes
Safety						
R047-D2.3-L2-L3-L4-B5-B6-B8-B10	Any smart control algorithms, AI-components and digital twin models shall not adversely affect the safety of the system.	M	T		T3.1, T4.1, T5.1, T6.1	Yes

Building block specific requirements

Table 37. BB2 (high-speed vision-in-the-loop) requirements.

ID	Requirement	Priority	Verify	Tasks	Compliance
R088-D2.3-B2-P2	BB2 shall use standardised interfaces to the various involved layers (Layer 1 and Layer 2 specifically)	M	I	T3.1, T4.1	Yes
R089-D2.3-B2-P2	Must have physical mounting plate and volume claim	M	I	T3.1	Yes
R090-D2.3-B2-P2	BB2 must offer interfacing to industrial cameras via an industry-standard interface	M	D	T3.1	Yes, utilizes CoaXPres
R091-D2.3-B2-P2	BB2 can be connected to Matlab/Simulink	C	A	T3.1, T4.1	Yes

R092-D2.3-B2-P2	BB2 shall be modular with testable interfaces	S	I	T3.1	Yes
R093-D2.3-B2-P2	The low-cost BB2 implementation shall offer a vision position update rate of at least 75 Hz	M	D	T3.1	Yes, utilizes low cost hardware
R094-D2.3-B2-D4	Object detection and pose estimation model processes > 5 frames per second	S	D	T3.1	Yes
R095-D2.3-B2-P2	The high performance BB2 implementation must have frame rate of > 5 kHz	M	I	T3.1	Yes, 5KHz with 512x512 pixels
R096-D2.3-B2-P2	Must have Latency <300 μsec	M	I	T3.1	Yes
R097-D2.3-B2-P2	Must have Image frame size > 0.25 Mpixel	M	I	T3.1	Yes, 512x512 pixels
R098-D2.3-B2-P2	BB2 shall have a configuration interface to modify all (pre-defined) configuration parameters without requiring firmware changes	M	D	T3.1	Yes, uses configuration files
R099-D2.3-B2-P2	BB2 must be able to operate disconnected from connections to the internet	M	I	T3.1	Yes, standalone is possible
R100-D2.3-B2-P2	BB2 shall offer customizability such that non-standard algorithms can be implemented	M	D	T3.1	Yes, application interface allows for modular algorithms
R101-D2.3-B2-P2	BB2 shall have a cost target of <1000€, excluding camera, optics and motion drives (for lower frame rates)	M	I	T3.1, T7.1	Yes
R102-D2.3-B2-P2	BB2 shall have a cost target of <25000€, including camera; excl optics and motion drives. (for higher frame rates)	C	I	T3.1, T7.1	Yes

R103-D2.3-B2-P2	BB2 shall offer a scalable processing platform, to enable higher update rates or more complicated image processing pipelines if required	M	I	T3.1, T7.1	Yes, modular concept utilizing COTS components for heterogeneous computing
R104-D2.3-B2-P2	BB2 would preferably be based on open software tooling	M	I	T3.1	Yes, Linux OS and tooling
R105-D2.3-B2-P2	Must be safe for humans, products, machine/system and environment	M	D	T3.1, T4.1, T5.1	Yes

4.5.5 Capabilities & limitations

The High Performance Control system is highly scalable to serve the most calculation demanding applications. The architecture is based on PCIe interface boards and is therefore less suited for low battery powered applications. The strength of this set up lies in the fact that it combines High Performance Computing, Strict μ second real time behavior with a highly customizable IO-platform. The upside is that the system is highly scalable (i.e. with or without a GPU, depending on performance requirements) But an x86-host will likely dissipate 15Watt as a minimum and high performance setups with a GPU can easily draw in excess of 500Watts continuously.

4.5.6 Customization options and adaptations

Customization options are widespread. Platform is based on the much common X86-64 technology with it's wide range of processors from simple single core for low-dissipation embedded application to performance hungry 96-core workstations and servers. Adaptability to physical interfaces via PCIe interface boards. For multi-many-core processing acceleration there are various fast GPGPUs by AMD and NVIDIA, ranging from cost effective low-end cards to the high-end workstation accelerators.

Communication to external physical processes is also organized via PCIe. The demonstration system utilizes a CoaXPress framegrabber that directly interfaces to the GPU via shared memory as mapped via the Base Address Register (BAR).

Since not all physical processes can be served with standard hardware there is the option of communicating with an FPGA on a customizable PCIe carrier board with FPGA Mezzanine interfaces.

4.5.7 Methodology & toolchains

The toolchain for generation real time code consists of customizable C-code, to be compiled with the open source GCC compiler that is supplied with the applicable Linux distribution. Some proprietary extensions are required to utilize the GPU and framegrabber. These extensions are supplied free of charge by the various hardware suppliers upon purchase of their hardware.

4.6 Deep neural networks based system for deployment on robotic systems to enable recognition of the scene and understanding of complex situations in typical warehouses or production sites – load carrier pose estimation based on synthetic data. (IML, STILL). Optimization of these neural networks for FPGAs and MCUs with a focus on low-power (HS)

AI-based robotics perception solution for understanding complex situations in typical warehouses or production sites consists of two different components and sub-solutions. As these sub-solutions differ a lot according to their requirements, used technologies and implementation, they are described in separate sections. This section is dedicated to the load carrier pose estimation based on synthetic data, while the following section describes radar signal simulation for AI-driven scene recognition and complexity understanding.

4.6.1 Technology overview

To automate the process of picking up load carriers, such as pallets, it is necessary to determine the pose of the object to be picked up. Existing solutions often rely on fixed positions of the load carriers, which are marked with QR codes or similar markers. To avoid this additional infrastructure and limited flexibility, our solution aims to detect load carriers at any location within a brownfield environment based on RGB camera data to enable the vehicle to pick up the load carriers using appropriate control algorithms.

The solution uses machine learning based Pose Estimation models, but collecting real-world data is challenging as a huge number of images covering diverse scenarios and manually annotating the data are needed. Further, load carriers can have different appearances, carry various types of loads, and be perceived under different environmental conditions. Moreover, vision models trained on real-world data for one application often need to be retrained for new environments. To address these challenges, we are leveraging state-of-the-art simulation tools to generate synthetic and annotated images of load carriers. These tools make use of the latest rendering techniques and the computational power of GPUs to generate photorealistic images that closely resemble real camera data. By using simulated data, we overcome the limitations of real-world data collection and train a robust model for load carrier detection.

The solution consists of two main blocks as given in Table 38. Both software components are related to the development of AI solutions, therefore they are included in BB8. The *Pose Estimation Model for Load Carriers* considers requirements to work with camera data and estimates the pose within a reasonable amount of time. According to this, this software component is included in BB2, even if the needed processing rate is much smaller in comparison to other components in BB2.

Table 38. Software components of solution *Error! Reference source not found.* and their link to the IMOCO4.E architecture.

SW ID	Building Block(s)	Layer	Description
SW-021	8	4	Synthetic Dataset Generator
SW-022	2, 8	4	Pose Estimation Model for Load Carriers

An overview of the technology chain is given in Figure 119.

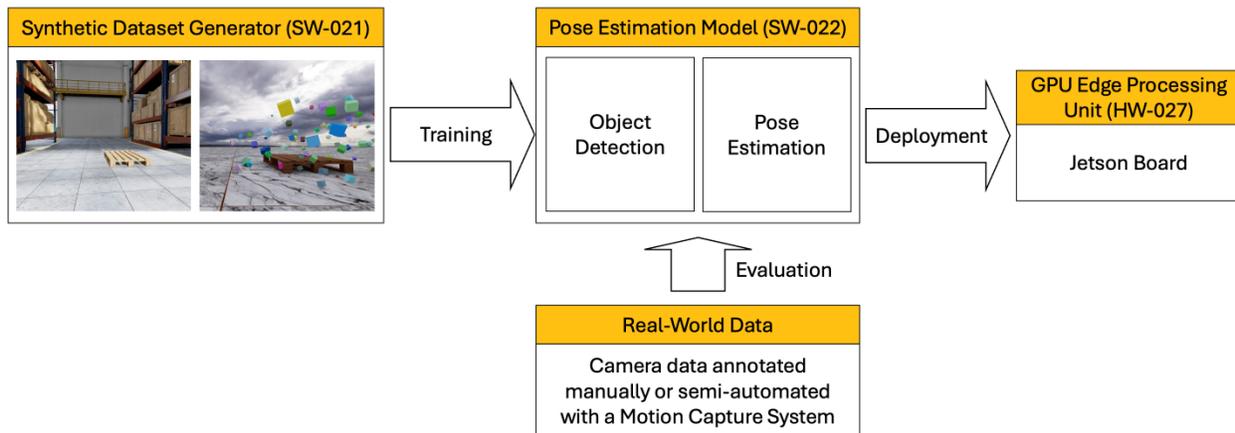


Figure 119. Overview of the technology chain of the used software components.

The **Synthetic Dataset Generator (SW-021)** is used to generate trainings data of simulated pallets. Therefore, a model of the pallet is randomly placed in simulation. Multiple Domain Randomisation techniques are used to create a trainings dataset which consists of a huge variety of image data, containing different lightning conditions, different backgrounds and much more. Since the simulation used to generate the synthetic data allows to determine the pose of the pallet in a rendered image, the Synthetic Dataset Generator module generates the images with annotation and camera information. Depending in the AI model which needs to be trained, the trainings data can be generated with different annotation formats.

The synthetic training images are used to train the **Pose Estimation Models for load carriers (SW-022)**, which consists of two components. First, an Object Detection approach is used to detect all the pallets given in an image. Afterwards, the image parts of the detected pallets are cropped and given to a Pose Estimation Model. This approach turned out to be the best choice during the project. Earlier approaches consisted of a Pose Estimation model only, but the combination of models turned out to be more accurate, robust, and performant. More details are given in the next section. This combination increases the training effort since both models are trained separately from each other and with different annotated trainings data. The Synthetic Dataset Generator module saves even more time with this approach since it generates both trainings dataset by just changing a configuration. The quality of the trained model is tested by real-world data which can be manually labelled. Further, some activities were done to annotate camera data semi-automated by tracking the camera and the object of interest via a Motion Capture System. Finally, the Pose Estimation Model is deployed to the GPU Edge processing unit (HW-027), a Jetson board on the demonstrator vehicle iGo neo (HW-026). Within the demonstrator a RGB camera (HW-028) is used to estimate the pose of real-world pallets on the image data.

4.6.2 Implementation aspects

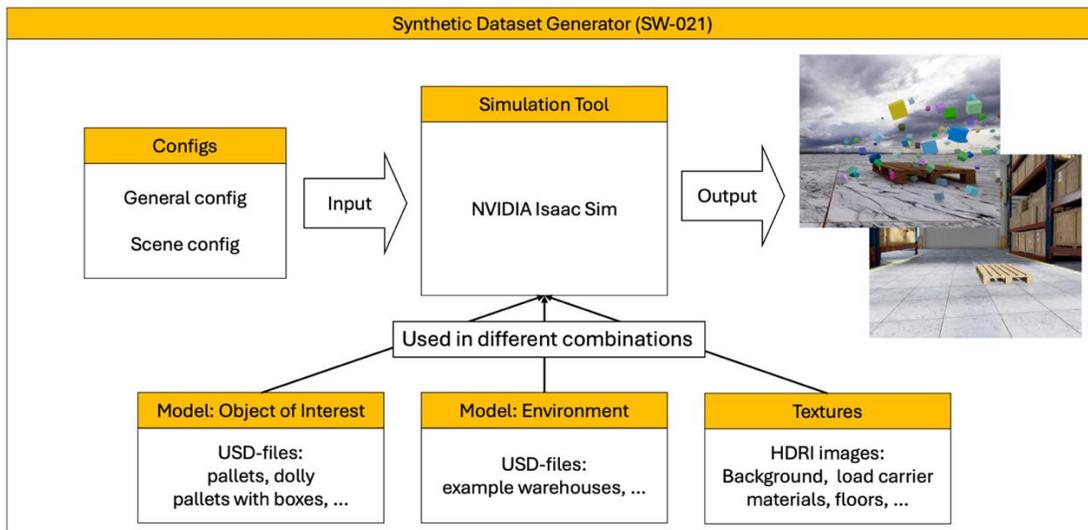


Figure 120. Implementation principle of the Synthetic Dataset Generator (SW-021).

The principles of the implementation of the **Synthetic Dataset Generator (SW-021)** are shown in Figure 120. As input different config files are used: one for the general config and multiple scene config files. The general config defines aspects which should be consistent for all the training data, such as the resolution of the camera. The number of scene config files depends on the user needs, it is possible to generate all the synthetic data with one file or a bunch of files can be used. Within the config the synthetic data which needed to be generated are characterised by defining the randomisation techniques which should be used (distance between load carrier and camera, range for the pose variation, path to the textures which should be used for randomisation). Further, the number of images to be generated is given. By using several scene config files, the user gains more control on which trainings data are generated and how they are distributed within the training dataset. The configs are considered from the software module and via Python APIs the simulation tool NVIDIA Isaac Sim is used to generate the synthetic images. They are the output of the Synthetic Dataset Generator. Within the generation process, different models highlighted in Figure 120 are combined to create different scenes in Isaac Sim. First, a simulation model of the object of interest is needed, therefore a pallet model was generated. Second, environment simulation models built from standard assets of Isaac Sim are used. Third, a huge variety of HDRI images for textures and the background is needed. As starting point, open-source available images are used, and further textures are generated with Stable Diffusion models. The more models and textures are provided to the Synthetic Dataset Generator, the greater the range of variety within the synthetic images.

The **Pose Estimation Models for load carriers (SW-022)**, shown in Figure 121, uses a YOLOv5 model for the Object Detection, remaining on a third-party implementation for training the model. It is trained independent of the overall Pose Estimation approach based on the generated synthetic data and allows to detect load carriers which are then marked with 2D bounding boxes. For the Pose Estimation DOPE is selected as model, again a third-party repository is used for training. The model was trained once with the standard synthetic data, once with the cropped outputs of the Object Detection applied on synthetic data. The second option turned out to deliver better results. After training the models are combined in a ROS2 node for inference. The RGB camera stream is delivered under the ROS topic /image and processed by the Object Detection model which detects all the load carriers and marks them with a 2D bounding box (see

the red box in Figure 121). The bounding box is then extended and transformed to a common image input size, the final cropped images act as input for the Pose Estimation model (see the blue box in Figure 121). Since the cropped image only contains the load carrier, the input size for the Pose Estimation model is much smaller compared to a solution without a preceded Object Detection. This increases the performance of the Pose Estimation model rapidly. The used model DOPE extracts the keypoints of the load carrier and uses them to estimate the 3D bounding box of the object (green 3D box in Figure 121). Usually, it is possible that multiple objects are included in the input image of a DOPE model, requiring more logical implementations to separate the different objects. Since each of the cropped input images contains only one load carrier, the pose estimation model can be further reduced resulting in a much more performant model. Finally, the keypoints of the estimated 3D bounding box are considered from a Perspective-n-Point (PNP) solver which calculates the 3D pose relative to the camera based on the camera intrinsics and 2D image points. Finally, the inference node provides the pose of the load carriers under a topic called /poses.

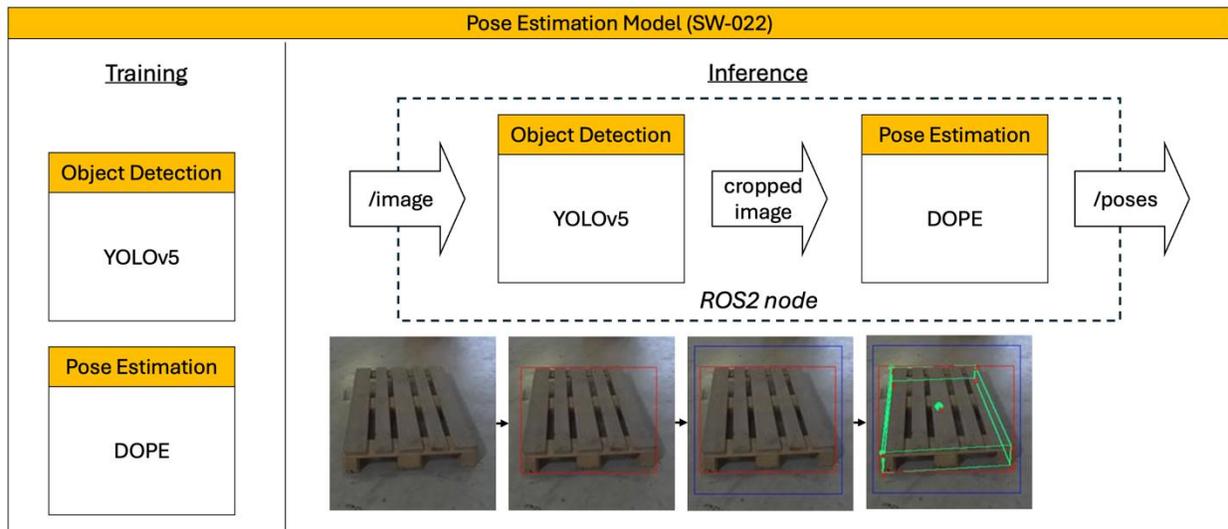


Figure 121. Implementation of the Pose Estimation Model (SW-022).

4.6.3 Results

The Synthetic Dataset Generator allows to create a wide range of synthetic image data – fully customizable by the user. The following options can be configured and randomised among others:

- The model of the load carrier
- Position and Orientation of the load carrier
- The camera position and the distance to the object of interest
- Camera characteristics, such as resolution or Field of View (FoV)
- Random HDR textures as background or simulated warehouse environments
- Textures for the load carrier, its load, and the floor
- Flying objects which distract the view on the objects

These configurations result in a huge variance of images as shown in Figure 122.

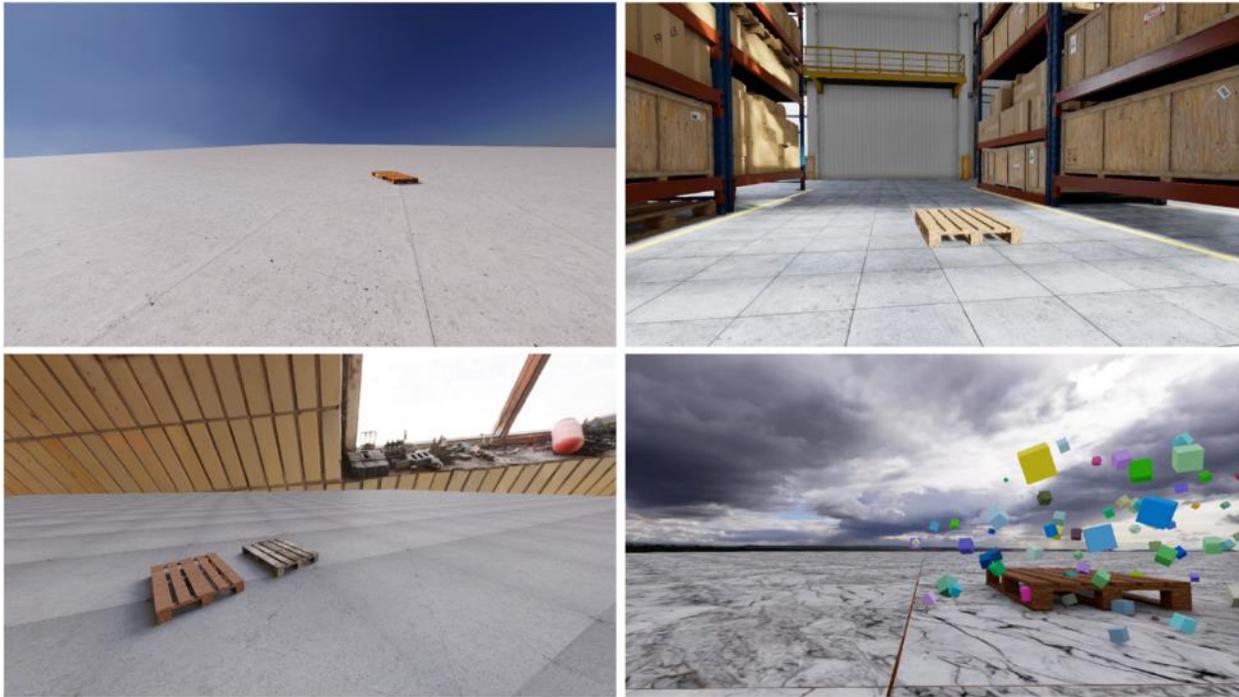


Figure 122. Sample synthetic training images including random background, different textures for floor and background, flying distractors as well as a simulated warehouses.

The trained **Pose Estimation Models for load carriers (SW-022)** detects pallet within the camera data. First, all pallets are recognised by an object detection model. The resulting output of the model is shown in the red bounding box in Figure 123. The bounding box is then extended and the image pixels in the resulting blue bounding box are cropped. These image crops act as input for a pose estimation model, which detects the outer boundaries of the pallet as key points and calculates a 3D bounding box around the object of interest. Overall, the pallet is detected, and the pose estimated based within the real-world data of the iGo neo demonstrator. The model fully trained on synthetic data is capable of provide the pose data of the pallet and allows the demonstrator to pick it up without additional sensor data like LiDAR or depth information.

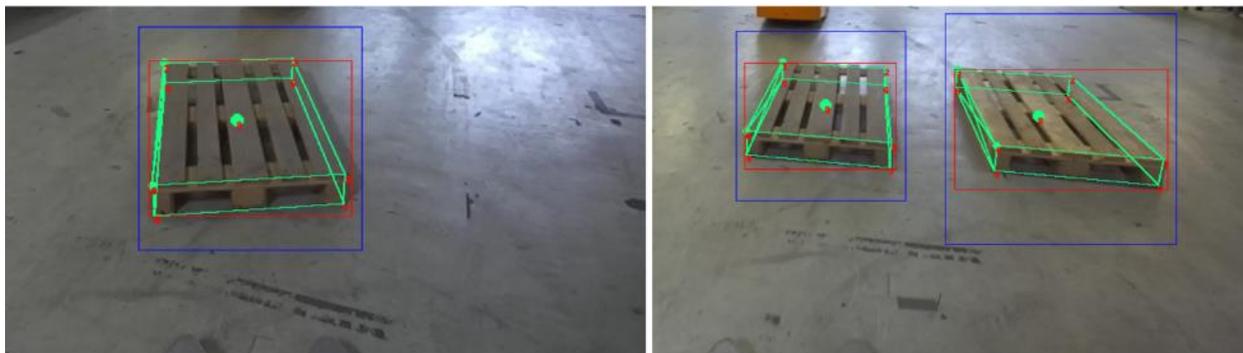


Figure 123. Detected pallets including a red 2D bounding box from the Object Detection, the image crop marked in blue and the final green 3D bounding box of the Pose Estimation. The images are manually cropped to focus on the detected pallet, the FoV of the used camera stream is considerably larger and includes different environment objects.

4.6.4 IMOCO4.E Requirements

Requirements with respect to the used **GPU Edge processing unit (HW-027)**:

- **Req-D3.1-D3-hw**: *The Edge Device shall be equipped with a high-speed Ethernet interface.*
- **Req-D3.1-D3-hw**: *The edge processing unit shall be able to infer state of the art deep neural network architecture with at least 5 frames per second.*

All the GPU Edge processing unit requirement are fulfilled. The final inference rate of the model is still pending since it depends on the finally deployed model and its weights, but the current approaches are faster than 5 fps.

Requirements with respect to the used **RGB(D) camera (HW-028)**:

- **Req-D3.1-D3-hw**: *The sensors shall capture Information in RGB format. Optionally, depth information shall be acquired. The acquisition rate shall be at least 10 frames per second.*
- **Req-D3.1-D3-hw**: *The camera resolution shall be at least 640x480.*
- **Req-D3-003**: *The resolution of the load detection camera must be above 5 MP.*
- **Req-D3-003**: *Sensors must have a reader / controller connected to upper layers (through BB1) by USB or Ethernet.*

All the camera requirements are fulfilled. The as optionally mentioned depth image data of the camera are not needed for the developed solution.

Requirements with respect to the usage of the **GPU Edge processing unit (HW-027)** and **RGB(D) camera (HW-028)** on the demonstrator vehicle **iGo neo (HW-026)**:

- **Req-D3.1-D3-hw**: *The camera field of view shall be in a reasonable range to capture all surrounding objects in both driving directions (e.g., more than 60 degrees).*
- **Req-D3.1-D3-hw**: *The environmental requirements for the camera and edge device shall fulfil current standards and guidelines for intralogistics usage.*
- **Req-D3.1-D3-hw**: *The vision-based solutions shouldn't require different cameras if possible.*

The actual demonstrator fulfils these requirements. They need to be double checked for the final demonstrator, but failing here is not likely.

4.6.5 Capabilities and Limitations

The unique selling point of this solution is easiness of generating a large amount of trainings data for ML based Computer Vision models and the flexibility of the pipeline. Since the simulation allows us to generate high-quality images and to access pose data and the geometry of the objects of interest, we can produce ready-to-use large datasets for ML training within hours. This is a huge improvement for ML projects, which typically suffer from long periods where trainings data are not available since varying real-world data and their annotations are hard to generate. By applying different Domain Randomisation techniques, we see good results for the Sim-to-real transfer and with the trend of pretrained ML models and large Foundation Models, the used models are often already trained on huge real-world databases.

