ITEA 3

OPTIMUM

OPTimised Industrial IoT and Distributed Control Platform
for Manufacturing and Material Handling

Deliverable 6.3
Description of the Material Handling Demonstrators

Deliverable type:	Document
Deliverable reference number:	ITEA 16043 | D6.3
Related Work Package:	WP 6
Due date:	2021-04-30
Actual submission date:	2021-05-18
Responsible organisation:	DEMAG
Editor:	Giuliano Persico
Dissemination level:	Public
Revision:	Version 4.0

Abstract:	This document provides an overview on the current state of the final real demonstrators. These demonstrators are specified and realized, suitable to demonstrate and evaluate the most relevant use cases from the material handling domain.

Keywords:	OPTIMUM, demonstrator, evaluation, material handling, specifications, assembly scenario.

Table_head	Name 1 (partner)	Name 2 (partner)	Approval date (1 / 2)
Approval at WP level	Metin Tekkalmaz (ERSTE)	Klaus Hanisch (TARAKOS)	26.04.2021
Veto Review	No Veto		18.05.2021

Editor

Giuliano Persico (DEMAG)

Contributors

Marius Freitag (DEMAG)
Fabian Hölzke (URO)
Hannes Raddatz (URO)
Matthias Riedl (IFAK)
Duy Lam Tran (IFAK)
Karsten Meisberger (NXP)
Hagen Borstell (THORSIS)
Henry Cermann (TARAKOS)
Klaus Hanisch (TARAKOS)
Metin Tekkalmaz (ERSTE)

Executive Summary

This material handling demonstrator specification and realization report provides a description of the MH-Demonstrators for the OPTIMUM project at DEMAG in the Research Factory in Germany in Chapter 1 and at ETRI site in South Korea in Chapter2.

The demonstrator is described within this document in the following format:

- Overview of the demonstrator
- Design of the demonstrator
- System architecture description
- Visualization and Engineering Tools
- Process flow in the demonstrator line
- Testing and verification.

Table of Content

Figures

Tables

1 DEMAG Material Handling Demonstrator

1.1 Overview

The MH-Demonstrator built in the DEMAG Research factory in Wetter includes typical different machines that are utilized for the indoor material flow in industrial processes.

In a factory hall with an area of approx. 280 m² and a height of 10 m, two industrial cranes with each two trolleys and hoists have been installed and realize the “floor free” material transport. On the floor, there is a manually driven forklift and an AGV-prototype that are in charge for the “floor bounded” material transportation.

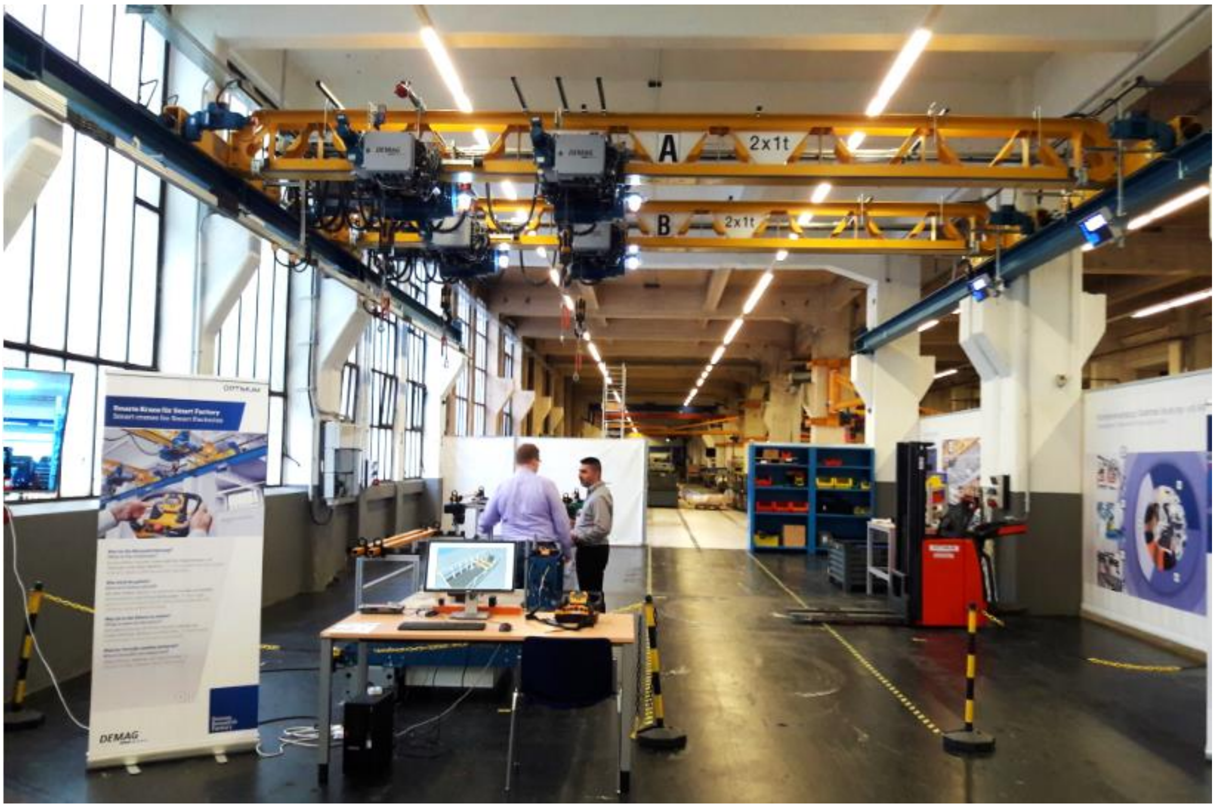


Figure 1: MH-Demonstrator in the DEMAG Research Factory

To demonstrate the innovative assist functions implemented during the OPTIMUM project, we have defined a “Demo-Assembly”, so that all assist functions can be utilized and validated in a logical context. For the assembly scenario, in total four assembly places have been installed in the hall. The assembly activities are worked out by two human operators that are also in charge to control the machines needed for the material flow: cranes, AGV and forklift.

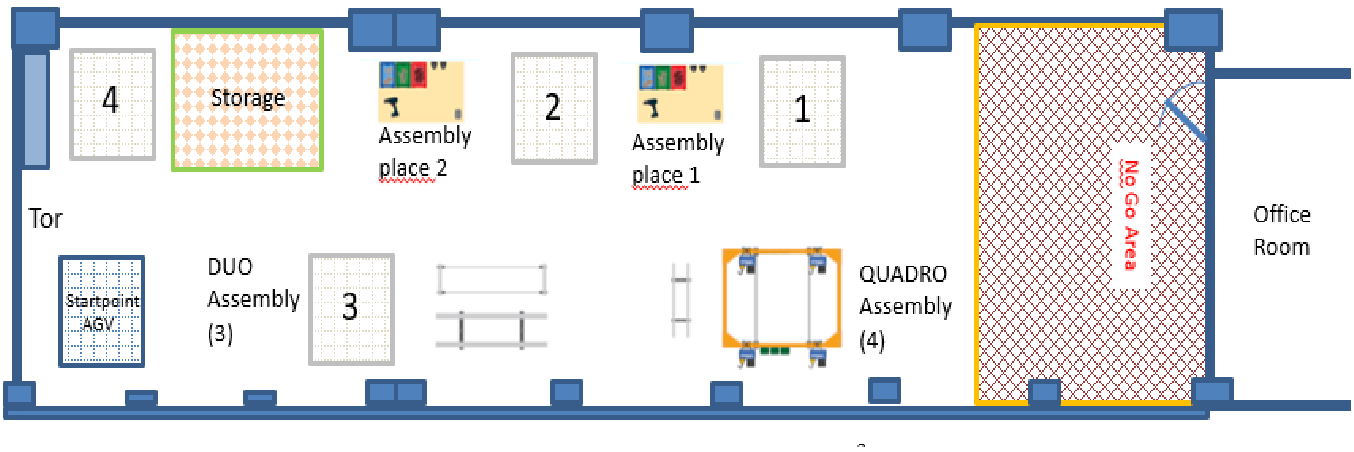


Figure 2: Overview assembly scenario

The assembly process consists in three different steps – see also Fig. 3-6.

First, a Demag chain hoist is completed with the control board, the brake and the electrical cover. This happens at the assembly places 1 and 2.

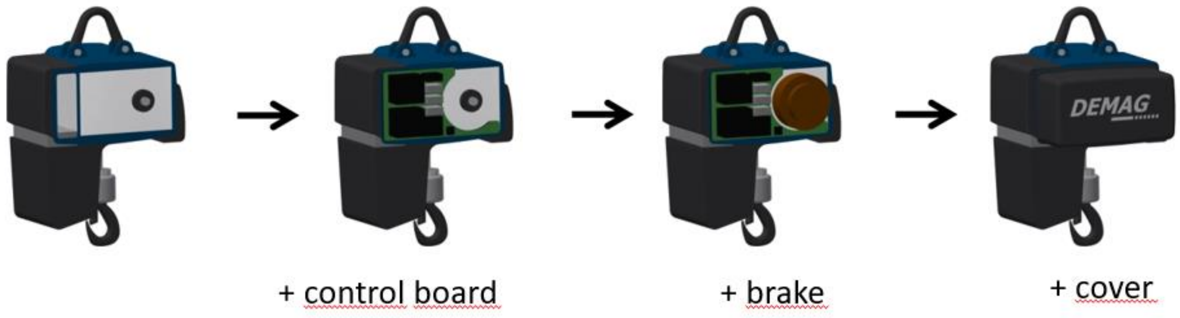


Figure 3: First assembly step

In the second step, two chain hoists are connected to a profile bar to form a "Duo". For the Duo-assembling there is a dedicated "Duo-assembling-station".

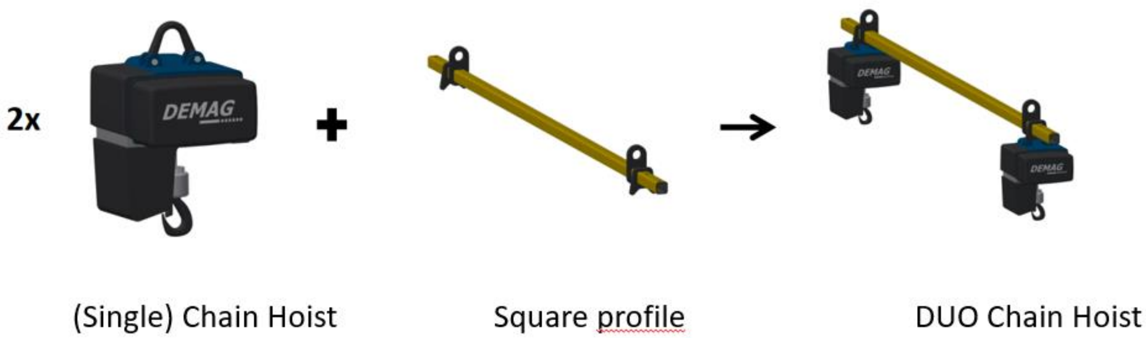


Figure 4: Second assembly step

At the third workstation, two duo chain hoists are assembled with two cross beams to form a “Quadro”. Also, for the Quadro-Assembling, there is a dedicated assembly station available.

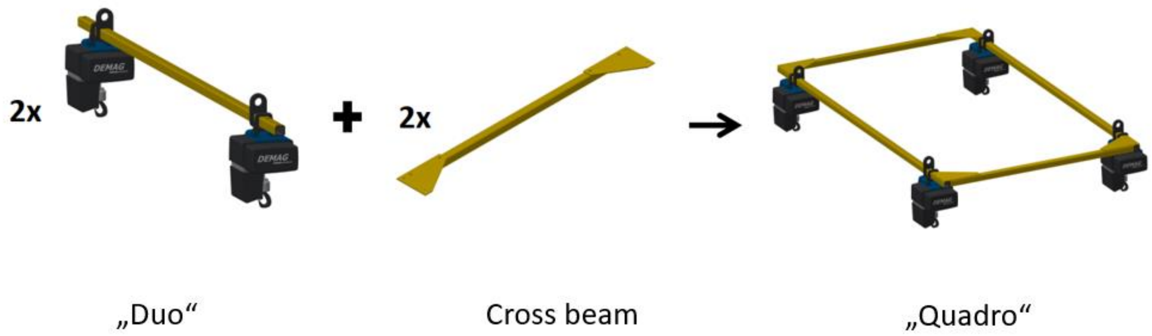


Figure 5: Third assembly step

Figure 6: Assembly and mounting stations

The chain hoist represents in this scenario a “single load” that can be safely transported with just one attachment point. The “Duo” represents a “long load”: These long loads need two attachment points to be safely transported by crane. The “Quadro” represent “flat loads” (i.e. aircraft wings) that need more than two attachment points for a safe transportation.

For this the two cranes can be utilized in “Tandem-Mode” in different configuration - depending on the load type, so that a safe transportation of the different load shapes is always possible.

The following pictures show as example the different situations in the MH-demonstrator:

A) Single attachment to a compact load
B) Double attachment to a long load
C) Quadruple attachment to a flat load

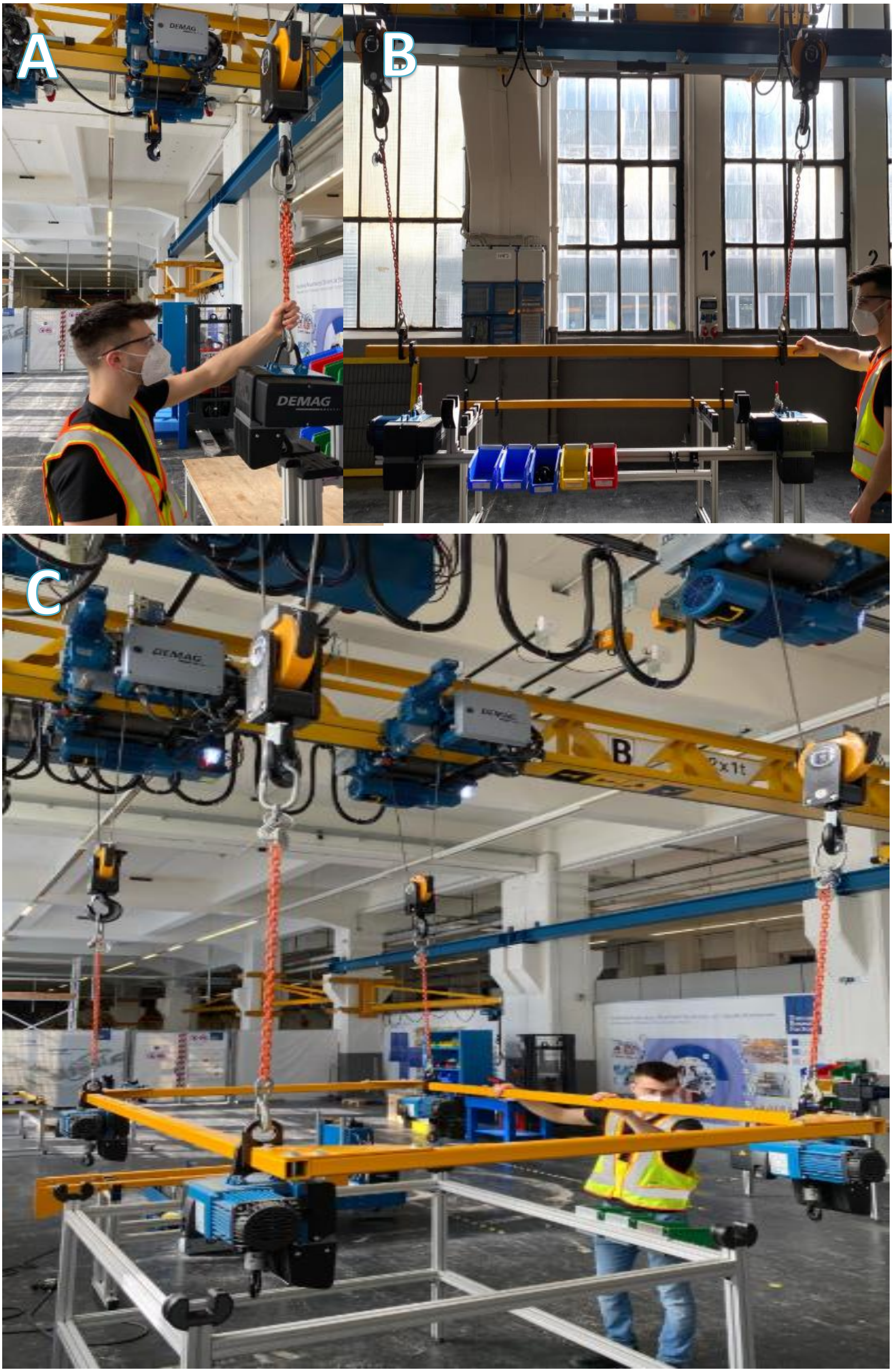


Figure 7: Handling of different load types

The material movements between the different stations are handled by the two overhead cranes and on the floor – depending on the scenario - by forklift and/or AGV.

Main target of the MH-Demonstrator is not just to demonstrate the innovative assist functions, but also to validate that the utilization of such assist function is improving the efficiency of the process and shortening the total assembly time.

For this purpose, the defined assembly process will be worked out utilizing:

a) Conventional MH-machines (as reference value)
b) OPTIMUM improved MH-machines, where the assist functions will support the operators targeting a higher efficiency and safety.

The assembly time done by humans in each assembly station will be the same for both scenarios a) and b). The time differences are created by the utilization of the assist functions of the MH-machines and the higher efficiency of the material flow.

The assembly process in the demo line is described more in detail in section 1.6 Process flow in the demonstrator line.

1.2 Demonstrator design / Pilot plant design

The MH-Demonstrator includes typical different machines that are participating to the material flow in an industrial process. Main targets are the realization of the connectivity between the different machines and the indoor localization of machines and operators.

Based on that, innovative assist functions will be implemented. The assist functions will support the operators targeting a higher efficiency and safety.

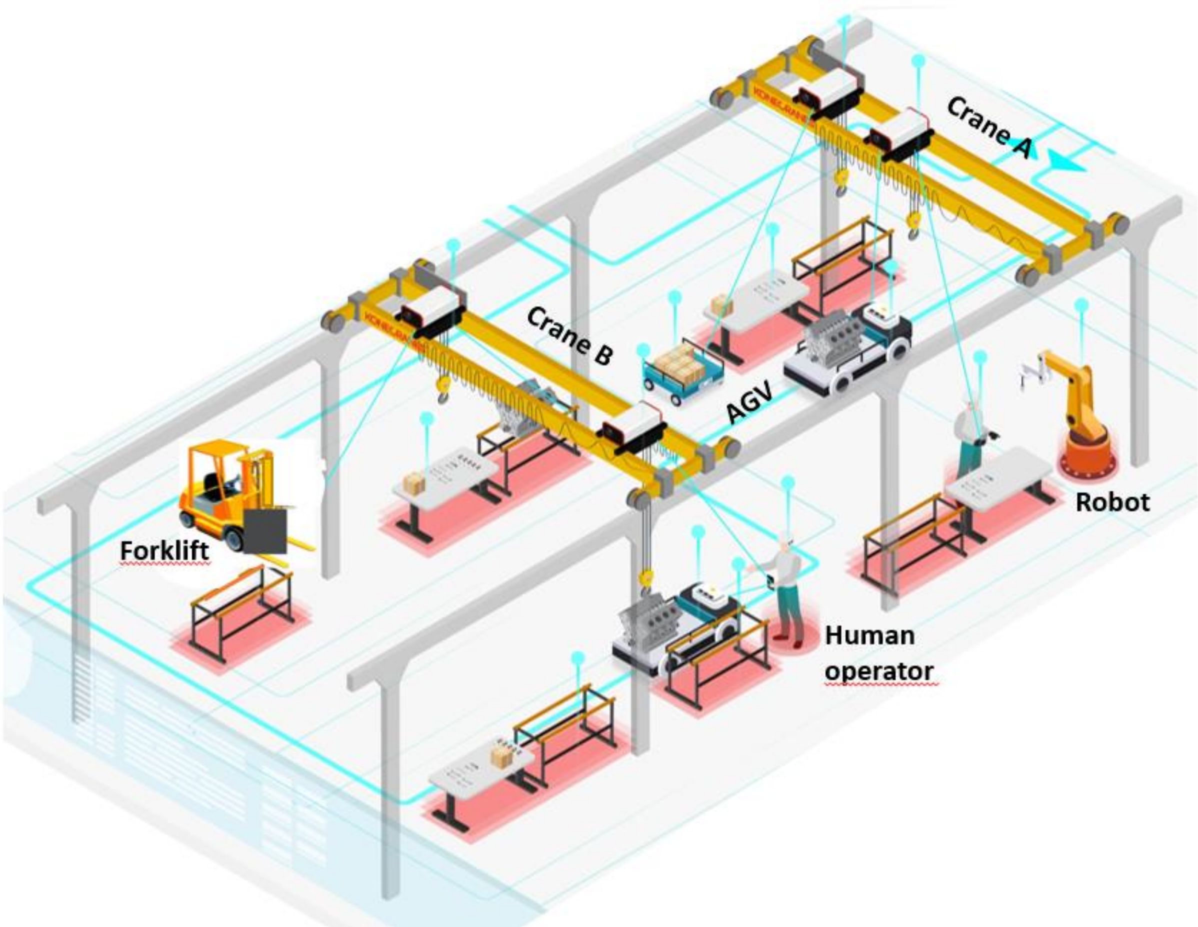


Figure 8: Principle of MH-Demonstrator

In the DEMAG Research Factory there are “real” industrial cranes and other machines for material handling available. The Forklift is a rented machine made by LINDE - Type L16, the Cranes and the AGV are made by DEMAG.

