| | |
|---|---|
| Doc ID | 20191127063 |
| Doc Creation Date | 22 07 2019 |
| Doc Revision | V0.3 |
| Doc Revision Date | 27 NOV 2019 |
| Doc Status | Final |

| Work Package | Deliverable ID |
|---|---|
| **WP3, Instrumentation layer design and development** | **D3.9 - Instrumentation Layer development final report** |

**Summary**

The document (summary report) provides a comprehensive description of Instrumentation Layer building blocks including implementation aspects on relevant pilot applications, use cases and demonstrators. It provides a detailed description of final research and development results as well as main functionalities of five Building Blocks (BB) - BB1 "Platform for Smart Sensors with Advanced Data Processing", BB2 "Real-time wireless sensors providing complementary feedback information", BB4 "High speed vision", BB5 "High performance servo amplifier" and BB10 "Control Specific Multi many core Platform" and their implementation aspects in four Pilots (Pilot 1 "Generic substrate carrier", Pilot 2 "12 Inch wafer stage", Pilot 3 "High speed packaging", Pilot 4 "Big CNC machining"), two Demonstrators (Demo 1 "Manufacture of an Insulin Delivery System", Demo 2 "Injection mold industry") and three Use-Cases (UC1.1 "Power electronic for hoist and crane sector", UC1.3 "PAC based modular HW for machinery", UC2.2 "I-MECH platform validation on open modular robotic arm"). Furthermore, the document provides information about how to operate with BBs, communication interfaces, etc. In addition, the document provides description how all of the Work Package 3 (WP3) "Instrumentation layer design and development" BB stands into the overall I-MECH picture. Also, the document provides short descriptions and clear links to previous technical deliverables about the progress beyond the state of the art of each BB, considering different requirements and specification from the mechatronics/robotics industry. At the end of the document, conclusions and next steps are given.

**Author** Kaspars Ozols (EDI), WP3 leader

**Keywords: real-time wireless sensors, wireless sensor networks, WSN, smart sensors, sensors, wireless communication, embedded systems, hardware, low latency, data acquisition, wireless charging, industrial applications, robustness, reliability, pre-processing, communication protocols.**

Coordinator Sioux CCM
Tel. 0031 (0)40.263.5000
E-Mail info@i-mech.eu
Internet www.i-mech.eu

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

## Table of contents

## List of Figures

## List of tables

## (Open) Issues & Actions

Open Issues (and related actions) that need central attention shall be part of a file called "IAL - Issues & Action List – Partners" which can be found in the Goolge Drive Partner Zone.

| ID | Description | Due date | Owner | IAL ID |
|---|---|---|---|---|
| OI-xxxxx | | | | |

## Document Revision History

| Revision | Status | Date | Author | Description of changes | IAL ID / Review ID |
|---|---|---|---|---|---|
| R01 | Draft | 22-JUL-19 | OZOLS | Initiate | |
| R02 | Draft | 24-NOV-19 | OZOLS | Input received; document finalized for review | |
| R03 | Final | 27-NOV-19 | OZOLS | Review comments considered; document finalized | |

## Contributors

| Revision | Affiliation | Contributor | Description of work |
|---|---|---|---|
| R01, R02, R03 | EDI | Kaspars Ozols, Rolands Shavelis, Niklavs Barkovskis, Arnis Salmins | As an owner of this deliverable, EDI provided contribution to Sections 1, 2, 3, 5, 9 and reviewed all other contributions. |
| R03 | ZAPUNI | Roman Čečil, Vlastimil Setka, Martin Cech | Description of BB1 and BB2 and implementation aspects (Section 3, 4 and 5) |
| R01, R02 | INL, EDILASIO | Elisabete Fernandes, Gabriel Ribeiro | Description of BB2 implementation in Demo2 (Section 5) |
| R01, R02 | IKERLAN-IK4 | Roberto Pacios | Description of IKE Wireless sensor (BB2) in Section 4 and 5. |
| R01, R02 | CORREA | Jorge Marcos | Description of IKE Wireless sensor (BB2) implementation in Pilot 4. |
| R01, R02 | UCC | Michael Walsh | Description of the TNI Wireless Sensor (BB2) in Section 5 |
| R01, R02 | J&J Vistakon | Seamus Hickey, Fraser Wardrop | Description of TNI Wireless sensor implementation in Demo 1. |
| R01, R02, R03 | INGENIA | Francesc Marlasca | Description of BB5 in Section 3 and 7. |
| R01, R02, R03 | Tu/e, EVI | Claudio Scordino, Dip Goswami | Description of BB10 in Section 3 and 8. |
| R01, R02, R03 | TNO | Rob Snel | Description of High-speed vision system (BB4) and implementation aspects (Section 3 and 6) |

## Document control

| | Status | Draft | Draft | Final | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Revision | 01 | 02 | 03 | | | | | |
| **Reviewer Name** | **Role** | **Selection** | | | | | | | |
| Arend-Jan Beltman | Coordinator | X | | | | | | | |
| Kaspars Ozols | WP3 leader | X | X | X | | | | | |
| Francesc Marlasca | Reviewer | | X | X | | | | | |
| Martin Cech | Reviewer | | X | X | | | | | |
| Rob Snel | Reviewer | | X | X | | | | | |

## File Locations

Via URL with a name that is equal to the document ID, you shall introduce a link to the location (either in Partner Zone or CIRCABC)

| URL | Filename | Date |
|---|---|---|
| [Link] | D3.9 - Instrumentation Layer development final report V0.3 | 27-NOV-2019 |

# Literature

| Ref | Name | Publisher | Year |
|---|---|---|---|
| [1] | D2.1 “Updated state of the art and enabling technologies of motion control solutions for smart mechatronics and robotics” | I-MECH | 2017 |
| [2] | D2.2 “Updated state of the art and enabling technologies of motion control solutions for smart mechatronics and robotics systems” | I-MECH | 2017 |
| [3] | D2.3 “Overall requirements on I-MECH reference platform” | I-MECH | 2017 |
| [4] | D2.4 “General specification and design of I-MECH reference platform” | I-MECH | 2018 |
| [5] | D3.1 “Instrumentation Layer requirements and specification (first iteration)” | I-MECH | 2017 |
| [6] | D3.2 “Instrumentation Layer requirements and specification (final iteration)” | I-MECH | 2018 |
| [7] | D3.3 “Platform for Smart Sensors with Advanced Data Processing (BB 1)” | I-MECH | 2019 |
| [8] | D3.4 “Real-time wireless sensors providing complementary feedback information (BB 2)” | I-MECH | 2019 |
| [9] | D3.5 “High speed vision (BB 4)” | I-MECH | 2019 |
| [10] | D3.6 “High Performance Current amplifier (BB 5)” | I-MECH | 2019 |
| [11] | D3.7 “Control specific multi-many core HW (BB 10)” | I-MECH | 2019 |
| [12] | D4.7 “Unified framework for model-based design of motion control system” | I-MECH | 2019 |
| [13] | D3.8 “Portfolio of micromechanical sensors” | I-MECH | 2019 |
| [14] | D6.5 “Validation reports (final iteration)” | I-MECH | 2020 |

# Abbreviations & Definitions

| Abbreviation | Description |
|---|---|
| ACK | Acknowledgement |
| ADC | Analog to Digital Converter |
| API | Application programming interface |
| BER | Bit error rate |
| BB | Building Block |
| BiSS | Bidirectional/serial/synchronous |
| BLE | Bluetooth Low Energy |
| BSD | Berkeley Software Distribution |
| CNC | Computer Numerical Control |
| CoE | CANopen over EtherCAT |
| COTS | Commercially-of-the-shelf |
| CPU | Central Processing Unit |
| DMA | Direct memory access |
| DOF | Degrees of freedom |

| | |
|---|---|
| GSC | Generic Substrate Carrier |
| DSP | Digital Signal Processing |
| DSS | Direct Spread Spectrum |
| ECU | Electronic control unit |
| EMC | Electromagnetic compatibility |
| EMI | Electromagnetic interference |
| ESC | EtherCAT Slave Controller |
| FoE | File over EtherCAT |
| FPGA | Field-programmable gate array |
| GATT | Generic Attributes protocol |
| GFSK | Gaussian frequency-shift keying |
| GNC | Guidance, Navigation and Control |
| GPU | Graphics processing unit |
| HDL | Hardware description language |
| HTTP | Hypertext Transfer Protocol |
| HW | Hardware |
| HiL | Hardware-in-the-loop |
| IBVS | Image Based Visual Servoin |
| IC | Integrated Circuit |
| I/O | Input / Output |
| IP | Intellectual Property |
| ISA | Instruction Set Architectures |
| I2C | Inter-Integrated Circuit |
| LE | Large Enterprise |
| LSM | Linear Synchronous Motor |
| LTE | Long-Term Evolution |
| MBSE | Model Based System Engineering |
| MCU | Microcontroller |
| MEMS | Microelectromechanical systems |
| MIMO | Multiple-input and multiple-output |
| MiL | Model-in-the-loop |
| OPC UA | OPC Unified Architecture (interoperability standard for the secure and reliable exchange of data) |
| OS | Operating System |
| PCB | Printed Circuit Board |
| PDO | Process Data Objects |
| PDU | Protocol data unit |
| PER | Packet error rate |
| PHY | Physical layer |
| PiL | Processor-in-the-Loop |
| PLC | Programmable logic controller |
| PRR | Packet rejection rate |
| PWM | Pulse-width modulation |
| Qi | An open interface standard that defines wireless power transfer using inductive charging |
| RAM | Random-access memory |
| ROI | Region of interest |
| RPC | Remote Procedure Call |
| RTD | Research and Technology developers |
| RTOS | Real-Time Operating System |
| RX | Receiver/Receive |
| SDK | Software development kit |
| SE | System Exploitation objectives |

| | |
|---|---|
| SFTP | Secure Shell File Transfer Protocol |
| SI | System Integration objectives |
| SiL | Software-in-the-loop |
| SME | Small Medium Enterprise |
| SNR | Signal-to-noise ratio |
| SO | System Operational objectives |
| SoA | State-of-the-Art |
| SoC | System on Chip |
| SPI | Serial Peripheral Interface |
| SSH | Secure Shell |
| ST | Scientific and Technological objectives |
| STO | Safe Torque Off |
| TCP | Transmission Control Protocol |
| TSN | Time-Sensitive Networks |
| TX | Transmitter/Transmit |
| UC | Use-Case |
| UNI | University |
| USB | Universal Serial Bus |
| VGA | Video Graphics Array |
| VHDL | Very High Speed Integrated Circuit Hardware Description Language |
| WP | Work Package |

| Definition | Description |
|---|---|
| | |

# 1 About this document

The document (summary report) provides a comprehensive description of Instrumentation Layer building blocks and their implementation aspects on relevant pilot applications, use cases and demonstrators. The document is organized in nine sections. Section 1 provides description about the document. Section 2 provides an introduction about the Work Package 3 (WP3) “Instrumentation layer design and development”, which includes a description of how all of the WP3 Building Blocks (BB) stands into the overall I-MECH picture and gives short descriptions and clear links to previous technical deliverables about the progress beyond the state of the art of each BB, considering different requirements and specifications from the mechatronics/robotics industry. Section 3 provides information about how to operate with BBs, their communication interfaces, and other aspects. Section 4 - 8 provides detailed description of final research and development results as well as main functionalities of six BB and their implementation aspects in four Pilots, two Demonstrators and three Use-Cases. Last Section 9 provides conclusions and next steps for all BBs.

# 2 Introduction

## 2.1 WP3 overview

Work Package 3 (WP3) “Instrumentation layer design and development” is about layer 1: “Instrumentation layer” (See Fig.1).

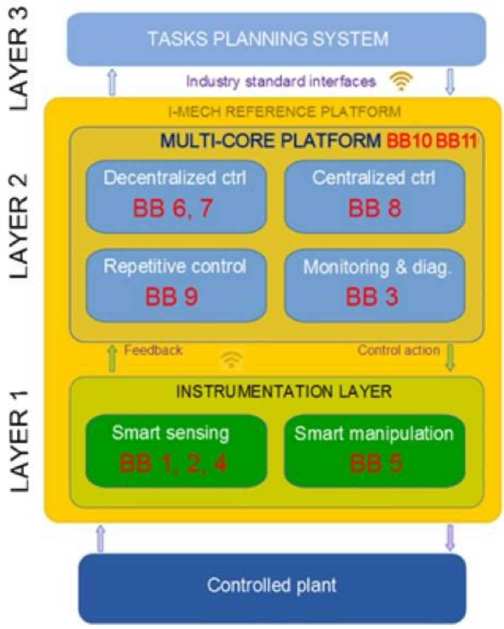



*Figure 1: Layered I-MECH architecture showing the allocation of I-MECH building blocks*

This layer addresses all necessary technologies to sense and manipulate the physical properties of the controlled system. This work package deals with the design of intelligent sensors, drives, actuators and multi-many core ECUs suited to work in smart mechatronic applications. Despite different principles (piezo, mems, induction, ...) size and power, I-MECH strives for a unified (where possible standardized) interface for sensor integration.

The work in WP3 is/was organized as shown in Figure 2.

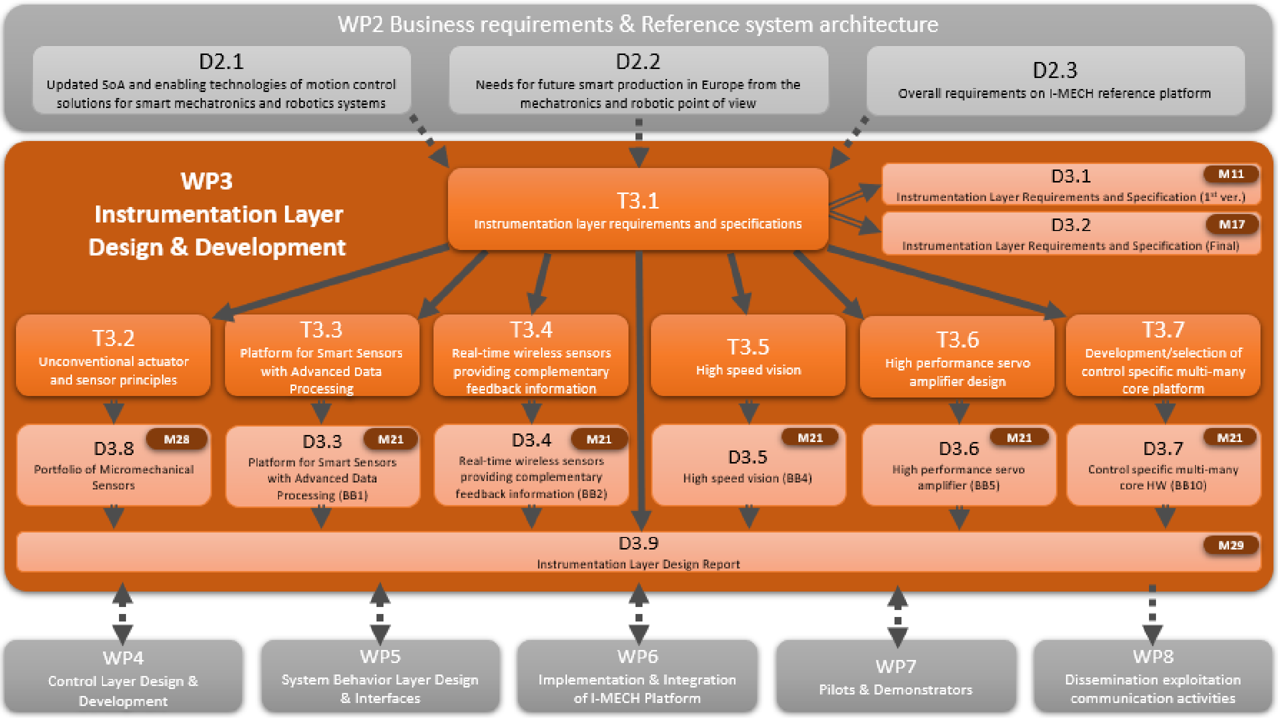



*Figure 2: WP3 implementation plan.*

First, the foundation of WP3 was researched and developed in Work Package 2 “Business requirements and reference system architecture” (WP2), where three important deliverables were delivered: D2.1 “Updated state of the art and enabling technologies of motion control solutions for smart mechatronics and robotics” [1], which provided a review of the relevant existing technologies in advanced and smart production automation and provided recommendations as to which option should be considered by this project; D2.2.”Needs for future smart production (manufacturing) in Europe from a mechatronic and robotics perspective” [2], which identified state-of-the-art and defined gaps in advanced production automation as well as defined high level industrial needs coming from the industrial participants; and D2.3 “Overall requirements on I-MECH reference platform” [3], which defined project width reference platform requirements.

Based on the available information from WP2 and its deliverables, first more detailed requirements and specifications for each Building Block (BB) and Pilot, Demonstrator and Use-case, was developed in two iterations, providing two deliverables: D3.1 “Instrumentation Layer requirements and specification (first iteration)” [5] and D3.2 “Instrumentation Layer requirements and specification (final iteration)” [6], which served as the basis for all the technical developments in WP3, resulting into six technological building blocks: BB1 “Platform for Smart Sensors with Advanced Data Processing”, BB2 “Real-time wireless sensors providing complementary feedback information”, BB4 “High speed vision”, BB5 “High performance servo amplifier” and BB10 “Control Specific Multi many core Platform” as cornerstones of the I-MECH reference platform and relevant deliverables D3.3 “Platform for Smart Sensors with Advanced Data Processing (BB 1)” [7], D3.4 “Real-time wireless sensors providing complementary feedback information (BB 2)” [8], D3.5 “High speed vision (BB 4)” [9], D3.6 “High Performance Current amplifier (BB 5)” [10], D3.7 “Control specific multi-many core HW (BB 10)” [11].

Each of the above mentioned technical deliverables for BBs (D3.3-D3.7) includes comprehensive analysis of state-of-the-art (SoA) and how each specific BB is going beyond SoA, considering different requirements and specifications from the mechatronics/robotics industry; description about the main functionalities of each BB; detailed technical description about each of the developed BBs with preliminary test results; and description about the I-MECH methodology at BB level as well as BB context of model-based systems engineering.

Together with other BBs, developed in WP4 “Control Layer design and development” and WP5 “System Behavior Layer design and interfaces”, all these WP3 technichical developments are integrated, tested and validated in WP6 “ Implementation and integration of I-MECH platform” and WP7 “Pilots and demonstrators”, while contributing to horizontal WP8 “Dissemination, exploitation, communication activities”.

This deliverable D3.9 provides a comprehensive description of the latest research results and developments of Instrumentation Layer BBs, including implementation aspects on relevant pilot applications, use cases and demonstrators.

## 2.2 Context within I-MECH concept

The main I-MECH project goal is to provide augmented intelligence for wide range of cyber-physical systems having actively controlled moving elements, hence support development of smarter mechatronic systems. To achieve this goal, there is a need for new developments and solutions in all three motion control system layers: *Instrumentation* (WP3), *Control* (WP4) and *System Behavior* (WP5) Layers. In the I-MECH project, in each of these Layers several key building blocks (BB) are developed to enable cutting edge reference motion control platform for non-standard applications where the control speed, precision, optimal performance, easy reconfigurability and traceability are crucial.

The ‘I-MECH Platform’ consists of a mix of existing and new subcomponents which are developed into more complex I-MECH BBs. The BBs that follow have been identified as key to meeting the I-MECH objectives. Eleven building blocks have been defined. Figure 1 visualizes which blocks are related to which part of the I-MECH reference model. The defined BBs are (WP3 BBs highlighted in bold):

- **BB1 Platform for Smart Sensors with Advanced Data Processing**
- **BB2 Real-time wireless sensors**
- **BB3** Robust condition monitoring and predictive diagnostics
- **BB4 High speed vision**
- **BB5 High performance servo amplifier**
- **BB6** Self-commissioning velocity and position control loops
- **BB7** Vibration control module
- **BB8** Robust model-based multivariable control
- **BB9** Iterative and repetitive control module
- **BB10 Control Specific Multi-many core Platform**
- BB11 RTOS for multi-many core platform

The key I-MECH challenge is to integrate these BBs into complex scenarios, as shown on manipulator example in Figure 3. The range of potential application is shown via I-MECH Pilots, Demonstrators and Use-Cases where WP3 addresses Pilot 1 “Generic substrate carrier”, Pilot 2 ”12 Inch wafer stage”, Pilot 3 “High speed packaging”, Pilot 4 ”Big CNC machining”, Demo 1 “Manufacture of an Insulin Delivery System”, Demo 2 “Injection mold industry” UC1.1 “Power electronic for hoist and crane sector”, UC1.3 ”PAC based modular HW for machinery” and UC2.2 “I-MECH platform validation on open modular robotic arm”.

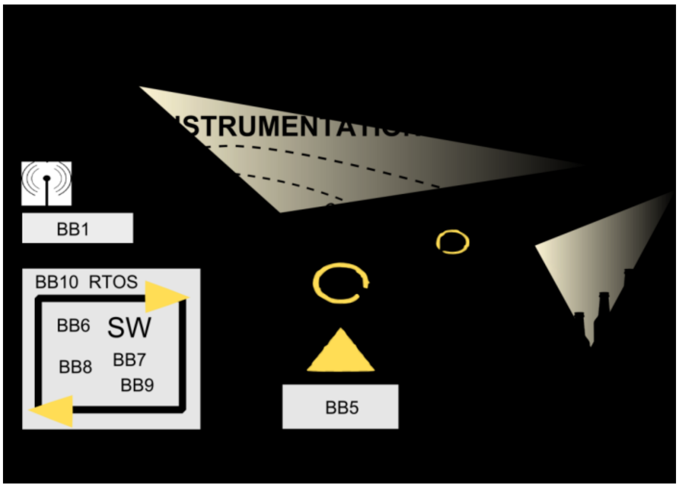



*Figure 3: I-MECH building blocks integration example - generic scenario applied on industrial manipulator*

In the global I-MECH picture, WP3 address several Scientific and Technological objectives (ST), System Integration objectives (SI), System Operational objectives (SO), and System Exploitation objectives (SE):

- *ST1* - to develop techniques for employment of advanced model-based methods for the design, realtime control and self-diagnosis of cyber-physical systems.
- *ST2* - to develop a smart *Instrumentation* Layer gathering visual and/or sensor information from supplementary instrumentation installed on the moving parts of the controlled system to enhance the achievable performance of the system.
- ST3 - To develop modular unified, Hardware and Software motion control building blocks implementing a service-oriented architecture paradigm, i.e. smart Control Layer.

- SI1 - To integrate the developed building blocks into a conceptual open platform for intelligent control of industrial mechatronic systems.
- SI2 - To prove the platform deployability on commercial HW – Use Cases 1.1. – 1.3.
- SI3 - To prove the platform deployability onto commercial industrial robots (fixed, modular) - Use Cases 2.1., and 2.2.