4.6.6 Customizations and Adaptations

The developed solution is highly customizable and can be transferred to other use cases and domains. The overall pose estimation solution can be customized to work with different objects of interest, e.g. with other load carriers like dollies or boxes. One constraint is a unique geometry of the objects of interest, estimating the pose of packages with a random size is not doable. Independent of this limitation, with some changes in the configuration file of the pipeline, the scenario, camera, and much more can be changed. After retraining the AI models, the solution should be applicable. This is true for other P/D/UC in IMOCO4.E or even independent projects which deal with vision-based Machine Learning approaches.

The Synthetic Dataset Generator is even more adaptable. Its basic aim is to generate synthetic images of a simulation model given as input. This can be used for a high variation of AI models. This solution already proves that the generator is usable for an Object Detection Model and a Pose Estimation Model with minor changes. By adding additional writers and other simulation models the component can be adapted for the training of other AI models.

4.6.7 Methodology and Toolchains

The **Synthetic Dataset Generator (SW-021)** is a software component based on Python – running in Ubuntu on a local machine or a server. It utilizes NVIDIA Isaac Sim (Release 2023.1.1) as simulation tool for generating the synthetic data, therefore a NVIDIA GPU is needed. Isaac Sim is integrated as Docker container to reduce the demand of install dependencies. The official Docker images provided by NVIDIA (https://docs.omniverse.nvidia.com/isaacsim/latest/installation/install_container.html) can be used without modifications, making the accessibility of the software much more convenient. By adding some simulation models of the object of interest and textures to create a dataset with a huge variety, the software module can be used to generate synthetic data for training machine learning models. During the project the module has proven to be easily usable to generate annotated data. By configuring the pipeline accordingly, different weightings can be generated in the training data. This is a convenient tool to train different machine learning models. With domain randomization techniques and the right trainings dataset design, the sim-to-real transfer works accurate and robust.

During the project there were multiple problems with Isaac Sim. As it is relatively new, it undergoes frequent changes and instabilities. New releases often resulted in altered system behaviour or new APIs, requiring adjustments to be made to the synthetic data generator. It is recommended to stay at one release for longer periods to avoid frequent problems caused by the release changes.

The **Pose Estimation Models for load carriers (SW-022)** combines state-of-the-art machine learning models (one for object detection, one for pose estimation) based on Pytorch to enable their combination. For the training, existing repositories are used. A Python based custom inference solution is used, which uses OpenCV among others to combine both models for an accurate and robust pose estimation. For the deployment, the inference solution is built as ROS2 (Humble) node to receive the camera images via a ROS topic. The estimated pose is published as ROS topic as well to be used from a control algorithm to pick up a pallet as part of demonstrator 3. A transformation of the models to TensorRT is possible, depending on the final inference rate requirements.

On the way to this solution, further difficulties were encountered in selecting the appropriate machine learning model. Finding suitable models that work well even with a larger distance between the palette and the vehicle was not easy. Additionally, there were deployment issues with the identified models. In several cases, the models showed good inferences on real image data after completing training. However, when

deployed on the Jetson Board using solutions like the ROS GEMs promoted by NVIDIA, the pallets were no longer recognized.

4.7 Deep neural networks based system for deployment on robotic systems to enable recognition of the scene and understanding of complex situations in typical warehouses or production sites - radar signal simulation for AI-driven scene recognition and complexity understanding (DTT, STILL)

4.7.1 Technology overview

DTT has contributed to Task 3.4 in WP3 and BB8. The main objectives of the task are to develop AI-based perception models to enable the recognition of the scene and to understand complex situations. This work is related to the Solution 5 of this task.

In modern robotic sensor systems, vision-based approaches, including stereo vision, depth cameras, and event cameras, are commonly employed. While these methods have successfully addressed numerous challenges, they exhibit limitations in low-light conditions, feature-less scenarios, agile movements, and harsh environmental conditions. To overcome these limitations, the integration of radar sensors is practiced in many applications, leveraging their robustness, low weight, and compact size. Realistic simulation has become integral to research, particularly with the rise of data-driven techniques like machine learning in robotics. Simulation tools enable algorithm implementation in a virtual environment, enhancing flexibility and safety for exploring new methods.

However, existing radar simulation models are often simplistic, generating data loosely in terms of range and velocity. Unlike camera models, radar models lack sophistication and fail to capture the complex information embedded in radar signals, which depend on object shape, size, and material properties. Additionally, multi-antenna configurations are frequently overlooked in current models. Developing a realistic radar simulation framework is crucial for studying algorithms and learning techniques in real-time using raw sensor data. Configurable radar sensors in simulation offer the added advantage of determining optimal sensor configurations for specific applications.

In the past, radar sensor simulations have relied on sensor model approximations using geometrical optics and ray-tracing techniques [1]. Some existing works attempt 77 GHz FMCW radar simulator for phased-array sensors, others use physics-based electromagnetic solvers for generating radar features and extracting range-Doppler maps in ADAS applications and also millimeter-wave radar simulators for traffic scenarios. However, these works primarily generate features like range or Doppler velocity, rather than raw data obtained from real-world radars, limiting the application of different signal processing techniques for feature extraction. Moreover, some works also attempt radar simulation framework embedded in a robotic context, assuming a single antenna radar without considering wave penetration effects and multiple-input multiple-output (MIMO) processing techniques. The existing approaches, therefore, fall short of providing a comprehensive simulation of raw radar data with consideration for advanced processing techniques and material penetration effects. Some of the existing approaches work for scenes with stationary objects, whereas a few of them are suitable for dynamic scenes where both the radar and objects inside the scene are moving.

4.7.2 Implementation aspects

We are using MATLAB and Blender to simulate radar signals. In the first phase of the implementation, we used the ray tracers method [1] to reduce the computational load since accurately simulating radar signals requires simulating the electromagnetic wave which is computationally expensive. Here, the ray tracer is not directly used, rather the rendering results of Blender are utilized for relevant calculation.

The fundamental concept is to consider each pixel in a rendered image as a radar point target, utilizing the superposition of their individual signals to create the simulated raw data. Here, the scene definition has been defined in the Blender with the material radar scattering characteristics, and the geometry of the scene has also been defined. Then radar modulation is used to encode the transmitted signal into the electromagnetic wave from the simulated radar. The transmitted signal contains information on the target’s distance, angle, velocity, and other data. Using Blender, we created radar antennas, transmitters, receivers, and other components, and then, rendering tools were used to simulate the radar modulation process. To generate the micro Doppler raw signals, we are currently working on radar signal processing.

The basic idea here is to treat every single pixel of the rendering result of Blender as a point target. As a result, the superposition of the signals of all pixels becomes the simulated raw signal. The simulation model can be described, according to [1], as below:

$$\underline{x}(n_s, n_p) = \sum_{n_x n_y} A \exp\left(2\pi j\left(\frac{f P \underline{u}}{c} + \frac{2Br}{T_c} T_s n_s + \frac{2fv}{c} T_p n_p\right)\right) + \underline{n}(n_s, n_p)$$

In the above equation, A , \underline{u} , r and v correspond to the amplitude, electrical signal, range and radial velocity of every pixel. The diagram, in Figure 124, illustrates an overview of extracting these target parameters (A , \underline{u} , r and v) from the outputs generated by Blender.

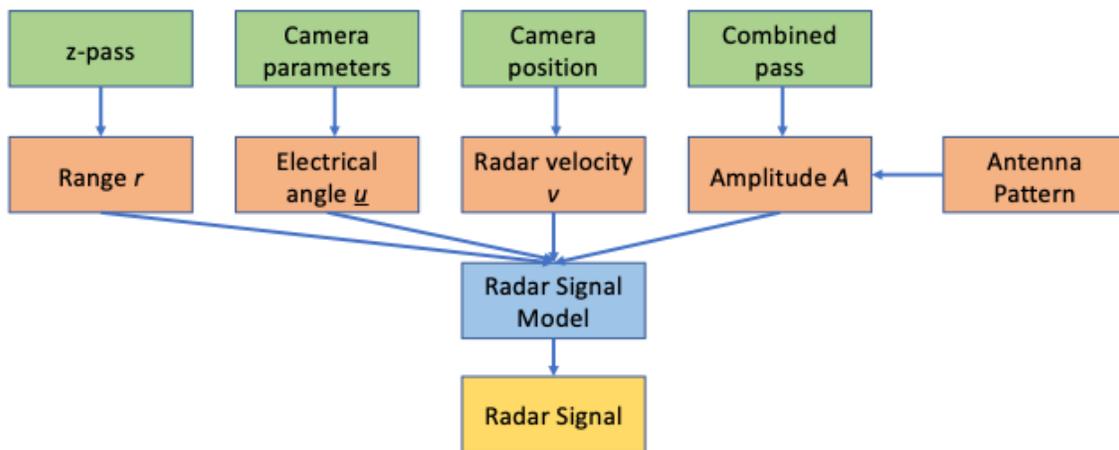


Figure 124. Extraction of target parameters from Blender outputs [1].

Although this approach simulates radar signals more accurately than simple point-target simulations, it has limitations such as in case one object is occluded by another object in the scene, this simulation only

considers the nearer object while a radar system is capable of measuring reflections from both the objects. Moreover, this system is suitable for static environments where the objects do not change positions.

To address this limitation and pursue radar simulation for a dynamic environment where both the radar and objects move, we looked into another approach [2] of radar simulation. This approach corresponds to a real-time radar simulation framework for perception-driven robotic applications. Built on the Unity game engine, this system utilizes its inherent rendering pipeline and incorporates custom shaders, enabling real-time radar signal generation at the GPU level. The choice of Unity is motivated by two main factors: (i) its capacity to simulate realistic scenarios with diverse environmental conditions and support cutting-edge physics engines, and (ii) its ease of integration into existing frameworks, facilitating extensions to established simulation pipelines. Figure 125 depicts a schematic overview of the simulation pipeline and also a high-level schematic overview of the Frequency-modulated continuous wave (FMCW) radar principle.

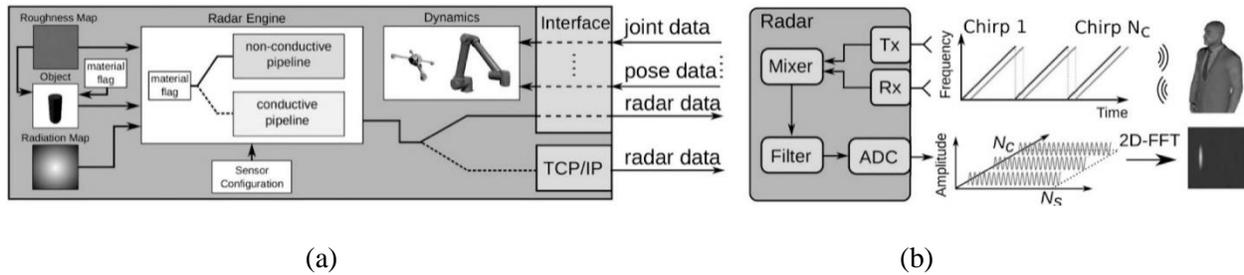


Figure 125. a) Schematic overview of the simulation pipeline. (b) High-level schematic overview of the FMCW radar principle. The signal paths are replicated when more receiver or transmitter antennas are used [4].

4.7.3 Results

The blender-MATLAB implementation is tested on a blender scene with two detectable objects as shown in Figure 126. The radar moves along a predefined path when the simulation process is initiated. The simulation collects the following data:

- a) The z-pass gives the distance of the objects to the camera plane for each pixel, see Figure 127.a.
- b) The combined-pass is the result of Blender ray tracer, namely the radiation intensity in various directions (image brightness for all pixels, see Figure 127.b).

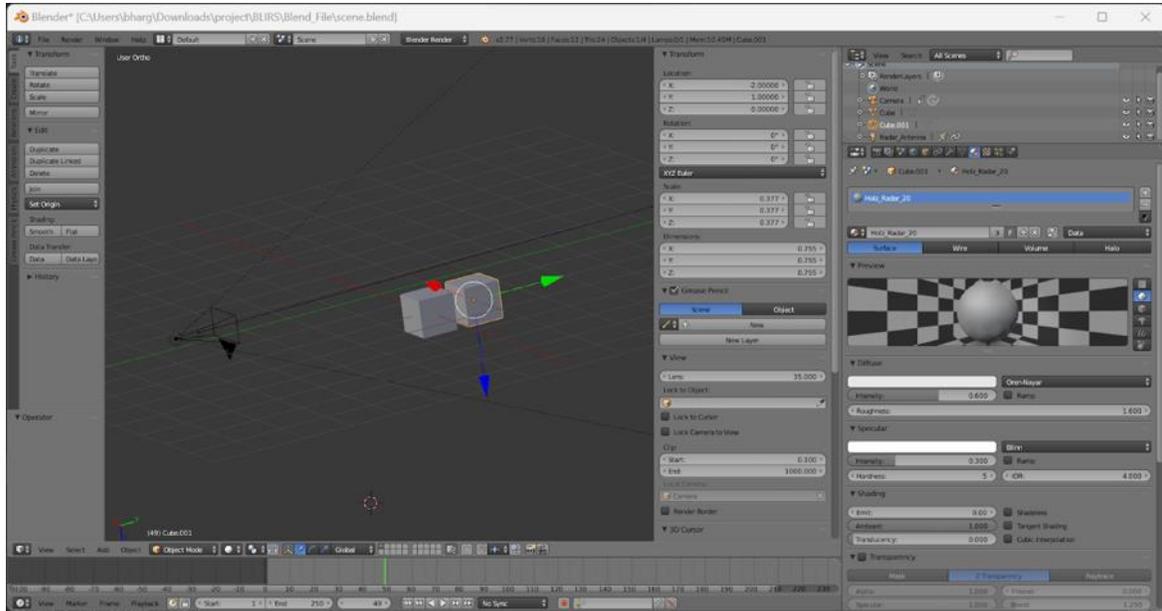


Figure 126. Sample scene in Blender showing the virtual radar with two detectable objects.

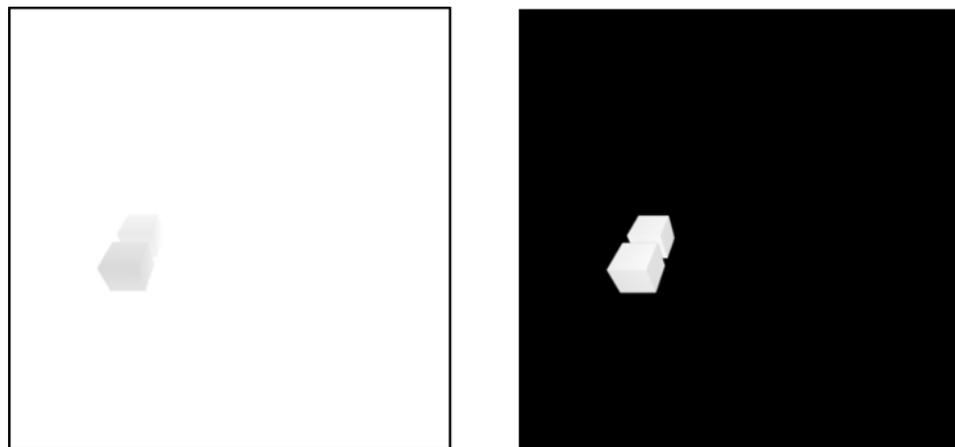


Figure 127. a) Z-pass, b) combined-pass.

Following the blender data generation, the MATLAB is used to generate the radar raw data in a complex number format. The data is then processed using to find the number of targets the radar as detected in the given scene. The plot in Figure 128 shows the number of targets detected for n number of frames defined by user in the simulation properties.

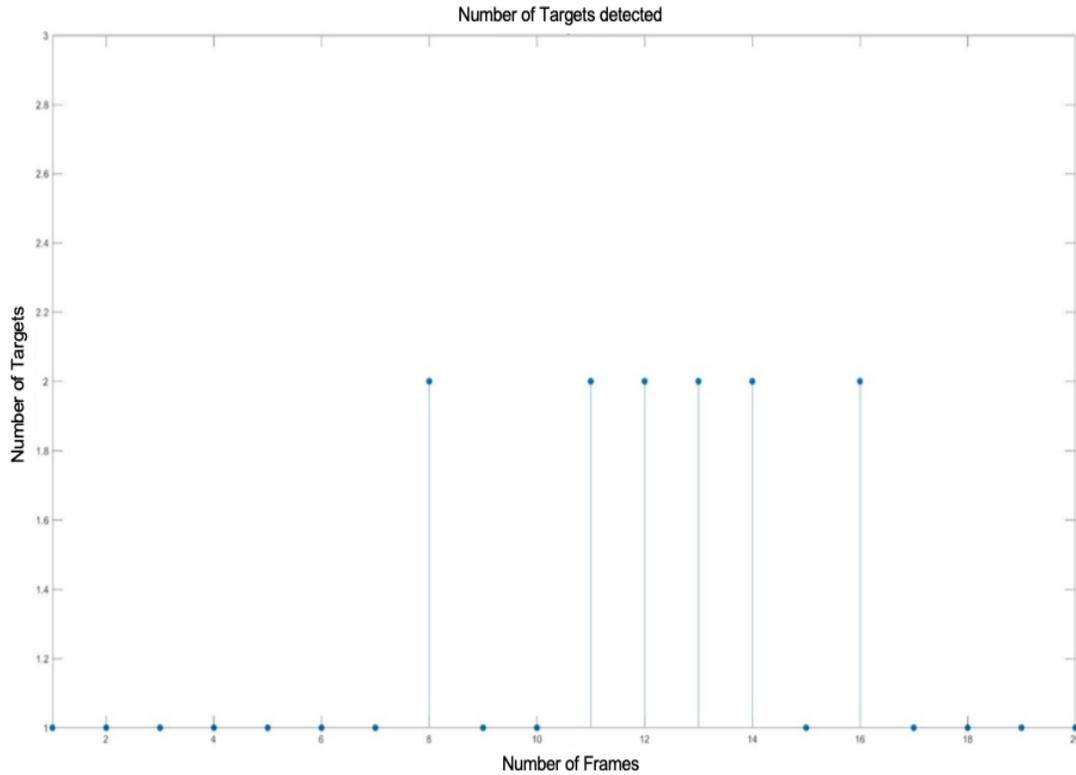


Figure 128. Number of targets detected for each frame.

The Unity-MATLAB implementation is tested on a similar scene with two detectable objects as shown in Figure 129. The approach is tested in two configurations – 1) the radar and the objects are in a stationary position and 2) the radar and one detectable object are in a stationary state with other detectable object moving away from the radar’s field of view.

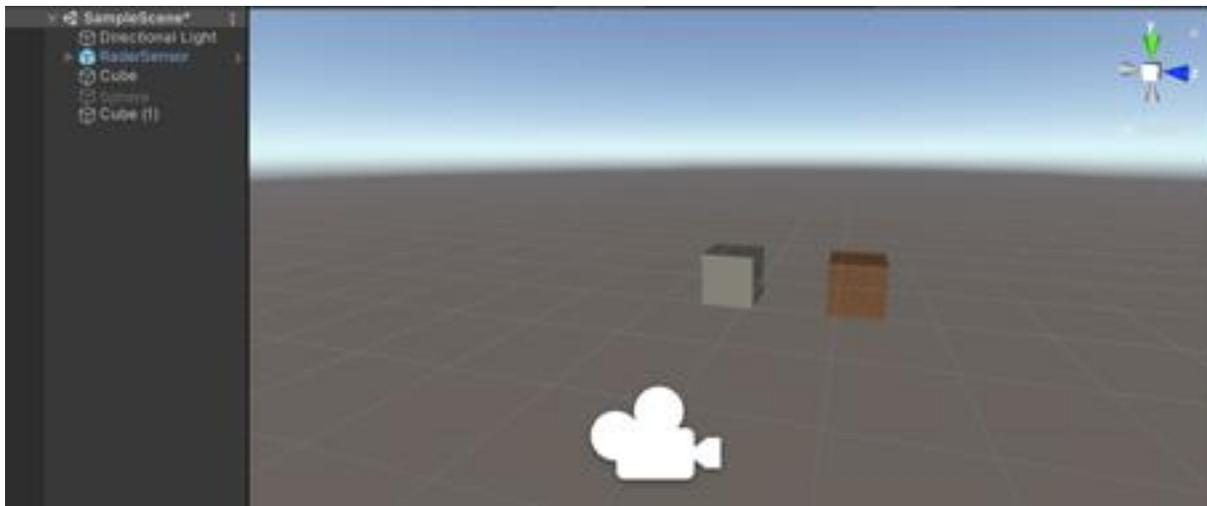


Figure 129. Sample scene in Unity 3D showing the virtual radar with two detectable objects.

Figure 130 illustrates the range of detected objects for m user-defined samples in the simulation properties for test case 1, while Figure 131 depicts the range of detected objects for m user-defined samples in the simulation properties for test case 2.

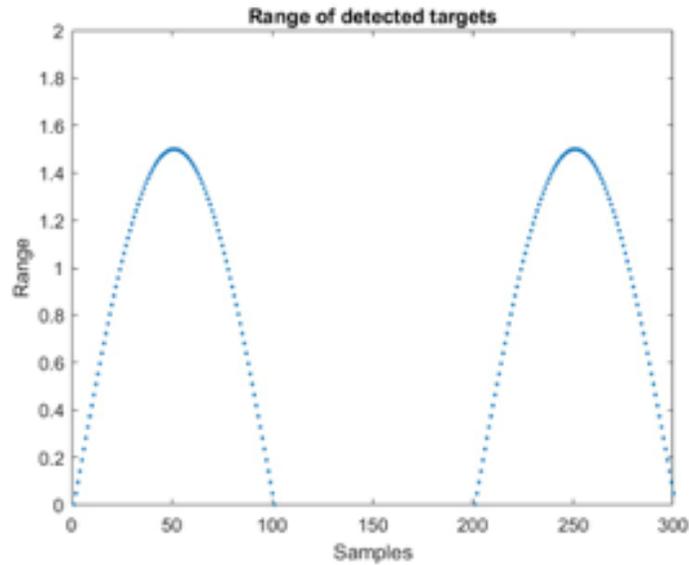


Figure 130. The range of the detected objects with radar and detected objects in stationary state.

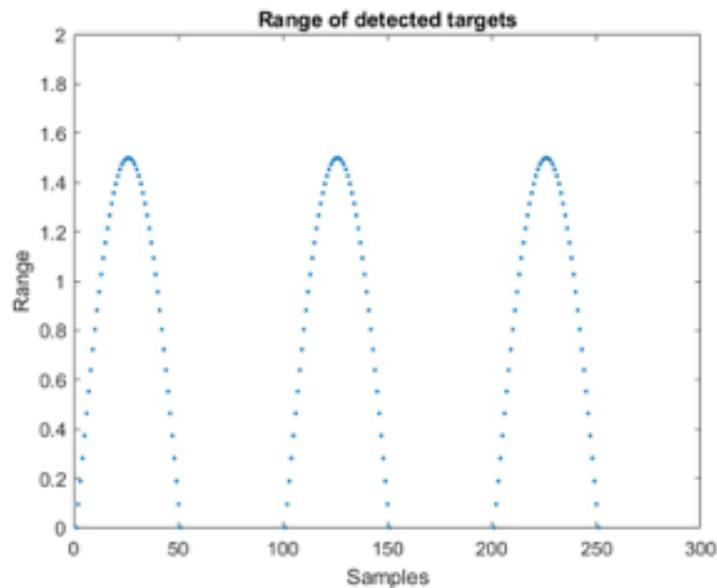


Figure 131. The range of the detected objects with radar and one of the detectable objects in stationary state and one of the detectable objects disappearing from the scene.

4.7.4 IMOCO4.E Requirements

This activity meets the following requirements:

Table 39. Requirements for Radar Signal Simulation for AI-driven Scene Recognition and Complexity Understanding.

ID	Requirement	Priority	Verify	Tasks
R063-D2.3-L2	The optimised Neural Networks must be able to run on the available (brownfield) hardware platforms.	M	D	T3.3, T3.4
R016- D2.3-U3- hw-sw	Co-existence of Information Technology (IT) and Operation Technology (OT) on the same network infrastructure will be supported.	M	I	T3.3, T3.4
R020- D2.3-L2- L3-L4-B5- B6-B8- B10-P4	Algorithms, AI-components and digital twin models that are intended for real-time deployment shall not adversely affect the responsiveness of the system to user requests.	M	T	T3.1, T4.1, T5.1
R047- D2.3-L2- L3-L4-B5- B6-B8- B10	Any smart control algorithms, AI- components and digital twin models shall not adversely affect the safety of the system.	M	T	T3.1, T4.1, T5.1, T6.1
R197-D2.3	All (generated) software for hardware targets should also run on digital twin (unmodified).	S	D	T3.1, T4.1
R187-D2.3	Support and be operational in multiple Pilots/Demos/Use-cases	S	I	T3.1, T4.1, T5.1

4.7.5 Capabilities and Limitations

Our solution offers customizable AI-based perception models tailored for various scenarios, specializing in scene recognition and understanding complex situations. It can be tailored to various scenarios that necessitate AI-based perception models for scene recognition and comprehension of complex situations. It includes a real-time radar simulation system built on the Unity game engine, leveraging custom shaders and GPU-level processing for efficient signal generation within dynamic environments. This comprehensive approach enables high-fidelity simulations, real-time performance, and versatility across industries, though it may require expertise for development and integration and could incur hardware and cost limitations.

The unique selling point of this solution is that it stands out with its holistic approach, combining AI-based perception models and real-time radar simulation for comprehensive understanding of dynamic environments. Leveraging GPU-level processing and custom shaders ensures optimal performance and fidelity. Its versatility and customization options cater to a wide range of industries, while its future-proofing capabilities make it adaptable to evolving technological landscapes.