The main technical data of the forklift are listed below.

Table 1: Technical data of forkliftModel	Load capacity	load Stroke	Driving speed	Turning radius	Battery voltage	nominal capacity
L16R	1,6 (t)	2844 (mm)	6 km/h	1681 (mm)	24 (V=)	250 (Ah)

Further info about the forklift can be seen under [1]: Plattform-Hochhubwagen L14 – L20 AP von Linde Material Handling (linde-mh.de)

The AGV is a prototype build by DEMAG that has been integrated in the Research Factory to be able to demonstrate machine to machine communication and collaboration between floor-free and floor-bounded MH-machines.

The main technical data of the AGV-prototype are listed below.

Table 2: Technical data of AGV

AGV Model	nominal load	drive-through dimension	Driving speed (max)	nominal voltage	nominal capacity
DEMAG	1,0 (t)	1,9 (m)	60 (m/min)	24 (V=)	165 (Ah)

The AGV control is from BLUEBOTICS, Type ANT®

[1] https://bluebotics.com/autonomous-navigation-technology

This control allows a natural feature navigation (sometimes called ‘natural navigation’, ‘free navigation’ or ‘SLAM navigation’) uses laser safety scanners (from SICK) to first identify and then match permanent features in the environment, such as walls, pillars, and machines. The paths followed are virtual, no needs to install marks or sensors in/on the factory floor for path definition.

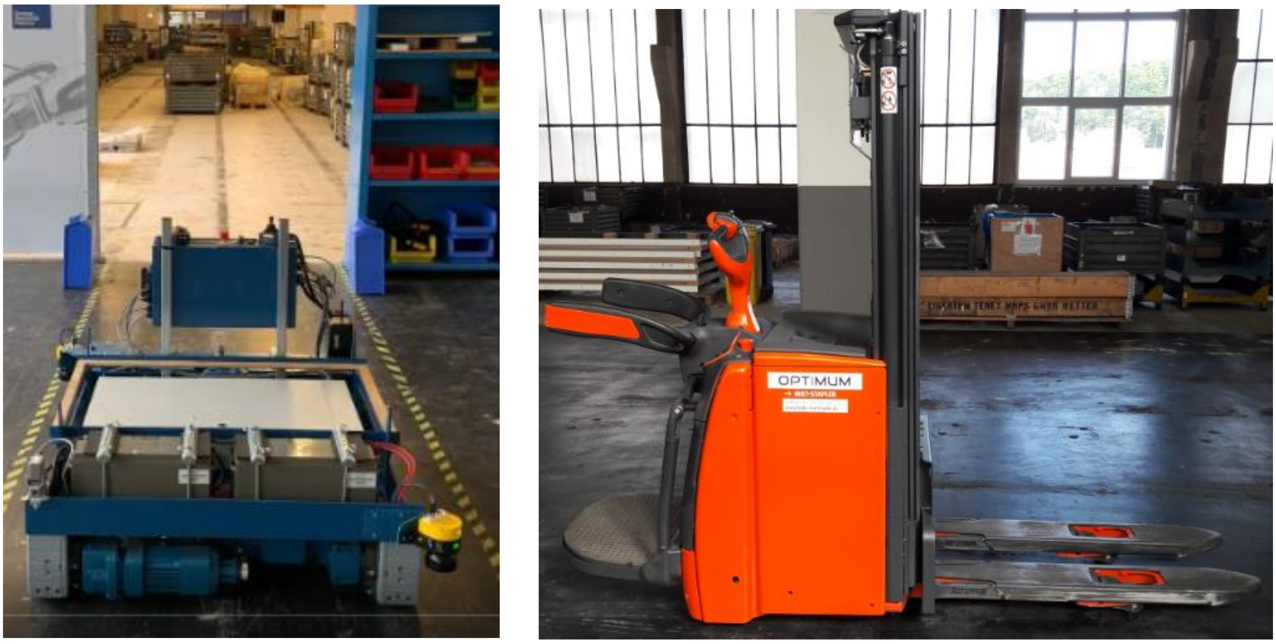


Figure 9: AGV Prototype and LINDE Forklift in the DEMAG Research Factory

The two overhead cranes are made by DEMAG – Type EVKE DMR3 [2].

Each Crane has two trolleys with a rope hoist. The crane control is a standard control Type "DEMAG Safe Control" [3].

The main technical data of the cranes are listed below.

Table 3: Technical data of cranes

Crane Model	Crane nominal load	Nominal load per hoist	Driving speed trolley/crane	Hoisting speed	Bridge length	nominal voltage
EVKE DMR	2,0 (t)	1,0 (t)	40/60 (m/min)	1,3 / 8,0 (m/min)	7,83 (m)	3 x 400 (V~)

Each crane has two trolleys and each trolley one hoist, like shown in Fig. 10.

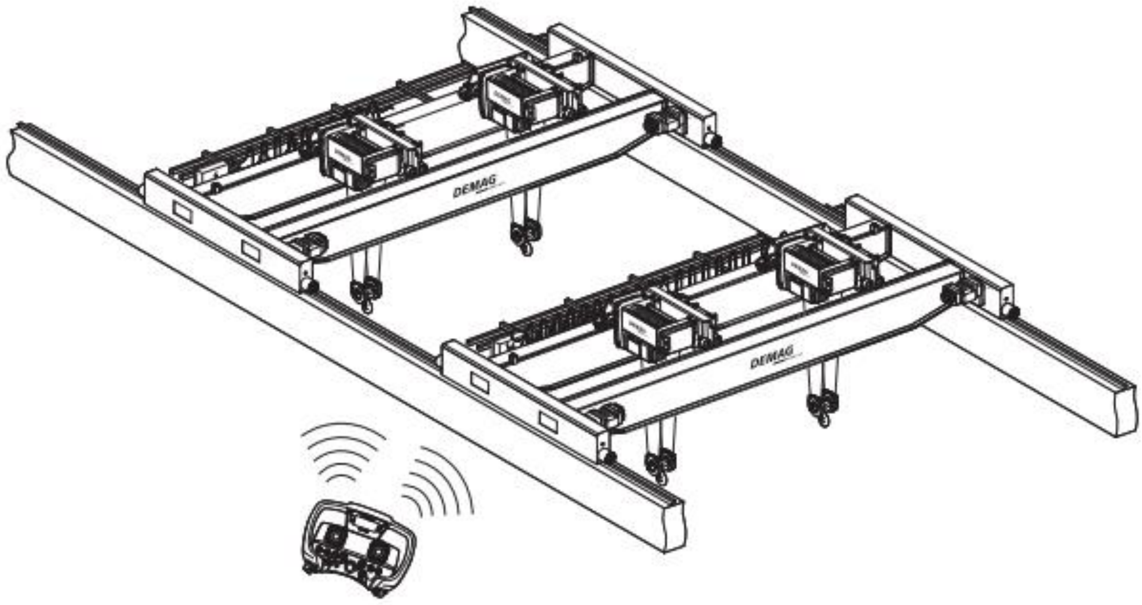
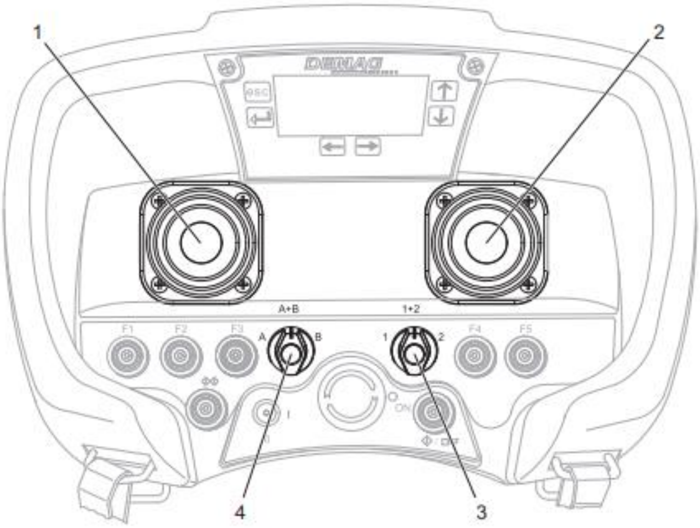


Figure 10: Demag Smart-Tandem 2 – 2 tandem control for cranes

The operator can control the crane movements wirelessly with the DRC-JS radio control [4]. Also, the selection of the operational mode is done by dedicated switches – Figure 11



1	Travel switch	2	Hoist switch	3	Crab selector switch	4	Crane selector switch (only for group control mode)
-	Forward/reverse long travel	-	Lifting/lowering	(1)	Crab 1	(A)	Crane A
-	Left/right cross travel			(1+2)	Both crabs in parallel mode	(A+B)	Both cranes in parallel mode
				(2)	Crab 2	(B)	Crane B

Figure 11: Demag DRC-JT(S) radio control

The trolley selector switch (3) enables the trolleys to be controlled separately (trolley 1 or 2) or together (1+2). The crane selector switch (4) enables the cranes to be controlled separately (crane A or B) or together (cranes A+B).

The radio control system consists of 2 DRC-JT(S) joystick transmitters and 2 DRC-DR radio receivers with a CAN-bus connection for 2 cranes and 4 trolleys.

Tandem installations feature separate control and group control modes. The control mode refers to the entire installation; therefore, it always applies to both cranes and results from the way in which the cranes and their controls are assigned to each other:

- **Separate control (Single mode) - each crane is controlled by its own control system**
  In this operating mode, each crane is controlled by its own radio control device.
  If there are no safety-relevant special functions which might result in mutual interference of the cranes (e.g. load-dependent anti-collision control) and
  the parameters of all controls of the Smart-Tandem 2 – 2 system are correspondingly programmed, both cranes can be used independently of each other in this separate control mode.
- **Group control (Tandem mode) - both cranes are controlled by the same control system**
  Group control mode enables lifting and transport operations to be performed that are partly not in a safe manner possible with individual cranes:
  ● Lifting long and bulky loads,
  ● Lifting loads that weigh more than the maximum load capacity of an individual crane.
  One of the two radio control units must be logged off to switch from separate control to group control mode. The remaining radio can control both cranes with all trolleys.

Both cranes can be controlled (also independently of each other) in group control mode. The cranes and trolleys to be controlled are determined by the active control system:

● Both cranes are controlled in parallel in tandem mode (crane selector switch: A+B).

● Only one crane is controlled in single mode (crane selector switch: A or B).

When the cranes are operated as a group, it must be ensured that the distance between the cranes does not exceed the range of the radio link of approx. 50 m.

Figure 12: DEMAG EVKE cranes and DRC-J radio control station in the DEMAG Research Factory

1.3 HW architecture

Figure 13 shows the block diagram of the distributed control for a crane:

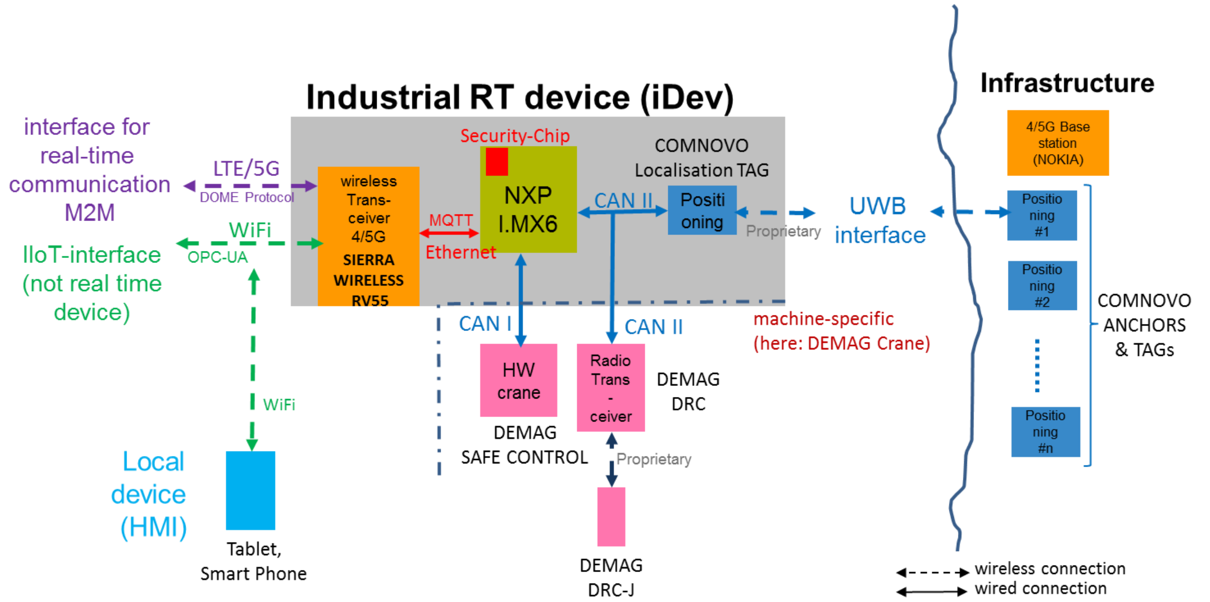


Figure 13: Block diagram of the crane distributed control

The Industrial Device (iDev) in the grey box represents the distribute control platform;
All actors (machines & operators) involved in the material flow are equipped with iDevs.
The iDevs communicates with each other utilizing the OPTIMUM-infrastructure.
This infrastructure consists in 5G wireless communication and UWB-localization networks.
The following chapters describe more in detail the iDev components.

1.3.1 Embedded Control Board

For the demonstrator, we have implemented the powerful and cost effective i.MX6 evaluation board from partner NXP.

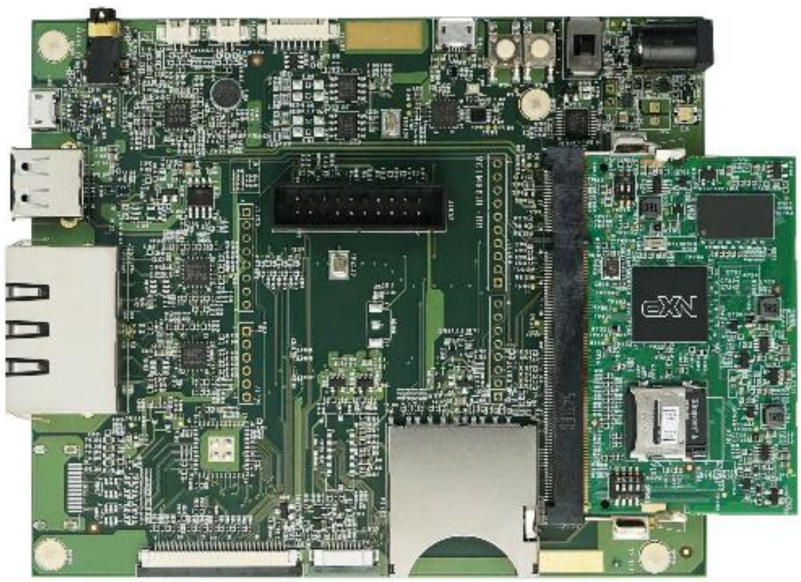


Figure 14: i.MX6 Evaluation Board

The i.MX 6ULL processors represent NXP's latest achievement in integrated multimedia-focused products offering high-performance processing with a high degree of functional integration, targeted towards the growing market of connected devices.

The i.MX 6ULL is a high performance, ultra-efficient processor family featuring NXP's advanced implementation of the single Arm Cortex®-A7 core, which operates at speeds of up to 900 MHz. i.MX 6ULL includes an integrated power management module that reduces the complexity of the external power supply and simplifies the power sequencing.

Each processor in this family provides various memory interfaces, including LPDDR2, DDR3, DDR3L, Raw and Managed NAND flash, NOR flash, eMMC, Quad SPI, and a wide range of other interfaces for connecting peripherals, such as CAN, WLAN, Bluetooth™, GPS, displays, and camera sensors.

Figure 15 shows the main features, Figure 16 the block diagram.

Technical and Functional Specifications

Processor	NXP i.MX6ULL 900 MHz Arm Cortex-A7 core, MCIMX6Y2DVM09AB
Power management	Discretes
Memory	▪ 4Gb DDR3L SDRAM, 400 MHz ▪ 256 MB Quad SPI Flash ▪ MicroSD connector ▪ Footprint for eMMC ▪ Footprint for NAND Flash
Size	2.66 inch x 1.27 inch (6.76 cm x 4.24 cm), 4-layer board
Display board interface	▪ LCD expansion port connector ▪ HDMI connector and footprint for HDMI transmitter
Audio	▪ Audio codec ▪ 3.5 mm Stereo Headphone output ▪ Mono-Microphone input on board ▪ Left and Right Speaker Out connectors
Connectivity	▪ One USB 2.0 Micro-B OTG connectors ▪ One USB 2.0 Standard-A host connectors ▪ Two Ethernet (10/100T) connectors ▪ Dual CAN connector ▪ SD/SDIO connector
Camera	Parallel camera connector
Sensors	▪ NXP® MAG3110 eCOMPASS ▪ NXP FXLS8471Q accelerometer ▪ Footprint for gyroscope
Debug	▪ 20-pin standard JTAG connector ▪ UART to Micro USB connector
Expansion port	Arduino header
Size	5.12 inch x 4.25 inch (13.0 cm x 10.8 cm), 4-layer board

Figure 15: i.MX6 technical Specification

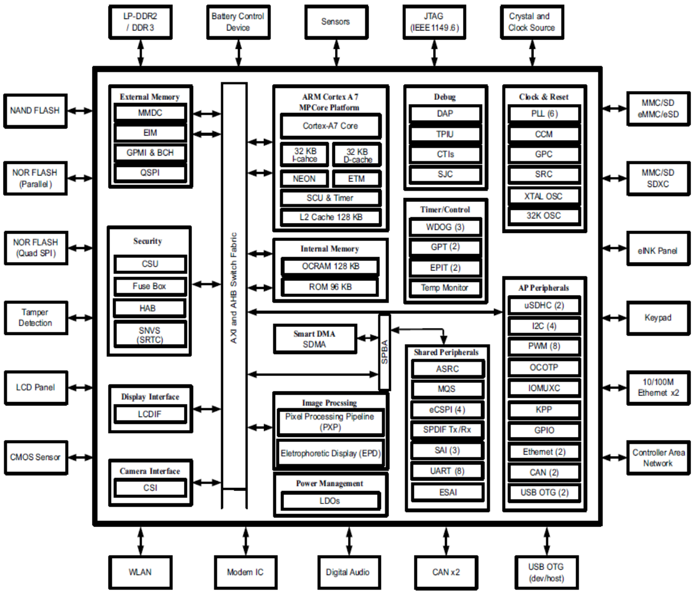


Figure 16: Block diagram shows the functional modules in the i.MX 6 processor system

For more information about NXP eval board products see [5] [i.MX Evaluation and Development Boards | NXP Semiconductors]

1.3.2 Indoor real time localization system (RTLS)

The indoor localization system is based on the UWB-technology and consists in stationary anchors (A) and movable Tags (T).