- SO1 - Industrial printing (Pilot 1) – Generic substrate carrier (GSC) for digital printing for graphics and/or functional electronics
- SO2 - Semiconductor production (Pilot 2) – A wafer stage with high precision stepping performance
- SO3 - High speed packaging (Pilot 3) - In-line filling & stoppering machine, Tea bag machine for cotton thread knot technology
- SO4 - Big CNC (Pilot 4) - Smart machining tools and high precision CNC milling machines

- SE1 - To establish so called I-MECH Center (led by SCC) which shall ensure sustainable cooperation between consortium partners after the project termination. It will be open for new interested parties (SME, LE, RTD, UNI) coming outside the I-MECH consortium. Consequently, it is believed that, through such center, I-MECH will become a European solution desk for advanced motion control in cyber-physical systems

## 2.3 Context within I-MECH platform

As described in D2.4 “General specification and design of I-MECH reference platform” [4], the I-MECH reference platform will enable the demonstration of all key I-MECH principles and the validation of building blocks. WP3 BBs supports both decentralized (left) and centralized (right) I-MECH platform architectures (see Figure 4). Besides that, WP3 BBs support MIL/HIL/PIL configurations/setups. Also, WP3 BBs supports configuration and monitoring from layers 2 and 3 and implement following I-MECH principles and features: interoperability, modularity, (self-)diagnosis, self-reflection, maintainability.

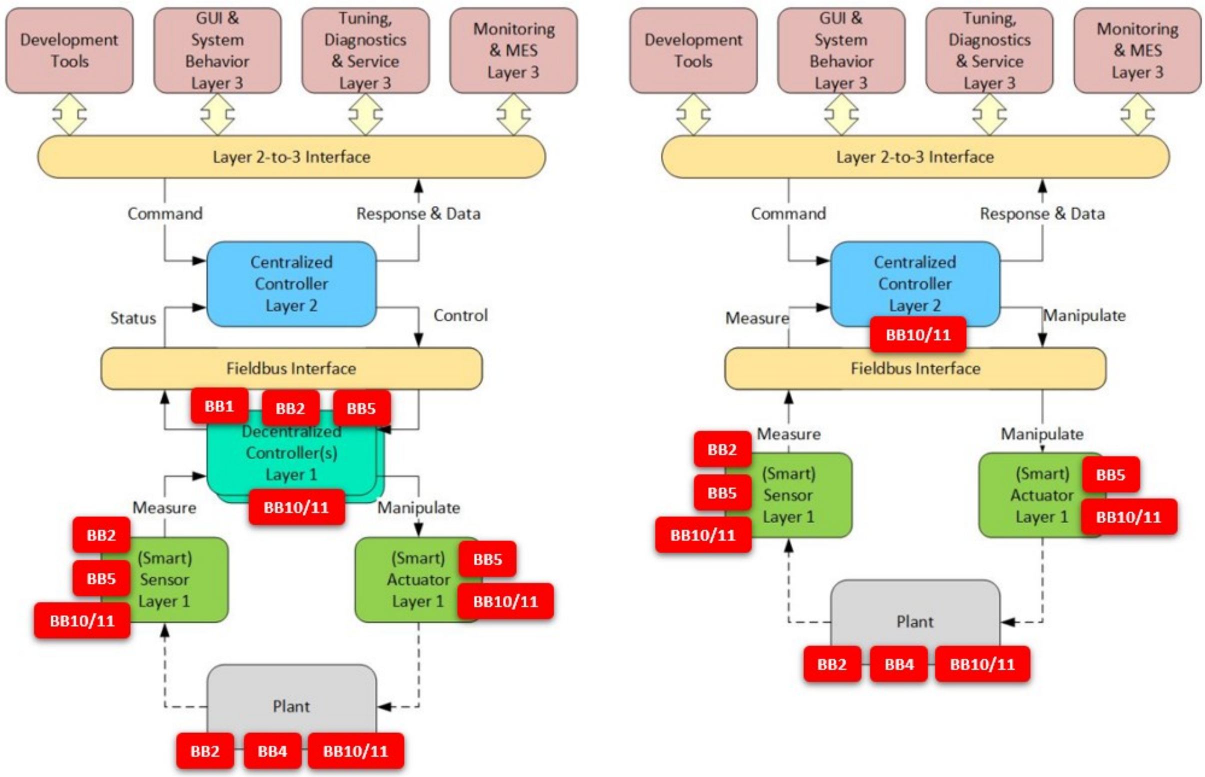



*Figure 4: Position and functionalities in decentralized (left) and centralized (right) I-MECH platform architectures of WP3 BBs.*

A complete overview of the I-MECH reference platform architecture and BB2 position in D2.4 [4].

# 3 How to operate with Instrumentation layer Building Blocks

In this section, basic principles on how to operate each WP3 Building Block (BB) are described. Each BB, has its own aspects and variations on how to use them, for example, how to configure a device, how to choose the best settings, and so on.

## 3.1 BB1 - Platform for Smart Sensors with Advanced Data Processing

BB1 (Common Platform) is a modular device. There is up to 6 slots for I/O cards and several types of I/O cards are available. To operate BB1, first it’s necessary to choose a set of I/O card hardware types which fit individual application needs. This choice also determines which hardware interfaces to sensors, actuators and other devices are available.

BB1 is typically part of I-MECH Layer 1 with EtherCAT fieldbus interface to Layer 2. In this case BB1 act as a Slave Device in the EtherCAT network.

In some less demanding applications, BB1 can also act as integrated Layer 1+2 device with control algorithms and EtherCAT Master inside. Example of this scenario is I-MECH Use Case 1.1 where one or more drives are controlled over EtherCAT from algorithm running on BB1 device.

Core of BB1 implementation is an FPGA to which all I/O cards and fixed interfaces are connected (described in D3.3 [7], ch. 3.6). FPGA firmware can be built automatically from predefined components (IP cores) and optional Customized Core implemented in Simulink. Firmware build is controlled by a configuration file which needs to be prepared for each individual application. This file defines:

- Base hardware variant - multiple types of FPGA chips with different capacity, performance class and temperature range are available.
- Operation mode of the two Ethernet interfaces connected to FPGA - EtherCAT Slave or standard Ethernet (which is capable of EtherCAT Master function implemented in software).
- List of I/O cards in individual slots and functional mode of each I/O card, the I/O Card Core. The same hardware, like generic RS-422 interfaces, can be used for different functions. Multiple IP cores can be implemented for the same hardware with different functional modes, like BiSS-C Master or Slave or S0S90 emulation.
- If the Customized Core should be used, which variant, and all direct interconnects between IP Cores for individual I/O cards and Customized Core. I/O Card Cores usually have predefined connection points which can be used to get data from hardware inputs to Customized Core and back.

If Customized Core is required for the application, suitable template needs to be prepared in Simulink with interfaces matching to the firmware configuration file. In current state, this needs to be done manually. Feasibility of automatic generation of such template should be investigated.

Then, Simulink with HDL Coder toolbox can be used to implement custom algorithms inside FPGA. This follows the I-MECH MBSE approach and the algorithm can be verified in Simulink before building the FPGA code. Using standard Simulink HDL Coder tools, additional parameters and signals can be mapped to predefined memory space to be then exported over EtherCAT as process data or service parameters.

For EtherCAT interface, there is another configuration file which defines process data layout and available service parameters. Standard CoE (CANopen over EtherCAT) is supported for asynchronous parameter access. This file is then automatically used by BB1 Management Subsystem (described in D3.3 [7], ch. 3.7) to initialize Process Data Mapper registers inside FPGA. Process data are composed from memory-mapped registers available for each I/O Card Core and Customized Core.

Configuration files and source files of Customized Core can be then used to build BB1 Firmware Package by BB1 SDK. BB1 SDK is a set of VHDL source files, build scripts and standard build tools including Intel Quartus and Matlab/Simulink HDL Coder. Build can be executed manually on local development workstation or automatically on a build server which follows the I-MECH Portal concept and provides a better reproducibility.

BB1 Firmware Package, the output of the build process, can be then downloaded to BB1 using standard FoE (File over EtherCAT) protocol or using a standard web browser and embedded web server running on BB1.

To access BB1 web server and other diagnostics tools, additional Ethernet interface is available with standard TCP/IP network protocol and application protocols like SSH, SFTP, HTTP. Thanks to open operating system running on BB1 Management subsystem, the RT-Linux, other application protocols and features can be added with very low effort.

BB1 also integrates Real-Time Application Subsystem (described in D3.3 [7], ch. 3.8). It allows to implement a custom software algorithm running on fully isolated ARM CPU core to get the best performance (latency, jitter) possible. Currently this application needs to be implemented in a C language using predefined template. Source files can be compiled to binary form using freely available GCC compiler. Resulting binary file can be downloaded to BB1 using SFTP of FoE protocols. BB1 management Subsystem contains module which allow loading and unloading this application file to isolated CPU core and laso mechanism for asynchronous communication with running application for tuning of parameters and tracing. Real-Time application is typically executed with fixed time period and have access to all I/O data registers by a simple API.

Next to the BB1 Common Platform, two Application Specific Platforms have been developed for Pilot 4 and Demonstrator 1. Operation of these variants are described in ch. 4.2.3 / 5.2.2 and 4.2.7 / 5.2.6 as both these solutions are composed from BB1 and BB2 sub-parts.

## 3.2 BB2 - Real-time wireless sensors providing complementary FB information

To operate BB2, first it's necessary to create a *Matlab Simulink* model from the developed library blocks: BLE master and slave devices; sensors; 802.11b interferer; transmission channel (noise, path loss, interference) (for more details see Section 5.1.8 and deliverable D4.7 "Unified framework for model-based design of motion control system" [12]). The library allows to build BLE models for simulating data transmission between data acquisition (sensors) and data reception sides, by selecting the necessary configuration and setting for target application (e.g. number of slaves, latency, data rate, etc.). When the preferred model has been created and tested, a configuration file for BB2 hardware can be formed and implemented on device. For that purpose, a BB2 hardware (EDI real time wireless sensor, described in D3.4 "Real-time wireless sensors providing complimentary feedback information (BB 2)" [8]) is necessary. The available hardware use standardized interfaces - I2C or SPI for connection of different sensors; BLE5.0 or 802.15.4 (which both can be used in low-latency configuration) or optional low latency proprietary protocol for wireless communication; and EtherCAT for communication to upper layers through BB1). When BB2 hardware is configured, it can be deployed in specific application. Industrial casing and different mounts are available to be able to put the device on differently shaped objects, surfaces and environments.

## 3.3 BB4 - High speed vision

BB4 modules are part of I-Mech layer-1 and interface to layer-2/layer-3 by means of fieldbusses (Ethernet or EtherCAT master). Application configuration-settings are stored in text-based configuration files to be interpreted at the instance of application-launch. These settings are human readable and must be tuned to the specific application.

BB4 modules can be utilized in two modes, depending on the requirements of the application.

- "Smart sensor mode" in which the motion-control-loop is handled over layers-2/layer-3.
- "Local mode" in which the motion-control-loop, including actuation, is operated in the BB4 module and layers-2/layer-3 supply supervisory control only. In this mode the communication overhead is minimised for optimal performance.

The smart sensor modules can in essence be modeled as a linear transformation with latency:

$$H_{smart\ sensor} = K\, e^{-\tau s} \quad (1)$$

Where $\tau$ is the signal latency in seconds, as measured from the beginning of the exposure of the camera until availability of the signal to layer2-/Layer-3. K is the sensor conversion constant. In case the BB4 modules are utilised in local mode the simulation model is adapted to a local feedback loop in which the transformation is given by:

$$H_{local\ mode} = \frac{Hcontroller.Hactuator}{1 - Hcontroller.Hactuator\ .Hsmart\ sensor} \quad (2)$$

The latencies are application specific and should be benched before utilisation of the simulation.

## 3.4 BB5 - High performance servo amplifier

The interface with BB5 is done by means of an EtherCAT network. For establishing communication with the amplifier, the following pattern must be used: CoE EtherCAT register = Key+0x2000. The different Keys for configuring the amplifier, are described in the following tables:

**Max current**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Max current | 0x1E0 | 0x6073; 0x00 | 0 to max. drive peak current | A |

This parameter is used to limit the maximum target current value. It is expressed in A.I2T Parameters

**I2T Parameters**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Peak current | 0x101 | 0x2101; 0x00 | 0 to max. drive peak current | A |
| Peak time | 0x102 | 0x2102; 0x00 | Any | ms |

These parameters allow to configure the I2T parameters to protect the drive and/or motor in front of exceeding their thermal limit.

**Current A PI parameters**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current quadrature loop Kp | 0x500 | 0x2500; 0x00 | Any | V/A |
| Current quadrature loop Ki | 0x501 | 0x2501; 0x00 | Any | V/A |
| Current quadrature loop max. out | 0x502 | 0x2502; 0x00 | -max. voltage to max. voltage | V |
| Current quadrature loop min. out | 0x503 | 0x2503; 0x00 | -max. voltage to max. voltage | V |
| Current quadrature loop Kr | 0x504 | 0x2504; 0x00 | 0,0 to 1,0 | - |

These parameters allow to configure the constants and parameters of the PI controller used for current A regulation. See table above for further information. This set of registers are also used to configure the PI of the current amplifier mode for phase Q.

**Current B PI parameters**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current direct loop Kp | 0x505 | 0x2505; 0x00 | Any | V/A |
| Current direct loop Ki | 0x506 | 0x2506; 0x00 | Any | V/A |

| | | | | |
|---|---|---|---|---|
| Current direct loop max. out | 0x507 | 0x2507; 0x00 | -max. voltage to max. voltage | V |
| Current direct loop min. out | 0x508 | 0x2508; 0x00 | -max. voltage to max. voltage | V |
| Current direct loop Kr | 0x509 | 0x2509; 0x00 | 0,0 to 1,0 | - |

These parameters allow to configure the constants and parameters of the PI controller used for current B regulation. See table above for further information. This set of registers are also used to configure the PI of the current amplifier mode for phase D.

**Current reading values**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current A value | 0x038 | 0x2038; 0x00 | min. measurable current to max. measurable current | A |
| Current B value | 0x039 | 0x2039; 0x00 | min. measurable current to max. measurable current | A |
| Current C value | 0x03A | 0x203A; 0x00 | min. measurable current to max. measurable current | A |

These parameters contain the current actual value of each motor phase.

**Current set-points**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current A set-point | 0x01C | 0x201C; 0x00 | -max. drive peak current to max. drive peak current | A |
| Current B set-point | 0x01D | 0x201D; 0x00 | -max. drive peak current to max. drive peak current | A |

These parameters are used to command the target values for both current inputs. The polarity indicates the direction of rotation.

**Current loop status**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current loop status | 0x517 | 0x2517; 0x00 | See below | - |

These parameters helps to understand the status of the control loop. It has the following structure

| Block | Current D/V loop status | | | | Current Q/U loop status | | | |
|---|---|---|---|---|---|---|---|---|
| Bits | 31 | 30...18 | 17 | 16 | 15 | 14...2 | 1 | 0 |

| Meaning | Current D/B loop enabled | Reserved | Upper saturator active | Lower saturator active | Current Q/A loop enabled | Reserved | Upper saturator active | Lower saturator active |
|---|---|---|---|---|---|---|---|---|

**Current control commands**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current quadrature / A control loop command | 0x098 | 0x2098; 0x00 | -max. drive peak current to max. drive peak current | A |
| Current direct / B control loop command | 0x099 | 0x2099; 0x00 | -max. drive peak current to max. drive peak current | A |

These parameters show the value that is commanded to each current control loop (after the effects of the current offset and reference filters)

**Current loop feedback bypass**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current loop feedback bypass | 0x207F | 0x207F; 0x00 | 0 - Current feedback connected<br>1- Current feedback bypassed | - |

This parameter allows bypassing the current feedback in the current loops, and setting the current readings to 0. This allows using the current loops in open loop.

**Current loop rate**

| Name | Key | CoE id | Value range | Units |
|---|---|---|---|---|
| Current loop rate | 0x458 | 0x2458; 0x00 | - | Hz |

This parameter indicates the frequency at which the current control loop is running.

## 3.5 BB10 - Control Specific Multi many core Platform

BB10 (“Control specific multi-many core platform”) dealt with selecting a hardware platform suitable for implementing the RTOS envisioned in BB11. The activities started by collecting the large amount of requirements coming from pilots, use-cases, I-MECH specifications, etc. These requirements were scattered among several deliverables, and therefore BB10 needed a work of analysis and synthesis of all requirements, then summarized in deliverable D3.7 [11]. A survey about the existing platforms available on the market (available in the deliverable D3.7 as well) allowed to compare the collected requirements against possible hardware selections. Finally, a poll among the involved partners allowed to make the final selection.

BB10 allowed to learn a couple of lessons:

- Industrial automation is still highly dependent on x86 platforms, mainly for running legacy applications and operating systems. At the same time, there is the need/desire of looking forward to possible hardware alternatives.

- Even in the x86 domain, there is no "one fits all" platform, as each company operating in this field tries to optimize the costs by selecting the minimum platform that satisfies its own requirements.

Two main classes of target platforms are considered in I-MECH. First, x86-based architecture on a commercially-of-the-shelf (COTS) board. We refer to that as the COTS platform. Second, an FPGA based customized architecture which we refer to as the FPGA platform.

BB10 tried to address current and future trends by selecting two different platforms:

- An x86 multi-core platform capable of running legacy applications and operating systems (e.g. VxWorks). The reference platform selected has been Advantech ARK-3520P with the i5-6440EQ processor or its counterpart by manufacturer B&R (i.e. APC910 with the same CPU).

*Figure 5. Advantech ARK-3520P*

- The Enclustra Mercury XU5 platform based on the Xilinx Ultrascale+ SoC with ARM cores and FPGA.

*Figure 6. Enclustra Mercury XU5 platform*

Since both are COTS platforms, they can be easily obtained by contacting on-line resellers.

The installation procedure for the former platform consists in installing the Ubuntu 18.04 Linux distribution, then the Xen hypervisor[1] and finally setting up the ERIKA RTOS as DomU guest of Xen. The steps for the installation are available at the following webpage:

http://www.erika-enterprise.com/wiki/index.php?title=ERIKA3_on_the_Xen_hypervisor

The installation for the latter platform is to introduce the platform as a custom target in the Simulink environment. This enables the user to perform the processor-in-the-loop (PIL) and hardware-in-the-loop (HIL) simulations on the targeted FPGA platform. The detailed instruction and further information for the installation can be found in WP4 deliverable D4.9.

# 4 BB1 - Platform for Smart Sensors with Advanced Data Processing (ZAPUNI)

The main BB1 requirements, design and technical developments are described in detail in D3.3 "Platform for Smart Sensors with Advanced Data Processing (BB1)" [7].

[1] Xen hypervisor https://xenproject.org/

In this section, we provide a brief description of Building Block functionalities and implementation aspects in **three Pilots** (Pilot 1 “Generic Substrate Carrier”, Pilot 2 “12 Inch wafer stage” and Pilot 4 “Big CNC”), **three Use-Cases** (Use-Case 1.1 “Power electronic for hoist and crane sector”, Use-Case 1.3 “PAC based modular HW for machinery”, Use-Case 2.2 “I-MECH platform validation on open modular robotic arm”) and **one Demonstrator** (Demo 1 “Process Monitoring and Predictive Maintenance for LSM”).

As described in D3.3 [7], ch. 2.2, most of the implementation share a modular *Common Platform* architecture. However, due to a special requirements and tight integration with brown-field technology, Pilot 4 and Demonstrator 1 defined their own *Application-specific Platform* architectures. But still all implementations share as much of common concepts as possible.

## 4.1 Functionalities

BB1 is a complex technology consisting of modular hardware and several software layers. For better convenience, we provide a brief overview of these functionalities in this chapter. Much more technical details can be found in the whole D3.3 [7].

Here we focus on the BB1 *Common Platform* as it provides the most of functionalities required by majority of I-MECH applications.

**Overview**

BB1 is mostly a Layer 1 hardware device which provides a flexible interface from various electrical field-level interfaces (analog signals, encoders, specialized communication protocols) to Layer 2 controller over EtherCAT fieldbus. The key implementation points are real-time performance and wide software customizability.

**Hardware**

BB1 hardware design is described in D3.3 [7], ch. 3.4. The system consisting of CPU board, backplane and up to 8 I/O cards. CPU board is based on SoC semiconductor device integrating FPGA logic core and multiple ARM CPU cores with three Ethernet interfaces which can have various functions depending on FPGA firmware. Several I/O cards were designed and prototypes manufactured to meet various requirements of related I-MECH applications.

**FPGA Subsystem**

An unified framework for FPGA system architecture was developed. It allows easy handling of modular nature of BB1 by a simple configuration file which defines components used for particular application and their interconnect.

Core functionalities shared by almost all applications are (non exhaustive list):

- **EtherCAT Slave Controller** with high performance (tested up to 20 kHz cycle).
- **Time synchronization** of all components to EtherCAT Distributed Clock reference.
- **Process Data plane** to manage the whole signal chain from I/O, through custom algorithms to EtherCAT and back entirely in the FPGA for the best performance possible.
- **System description** to identify build version, which components at which version are present in the system at which memory locations.

**Management Subsystem**

At least one of the ARM CPU cores is dedicated for system management purposes. This includes various usually non-time-critical tasks like system configuration loading to all components, EtherCAT state machine (Slave Stack), diagnostic protocols (CoE, FoE, HTTP, OPC UA, SSH, …) and reliable firmware update services. To simplify implementation of such features, standard Linux operating system is used with RT-PREEMPT extension which also allow execution of less critical and less performance demanding control applications on the CPUs.

**Real-Time Application Subsystem**

BB1 hardware has enough computing power to run dedicated local control loops on the ARM CPU(s) at performance level fairly above EtherCAT fieldbus. To provide such functionality to end users, AMP (Asymmetric Multiprocessing) approach is used - one ARM CPU can be fully isolated from Management Subsystem and dedicated to control task. A new implementation which allows this next to Linux operating system was developed as part of BB1.

## 4.2 Implementation aspects

In this section, we show how BB1 technology were further developed, deployed and tested in different Pilots, Use-cases and Demonstrators.

### 4.2.1 Pilot 1 “Generic Substrate Carrier”

In Pilot 2, BB1 Common Platform will be the decentralized I/O unit with embedded data processing. The two sin/cos encoders signals from belt unit are connected directly to BB1, custom algorithm is implemented to recover the true position information from these signals, and this data is sent to Layer 2 for feedback control over EtherCAT fieldbus. Another sin/cos encoder provides position information from main belt traction motor side.

The belt position information is also used to synthesize virtual S0S90 sensor output for external device which needs to know the belt position in real-time with very high precision and very low latency. To meet this requirement, the whole signal chain is implemented fully in the FPGA using Customized Core functionality of BB1. 1 MHz sampling rate is used for sin/cos analog inputs and following signal chain. Another virtual S0S90 output is synthesized based on data received from Layer 2 controller over EtherCAT. Many parameters for tuning and diagnostics are also exported over EtherCAT.