The strengths of our solution are that it offers quite versatile advantages in the domain of scene understanding, including high-fidelity radar simulations, real-time performance driven by GPU-level processing, and versatility across industries thanks to its customizable AI-based perception models. Its integration with the Unity game engine provides a user-friendly development environment, while its comprehensive approach ensures holistic solutions for understanding dynamic environments. While the solution boasts high fidelity and real-time performance, it might face limitations such as potential hardware requirements for GPU-level processing, complexity in development and integration, and the labor-intensive nature of training data generation for AI models. Additionally, the costs associated with building and maintaining such an advanced solution may be a consideration for some users.

4.7.6 Customizations and Adaptations

This solution's adaptability extends its utility beyond its primary use in IMOCO4.E, catering to diverse scenarios and challenges with customizations and adaptations. It excels in dynamic scene recognition, offering optimized performance in low-light conditions, feature-less environments, agile movements, and harsh conditions like dirt, dust, or moisture. By tailoring the solution to specific requirements, it ensures reliable operation where traditional vision-based approaches may struggle, maintaining accuracy and responsiveness even in fast-paced environments. Versatile applications across Platforms, Devices, and Use Cases (P/D/UCs) further highlight its flexibility and scalability, while continuous modifications and extensions future-proof its capabilities for evolving needs and advancements. In essence, this solution provides tailored solutions for a wide range of industries and applications, empowering users to overcome challenges and achieve optimal performance in dynamic environments.

4.7.7 Methodology and Toolchains

The methodology employed for radar signal simulation involves leveraging MATLAB and Blender in the initial phase, utilizing ray tracing methods to reduce computational load while maintaining accuracy. Blender's rendering results are utilized for relevant calculations, with each pixel in the rendered image representing a radar point target. By considering the superposition of individual signals from these targets, simulated raw data is created, encoding information such as distance, angle, and velocity. While this approach provides more accurate simulations compared to simple point-target methods, it has limitations such as occlusion issues and suitability only for static environments. To overcome these limitations and enable radar simulation in dynamic environments, a real-time framework is developed using the Unity game engine. This system utilizes Unity's rendering pipeline and custom shaders for real-time radar signal generation at the GPU level, offering realistic scenarios and ease of integration into existing frameworks. Lessons learned include the importance of balancing computational complexity with simulation accuracy and the value of leveraging versatile platforms like Unity for dynamic environment simulations.

4.7.8 References

- [1] M. Ouza, M. Ulrich and Bin Yang, "A simple radar simulation tool for 3D objects based on blender," 2017 18th International Radar Symposium (IRS), Prague, Czech Republic, 2017, pp. 1-10, doi: 10.23919/IRS.2017.8008254.
- [2] C. Schöffmann, B. Ubezio, C. Böhm, S. Mühlbacher-Karrer and H. Zangl, "Virtual Radar: Real-Time Millimeter-Wave Radar Sensor Simulation for Perception-Driven Robotics," in IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 4704-4711, July 2021, doi: 10.1109/LRA.2021.3068916.

4.8 Deep learning algorithm for Vision, audio, acceleration, pressure sensor data analysis; Ultra-low power ASIC/FPGA hardware implementation of neural networks (Reexen)

4.8.1 Technology overview

CIM (computing in memory) is expected to break the von Neumann architecture and is inevitable in the post-Moore era: CIM is to embed computing power inside memory and use new computing architecture to perform 2D/3D matrix multiplication/addition operations. The logic and advantage of CIM is to directly use memory for data processing and calculation, thereby integrating storage & calculation in the same area of the same chip, which can eliminate von Neumann bottlenecks.

Suitable for large-scale parallel scenarios such as deep learning with large amounts of data @low precision.

Improve computing efficiency thousands of times & reduce costs: CIM is to break the storage wall, eliminate unnecessary data movement delays and power consumption, and use memory units to increase computing power;

Provide higher computing power & efficiency in specific fields: CIM architecture eliminates the boundaries between computing and storage and completes calculations directly in memory. Non-von Neumann architecture provide greater computing power (> 1000TOPS) and higher efficiency (> 10-100TOPS/W), significantly surpassing existing ASIC chips;

CIM represents the mainstream of future AI chips: In addition to AI computing, CIM can also be used in neuromorphic chips, which can reduce unnecessary data movement and use memory units to participate in logical calculations and improve computing. It is to increase the number of computing cores on a large scale without changing the area.

4.8.2 Implementation aspects

Reexen: Digital-Analog Mixed-Signal, High Speed Analog and CIM based high-speed/efficiency processing

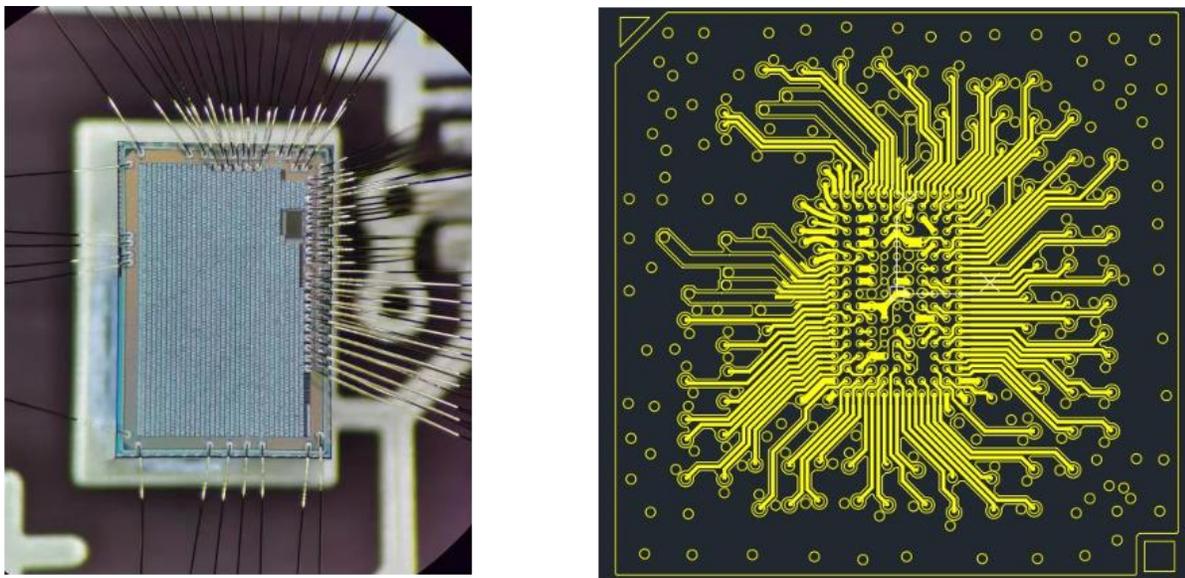


Figure 132. ADA200 CIM TEST Chip.

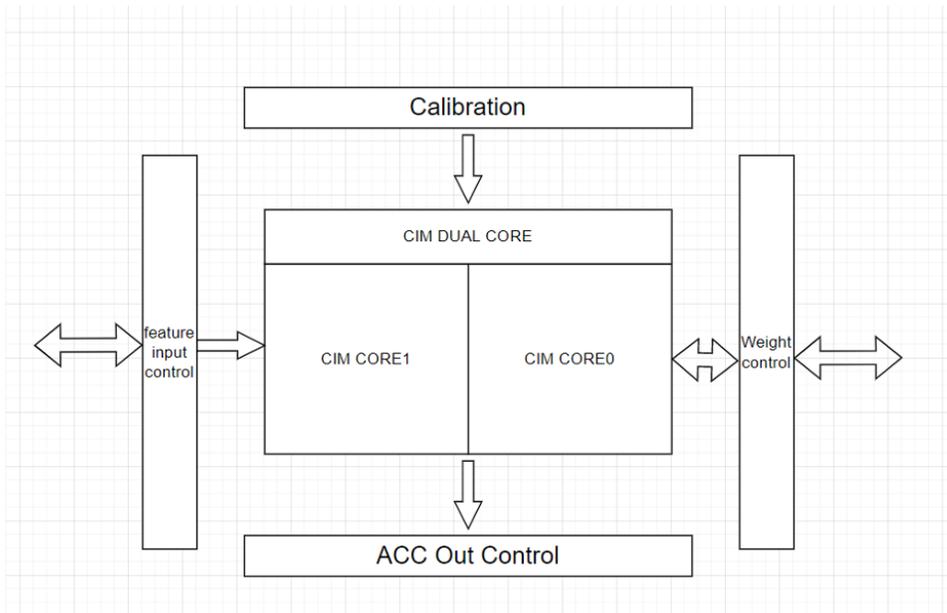


Figure 133. 3 data flow and CIM energy-efficient AI processors.

This processor serves as a core component of visual processing technology, primarily applied in visual collaborative processing tasks for devices. It is capable of handling complex visual signals across both temporal and spatial dimensions, and further enhances the accuracy and richness of visual experiences by fusing multiple signal sources.

The core strength of this processor lies in its efficient workload offloading capabilities. The visual processing tasks of the main control chip, significantly reducing power consumption. This workload offloading not only enhances the overall system performance but also extends the lifespan of the device.

Moreover, centered around this core functionality, the processor possesses robust scalability. As technology continues to advance, it can adapt to new application scenarios and demands, providing robust technical support for the future development of AR/VR devices. Whether it's handling more complex visual processing tasks or delivering superior visual experiences, this processor can offer robust support, driving continuous advancements in visual technology.

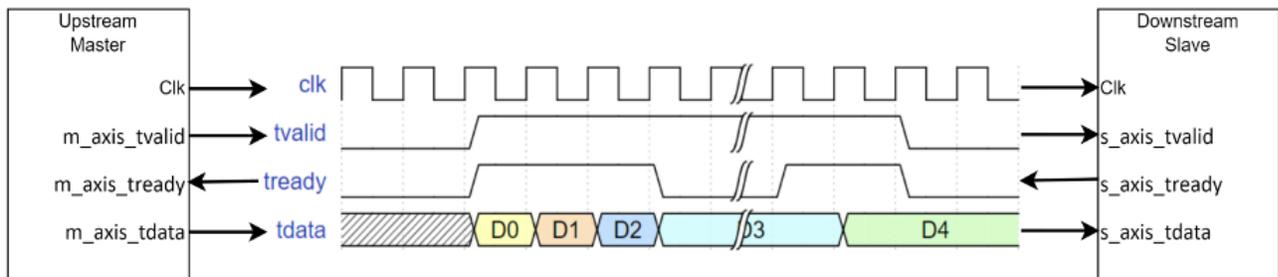


Figure 134. Chip clock logic.

4.8.3 Results

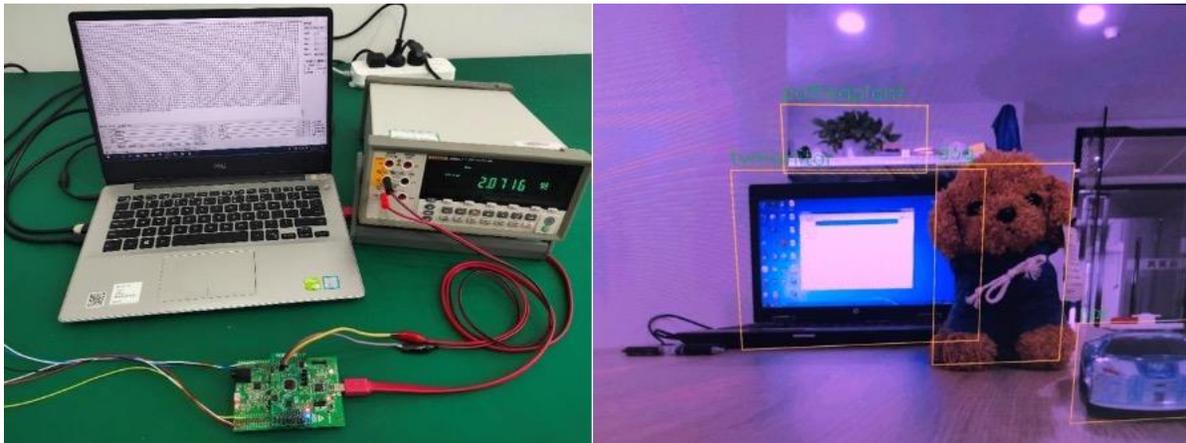


Figure 135. Test ADA200 Chip and YOLO Result.

- 100-run on one column in one chip, 100% correct computation;
- 10-run on one column in three chips, 100% correct computation
- 1-run on >20 columns in one chip
- max 1 step mismatch in a few columns under investigation weights stored in CIM cores while input data and feature maps flow through different cores

In today's world, augmented reality (AR), virtual reality (VR), robotics, and autonomous driving are rapidly transforming our lives. These technologies rely heavily on sensors to capture and process vast amounts of data in real-time. However, with the increasing complexity and scale of these applications, the demand for high-energy-efficiency processing chips has become increasingly urgent. Sensors play a crucial role in these technologies, providing the necessary information for precise navigation, decision-making, and environmental perception. However, the continuous stream of data generated by these sensors requires significant computational power, often leading to high energy consumption. This is a major challenge, especially for mobile devices and autonomous vehicles that need to operate efficiently for extended periods. High-energy-efficiency processing chips are crucial to addressing this challenge. These chips are designed to provide powerful computing capabilities while minimizing power consumption. By optimizing the chip's architecture and using energy-efficient technologies, manufacturers can create processors that can handle the demands of AR/VR, robotics, and autonomous driving without draining energy resources. In conclusion, the need for high-energy-efficiency processing chips is paramount in the age of AR/VR, robotics, and autonomous driving. By developing more efficient chips, we can enable these technologies to reach their full potential, transforming our world in ways we can only imagine.

4.8.4 IMOCO4.E Requirements

Table 40. Requirements for solution.

ID	Requirement	Priority	Verify	Task	Result
Req-D3.2-L1-hw	Used camera needs to provide RGB images.	M	T	T3.4	Achieved

Req-D3.1-L3-sW	The perception model should handle as less different data as possible(input: img-data + optional depth data; output: pose)	S	T	T3.4	Achieved
Req-D2.3	BB1 should have an interface with camerasensors.	M	T	T3.4	Achieved
Req-D2.3	BB1 should have enough memory to allow for buffering more than 6images from the camera sensors	M	T	T3.4	In progress

4.8.5 Capabilities and Limitations

With the evolution of CIM, the prerequisites for ensuring that Neural Networks or algorithms running on hardware can achieve commercial precision are:

Higher storage density(MB/mm2)

Higher area efficiency (TOPS/mm2)

Higher energy efficiency (TOPS/W)

Increasing Demand for High-Performance Computing: As the world becomes increasingly reliant on technology, the need for faster and more efficient computing solutions is on the rise. Memory-compute integrated chips (MCICs) offer a unique advantage in this regard, combining memory and processing units into a single chip, thereby significantly reducing data transfer latency and enhancing overall system performance. This makes MCICs ideal for high-performance computing (HPC) applications, such as artificial intelligence, machine learning, and big data analytics.

Need for Energy-Efficient Solutions: With the escalating concern about energy consumption and its environmental impact, the market is looking for more energy-efficient computing solutions. MCICs are designed to reduce power consumption by eliminating the need for separate memory and processor chips, which often consume significant amounts of energy. As a result, these chips are becoming increasingly popular in markets that value energy efficiency, such as mobile computing, IoT devices, and wearable technology.

Growing Trend of Edge Computing: The emergence of edge computing is another key driver for the demand for MCICs. Edge computing refers to the processing of data at the peripheral devices rather than relying on remote servers. This requires powerful and efficient chips that can handle both data storage and processing tasks. MCICs, with their ability to integrate memory and compute functions, are well-suited for edge computing applications, enabling faster and more responsive systems. As edge computing continues to gain popularity, the demand for MCICs is expected to grow significantly.

Strengths and weaknesses:

Integration of Advanced Process Technologies: The development of memory-compute integrated chips (MCICs) requires the integration of multiple and advanced process technologies. This integration is complex and challenging, as it involves combining different materials, devices, and circuit designs into a single chip. The complexity of this integration process increases the difficulty of research and development, requiring a significant amount of expertise and technological capabilities.

High Research and Development Costs: Developing MCICs involves substantial financial investments in research and development. The need for advanced process technologies, coupled with the complexity of integrating multiple functions into a single chip, drives up the costs significantly. Additionally, attracting

and retaining talent with the necessary expertise to carry out this research can also be costly. (R&D COST)

Talent Shortage: The development of MCICs requires a highly skilled workforce with expertise in multiple fields, including semiconductor processing, circuit design, and systems integration. However, there is currently a shortage of such talent, as the field of integrated chip development is highly specialized and requires a deep understanding of various technologies. This talent gap can pose a significant challenge to the development and commercialization of MCICs.

4.8.6 Customizations and Adaptations

Customization and adaptability are crucial. To cater to the needs and usage scenarios of different user groups, we can adjust product features, design elements, and user interfaces based on specific applications. For instance, we can add customized features for specific user groups or integrate the product with external systems, such as other software, hardware, or services, to provide a more comprehensive and personalized experience.

Customization Settings: Users can adjust various product settings according to their preferences and needs, including interface themes, font sizes, color schemes, and more.

Theme Changing: Products often provide multiple themes for users to choose from, satisfying different aesthetic preferences.

Third-Party Service Integration: Users can integrate the product with third-party services (such as social media, cloud storage, etc.) to expand the product's functionality and application scope.

Feature Enhancement: Based on user feedback and market demand, we can enhance the product's features, such as adding new algorithms, optimizing performance, and more.

Security Improvement: As cybersecurity issues become increasingly severe, we may need to strengthen the product's security, such as adopting more advanced encryption technologies, adding authentication mechanisms, and more.

Multilingual Support: To cater to the needs of global users, we can expand the product to support multiple languages and cultures.

4.8.7 Methodology and Toolchains

In traditional digital PE designs, even when the memory is placed close to the MAC computation unit, the energy consumption associated with data movement cannot be ignored. In MIT's Eyeriss (65nm), under the row-stationary dataflow, the power consumption of the local scratch pads reaches 5 to 10 times that of the MAC.

On the premise of maintaining the advantage of fewer partial sum moves, mixed signal CIM minimizes the area overhead outside of the storage part in MAC operations, so the storage density of the CIM part can be greatly increased. Specifically, in the pipeline architecture adopted to reduce the storage space and movement of output feature maps in the demand for real-time processing on mobile/edge platforms, an increase in CIM storage density is beneficial in reducing the gap between the required computing power without updating CIM parameters and the maximum computing power that CIM cores can achieve, reducing CIM parameter updates or/and improving the utilization of CIM cores.

4.9 Real-time predictive models, accelerated for embedded processing accounting for latencies and deficiencies in gesture recognition (EMD)

4.9.1 Technology overview

As UC3 partner, EMD’s expertise relates to embedded AI potential at both the local end and the remote end of the architecture. The two previous solution sections (2.5 and 3.4) have provided background to UC3 with reference to ToF sensors (ADI) and the lower layer 1 edge infrastructure (UCC/Tyndall and ADI). With this in mind, this solution section will present the UC3 technology overview from an AI perspective and with reference to the IMOCO4.E architecture.

As already presented in previous sections, UC3 has both a local sensor end architecture (where the user will operate) and a remote robotic systems end architecture. The next two sections present a discussion on AI models, methods, and technical approaches in an embedded processing context. Both the local and remote architecture diagrams from previous UC3 solution sections are reproduced below for reader convenience.

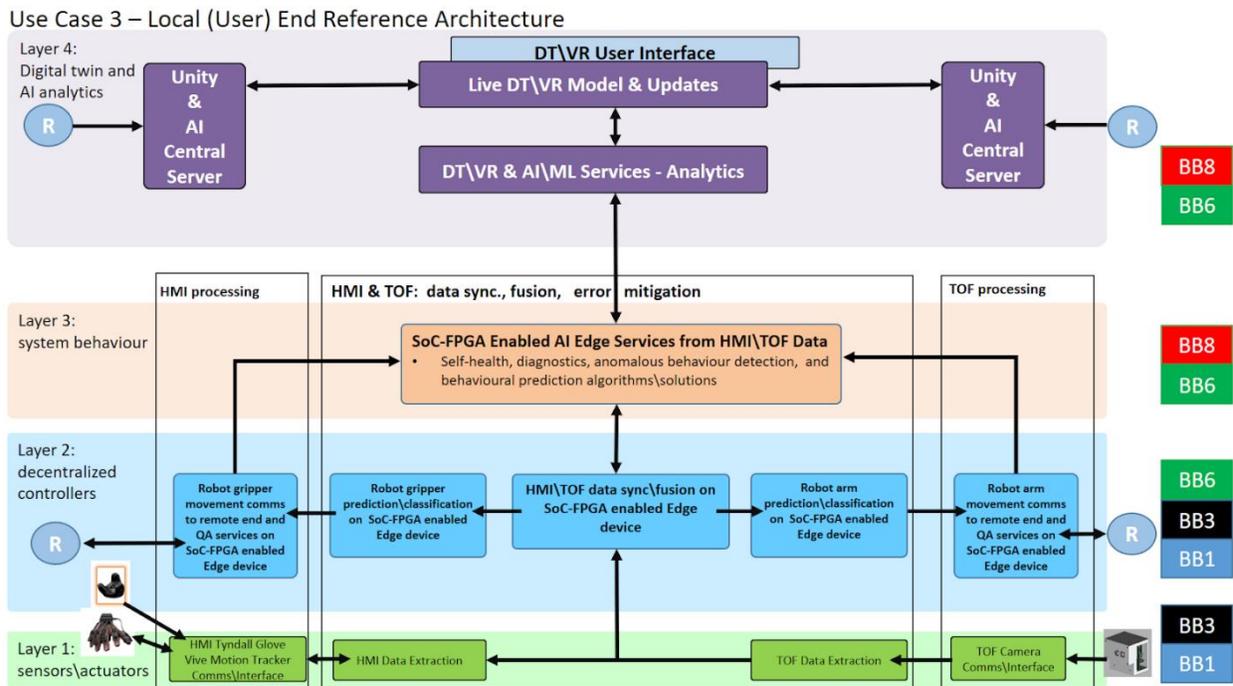


Figure 136. Use Case 3 - Local Sensor End Architecture.

Use Case 3 Architecture - Local End – Embedded Processing: As per the above figure there is embedded AI processing provisioned at Layer 1 (L1), sensors/actuators layer. Here EMD is working closely with ADI at their Catalyst facility. AI processing is working across server and edge devices with the deployment of various algorithms for vision and gesture analytics. As per the diagram above, the local end Layer 2 (L2) is interfaced with Layer 3 (L3), system behaviour and both layers will provide robot movement processing and analytics that will evolve and be complimented by various AI and machine learning (ML) tools and techniques. Note that L2 is engineered to be able to function independently of L3 as and when required. The potential for the L3 system behaviour *Embedded AI Edge Services* is where future AI engineering is expected to take place. Such engineering may relate to services to the Digital Twin (DT) and

Virtual Reality (VR) Layer 4 (L4) or apply to advanced sensory fusion for machine learning processing and updating.

To date, most of the embedded processing has been conducted for the local sensor end of the UC3 platform. Please refer to the next section for further details on the embedded processing at the remote end.

Use Case 3 – Remote (Robot) End Reference Architecture

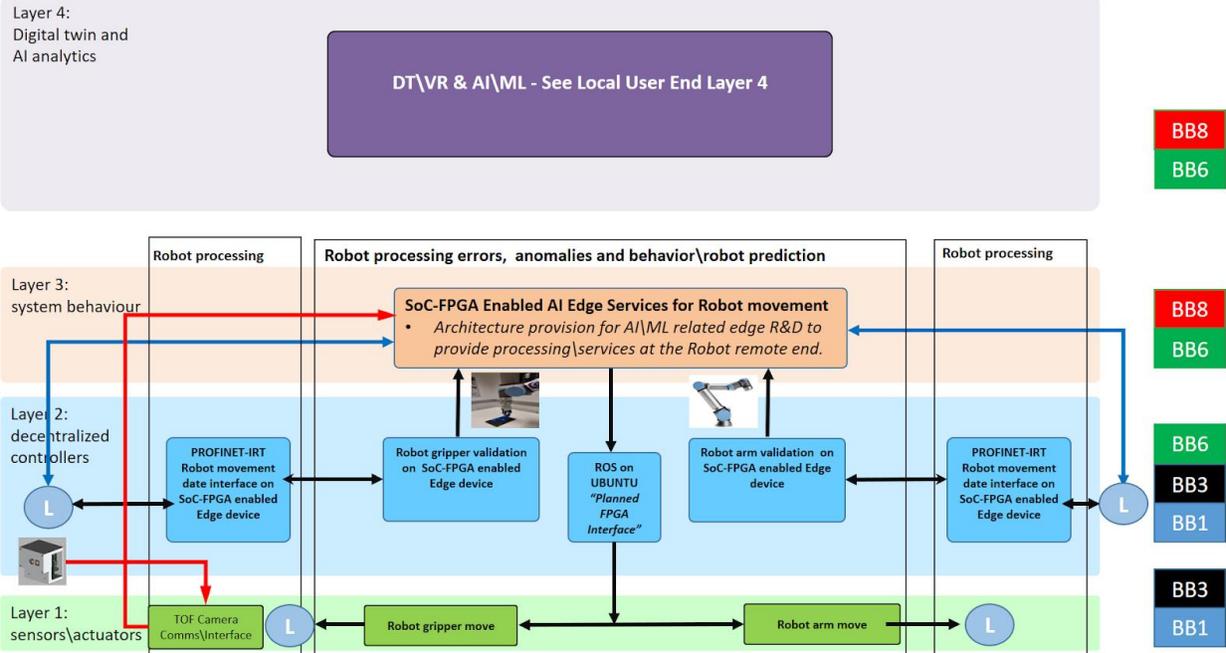


Figure 137. Use Case 3 - Remote Robot End Architecture.