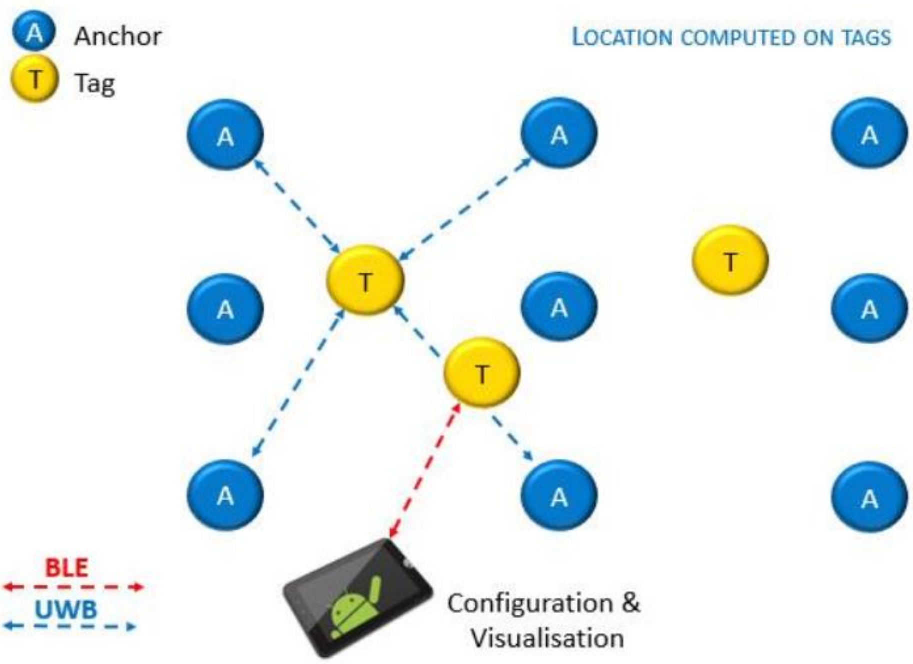


Figure 17: Example for RTLS

The device-HW are provided by the OPTIMUM partner company COMNOVO.
The SW running in all localization devices is made by COMNOVO.

In the OPTIMUM factory hall, there are eight anchors fix installed accordingly, to get a good connection with the moving tags in all possible directions and positions of material flow – see Figure 18.

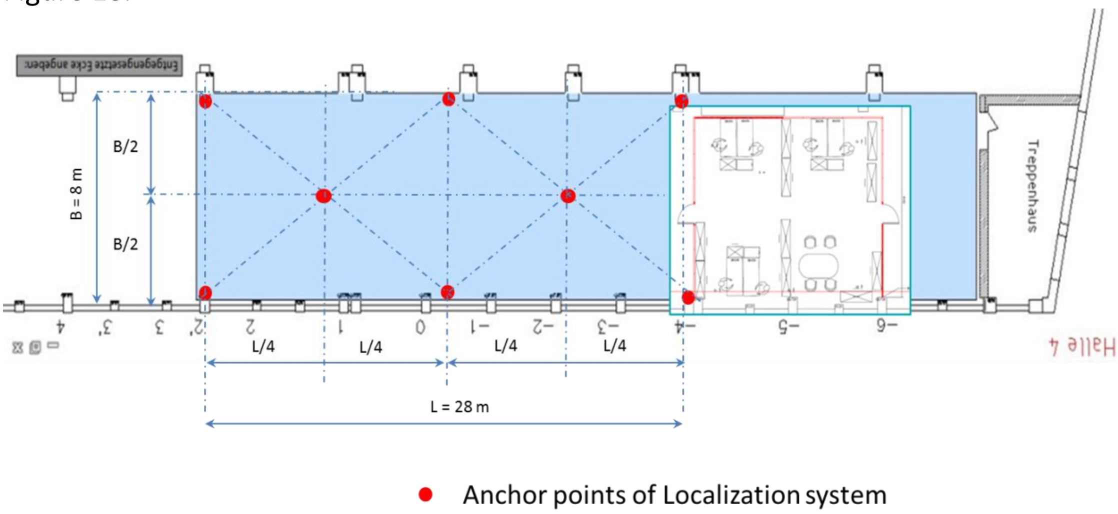


Figure 18: Position of the eight anchors in the hall

The tags calculate their 3D-position utilizing TOF (**t**ime **o**n **f**ly) -Algorithms.
The X- Y- and Z-positions are transmitted with a rate of 100 ms at 125 kBaud via CAN-Bus to the iDev.

Figure 19: COMNOVO and DECAWAVE Tags

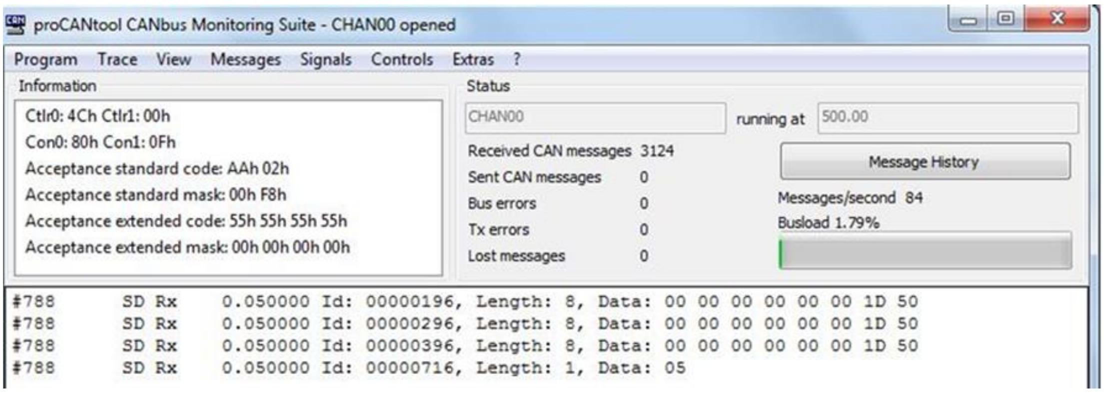


Figure 20: Screen shot on CAN data positions transmitted by the tag

1.3.3 5G Network Infrastructure (Base Station)

The components for 5G infrastructure in the MH-demonstrator hall are delivered by NOKIA: It is a 5G SA system based on Nokia Digital Automation Cloud Nx gNBs, On-Premise Core Initially based on 3GPP rel. 15.
Since April 2020 we are utilizing the dedicated frequency band for private industrial networks (B43: 3,70 – 3,80 GHz).

- Operation mode: TDD
- Broadcast method: W7D
- Spectrum 3,7 – 3,8 GHz (Bandwidth 100MHz)
- Radio transmission power: 500 mW
- frequency band allocation by BnetzA – Köln - Allocation-no: 14557414

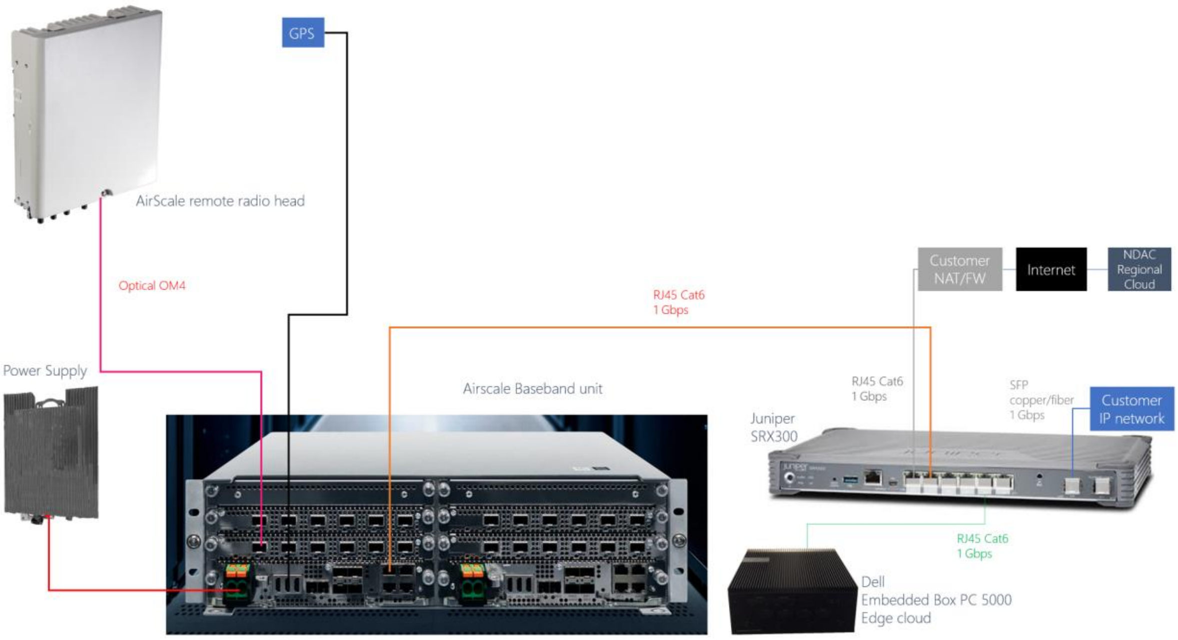


Figure 21: 5G installation in the research factory

1.3.3.1 Wireless Transceiver

To connect the different machines to the wireless network we evaluated different devices from technical and commercial points of view. The selected solution adopts the router AirLink® RV55 from the company SIERRA WIRELESS.

Figure 22: AirLink RV55 – SIERRA WIRELESS

The AirLink® RV55 is a rugged, compact, LTE and 5G-Advanced Pro router. The RV55 router is designed for connecting critical remote assets, infrastructure, and vehicle fleets. The RV55 provides real-time remote connectivity for SCADA, distribution management systems and metering. The RV55 is suitable for several frequency bands and fits also to B43 for the OPTIMUM MH-Demonstrator. The RV55 also supports new unlicensed bands (e.g., CBRS - Citizens Broadband Radio Service and the coming 5G technology) to support dedicated private services.

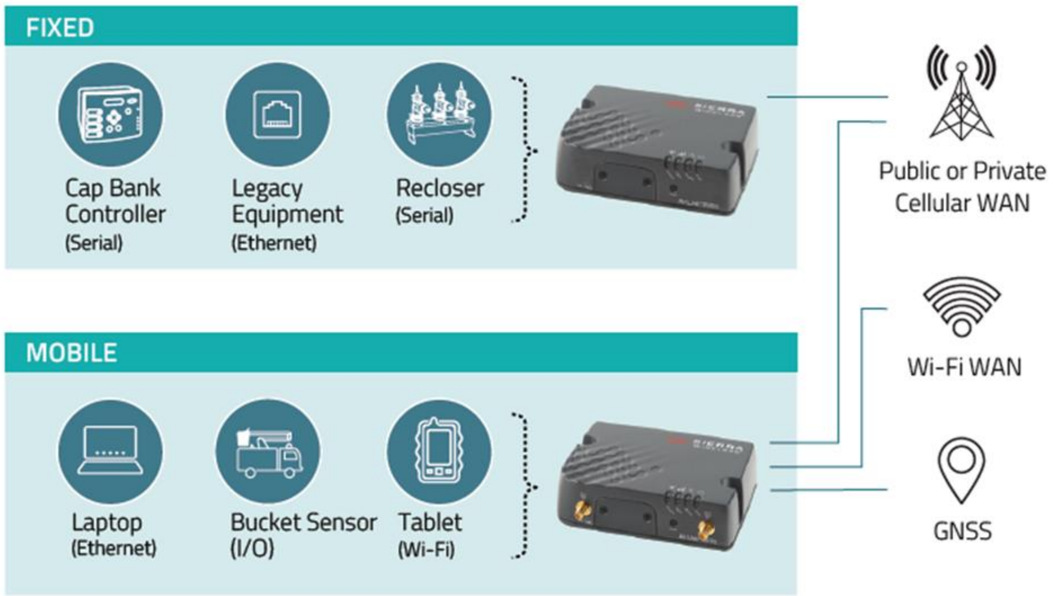


Figure 23: Typical applications of the router

For more details see [6] [Sierra wireless AirLink RV55 Manuals | ManualsLib]

1.3.4 Enhanced HMI

To initiate and control the new machine's functions (assist functions) there is the need of an enhanced HMI for the operators. Typical SotA devices are tablets and smart phones.

In the market, there is a huge variant of such devices with different sizes, features and operating systems.

At DEMAG company we have some experience with components for industrial applications. These devices are ruggedized for typical use cases in MH-Domain.

Figure 24: Enhanced HMI Device – PANASONIC Tablet FZ-M1D107YT3

Key Features

- Processor: Intel® Core™ i5-7Y57 vPro
- RAM: 4 GB
- Op. System: Windows 10 Pro
- Display-Size: 17,8 cm, 7" Active matrix (TFT)
- Color-LC-Display 1280 x 800 (WUXGA) with glare-free capacitive touch screen, operable with gloves (up to 700 cd/m²)
- Protection degree: IP65 (MIL-STD 810G), shock resistant, water and dust resistant
- Interfaces: Flexible configuration connection (LAN, serial interface, 2D barcode scanner, USB 2.0) and integrated extension for business use (NFC, UHF RFID, SmartCard reader)

It is possible to also install a Linux operating system on this device.

Figure 25 shows the implemented Graphical user interface. A detailed description of the GUI can be found in Deliverable [7] GUI Specification for Material Handling Use Case, OPTIMUM_PPR6_Attach_5_GUI_v1.9.docx

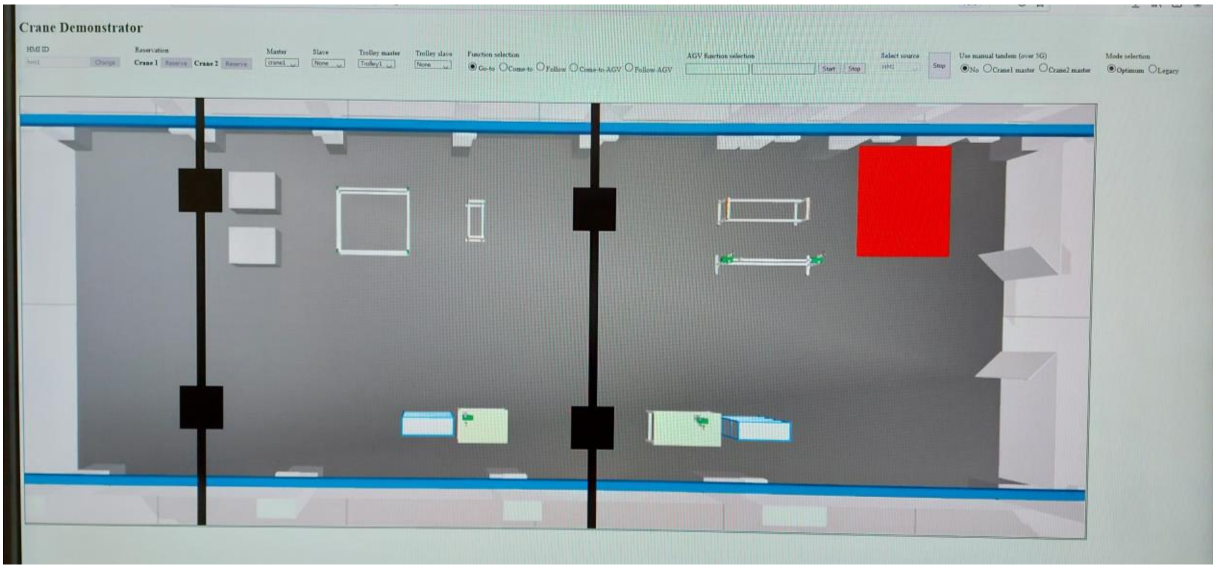


Figure 25: HMI Graphical user interface (2D-Version)

1.3.5 Machine specific components

Depending on the Use-Case owner, different – machine specific - controls are implemented. This controls realize the basic functions and the machine-safety.

1.3.5.1 Crane Control

For the MH-Demonstrator the DEMAG-cranes utilize the SAFECONTROL® control unit.

This is a standardized control solution for DEMAG-cranes that utilize CAN-Bus Interfaces for the internal communication. To control the crane the operator utilizes so called wired control pendants or wireless radio devices of the DEMAG DRC-Family.

The SAFECONTROL® is described in the document [3] “Operating instructions DMR 3 – 16 SafeControl” – Id. Nr. 211 809 44.

The DRC-J® is described in the document [4] “Operating manual/assembly instructions DRC-J/-JS D3 joystick transmitters” – Id. Nr. 211 271 44.

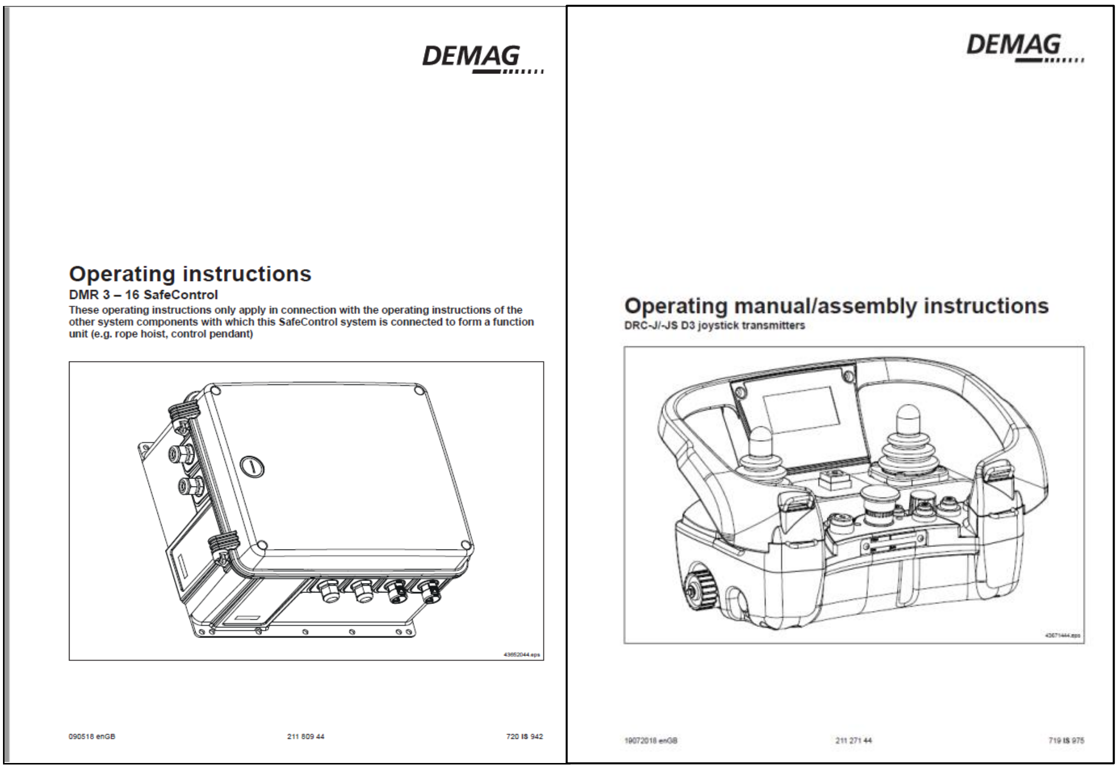


Figure 26: Operating instruction manuals

For wireless command transmission, there are - depending on the application - various operating devices whose equipment varies in terms of the number and type of operating elements. However, the proprietary radio technology used in the DRC D3 is identical:

- DRC D3 frequency band: 2.4 GHz ISM band (2405 - 2480 MHz)
- Radio output power (ERP): Max. 20 dBm (100 mW)
- Typical range: approx. 100 m

In the MH-Demonstrator the provided radio device is Type DRC-J D3®.

DRC-D3 transmitters and receivers comply with the DIN EN 300328 V 1.8.1 standard, which has been binding since 2015 and regulates coexistence with other radio applications in the 2.4 GHz ISM frequency range (e.g. WLAN access points).

The radio transmission method combines different transmission mechanisms: a frequency hopping system ensures that the radio transmission is robust and highly resistant to interference - an adaptive characteristic (Listen Before Talk), combined with this allows interference with neighboring radio applications

In the DEMAG Research Factory, two cranes with two trolleys/hoists each can be used as a single machine or as a combination (tandem operation). For such tandem applications, operator interfaces of type DRC-J D3 are typically used - see Figure 27.