BB1 hardware set used for Pilot 1 is described in D3.3 [7], ch. 3.2. Interface for position sensors with plain analog output will be finally implemented using COTS I/O unit instead of BB1 because it can not benefit from BB1 advanced features.

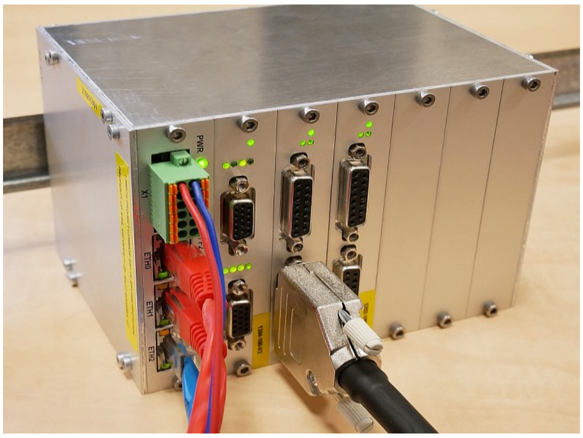


*Figure 7: BB1 box with all I/O cards for Pilot 1, connected one encoder for test.*

### 4.2.2 Pilot 2 “12 Inch wafer stage”

In Pilot 2, BB1 Common Platform will be the encoder interface in which all encoder signals are processed and sent to Layer 2 for feedback control over EtherCAT fieldbus. There are up to 4 S0S90 (quadrature) encoders and up to 5 sin/cos encoders BB1 also offers the option to forward encoder signals to current amplifiers that need the encoder signal for commutation. Evaluation of both encoders types is implemented as a standard functional modes of relevant I/O Cards.

Due to a complex mechanical structure of the machine and different orientations of sensors vs. motors, commutation angle for some motors needs to be composed from multiple encoders in real-time. This function requires maximal latency far beyond EtherCAT limits, so the direct BiSS-C connection with each drive is implemented. Algorithm for encoder data fusion is implemented using Simulink HDL Coder as Customized Core for BB1 FPGA. The BiSS-C slave function is implemented as a standard functional mode of BB1 I/O card with 4 RS-422 interfaces.

BB1 hardware set used for Pilot 2 is described in D3.3 [7], ch. 3.2.

*Figure 8: BB1 box with all I/O cards for Pilot 2.*

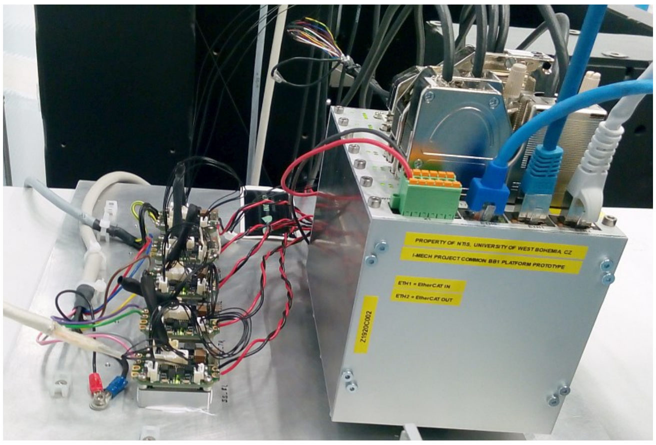


*Figure 9: BB1 box at Pilot 2 test setup with 5 pcs of BB5 drives.*

### 4.2.3 Pilot 4 "Big CNC"

In Pilot 4, BB1 manages data acquisition and processing before sending them remotely. The instrumentation layer of BB1 comprises all the sensors needed to take and monitor working parameters, the microprocessor and the electronics needed for signal processing and conditioning.

The schematics with all the elements are shown in the following picture:

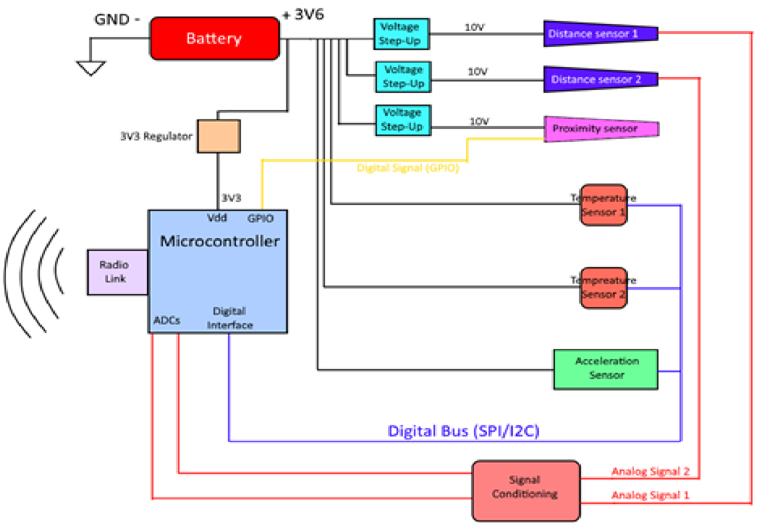



*Figure 10: Schematic of BB1 used in Pilot 4.*

The instrumentation layer components can be divided into two different categories depending on whether they will also be used by other Pilots or not. In this regard, BB1 makes use of specific temperature and distance sensors, and an accelerometer, as well as the circuitry for elements interconnection. Apart from this, it also uses a generic radio-emitter and a micro-controller as it can be seen in the next picture.

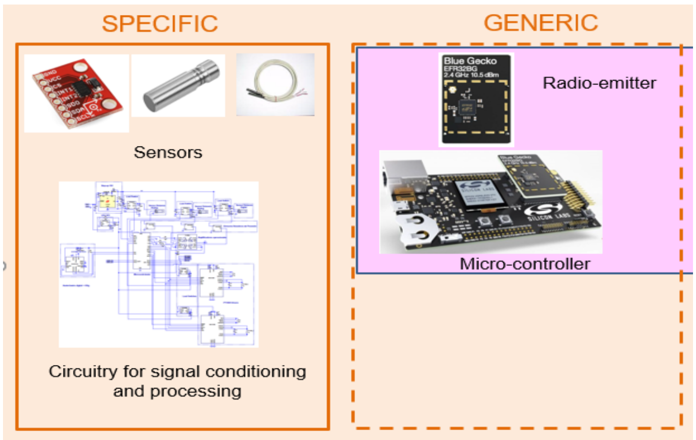



*Figure 11: BB1 used components.*

Finally, all the electronic components, except the sensors are integrated in a PCB specially designed for the geometrical constraints. The sensors are plugged into the PCB with standard connections placed at the right corner of the PCB. The result of the instrumentation layer associated to BB1 can be seen in the next figure:

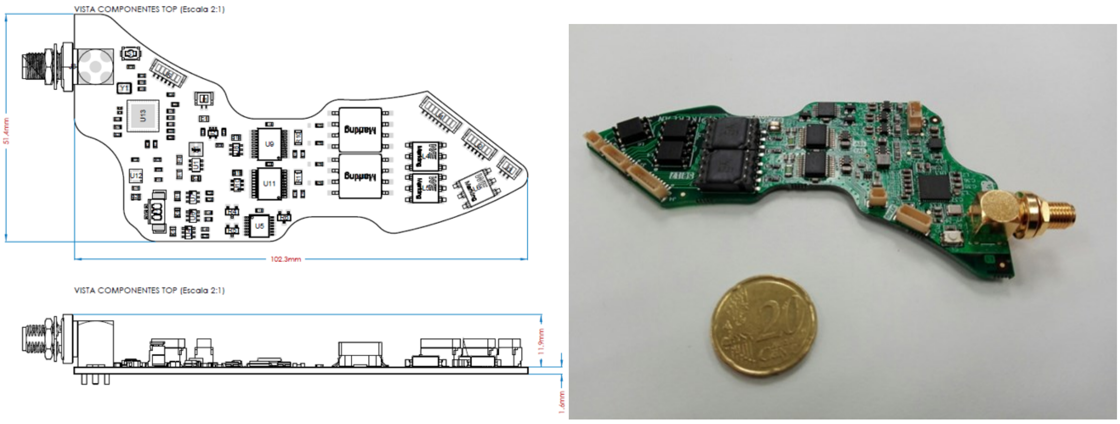



*Figure 12: BB1 developed by IKE.*

### 4.2.4 Use Case 1.1 “Power electronic for hoist and crane sector”

In Use Case 1.1, BB1 Common Platform provides functionalities of both Layer 1 and Layer 2. Using I/O Card with wireless transceivers (part of BB2) it can receive data from distant wireless sensor nodes. In this application, sensor node is typically placed on the load or on the hook of industrial overhead crane. Measured data from the sensor are intended to be used as a feedback for active anti-sway algorithm. Thanks to feedback information, also unintended movements of the crane load caused by external forces (e.g. wind) or non-ideal system model can be corrected.

These anti-sway algorithms can be executed directly on BB1 which in this case also acts as a specialized Layer 2 controller. BB1 software implements EtherCAT Master interface to one or more industrial motor drives which control individual axes of the crane. Also inputs from manual push buttons which are normally processed in the drive firmware itself are in our scenario passed to the BB1 over EtherCAT. BB1 provides velocity setpoint based on the anti-sway algorithm and feedback information from wireless node(s) to the drive(s) over EtherCAT.

BB1 hardware set used for Use Case 1.1 is described in D3.3 [7], ch. 3.2.

### 4.2.5 Use Case 1.3 “PAC based modular HW for machinery”

The purpose of Use Case 1.3 for BB1 Common Platform is to provide a test bench for verification of various BB1 functionalities in a laboratory environment. BB1 is used as I/O interface with EtherCAT data exchange to Layer 2 controller. Several devices and signals for test and verification of BB1 are available:

- 4 pcs of Schneider Electric LXM32 servo drives with EtherCAT and BiSS-C interface modules. Each drive also contains S0S90 (quadrature encoder) inputs and outputs.
  - To test BB1 EtherCAT Slave function interoperability with other Slave devices, S0S90 inputs and BiSS-C Slave encoder emulation of BB1.
- 4 pcs of synchronous motors with sin/cos encoders connected to LXM32 drives.
  - To test sin/cos inputs of BB1 and compare with values from drives itself for verification.
- 2 pcs of linear encoders with BiSS-C interface mounted on cartesian robot.
  - To test BiSS-C Master encoder interface of BB1.
- 4 pcs of S0S90 encoders mounted on partially flexible belts driven by synchronous servo motors or optionally asynchronous motors.
  - To check S0S90 inputs of BB1 including synchronization aspects of the whole system.
- TECO commercial platform PAC (Programmable Automation Controller) with software capable of EtherCAT Master and Motion Control.
  - To check EtherCAT interoperability of BB1 with EtherCAT Master software used on TECO platform.

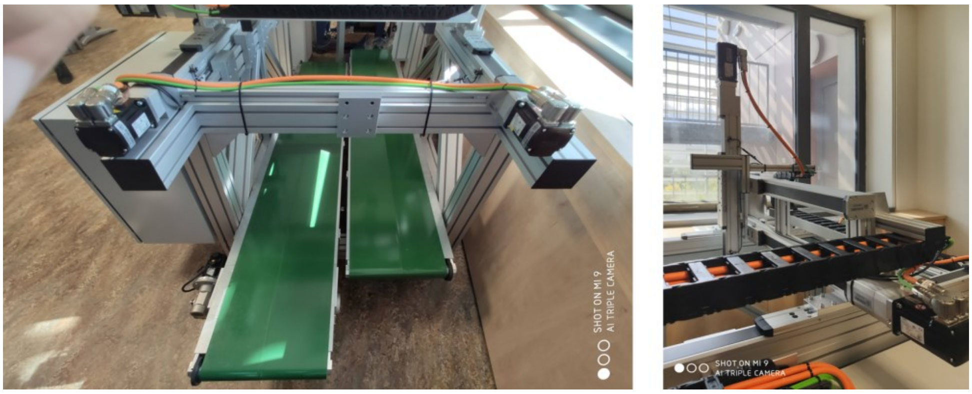


*Figure 13: View on the Use Case 1.3 machine used for various test tasks with BB1.*

### 4.2.6 Use Case 2.2 “I-MECH platform validation on open modular robotic arm”

Use Case 2.2 is based on open modular robotic arm platform developed by ZAPUNI outside I-MECH scope but used to test and verify several hardware and software I-MECH Building Blocks including Common Platform in laboratory environment.

The robotic arm consisting of 7 joints (axes), each with synchronous motor and two precise encoders - on motor side and load side on gearbox output. BiSS-C standard is used as an interface between encoders and drive units. This was used to test several functionalities of BB1 Common Platform related to BiSS-C (Master to read encoder, Slave to emulate encoder, Passive Master to listen on encoder used by the drive).

BB5 is used in place of all drive units with EtherCAT interface to Layer 2, so this platform was also used for several tests of interoperability between BB1 and BB5 - related EtherCAT, BiSS-C and SPI interface. Especially the timing aspects of EtherCAT communication can be evaluated in this setup thanks to the integration of the whole data processing into FPGA on BB1. We can check if there are no unexpected latencies in data exchange and interfaces.

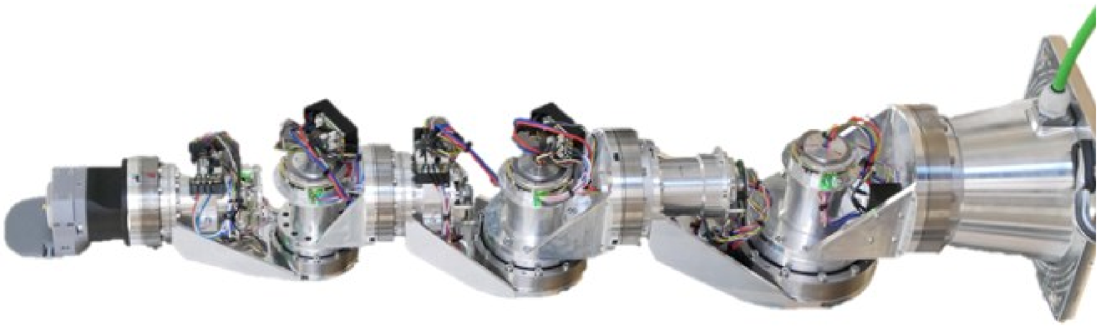


*Figure 14: View on the Use Case 2.2 7 DoF open modular robotic arm.*

# 5 BB2 - Real-time wireless sensors providing complementary feedback information

The main BB2 technical developments and initial test results are described in great detail in D3.4 “Real-time wireless sensors providing complimentary feedback information (BB 2)” [8]. As highlighted in the D3.4 document, BB2 consists of seven core functionalities (sensors, real time data acquisition, reliable data transmission, low power consumption, extended operation time, wireless charging, low cost solutions), implemented in four real-time wireless sensors - EDI+ZAPUNI real-time wireless sensor, IKE real-time wireless sensor, TNI real-time wireless sensor and INL real-time wireless sensor. In this section, a brief description of implemented approaches for each real-time wireless sensors are given together with descriptions of implementation aspects in two Pilots (Pilot 2 “12 Inch wafer stage” and Pilot 4 “Big CNC”), one Use-Case (Use-Case 1.1 “Power electronic for hoist and crane sector”) and two Demonstrators (Demo 1 “Process Monitoring and Predictive Maintenance for LSM” and Demo2 “Injection mold industry”

## 5.1 Functionalities

In general, based on the requirements and specification defined in D3.2 “Instrumentation Layer requirements and specification (final iteration)” [6], the purpose of all real-time wireless sensors is to acquire the data from sensors (e.g. accelerometer, force, corrosion, etc.) with as low as possible latency (<500us), while ensuring reliable data transmission and low power consumption. In addition, the aim is to extend the operation time by introducing wireless charging feature. In order to provide low cost solution, commercially available low-cost components (sensors, MCU, transceiver, etc.) are used. Each of those functionalities are described below.

**Sensors**

In order to improve performance of different robotic and mechatronic systems, it is important to measure various parameters of them. Both novel sensors (e.g. those developed in D3.8 “Portfolio of micromechanical sensors” [13]) as well as COTS sensors can provide the necessary information to perform this task. Even though, different Pilots, Use-cases and Demonstrators needs different sensors, developed real-time wireless sensors share the same communication interfaces: SPI and I2C, so functionally they are compatible with a series of different sensors. In developed real-time wireless sensors we use and demonstrate following sensors - accelerometer, gyro, magnetic field strength, force, corrosion, temperature, digital and analogic distance, blue light intensity, etc.) ensuring one of the most important functionalities - ability to sense. More details available in [8].

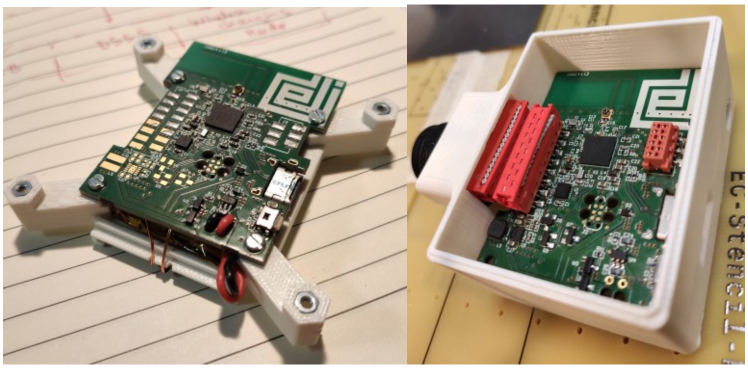


*Figure 15: EDI real-time wireless sensor hardware with integrated inertial sensors*

**Real-time data acquisition**

EDI real-time wireless sensor hardware (BB2) supports multiple radio protocols for near real-time data transmission, for now supported solutions are BLE(best for general connectivity), 802.15.4(most resistant to interference), custom 2.4Ghz GFSK protocol(lowest latency). Based on target necessities any of the above or any combination can be used.

EDI node supports real-time streaming of single sensor with latency under 500us (full speed accelerometer, gyroscope or magnetometer) however, if custom 2.4Ghz GFSK protocol is used packet rate of up to 2Khz can be achieved, which in combination with multiple samples per packet can provide full speed (4.5Khz per axis) accelerometer streaming with latency still under 500us.

In order to provide the required data flow from BB2 wireless sensors to BB1 controller it is essential to use proper communication protocol and transmitter hardware (gateway) that passes the remote data to the controller (and vice versa). In scope of the I-MECH project 2 solutions have been utilized by ZAPUNI - RETIS Gateway and BLE GATT Gateway.

*RETIS Gateway*

In many robotic and mechatronic systems, low latency (<500us) real-time data acquisition is crucial to ensure optimal function of the system (see example of Use-Case 1.1 “Power electronic for hoist and crane sector” in Section 5.2.3). With regards to the requirements we have used the results of the TAČR Delta project TSN-CPS (Reliable Time-Sensitive Networks in Distributed Cyber-Physical Systems for Real-Time Control Industry 4.0 Applications), TF04000048, carried by ZAPUNI. The mentioned project was focused on the design of low latency, low jitter, reliable wireless communication solution for industrial applications. The RETIS protocol and hardware structure developed there were integrated into the BB1/BB2 modules and ensure seamless data exchange between BB1/BB2 blocks.

The RETIS communication protocol is proprietary protocol inspired by LTE, 5G and 802.1 TSN concepts. This MIMO wireless system working simultaneously at multiple frequency channels or even bands, allows highly deterministic and reliable data exchange between controller and sensor or actuator nodes with latency below 300 μs and refresh rates above 1.5 kHz. Moreover, sub-μs clock synchronization between all nodes allows synchronized operation at all nodes, which is essential for control of fast systems.

The devices in the network have assigned time-frequency transmission slots and thus setting deterministic data exchange network and effectively avoiding packet collisions. The possibility to send identical data in multiple T-F slots allows to add redundancy and increase communication robustness in demanding applications where data loss is not an option. Another possibility is to use multiple receivers with diversified antennas to diminish reception losses. At last but not least - by default the IEEE 802.15.4 PHY is used, which uses DSS (Direct Spread Spectrum) modulation techniques that are less prone to interference.

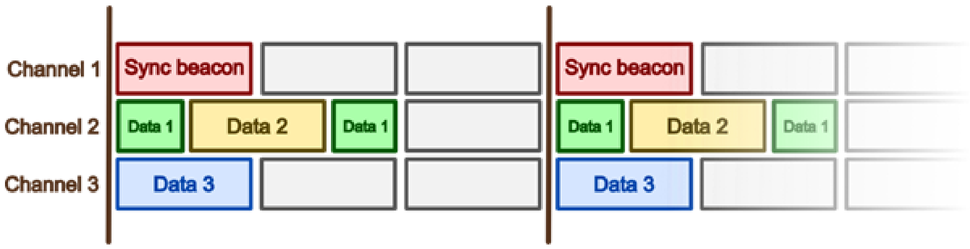



*Figure 16: Time-frequency division multiplexing used in the RETIS wireless solution. The T-F slots are granted to nodes by network controller and thus efficiently mitigating packet collisions. The number of RF channels used is configurable and mostly ranges from 1 to 4.*

*BLE GATT Gateway*

Some of the industrial applications do not have strict requirements on the latency as mentioned above. Instead connectivity to large portfolio of sensors from various manufacturers is required. And the Bluetooth Low Energy GATT protocol is the frequently used solution. For this case a BB1 BLE 5.0 GATT gateway was developed.

The hardware structure of the BLE gateway was inspired by the above mentioned RETIS gateway. It contains 4 independently configurable BLE transceivers which can be used as:

- master in the BLE 5.0 centralized network (master-slave). The device collects data from slave sensor/data nodes and provides them to BB1. It can also distribute computed results to the wireless nodes (remote actuators).
- slave in the BLE 5.0 centralized network. The device exchanges data (in/out) with its master. Gateway with one/more transceiver(s) configured as master in one network and second transceiver configured as a slave in another network allows to create hierarchical network structure (Figure 10).
- peer node in the BLE 5.0 mesh network
- redundant peer node in the BLE 5.0 mesh network. Including multiple gateway transceivers in the same BLE 5.0 mesh network allows to use multiple receive paths and aggregate RX data to get more reliable and robust data source.