Use Case 3 Architecture – Remote End – Embedded Processing: With reference to the remote end of UC3, robot movement coordinates received at L3 are the inputs to a set of AI and ML services that can be future engineered and deployed at the remote end. In particular, the use of ToF technologies at the remote end of the platform can provide potential to embed vision based detection capabilities to the platform, which can then be used to provision AI powered object and robotic movement recognition functionality. Such object and robot movement perception capabilities endowed to the remote end of the teleoperation platform can also offer potential to use task based knowledge to part complete work in advance on behalf of a user. Such applied functionality using ToF and AI technologies offers the potential to research novel approaches in addressing end-to-end latency challenges in robotic teleoperations.

4.9.2 Implementation aspects

This section primarily discusses the implementation aspects of the embedded AI processing at the local and remote ends of UC3.

Use Case 3 Architecture - Local End – Implementation: This section provides a general overview of the embedded processing research and engineering under development at the local end of the platform. The reader is referred to D3.5 where detailed discussion have been provided on the various approaches implemented.

Gesture Recognition: In the teleoperations robotic system, hand gestures are used for various purposes, including for system safety, such as sending an emergency stop command in the event of a hazardous situation. Gesture control systems typically involve the use of sensors, cameras, or other devices to capture human gestures, and then AI algorithms to interpret these gestures into commands or to control robotic actions. As part of the research conducted for real-time hand gesture recognition, a total of 11 different pretrained neural networks were evaluated. The fastest method identified for real-time hand gesture recognition was **Mediapipe**, which enabled inference at 30 frames per second (FPS) on a simulated system. Gesture control advancements are seen as a key enabler of robotic teleoperations and a core research track for UC3. An example of this in operation is demonstrated in the figure below.

In relation to future research, the model performance will need to be evaluated and optimised once it is deployed to the embedded edge device (PolarFire) in order to confirm that inferencing does not suffer any significant degradation in performance.



Figure 138. UC3 AI and ML research on gesture recognition.

ToF Sensors: The ADI ToF section has already provided details of the technologies used and the specific ADI embedded processing and AI algorithms used in order to guide the robot. Working with ADI, EMD are also addressing various latency aspects (see below section) of the ToF sensor setup for robotic control. The next stages of the ToF research work with ADI will involve a number of tasks to port from the typical PC architecture to the embedded edge device (PolarFire). This transition to the edge can be expected to introduce latency not only in relation to the ToF processing but also in relation to AI model inferencing on the edge device. The characterisation of the embedded processing versus the PC architecture based processing will form part of future testing to be reported into WP6 final deliverables for UC3.

Latency Reduction: The teleoperations robotic platform link between the local and remote locations may suffer from high latency in certain settings. The delay caused by the high latency in the network link is a significant problem for any teleoperations robotic platform. Reducing the latency is crucial to improving the operator's effectiveness and the overall performance of the system.

An approach to address latency using selected AI algorithms and techniques is underway as part of the UC3 research efforts. AI algorithms and techniques that can predict time series data with low processing latency and high accuracy are being researched and evaluated. For example, the Kalman Filter is a widely used algorithm for time-series prediction. The Kalman Filter approach can be modified with AI techniques to improve time-series prediction by using machine learning algorithms to estimate the process and measurement of noise covariance matrices. Another approach is to incorporate deep learning models to learn the dynamics of the system and update the Kalman filter's state estimation and error covariance matrices.

In terms of process, potential latency reduction algorithms are tested in isolation of the final teleoperations robotic system (“offline”). Algorithms are evaluated for their overall latency reduction capabilities and the amount of error introduced. Once an optimal algorithm is identified, it will be deployed to the PolarFire edge devices for testing as part of the overall system (“online”).

Use Case 3 Architecture - Remote End – Implementation: The UC3 architecture at the remote end currently provides for embedded AI processing and analytics in relation to sensor data, robotic movements and ToF vision data.

Object Recognition and Pose Estimation: The remote UC3 platform edge processing infrastructure can provision for the integration of semantic object recognition as UC3 moves up the TRL scale as part of future research. The approach to object recognition is to use 6D pose estimation to recognize the location and orientation of objects in 3D space at the remote end of the system. To achieve this, a 3D ToF sensor will capture an image of an object; AI algorithms will conduct feature extraction and image segmentation, to identify the object in the remote scene. Once an object is identified, a 6D pose estimation algorithm will be applied to estimate the position and orientation (poses) of each object in 3D space. These pose predictions can then be processed by the teleoperations robotic system, to command the robot to pick up specific objects in a typical industrial scene.

Remote Task Based Knowledge: The UC3 partners, during the development of the IMOCO4.E guided architecture, have deeply considered the importance of embedded AI and data analytics at the remote end. As discussed ToF guided object recognition has the potential to embed robot intelligence in handling and managing a specific teleoperation task. Combining remote object recognition capabilities with a deep semantic understanding of how a specific task is carried out to a defined standard (task based knowledge) opens up the potential to further minimise the latency from the local user end to the remote robotic end of the teleoperations platform. Unfortunately, this very exciting field of AI task/context awareness is outside the scope of the current project but it is certainly a research domain to be built upon as part of any future research and development projects.

4.9.3 Results

This section provides outline guidance on a selection of tests/results that have been conducted or that are currently underway. The reader is referred to the deliverables of WP6 for on-going updates in relation to the tests/results outlined below.

Real-Time Hand Gesture Recognition inference time: The latency in our system may be reduced by defining sequences of pre-programmed commands and triggering them using hand gestures. Eleven different pretrained neural networks were tested for their accuracy, size, average inference time, and mean frames per second in real-time hand gesture recognition.

Results: The fastest approach was found to be Mediapipe, which achieved inference at 30 FPS in offline testing. The next phase of this testing will involve the deployment of the algorithms at the ADI Catalyst centre using their UR5e robot infrastructure. Focus will be on performance and latency as code bases move from PC based to edge architecture.

Time-series Prediction for Human Motion Prediction (Offline): Description: The aim of this test is to reduce the latency between user input and the motion of the robot by researching and testing time-series prediction algorithms offline. Several potential algorithms will be identified, including Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells, Convolutional Neural Networks (CNNs) with sliding windows, and Kalman Filters. The selected algorithms will be implemented using Python or C/C++ and open-source libraries such as PyTorch and NumPy.

Results: The Kalman Filter emerged as the optimal choice, offering a compelling balance between high prediction accuracy and minimal processing time. While more complex algorithms demonstrated the potential for enhanced predictions, their performance was inhibited by the absence of a larger training dataset. The accuracy of predictions could be further refined by incorporating task-specific information. In our study, we used random hand movements without a predefined task. Thus, a task-aware approach may allow further performance improvement in the future.

Time-series Prediction for Human Motion Prediction (Online): The aim of this test is to further test the subset of algorithms discussed above. To operate efficiently on resource-limited edge devices, where techniques such as quantisation will be employed. The latency reduction and prediction error of the algorithm will be measured using established metrics, namely Total Latency Reduction and Total Prediction Error. The algorithm's performance will be tested under varying network conditions, such as intermittent packet loss, bandwidth limitations, or network congestion.

Results: The algorithms underwent assessment under constrained conditions, specifically with a limited frame rate at the local end (~5fps due to limitations in communications bandwidth of the edge device). Despite this major limitation, the outcomes yielded were reasonably favourable. Full end-to-end testing will comprehensively address latency, network variability and potential packet loss, but remains pending at the present time.

4.9.4 IMOCO4.E Requirements

This section presents selected requirements related to this discussion along with a current update. Please note that the detailed UC3 requirements are provided in selected WP3 deliverables.

Table 41. Requirements for solution.

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
Tools/Toolchains					
Req-D3.2-U3-1-hw-sw-com	Software development using the VectorBlox SDK which will involve research and deployment of CNNs at the local end for the mapping of HMI-IMU and ToF sensor data streams to robot activations at the remote end	I-D	TEST-050	IN PROGRESS	Investigating VectorBlox

	and as required for applied task specific object detections at the remote end.				
Req-D3.2-U3-4-hw-sw-com	Research, engineering and development of an applied sensory fusion algorithm that has functionality to fuse 3D ToF depth imagery (ADI ToF camera), motion tracking sensors (Vive motion trackers) and IMU data (from the Tyndall glove) in order to estimate/predict human arm, wrist and finger movement estimates for translation to robot arm and gripper movements at the remote end.	I-D	TEST-050	IN PROGRESS	Currently focusing on the ADI ToF sensor for PolarFire integration at local end.
Req-D3.2-U3-5-hw-sw-com	Research and investigation into how low-level robot coordinate geometry data can be translated into a sub-set of higher-level gestures and such that AI\ML techniques may be applied at the user local end. As a result, opportunities may exist, such that auto-prediction may be investigated in the context of latency reduction between local user and remote robot ends.	I-D	TEST-050	IN PROGRESS	Currently focusing on hand gesture recognition for latency reduction. Further task-specific knowledge will be required for AI based prediction from hand co-ordinates.
Req-D3.2-U3-7-hw-sw-com	For high-level gestures generated/predicted at the local or remote end, then there is a requirement for the conversion of communicated gestures into a series of robot commands at the remote end for arm movements and precision gripper movements based on the pre-defined industrial task activation steps.	I-D	TEST-050	IN PROGRESS	Ongoing for all partners working on robot geometry algorithm processing.

Req-D3.2-U3-10-hw	Research and development using generally available open-source APIs and SDKs to develop, test and re-engineer (as applicable) detection algorithms for the various 3D ToF camera data used in the use case sensor architecture.	I-D	TEST-050	IN PROGRESS	Ongoing as part of AI/ML algorithms research and development activities.
General					
Req-D3.2-U3-1-hw-sw	Develop a small set of initial test-cases to focus the use case three. For example, tele-operation without digital twin. Two scenarios are described in detail in D3.5.	I-D	TEST-050	IN PROGRESS	Planned for Q2 2024.

4.9.5 Capabilities & Limitations

The unique selling points (USPs) of the system lie in its vision-based gesture recognition employing embedded AI algorithms; Time-of-Flight (ToF) sensors; real-time control of remote robot activations and movements; along with the flexibility to customise gestures and action sequences for diverse applications.

The UC3 strengths include high accuracy in recognising hand gestures, near real-time responsiveness for controlling robot movements, and easy customisation of gestures and action sequences for adaptability to various scenarios. Moreover, precise movement control is ensured through integration with robot actuators and inverse kinematics, while robustness against ToF sensor noise and signal failures is achieved via embedded edge AI and statistical algorithms, enhancing overall system reliability.

The UC3 system also presents a number of identified weaknesses and limitations. The primary limitation lies in the edge device's processing power, leading to lower frame rates and thus increased system latency. Other limitations include reduced accuracy in extreme lighting or high-interference environments, and the need for technical expertise in setup, calibration, and customisation activities.

Additionally, constraints in ToF sensor range and resolution, as well as sensitivity to environmental changes, potentially affecting sensor performance and pose challenges. Other minor limitations include sensitivity to lighting conditions affecting sensor accuracy, interface compatibility constraints with other systems, and potential integration challenges with existing infrastructure or platforms.

4.9.6 Customizations & Adaptations

The gesture recognition system allows the addition of new gestures with minimal data (as few as 10 training samples). These gestures can be associated with new actions or action sequences, and can be adapted to specific scenarios as necessary. The system's versatility allows integration into a range of scenarios, spanning from interactive presentations to various industrial settings.

The current UC3 system uses a Kalman filter to smooth and predict the user's input. This approach is task-agnostic, meaning it will perform well in a variety of situations not specifically covered by this use case. Further prediction improvements may be made by having semantic knowledge of the remote work area. For

example, certain robot actions on the remote end may be augmented with the exact location of the target objects within the workspace.

The research also has potential in other P/D/UCs where vision sensors are used for user guidance of object/robots in the real world. The embedded vision predictors of UC3 with further research and customisation have also got major potential in many forms of healthcare and assistance services.

4.9.7 Methodology & Toolchains

From a hardware perspective, the edge device (PolarFire) establishes connections to a Time-of-Flight (ToF) sensor via IP over USB and to a robot through Ethernet connectivity. The data flow is as follows: Frames are captured at a rate of 10 times per second, optimising for the edge device's data transfer capabilities. Upon capture, each frame undergoes processing to pinpoint the palm's location. Subsequently, hand pose estimation techniques are employed to identify key points of the hand and wrist.

These key points serve as inputs for gesture recognition, executed through the K Nearest Neighbours algorithm (KNN), enabling the system to discern specific gestures such as a thumbs-up or numerical gestures from one to five. Upon recognition, the corresponding action or action sequence is initiated. In instances where a palm is detected but no recognisable gesture is identified, the robot is dynamically moved to a position aligning with the user's wrist location in real-time.

Several key lessons have emerged in the development of this project. Firstly, despite meticulous design and testing of individual components, integration challenges proved inevitable, often presenting substantial time and resource hurdles. Secondly, we discovered that the use of dimensionality reduction when appropriate, significantly minimizes overall processing time.

In our specific UC3 scenario, we expedited this by identifying landmarks in the user's hand/wrist, and leveraging these key points for gesture recognition and robot movement with simple algorithms (KNN). Lastly, we learned that certain machine learning models trained with RGB images can be repurposed for (single-channel) IR images with minor adjustments, albeit with a slight reduction in overall accuracy. These insights have been instrumental in refining our approach and overcoming various obstacles encountered throughout the development process.

4.10 Adaptable and ultra-reliable Time Sensitive Networking to support real-time AI-based acquisition, control and processing (SED)

4.10.1 Technology overview

To meet the objectives of this task, a software component (SW-075) for monitoring and configuring TSN has been improved and developed. This component is intended to acquire and control network data and aims to be compatible with other TSN developments in the project to achieve a more comprehensive network system.

System monitoring must be done remotely and centrally, but to achieve this, a complete structure starting from each of the devices must be established. Figure 139 shows the complete structure of the monitoring tool, including modules and connections for each of the TSN network nodes, as well as for the Central Network Controller (CNC). This task focuses on the main monitoring developments, that are those related

to the TSN nodes. The CNC part, where various services are consolidated to centralize and control information, has been carried out in WP5 and is explained in more detail in D5.8.

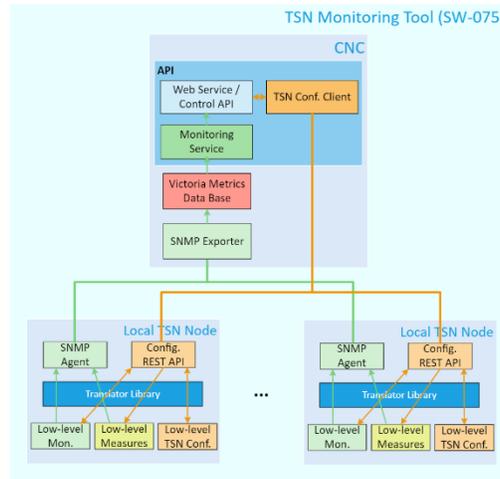


Figure 139. Structure of the TSN monitoring tool.

If we focus on the structure within the TSN nodes of Figure 139, we can see two distinct parts, one marked in green more focused on system monitoring, and another in orange centered on TSN configuration itself. We can consider a third part marked in yellow, composed of a low-level module focused on taking measurements.

Monitoring subsystem

The monitoring system, represented in green in the figure, has two main parts. The first of these is the low-level core of the monitoring subsystem of the node, which is responsible for obtaining information from TSN processes and deriving various parameters from them. It has several probes to monitor the clock's state and, depending on the number of activity cycles and various parameters, it can obtain values such as congestion and bandwidth used by the node.

The other part is an agent that uses the SNMP protocol to obtain all parameters of the node's state along with those obtained with the monitoring core. This agent is responsible for communicating with the SNMP exporter to transmit all the requested information, and with the node's REST API to visualize some of the data locally.

Configuration subsystem

This system, represented in orange in the previous figure, is partially composed of the different configuration cores that make up the TSN subsystem. We have applications, daemons, and APIs for configuring the TSN FPGA cores discussed in section 3.9. In order to extract information from the TSN configuration state, a specific library has been created for each of the parameters, following a hierarchical structure like that used by the SNMP protocol, to use them in interfaces in an analogous way.

The other main part of the subsystem is the node's REST API, from where TSN parameters can be configured. The data is received, and configuration commands are translated to the hardware using the mentioned library. There is also direct communication with the TSN configuration client in the CNC API, as configuration requests and information transmission to the node are performed from there. In Figure 140,

some of the possible configuration requests (POST) or current state queries (GET) in the node's API can be seen, with communication with the CNC being possible through HTTP requests.

Method	Endpoint	Description
GET	/v1/tsn/pps-hwtai-sec	Read Tsn Pps Hwtai Sec
GET	/v1/tsn/mon	Read Tsn Mon
GET	/v1/tsn/mon/{interface}	Read Tsn Mon Interface
GET	/v1/tsn/config-files	Read Tsn Config
GET	/v1/tsn/config-files/{file}	Read Tsn Config
POST	/v1/tsn/vlan	Apply Tsn Interface Vlan
POST	/v1/tsn/tas	Apply Tsn Interface Tas
POST	/v1/tsn/cam	Apply Tsn Interface Cam
POST	/v1/tsn/qci	Apply Tsn Interface Tas

Figure 140. Configuration options of TSN.

External measurements

This part refers to the yellow module in Figure 139. It is a monitoring functionality, but it operates in a completely different way from the rest of the cores. While the others focus on obtaining operational data from the node or simply transmitting its status using SNMP, this measurement core implements functions that allow relating multiple nodes, because latency measurements are distributed and we must correlate between different nodes to get meaningful measurements.

One of the first and most important functionalities of this core is measuring latencies between nodes for each of the traffic flows. For this purpose, the characteristics of the nodes and the flow are provided so that it can identify packets, and using the packet timestamp fields, latency calculations are performed. These calculations are transferred via the API to the CNC, which initiated the measurement request, and it will display a report of the results.

4.10.2 Implementation aspects

The implementation of this component has been carried out on the Xilinx Zynq 7000 family SoC and on the board processor regarding the node, that is on the TSN Z16 platforms. On the other hand, the implementation of the CNC part has been done on a Linux operating system, integrating the different software-created services to have a complete central system. Communications were conducted through the SNMP protocol and HTTP GET and POST requests messages.

First, unit tests were conducted with the TSN nodes, verifying the functionality of the monitoring and the possibility of configurations through its API. Then, the CNC was tested separately with simulated nodes, and once it was operational, both parts were combined to obtain the complete monitoring and configuration system. With the entire system integrated, monitoring tests covering various aspects were performed, including scenarios of saturation and equipment failure, as well as long-duration tests. Regarding the implementation in the pilots, an integration example has already been explained in section 3.9.

4.10.3 Results

During the development of the component, tests were conducted to verify their functionalities as explained previously. The first operational tests on the SW-075 were included in WP6, in D6.4, and the final results will be included in the last deliverable of WP6. These results will encompass laboratory data and other data collected from the successfully conducted integrations of this component.

4.10.4 IMOCO4.E Requirements

The requirements directly related to the SW-075 of the TSN monitoring tool are found in the next table.

Table 42. SW-075 Requirements.

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
R076-D2.3	Time Sensitive Network data plane will expose telemetry monitoring and configuration to a centralised TSN data management.	I		IN PROGRESS	Results to be reported in WP6.
Req-D3.2-B1	Continuous monitoring of the hardware to find faulty behaviors.	D		IN PROGRESS	Results to be reported in WP6.
R021-D5.1-B1	TSN Centralized Network Configuration to facilitate the network configuration and monitoring.	D		IN PROGRESS	Results to be reported in WP6.

4.10.5 Capabilities and Limitations

The main feature of this monitoring component is the variety of functions it offers. Although it is a recently developed system, it already provides several very interesting options for monitoring equipment. The fact that it has a central system with a user interface where all network data can be viewed and configurations can be made is a significant advantage. Additionally, it is a customized tool to extract complex data, as it offers not only the values provided by other tools via SNMP but also network measurements as explained. This, coupled with the unified configuration capability, is one of its strengths. Furthermore, new functionalities will continue to be developed, interfaces will be improved, and more configuration options will be added in the future to make it a complete product.

As for limitations, as mentioned earlier, it is a component in development with many possibilities for improvement, but for now, it includes the functions mentioned above. Another potential disadvantage is that while the CNC could serve any network, the implementations specifically made in the TSN nodes would need to be adapted for other types of equipment, making it not a direct solution that could be used with any network system.

4.10.6 Customizations and Adaptations

This component has been slightly adapted for each integration in this project, both in P1 and P2. However, since the main part used during integration is the user interface tool, the changes have been minor, and its primary use has been to monitor the proper functioning of each node and parameters, to configure the network, and to extract measurements that will be used in the final project results.

The customization possibilities for this monitoring component are lower than for the TSN platform, but this makes it easily reusable in practically any scenario without worrying about configurations. As mentioned earlier, this tool is still in development, so it will continue to be modified and extended with new functions in the future.

4.10.7 Methodology and Toolchains

The toolchain used in this component encompasses a wide range of tools. For the implementation of code within the TSN nodes, Vivado tool has been used for low-level FPGA programming using VHDL language. Additionally, C code has been utilized for testing and programming the device software, while high-level Python code has been employed for automations and function dispatching. For the user interface part, web languages such as HTTP, CSS, and JavaScript have been utilized, and databases like VictoriaMetrics have been programmed. Furthermore, network tools like Iperf for traffic injection and monitoring tools like Pyspark have been employed.

In the development of this component, we had to leverage this diverse set of tools, learn how to integrate them, and ensure that the whole system becomes intuitive and useful for the end users. We have learned that monitoring is a vital aspect of network systems, and providing centralized systems adds significant value to the final products.

4.11 General purpose AI operations speeding-up module, leveraging on FPGA fabric specific resources (SOC-e)

4.11.1 Technology overview

This development HW-017 introduces an AI video analytic application that has been implemented on an edge-computing device. The primary focus of this work is to demonstrate the device's capability to accelerate the inference of AI models and video compression using dedicated hardware. Moreover, the critical information, including the location and size of the objects detected, is transmitted as hard-realtime traffic using deterministic Ethernet. The work carried out by SoC-e within task T3.4, provides an overview of the approach taken, including the hardware and software used, as well as the design flow followed to implement the solution.

Convolutional Neural Network for Real-Time Video Analytics

DNNs are an umbrella term for a bunch of different neural network architectures. One of these architectures is the Convolutional Neural Network (CNN). Since the standardization of machine learning in various industry sectors, much research has been done on this type of network.

The neural networks on the Edge in real-time systems work by processing data instantly using specialized algorithms optimized for speed and efficiency. These neural networks are designed to work on resource-constrained devices, such as mobile phones or embedded systems, achieving a high level of accuracy in detecting the objects they are trained to detect. The main principle behind the detection process is using a CNN trained for classification, to divide an image into quadrants and move along the quadrants to determine the object's position.

According to this principle, different types of neural networks focused on object detection were developed. Among the most important is R-CNN, based on the selective selection of the regions of interest in the image. Another is the SSD network, which extracts feature maps first and then obtains the location of objects. RetinaNet, is a network based on the focal loss to eliminate the imbalance between classes. It assigns different weights in regions where it is difficult to distinguish between the object's classes searched and classes that do not belong to the one searched. Finally, YOLO simultaneously predicts multiple bounding boxes and object class probabilities.

This task proposes the use of YOLO primarily for its speed of execution. In applications that aim to provide real-time road safety enhancement, a fast response speed is imperative to ensure timely detection. At this point, YOLO is a network providing higher velocity. This makes it a suitable option to implement on the Edge. This feature together with a good level of accuracy has introduced this type of network in various systems such as those based on traffic monitoring or people detection in video surveillance systems or even agriculture.

As for the basic operation, it is based on the commented CNN detection principle. The network divides the input image into cells that predict bounding boxes independently of the rest of the cells. These boxes can contain objects matching the searched classes. It calculates the features of each cell to predict both bounding boxes and object classes. YOLO makes use of the Darknet CNN for feature calculation. Darknet is a pre-trained network to facilitate the extraction of useful features for prediction.

Once the features are calculated, the network predicts the bounding boxes and probabilities of the object classes in each cell. These predictions are composed of five values: the (x; y) coordinates referring to the center of the bounding box relative to the cell ends, the height and width relative to the whole image, and the score of containing an object. Before prediction, it removes detections that do not exceed a defined confidence threshold. The final result includes the location of the bounding box, the class, and the probability that the object belonging to that class is correct.

4.11.2 Implementation aspects

SoC Architecture

The combination of AI-FPGA technologies provides a viable option with conditions that enable the efficient performance of real-time applications. A hardware architecture capable of performing the different necessary tasks is developed.

With the focus on the performance of this dedicated system, two key elements make up the hardware architecture of the system. Firstly, Xilinx Deep Learning Processor Unit (DPU) represents a programmable engine dedicated to the acceleration and execution of convolutional neural networks in embedded environments. Secondly, Multiport-TSN Switch (MTSN) IP block performs the transmission of critical information on the size and positioning of the detected objects.

The following figure shows the design of the hardware system architecture describing the communication between the Programmable Logic (PL) and the Processor System (PS). Along with the hardware design, the software tools needed to run the system are implemented. The main elements are described below.