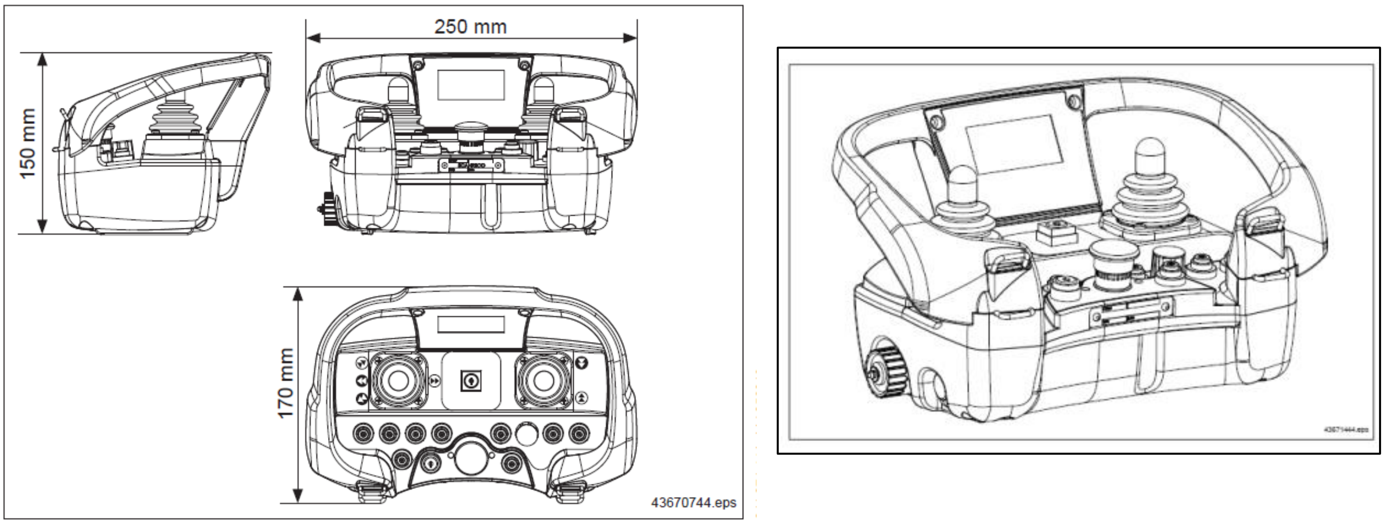


Figure 27: Radio device Type DRC-J D3®

The DRC-J (J = joystick) transmitter typically includes

- A stop button (All DRC D3 radio control systems meet Category 3 and Performance Level (PL) "d" requirements of the relevant crane standards for the stop function).
- A single axis joystick for lifting/lowering
- A two-axle joystick for crane and trolley travel (long and cross travel)
- A key switch for switching the transmitter on/off
- One button for horn/start
- Up to seven additional buttons for further crane functions (F0 - F6)
- Two 3-step selector switches (1, 1+2, 2) and (A, A+B, B) for trolley or crane selection
- An LCD display with keys for menu navigation

1.3.5.2 AGV Control

For the AGV-Prototype a BLUEBOTICS Control Type “ANT lite +” has been implemented.

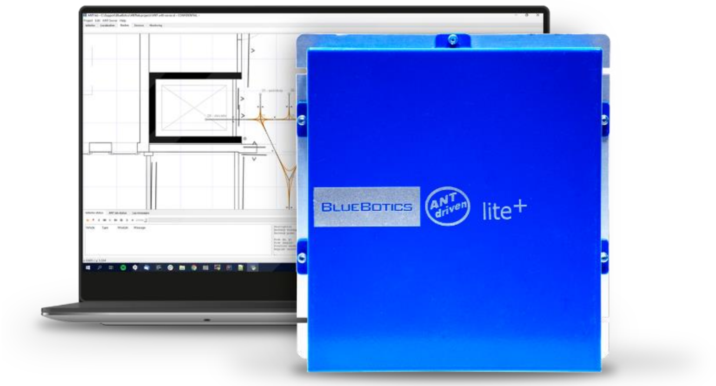


Figure 28: AGV control

ANT® lite+ is a vehicle control and positioning system for AGVs, automated forklifts and mobile robots. This complete natural feature navigation solution calculates the vehicle’s position (localization), controls its motion, and is interfaced directly with the vehicle’s safety laser scanners (SICK, Type…). ANT® lite+ uses laser scanner data and odometry to localize the vehicle in the map, with permanent structures (features) in the environment acting as references. The accuracy range for positioning is ±1 cm /±1°. Based on the X, Y and angle coordinates, ANT® lite+ controls the vehicle’s motion.

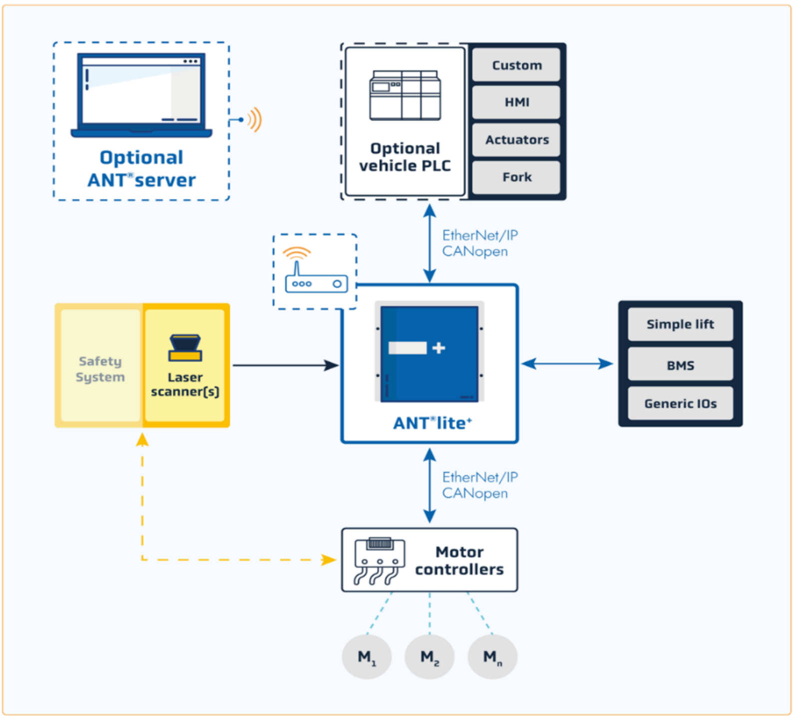


Figure 29: AGV System architecture

1.3.6 Wearable

1.3.6.1 Basic Concept

The wearable IMU captures acceleration and position data and performs an embedded gait analysis. The wearable, together with the UWB localisation, acts as a part of the hybrid localisation. The IMU must be attached to the operator's shoe. The complete data flow is seen in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

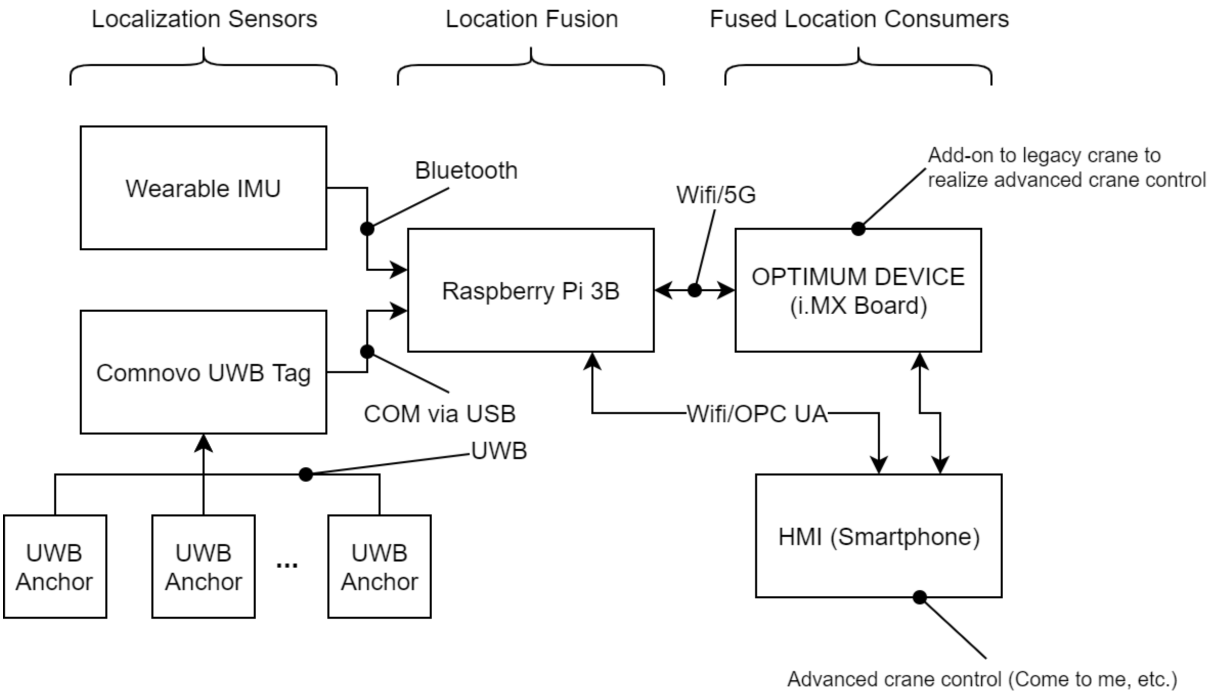


Figure 30: Data Flow of the Operator Localization

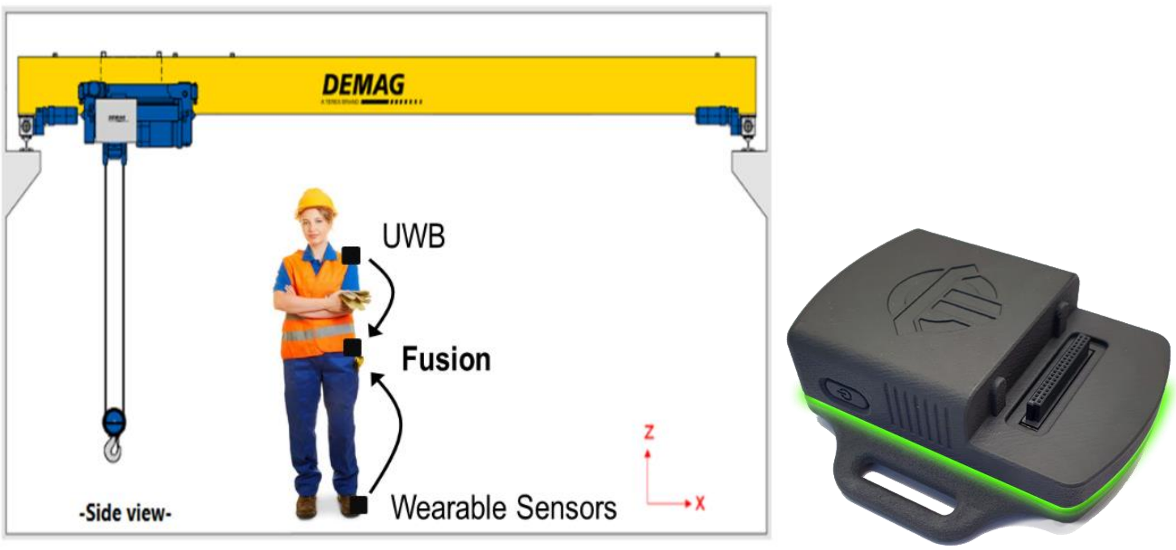


Figure 31: Use case scenario and wearable IMU

1.3.6.2 HW/SW-Specs

The wearable IMU consists of following main electronic components:

- STM32L452VETx – Microcontroller
  [8] https://www.st.com/en/microcontrollers-microprocessors/stm32l452re.html
- ESP32-WROOM-32U – Wireless connection
  [9] https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf
- BNO055 - Absolute Orientation Sensor
  [10] https://www.bosch-sensortec.com/products/smart-sensors/bno055.html#documents
- LEDs for status indication
- Pushbuttons (Power button, Reset Button, Pairing Button)
- Programming interface
- Battery and charging system with micro USB port

The BNO055 is an "absolute orientation sensor" with acceleration sensor, gyroscope and compass sensor. It is connected to the central STM32 microcontroller via I2C and can also be switched on and off via GPIO. The Thorsis Technologies IMU Library contains functions for initializing the I2C peripherals and for communicating with the BNO055 chip.

The wearable contains a firmware which determines the step length vector from the raw IMU data. If a step has been detected by means of ZUPT algorithm the step vector is sent to a processing unit via Bluetooth. More precisely, a raw data packet as well as the step length and direction of movement is sent.

The receiver side acknowledges each received step in the console. The processing unit is currently specified as:

- Raspberry Pi 3 Model B Rev 1.2
- Raspberry Pi 3 Model B+ Rev 1.3
- Raspberry Pi Zero W Rev 1.1

The processing unit combines the absolute position of the UWB positioning system with the relative movement data of the wearable IMU. The main component for a fused position is a Kalman filter architecture that models the users' movement change using the wearable IMU and combines the predicted position with the UWB measurement. Simultaneously, a continuous calibration of the users' orientation is achieved, using the step direction change from the IMU and the movement trend from the past UWB measurements in a separate Kalman filter. This calibration facilitates the continued tracking of the user in the case of missing or corrupted UWB positioning. As a side effect, the calibration also allows for a plausibility check of UWB data, by comparing the expected movement from the IMU data, with the measured position from UWB.

The processing unit publishes the users' location via OPC UA, using a lightweight version of the IIoT Platform.

1.4 SW Architecture

1.4.1 Overview

The software components are highly distributed and will be assigned to specific hardware components during engineering or setup of the equipment. First kind of software is IT related, e.g. the App Store, running on the PC or components with no direct control tasks e.g. to observe the movements. Such IT related components also run at machine side on the controller, on mobile devices or as wearables by the operators.

The second kind of software is control related which runs on the controller to directly interact with the control system of the machines. In Figure 32, the OPTIMUM controllers are illustrated with white colour while the yellow ones are the control systems, i.e., SafeControl on the cranes and ANT® lite+ system on the AGV.

In OPTIMUM, the whole system is protected by a firewall and specific access rights to the DEMAG sub network. It is guaranteed, that the access is limited to the App store or dedicated engineering tools.

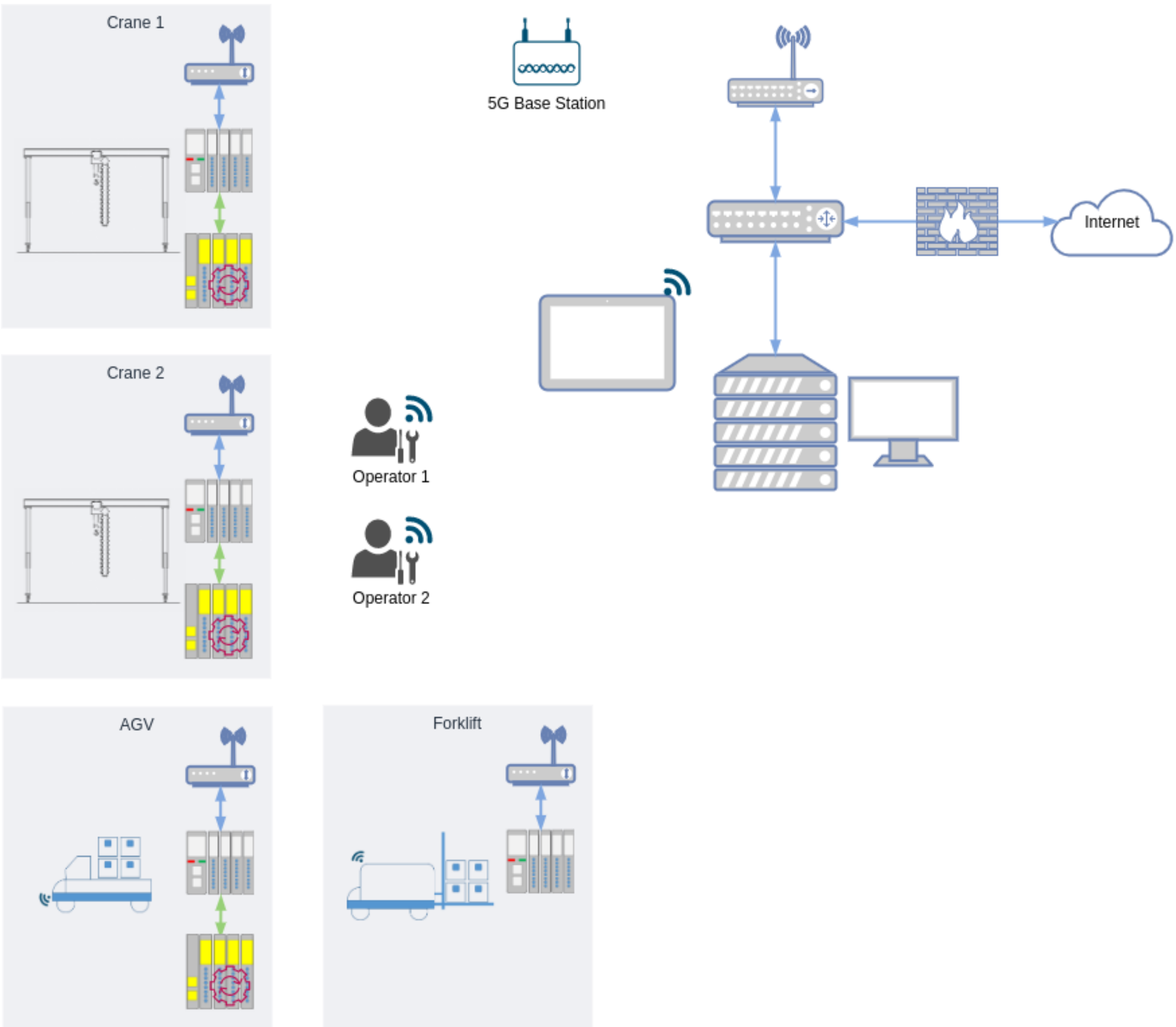


Figure 32: Architecture of involved control components

Out of scope is the configuration of the wireless 5G network. This is part of the implicit infrastructure and has no direct relationship to services of OPTIMUM software.

The specifics of the software components developed in OPTIMUM will be described in the next sub sections.

1.4.2 IIoT SW

The IIoT component of the OPTIMUM software is responsible for informational data exchange between industrial devices, human operators on behalf of HMI devices and other services like facility visualization. Furthermore, semi-autonomous functions can be invoked via IIoT that forwards these requests to the DCP component. The IIoT component has been developed in WP2.

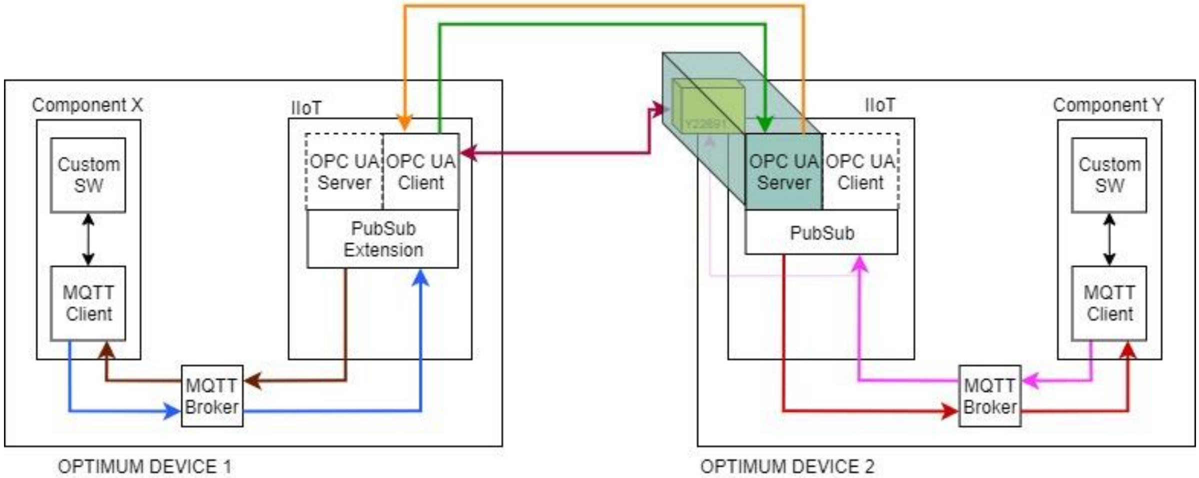


Figure 33: Communication flow between component X and Y of different devices using IIoT

As shown in Figure 33, the IIoT consists of an external interface based on OPC UA that enables an encrypted and authenticated communication with other OPC UA devices. The information data of an OPTIMUM device is provided as an OPC UA data model that can be read from any OPC UA client. The data model used in the DEMAG demonstrator consists of a common base for all OPTIMUM device and a device-specific part following a predefined hierarchy. Additionally, the data model integrates the OPC UA Companion Specification “OPC UA for Machinery” that provides a vendor-independent and cross-industry set of machinery information similar to a machine nameplate and a standardised structure for accessing the component of the devices. This enables a basic M2M communication between OPTIMUM devices and other OPC UA devices and clients that follow this specification.