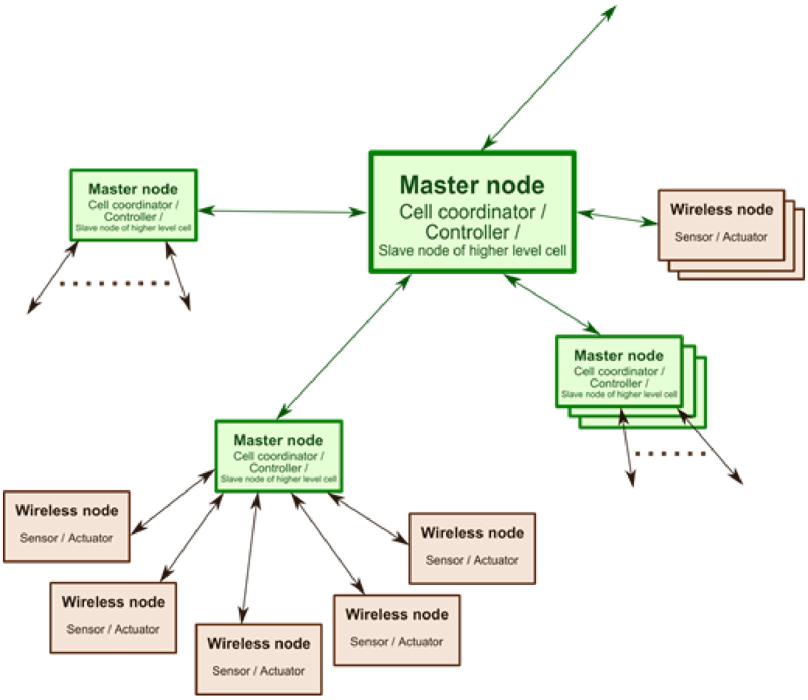



*Figure 17: Hierarchical network structure using multiple gateway transceivers in master/slave modes. Applicable both to the RETIS / BLE gateways (or their mixture).*

### Reliability

Before configuration of communication, the node can listen to whole target protocol spectrum and conclude optimal channel for communication (for non BLE case). Radio link quality is constantly measured with each received packet to provide communication quality feedback however, for now no channel hopping scheme is implemented. For non-BLE case ACK packets can be used to signal gateway of successful data reception. For very noisy environments solution based on 802.15.4 PHY standard should be used for best reliability.

### Low power consumption

Low latency and fast update rates allow advanced control strategies exploiting auxiliary load-side measurements to be employed. As in many practical applications, the control system needs input data measured at places where it is hard or even impossible to deploy a wired solution (rotating, moving or hanging parts, etc.), it is especially important that developed real-time wireless sensors are energy efficient with as low as possible power consumption. To enable this functionality, SoA MEMS, MCU, etc. components are used to ensure maximum operation time without charging.

BB2 software is designed to provide maximum data rate between sensor and radio, which means relatively high power consumption. It is done using multiple DMA channels to minimize microcontroller awake time. This way we are able to achieve under 10 mA average power consumption on EDI real-time wireless sensor HW (see Fig.8) while streaming sensor readings via Bluetooth. Using custom 2.4Ghz GFSK and 802.15.4 solutions power consumption is higher, but it is still under 25 mA and can be reduced greatly by tweaking sensor rate and node duty-cycle. Specific test results are available in deliverable D6.5 “Validation reports (final iteration)” [14]. On the other hand, the TNI wireless sensor uses WiFi to transmit the data, which during transmission uses a peak power of 210 mA. However, in a normal condition of operation the data is transmitted once per day. The battery used by the TNI sensor has a capacity of 1230 mAh, given this capacity and given that the transmission happens once per day, the number of days without charging amount to 86 days. Furthermore, Fig. 11 below shows what happens when the data is transmitted more than once per day and what if the Bluetooth is used instead of the WiFi.

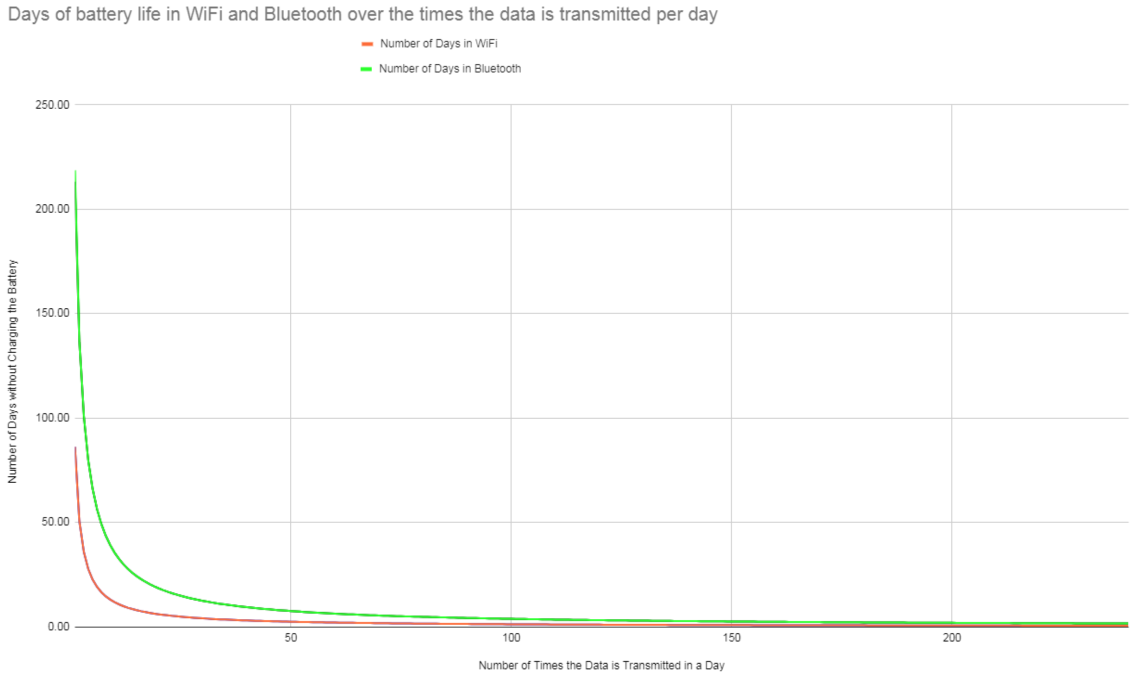



*Figure 18: Number of days vs amount of times data is transmitted per day in WifI and Bluetooth*

**Extended operation time**

Prior to BB2 deployment and Pilot integration, the corresponding electronics were characterized in terms of energy consumption in order to predict the device operation time. There are two main components in the power consumption; First, the energy needed for the sensors and the rest of the system to perform a measurement, and second the energy needed for the system to be always aware and be able to receive the trigger to start the measurement (sleep mode).

As an example, we take Pilot 4. For this system two sensors (and measurements -digital and analogical distance sensors-) are needed to perform and monitor the Head Movement action. An additional analogical sensor for the Tool Change action and two temperature probes to measure the temperature of the inner bearings. Apart from this, all the sensors are connected to a microprocessor which controls and handles operation and data sending.

In a normal operation mode as defined in the requirements and specifications, the daily energy consumption simulated in the lab lies at 13,05 mAh. The selected battery to power the system has a capacity of 5800 mAh. Therefore the expected operation time is 1.2 years. As it can be seen in the Figure 12, 48% of this energy is consumed by the system only to be awake and listening for any command to start measuring. It has to be said that the micro has an ultra-low power consumption mode to run in this case. As soon as it receives the signal, it activates the rest of the functionalities needed for the measurement.

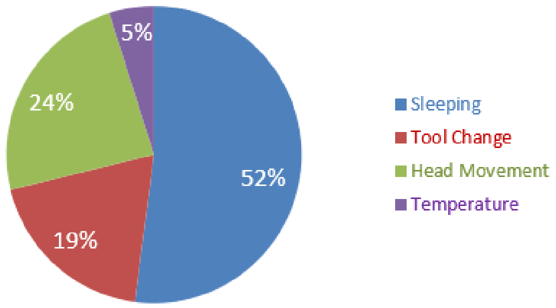



*Figure 19. Energy Consumption of Integrated Wireless Sensors.*

Similar approach and setup is used in other wireless sensors. This fact importantly limits the operation life time, so that IKERLAN and EDI has introduced wireless charging capabilities in the instrumentation layer (see below).

**Wireless charging**

As in many applications there is a need for high refresh rates of data, which has a huge impact on operation time, two wireless charging solutions, not described in [8] were developed to extend the operation time. Description of them is provided below:

*IKERLAN Wireless charging solution*

IKERLAN has been characterizing Qi-compliant or proprietary evaluation kits for Texas Instrument (see following picture).

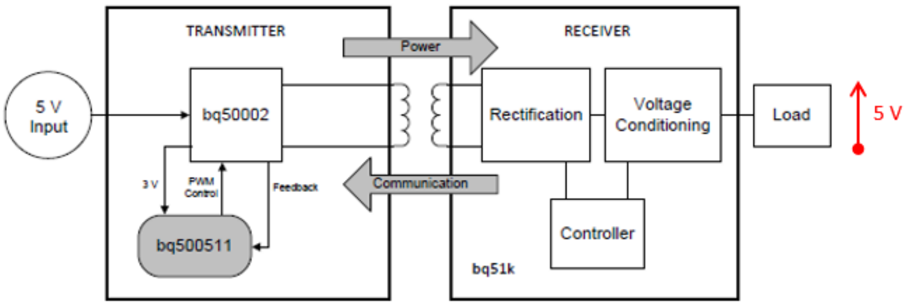



*Figure 20. Block diagram of IKE wireless charging.*

The Qi protocol works as follows:
1.- A concrete load is chosen by the user depending on the application
2.- The system requires a 5V input signal and tries to deliver always 5V at the output
3.- When the coupling between the transmitter and the receiver is perfect in terms of separation distance and alignment, the power transfer maximizes and reach the levels defined in the datasheet (around 75%)
4.- If the coupling decreases, the receiver tries to increase the current by decreasing the resonant frequency of the receiving coil.
5.- This frequency is communicated to the transmitter that adjusts the resonant frequency of the emitter to the same one.
6.- As the frequency decreases, more current is needed from the source to keep the power constant.

$$I = \frac{V}{X_L} = \frac{V}{\sqrt{R^2 + \omega^2 L^2}}$$

7.- There is a point for which the transmitter cannot deliver more power and this must be taken from the source. At this point is when the efficiency of the system starts decreasing.
In the following picture, we can see how the efficiency decreases for different loads as the distance between resonant coils increases:

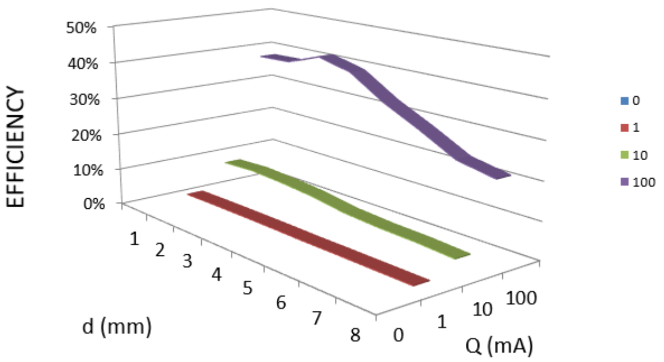



*Figure 21. Efficiency decrease based on different loads as the distance between resonant coils increases.*

The control algorithms implemented in the Qi protocol adjust the frequency of the resonant coils to deliver the same power. We measured the frequency with an oscilloscope to in detail characterize this process:

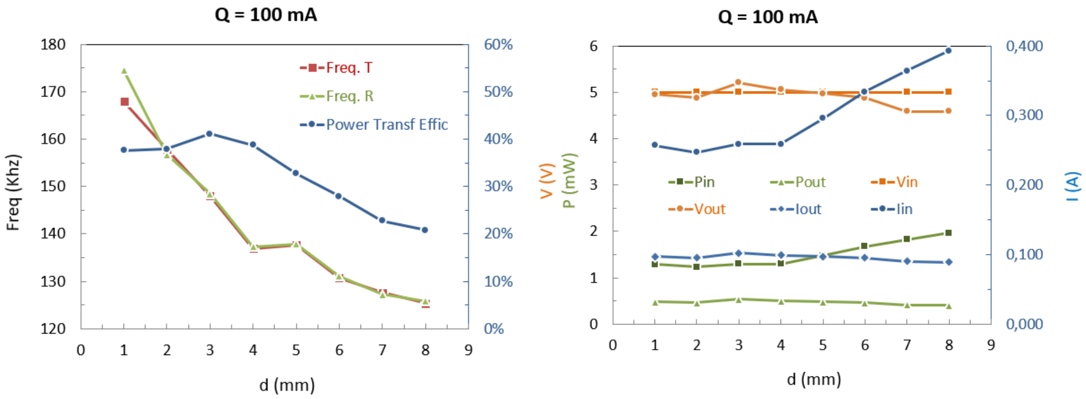



*Figure 22. a) Frequency measurements b) Voltages and currents in and out.*

When the distance between the two coils is larger than 4 mm, the system tries to decrease the frequency even further, but the emitting electronics are not able to deliver more current, so that this is taken from the source decreasing considerably the efficiency of the process.

As a conclusion, the wireless power transfer process is very sensitive on the distance and alignment between the two inductive coils. This makes it necessary to think carefully on the mechanical design in order to guarantee small tolerances that ensure proper power transfer. In any case, the transmitted power is very small in comparison with the power requirements. The use of different coils could in principle enhance this power transfer and will be evaluated as an alternative strategy for the integration of the Pilot once WP3 finishes.

*EDI wireless charging solution*

EDI developed a wireless charging system, deviating from the QI standard. The principle of operation is shown in the picture below.

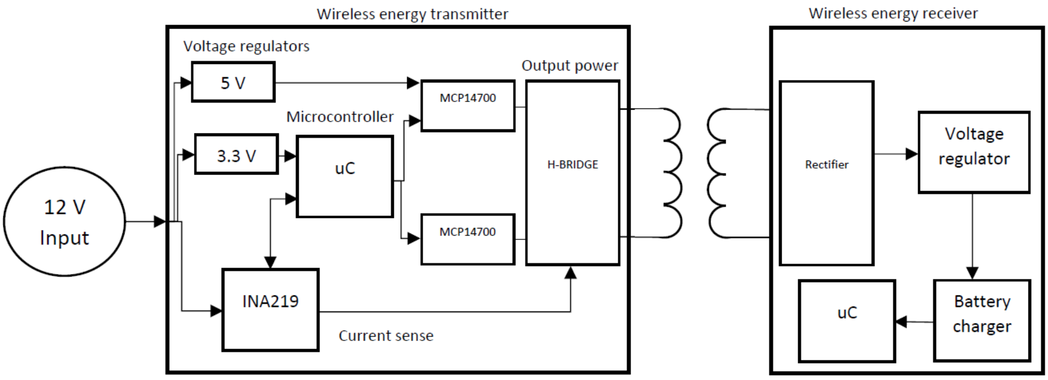



*Figure 23. Block diagram of EDI wireless charging.*

A non-standard solution was devised to improve power transmission of the wireless charging station by deviating from the QI standard, mainly to increase the transmit power and transmission distance. The operating principle of the wireless charging station is similar to the QI standard principle, but the components used differ. In order to increase the output power, the supply voltage is increased to 12V. Instead of a BQ microcontroller, a dsPIC microcontroller are used to control the H-bridge output frequency and duty cycle. The transmitter consists of microcontroller, mosfet drivers, H-bridge, power measurement chip, and LC circuit. The receiver side consists of LC resonant circuit, a diode bridge rectifier, an amplitude demodulator and a battery charger. The main benefits of the system are low cost design, power transfer increased around 50W compared to QI standard, customizable power transfer scenarios, compatibility with QI standard systems. The disadvantages are the complexity of the operating algorithms in the implementation of various applications, the power limiting factors resulting from the coil dimensions and the peak value of the H-bridge load current.

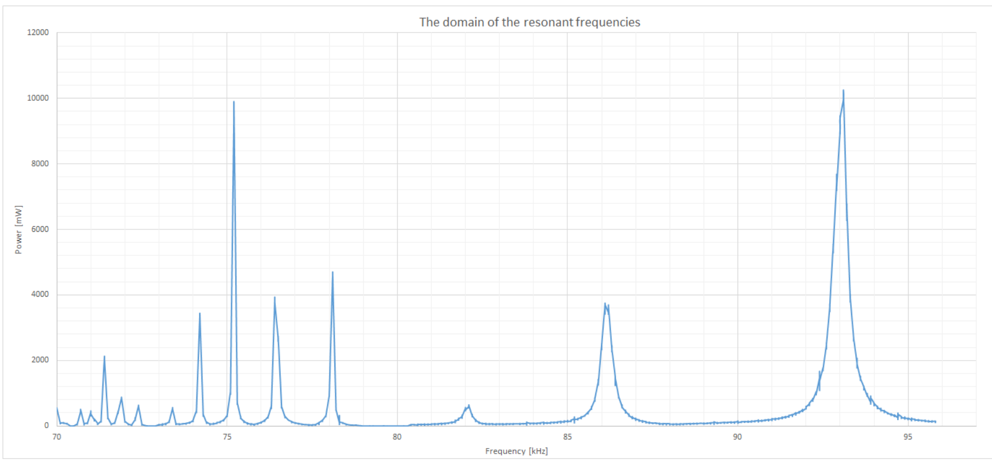



*Figure 24. Acquired power by increasing the source frequency in the range of 70 to 100 kHz.*

The figure shows the measured power by increasing the source frequency in the range of 70 to 100 kHz. Output data is used in scenarios to select the effective transmission frequency, object detection by consumed output power and in case of various security solutions to protect the wireless transmitter and receiver. With a 500mAh battery at a charging current of 500mA, the battery is expected to be charged in 1.2 hours. Power transmission distance is planned up to 120mm.

**Low cost solutions**

To enable this functionality, SoA low cost MEMS, MCU,SoC, etc. components/modules are used to reduce the costs.

For example, for EDI real-time wireless sensors hardware, the costs of one sensor node is ~€80.00 (when manufacturing 1000+ items, the costs could be reduced by 50% or even more). The final price of the product still could be higher, as it has to cover R&D expenses. The high the volume of sold sensors, the lower the price.

Another example is IKE wireless sensor. In order to evaluate the cost of the solution, one has to take in mind and differentiate three concepts: price, cost, and value. Additionally, the cost of the solution has to be evaluated in comparison to the price of the machine where it is integrated. In the case of Pilot 4, the price of one of these big CNC machines is around 1M€, while the cost of the complete solution taking into account all the components can be around 600€. The latter excludes the developed knowledge that can be paid off by selling on value the same device several times for different clients and applications. Considering also that added value offered by the solution to the containing machine, this can be regarded as a very low cost solution since there was no possibility prior to this project to directly monitor these parameters.

Last but not least, for TNI wireless sensor, to maintain costs as low as possible they used only widely available sensors that suited the system requirements. During the prototyping phase, the number of sensors manufactured and assembled were 10, yielding to a total price of approximately 420 euro per wireless sensor board. This cost can further decrease when the scalability rises up. For example, if 10000 sensor boards are produced the cost per board can decrease of an order of magnitude, bringing the price from more than 400 to less than 100. Another solution could be reducing the size and the total number of components available on the board and this can easily be done once the prototyping phase is concluded.

**Bluetooth Low Energy (BLE) wireless sensor network library**

A specialized MATLAB *Simulink* library (Fig. 18) for building the BLE models for simulating the data transmission between data acquisition (sensors) and data reception sides has been created. The library includes configurable blocks of:

- BLE master and slave devices;
- sensors;
- 802.11b interferer;
- transmission channel (noise, path loss, interference).

The library also includes a block for diagnostics, which allows to observe the battery status of the data acquisition device (BLE slave); inspect various signals encountered in BLE devices during simulation (channel frequency, transmit enable, receive enable, etc.); and provides the measurements of the bit error rate (BER), packet error rate (PER), and packet rejection rate (PRR) in noisy channels.

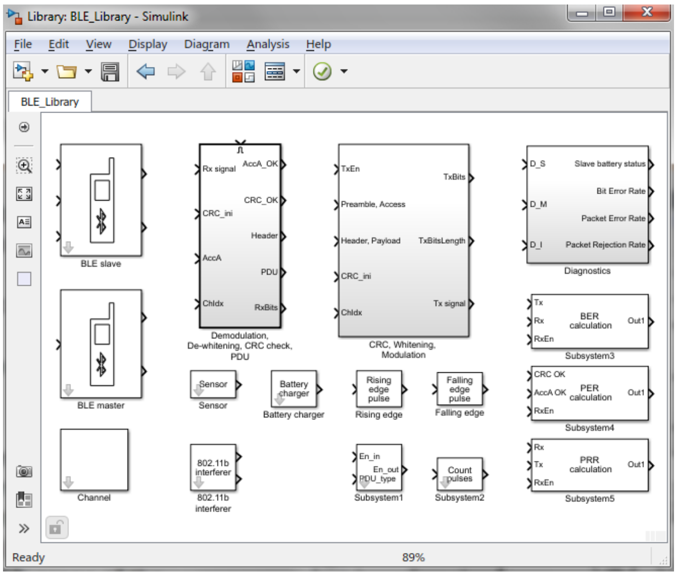



Figure 25: BB2 Simulink library

For generating the initial BLE models, a special MATLAB function has been written, which allows to specify the locations of the devices for taking into account the distances between them (Fig. 19 and Fig. 20).

```
xy_BLE_master=[0,0; 0,10]; %coordinates (m) of 2 master devices
xy_BLE_slave=[10,0; 10,10];%coordinates (m) of 2 slave devices
xy_Interf=[-10,0];          %coordinates (m) of 1 802.11b interferer
new_BLE_model('BLE_model',xy_BLE_master,xy_BLE_slave,xy_Interf)
```

*Figure 26: MATLAB code for generating the BLE model consisting of two masters and two slaves and one interferer*

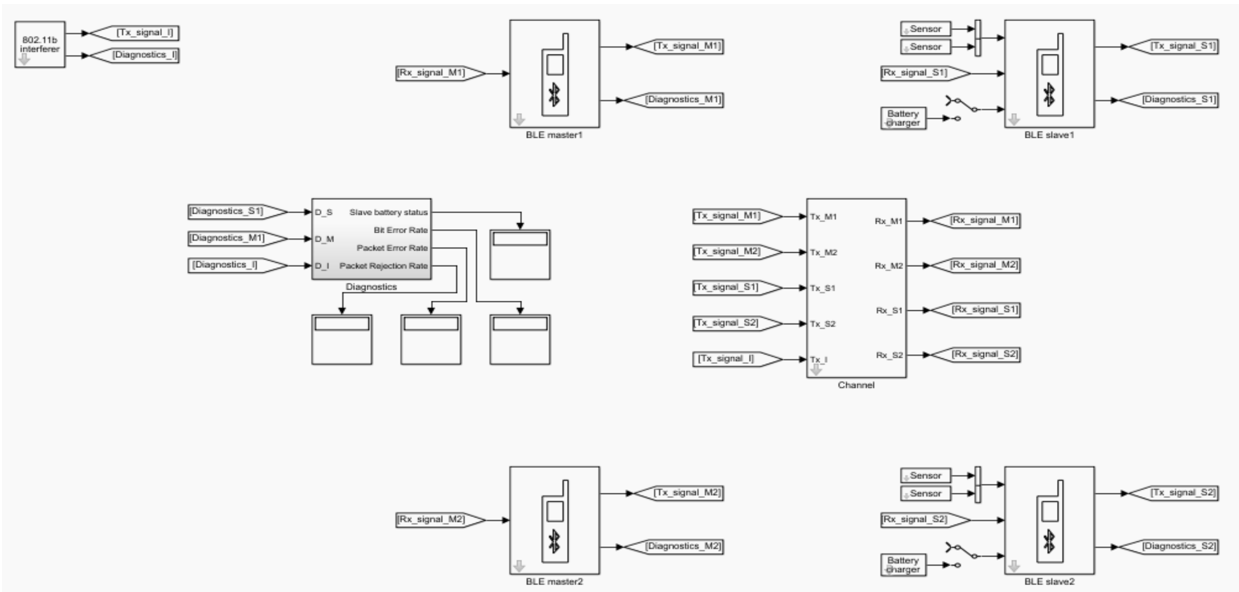



*Figure 27. The BLE model consisting of two masters and two slaves and one interferer located according to their given coordinates*

For each sensor, the number of bits per sample can be specified. The sensors are connected to the slave device, which determines the sampling instances of the signals to ensure low latency between data acquisition and reception sides. The number of sensors can be varied by connecting or removing them from the concatenating vector.