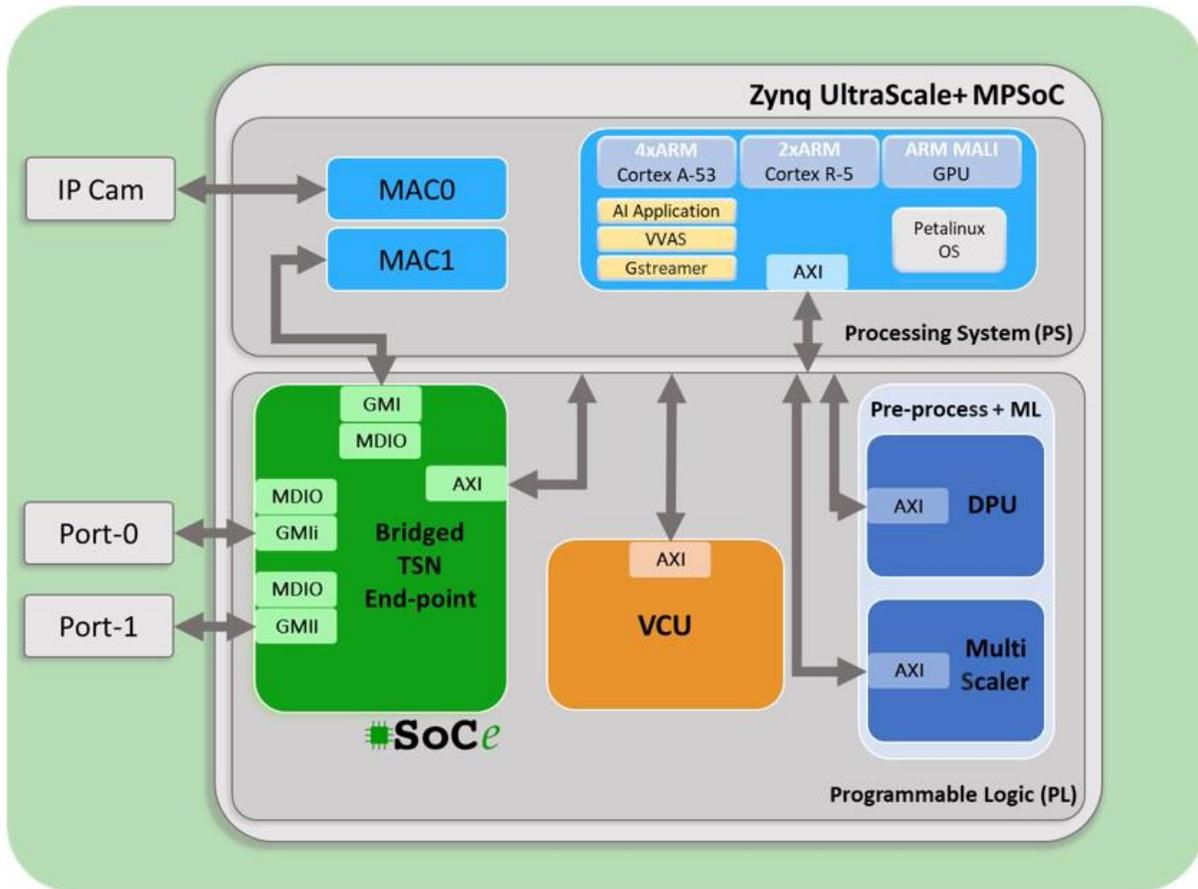


Figure 141. Integration of DPU and Time-Sensitive Networking capabilities in the endpoint.

The PS region contains the Zynq XCZU7EV-2FFVC1156 processor which consists of four ARM Cortex A-53 and two ARM R-5. For Ethernet communication, a Gigabit Ethernet MAC (GEM) and Management Data Input/Output (MDIO) interfaces are enabled using EMIO for connection with PL. The operating system (OS) implemented is Xilinx Petalinux Linux distribution. The application run on this OS.

The software installed on Petalinux consists of libraries for capturing/displaying images through the multimedia framework GStreamer, the Xilinx Vitis Video Analytics SDK (VVAS) solution, the DPU runtime libraries and the user hardware configuration files.

As for the PL region, it is composed of the DPU and MTSN IPs to perform the main functions. In addition, image processing tasks use two core IPs, the Video Codec Unit (VCU) and the Video Multi-Scaler.

The DPU can be configured based on architectures with different levels of parallelism of the convolution unit. The requirements of programmable logic resources to achieve higher or lower performance depend on these architectures. Other characteristics like RAM usage, ReLU type, or average pool are available. Depending on the attributes, they require more or fewer resources.

The IP MTSN is part of the project in critical information traffic via deterministic Ethernet. Designed to support Ethernet transmissions based on the previously explained TSN standard, this switch allows for implementation solutions where flows guarantee bandwidth and deterministic latency. The MTSN integrates 10/100/1000 MACs for PHY interface rates of 10/100 Mbps and 1 Gbps for use with any type of PHY interface (e.g., MII, GMII, RGMII) depending on the application.

The Multi-Scaler IP pre-processes the video frames captured by the IP camera. The target is to process the input data to modify it as required for the model color code, mean and scale specifications.

The VCU core encodes and decodes the video streams in the H.264/H.265 standards. The Zynq MPSoC EV device implements this hardware unit in PL and PS. It has the video codec processing hardware blocks implemented in PL. The video input and output interfaces, such as HDMI, DisplayPort and MIPI-CSI, are implemented in PS.

Finally, an FMC device (VITA57) is included for high-speed serial connections (fiber optic or copper) on the MGT interfaces of the FPGA. To provide the output of the two MTSN ports, use is made of two GMII/SGMII cores located in the MGT region dedicated to the FMC connector of the device generating an SGMII output synchronized with the FMC clock.

The design of the hardware architecture is completed by the clock signals, resets and AXI blocks that make up the final infrastructure.

The proper training of the neural network is crucial for the optimal implementation of the system. The implementation phases include selecting the dataset for the detection of the different objects, modifying the YOLO parameters for the dataset, optimally performing the training and testing, quantifying the neural network for its implementation in the design, and creating the model based on the architecture defined for the system’s IP DPU.

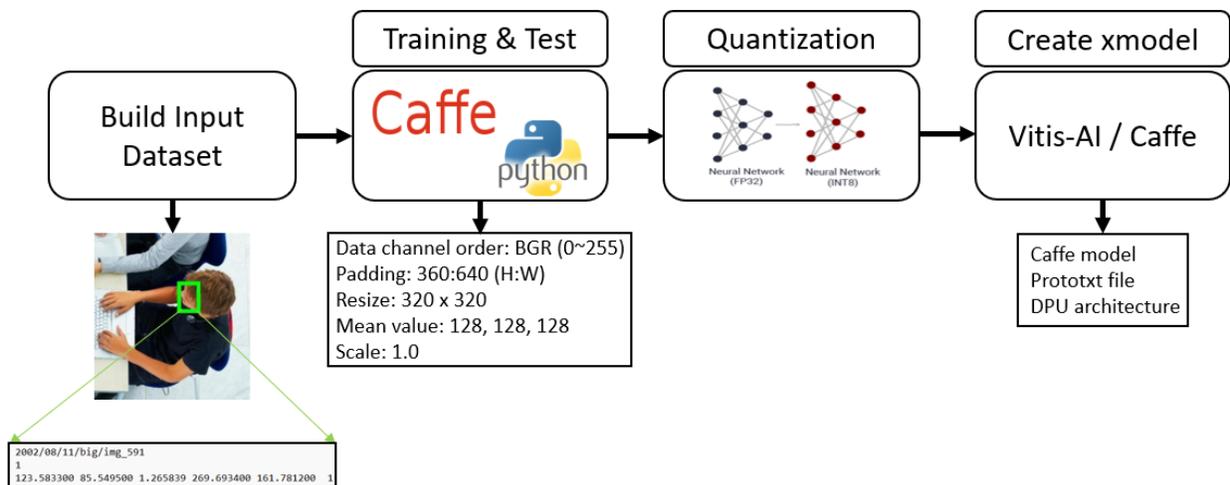


Figure 142. Fully ML Design Flow achieved for YOLO Model.

Data flow

The execution of this system is done using the GStreamer framework for RTSP transmissions. The application launch command runs the video source capture, the Multi-Scaler configuration kernel, the DPU acceleration, the bounding box generation software, the transmission of the image to a VLC server, and the information via serial port. The data flow is summarized in the following figure:

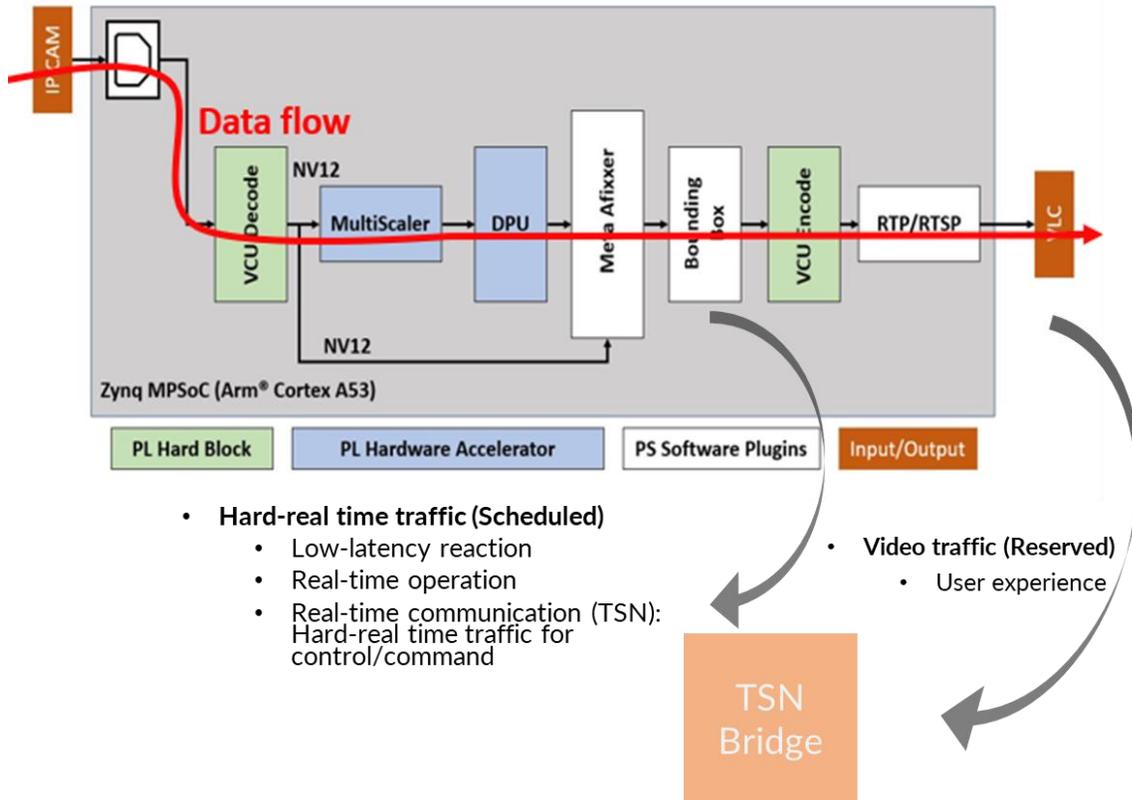


Figure 143. Embedded software for latencies measurements development and for bounding boxes position and size identification.

The launch set-up consists of a switching hub that connects an IP camera, the SoC system, and a VLC server on the host PC. That form the system network, so they must be assigned the corresponding IP addresses (192.168.4.X).

The images enter and leave the system encoded in H.264. The VCU block decodes these frames into raw data, so the Multi-Scaler preprocesses it. The kernel generated for this block transforms from NV12 to BGR format and rescales the image to the size configured for the neural network, 416x416.

Once the images are processed, the DPU block accelerates the neural network that generates the detection by executing the operations included in the `xmodel' file generated in section \ref{SoC Architecture}. This system obtains the precision values of the detection, as well as the vectorial position of the elements included and their size. The next step is to apply the bounding box generation software. These boxes are generated from the vector information obtained by the YOLO network.

The final information is encapsulated in the MTSN block used for Ethernet communication after frame encoding, again through the VCU, and sent to the VLC server for detection result display. The TSN traffic classification includes the following:

- Location and size of the bounding box of the detected object: This is hard-real-time traffic, and it is communicated using a customized NGVA Brake model frame format as Scheduled Traffic. One slot is reserved for this communication on TSN configuration.
- Post-processed HD video image: The video stream that includes the bounding boxes is transmitted to the monitors as Reserved-Traffic. This communication is characterized as soft real-time with a reserved percentage of the data throughput. The latency shall be minimized but is not as critical as the data sent as Scheduled traffic.

- Remaining traffic: The remaining traffic is characterized as Best Effort traffic and shares one slot of the TSN communication with the video traffic.

4.11.3 Results

The implementation of the SoC system has been done in a XCZU7EV-2FFVC1156 AMD-Xilinx MPSOC device. The DPU configuration is 1x 4096 and the system is fed with video images in 1280x720 HD resolution and a framerate of 25 FPS. The resources required for this implementation are summarized in the following table:

Table 43. Logic resourced required for the system implementation.

Resources	Required	Available
CLB LUT	92062	230400
LUT Logic	81536	230400
LUT Memory	10523	101760
Block RAM	243	312
Ultra RAM	47	96
DSP	748	1728
RESOURCES REQUIRED IN A XCZU7EV-2FFVC1156 MPSOC DEVICE 1x 4096 DPU configuration The system is fed with video images in 1280x720 HD resolution and a framerate of 25 FPS		

The most used resource of this system is the Block RAM which is close to 80%. Other configurations show a higher value due to a second DPU usage. That is the limiting resource that prevents the inclusion of more DPU cores in the system.

We have completed the integration in the design an FMC extension board for Fiber Optic/Copper networking:



Figure 144. Integración en el diseño de una tarjeta de ampliación FMC.

This Edge Intelligence system has been evaluated using different DPU architecture configurations for performance analysis in terms of latency and resources used by the system as summarized in the following table. For testing purposes, the system is fed with video images in 1280x720 HD resolution and a framerate of 25 FPS. The DPU architectures used are B4096, B3136, B2304, and B1600. Due to the available resources, the system based on the two architectures with the highest operations per cycle implements a single DPU. B2304 and B1600 architectures system implementation use two cores.

Table 44. Latency results for the Real-Time Video analytics application.

DPU Architecture	Cores	Latency
B4096	1	86 ms
B3136	1	120 ms
B2304	2	153 ms
B1600	2	215 ms

This figure shows the TSN traffic measured in the prototype. The network traffic corresponds to the hard real-time information. In this case, the position and size of the detected objects. The traffic in green is the soft-real real-time one—specifically, the post-processed video sent to the monitors that includes the bounding boxes drawn. In blue, the remaining best-effort traffic is represented. This experiment's traffic is composed of artificially generated noise that overloads the network capacity to value the Quality-of-Service managed by the TSN shapers.

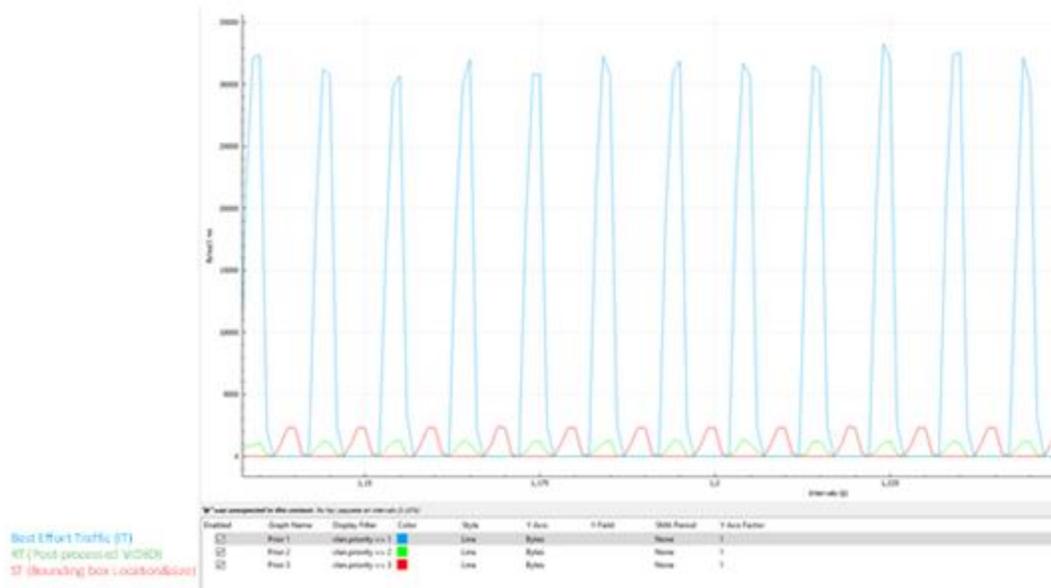


Figure 145. TSN traffic in the concept-proof system.

4.11.4 IMOCO4.E Requirements

This development has been tested against the same requirements as HW-016 also design by SoC-e for this project, as all the functionalities defined here have been tested in the context of the platform developed in T3.3.

Table 45. Requirements for General purpose AI operations speeding-up module.

Req ID	Requirement Description	Verify	Test ID	Result	Comments & Rationales
R1-D4.2-P3	BB4 should provide TSN support	T	TEST-056	PASS	Connection’s bandwidth has been measured among the devices.
R073-D2.3	IEEE 802.1Q Mixed-critical communication support	T	TEST-056	PASS	Several data flows with different priorities haven been sent among devices.
R076-D2.3	Time Sensitive Network data plane will expose telemetry monitoring and configuration to a centralised TSN data management	D	TEST-056	PASS	The configuration and monitoring of SoC-e’s TSN switch has been tested using a remote management protocol
R078-D2.3	Sub-µs time synchronisation based on IEEE 1588 and IEEE 802.1AS	T	TEST-056	PASS	The synchronization among devices has been correctly tested using 802.1AS protocol
R079-D2.3	End to End deterministic latency for time-constrained TSN data streams.	T	TEST-056	PASS	Connection’s latency has been measured among devices.

4.11.5 Capabilities and Limitations

Real-time video analytics based on Artificial Intelligence is an intensive computation. When there is a very short time to obtain results is the key, having the chance to infer the Neural Network models on the edge using hardware accelerators is a value. Additionally, combining this capability with deterministic networking and computation will allow deploying intelligent and trustable systems.

This proposal moves forward the start-of-the-art combining Edge Intelligence and Time-Sensitive Networking. Edge Intelligence concept comprises edge computing hardware and software. Artificial intelligence provides many neural network models and a rich design flow to deploy powerful devices for real-time video analytics. Time-Sensitive Networking is the new Ethernet capable of communicating critical real-time data and a massive best-effort. TSN enables in these systems distributed control and full interoperability among subsystems.

The acceleration obtained inferring the NN model using one DPU on the Edge is approximately 25 times faster than a 4x core ARM CORTEX A53 running a float 32 model and 12 times faster than that CPU running a 8 bit integer model.

As an example, this reduction of the latency time enables a fast detection of the position and size for a given object. This critical information can be transmitted as hard real time traffic through the remaining

subsystems using TSN. The enriched video can be transmitted as reserved traffic through TSN, ensuring not interrupting the video, but with less criticality than the previous traffic.

This proposal utilizes a configurable DPU unit for inferring the neural network (NN). While this solution may not be as resource and performance optimized as implementing a custom hardware model designed specifically for the trained NN, the flexibility of AMD-Xilinx DPUs enables swift adaptation of the system to retrained NNs. Furthermore, it streamlines the process of integrating this acceleration with other subsystems, such as user-defined software.

Regarding the video input, the current implementation receives the stream through a gigabit Ethernet interface connected to an Ethernet MAC implemented on the PS section of the MPSoC. This MAC moves the data through DMA to the external DDR memory. This approach is a drawback if the system requires a video preprocessing operation because the frame is not input directly into the PL section.

4.11.6 Customizations and Adaptations

As stated before, a more optimized implementation of this approach would include a custom DPU unit for a specifically trained NN. This adaptation is significant because the design process's optimization, quantization, pruning, and microcode generation steps would not apply to this custom DPU core.

More practical customizations and adaptations refer to changing the streaming data flow. Using programmable MPSoC allows for defining very different paths. For example, the video input can be set in the programmable section of the device, supporting direct hardware video pre-processing or postprocessing.

In addition, it is worth mentioning that the proposed design is fully scalable. In cases requiring higher computing performance, it is feasible to integrate more DPUs in parallel. Due to the size of the final implementation, it may be necessary to use a more significant FPGA device.

4.11.7 Methodology and Toolchains

To fulfill the main objective of the present research, it is necessary to carry out adequate training of the neural network. The diagram shown in the following figure summarizes the steps involved. Using AMD-Xilinx AI technology, there are two options to implement an AI design. Suppose the Neuronal Network required for your application fits with one of the pre-trained examples provided in the AMD-Xilinx ecosystem. In that case, the implementation effort is significantly reduced, as stated in the figure. However, if a complete training process of the NN is required, it is necessary to follow all the steps identified as the “Full Flow” in the picture. Both approaches have been followed in the research activity carried out in this project. In the full flow, WIDER Face kaggle dataset has been used to train the network.

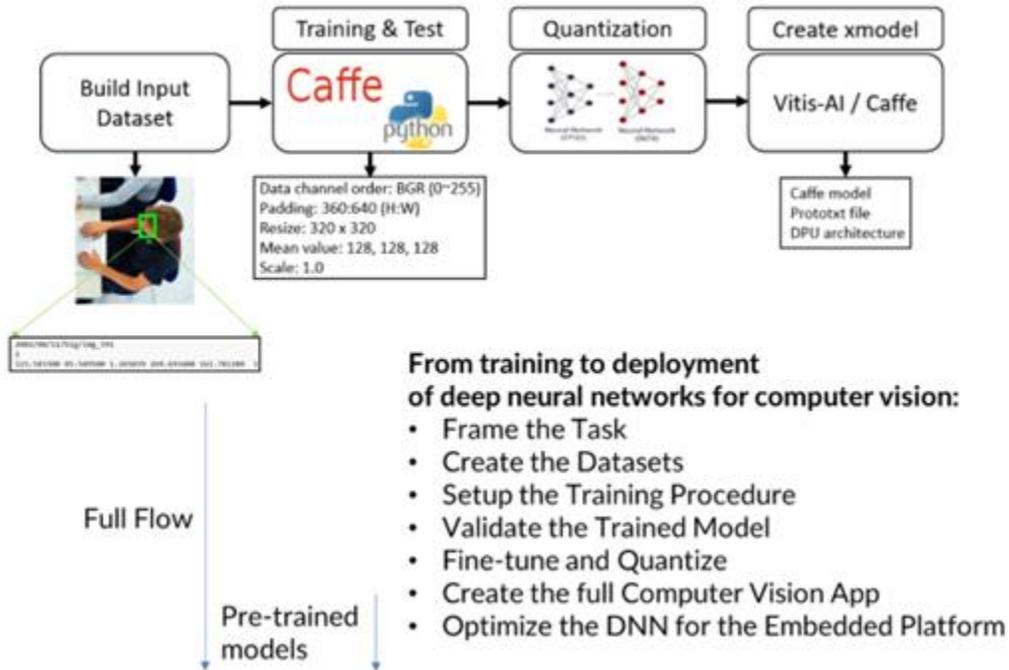


Figure 146. AI complete design flow using Pre-trained models or "Full Flow".

All these steps are carried out by the Vitis AI compiler as shown in the following figure. An environment offered by Xilinx that includes the libraries of the main frameworks for the development of neural networks as well as additional tools for the generation of models implementable on Edge.

AMD-Xilinx AI Framework

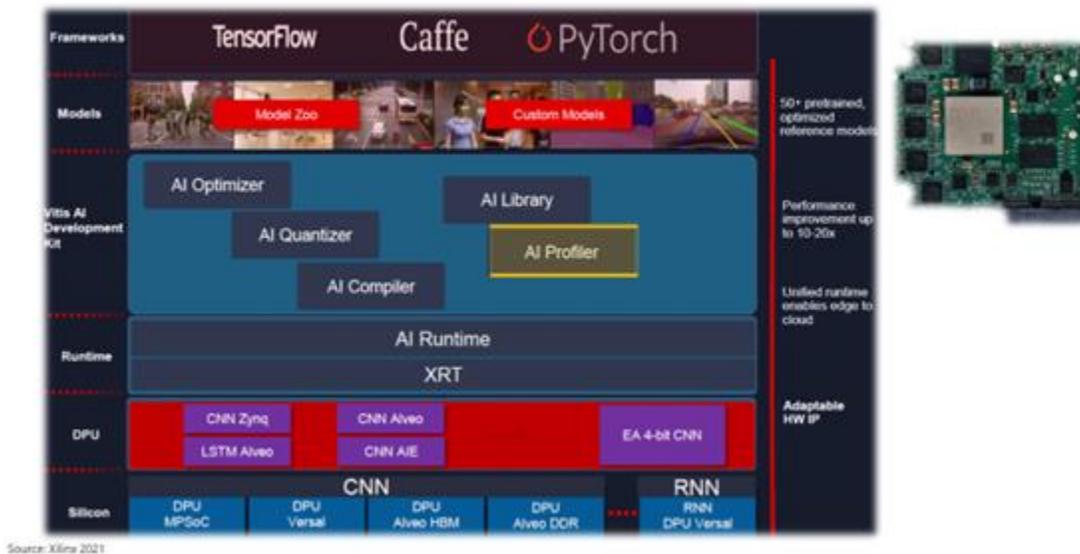


Figure 147. Vitis AI Design Flow.

Firstly, it is necessary to adapt the input data to the network to train the model. For the face shape learning application, each image must provide the vector position of the learning target.

The Densebox network is trained and tested using the Caffe framework for machine learning. The input image size is 320x320 pixels, and the mean and scale value is 128 and 1, respectively. Once tested, it is necessary to quantize the model using integer computational units and the representation of weights and activations by lower bits.