The second interface of the IIoT component is for device internal communication between components of one device. This communication is based on MQTT and allows other components of the device to use the OPC UA capabilities provided by the IIoT component for external communication.

The IIoT component translates OPC UA requests into MQTT messages and vice versa. This allows the invocation of semi-autonomous functions via an HMI that is connected via OPC UA to the IIoT component. The function request of the HMI is validated and converted into a MQTT message and send to the DCP via an MQTT broker. Furthermore, components of the OPTIMUM device can instruct the IIoT via MQTT to retrieve data of other devices once or on a subscription base. As an example, this allows the DCP to request the location information from the wearable.

Two different IIoT versions are used in the DEMAG demonstrator. The cranes and the AGV are equipped with the (full) IIoT component as described above. The fork lift and the wearable run a lightweight version of the IIoT component. Since the fork lift and the wearable have no active control tasks and offer only their location data, no DCP component is required. Additionally, the CAN interface of the UWB location tags is replaced by a serial connection. The lightweight version of the IIoT component contains also the OPC UA interface and the data model, but the MQTT interface is replaced by an inter-process-communication (IPC) to a process that reads the UWB location data from a serial interface. The lightweight OPC UA data model does not provide the invocation of semi-autonomous functions.

The IIoT component for crane and AGV contains OPC UA methods to activate semi-autonomous functions like *GoToPosition*, *ComeToMe* and *FollowMe*. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows these method calls and the data model of a DEMAG crane with two trolleys. *Method_JSON* is a general-purpose method that can be used to transmit any information to a specific component of the target device using JSON data encoding. Depending on the method, different input and output arguments are available. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows a screenshot of the OPC UA client UAExpert that calls the method *ComeToMe* with specific input arguments. The output argument reflects the response of the IIoT component of the target device.

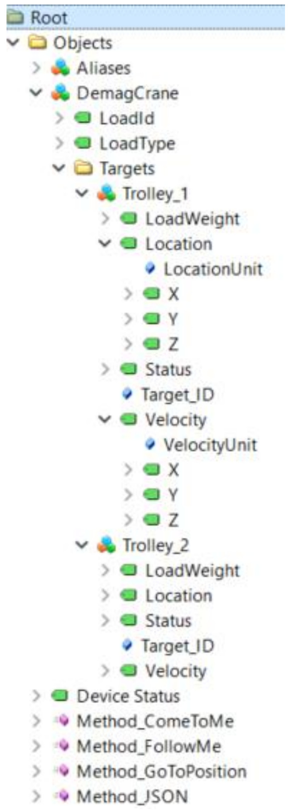


Figure 34: Data Model of a DEMAG crane

The methods *GoToPosition*, *ComeToMe* and *FollowMe* can be combined with the *FollowMachine/Tandem* function that allows simultaneous movement of multiple cranes and trolleys. The function can be activated by adding values to the input arguments “Slave Devices” and “Slave Trolleys”.

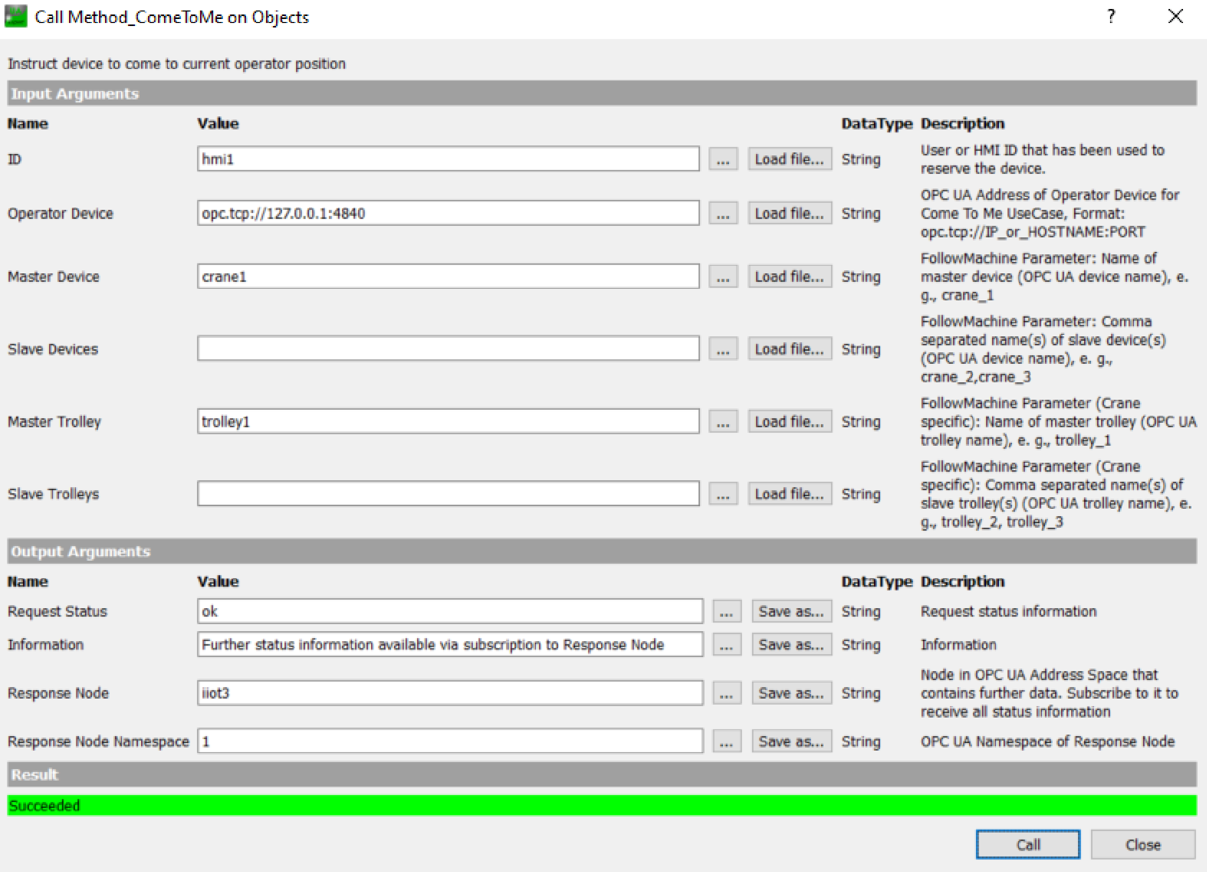


Figure 35: Input and Output Arguments of ComeToMe Method using OPC UA client UAExpert

1.4.3 DCP SW

The control applications on the crane and the AGV are applications run on the DCP developed in WP3. In the concept of the DCP, an application is composed of multiple objects connected via links. This allows the modularization of the application. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows an overview of the control application and the interfaces to other components in the system.

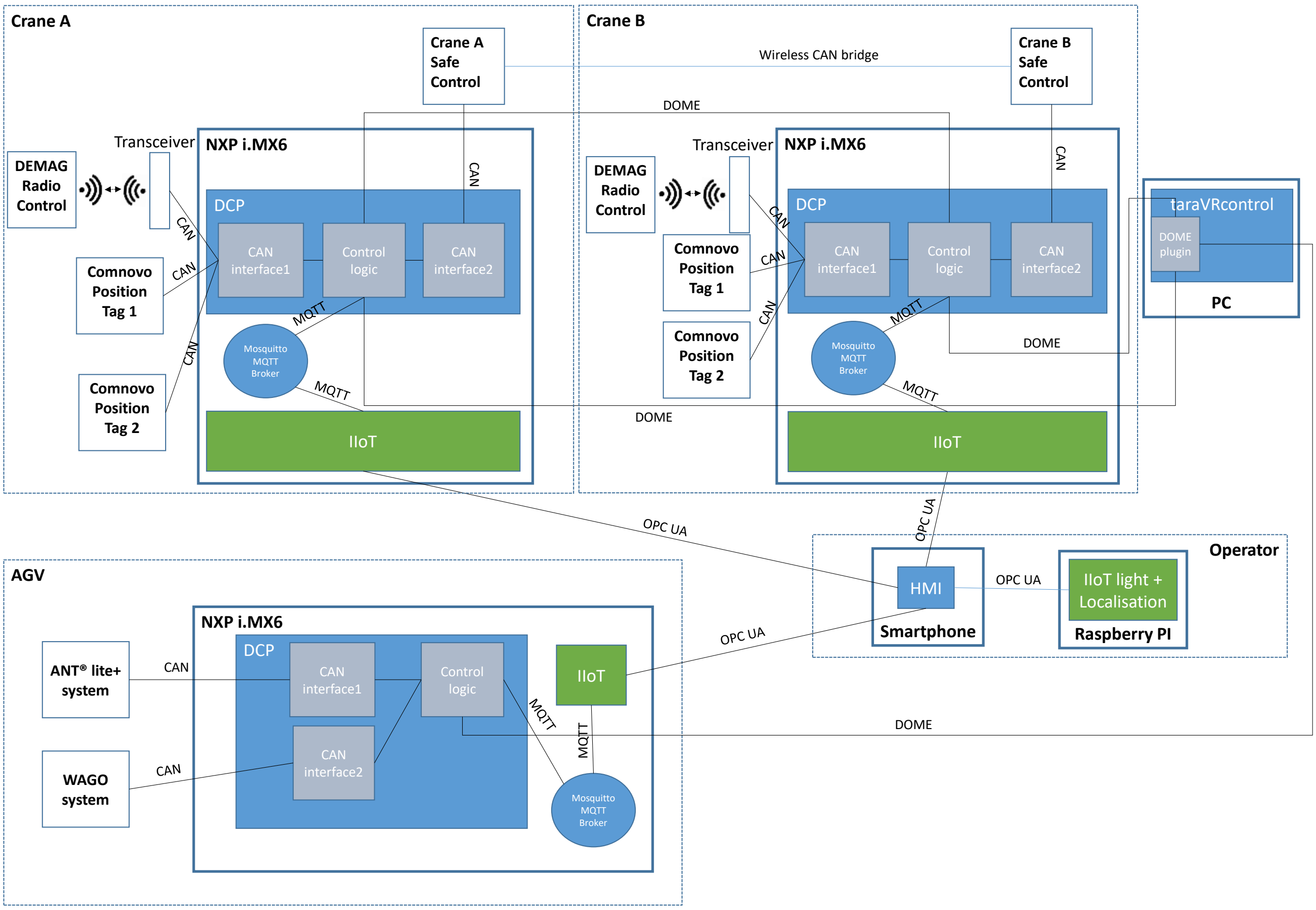


Figure 36: Control application and the interfaces to other components

In the control application, the objects can be generally categorized into interface objects and control objects. The interface objects are used for the interfacing with hardware parts of the system e.g. DEMAG radio control, crane safe control, Comnovo position tags and AGV ANT® lite+. In the MH-Demonstrator, only CAN bus is used so a CAN module is developed and used for both crane and AGV. On each, there are two CAN objects, which correspond to two CAN interfaces on the imx6 board. On the crane, the first CAN interface connects to the radio control and position tags. The second interface connects to safe control. On the AGV, the first interface connects to the ANT® lite+ system while the second one connects to a WAGO system, which controls the power source to the AGV control system and provides digital input. The CAN module uses SocketCAN, provided by the operating system of the embedded board. SocketCAN supports different CAN interface types such as native CAN, serial based CAN (slcan). It also supports different bitrates. All of these are configurable in the configuration files of the control application.

The control logic is the part that defines the functionalities of the crane or the AGV. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows the overview of control logic part of crane.

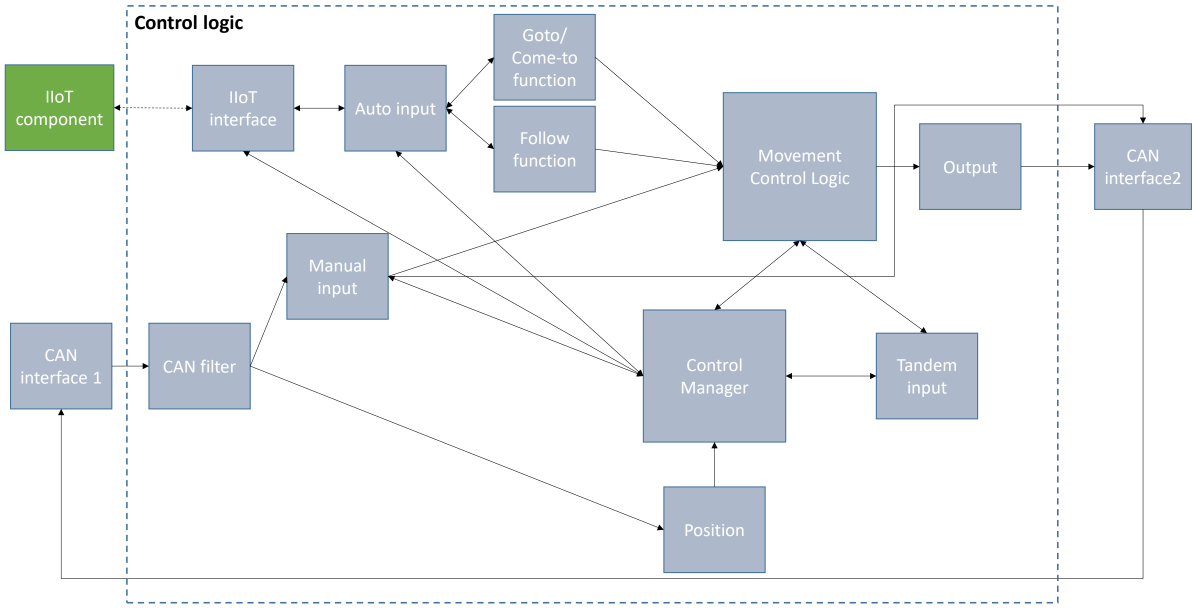


Figure 37: Control logic of crane application

The crane control application support two modes, which can be configured using an HMI:

- **Legacy mode:** In this mode, the crane behaves like a conventional crane, which can be configured into separate control mode or group control mode as described in **Fehler! Verweisquelle konnte nicht gefunden werden.**. The crane is controlled solely by the radio control and the tandem is performed via the wireless CAN bridge. In this mode, the control logic acts simply as a bridge between two CAN interfaces, no data processing is performed on the crane CAN messages. The position data from the position tag is still processed and provided to the visualization tool.
- **Optimum mode:** The crane must be configured into separate control mode before activating this mode i.e. each crane is controlled by its own radio controller. When optimum mode is activated, smart functions and other advanced functionalities can be used. Data coming from the radio control is processed to detect the movement commands, which are used for more advanced functions e.g. collision avoidance. This mode also supports tandem movement. The difference is that the new tandem is performed over the 5G network instead of the legacy wireless CAN. The new tandem is activated using the HMI. The movement commands from master crane are sent directly to the slave crane using DCP protocol.

Crane control logic can be generalized into following objects/groups of objects:

- **CAN filter:** CAN interface 1 is connected to both radio control and position tags. This object is used to separate the data coming from these two sources.
- **Position:** This group of objects processes the CAN messages to extract the position and velocity data. The data is fed to the control manager for further use and is also converted to format that used by taraVRcontrol for visualization.
- **Manual input:** This object is responsible for handling data from radio control. In Legacy mode, this object simply forwards everything to CAN interface 2. In Optimum mode, when manual mode is enabled, this object examines the PDOs, which contains info about the selected trolley, direction and velocity, to extract required info for further use. Other CAN telegrams are forwarded directly to the other CAN interface. When smart functions are activated, this object will block all movement commands coming from radio control.

- **Auto input:** This object is responsible for the handling of smart function commands coming from IIoT component. Based on the command, it will enable the corresponding functions such as goto/come-to or follow.
- **Goto/Come-to function:** This object is used for both goto and come-to functions because basically, they are the same. This object continuously checks the crane current position and the target position to produce the movement commands on x, y and z axis separately. Once the target is reached, it reports back to **Auto input** to end the smart function.
- **Follow function:** This object is similar to goto/come-to object. But instead of receiving target position once at the beginning, it keeps getting the updated position of the target in order to follow. The function will not end when the target is reached. The function only ends when receiving command from IIoT component or when no new target position is received for a predefined time period.
- **IIoT interface:** This object is responsible for handling requests and providing responses to IIoT component via MQTT. Since IIoT uses a different data structures from object's interface port, this object is also used for further data exchange between the control application and the IIoT component e.g. position data, status data etc.
- **Control Manager:** This group of objects is used for:
    - Configuring the crane application based on the selected mode (Legacy/Optimum)
    - Enabling/Disabling the required inputs (Manual/Auto/Tandem). It makes sure that only one source of input is enabled at a time. One exception is in the case of master crane in tandem mode, the tandem is enabled along with a selected source such as manual or auto input.
    - Keeping track of crane related info: status, position etc.
    - Enabling tandem function when this crane is selected as tandem master.
    - Collision avoidance
    - Path planning
- **Movement Control Logic:** Each object of this group is responsible for the movement on x, y and z respectively. The object receives the movement commands coming from the internal logic and checks with the blocking information from **Control Manager** to prevent collision.
- **Output:** This object is used to translate from the generic control data to the CAN telegrams which are then sent via CAN interface 2 to the crane safe control. It is also used for additional functions such as turning on the blue light or producing heartbeat to the safe control when smart functions are activated.
- **Tandem input:** This group of objects is used when tandem is activated. On the master, the tandem master object is enabled and, on the slave, the tandem slave object is enabled. The master object is enabled along with a selected source of input (Manual/Auto) to forward the movement commands to the slave crane. On the slave crane, when tandem slave object is enabled, other inputs (Manual/Auto) are disabled to only accept the commands from master crane.

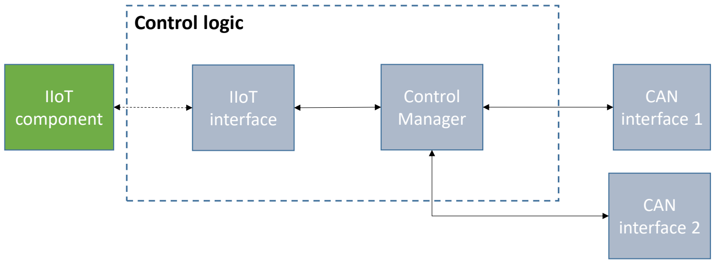


Figure 38: Control logic of AGV application

The ANT® lite+ system on the AGV is responsible for controlling the AGV as well as obtaining the position information. To integrate the AGV into OPTIMUM platform, the DCP application is developed to interact with this AGV control system via CAN bus on CAN interface 1. The DCP application is also used to interact with a WAGO system on CAN interface 2 for additional functionalities such as supplying power and providing input signal. The required function for the AGV is to go to predefined positions on the map, which are stored in the ANT® lite+ system. The control logic of the DCP application is therefore, much simpler than the one on the crane. The control logic, which is shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**, contains following objects:

- **IIoT interface:** is used to handle commands from IIoT as well as provide position information obtained from AGV system to IIoT data model.
- **Control Manager:** This object is responsible for:
    - Starting up the ANT® lite+ system: the power to the ANT® lite+ system is controlled by DCP application via a WAGO system. This is to ensure that the OPTIMUM control hardware is started up before the AGV control system. Otherwise, the AGV control system will stop listening on the CAN bus.
    - Extracting position and rotational information from CAN messages coming from AGV control system. This info is then fed to the IIoT information model and to taraVRcontrol for visualization.
    - Commanding AGV control system to go to a desired position: on the AGV control system, there is a map which contains some predefined points where the AGV can go. The **Control Manager** object is responsible for generating CAN message which commands the AGV to go to one of these points.

Providing digital input to AGV control system: the goto function on the AGV is triggered by a physical switch. In order to fully automate this function, the physical switch is replaced by a digital output of the WAGO system, which in turn, is controlled by the **Control Manager** object.

1.4.4 Security Server SW

The so called Security Server is first of all a backend server running services that are needed to fulfil security requirements of the system. A webservice is used to wrap several services in an interface presented to the devices in the network of the OPTIMUM system.

The Security Server is based on a Linux operating system and supports the container service Docker, which is used to run the several services hosted on the Security Server.

1.4.4.1 Webservice

The webservice provides the interface for services for the OPTIMUM devices running in the background of the server such as a Certificate Authority or a user and device database. It is implemented using a RESTful API based on the Spring framework. Spring offers a lot of ready to use packages that makes it relatively easy and secure to set up different features in the webservice without starting from scratch.