For the slave device, the following parameters are configurable:

- Protocol data unit (PDU): ADV_IND or ADV_DIRECT_IND, which is sent by the slave in Advertising State and received by the master in Initiating State ([1] Part B 2.3);
- Advertiser's address: public or random ([1] Part B 1.3);
- Time between two consecutive PDUs ([1] Part B 4.4.2.3);
- Advertising interval ([1] Part B 4.4.2.2.1);
- Sampling parameters: sampling rate of the signals from the sensors, data size in one data packet, number of packets per connection interval, connection interval – all these numbers are provided in sets due to they must conform with each other to ensure uniform sampling at rates higher than 0.133 kHz;
- Battery parameters: voltage, capacity, idle state power consumption, energy spent per bit sent, energy spent per bit received.

One slave is allowed to communicate with only one master device simultaneously.

For the master device, the following parameters are configurable:

- Initiator's address: public or random ([1] Part B 1.3);
- Scan window ([1] Part B 4.4.4) – master enters the Initiating State and starts to listen on the primary advertising channel for the duration of the scan window for a connectable advertisement;
- Scan interval ([1] Part B 4.4.4);
- Inter frame space ([1] Part B 4.1.1) – the time interval (150 us) between two consecutive packets on the same channel index;
- Connection interval ([1] Part B 4.5.1) – time between the start of two consecutive connection events (must conform to sampling parameters set by the slave);
- Transmit Window Offset ([1] Part B 4.5.3);
- Transmit Window Size ([1] Part B 4.5.3);
- Connection Supervision Timeout ([1] Part B 4.5.2) – enables detection of link loss for fast termination of connections that fail to establish (both the master and the slave use a connection supervision timer, which resets upon reception of a valid packet, however, if the timer reaches six times the connection interval, the connection is considered lost);
- Connection Slave Latency ([1] Part B 4.5.1) – defines the number of consecutive connection events that the slave device is not required to listen for the master (not implemented, this parameter is set to zero – slave device listens at every anchor point, thus ensuring higher signal sampling rates);
- Master SCA ([1] Part B 2.3.3.1) – worst case master's sleep clock accuracy, which determines window widening for the slave device ([1] Part B 4.5.7) – not implemented.

One master is allowed to communicate with one or more slave devices simultaneously depending on the chosen data acquisition scenario. If the sampling interval of the sensor nodes exceeds the minimum connection interval of 7.5 ms, then multiple BLE slaves are supported, while only one slave is allowed if the sampling interval is less than 7.5 ms due to MD bit is exploited in the packets sent by the slave.

For the channel, the following parameters are configurable:

- Transmit power of BLE devices;
- SNR of the channel;
- Interferer (802.11b): on/off;
- Interferer power.

One simulation example of two masters and one slave device is shown in Fig. 21. From the messages on the right side of the screen it follows that both masters (M1 and M2) receive the advertising packet from the slave (S1) at the same time (0.131 ms); therefore both devices send their responses to S1 after 150 us and enter the connection state after the initiating packets are sent at 0.633 ms. The slave device successfully decodes the response from M1 at 0.633 ms and enters the connection state. After receiving the first data packets (S1 from M1 at 3.226 ms and M1 form S1 at 3.496 ms), the connection is established between S1 and M1, while M2 leaves the connection state at 45.633 ms due to not receiving the response from S1. The example also shows that when an interferer is switched on (the lower graph) and its frequency is close to M1 and S1 frequencies (the upper graph), then almost all the packets between M1 and S1 are lost.

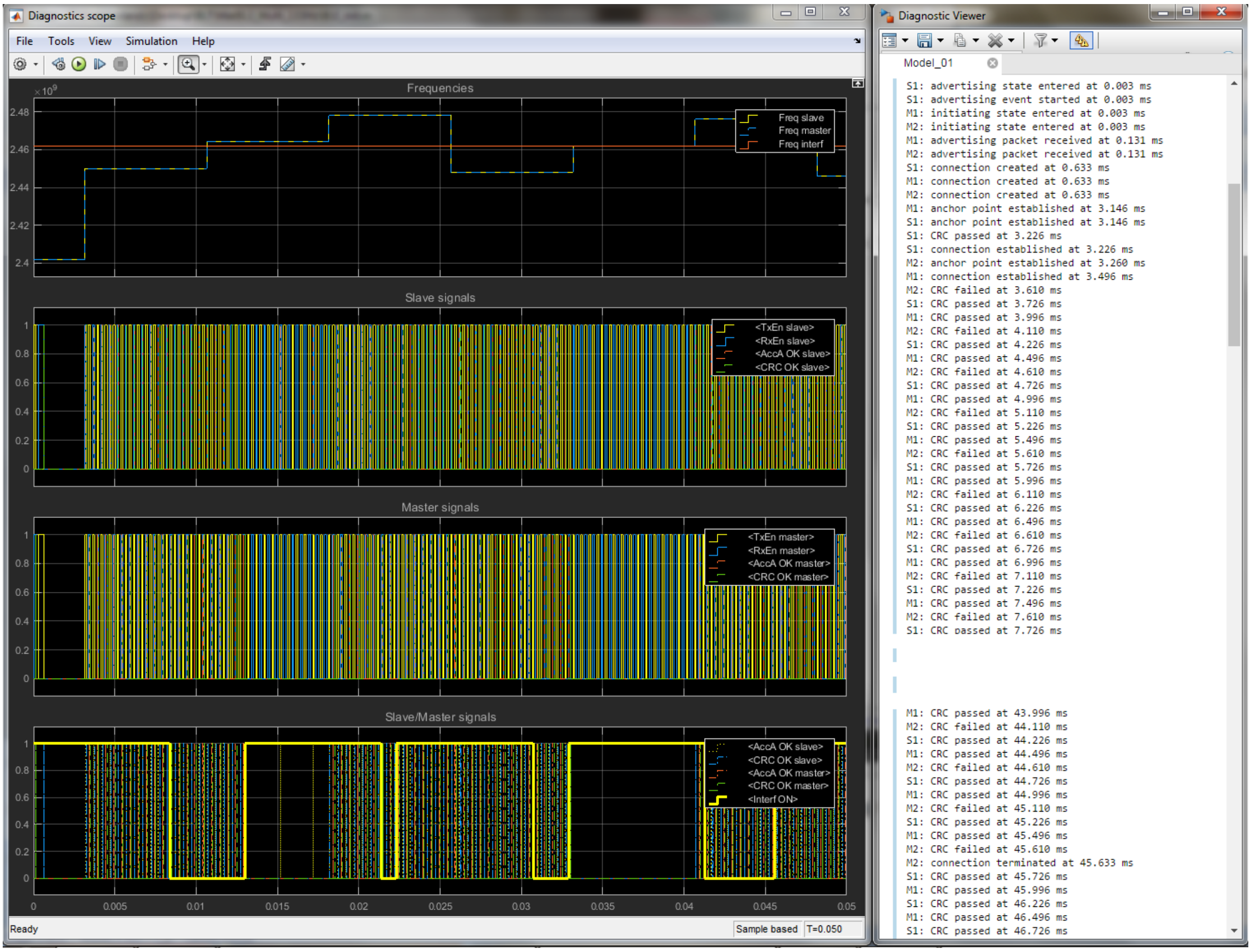



*Figure 28.: Diagnostics scope of simulation of two masters and one slave device with an interferer switched "on"*

## 5.2 Implementation aspects

As mentioned at the beginning of section 5, four different real-time wireless solutions were developed for different Pilots, Use-cases and demonstrators. Still, all of them are sharing these key functionalities. In this section, we show how those wireless solutions were further developed, deployed and tested in different Pilots, Use-cases and demonstrators.

### 5.2.1 Pilot 2 "12 Inch wafer stage"

With the aim to increase the overall performance of the system, as can bee seen in Fig. 22, EDI real-time wireless sensor is used in the 12-inch wafer stage, which is part of the Nexperia ADAT3 pick-and-place platform. Wireless sensor is mounted on the moving part of the Pilot 2. 2 axis acceleration data is transmitted to gateway, which converts +/-2g acceleration data of 12 bit precision to analog voltage output to 0...1.8 volt range.

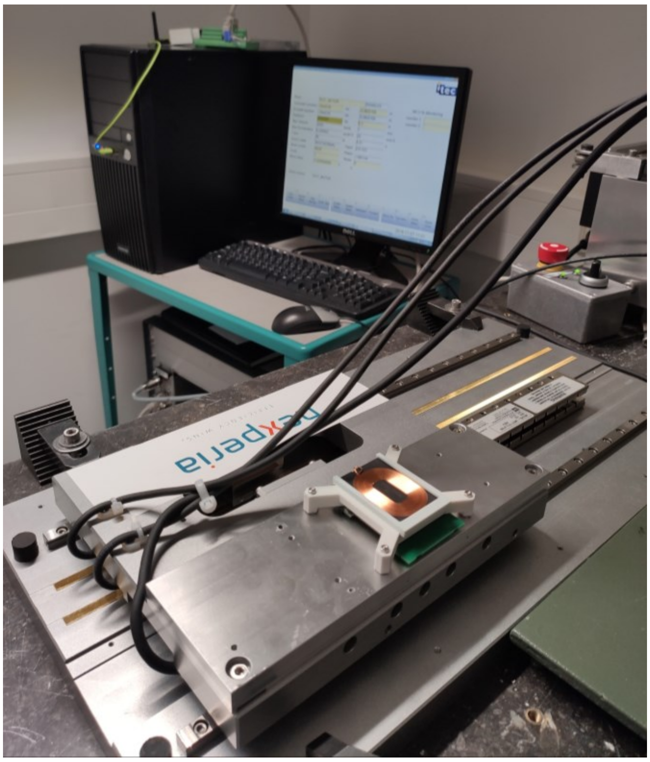


*Figure 29. EDI real-time wireless sensor deploy on Pilot 2*

Analogue data is further used to represent the responsiveness of motor encoders. Together with processing latency, the delay between data measure and display is 5ms. It is feasible for encoder monitoring, but could be vastly increased if digital communication and fast digital to analog converter would be used instead. In order to power the wireless sensor contactless energy transfer feature is used.

### 5.2.2. Pilot 4 “Big CNC”

In Pilot 4, IKE real-time wireless sensor software and hardware modules are used. The former has been reported in [8], while here latest results are reported. The following picture shows the instrumentation layer needed for Pilot 4:

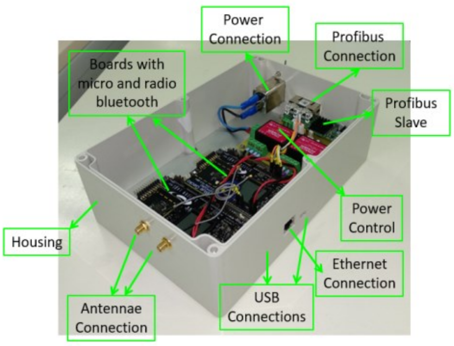



*Figure 30. IKE real-time wireless sensor*

The main components here are the boards with micro and radio Bluetooth. These operate as bluetooth slaves commanded by the Bluetooth master integrated in the emitting node (BB1). These are responsible for receiving the measured data and send them to the PLC of the CNC machine.The profibus slave is controlled by the profinet master that is in the PLC machine. It takes the data from the bluetooth slaves and convert them into something readable for the PLC of the CNC machine.

Once the BB is ready for operation, the next step is the BB integration and Pilot adaptation. In the case of Pilot 4, due to size constraints, the instrumentation layer of BB1 and BB2 had to be physically re-designed with a special housing according to the geometrical limits that the CNC machine implied.

The next picture shows in the left the housing developed for the instrumentation layer of BB2 once it has been integrated in the machine. In the right, the whole setup, including BB1 and BB2 is also shown in a real working environment.

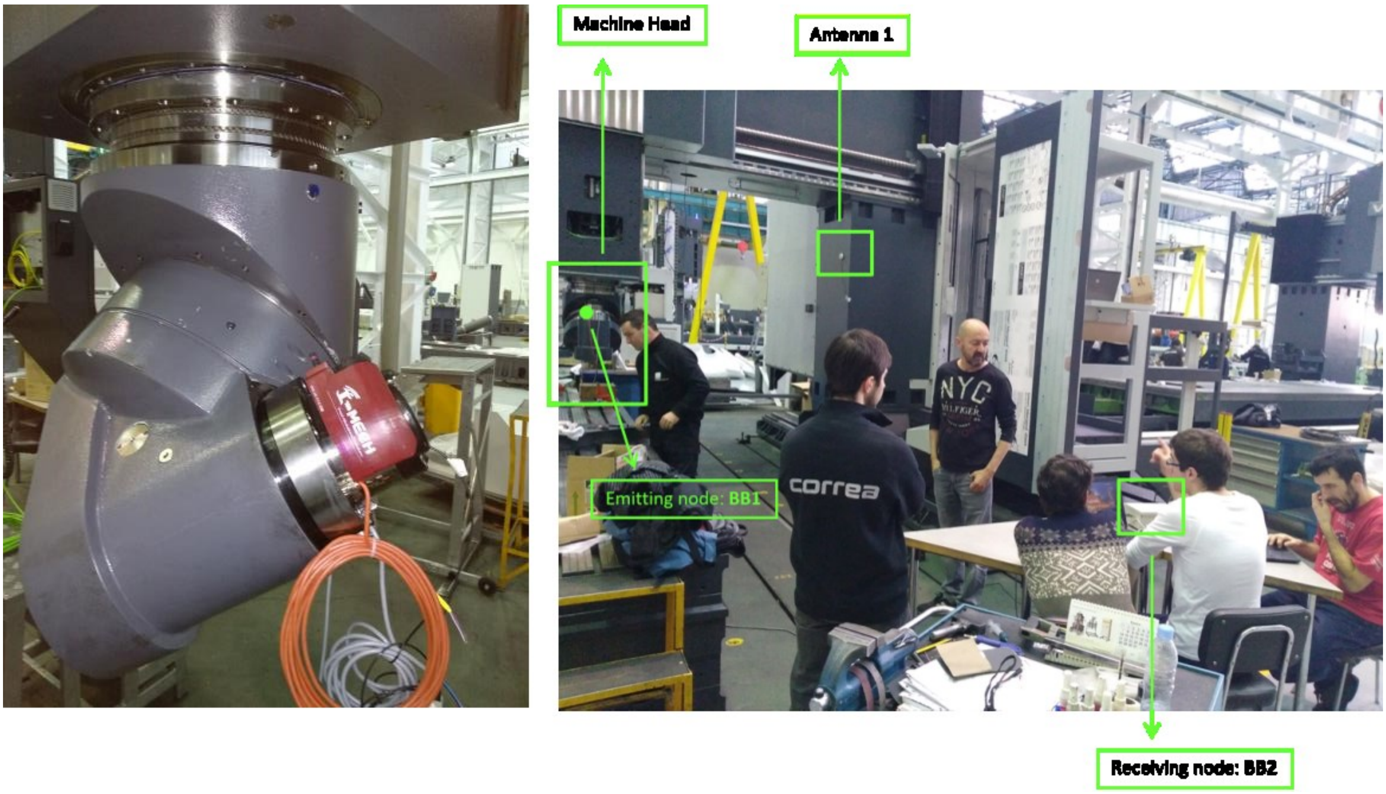



*Figure 31. Pilot 4 setup*

The block diagram with the setup for the Pilot including the instrumentation layer is shown below.

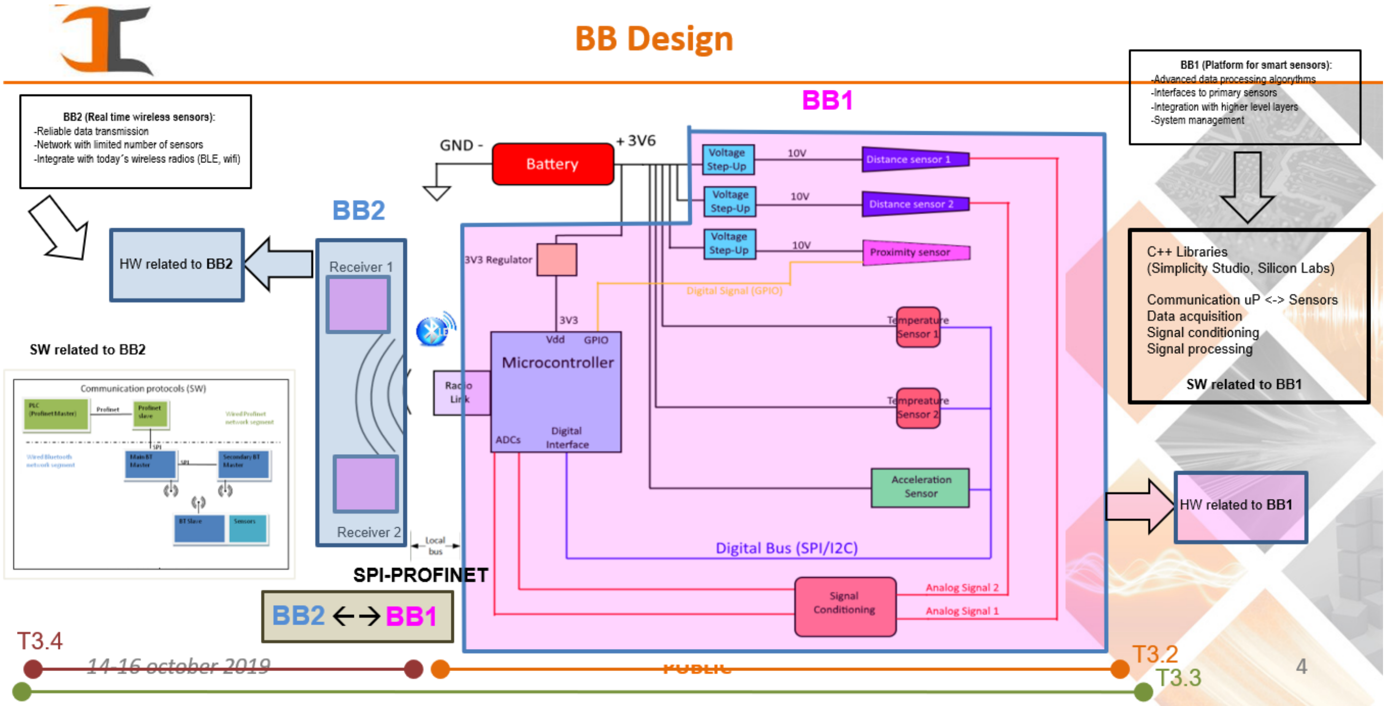



*Figure 32. The block diagram with the setup for the Pilot 4.*

BB1 is responsible for taking the data from the sensors and collected them in the microprocessor after signal processing and conditioning. BB2 in contrast, is responsible for sending the data wirelessly from the microprocessor placed in the rotating head of the machine to the PLC of the CNC.

Communications in this environment is a difficult task, since the head is always rotating and there are flying metallic particles resulting from the machining process that impede the electromagnetic waves from effectively reaching the receivers. This is the reason why special algorithms with time and spatial redundancy were developed. Therefore, the most critical aspect of the Pilot is testing if the signal is indeed reaching the PLC of the machine.

The next figure shows the signal coverage at the PLC when the head is continuously rotating in a normal machining process. As it can be seen, the communication is never lost and the coverage is acceptable.

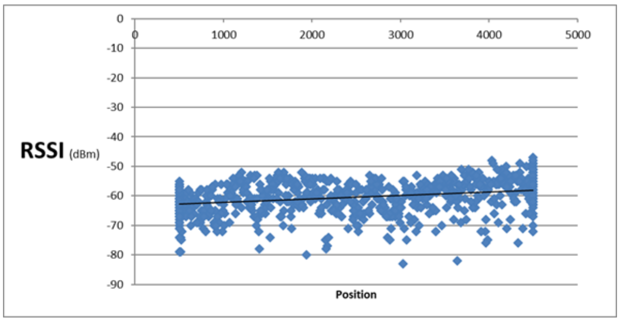



*Figure 33. Signal coverage at the PLC when the head is continuously rotating in a normal machining process*

### 5.2.3 Use-Case 1.1 "Power electronic for hoist and crane sector"

In this use-case, EDI+ZAPUNI real-time wireless sensor enables an active anti-sway feature in the Gefran Hoist and Crane. Now, the system is able to recognize acceleration and angle position of the load, thus allow to improve displacement time; provide ability for non-skilled operator to use the machine safely and with higher performance; and eliminate load oscillations. The overall system will consist of four main components: EDI+ZAPUNI real-time wireless sensor, wireless gateway, algorithms (BB7 “Vibration control module”) and the crane itself + corresponding systems (Gefran) (drives and mechanical parts).

*Figure 34. EDI+ZAPUNI real-time wireless sensor deployed on a crane.*

The wireless sensors are used to supply accelerometer and gyroscope data to BB1 using the RETIS proprietary communication protocol developed by ZAPUNI. Each sample consists of 3-axis accelerometer data and 3-axis gyroscope data obtained with 100Hz frequency. During the tests vibration noise from motor and overall system was measured and conclusion was made that mount and sensor used in combination with BB7 “Vibration control module” algorithms was sufficient for active crane balancing. Below raw accelerometer sample data from tests carried out can be seen. Great noise areas is where motor starts its movement - after stopping, sway movement clearly can be seen event without any filters applied.