Quantization is necessary as the inference process requires high computational power and bandwidth to achieve optimal latency and throughput values on the Edge. During this step, the 32-bit floating point weights and activation values used during network training are converted to 8-bit integers to reduce the complexity of the requirements while maintaining the accuracy of the model. For this process, the data layers file, called prototxt, and the file with the weights of the pre-trained model, defined by the extension caffemodel, are needed. The Vitis AI compiler uses the files to generate the quantized network description prototxt file and quantized Caffe model parameter caffemodel file.

Finally, the quantified Caffe model, the prototxt file with the network specifications, and the architecture defined for the DPU used to generate the implementable model file through the Vitis environment. The generated output of the flow is a xmodel file that contains the required instructions and network information for the DPU to run the model on the embedded system.

It should be noted that the implemented model is only suitable for this specific application and neural network. For similar applications based on real-time object detection, it is not necessary to change the hardware design, but the implementable model must be regenerated so that these new weights and layers of the network are correctly interpreted by the IP block. In order to do this, it is necessary to repeat what has been discussed above regardless of the AI framework used (Caffe, Tensorflow or PyTorch).

4.11.8 References

- [1] G. Ananthanarayanan, P. Bahl, P. Bod'ik, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, p. 58–67, jan 2017. [Online]. Available: <https://doi.org/10.1109/MC.2017.3641638>
- [2] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, "Live video analytics at scale with approximation and fDelay-Toleranceg," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017, pp. 377–392.
- [3] C.-C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu, P. Bahl, and M. Philipose, "Videoedge: Processing camera streams using hierarchical clusters," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 115–131.
- [4] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2722–2730.
- [5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [6] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.
- [7] S. Shen, M. Sadoughi, M. Li, Z. Wang, and C. Hu, "Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries," *Applied Energy*, vol. 260, p. 114296, 2020.
- [8] S. Setiowati, Zulfanahri, E. L. Franita, and I. Ardiyanto, "A review of optimization method in face recognition: Comparison deep learning and non-deep learning methods," in *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2017, pp.1-6.

- [9] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai et al., “Recent advances in convolutional neural networks,” *Pattern recognition*, vol. 77, pp. 354–377, 2018.
- [10] M. Shin, M. Kim, and D.-S. Kwon, “Baseline cnn structure analysis for facial expression recognition,” in *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*. IEEE, 2016, pp. 724–729.
- [11] D. Palaz, R. Collobert et al., “Analysis of cnn-based speech recognition system using raw speech as input,” *Idiap, Tech. Rep.*, 2015.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
- [14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [15] C.-J. Lin and J.-Y. Jhang, “Intelligent traffic-monitoring system based on yolo and convolutional fuzzy neural networks,” *IEEE Access*, vol. 10, pp. 14 120–14 133, 2022.
- [16] H. H. Nguyen, T. N. Ta, N. C. Nguyen, H. M. Pham, D. M. Nguyen et al., “Yolo based real-time human detection for smart video surveillance at the edge,” in *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*. IEEE, 2021, pp. 439–444.
- [17] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, “Apple detection during different growth stages in orchards using the improved yolo-v3 model,” *Computers and electronics in agriculture*, vol. 157, pp. 417–426, 2019.
- [18] K. P. Seng, P. J. Lee, and L. M. Ang, “Embedded intelligence on fpga: Survey, applications and challenges,” *Electronics*, vol. 10, no. 8, p. 895, 2021.
- [19] IEEE 802.3 ETHERNET WORKING GROUP, “IEEE 802.3 ETHERNET,” <http://www.ieee802.org/3/>, 2018.
- [20] IEEE Time Sensitive Networking Task Group, “IEEE 802.1 Standards,” <http://www.ieee802.org/1/pages/tsn.html>, 2018.
- [21] NORTH ATLANTIC TREATY ORGANIZATION, “AEP-4754 NATO GENERIC VEHICLE ARCHITECTURE (NGVA) FOR LAND SYSTEMS VOLUME III: DATA INFRASTRUCTURE,” <https://www.natogva.org/>, 2018.
- [22] System-on-Chip engineering S.L., “Multiport TSN Switch IP,” <http://soc-e.com/mtsn-multiport-tsn-switch-ip-core/>, 2018.

5 Solutions of Task 3.5: High Performance servo drives, variable speed drives (BB7)

5.1 Overview

5.1.1 Solutions of T3.5

Task 3.5 has two solutions. First is a dc servo drive platform Advanced Motion Controller (AMC) developed by UWB and REX. Second solution is methodology for reduction of electromagnetic interference (EMI) provided by EMCMCC.

5.1.2 Addressed ST objectives and KPIs

Solutions of the task addressed scientific and technological development objective ST2 of IMOCO4.E.

ST2 is focused on developing a smart Instrumentation Layer that gathers and processes visual and/or sensor information from supplementary instrumentation installed on the moving parts of the controlled system. The aim is to enhance the achievable performance and energy efficiency through this approach.

BB7 plays a crucial role in contributing to the goals of ST2 by providing a component for implementing advanced motion control. This component, the AMC servo drive, offers advanced functionality that enables system developers to implement their own smart algorithms directly within the drive's firmware.

BB7 KPIs:

KPI 1

Advanced HW platform for low power DC servo drives. The platform provides HW scalability, low dimensions, rich set of IO connectivity and high performance EtherCat control interface. Intended main field of use is collaborative robotics.

IMOCO4.E target

Scalable servo drive platform for DC drives up to 500W. Rich IO connectivity specified with regard to requirements of collaborative robotics applications. Connectivity for use of additional sensors and controls over provided io and communications without the need of additional HW.

Results

The AMC servo drive offers a hardware platform that meets the specified requirements. During IMOCO4.E, a key issue of discontinued electronic components was successfully addressed by developing a new hardware version with obsolete components replaced. The platform provides a rich set of interfaces for connecting various sensors.

KPI 2

Servo drive firmware for servo drive HW platform supporting IO connectivity, communications with peripherals and motor control algorithms like FOC, BLDC, etc. Tools for configuration and control of the servo drive. User programmable autonomous function of servo drive for selected applications (without the need of upper layer control system).

IMOCO4.E target

Firmware supporting configuration and control of all IO connectivity and communications of the servo drive. FOC algorithm for motor control with focus on collaborative robotics applications. Firmware supporting implementation of user algorithms. Tools for configuration and control of the servo drive with support of all available IO connectivity and communications. Tools for user programming of internal algorithm for standalone applications of the servo drive.

Results

The drive's firmware offers comprehensive support for all available IO connectivity and communication interfaces. Additionally, the drive supports Field Oriented Control (FOC), enabling precise motor control in demanding robotics applications. The control software for the drive facilitates the straightforward definition of user algorithms by utilizing REXYGEN Studio to draw functional schematics using functional blocks. These user-defined algorithms can then be loaded and executed directly within the drive's firmware.

5.2 EMI reduction methods (EMCMCC)

The use of ultra-fast high-performance drives has in common that the switching speed of the PWM stages has to be increased to enable a larger open-loop bandwidth. Due to the closed loop; by sensing the actuator' currents and/or using encoders, and AI at the actuation control loop, the intended motion profile is adhered with high fidelity. All influences like stiffness, friction, variable loads, actuator non-linearity and even wear-out are continuously compensated for by the algorithms developed by the other partners of the IMOCO4.E project.

As such, the relation between the intended motion profile and the 'adjusted' motion 'action' has become a multi-functional relation which is no longer forecastable. As such, the intended velocity, acceleration and jerk (VAJ) profile has lost its one-on-one relation between with what is happening at the drive/ actuator w.r.t. the currents through the actuator, being considered to be the main cause for EMI/ unintended crosstalk

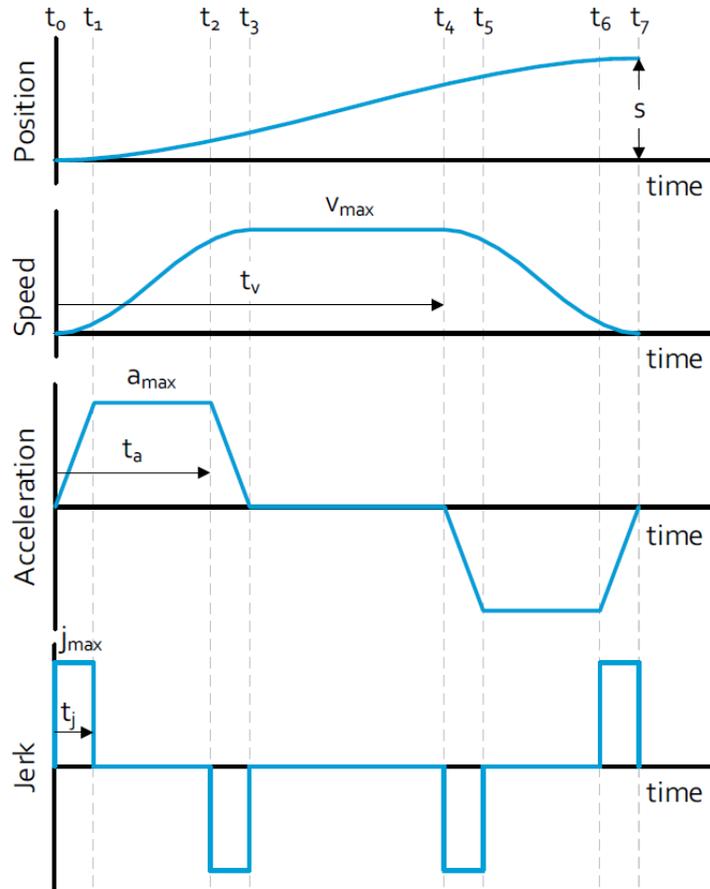


Figure 148. Example of an intended VAJ profile, before local feedback.

What has enhanced is the fact that the PWM drives used with these ultra-fast high-performance drives have definitely moved from the analog to DSP control domain. With most drives, the (X-tal based) clocking used with the DSP is fixed, from which the PWM frequency and duty-cycle profiles are derived.

Similar to digital sound, the DSP will run at several hundreds of MHz, up into the GHz-region, where the audio stream runs at 192 kB/s to enable stereo sound with 22,1 kHz bandwidth, typically with 16 bit going up to 24 bits resolution. The SACD sampling rate is 2822,4 kHz and the resolution is just **one bit**. A stereo SACD recording can stream data at an uncompressed rate of 5,6 Mbps and still provide 22,1 kHz audio bandwidth at 24 bits i.e. more than 128 dB of dynamic resolution (theoretically). Using oversampling up to 128 times the sample's period provides options to trade signal fidelity in resolution and/or bandwidth.

By knowing the one-before last sample change in advance, the change in actuator current between the passed, the actual and the upcoming sample, a 2nd-order moving average prediction can be derived. However, this moving average value needs to be calculated in a fraction of a sample period to be beneficial for compensation.

In addition, the actual actuator common-mode voltages and currents need to be measured accurately, processed w.r.t. the EMI noise elimination required, and injected into the actuator signal interfaces, all taken into account the processing latency as well as the latency (delay) for the signal conditioning, the correction

for frequency dependency of the whole EMI noise correction chain: sensing, conditioning, A/D, digital filtering and processing, D/A, amplification and injection path.

5.2.1 Technology overview

For fixed frequency DC/DC converters, active noise cancelling i.e. reducing modules have become available. Examples are TDK or VICOR of whom the last supports a QPI family of active EMI filters which provide conducted common-mode (CM) and differential-mode (DM) attenuation from 150 kHz to 30 MHz. Their proprietary active-filtering circuit provides superior attenuation at low frequencies intended to support EN Class B conducted RF emission limits. In comparison to passive solutions, the use of active filtering reduces the volume of the common-mode choke, providing a low-profile, surface-mount device. Smaller size filters save valuable board space/ volume, reduce height, power losses and weight.

At the lower (AC-mains) frequencies, active mains harmonic compensation systems exist, e.g. Hyteps, Danfoss, ABB, TDK or YT- Electric (amongst many other brands).

The Active Harmonic Filter (AHF) is a comprehensive solution to the power quality issues of power grid, such as harmonics, reactive power and 3-phase load unbalances. An AHF is in parallel to power grid can detect the harmonic wave in power grid in time, generate the reverse-phase compensation current through the converter and dynamically filter the harmonic wave in power grid.

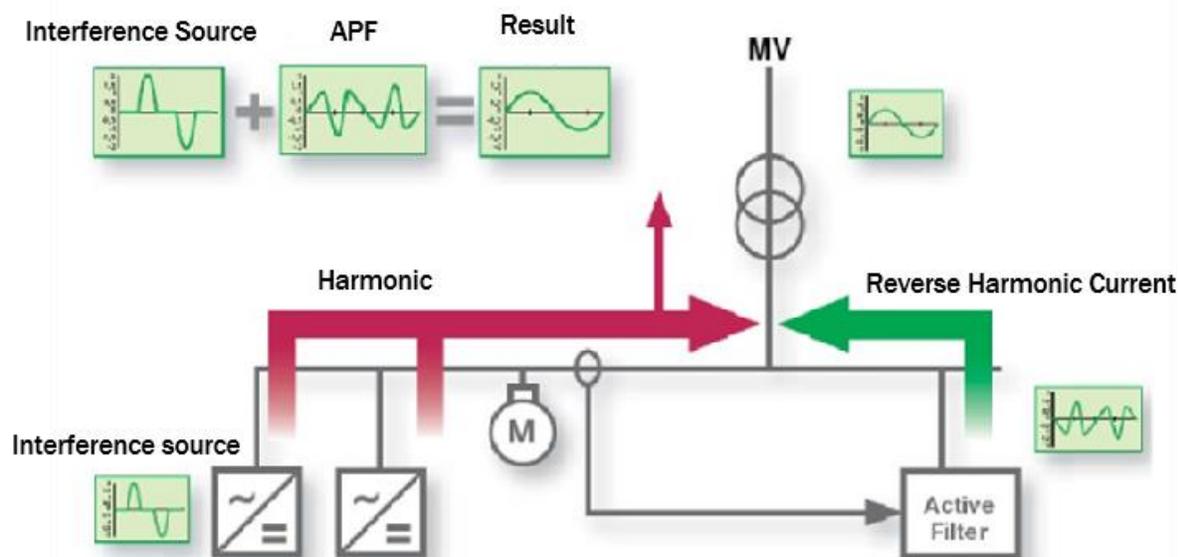


Figure 149. Example of active mains harmonic compensation circuitry.

The operation of an AHF is typically unaffected by multifarious load forms and power grid structures. It will not produce harmonic oscillation (resonance) along with the system, thus perfectly realizing harmonic control under diverse loads. An AHF can fully accomplish dynamic reactive compensation, control the capacitor switching and ultimately heighten the power factor of power grid. Meanwhile, an AHF has the function of controlling the three-phase load current unbalance, thus comprehensively handling various power quality matters in power grid.

Active filtering, to be used at the output of a motion drive towards the actuator, is dependent on a great number of parameters:

- DC-bus voltage and currents, thus motion power
- Switching frequency
- Motion bandwidth required
- Cable length i.e. capacitance, shielded/non-shielded
- Motor impedances, both in common mode to the housing as over the windings

and impacts EMI noise from the application as well as unintended crosstalk to other signals used with the motion system. Furthermore, cable and winding isolation as well as bearing corrosion is affected, when filtering at the output of a drive remains unconsidered.

5.2.2 Implementation aspects

The active adaptive EMI noise cancelling solution has not been realized successfully caused by the constraints given, see section 5.2.1. Instead of that, an effective solution has been found in relatively simple requirements on the interface cable/ wiring between the drive and the actuator. Kirchoff ‘laws need to be obeyed such that:

- All functional signals in-between the drive and the actuator need to be confined on the inner wiring of the cable such that:
- The vectorial sum of all currents at the cross-section of the inner wiring needs to become equal to ‘zero’. Current through the outer shield shall be avoided and may only occur under fault conditions.
- At any cross-section of the drive-to-actuator cable, the vectorial sum of all the voltages on the inner wires needs to be constant (static only) against the outer shield.

N.B. In case no outer shield is used between the drive to actuator cable, no functional current may flow through the motion application’ frame and/or other cables between the actuator and drive.

N.B. In case a shielded cable is used, the cable shield shall be mounted electrically, both to the metal drive housing as well as the metal terminal box of the actuator.

The above implies that all functional as well as stray signals need to be ‘closed’ into their own ‘as small as possible’ loops, back to their sources without unintended bypasses or barriers. It implies that every signal wire/wires need(s) to have its own ‘return’ adjacent AND that, by circuit measures, the sum of currents has to be enforced to become ‘zero’.

However, even differential signals are, by default, not fully complementary. Loop impedance may differ slightly, resulting in different amplitudes of voltages and currents. Even differences in rise and fall times will result in ‘delta’ current in its reference. With PWM drives, the rise and fall times of each stage: high-side and low-side switch, will never be equal and there will be dead-time in the sequence break-before-make. Due to the inductances involved with the actuators, currents will be continuous, clamping the resulting voltages against the crowbars at the drive to the bus-voltages, leading to high dV/dt , followed by di/dt .

With 3-phase systems, using 3 separate shielded cables in-between the drive and the actuator, these can never satisfy the above condition: $\Sigma(i) = 0$, as the main currents flow in-between the inner wires of the 3 separate shielded cables. dV/dt on the actuator windings (= capacitance) against the actuator’s house will cause steep currents as a result, even at low voltages and fast switching, as $I_c = C \cdot dV/dt$.

Example: when 12 Volt is switched in 1 ns across 1 nF, 12 Amps of peak current will result. When confined within a two-wire shielded cable, hardly any crosstalk to other cables will result. When flowing on the exterior of the shielded cable (coax), serious crosstalk will result. Faster switching, e.g. 60 ps has been measured with GaN drives, which will increase the peak currents further.

5.2.3 Results

Based on a low-voltage Nidec drive-motor application, various options for application were analysed. Between best-case versus worst-case applications, all similar functional in their operation, differences can be noted up to 60 dB w.r.t. the externally measurable EMI noise i.e. potential crosstalk.

N.B. During this IMOCO4.E project, this solution has been implemented in an electrical vehicle (Lightyear-One) with in-wheel motors. Between a conventional OEM application and the optimized circuit implementation, also 60 dB of difference (enhancement) had been noted.

With a generic setup, also tests were made using a 2-wire braid- and foil-shielded cable with and without current compensation network. 1 meter of cable is used together with 2 x 22 μH chokes to provide an unbalance-to-balance conversion for the current transferred. The center wire current has been normalized

Measurement: Transmission / Reflection

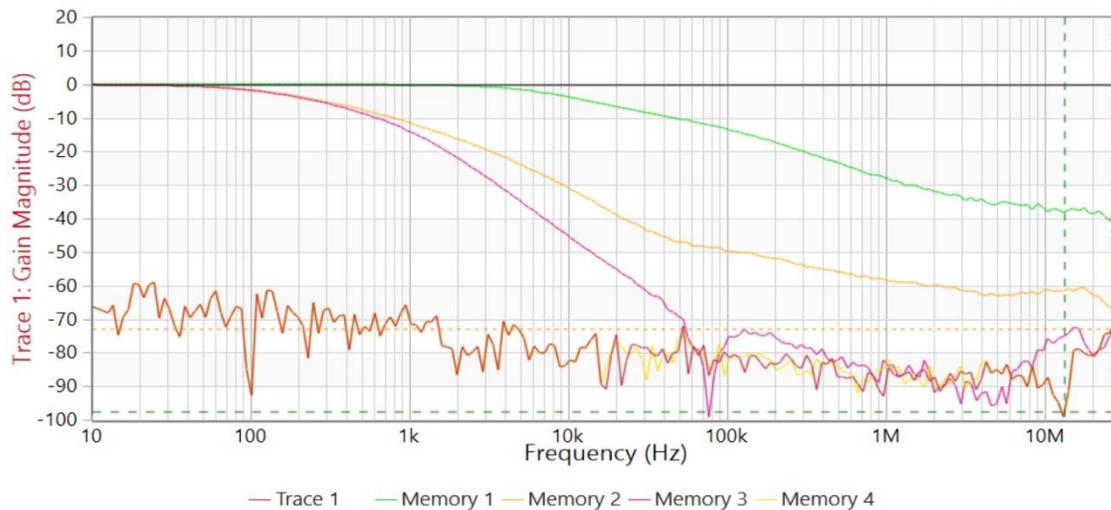


Figure 150. Effect on outer screen current w.r.t. measures taken:

- 0 dB = normalized reference current (inner wire)
- Green = effect of braid/foil shield on cable
- Yellow = effect of braid/foil shield on cable + common-mode chokes 2 x 22 μH
- Pink = effect of braid/foil shield on cable + common-mode chokes 2 x 22 μH + outer choke
- Red = noise floor of measurement system.

When the inductance of the common-mode choke would be enhanced to 700 μH i.s.o. 22 μH, the effect becomes even more prominent, starting at lower frequencies, and more than 60 dB can be achieved, see Figure 150.

Measurement: Transmission / Reflection

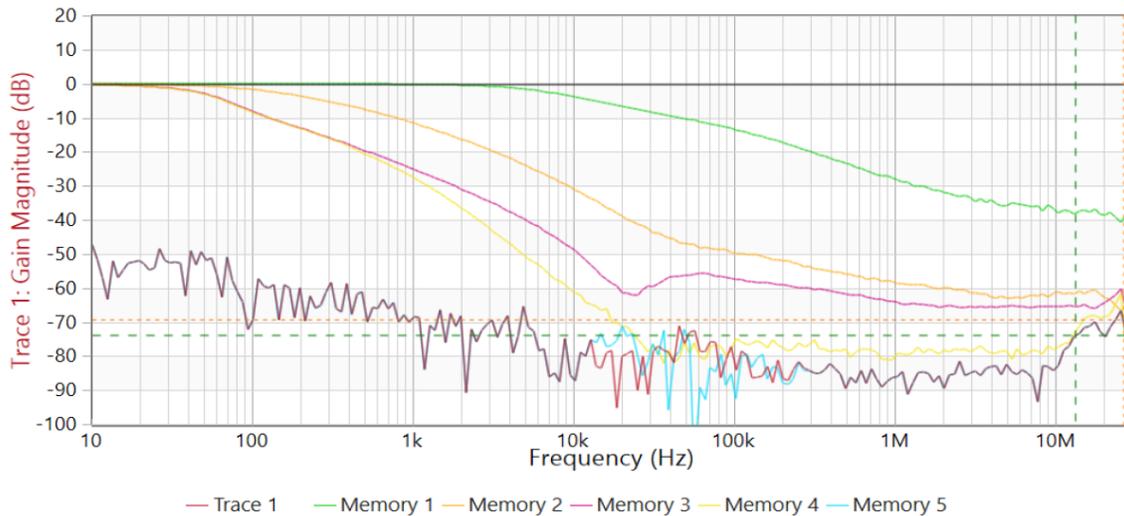


Figure 151. Effect on outer screen current w.r.t. measures taken:

- 0 dB = normalized reference current (inner wire)
- Green = effect of braid/foil shield on cable
- Yellow = effect of braid/foil shield on cable + common-mode chokes 2 x 22 μ H
- Pink = effect of braid/foil shield on cable + common-mode chokes 2 x 700 μ H
- Black = noise floor of measurement system.

For frequencies above 10 kHz, 60 dB of PWM switching noise can be reduced by simple circuit measures, easy to apply with hardly any power losses.

N.B. With longer cable lengths, either higher common-mode inductances are needed, or somewhat less current reduction improvement will be achieved.

5.2.4 IMOCO4.E Requirements

There were no specific requirements w.r.t. EMI noise reduction specified, other than being equal or better than the active noise cancellation developed for stationary cases. The solution found also starts at low frequencies ~ 100 Hz and is useful till 30 MHz. The active noise compensation was limited in bandwidth till about 5 MHz due to the signal fidelity required.

5.2.5 Capabilities and Limitations

The compensation method found is rather compact, doesn't cause additional power loss but does require a different approach w.r.t. cable-connector selection and usage. It is passive and will be based on material selection in combination with the cable length and the drive and actuator impedances involved.

The method provided, being a passive solution, thus reciprocal, is also applicable to enhance the immunity performance of sensor systems and communication paths by taking measures within the wiring of the system.

The solution offered is NOT useful with wireless communication or power transfer systems.

5.2.6 Customizations and Adaptations

For ‘each’ application, the measures need to be optimized to minimize the crosstalk and reduce the EMI emission from the drive/ actuator combinations used.

5.2.7 Methodology and Toolchains

The use of unbalance-to-balanced signalling is known from CCTV applications with analogue signalling and long before that with RF antenna techniques. With the application to smart drive systems, the solution is new and very efficient (when properly dimensioned). The addition of a ferrite ring over the shielded motor cable (as provided as a solution by a great number of drive manufacturers) is orders of magnitude less efficient as the one offered.

5.3 AMC - Advanced Motion Controller (REX CONTROLS)

5.3.1 Technology overview

AMC (Advanced Motion Controller, Figure 152) is a DC servo drive with EtherCAT® communication, designed to provide compact dimensions, rich set of IO connectivity, high computing power and allow control of brushed and brushless motors with power up to 500W.



Figure 152. AMC servo drive.

The drive was designed at New Technologies for the Information Society, Research Centre of the Faculty of Applied Sciences, University of West Bohemia. The device was licensed by REX Controls s.r.o. with intention of further development of required functionality and commercialization. REX Controls s.r.o. has implemented support for its REXYGEN system, which provides drive’s user the ability to define custom algorithms running directly within the drive.