For the security aspect of the webservice with regards to the HTTP protocol and the user login Spring Security has been chosen. It provides built-in functionality for user and authorization handling as well as for security related topics like e.g. CORS. Using that framework different aspects of the HTTP communication can be easily be addressed:

- Comprehensive and extensible support for user authentication
- Sharing authorizations in the OPTIMUM network
- Easy integration of JWT for session control
- Easy integration of database access using ORM with Hibernate

The webservice offers different endpoints to the network participants. That covers user login endpoints for 2FA authentication as well as endpoints for certificate creating, signing and checking.

The architecture of the webservice is open to integrate additional functionality of different type so that it is highly extensible.

It has been setup as a Docker image to ease automated usage and update. Starting and restarting the container does a complete pull of the GitHub repositories master branch and recompiles the target completely before starting.

1.4.4.2 Certificate Authority

OPTIMUM needs to have its own PKI so that a company using OPTIMUM can have full control over the participants within the network. In order to achieve this goal a local service for managing certificates – a so called Certificate Authority (CA) – is running on the Security Server.

This CA covers a variety of functionalities with regards to certificate handling:

- Issuing normal and also short-lived certificates
- Renewal
- Revocation
- Signing

The software package used to provide these services is Smallstep. It is an online CA for secure, automated X.509 and SSH certificate management.

Smallstep has been set up on the Security Server as a Docker container running locally. Its functionality to the outside world is wrapped by the OPTIMUM webservice. Thus, the access to the CA functionalities can easily be controlled by the webservice through dedicated endpoints using user authentication and authorization.

1.4.4.3 NFC based User Authentication

For User Authentication a two-factor authentication procedure has been implemented. Therefore, a solution has been chosen that integrates the login process into the HMI app on the tablet or smartphone that is used by the operator during work time. As a basic requirement the tablet / smartphone needs to have an NFC interface.

The first step of user login is done by presenting the operator badge to the tablet. After hitting the login button, the Xamarin based HMI app is requesting the identifier for the application as well as for the file within the application on the DESFire card from the RESTful webservice on the backend server. An NFC library is handling the access to the NFC card using this information. In the current implementation a NXP DESFire EV2 card has been chosen to store the user information for login.

The NFC library reads the needed information from the chosen file within the selected application and returns it to the webservice. Here the card content needs to be decrypted by a SAM containing the symmetric keys use for encryption. The decrypted card content, e.g. the personnel number of the operator, is used to check if the requested person does exist in the database and thus is known to the system. In a next step the HMI app is requesting the operator's password as the second factor for authentication.

Once the password has been checked by the webservice the personal information of the operator is read from a database in the background. That comprises not only the name and other personal information but also the authorizations granted to the operator.

The authorization information is passed back to the HMI app where it is used to create a certificate signing request (CSR) for a short-lived certificate.

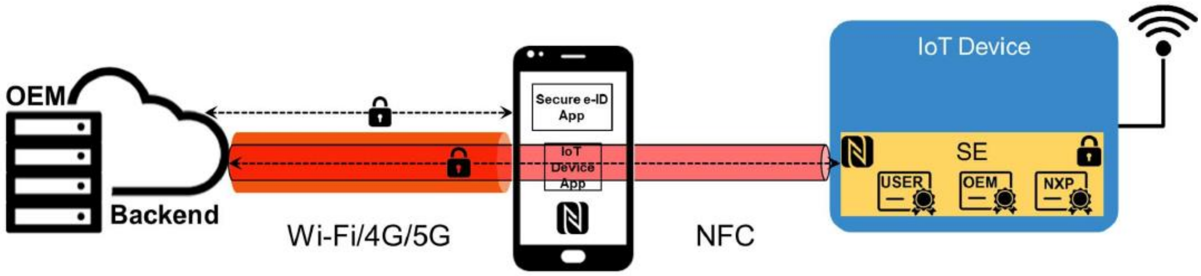
This CSR is sent to dedicated endpoint of the webservice that passes on the CSR to the OPTIMUM CA working in the background of the backend server.

The OPTIMUM CA checks the CSR and in case of success signs the certificate, which is eventually sent back to the HMI app. Now the mTLS secured connection to the webservice can be closed and the HMI app can connect to an OPTIMUM device using the new issued short-lived certificate.

1.4.4.4 Provisioning

When a new OPTIMUM device has to be added to the OPTIMUM network it needs to be equipped with a security anchor that allows seamless network integration. The security anchor in this case is a device certificate containing device unique information. This certificate has to be created and stored on the Secure Element (SE) of the OPTIMUM device during the provisioning phase.

The solution for this task is a smartphone app that establishes first a TLS secured connection to a backend service on the one side and then an NFC connection to the OPTIMUM device to be integrated on the other side.



It makes use of the ability of the SE that is embedded in the OPTIMUM device to operate powered only by the NFC field.

Once the OPTIMUM device is held into the NFC field of the smartphone the SE starts to communicate. The smartphone app in the middle between backend service and Secure Element acts as an agent to establish a direct SCP03 connection between the backend service and the OPTIMUM device.

Using this connection, the backend service requests information about the OPTIMUM device in order to create a device certificate that contains unique information about the OPTIMUM device and stores the certificate in the SE.

Once that is done the provisioning is complete and the OPTIMUM device can be used in the OPTIMUM network.

1.4.5 App Store

The Application Store is a web application that employs a NodeJS backend and a ReactJS and MaterialUI frontend. It requires user authentication and authorization from the client for performing tasks. The user database is held in a local file on the backend server as JSON data.

The frontend is a ReactJS application, transpiled into javascript by react transpiler and served as static content by the Express web server. The frontend is served from the url base path "/", and the backend is served from the url base path "/api".

Application Store can be run as a Docker container by first building the images and then starting the containers using the Docker compose tool. The complete Application Store suite has 3 services: Application Store server, Authentication server and Docker Registry server. All of these services must be configured properly and started in order to run the Application Store application.

Fehler! Verweisquelle konnte nicht gefunden werden.39 depicts the modules and connections of the Application Store. For more information on App Store, please refer to deliverables D4.6 and D4.7.

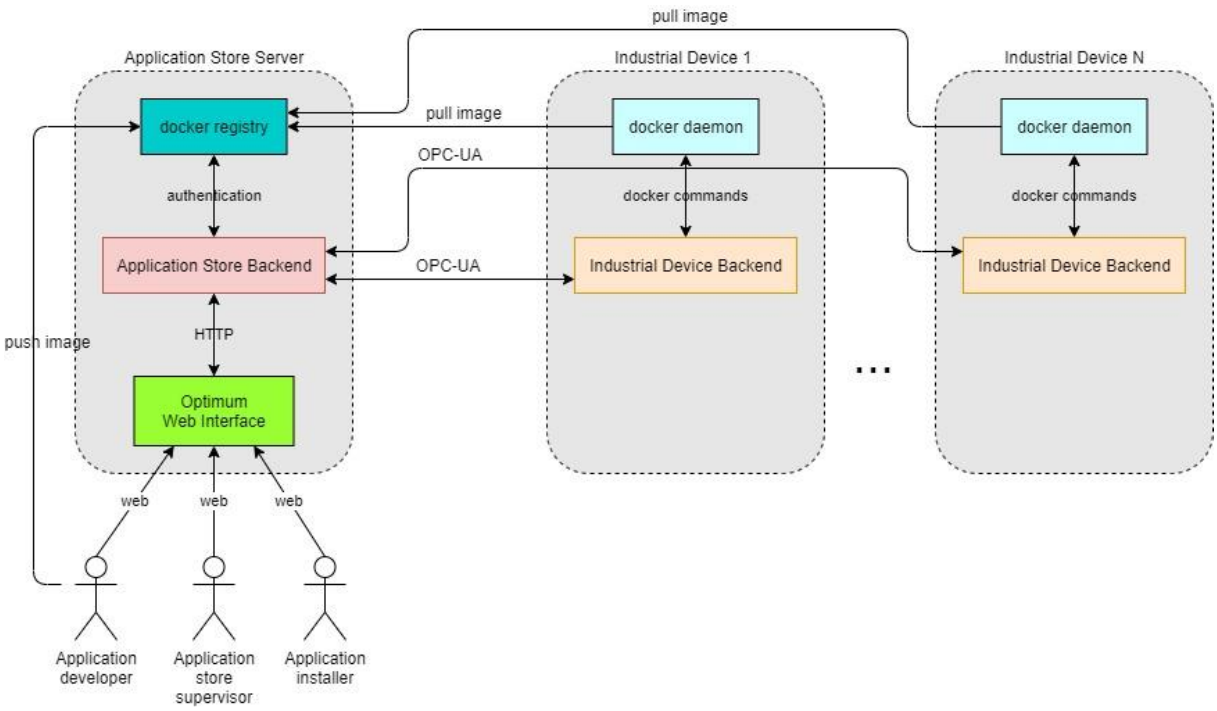


Figure 39: Modules and connections of Application Store

1.5 3D-Visualization and Engineering Tools

The creation of the "digital twin" of the assembly hall with all actors – operators and machines- was worked out with the 3D-visualization tool ”TaraVRbuilder” from the OPTIMUM partner TARAKOS. With this tool the scenario can be visualized and simulated.

This visualization software is often used when logistics and assembly processes are planned, optimized and implemented. The CAD plan of the hall served as the basis for a 3D scenario. The components that can be animated, such as vehicles, work and assembly stations, goods to be conveyed or storage systems, are taken from the object libraries contained in taraVRbuilder. They can be placed with the mouse and then animated with parameter inputs. Some Demag's predominantly customized components were designed in the Inventor 3D CAD system and imported into taraVRbuilder via suitable interface.

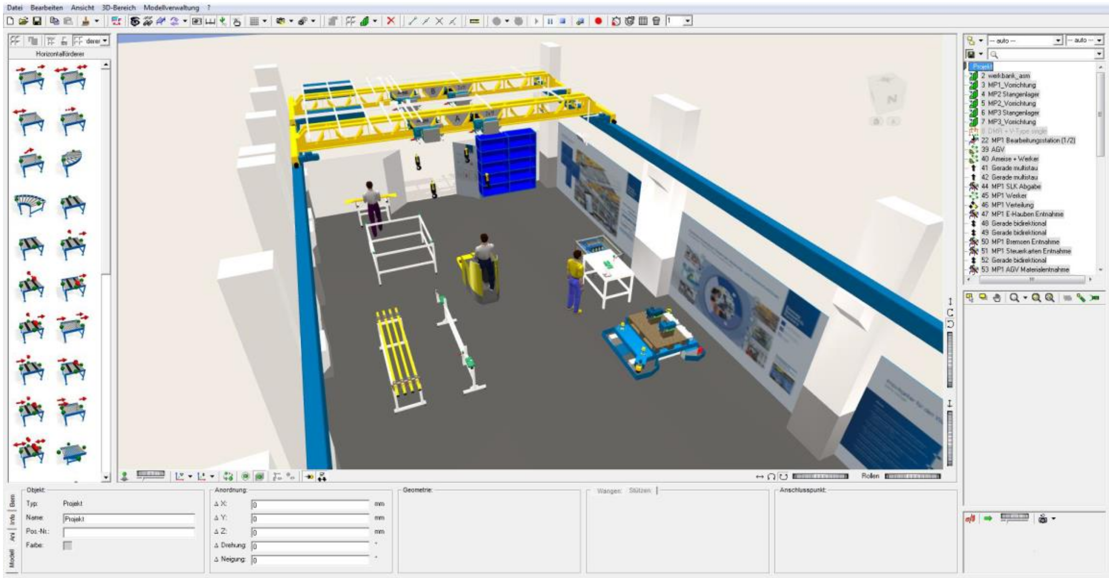


Figure 40: 3D modelling Tool “TaraVR-Builder” from TARAKOS

Figure 41: Material flow simulation in the DEMAG research factory

Signals and data from the processes are recorded and analyzed in real time in the research factory. Another Tarakos product serves this purpose: the “TaraVRcontrol” software tool is used for 3D process visualization in automation technology. The "digital twin" of the research factory created in “TaraVRbuilder” is read in and can be animated with real data from PLC or OPC servers, taking into account all the dependencies already defined. The position of the cranes, the movement of the hook and other properties follow the signals from the automation technology.

By further developing the visualization and simulation tools used today, we have in the OPTIMUM project simplified these engineering steps and apply them in a methodical way.

Thereby the increased complexity can be mastered and the transparency can be increased. OPTIMUM even goes one step further and requires not only suitable 3D models of the machines, but also corresponding data models; So-called "digital twins" can be developed later and a virtual image of the smart factory can be created, which can be used at a later date for plant visualization and monitoring by the customer during operation, in the utilization phase of the factory.

The 3D modelling is to be carried out using libraries which the machine manufacturers make available to the Smart Factory customers. As mentioned above, this includes not only the 3D geometries but also the data models.

The 3D engineering tools were used during different steps of the demonstrator engineering process. At first a 3D model of the DEMAG-Research Factory was designed to visualize the demonstrator. The basis for this were implementations in WP5, which enable the visualization of different types of load handling in the planning tool without the user having to do any programming.

For the next step, the created 3D-model was used for the realtime-visualization. Also in WP5 a Plugin was developed, to connect the DCP with the visualization-tool taraVRcontrol.

This real-time visualization shows:

- cranes and trolleys at their current position
- movement and rotation of AGVs and human operators
- Status of the cranes (e.g. active assistance function)
- Preview of the current movement path

Figure 42: **Demag cranes and associated real-time visualisation**

Third contribution will be a software application for “hardware in the loop” DCP-simulation. Digital behavioral models interact with the DCP. The DCP sends orders to cranes and AGVs which will be executed by the digital models. Furthermore, the simulation contains digital models of workers and operators, which move at randomized paths inside the factory and communicating their position to the DCP. This allows to check the DCP functionalities, in particular corresponding to moving obstacles, before hardware installation at the shopfloor.

1.6 Process flow in the demonstrator line

The assembly process of the chain hoists in the production line is realized in two different scenarios:

1. material flow and handling utilizing conventional machines
2. material flow and handling utilizing the enhanced OPTIMUM machine functions

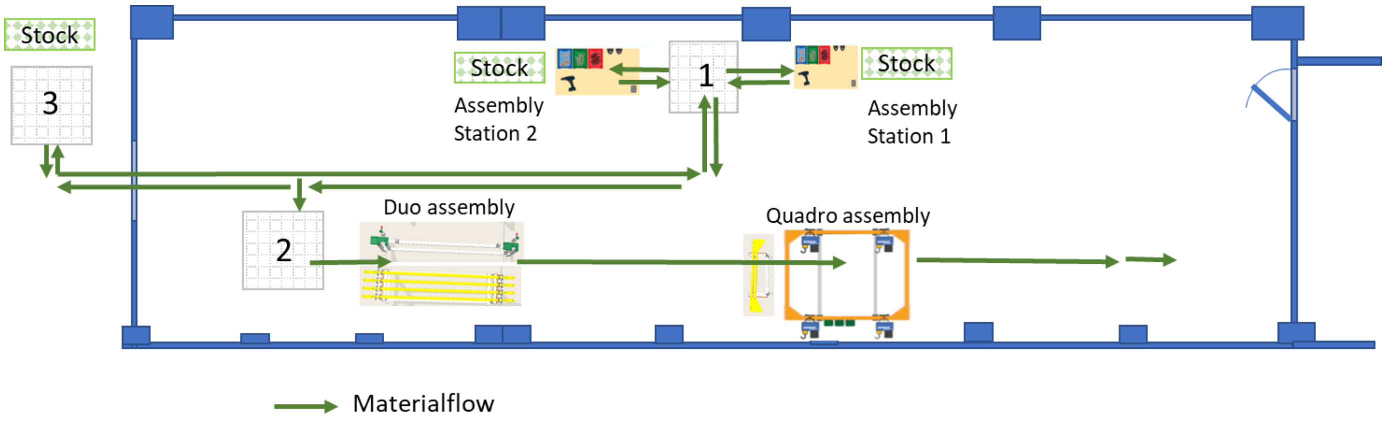


Figure 43: Layout of assembly line – Top view

	Action	Actor	Control by	Start Position	Target Position
1	Material Transportation	Forklift	manually driven	3	1
2	Material Transportation	Crane 1	Operator 1	Forklift	Assembly Place 1
3	Material Transportation	Crane 2	Operator 2	Forklift	Assembly Place 2
4	Assembly single Chain Hoist				
5	Material Transportation	Crane 1	Operator 1	Assembly Place 1	Forklift
6	Material Transportation	Crane 2	Operator 2	Assembly Place 2	Forklift
7	Material Transportation	Forklift	manually driven	1	2
8	Material Transportation	Crane 2	manually driven	Forklift	Duo Assembly place
9	Material Transportation	Crane 2	manually driven	Duo Assembly place	Forklift
10	Material Transportation	Crane 2	manually driven	Forklift	Duo Assembly place
	Material Transportation	Forklift	manually driven	2	3
11	Assembly Duo Chain Hoist				
12	Material Transportation	Crane 1 + 2	manually driven	Duo Assembly place	Quadro Assembly place
	repeat steps 1 to 12				
25	Assembly Quadro Chain Hoist				
26	Material Transportation	Crane 1 + 2	manually driven	Quadro Assembly place	Output station (ready)

Table 4: Process steps in scenario 1

	Action	Actor	Control by	Start Position	Target Position
1	Material Transportation	AGV	automatic	3	1
2	Material Transportation	Crane 1	Come to me	AGV	Assembly Place 1
3	Material Transportation	Crane 2	Come to me	AGV	Assembly Place 2
4	Assembly single Chain Hoist				
5	Material Transportation	Crane 1	Go To	Assembly Place 1	AGV
6	Material Transportation	Crane 2	Go To	Assembly Place 2	AGV
7	Material Transportation	AGV	automatic	1	2
8	Material Transportation	Crane 2	Come to me/Go To	AGV	Duo Assembly place
9	Material Transportation	Crane 2	Come to me/Go To	Duo Assembly place	AGV
10	Material Transportation	Crane 2	Come to me/Go To	AGV	Duo Assembly place
	Material Transportation	AGV	automatic	2	3
11	Assembly Duo Chain Hoist				
12	Material Transportation	Crane 1 + 2	Tandem Go To	Duo Assembly place	Quadro Assembly place
	repeat steps 1 to 12				
25	Assembly Quadro Chain Hoist				
26	Material Transportation	Crane 1 + 2	Tandem Go To	Quadro Assembly place	Output station (ready)

Table 5: Process steps in scenario 2

The process has been first planned, visualized and simulated (as described in the previous chapter 1.5) and finally executed and verified in reality.

1.7 Testing and verification

The total time to assembly a Quadro-Chain-Hoist have been simulated for both scenarios:

1. assembly time = 13’ 41”

2. assembly time = 11’ 12”

The comparison of the assembly time shows a time saving of approx. 18%

The process steps and the related time for assembling have been the same for both scenarios. The time saving is gained by the activation of the assist functions of the cranes

- Come to me
- Go To
- Follow AGV

and the autonomous movements of cranes and AGV.

The first tests in the real scenario seem also to be giving good results: Considering the small factory area in the demo-line is the saving already quite impressive.

More in-depth and detailed tests will be carried out in the coming months in D6.4.

2 ETRI Demonstrator

2.1 Overview

The Korean consortium conducted an experiment to verify the scenario of automatically controlling robots through HMI. Both robots and HMI are implemented to communicate wirelessly through the network architecture developed by the OPTIMUM consortium. We have constructed a scenario where robot 1 is continuously executed "Go to" function, and robot 2 executes "Follow machine" function to follow the movement of robot 1. Through this experiment, we were able to verify that the real-time communication of the optimum architecture is suitable for carrying out the scenario.