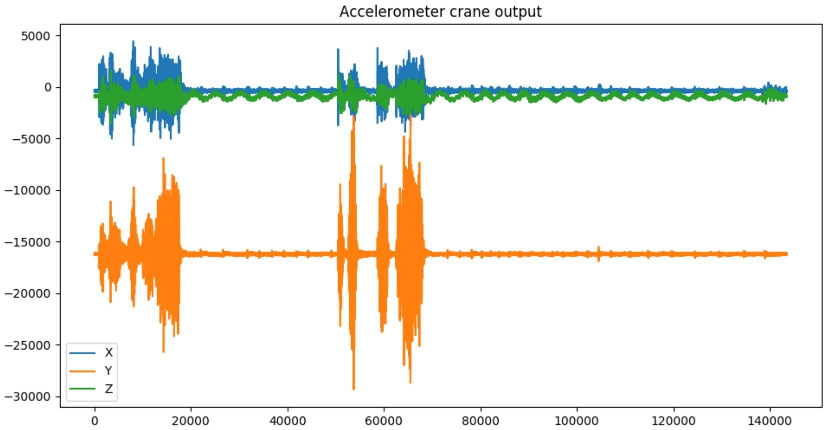



*Figure 35. Accelerometer measurements from UC1.1*

### 5.2.4 Demonstrator 1 "Process Monitoring and Predictive Maintenance for LSM"

In Demonstrator 1, TNI real-time wireless sensor solution was developed to monitor the health of product carrying pallets in the contact lens manufacturing process with a view to enabling a predictive maintenance model for the transportation layer, namely the Magnemotion Quickstick Linear Synchronous Motors (LSM). The parameters identified for the condition monitoring were the vibration experienced by the pallet, the magnetic field strength of the LSM system relative to the lens carriers, the temperature of the product on the pallet through the various process steps and the light intensity experienced by the lenses as they pass through the blue light curing tunnel. All of the identified sensor parameters had to fit the wireless communication requirement, as the moving carriers necessitated a wireless data transfer. The sensor solution was also required to have an operating life in the region of weeks/months and so an integrated energy harvesting coil was developed to recover electromagnetic energy from the LSM as the carriers cut the magnetic field.

At this point of the I-MECH project the wireless sensor platform and energy harvesting coil have been tested on the Magnemotion Quickstick LSM system of industrial production lines in J&J Vision Care, as well as new production lines which have been built but have yet to be validated. The tests carried out during this phase were:

- Position tracking algorithm
- Smart data transmission
- Energy harvesting circuit

The testbed used during this demo is a new build of the 5GT line model currently used in J&J Vision Care. The line is currently in the SAT phase of the build and will be commissioned and validated in early 2020. The line is made up of 3 shuttles (their movement is always vertical) which move the vehicle from one LSM track to the others and 9 LSM tracks (their movement is always horizontal). Four of these 9 tracks are used for the blue light curing tunnel (these tracks work in parallel) and the rest are generic LSM tracks. The main purpose of the position tracking algorithm is to identify when the vehicle has entered the blue light cure tunnel. This is important because during this time, the vehicle stops at the entrance of the blue light cure for approximately 15 seconds (see figure 28), thus, if the wireless platform is able to identify this window it is possible to send all the necessary data and given that the WiFi Access Point is located in close proximity of this spot, the data loss is highly reduced.

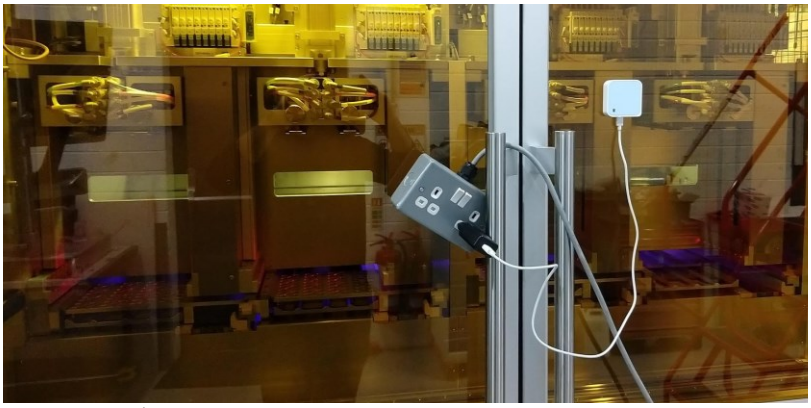


*Figure 36: Entrance of the 4 UV tracks and AP connected as close as possible to the entrance*

The idea behind the algorithm is that by integrating the acceleration data is possible to detect a positive or negative acceleration on a given axis. In particular, using the y axis (direction of travel of the shuttles) is possible to detect the acceleration of the shuttle and to define the direction of the acceleration. A negative and a positive acceleration look like in figure 29.

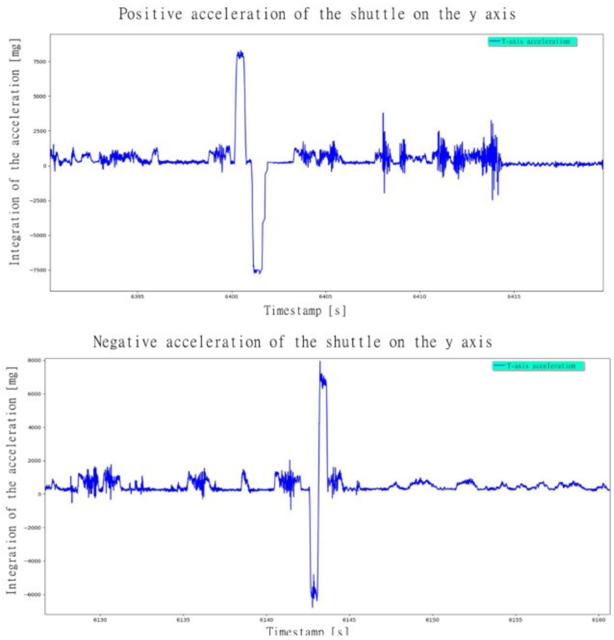



*Figure 37: On top, a positive acceleration seen in the data, on the bottom a negative acceleration*

Thus, the shuttles are easy to detect, the direction of movement of each shuttle is known, hence it is possible to identify every single shuttle on the line. Once a shuttle is identify it is also possible to identify which track comes next. Following this logic, the wireless sensor system is able to locate itself on the line without any external input. In this way, when the wireless sensor estimate its position to be at the beginning of the UV tunnel, it sends the data to the server. During this demo, the only data sent from the embedded system is the log of the current position and the log of the maximum value that the energy harvesting has found for every loop. The data was sent at the correct moment which proved that the algorithm was working correctly. Furthermore, Fig. 30 shows that the algorithm was able to properly detect the different shuttles, where the blue data is the acceleration data and the yellow data is the  estimated position of the algorithm.

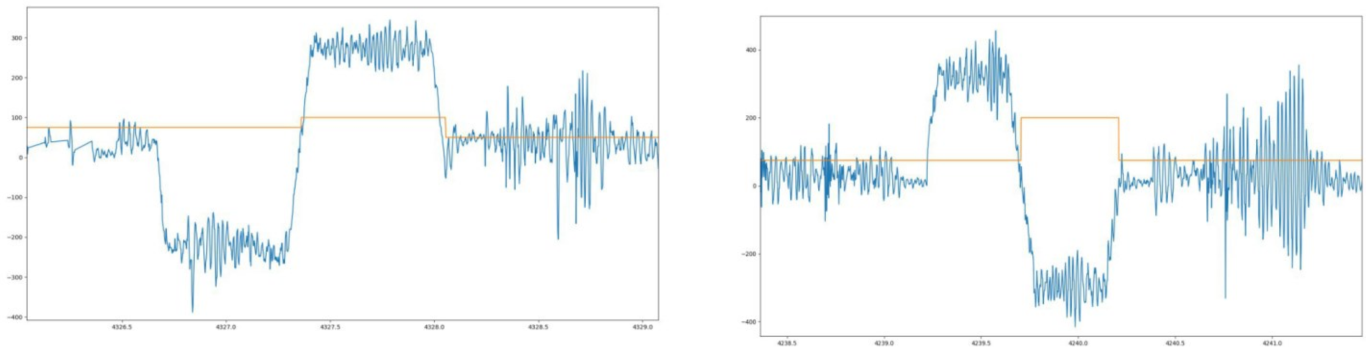


*Figure 38: On the left a shuttle moving in a negative acceleration, on the right a shuttle moving in a positive acceleration*

Finally, the energy harvesting circuit's purpose for this demo was to validate that the energy harvesting is feasible and that the circuit will turn on even at the lower voltages. For this reason, the harvesting circuit wasn't charging a battery but it was transforming the magnetic field into voltage, rectifying the signal and storing the data. This data showed that the circuit turned on as planned and the maximum peak voltage rectified by this version of the energy harvesting circuit was 140 mV whereas the average voltage was 18 mV. This results are impressive if is considered the low speed of the vehicle and the little available space on board of the vehicle to host the coil.

### 5.2.5 Demonstrator 2 "Injection mold industry"

In Demo 2, INL wireless sensor was implemented with 3 types of sensors in order to control two types of predictive maintenance, the first one during injection of plastic parts, and the second one during the storage of the tool considering is corrosion.

The following photo provide the location of the three sensors and their description, the force sensor, the vibration sensor and the corrosion sensor. All the sensors send data through bluetooth as explained in the previous points, to a computer where is possible to see the live data.

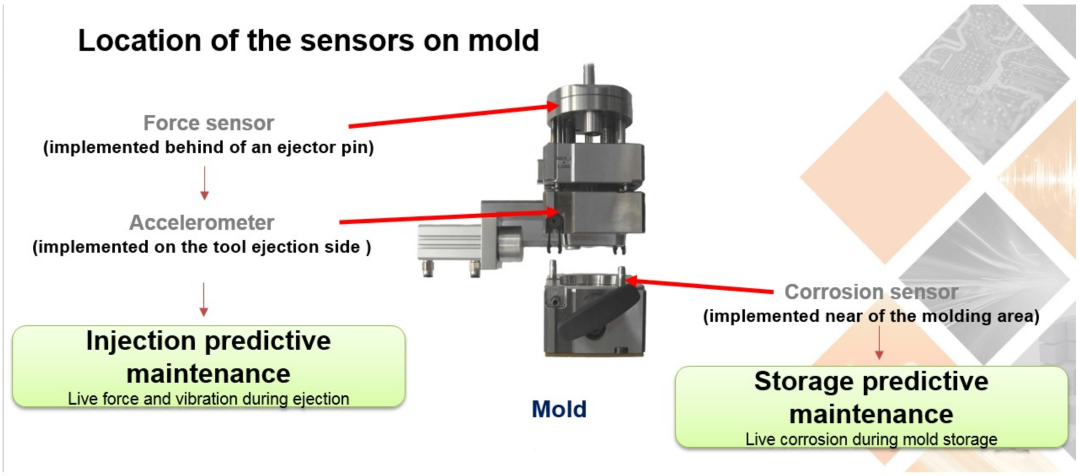



*Figure 39. Demonstrator 2 setup with 3 sensors.*

All the sensors send data through bluetooth as explained in the previous points, to a computer where is possible to see the live data. The following photo shows the type of data for each sensor.

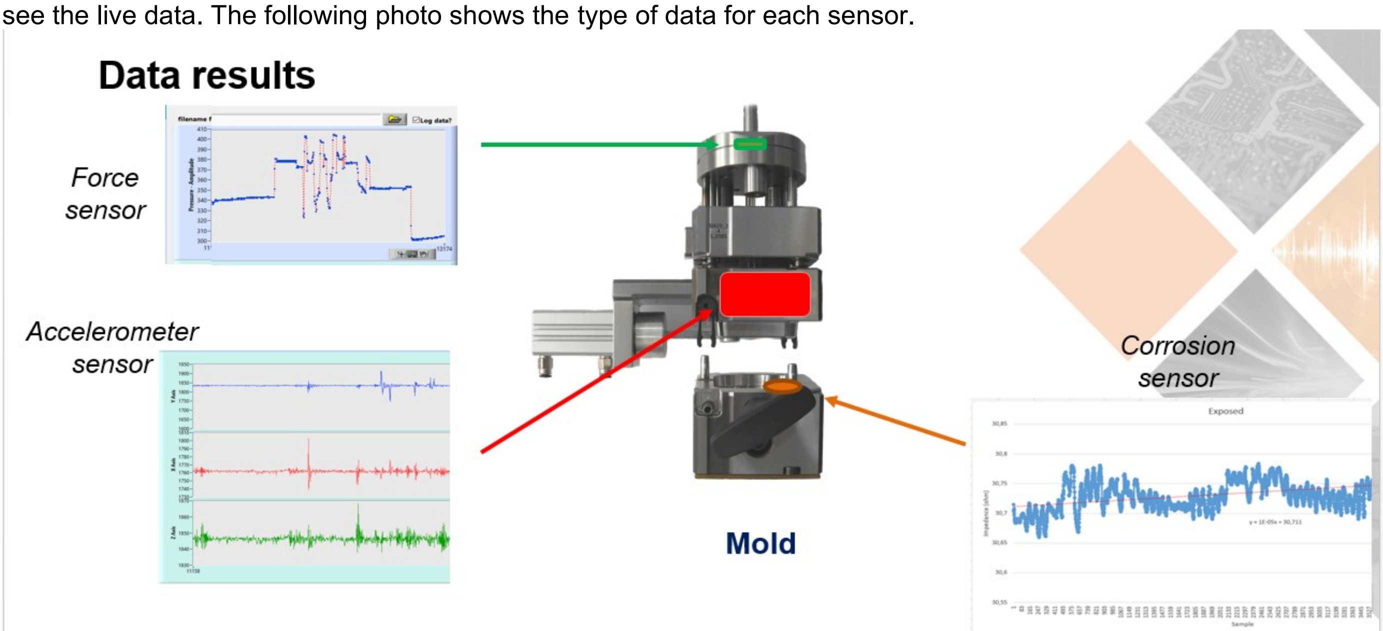



*Figure 40. Demonstrator 2 setup with measurements from 3 sensors.*

# 6 BB4 - High speed vision (TNO)

In previous documents D3.1 [5] and D3.2 [6] requirements derived from the Pilots and Use Cases have been transformed into requirements to the various Building Blocks (BB). The designed implementation of BB4 High speed vision has been described in detail in D3.5 [9]. Key performance parameters in visual servoing is low signal latency. This means latency as measured at the start of the exposure of camera pixels, ending at the actuation of mechanical output quality. Excess latency means excessive phase lag which in turn limits stability and control bandwidth of control loops. This is a major technical challenge, since camera information is expansive and data throughput is expensive in terms of system cost price. Because the cost and performance requirements of Pilots and Use Cases have a wide variation, three distinctive systems have been designed and built. For reference see D3.5 [9].

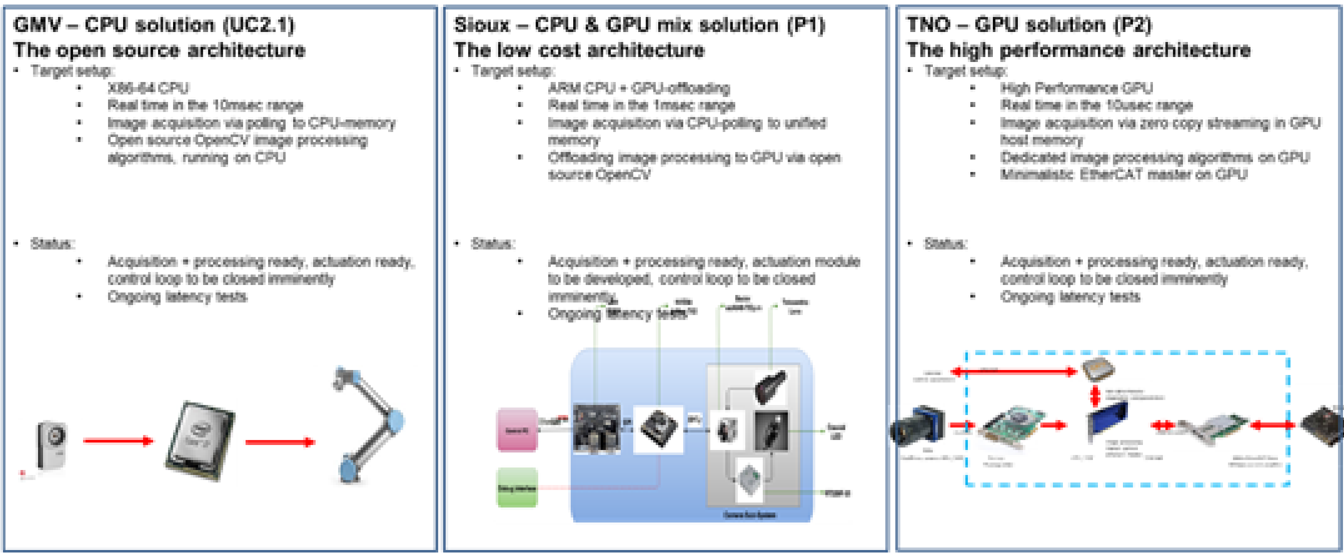



*Figure 41. Three BB4 high speed vision implementations to three sets of requirements*

Systems developed in BB4 may be applied as smart sensor over Layer 2 or as low level autonomous controller receiving position set points from Layer 2 as illustrated in Figure 33.

The work in BB4 during year 2 concentrated on realizing and optimizing technology. Progress of the optimization of the signal latency over the year is visualised in the figure below.

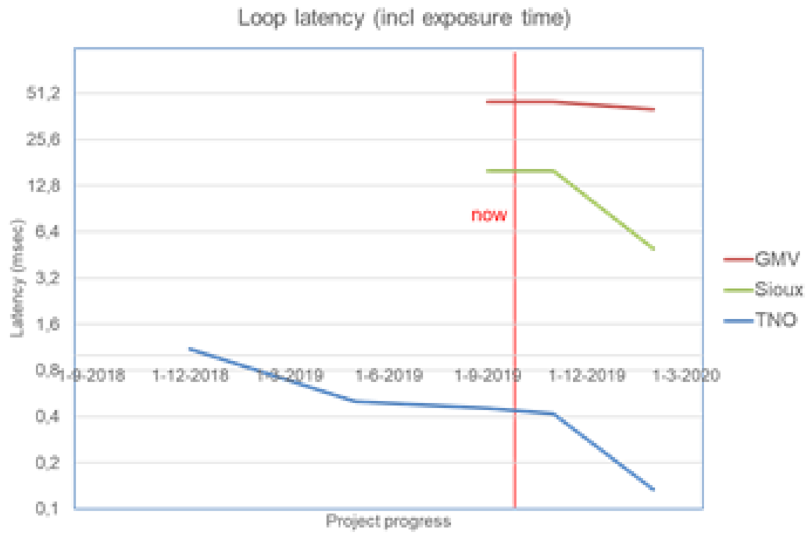



*Figure 42. Three setups with scalable performance*

# Functionalities

### Description of The Application by Sioux CCM for Pilot 1

The control architecture as developed by SCC (Sioux CCM) is based on a standard USB3 camera in combination with a Nvidia Jetson processing board. The scope and functionality of this BB4 configuration is well-defined as a high-speed smart vision sensor: A layer 1 device providing sensor data to a controller embodied in layer 2. In the case of Pilot 1, the sensor data consists of the absolute X- and Y-position of the belt (measured at two locations). This sensor data is thus used to realize X, Y and Rz position feedback to the controller.

The sample rate and latency of this BB4 implementation are insufficient to realize a stable velocity control of the belt speed. For this purpose, the rotary incremental encoders on the motor shaft is used. The complete controller design is depicted in the figure below.

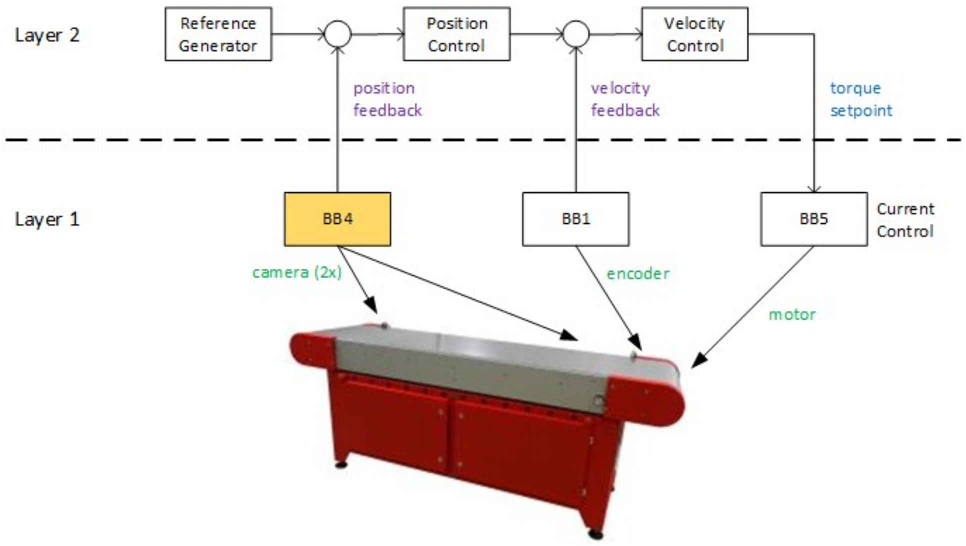
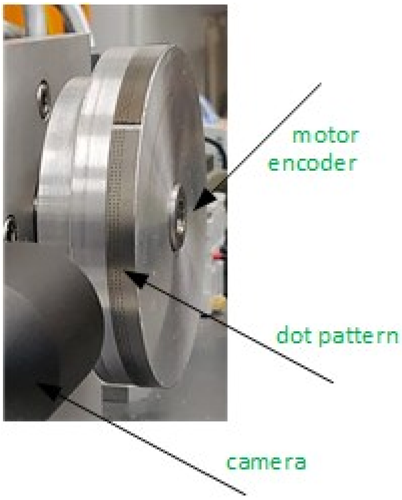


*Figure 43. The figure above shows the envisioned final implementation of BB4 for Pilot 1.*

So far, this configuration hasn't been realized, it involves the application of a specific dot pattern on the belt. For practical and operational reasons, this pattern hasn't been realized yet.




What has been realized is a demonstrator setup to show the essential core functionality of this BB4. This demonstrator setup consists of a single camera observing the same dot pattern as designed for the belt, which has been mounted on a rotary wheel instead. So basically the same Layer 1 components are present in the demonstrator, and the position loop can be closed in the Layer 2 controller of the demonstrator.

Note: in this demonstrator, the wheel is mounted with a small wedge, such that the horizontal dot pattern position starts to 'wobble' while the wheel is rotating. The camera is mounted on a horizontal moving slider, and will try to 'follow' the pattern, resulting in a sinusoidal motion.

**Description of The Application by TNO for Pilot 2**

The control architecture as developed by TNO (GUAPA - GPU Accelerated Acquisition, Processing and Control Architecture) enables general purpose Linux systems to offload data streams from the CPU/OS to a parallel data path. This is to be demonstrated on a hardware in the loop demonstrator to validate the requirements of Pilot2 – 12" inch wafer table. The system comprises of a high speed camera that is connected to a frame grabber via a 4 channel CoaXPress-link to a digital control computer that emits current-values to current amplifier (BB5). The demonstration is to move over a 2D grid by means of a visual servoing system, such as required for moving over a diced semiconductor wafer with the intent of pick and place. System, wafer pattern and basic signal processing are depicted in the figure below.