There are two main target application areas, which helped to define the drive's features and parameters.

The drive prioritizes research applications, offering a robust system for measuring and visualizing internal variables and signals. Users can select specific signals or variables for real-time monitoring, providing valuable insights into algorithms under development and research. This functionality enhances the drive's value for research activities, empowering researchers with powerful tools for analysis and experimentation.

The second target application area is robotics. The drive IO connectivity and mechanical design was proposed based on actual requests from our robotics applications.

The drive is built around Infineon XMC4800 MCU with integrated EtherCAT®

Main parameters of the drive:

- *MCU core* ARM Cortex-M4 @ 144 MHz
- *Supply voltage* 8 – 48V DC
- *Continuous current* 10A
- *Peak Current* 25A
- *PWM switching frequency* 24kHz
- *Supported motor types* DC brushed and brushless
- *Supported feedback* Digital incremental encoder, Encoder with Biss-C, Digital halls
- *IO* 2x incremental encoder, 2x Biss-C/SSI, CAN, RS422, UART/SPI, I2C, 2x AIN, 1x AOUT, 3x DIN, 2x DOUT, BRAKE, STO, EtherCAT®
- *Loop EtherCAT®* 1 kHz
- *Loop speed / position* 8 kHz
- *Loop current* 24 kHz
- *Current measurement* Sigma-delta 16 bit / 24 kHz
- *Dimensions L x W x H* 45 x 35 x 25 mm

The hardware architecture of the drive is shown in Figure 153.

Internal power supplies are highlighted with red color. Input voltage, with operating range from 7V to 48V, is passed to two step-down switching converters, which both output voltage with 5V level. First converter is dedicated to powering the CPU and internal circuits. Voltage with level 5V is passed to another step-down switching converter making 3V3 voltage level, which is required by CPU and other circuits. The second converter makes voltage with 5V level, which is used for selected IO connections and for powering position encoders, etc.

Another step-up switching converter which provides 15V voltage level for powering analog output circuit. There is another step-up switching converter which provides 15V voltage level for powering MOSFET driver circuit. Current measurement circuit is present for two of three phases. Phase current measurement circuit uses galvanically isolated Sigma-Delta AD converters. The ground level for power supply of the AD converter floats on the potential of the phase. Therefore, the supply voltage for the AD converter must float on the phase potential. This is achieved by using a step-down switching converter providing voltage with 5V level (relative to the phase potential).

As phase current measurement is present for two phases, this results in 7 switching power converters used in the drive.

Blocks highlighted with violet color are related to the power output of the drive. It also includes Safe Torque Off related circuit. The power output of the drive is realized by three half-bridge legs, implemented by two N-MOSFET transistors for each phase and a MOSFET gate driver.

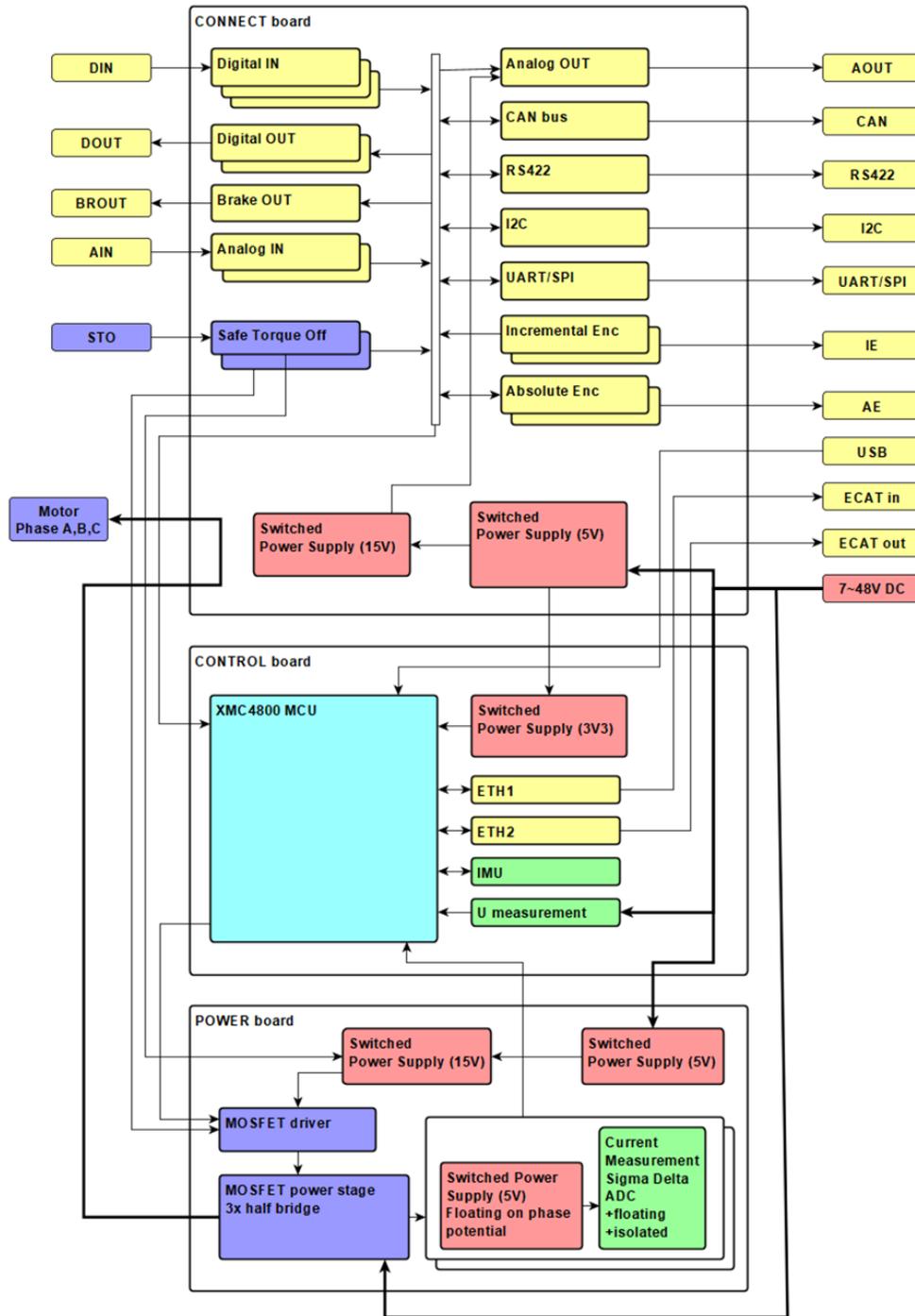


Figure 153: HW block diagram

Safe Torque Off functionality is implemented by two hardware channels, where each channel directly and independently on the CPU, controls physical disabling of phase output. The STO functionality is implemented by two independent HW methods, where first method disables the MOSFET driver, and second method disables the supply voltage for the MOSFET driver.

Green color in the block schematics highlights internal measurements. This includes measurement of motor phase current, supply voltage level and inertial measurement information. The drive features an internal 9-axis Inertial Measurement Unit. In robotics applications, where the drive is placed in moving parts (joints) of the robot, data gathered by the IMU can be used to provide data about the space position of the drive or gathering data for vibration analysis.

Yellow color highlighted blocks cover all inputs, outputs and communications. Key part is an EtherCAT® communication. The XMC4800 microcontroller provides dedicated hardware support for EtherCAT® communication including dedicated Ethernet MAC and DMA channels, which are necessary to offload the CPU load related to communication data manipulation.

The IO connectivity set was proposed based on our experiences and requirements from previous robotics projects. The drive IO connectivity can reduce the wiring requirements to attach various sensors and actuators. Such sensors and actuators can be connected as drive peripherals and data related to those peripherals can be transferred through EtherCAT®.

The drive provides general digital inputs and outputs, analog inputs, and output. Dedicated digital output to control motor brake is required for robotics applications, where inactive robotic manipulator must be mechanically stable. Communications like CAN, RS422, I2C, UART/SPI can be used to connect various peripherals. Use-case of some of those IO connectivity will be described in the section about our demonstrator, 7 joints collaborative robotic manipulator, which use 3D-mouse as control peripheral together with button for control and beeper for user feedback.

The IO connectivity also covers position encoders. The drive allows connection of up to 2 incremental encoders and 2 absolute encoders. Absolute encoders with Biss-C or SSI communication are supported.

USB communication provides high speed data transmission between the drive and PC control application.

Figure 154 summarizes main tasks performed by drive firmware.

The drive firmware executes various tasks covering communication, state handling, measurements, regulation, motor control, signal generation and other areas.

Most of the tasks performed by the drive are synchronized to PWM frequency of signals generated for power output stage. The key part of this synchronization is implemented by configuration of three timers generating PWM signals for the power stage. These timers are configured and synchronously started, providing PWM signals with frequency of 24 kHz. Most of the other tasks, performed by the drive firmware, are synchronized to these timers. These timers will be referenced in further text as "PWM timer".

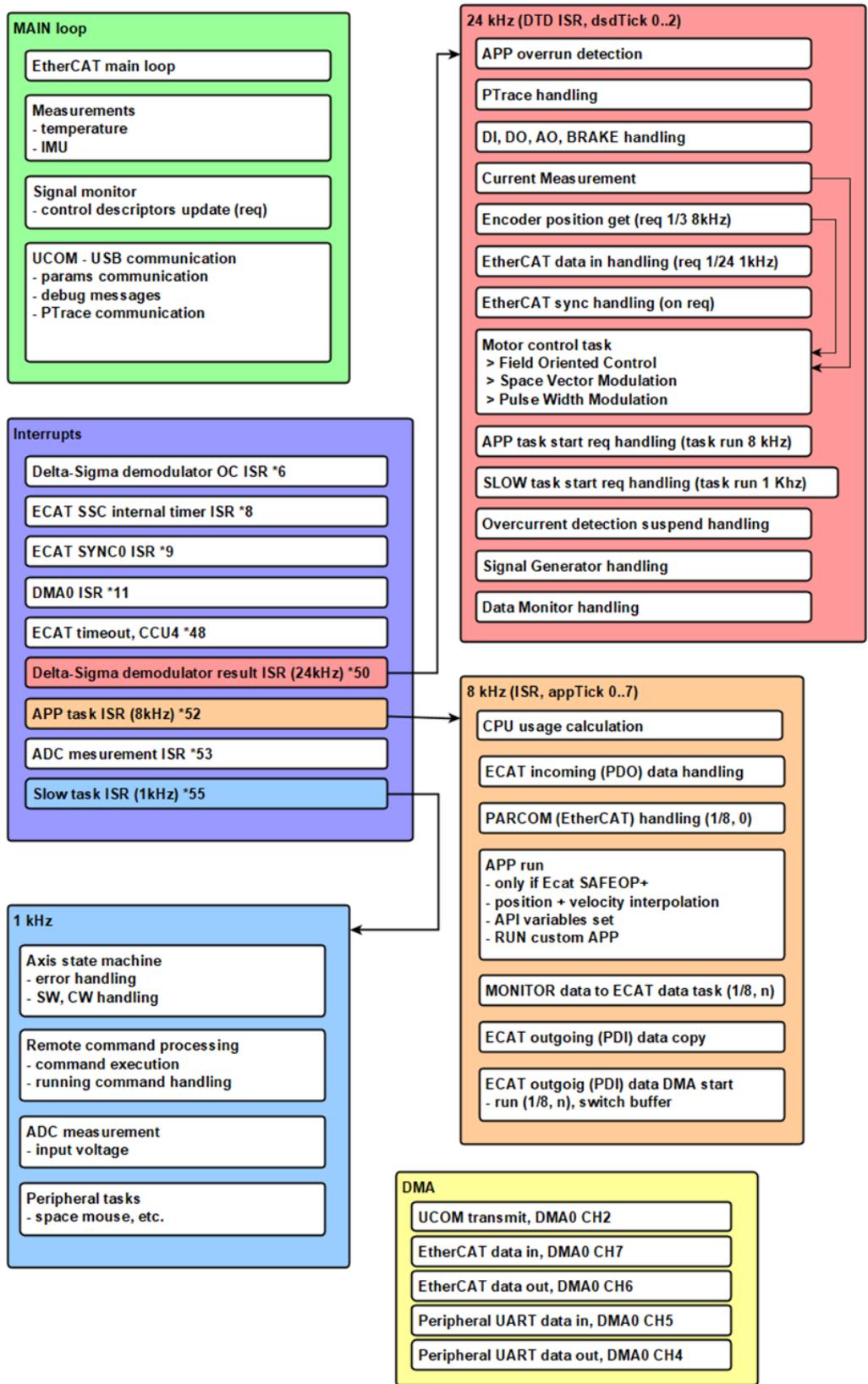


Figure 154. Firmware main tasks.

Figure 154 shows a list of significant tasks, grouped together by their execution instant.

Main loop (green color) lists tasks running in main loop using remaining time of the MCU.

Those tasks are called non-synchronously to PWM timer, although some may synchronize internally to various firmware events. Tasks are implemented to last reasonable short time intervals to allow execution of all main loop tasks in sufficient intervals.

The violet-coloured tasks are executed through interrupts with varying priorities. These interrupts are prioritized, with the higher the interrupt in the list, the higher its priority. Each interrupt's priority is indicated with an asterisk symbol followed by a number, where *9 indicates priority 9. The lower the number, the higher the interrupt priority.

Most interrupts are synchronized to PWM timer, either directly or indirectly.

Directly synchronized interrupt is “Delta-Sigma demodulator result ISR” (red colour) occurring when current measurement Sigma-Delta AD conversion result is available, which is always at the same position in the PWM timer cycle.

Current measurement performance has a direct impact on quality of motor control. Implemented method of current measurement uses isolated AD converter measuring voltage drop on shunt resistor. AD converter provides Sigma-Delta modulated digital stream, which is decoded by XMC4800 Sigma-Delta demodulator hardware.

Current measurement is synchronous to PWM signal, starting based on compare value match on PWM timer and lasting defined part of PWM period. The compare value and measurement time is properly defined to provide average current over whole PWM period, with small time intervals around switching moments omitted. The end of current measurement generates the interrupt.

Interrupt with level 6, the highest priority is "Delta-Sigma demodulator OC ISR". This interrupt is generated by the overcurrent detection feature, which uses the same Delta-Sigma signal from the AD converter as other tasks, but with different demodulator settings to provide fast and accurate current measurements at a result frequency of 3MHz. The instantaneous current is compared with pre-defined levels by the hardware, and when the current exceeds the specified value, the overcurrent interrupt is generated.

Interrupt with priority level 8 is handled by EtherCAT® stack and is related to internal 1ms timer used by the stack.

Interrupt with priority level 9 is handled by EtherCAT® stack and is triggered by SYNC0 EtherCAT® event.

Interrupt with priority level 11 is related to DMA transfers, call user defined callbacks and transfer specific handles events.

Interrupt with priority level 48 is called, when no EtherCAT® packet is received over defined time. This provides detection of communication failure and switches the drive into error state when EtherCAT® communication failure occurs.

Current measurement interrupt with priority level 50 is called during each PWM cycle at a frequency of 24 kHz. Important tasks are performed within this interrupt, including:

- Application overrun detection. As the application may be defined by the user, too complex application execution may exceed the available execution time for the application. If an application overrun is detected, then the drive enters an appropriate error state. If the application overrun is enabled as a PTrace (Process Trace) trigger, then relevant PTrace actions are performed.
- Handling of digital and analogue inputs and outputs and brake output.
- Calculation of measured current with respect to AD converter offset and gain for each channel.
- Conditional reading of encoder position.
- Start of DMA transfer of incoming EtherCAT® process data, input buffer switching (executed only if new EtherCAT® data were received).
- EtherCAT® synchronization, alters PWM cycle to maintain synchronicity to EtherCAT® (only if enabled).
- Motor control task, for Field Oriented Control this covers FOC, SVM (Space Vector Modulation), and PWM update request.
- Request for application task start (set request for application interrupt, execute on each 3rd cycle, resulting in an 8 kHz application rate).
- Request for slow task start (set request for application interrupt, execute on each 24th cycle, resulting in a 1 kHz slow task rate).
- Handling of overcurrent detection disable requests.
- Handling of the internal Signal Generator feature.
- Handling of the internal Signal Monitor feature.

Application task interrupt with priority level 52 covers following tasks:

- Calculation of CPU utilization
- Incoming EtherCAT® process data handling (copy data to destination variables)
- Handling of cyclic communication of drive parameters through EtherCAT®
- Application execution, interpolation of velocity and position requests
- Signal Monitor gathered data processing (copy to EtherCAT® frame buffer)
- Outgoing EtherCAT® parameters data processing (copy to EtherCAT® frame buffer)
- Start DMA transfer for outgoing EtherCAT® frame (every 8th cycle)

The interrupt with priority level 53 handles the ADC conversion results for measuring the supply voltage and internal drive voltage.

The slow task interrupt with priority level 55 covers several tasks including the state machine for the drive and axis, error handling and timing, processing of Control Word and Status Word, execution of user commands, triggering of ADC measurement, and handling of connected peripherals such as the Space Mouse.

To improve the efficiency of various data transmissions, DMA transfers are used. The DMA transfers, shown in yellow in Figure 154, are used for transmitting data streams over USB, for both directions of EtherCAT® data transfers, and for both directions of peripheral data communication over UART.

Typical control application is shown in Figure 155.

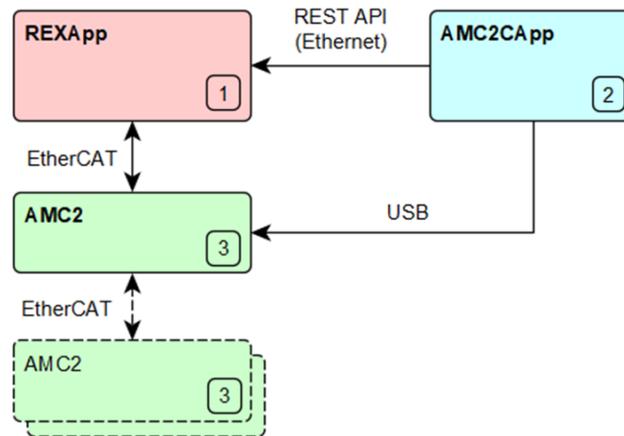


Figure 155. Typical control application.

The application allows setup, management, monitoring, and tuning of drive internal parameters and parameters of user-defined applications.

The drive firmware defines parameters that are published to the control application. Parameters can be writeable and persistent. A writeable parameter value can be modified from the control application, while a persistent parameter retains its value between restarts and power cycles.

The application can communicate with the servo drive by two methods. The first communication method is directly through USB. The second method is through REST API to the REXYGEN application, which communicates with the drive through EtherCAT®. The second method allows communication with any drive connected to EtherCAT®.

The application also shows debug messages from the drive, provides tools like “Signal Generator”, “Signal Monitor”, “Process Trace”, “User Script”, “Firmware Update” and “MicroREX custom application”.

The "Signal Generator" feature (shown in Figure 156) can generate a defined signal (square, sinus, pseudo-random) and inject it directly into an injection point in the drive firmware. The injection point can be either output voltage, request for current regulator, request for velocity regulator, or request for position regulator. This allows direct tuning of each regulator with real-time assessment of tested parameters.

Another powerful tool is the "Signal Monitor" feature (shown in Figure 157 and Figure 158). It allows real-time visualization of specified parameters in time charts. Any parameter provided by the drive firmware can be visualized. The sample rate for visualization is selectable, and the full sample rate of 24 kHz is available.

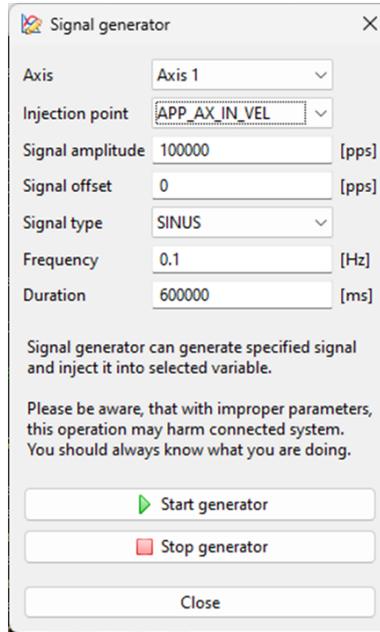


Figure 156. Signal generator.

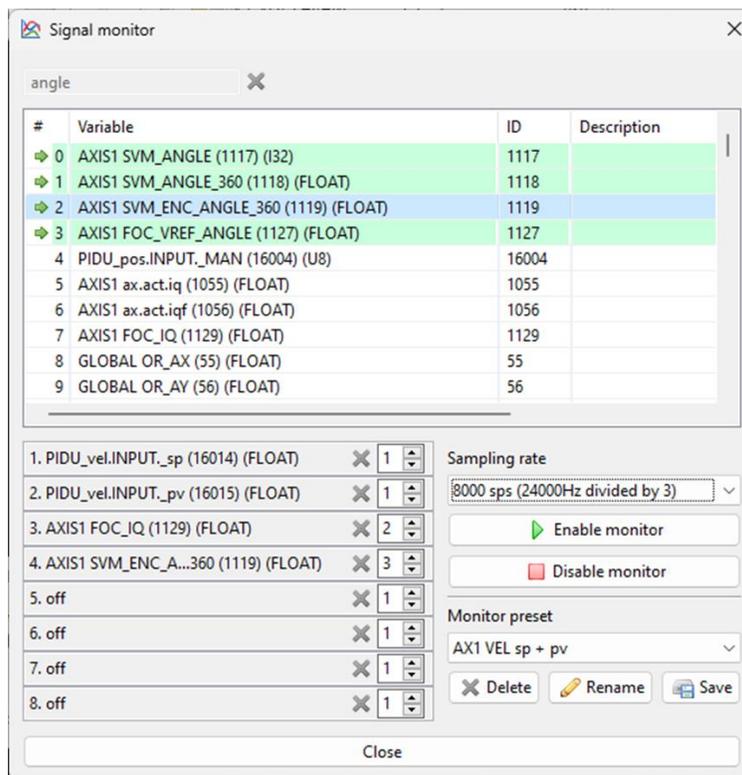


Figure 157. Signal monitor.

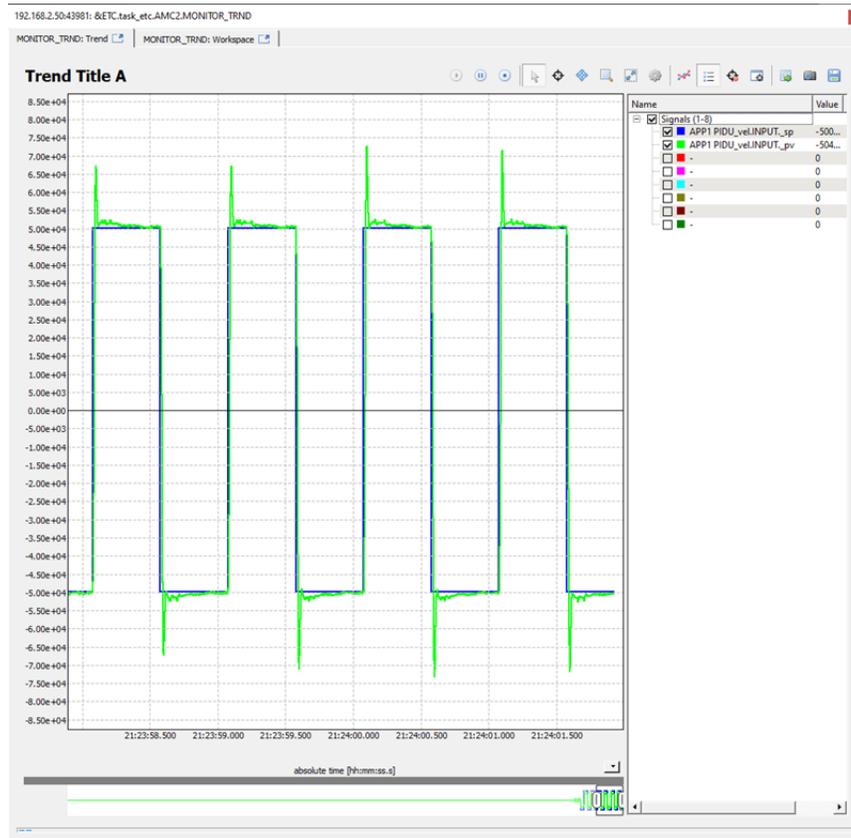


Figure 158. Signal monitor chart.

The "Process Trace" feature (shown in Figure 159) provides insight into the drive's internal processes and is mainly intended for tuning the real-time performance of the drive. It can visualize the timing of specified internal processes. The start of taking the snapshot is either user-triggered, or it's possible to define a trigger on variable value or other conditions to take a snapshot of the timing at the exact point of interest.

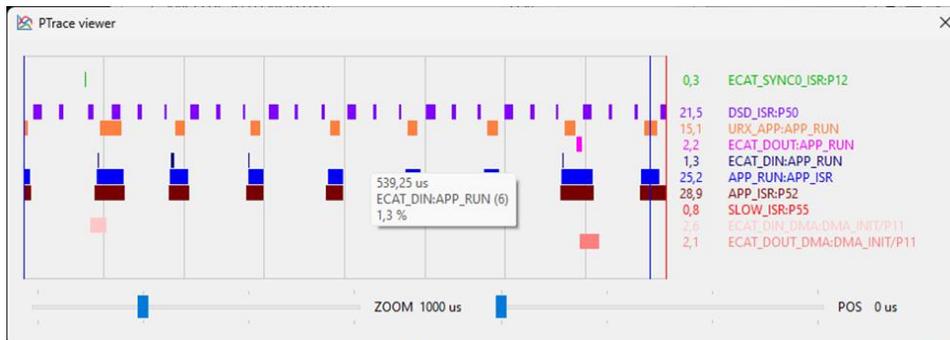


Figure 159. Process trace.