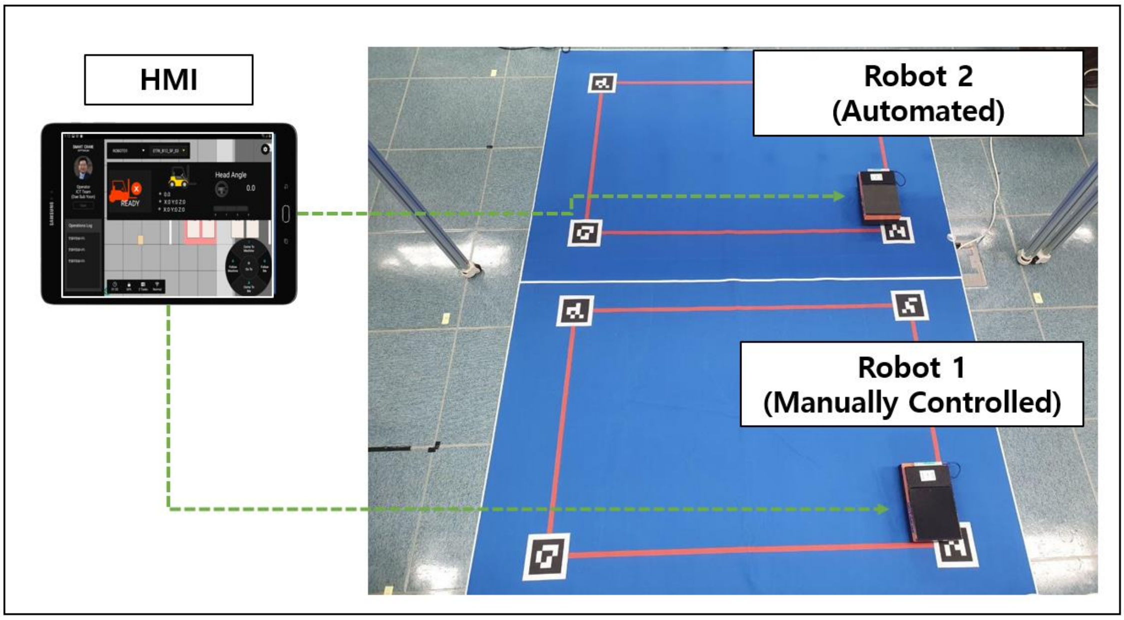


Figure 44: Overview of the test environment of the Korean Consortium

2.2 Demonstrator design

2.2.1 MAP design

The two robots used in the demo are driven on their respective maps. The width and height of the map are 165 mm x 270 mm, respectively. In the HMI, the coordinates that the robot can input are set to a maximum of 1500 for x and a maximum of 1000 for y.

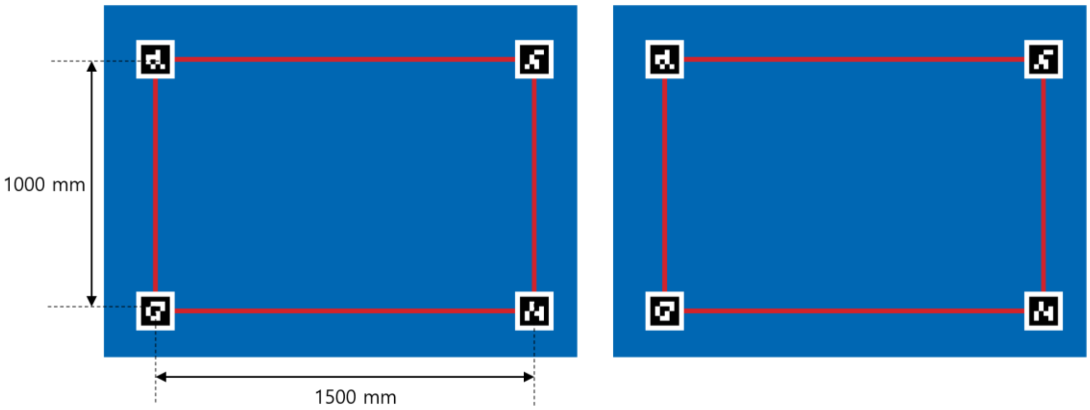


Figure 45: Map design

2.2.2 Robot design

The robot contains two wheels, each of which is driven by an independent motor. We implemented two-wheel robot by using "Gopigo 3" manufactured by Dexter. It has motors with encoder function, a control board, and a battery. Raspberry Pi 3B+ was combined with the robot, and the program was designed so that the robot can perform the tasks.

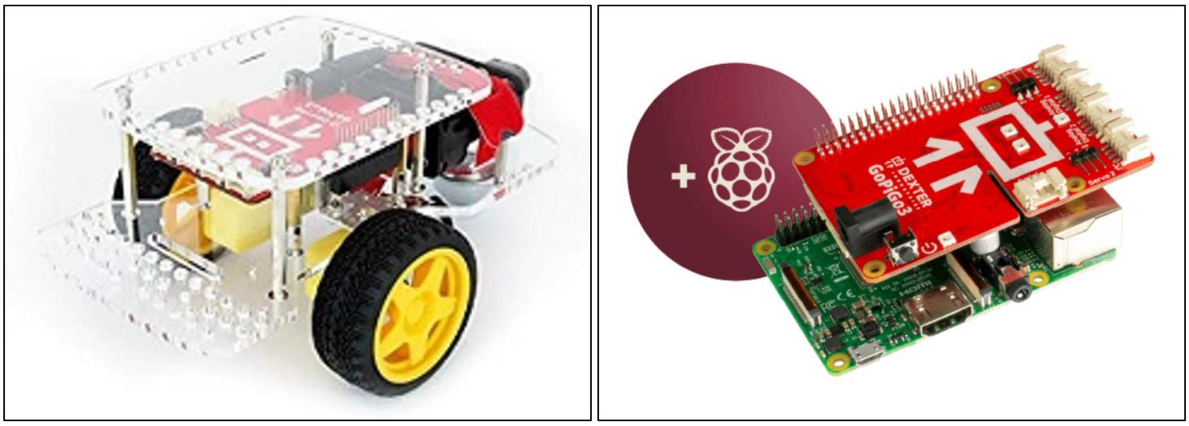


Figure 46: Two-wheel Robot (Gopigo 3)

The key features of Raspberry Pi 3 B+ are as follows.

"Raspberry pi 3B+" Key Features

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

In order for the robot to receive coordinates from the HMI and move to the corresponding position, the robot has to know its own position. The localization method using UWB usually generates an error value of about 30 cm. Because the map used in the demonstration is small, so the effect of the error value is very large.

Considering the size of the demo map, we used a vision-based localization sensor with a small error and enabled the robot to perform self-position estimation. SLAM sensor T265 manufactured by Intel is used and connected to a Raspberry Pi via USB. It includes two fisheye lens sensors, an IMU and an Intel® Movidius™ Myriad™ 2 VPU. It has the advantage of reducing the computational load of the Raspberry Pi because the operation processing required for SLAM is processed by the built-in VPU and only the result value is transmitted to the Raspberry Pi.

Figure 47: Vision-based localization sensor (Intel Realsense)

The key features of the T265 sensor are as follows.

"Realsense T265" Key Features

- Intel® RealSense™ V-SLAM technology
- Intel® Movidius™ Myriad™ 2.0 VPU
- 2 fisheye lenses with combined 163° FoV (+/- 5°)
- BM1055 IMU
- USB 3.1 Gen 1 Micro B connectors
- Dimensions: 108 x 24.5 x 12.5 mm
- Two M3 0.5 mm mounting sockets
- Camera supplied with tripod
- OS (System requirement): windows 10 / Linux

In order to combine the robot and the sensor, the upper part of the robot was designed and manufactured as a 3D printer. The horizontal and vertical length of the upper part are 165 mm x 270 mm, respectively.

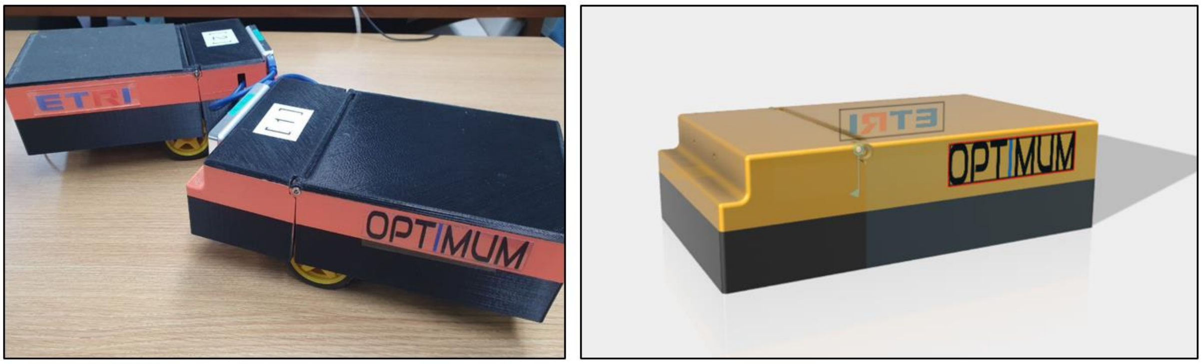


Figure 48: Robot mock-up

2.2.3 HMI design

HMI is designed to operate on Android OS, and we applied it to two devices. The difference between the two HMI devices is whether or not they have an NFC module. A version with an NFC-based login system and a version with a login system through typing were produced. Recently released Android OS tablets tend to be released without NFC modules.

Figure 49:50 HMI Device (Galaxy tab active2 / Galaxy tab S4)

2.3 HW architecture

2.3.1 Overview

The hardware consists of an HMI device and two material handling robots. The robots are operated in software by the Raspberry Pi3 B+ and each device has IIoT, DCP software and device controller as well.

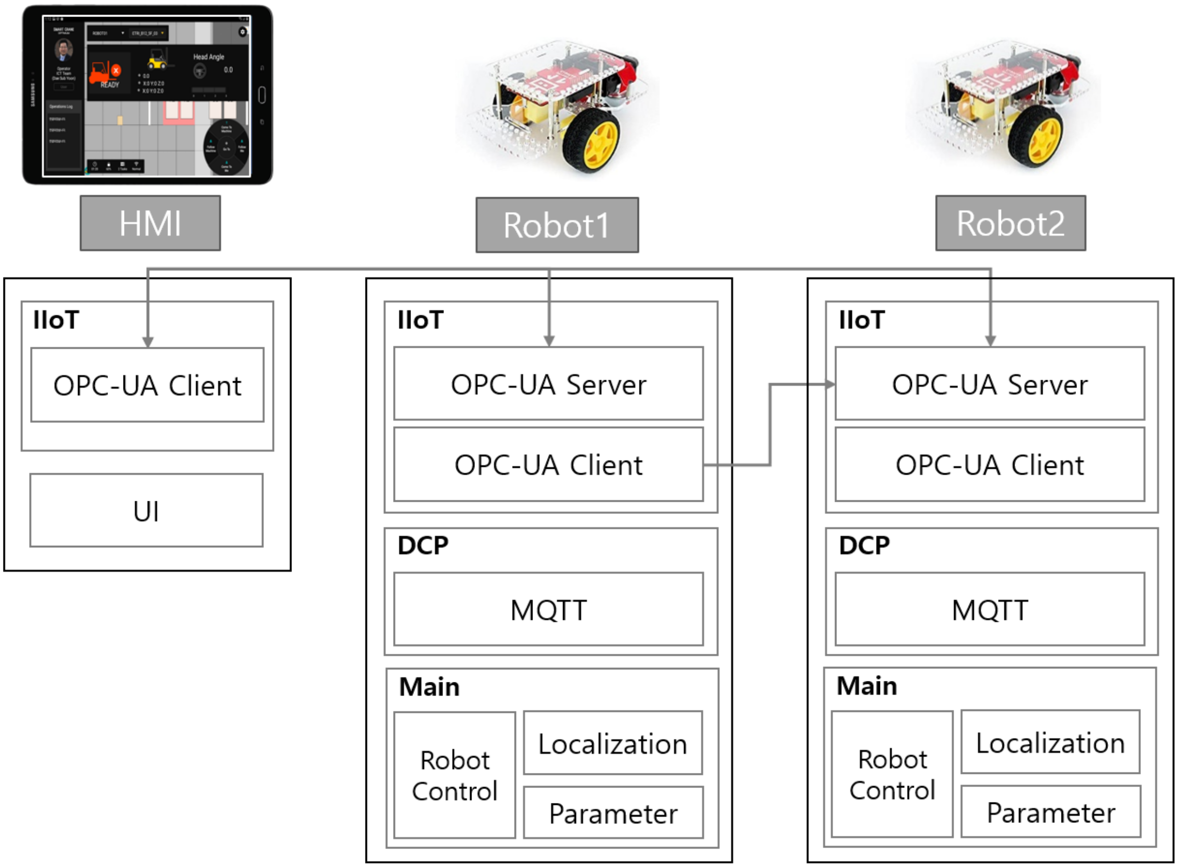


Figure 51: HW architecture of robots and HMI

2.3.2 Localization

The localization sensor creates a surrounding map through visual-SLAM technology and performs self-position estimation. Since the sensor sets the position to the zero point when power is supplied, the robot has to be operated after placing it at a preset starting point.

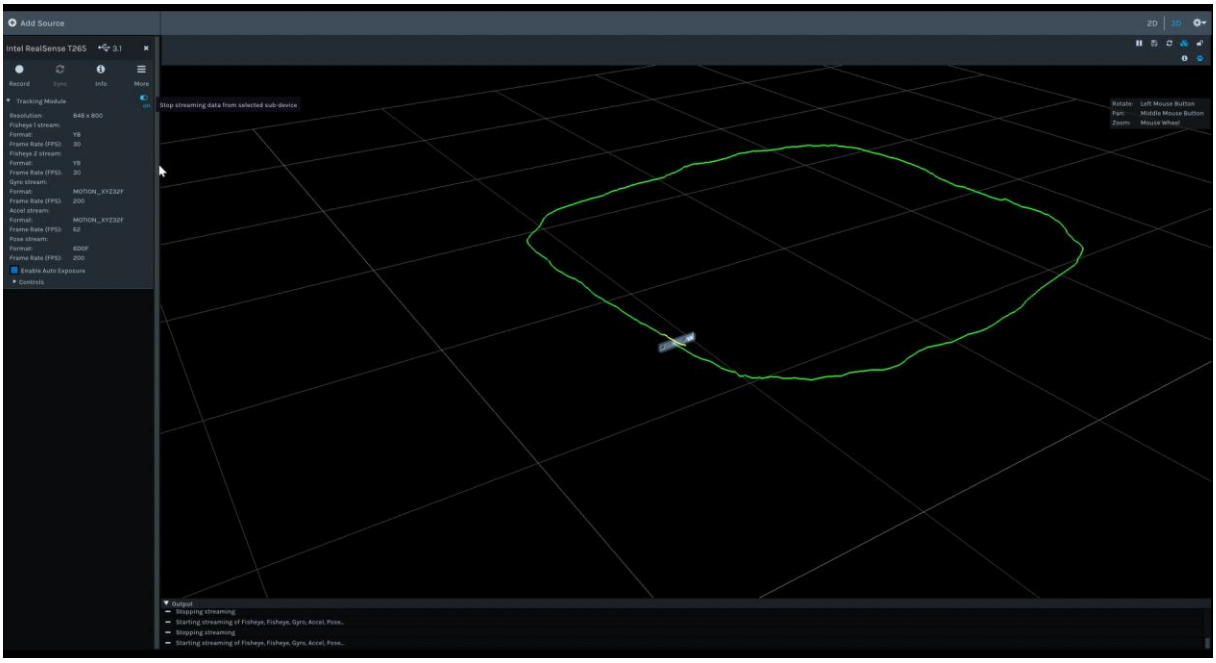


Figure 521: Result of localization using localization sensor

Raspberry pi is set to receive values 10 times per second from the localization sensor. The following values are derived from the values transmitted from the localization sensor, and these values are stored in the parameter module of the robot.

- Position [Unit: mm]
- Heading angle [Unit: degree]
- Speed [Unit: m/s]

In order to verify the performance of the localization sensor, the robot was driven around the map. As a result of the performance, it was confirmed that the robot was within 3 cm of the error range.

2.3.3 Local Network Environment

As shown in the figure below, the network for the test bed composed of cranes and robots is composed mainly of wired/wireless routers. The network for controlling the moving robot uses WiFi, and each robot has a fixed wireless IP setting. The network for controlling small cranes uses a wired LAN, and a fixed IP is also set.

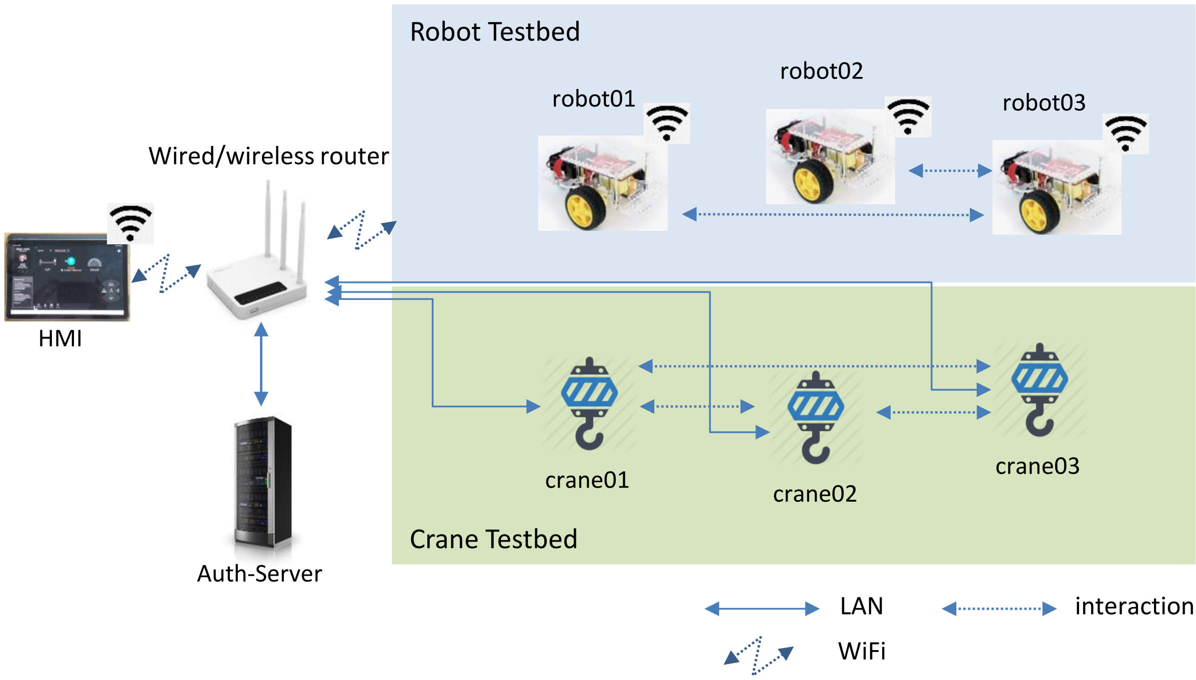


Figure 532: Robot/Crane Testbed Network Environment

2.3.4 Enhanced HMI

For better authentication and control, two types of tablets were used for the HMI app for the operator. With NFC-enabled tablets and non-supported tablets, authentication processing through NFC tagging and ID/Password authentication are supported.

Type	Model	Specification/Function
Non-NFC HMI	Samsung galaxy tab, sm-t830	• Spec - OS: Android 10 - Display: 10.6 inch(2660x1600) - Network: WiFi, BT, • Function - ID/PW authentication - Crane/Robot Control - Crane/Robot location monitoring - Crane/Robot status monitoring
NFC Enabled HMI	Samsung galaxy tab, Active2	• Spec - OS: Android 9 - Display: 9 inch (1280 x 800 (WXGA)) - Network: WiFi, BT, NFC • Function - NFC enabled authentication - Crane/Robot Control - Crane/Robot location monitoring - Crane/Robot status monitoring

Table 6: Types of HMI-devices

2.4 SW Architecture

2.4.1 Overview

The software is installed for each robotic device and is classified into IIoT components and DCP components. Distributed robots are connected through IIoT components and interact with each other to transmit and receive data in both directions. A DCP component is placed in the lower layer of the IIoT component. The DCP component consists of a controller that collects data generated by the robot and an MQTT client that packages the collected data and transmits it to the IIoT component. Figure 53 shows the composition of the entire software. The core components of the software consist of OPC UA and MQTT protocols and functions to operate them.