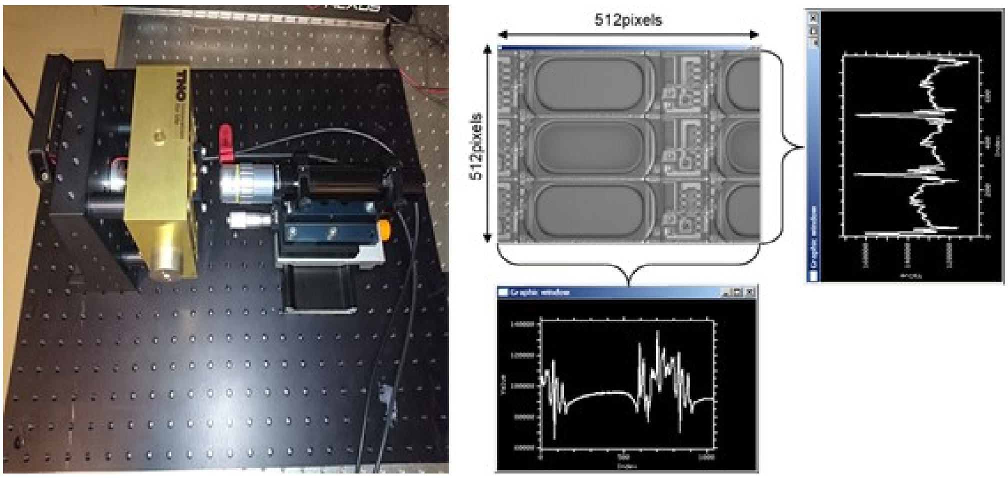



*Figure 44. System, wafer pattern and basic signal processing*

System and signal processing intermediate results of the setup by TNO

The benefit of the GUAPA system is that control-pipeline from sensor to current-amplifier is decoupled from the host-CPU, thereby eliminating the need for a real time operating system. GPU's with several thousands of calculation engines and vast memories allow for high performance control and access to signal- and image-processing libraries. See Figure below.


*Figure 45. Setup for BB4*

**Description of The Application by GMV for Use Case 2.1**

platform-art© is the GMV's dynamic HIL test bench used for validation of space GNC systems, and the Use Case 2.1 defined for I-MECH. In platform-art©, two industrial robots simulate the dynamic behaviour of two spacecraft in rendez-vous and docking operations. The sensors on-board the spacecraft mock-ups mounted on the industrial robots produce the same measurements as in space environment, allowing their validation. Also, the spacecraft mechanical devices (docking devices, manipulators, etc.) can be tested in an environment that simulates the dynamics of real spacecraft.

platform-art© is composed of the following main subsystems:

- Target: A mockup of the serviced space vehicle mounted on one of the industrial manipulators, simulating the dynamics of the serviced spacecraft.
- Chaser: Validated equipment mounted on the second industrial manipulator, simulating the dynamics of the servicing, active chaser spacecraft. The validated equipment can be composed of sensors (cameras, LiDar, etc.) and/or a gripper/docking devices (including manipulators) to be validated.
- Control system based on manipulators controllers (Kuka), DSpace (for simulation of dynamic behavior in space) and RTOS motion controller (for real time control of the system).
- Illumination system: A third Cartesian manipulator equipped with is used to generate realistic illumination simulating the sun.

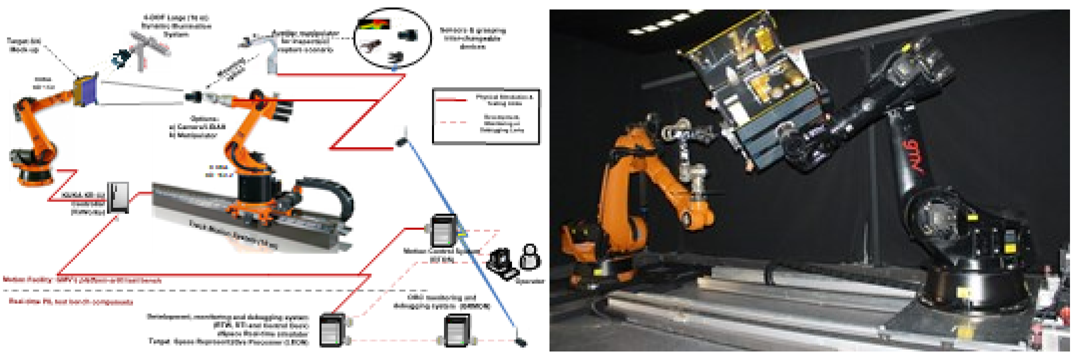


*Figure 46. Left: platform-art© subsystems. Right: platform-art© setup in a berthing application, where the chaser manipulator (orange) is equipped with a berthing manipulator, and the target manipulator (black) is equipped with a satellite mock-up.*

platform-art© is used for simulating activities that include the precise approximation between space vehicles or between a space vehicle and other bodies in tasks like rendez-vous, space debris removal, automatic assembly of large structures, etc. In order to simulate these tasks, a camera, a LiDar or similar sensors are installed in the chaser manipulator. The final approximation in these applications requires a controller capable of implementing vision based guidance algorithms. In some docking/berthing applications the chaser spacecraft is equipped with a docking manipulator and in this case, visual servoing or similar algorithms are used to guide the berthing manipulator during the final approach phase.

platform-art© can also be used to simulate activities that imply contact between space vehicles, such as the final phase of a docking/berthing operation. In this case, an impedance controller can be used to generate a compliant behavior in order to achieve a safe and stable berthing. However, vision-based control algorithms can still be used in this case for achieving a correct alignment of the berthing mechanisms until the operation is completed.

## 6.2 Implementation aspects

### 6.2.1 Pilot 1 “Generic substrate carrier (GSC)”

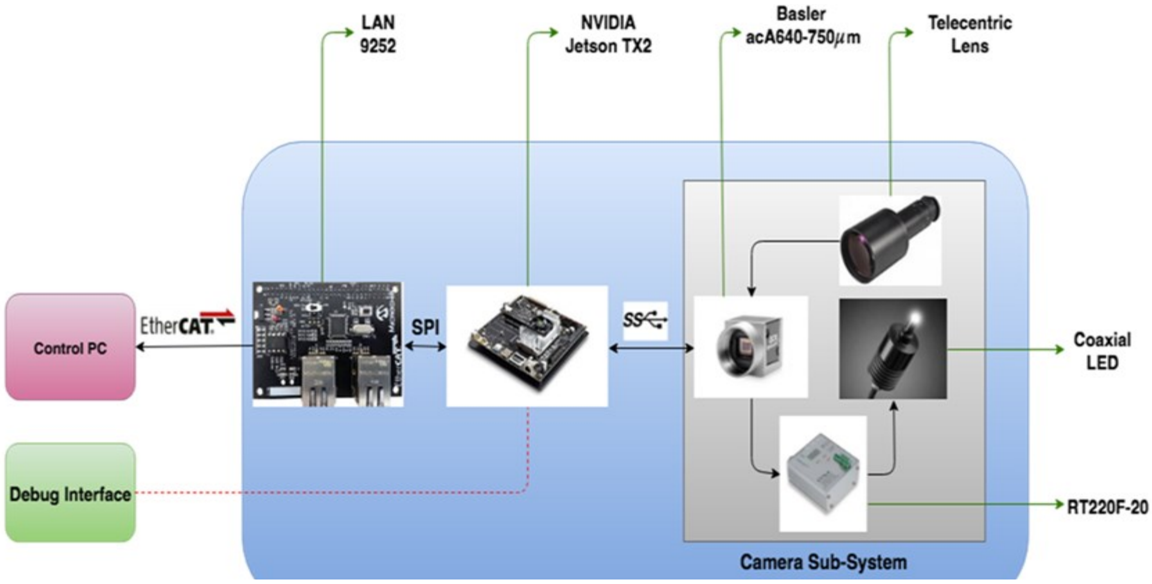
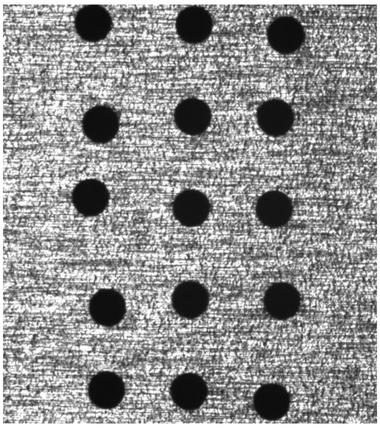



*Figure 47. Pilot 1 setup.*

For Pilot 1, the Jetson TX2 module is selected with a Basler USB3 camera (type: acA640-750um). Such a system can achieve the specified frame rate of 1 kHz. The total closed-loop latency is targeted around 3 ms, so the position control bandwidth limit will then be around 10 Hz.

The Jetson TX2 module is connected to an ESC (EtherCAT Slave Controller) via the SPI bus. The ESC is configured with the necessary data objects for the application; the most relevant are the X- and Y-positions that are available in the PDO (Process Data Objects) of the EtherCAT data packets.



The USB3 camera is a Basler, acA640-750um, which has a sensor with VGA resolution (640x480). In order to increase the frame rate, a ROI with 256 rows is read. Thus, the camera is able to capture and transmit images at roughly 1000 fps.

The lens is designed to show a region of 5 mm vertically, enough to show 4 consecutive rows of pattern dots (actually, a 5th row is visible, but this row is allowed to cross the image boundary). Following the ROI resolution, the pixel size is 20 µm.

Finally, the pattern is illuminated with a flash LED: The LED is driven by a special LED driver. This driver is sourcing ultra-short (~7.5 µs) ultra-high (~20A) current pulses, such that the motion blur becomes insignificant while the image intensity remains sufficient for an adequate camera exposure.

The image processing algorithm measures each dot position with subpixel accuracy.
After calibration of the camera optics, a dot position accuracy of less than 1 µm is obtained. This algorithm is running on the CPU during the development, but since the computation is time-consuming (~14 ms) the algorithm must eventually be transferred to the GPU. This image processing off-loading to the GPU is done via the open-source OpenCV library.

Table 1. Achieved and targeted latencies for Pilot 1.

| Item | Actual latency | Target latency |
|---|---|---|
| Exposure | < 0.01 ms | < 0.01 ms |
| Image readout | 1 ms | 1 ms |
| Image processing | 14 ms | 1 ms |
| EtherCAT + control | 7 ms | 0.4 ms |

The actual image processing algorithms are still running on the CPU, and are not optimized for speed. By off-loading to the GPU as well as by refactoring and tuning the algorithms for optimal speed, the target processing latency should be obtained.

The large EtherCAT latency is due to the current SPI bus implementation between the Jetson processor and the ESC (EtherCAT Slave Controller). This is mainly a limitation of the Jetson SPI software driver, but possibly also of the Jetson SPI hardware. In order to reduce this latency to the desired target value, either the Jetson SPI kernel driver has to be improved or a different communication bus between the Jetson and ESC must be chosen.

### 6.2.2 Pilot 2 “12 Inch wafer stage”

A high speed vision system has been developed during year 2. This system comprises of the following components.

**Camera.** The camera employed for visual servoing is an Optronis CP80 equipped with a 3 megapixel monochrome sensor. In addition to the high image rate in global shutter mode, it allows for reducing the region of interest in both directions, allowing for frame rates up to 20KHz. The pixel sensitivity of 3.8V/lux.S is enhanced due to micro lenses.

**GPU/Host.** The GPU is selected on its ability to allow sustained DMA into memory. Due to 1792 cores and 8GB of memory it is capable of more than 5 TFLOPS, assisted by the 64 GFLOPS of the Intel Core-I7 host computer. This GPU perform all image processing and control algorithms and in addition it hosts the EtherCAT master.

**Amplifier.** The utilized amplifier is an EtherCAT-connected current amplifier as defined in BB5. Maximum update rate is above 5KHz, whereas the signal latency from input to current-output is 100usec.

**Manipulator.** The manipulator is a 2-degrees of freedom translation stage, designed by TNO in 2015. The linear guides are formed by two wire sparked leaf spring sets in parallel configuration. The body is actuated by two Bei Kimco voice coil actuators. First eigen-frequency of the motion stage is over 250Hz.

**Illumination system.** Since the camera exposure time is very short (50usec) a high intensity LED + parabolic reflector has been applied to illuminate the scene.

**Software configuration.** Software comprises of configuration- and launch files on the Linux X86-64 host computer; starting the camera and the dataflow. A multitude of parameters may be set via JSON-files.

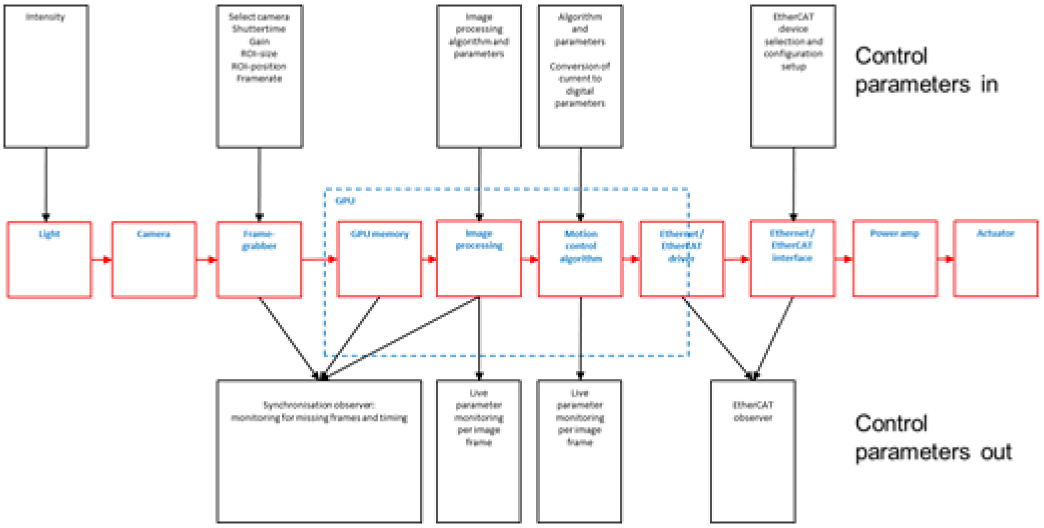



*Figure 48. Signal flow and configuration parameters*

After launching the vision and control application, the host monitors and visualizes the live process on the GPU.

**Signal processing and control.** The signal processing as applied as yet comprises of integrating the image into a vertical and horizontal line. These patterns are cross correlated to a reference as to detect sub-pixel shifts. These two signals are compared to a setpoint on two SISO PID controllers.

**Software development and performance tuning.** Software performance has been benchmarked during development as to balance optimization according toTable 1. Where possible open source software has been utilized.

Table 2. Latency optimization during the 2nd year for Pilot 2

| 512^2 design | | | | Target 2018/6 512^2 | Verification 2018/12 512^2 | Redesign 2019/7 512^2 | Verification 2019/8 512^2 | Outlook 2019/11 512^2 | Outlook 2020/1 512^2 | |
|---|---|---|---|---|---|---|---|---|---|---|
| # | Description | Size (byte) | Bandwidth (Mbyte/sec) | Latency (usec) | Latency (usec) | Latency (usec) | Latency (usec) | Latency (usec) | Latency (usec) | Conclusions / Comments |
| 1 | Camera sensor exposure | | | 50 | 50 | 50 | 50 | 50 | 50 | Camera settings application specific |
| 2 | Camera readout | 262144 | 1280 | 205 | 205 | 204 | 204 | 204 | 202 | 2019/11 system is bandwidth limited by the camera |
| 3 | Framegrabber to host memory | 262144 | 6450 | 1 | 40 | - | - | - | - | Reduced to zero by design: now single copy DMA to GPU |
| 4 | Host memory to GPU memory | 262144 | 12800 | 1 | 5 | - | - | - | - | Reduced to zero by design: now single copy DMA to GPU |
| 5 | Image to parameters in GPU | 262144 | | 40 | 40 | 40 | 40 | 30 | 20 | 2019/11 Application specific. Optimization required |
| 6 | Motion PID | 8 | | 1 | 5 | 5 | 5 | 2 | 2 | 2019/11 Application specific. Optimization required |
| 7 | Motion setpoints to fieldbus | 45 | | 1 | 5 | 5 | 5 | 5 | 2 | 2019/11 CPU/OS involvement, to be replaced by DMA |
| 8 | Fieldbus 1000Mbs to 100Mbs rate conversion | 45 | | 1 | 1 | 1 | 1 | 1 | 1 | |
| 9 | Fieldbus to motor current | 45 | | 20 | 750 | 20 | 150 | 150 | 33 | 2019/11 change fieldbus protocol to UDP |
| | + latency sum (usec) | | | 325 | 1094 | 325 | 456 | 442 | 191 | |
| | maximum update rate (Hz) | | | 6762 | 6116 | 6902 | 6902 | 6902 | 9806 | |
| | green: verified by measurement | | | | | | | | | |
| | orange: high confidence estimation | | | | | | | | | |
| | red: low confidence model based estimation | | | | | | | | | |

## 6.2.3 Use Case 2.1 "On-ground validation of space GNC systems through the use of robotic devices"

A vision based control system has been developed as an implementation of IMECH BB4 [D3.5] according to the requirements specified for Use Case 2.1 in [D3.1] and [D3.2]. The following subsections describe this implementation.

**Hardware**

**Camera**. The camera employed for visual servoing is an Ueye from IDS, model UI-3013XC. The camera is controlled through a USB 3.0 interface with a USB 3.0 micro B connector and it is mounted on the tip of the UR5 through a mechanical interface. In the preliminary tests, a 1280 x 960 image is received in BGR8 format at a framerate of 15 fps and with an exposure time of 33.4 ms. It uses its own proprietary software to capture and receive the camera images through the USB port.

**Processor**. The computer used for vision processing and visual servo control is based on an Intel i7-6600U processor (2.60GHz) with 8GB of RAM memory. The operating system is Ubuntu 16.04. This computer interfaces directly the camera and the UR5 controller through its USB and Ethernet interfaces and executes all the image processing and vision-based control algorithms described in the next section.

**Robotic Arm**. The robotic arm used to move the camera and simulate the berthing manipulator is an UR5 from Universal Robots with the CB2.0 control box. It is a 6 DOF collaborative manipulator that can be controlled in position or speed in joint or Cartesian space. The arm receives joint speed commands with a frequency of 100 Hz. The communication with the controller is done through Ethernet with a TCP/IP connection.

**Software**

**Image processing module**. Two main approaches can be used to detect the visual features on the image allowing the execution of visual servoing algorithms: Fiducial Markers or Model Based Detection. The preliminary tests have been performed using markers due to the simplicity of this approach. A set of 6 total markers is distributed on the berthing interface. The marker type selected library is AprilTags, since it is a fast library with a low computational load. Its licence is BSD and it is integrated with OpenCV, a widely used computer vision library. For future tests, a 3D model based approach is planned to be used for cases in which markers cannot be placed in the target. The image processing module is called at the beginning of the control loop: a new image is captured and then this module is called to extract the features (2D positions of the corners of the markers). These features are then fed as the input of the visual servoing algorithm. In the tests performed, the time spent taking and processing the image is the main factor defining the overall control frequency.

**Visual servoing module**. An Image Based Visual Servoing (IBVS) algorithm in an eye-in-hand configuration is implemented by this module. The control loop input is based only on 2D features present in the camera image, and not on a 3D estimation of the pose of the target. However, the algorithm requires an initialization based on the desired positions of the features. The desired image coordinates and depth in the camera frame must be provided for each feature to model the movement of the camera so the current and the desired features can be matched. To initialize the algorithm, a procedure detects the markers, estimates their 3D poses and gets their coordinates on the image and their depth. With the information of the depth, the scaling problem is solved at the initialization phase and during the control, the algorithm does not need to estimate the 3D pose of the markers for each frame. This module has been fully developed during the activity and it has been coded in C++ language. The Eigen third party math library was used to perform matrix computation efficiently.

**Control module**. To convert the camera speed commands to joint speed, an inverse jacobian approach was chosen. The jacobian matrix can be obtained through the partial derivatives of the equations of the forward kinematics. That relates the joint and the cartesian space velocities, so the inverse of this matrix allows to transform the cartesian speed into joints for a given joint pose. If the matrix is not invertible, the Moore-Penrose pseudo inverse is computed.

The vision based control algorithm is implemented in two different threads:

- A slow loop (10Hz) is used to acquire images, process them (feature detection) and run the visual servoing algorithm.
- A fast loop (100Hz) is used to acquire the arm position, compute the Jacobian and generate the joint speed commands towards the UR5 robot.

# BB5 - High performance current amplifier (ING)

## 7.1 Functionalities

After gathering the largest common denominator of the requirements, the features that were finally implemented in the motion drive/ amplifier (BB5) are the ones appearing in this section.

At this point, is important to highlight that BB-5 can be used as a current amplifier like required in the i-mech project but can also be used as a complete servo drive closing the current, position and velocity loops by itself. Apart from this, the BB-5 HW leaves one of the processors open to be programmed by the end customer to give them the chance of implementing their own algorithms or closing the loops by themselves.