During testing or development, it's necessary to set certain parameters to specific values and frequently repeat this procedure. To streamline and simplify such a process, the control application provides a "User script" feature. This feature allows the user to define a named custom script that can execute the required command sequence and be stored for later use.

The control application allows for updating the drive firmware through the "Firmware Update" feature. This function can provide updated core firmware, but it also provides a way to load a user application into the drive.

The "MicroREX custom application" feature (shown in Figure 160) allows the user to define a custom application graphically. The custom application is defined using REXYGEN Studio to draw the application schematics. This application is then deployed directly into the drive through the "Firmware Update" feature.

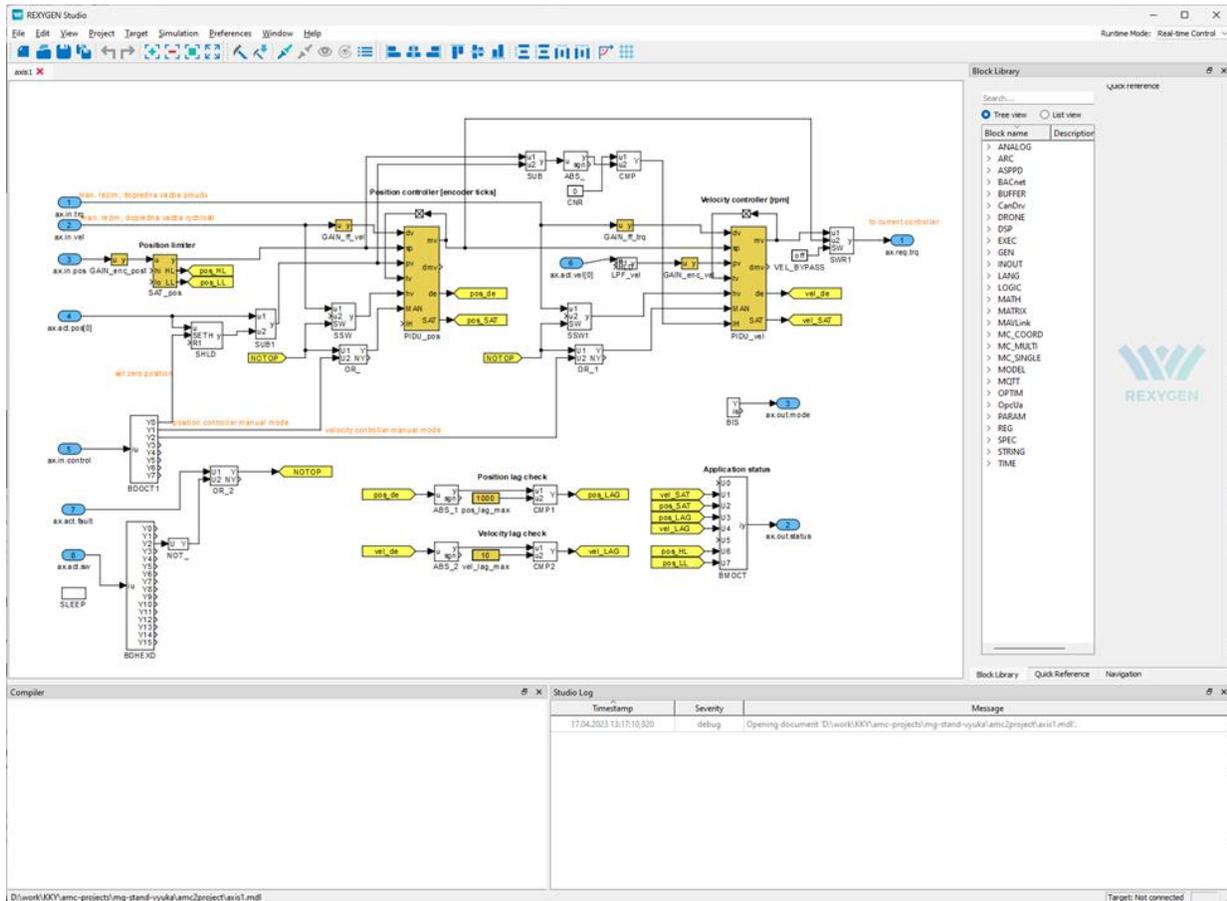


Figure 160. MicroREX application in REXYGEN studio.

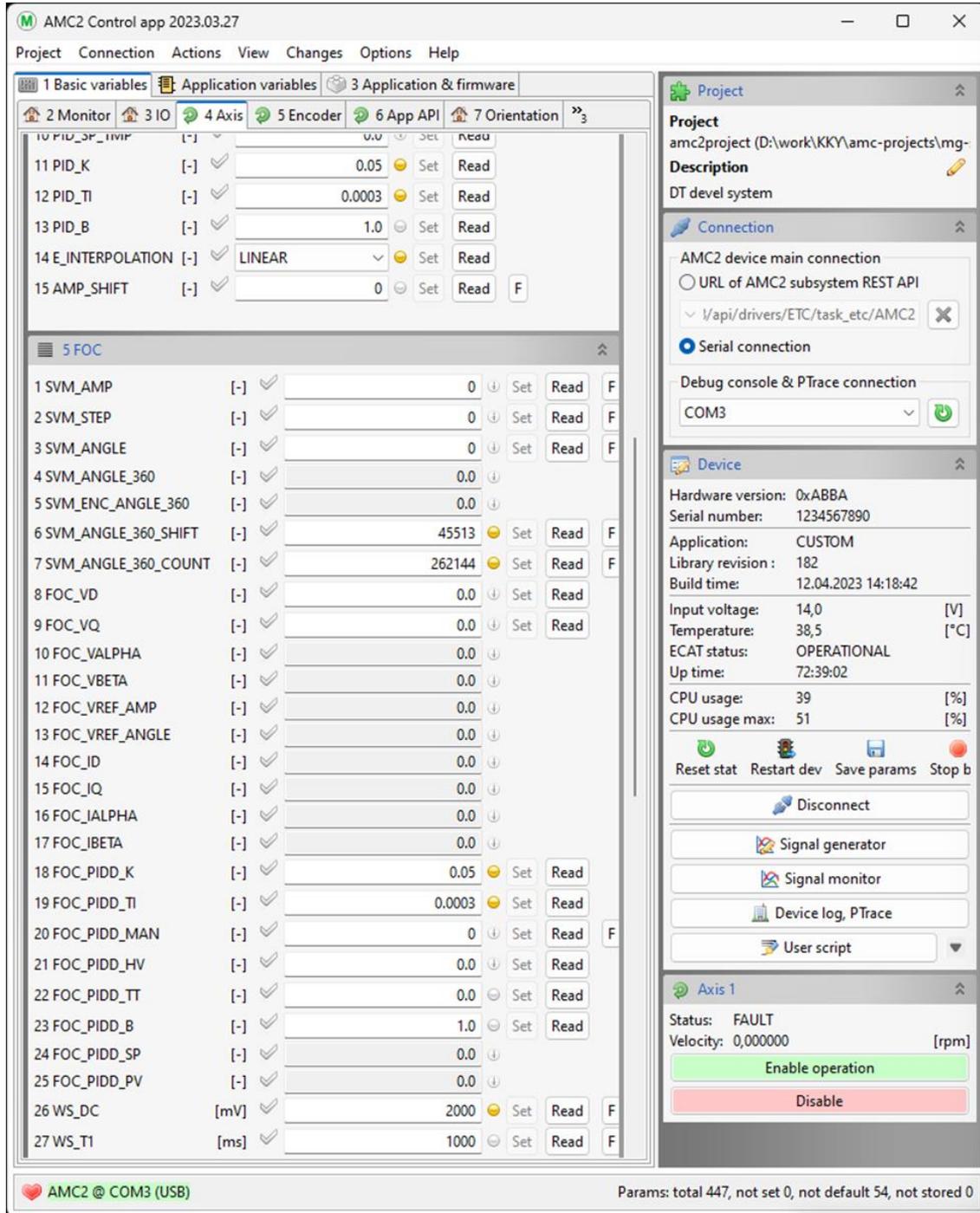


Figure 161: Configuration and control software

This feature, which allows the user to define a custom application running directly in the drive, fulfills requirement R179-D2.3. This feature is a crucial part of defining the test setup for other IMOCO4.E targets, such as research on smart control algorithms.

Figure 161 show main windows of drive configuration and control software.

Link to Hardware- and Software-Catalogue and Components-Integration

HW-Catalogue links: HW-020 – AMC servo drive

SW-Catalogue links: none

5.3.2 Implementation aspects

The servo drive firmware and support tools were extended to allow users the ability to define custom algorithms to run directly within the drive. The implementation of this functionality is based on three key components: the drive firmware, the drive control program, and REXYGEN Studio.

The user begins by creating their own algorithm, defining it graphically in REXYGEN Studio. This algorithm utilizes an API designed to facilitate interaction with the drive firmware. The API includes connection points between the drive firmware and the user application in multiple areas.

The first area comprises input variables, which include:

- in.trq: torque request
- in.vel: velocity request
- in.pos: position request
- in.mode: variable defined to set the mode of the user application
- in.control: variable defined to control the user application
- in.gpv1: general-purpose input variable

The next area encompasses state variables:

- act.sw: status word
- act.cw: control word
- act.fault: drive fault status
- act.iq: actual current
- act.iqf: filtered actual current
- act.vel[3]: actual velocity for encoders 1, 2, and 3
- act.pos[3]: actual position for encoders 1, 2, and 3

Lastly, we have requests for the drive firmware:

- req.trq: torque (current) request
- req.fault: application fault, allowing the application to shut down the drive

The user's application interacts with the API to execute user-defined functions. The basic application includes velocity and position regulators, enabling velocity or position control of the connected actuator.

The user defines their application graphically by connecting and parameterizing functional blocks. Available functional blocks cover several areas, including ANALOG, LOGIC, MATH, REG and more. Examples of functional blocks include ANALOG/ang for calculating average values, LOGIC/OR for logical sums, and REG/pidu implementing a PID regulator.

Detailed documentation and descriptions of all functional blocks can be found on the REXYGEN website.

The user's algorithm, defined in REXYGEN Studio, is then directly incorporated into the drive's firmware. This integration is achieved within the control application through three simple steps.

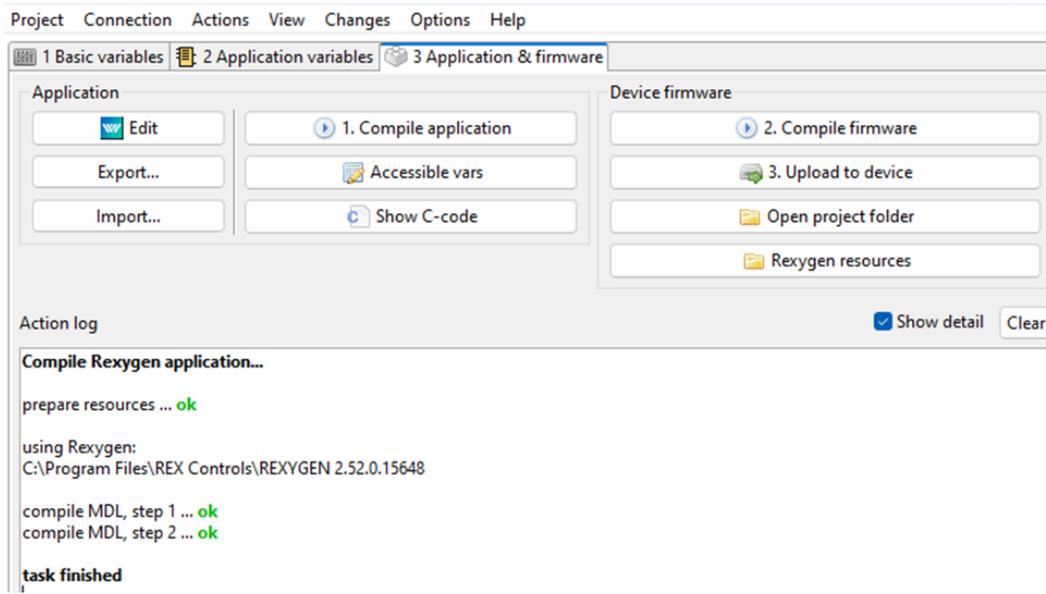


Figure 162: UI related to custom application

The first step involves compiling the user's application into programmatic code.

The second step involves creating new firmware for the drive by linking the user application with the drive's basic firmware.

The final step is to upload the new firmware into the drive. Throughout this process, users can also specify which user application variables should be visible and accessible at runtime.

The entire process, from defining the algorithm to uploading it to the drive, is very intuitive and simple. It can be accomplished with just a few clicks of the mouse in the drive control application.

The UI associated with definition of user application is show on Figure 162.

The functionality was implemented during the project and was utilized in the development and verification of Smart Control algorithms (BB5).

5.3.3 Results

The AMC servo drive and user application capabilities were used in research of Smart Control Algorithms (BB5). A test stand for repetitive control and other smart control algorithms is depicted in Figure 163. This setup allows for the evaluation and testing of various control algorithms, including repetitive control and other smart control strategies.

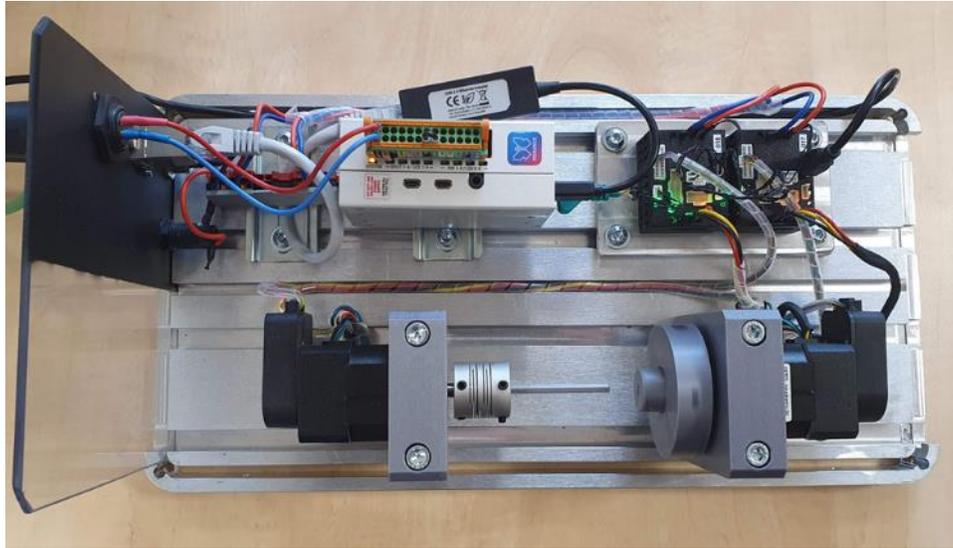


Figure 163. Test stand for smart control algorithm research.

An application for researching smart motor control algorithms was created to provide a platform for algorithms research and validation.

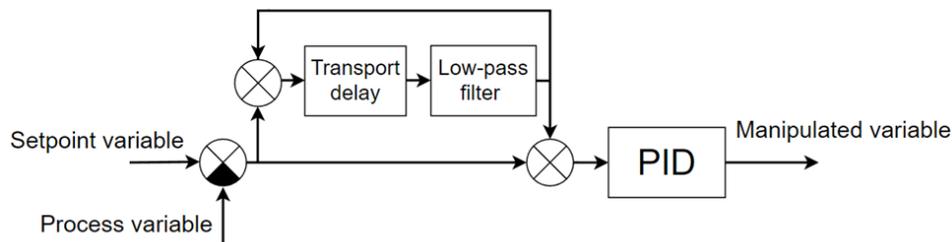


Figure 164. Basic repetitive control schematics.

The basic schematics of the repetitive control application are depicted in Figure 164. In this system, position serves as the manipulated variable.

The control algorithm utilized in this setup integrates a PID regulator and a repetitive compensator. The repetitive compensator comprises a transport delay with a predetermined period for processing an exogenous signal, along with a low-pass filter. This compensator acts as a model to predict the behavior of the exogenous signal, following the principle of an inner model. To maintain stability of the closed-loop system, inclusion of a low-pass filter is crucial, with the cutoff frequency carefully selected to determine the number of compensated harmonics.

The application operates directly on the drive, except for the transport delay implementation, with a cycle time of 125 microseconds. REXYGEN application and EtherCAT® communication with the drive operate with a cycle time of 1 millisecond. Required transport delay is implemented by dividing the application between the drive and the REXYGEN application. Data passed to the transport delay are oversampled, generated every 125 microseconds, and transferred every 1 millisecond, resulting in each EtherCAT® frame containing 8 samples of data. The REXYGEN application utilizes a round buffer to implement the transport

delay, compensating for EtherCAT® communication delay and internal processing in the drive. Consequently, the data from the REXYGEN application returns to the drive in the required 125-microsecond cycle, precisely implementing the necessary length of the transport delay.

Additionally, servo drives were deployed to a 7-axis robotic manipulator designed for automated Non-Destructive Testing, as depicted in Figure 165.



Figure 165. 7-axis robotic manipulator.

The deployment in a 7-axis non-destructive testing manipulator served as a comprehensive test of both the servo drive hardware and software, including the utilization of user-defined motion control algorithms. Measurements conducted on the manipulator using the servo converter's functions enabled the identification of mechanical oscillations within the robot structure. Additionally, verification was performed on the peripherals that can be directly connected to the servo drive unit, specifically the 3D mouse, buttons, and sound signalling.

5.3.4 IMOCO4.E Requirements

Table 46. Requirements that apply to High speed servo drives, variable speed drives.

ID	Requirement	Priority	Result	Comment
Interfaces and connectivity				
R150-D2.3	EtherCAT® connectivity	M	PASS	
R151-D2.3	Biss-C protocol over RS422 (abs. encoder)	M	PASS	
R152-D2.3	RS422 connectivity (abs. encoder)	C	PASS	
R153-D2.3	Incremental encoder connectivity	M	PASS	
R154-D2.3	UART connectivity	C	PASS	
R155-D2.3	CAN connectivity	C	PASS	
R156-D2.3	Digital IO connectivity	M	PASS	
R157-D2.3	Motor brake connectivity	M	PASS	
R158-D2.3	Safe torque off IO	M	PASS	
R159-D2.3	DC motor connectivity	M	PASS	
R160-D2.3	Servo motor connectivity (FOC)	M	PASS	
R161-D2.3	BLDC motor connectivity	W	PASS	
Maintainability (modularity, analyzability, testability)				
R162-D2.3-fw	Drive firmware parameters/variables monitoring (analyzability, testability)	M	PASS	
R163-D2.3-fw	Drive firmware internal process monitoring (analyzability, testability)	M	PASS	
R164-D2.3hw	Drive hardware modular design (modularity, testability)	M	PASS	
Performance				
R165-D2.3	Provide fast data transfer (oversampled variables) of selected parameters/variables over EtherCAT from/to drive to develop and test smart control algorithms, e.g. repetitive control, sample rate of up to 24 kHz	M	PASS	
R166-D2.3	EtherCAT® transfer rate with the servo drive shall be up to 1kHz	M	PASS	
R167-D2.3	Current loop rate up to 24 kHz	M	PASS	
R168-D2.3	Application loop rate up to 8 kHz	M	PASS	
Compatibility (interoperability, co-existence)				

R169-D2.3	Co-existence with I-MECH and IMOCO4.E BBs	C	PASS	Used in BB5
R170-D2.3	Compatibility with IEC 61800, ISO 15745, IEC 61784	S	PASS	
Usability (operability)				
R171-D2.3	The servo drive (BB7) shall be used for internal projects within the IMOCO4.E consortium (internally developed and used manipulators, test stands, etc., expected TRL 5)	M	PASS	
R172-D2.3	The servo drive (BB7) shall be used for external purposes (potential prototype applications outside the consortium, expected TRL 5)	W	IN PROGRESS	
Reliability (fault tolerance, availability)				
R173-D2.3	HW design/testing for high reliability.	S	PASS	
R174-D2.3	SW design/testing for high reliability. SW design to recover from possible faults, e.g. HW faults, communication, etc.	W	PASS	
Portability (adaptability, replaceability)				
R175-D2.3	Ability to adapt the drive hardware to different requirements (hardware, firmware), drive design is modular – allows easy modification for different power stage or io requirements, firmware architecture should reflect this concept	S	PASS	
Scalability				
R176-D2.3	BB7 shall support multiple drives on an EtherCAT® bus, the scalability will be achieved by adding multiple EtherCAT® devices in the system	S	PASS	
Tools/toolchains				
R177-D2.3	Tools for drive firmware parameters/variables monitoring	M	PASS	
R178-D2.3	Tools for drive firmware internal process monitoring	M	PASS	

R179-D2.3	Tools to research, develop, monitor, tune smart control algorithms	M	PASS	Used in BB5
Safety				
R180-D2.3	Requirements for electrical safety IEC/EN 61800-5-1: Adjustable speed electrical power drive systems - Safety requirements - Electrical, thermal and energy	M	PASS	
EMC				
R181-D2.3	Requirements for electromagnetic compatibility: IEC/EN 55011: applies to industrial, scientific, and medical electrical equipment operating in the frequency range 0 Hz to 400 GHz and to domestic and similar appliances designed to generate and/or use locally radio-frequency energy. These standard cover emission requirements related to radiofrequency (RF) disturbances in the frequency range of 9 kHz to 400 GHz. Measurements need only be performed in frequency ranges where limits are specified in Clause 6. For ISM RF applications in the meaning of the definition found in the ITU Radio Regulations (see Definition 3.13), these standard covers emission requirements related to radio-frequency disturbances in the frequency range of 9 kHz to 18 GHz.	S	IN PROGRESS	New hardware version is being verified
R182-D2.3	Requirements for electromagnetic compatibility: IEC/EN 61326: specifies requirements for immunity and emissions regarding electromagnetic compatibility (EMC) for electrical equipment, operating from a supply or battery of less than 1000 V AC or 1500 V DC or from the circuit being measured. Equipment intended for professional, industrial-process, industrial-manufacturing and educational use is covered by this part. It includes equipment and computing devices for - measurement	S	IN PROGRESS	New hardware version is being verified

	and test; - control; - laboratory use; - accessories intended for use with the above.			
Digital twin				
R181-D2.3	Real-time communication of signals/parameters/variables of drive subsystem functionality (e.g., FOC, velocity calculation) to be verified against the independent model (digital twin)	S	NOT STARTED	

5.3.5 Capabilities and Limitations

The servo drive, with its enhanced functionality of user-defined algorithms, is highly versatile and suitable for deployment in various fields. Its primary target area is robotics, with several new deployments planned within the next twelve months. Additionally, the drive is well-suited for research purposes, with the initial deployment planned in a research and educational model for smart control algorithms.

During the project, the development of an updated hardware version became necessary due to the obsolescence and unavailability of some key components in the original hardware design. Electrical compatibility testing for this updated version is currently in progress.

Some applications of the drive may exceed the computational power and control loop speed capabilities of the current platform. However, thanks to the modular hardware design of the drive, it is possible to enhance its computational power and overall parameters by modifying or inserting certain hardware components. As part of this evaluation, we are considering the possibility of extending the drive with FPGA hardware, which would provide additional computational resources for demanding applications. Additionally, the potential use of a Gallium-nitride-based power switching stage could enable remarkably high PWM frequencies required for extremely demanding motion control applications.

5.3.6 Customizations and Adaptations

We have ongoing plans to further extend the drive's hardware to meet the requirements of highly demanding applications. Our focus areas for improvement include enhancing computational power, increasing control loop speed, and raising PWM frequency. By addressing these current weaknesses of the platform, we aim to enhance its overall performance while retaining all existing advanced functionalities and tools that have already been developed. This expansion will also open up new areas of application for the drive, broadening its utility and effectiveness in various scenarios.

5.3.7 Methodology and Toolchains

The servo drive's support for the EtherCAT® communication standard makes it highly adaptable for various applications, facilitating its integration into target systems with ease.

A typical control system comprises a control computer equipped with EtherCAT® capabilities and one or more AMC servo drives connected to the control computer via the EtherCAT® network.

The tools required to deploy the drive into a typical target system include the following proprietary software tools:

AMC Control app:

- A PC application enabling comprehensive control over the drive's configuration and diagnostics.
- Provided by REX Controls s.r.o.

REXYGEN Studio:

- A PC application facilitating the definition of control algorithms
- allows definition of custom algorithm to be executed within the
- allows definition of algorithm running on control computer
- Provided by REX Controls s.r.o.

REXYGEN Runtime:

- Runs on the control computer, providing the control application functionality, including EtherCAT® communication.
- Provided by REX Controls s.r.o.

These tools collectively streamline the deployment process and enable efficient operation of the servo drive within the target system.

5.3.8 References

[1] REXYGEN software tools: <https://www.rexygen.com/>

6 Conclusions and next steps

Deliverable covers a significant number of solutions with a broad technical scope. Novel sensors (BB3), processing platforms (BB1), AI-based components (BB8), vision-in-the-loop (BB2) and drives (BB7) – each of these building blocks have several solutions, leading to a sizeable document. Nevertheless, as a public deliverable, it should be a comprehensive and valuable resource for the audience interested in perception and instrumentation solutions in an industry setting.

As WP3 concludes with this deliverable, activities related to the described technologies will continue in other work packages - WP6 and WP7, where final integration and validation of components will be performed and corresponding deliverables will still follow. In addition, developed solutions should lead to more dissemination possibilities in WP8 during the last period of the project.