Table 3. The functionalities (features) of the BB5 amplifier

| Electrical and power specifications | |
|---|---|
| **Minimum power supply voltage** | 8 $V_{DC}$ |
| **Maximum absolute power supply voltage** | 80 $V_{DC}$ (continuous)<br>85 $V_{DC}$ (peak 100 ms) |
| **Recommended power supply voltage** | 12 $V_{DC}$ ~ 72 $V_{DC}$<br>This voltage range ensures a safety margin including power supply tolerances and regulation during acceleration and braking. |
| **Internal drive DC bus capacitance** | 30 µF |
| **Logic power supply voltage (optional)** | 8 to 50 $V_{DC}$<br>Providing the logic supply is optional, as the drive is supplied from the DC bus (single supply) on its full operating voltage range. When supplied from logic, an intelligent switch will stop consuming from the DC bus. |
| **Nominal phase continuous current** | 30 A |
| **Maximum phase peak current** | 60 A @ 3 sec<br>Active current limiting based on power stage and motor temperature. |
| **Efficiency** | Up to 98% @ 20 kHz, 80 V, 30 A |
| **Bus voltage utilisation** | > 97% @ 20 kHz, 80 V, voltage mode, no load |

| Motion control specifications | |
|---|---|
| Standby power | ≥ 2.5 W<br>Lowest standby losses measured with dual supply at 12 V logic, with an active Ethernet communication, and commutation turned OFF |
| Supported motor types | • Rotary brushless (SVPWM and Trapezoidal)<br>• Rotary brushed (DC) |
| Power stage PWM frequency (configurable) | 10 kHz, 20 kHz (default), 50 kHz & 100 kHz |
| Current sensing | 3 phase, shunt based current sensing. 16 bit ADC resolution. Accuracy is ±2% full scale. |
| Current sense resolution (configurable) | Current gain is configurable in 4 ranges:<br>• 2.475 mA/count<br>• 1.352 mA/count<br>• 0.570 mA/count<br>• 0.379 mA/count |
| Current sense ranges (configurable) | Current ranges for the 4 configurable current gains:<br>• ±81.1 A<br>• ±44.3 A<br>• ±18.7 A<br>• ±12.4 A |
| Max. Current loop frequency | 75 kHz |
| Max. servo loops frequency (position & velocity) | 25 kHz @ 75 kHz current loop |
| Feedbacks | • Digital Halls (Single ended)<br>• Quadrature Incremental encoder (RS-422 or Single ended)<br>• Absolute Encoder (RS-422 or Single ended): up to 2 at the same time, combining any of the following:<br>  ◦ BiSS-C (up to 2 in daisy chain topology)<br>  ◦ SSI |
| Supported target sources | • Network communication (EtherCAT or CANopen*)<br>*CANopen is the communication enabled by default. In order to use EtherCAT, the FW must be updated. |
| Control modes | • Cyclic Synchronous Position<br>• Cyclic Synchronous Velocity<br>• Cyclic Synchronous Current<br>• Profile Position (trapezoidal & s-curves)<br>• Profile Velocity<br>• Interpolated Position (P, PT, PVT)<br>• Homing |

| Inputs/outputs and protections | |
|---|---|
| General purpose Inputs and outputs | 4 x non-isolated single-ended digital inputs - 5 V logic level & 3.3 V compatible. Can be configured as:<br>• General purpose<br>• Positive or negative homing switch<br>• Positive or negative limit switch<br>• Quick stop input<br>4 x non-isolated single-ended digital outputs - 5 V logic level (continuous short circuit capable with 470 Ω series resistance) - 8 mA max. current. Can be configured as:<br>• General purpose<br>• Operation enabled event flag<br>• External shunt braking resistor driving signal<br>1 x ±10 V, 16 bit, fully differential analog input for load cells or torque sensors. Can be read by the Master to close a torque loop. |
| Shunt braking resistor output | Configurable over any of the digital outputs (see above).<br>Enabling this function would require an external transistor or power driver. |
| Motor brake output | 1 A, 50 V, dedicated brake output. Open drain with re-circulation diode.<br>Brake enable and disable timing can be configured accurately.<br>PWM modulation available to reduce brake voltage and power consumption. |
| Safe Torque OFF inputs | 2 x dedicated, isolated (> 4 GΩ, 1 kV) STO inputs (from 3.3 V to 30 V).<br>The STO inputs include a current limiter at ~ 5 mA to minimize losses. |
| Motor temperature input | 1 x dedicated, 5 V, 12-bit, single-ended analog input for motor temperature (1.65 kΩ pull-up to 5 V included).<br>NTC, PTC, RTD, Linear Voltage Sensors , Silicon Based Sensors and Switches are supported. |
| Protections | • Hardcoded / hardwired Drive protections:<br>  ◦ Automatic current derating on voltage, current and temperature<br>  ◦ Short-circuit Phase to DC bus<br>  ◦ Short-circuit Phase to Phase<br>  ◦ Short-circuit Phase to GND<br>• Configurable protections:<br>  ◦ DC bus over-voltage<br>  ◦ DC bus under-voltage<br>  ◦ Drive over-temperature<br>  ◦ Drive under-temperature<br>  ◦ Motor over-temperature (requires external sensor)<br>  ◦ Current overload ($I^2t$). Configurable up to Drive limits<br>  ◦ Voltage mode over-current (with a closed current loop, protection effectiveness depends on the PID).<br>• Motion Control protections:<br>  ◦ Halls sequence / combination error (Pending implementation)<br>  ◦ Limit switches<br>  ◦ Position following error<br>  ◦ Velocity / Position out of limits |

| Communications for Operation | |
|---|---|
| CANopen | CiA-301, CiA-303, CiA-305, CiA-306 and CiA-402 (4.0) compliant.<br>125 kbps to 1 Mbps (default).<br>Note: when configured as CANopen the Ethernet ports can still be used to configure the drive. |
| EtherCAT | CANopen over EtherCAT (CoE)<br>File over EtherCAT (FoE)<br>Ethernet over EtherCAT (EoE)<br>Note: CANopen is the communication enabled by default. In order to use EtherCAT, the FW must be updated. |

| Environmental conditions | |
|---|---|
| Aluminium case | Yes (interface board not covered) |
| Isolation between case and live circuits | > 200 MΩ. Measured between PE (case) and GND_P and +SUP.<br>Maximum voltage between PE (case) and live circuits: 440 V continuous, 800 V impulse according to IEC 61800-5-1.<br>Note: The drive includes 2 nF capacitance between the power supply negative (GND_P) and the enclosure (PE). |
| Case temperature | Operation:<br>• -40 °C to +60 °C at full current (Minimum power up temperature is -30 °C)<br>• +60 °C to +85 °C with derated current<br>For further information, see Thermal Specifications below.<br>Storage:<br>• -40 °C to +100 °C |
| Maximum humidity | 5% ~ 85% non-condensing |
| ESD and EMC immunity | ESD immunity IEC 61000-4-2: ± 30 kV contact discharge , ± 30 kV air discharge<br>EFT immunity IEC 61000-4-4: > 40 A<br>Surge immunity: IEC 61000-4-5 IPPM > 8 A |

## 7.2 Implementation aspects

The motion drive/ amplifier designed has been implemented as a stack of 4 PCBs. The detail of each PCB composing the motion drive/ amplifier is the following one (from lower PCB to upper PCB):

- Power Stage (GaN switching elements)
- Motion Control Stage (DSP)
- Communication/Programming Stage (microcontroller)
- Connector/Interface board (signal interface insulation)

All the PCBs have been successfully designed and manufactured. The first three ones are stacked inside a metallic housing to have a proper thermal dissipation and have a 'Faraday cage' shielded enclosure to have a good EMC/EMI performance.

The above mentioned PCB stack, once assembled, has the following appearance:

*Figure 49. Photo of the realized prototype of the motion drive*

The most relevant characteristic of the PCB stack is the miniaturization of size containing the high power ICs and all the driving capabilities described in the section 7.1 (Functionalities). To have an idea of the size achieved, the following images can be a good reference:

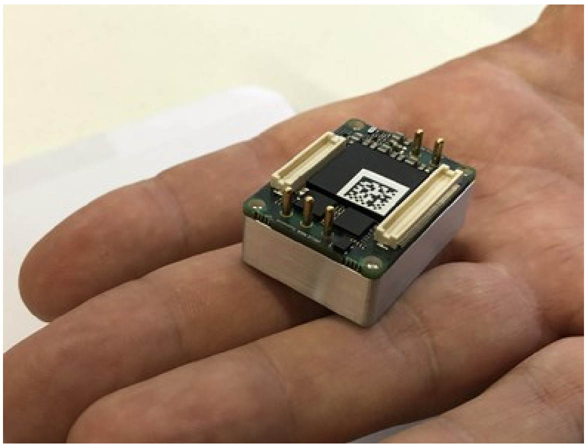


*Figure 50. Relative image of the realized prototype of the motion drive*

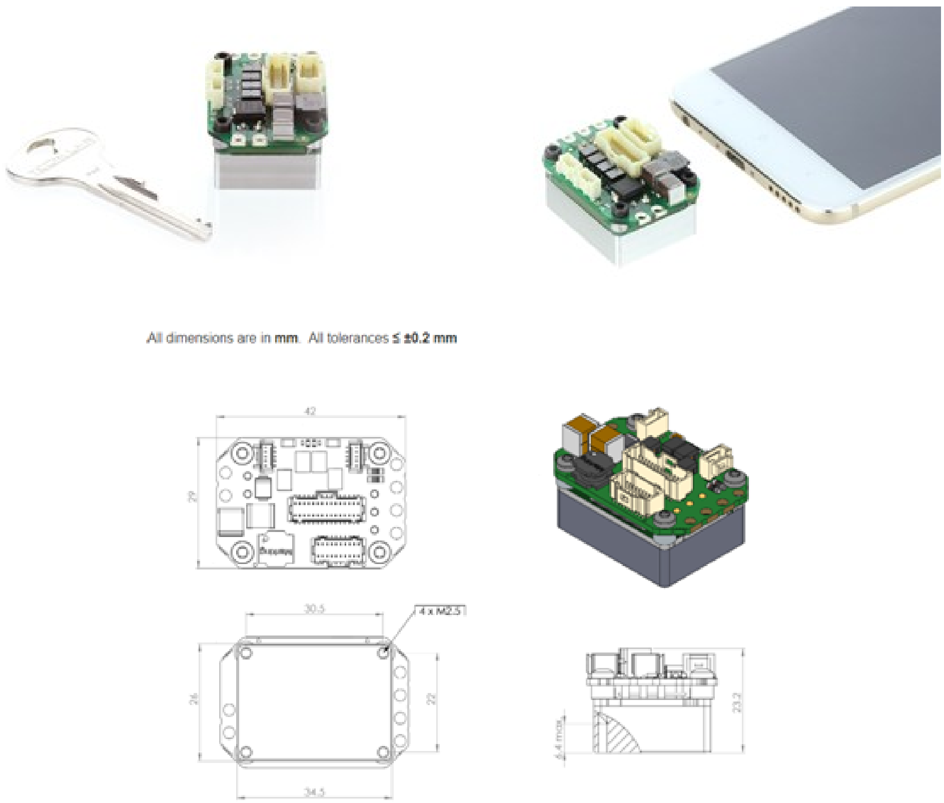



*Figure 51. Details of the realized prototype of the motion drive*

BB5 can be found in the following Pilots and use cases:

### 7.2.1 Pilot 2 “12 Inch wafer stage”

In this pilot, 6 current amplifiers have been integrated in the wafer stage machine. The movements of this machine have to be extremely precise, and also coordinated. In this pilot, the axis are coordinated in an EtherCAT network, and integrated in several different motors for moving the machine actuator.

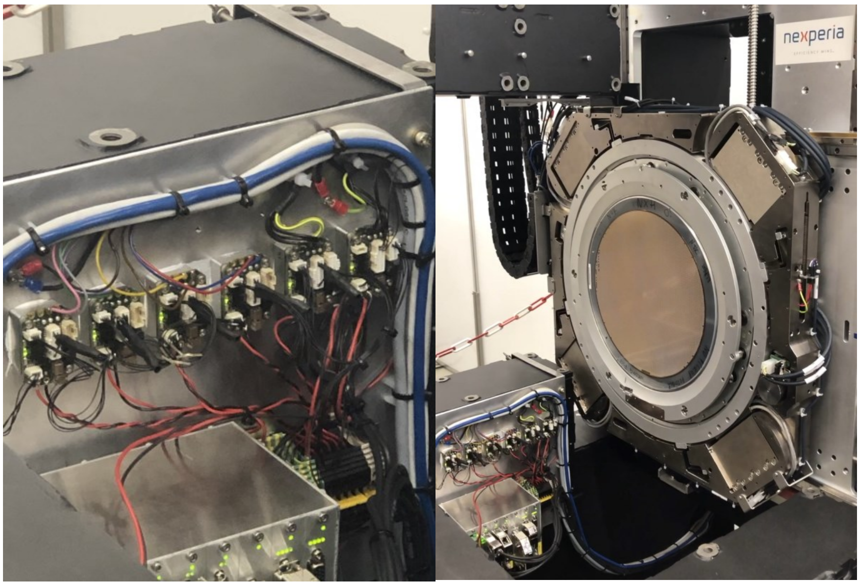



*Figure 52. Images of 6 BB5 units integrated in Pilot 2 (left) and generic picture of the machine and actuator (right)*

### 7.2.2 Use Case 2.2 “I-MECH platform validation on open modular robotic arm”

The current amplifier has been integrated in each of the 7 joints of the robotic arm. Traditionally, the robotic arms like this one  have an external cabinet with all the electronics installed, but this integration of the electronics inside the structure of the arm, has been possible due to the compactness of BB5.

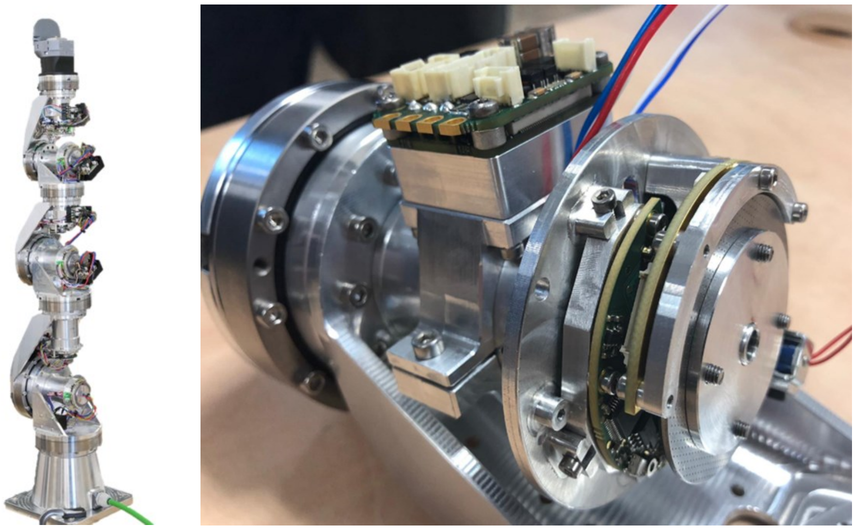


*Figure 53. Images of 7 BB5 units integrated in one arm (left) and detail of a robotic joint with BB5 attached (right)*

# 8 BB10 - Control specific multi-many core HW (EVI)

## 8.1 Functionalities

The two reference platforms selected for BB10 shows inner differences. First and foremost, they implement different Instruction Set Architectures (ISAs): the COTS platform is based on the long-term Intel x86-64 architecture, while the FPGA platform implements the ARMv8A instruction set. The two ISAs come from different vendors (Intel and ARM, respectively) and are not compliant. On the other hand, they both provide 64-bit instructions, including (different) support for hardware-assisted virtualization. Additionally, the second platform offers:

- An FPGA fabric, on which processing units with further ISAs (e.g. Microblaze) can be synthesized.
- A slow Cortex-R core meant for real-time processing.

From a performance point of view, the Intel i5 processor is known to overcome the ARM microcontrollers available on the second platform. On the other hand, the second platform offers the possibility of implementing some logic on the fast FPGA fabric, reaching performance not achievable through general-purpose processing units.

We want to point out that the differences between these two platforms represented a strong opportunity for improvement in the I-MECH project: adopting different approaches in BB11 for the real-time control allowed the project to further increase the investigation space and, in the end, evaluating a larger number of possibilities.

## 8.2 Implementation aspects

### 8.2.1 Pilot 1: Generic substrate carrier (GSC)

**FPGA platform**

The implementation targeted with BB11 is Pilot 1. The implementation is defined in an integration process where the BB functionalities are presented to the pilot owner and validated on the pilot. At the time of writing this report, the integration of the BB with Pilot 1 is ongoing. Through the integration meeting, the BB functionalities initially were presented to the pilot owner. The implementation of the BB on Pilot 1 was then decided to perform the steps proposed by the figure below:

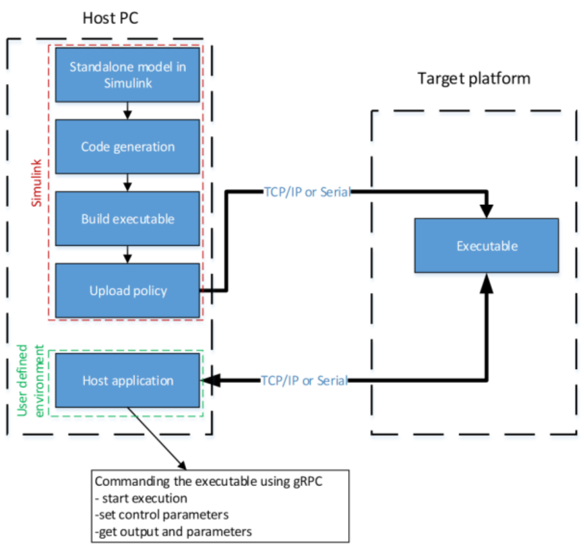



*Figure 54. The integration steps of BB11 with Pilot 1*

The integration in the Code generation, Build executable, upload policy, and executing on the platform are done. The remaining step is “host application” where the user can use the data tracing and parameter tuning via a host application else than Simulink. By achieving this step the integration of the BB with pilot 1 in layer 3 (gRPC framework) is validated.

**COTS platform**

The support of Pilot 1 with the COTS platform started by designing and developing the needed interface between Layer 2 and Layer 3 for monitoring and control. This also included the selection process of a suitable library for the communication. As explained in Deliverable D4.6, such an interface has been based on the gRPC technology by Google. gRPC is a new technology mixing the power of Google’s Protocol Buffers (for data serialization) with a Remote Procedure Call (RPC) interface. In particular, the Layer 2 exposes a gRPC interface, allowing the Layer 3 to perform common operations including model loading/starting/stopping, parameters setting and signals getting. The library on Layer 3 can be programmed through several languages, including C++, C#, Java and Python.

Then, we aimed at integrating the ERIKA RTOS with both Sioux’s EtherCAT proprietary stack and control code, as shown in the following picture:

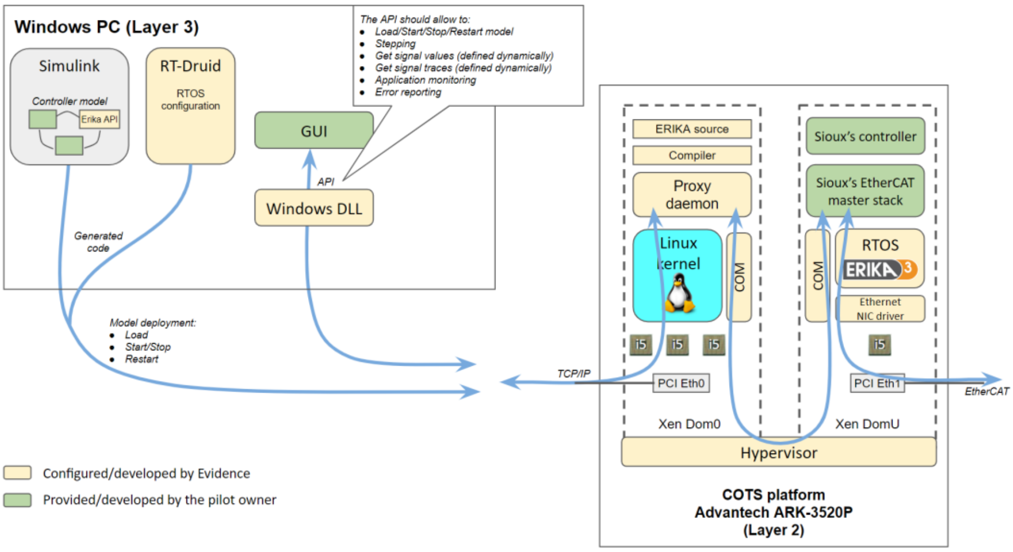



*Figure 55. BB10 COTS support for Pilot 1.*

This activity has been suspended by Sioux due to the uncertainty of the ERIKA licensing options after the end of the project. The man/months of partner EVI has been shifted towards the support of Pilot 3, where the porting of VxWorks on the hypervisor requested a much higher effort than originally forecasted (especially due to the lack of information by Windriver and B&R manufacturers).

## 8.2.2 Pilot 3 “In-line filling & stoppering machine, Tea bag machine”

Pilot 3 specified the requirement of using the VxWorks 6.9 RTOS for being able of supporting legacy code. Since VxWorks is not available for the FPGA platform, this pilot has been therefore only supported by the COTS platform.

The activity consisted in porting the VxWorks 6.9 BSP on top of the virtual machine exposed by the Xen hypervisor (see next picture). This has been a quite complex activity, due to the lack of support from the RTOS vendor. Once ported on the hypervisor, we have been able of running concurrent instances of the operating system for controlling different EtherCAT slaves. Moreover, we developed the mechanisms for letting the VxWorks RTOS reading/writing files on the filesystem of the virtualized device, as well as for a fast and real-time communication between the different instances of the RTOS.

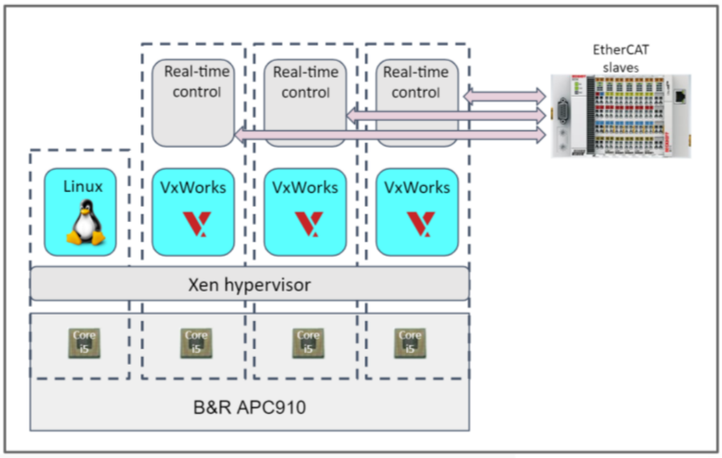



*Figure 56. BB10 COTS support for Pilot 3 - multiple VxWorks instances.*

# 9 Conclusions and next steps

The document provided the description of Instrumentation Layer building blocks (BB1 “Platform for Smart Sensors with Advanced Data Processing”, BB2 “Real-time wireless sensors providing complimentary feedback information”, BB4 “High speed vision”, BB5 “High performance servo amplifier” and BB10 “Control Specific Multi many core Platform” and their implementation aspects in four Pilots (Pilot 1 “Generic substrate carrier”, Pilot 2 ”12 Inch wafer stage”, Pilot 3 “High speed packaging”, Pilot 4 ”Big CNC machining”), two Demonstrators (Demo 1 “Manufacture of an Insulin Delivery System”, Demo 2 “Injection mold industry”) and three Use-Cases (UC1.1 “Power electronic for hoist and crane sector”, UC1.3 ”PAC based modular HW for machinery”, UC2.2 “I-MECH platform validation on open modular robotic arm”). In addition, the document provided description how to operate those building blocks. All the developments will be used in different combinations with other BBs from WP4 “Control Layer design and development” and WP5 “System Behavior Layer design and interfaces” and will be integrated, tested and validated in WP6 “ Implementation and integration of I-MECH platform” and WP7 “Pilots and demonstrators”, while contributing to horizontal WP8 “Dissemination, exploitation, communication activities”. Testing and validation reports from WP3 BBs will be available at the end of the project in WP6 and WP7 deliverables.

**Acknowledgement**
*This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737453. This Joint Undertaking receives support from the European Union’s Horizon 2020 research and innovation program and Netherlands, Czech Republic, Latvia, Spain, Greece, Portugal, Belgium, Italy, France, Ireland*