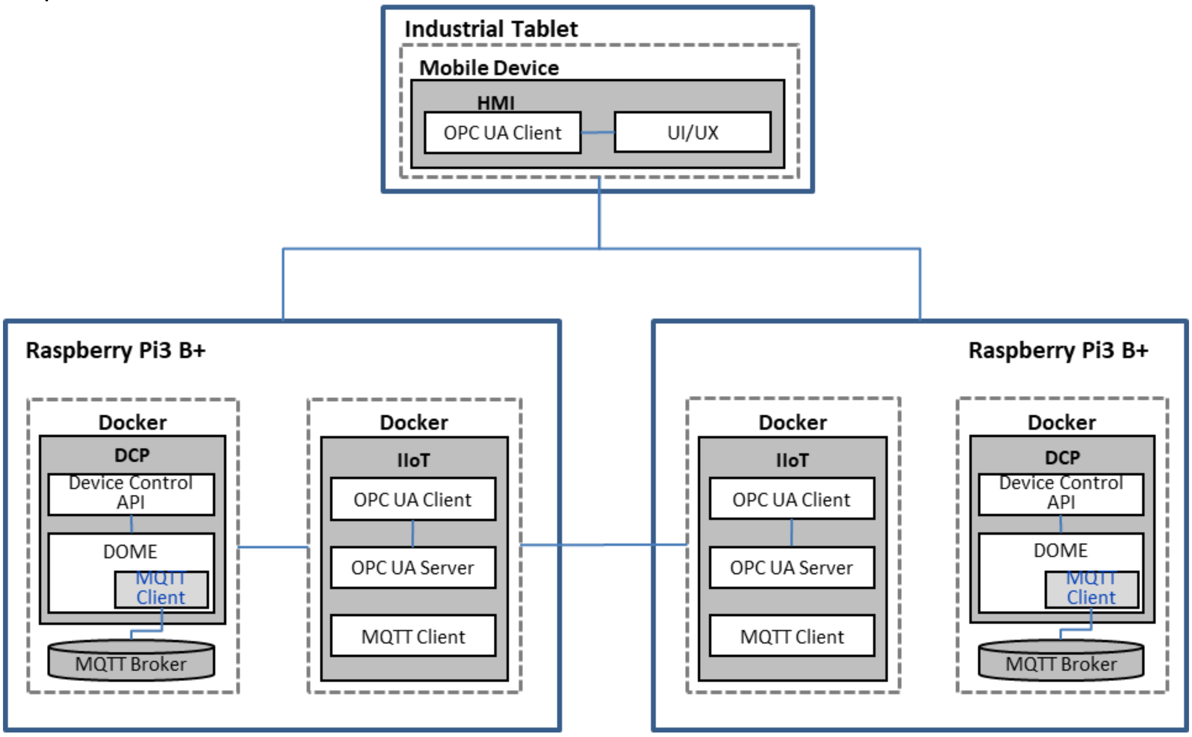


Figure 543: Conceptual software architecture

2.4.2 IIoT SW

IIoT platform consists of OPC UA Server and Client tool. The IIoT software was implemented based on the library provided by Open62541. The OPC UA server is equipped with an information model built to collect robot and crane data. Figure 54 shows the designed information model.

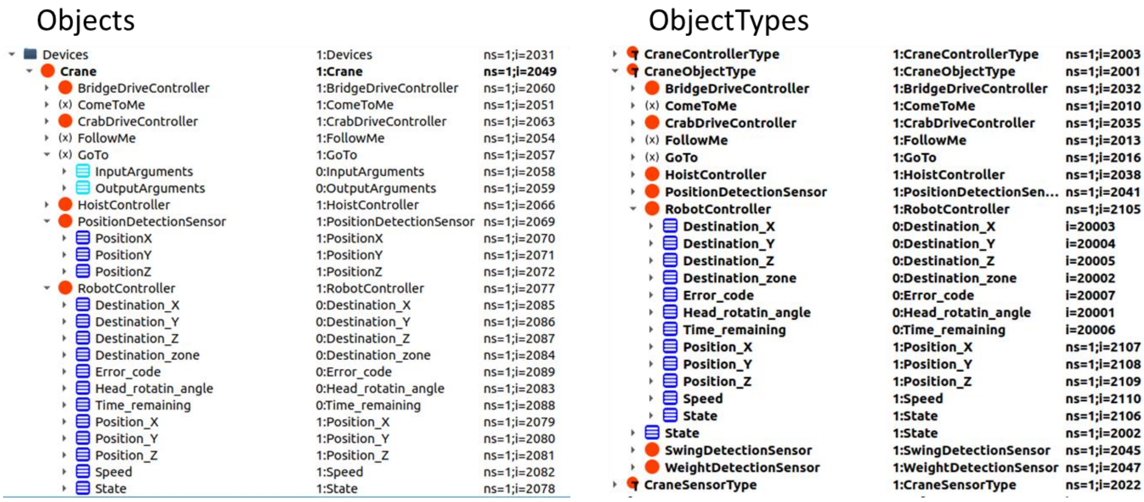


Figure 554: OPC UA information model

Figure shows a view of the designed information model on the OPC UA server. A user can connect to the designated OPC UA server using the OPC UA client and monitor the object tree configured through the defined information model.

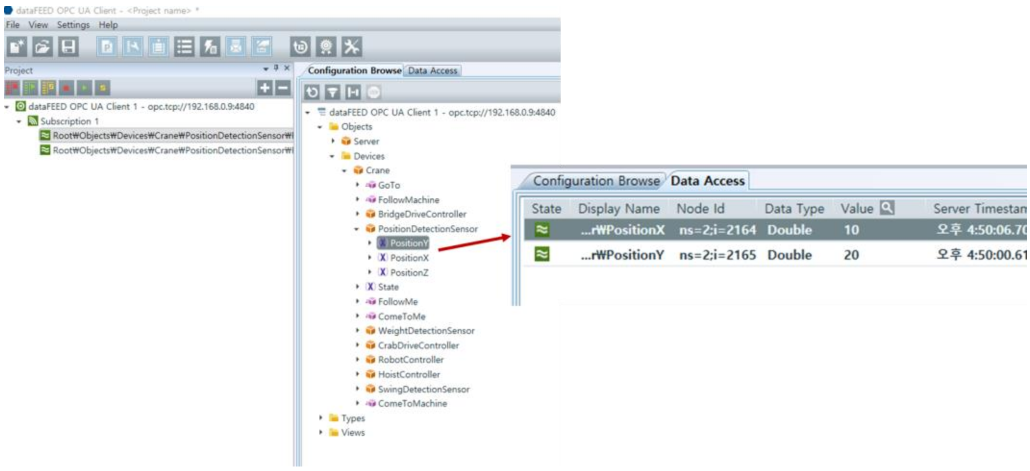


Figure 565: Client view of information model on OPC UA server

The IIoT software runs through the process of running the OPC UA server, importing the information model, and initializing the server to a state of waiting for client connection.

Figure 56 shows the view where the execution of IIoT software is completed by executing OPC UA server and initializing MQTT client for data exchange with DCP.

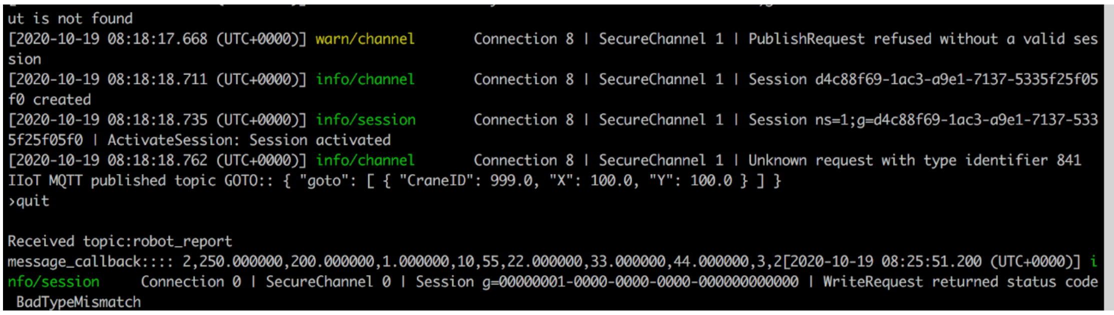


Figure 576: Display of running OPC UA server

2.4.3 DCP SW

The DCP software consists of a controller for robot data processing and an MQTT client that transmits the collected data from the local device to the IIoT and receives the commands requested from the HMI. The figure 57 shows a screen for exchanging messages between IIoT and DCP.

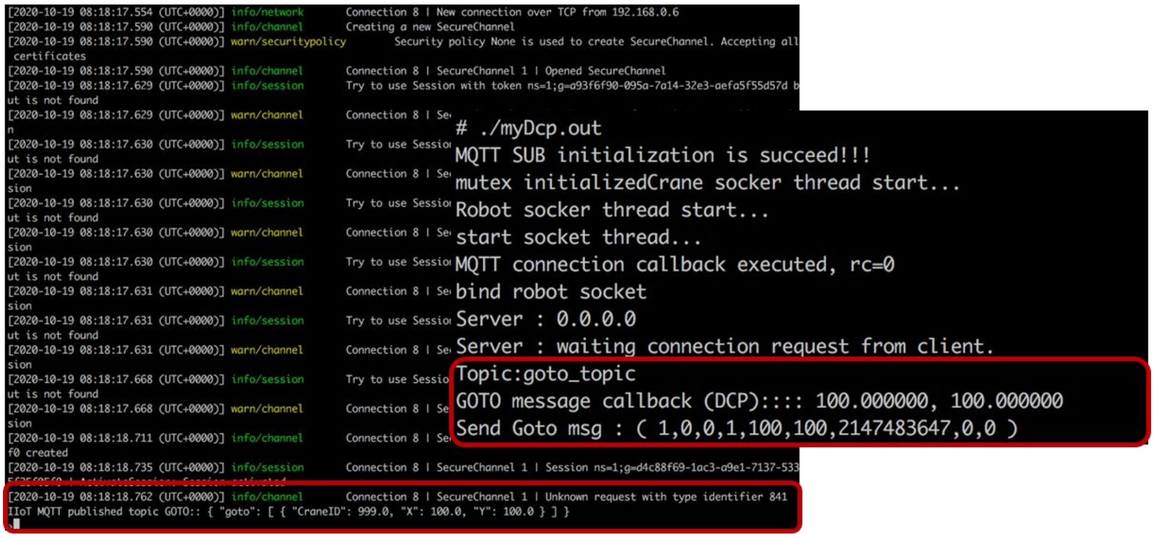


Figure 58759: Message exchange between IIoT and DCP

In order to execute a given command through the HMI, the robot should save variables for situation awareness and task execution. Variables are stored in the parameter module, and the stored information is as follows and these parameters are provided to DCP SW through TCP/IP.

- Robot state
- Request function (=task)
- Position (self)
- Heading angle (self)
- Speed (self)
- Destination coordinate
- Heading angle required when entering target coordinate
- Position (obstacle, other robot)
- Heading angle (obstacle, other robot)
- Speed (obstacle, other robot)

Robot generates traveling path every 100 msec when a position is updated or a new destination value is received. After generating traveling path, the robot performs the following algorithm to determine the wheel speed of each left and right, respectively.

- R_{go} and θ_{diff} are calculated by connecting the current position of the robot and the target coordinates with a straight line.
- Common wheel rotation speed on the left and right wheels is determined using the distance that the Robot 2 should move in the straight direction.

Wheel rotations speed of each of the left and right wheel is calculated by reflecting the difference between the current heading angle of robot and the heading angle to be prepared in the next step.

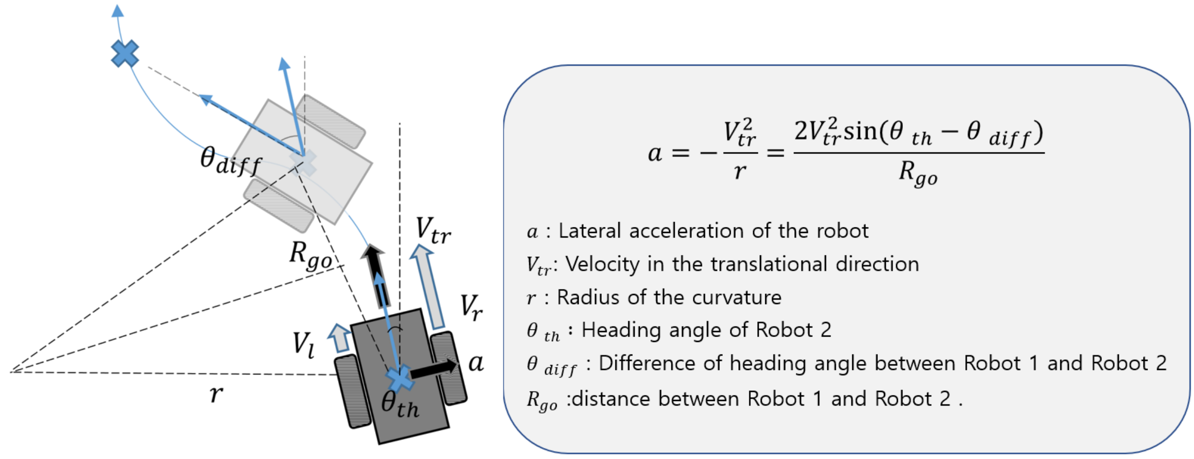


Figure 608: Robot control algorithm

Robot creates the shortest route to the destination point when a destination is determined. If an obstacle area is included in the shortest route, robot created a collision-free path. To create a collision-free path, information on the size of the robot, the safety distance, and the size of the obstacle area are required. It moves by creating a route that passes through the curved circumference of the obstacle area with the safety distance. Path is created periodically. Therefore, even if new obstacles are found during the movement of the changed path, it creates a collision-free path again in real time and moves to the destination.

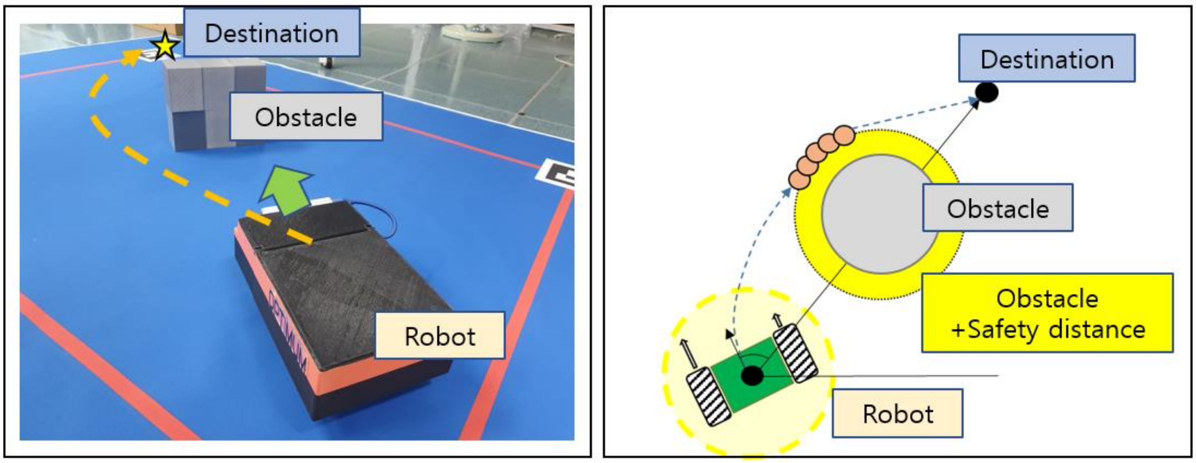


Figure 619:62 Obstacle avoidance

2.4.4 Certificate Authority

Authentication through NFC was implemented to have the following processing flow between HMI and authentication server.

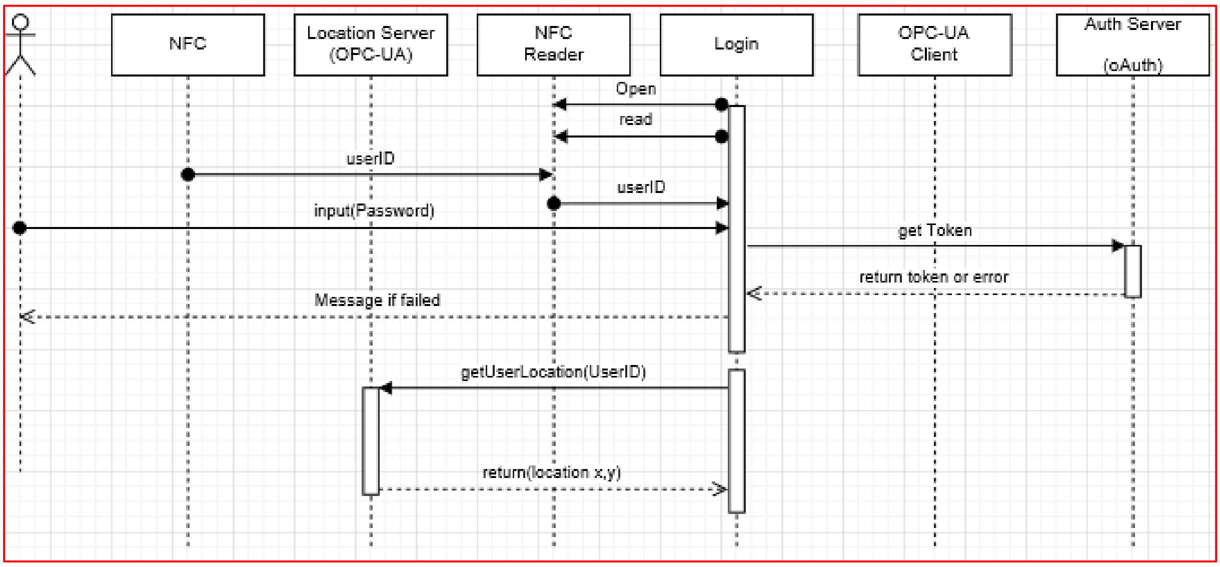


Figure 63: Authentication process flow through HMI app

When the operator tags the employee ID with NFC built-in to the HMI terminal, the terminal reads the user's unique ID and receives the user password. Through ID and password, the authentication server issues an authentication token for the user through OAuth service authentication. If an unregistered user, a login failure message is displayed.

The authentication server is based on Spring Boot and includes OAuth authentication processing function. Authentication service is provided by RESTful API. The following image is the result of the authentication token issuance through RESTful API, and provides the token type, authentication token and renewal token, and valid time.

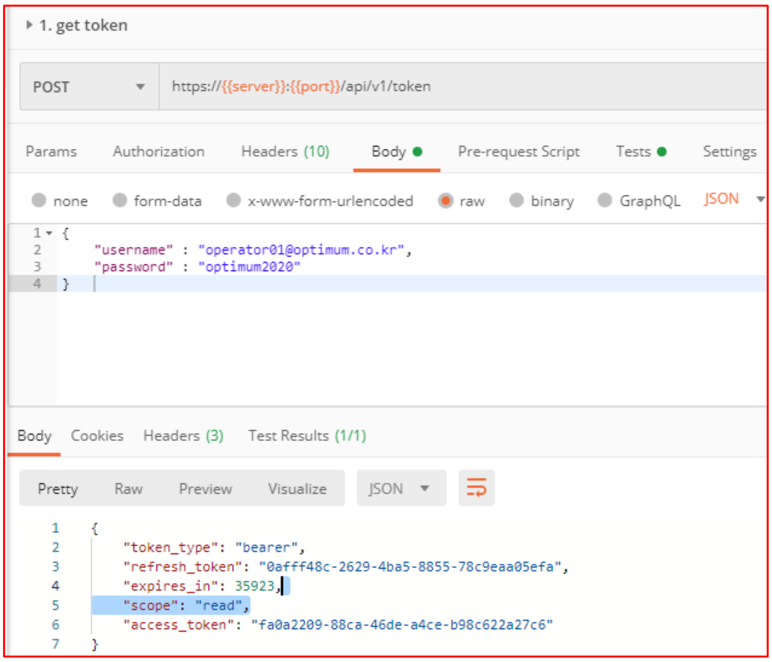


Figure 6465: OAuth-based authentication processing implementation

The Auth server is fixed to the IP of “192.168.0.100” in the system. The Auth server requires java (JDK or JRE 1.8), and executes it by entering an execution script or as follows.

```
C:\OptimumAuth\java -jar OptimumAuthServiceAPI-1.0.0_SNAPSHOT.jar
```

2.4.5 Webservice

To monitor real-time execution logs of robots and cranes, MQTT Log UI was developed and used for verification purposes. It was developed as a web UI based on nodejs to accurately identify logs or processing flows that may occur in the execution process of various use cases of IIoT and DCP. The UI is distributed in gitlab, etc., so it can be downloaded and install it in the required environment. In order to install MQTT UI, nodejs and npm should be installed according to the operating system. The following is an example of such an installation operation. Download it from the gitlab server, install it, and run it as follows.

```
git clone https://gitlab.com/aiblab/govproject/optimum/machine-log-ui.gi
t
cd machine-log-ui
npm install
npm start
```

Figure 66: MQTT Log UI run

After executing the module, access the following address through a web browser. http://localhost:8080

The monitoring UI must operate within the same network area that can be connected to the authentication server and the robot/crane, and by entering the address of the authentication server (https://localhost:9090), the target robot/crane log is obtained.

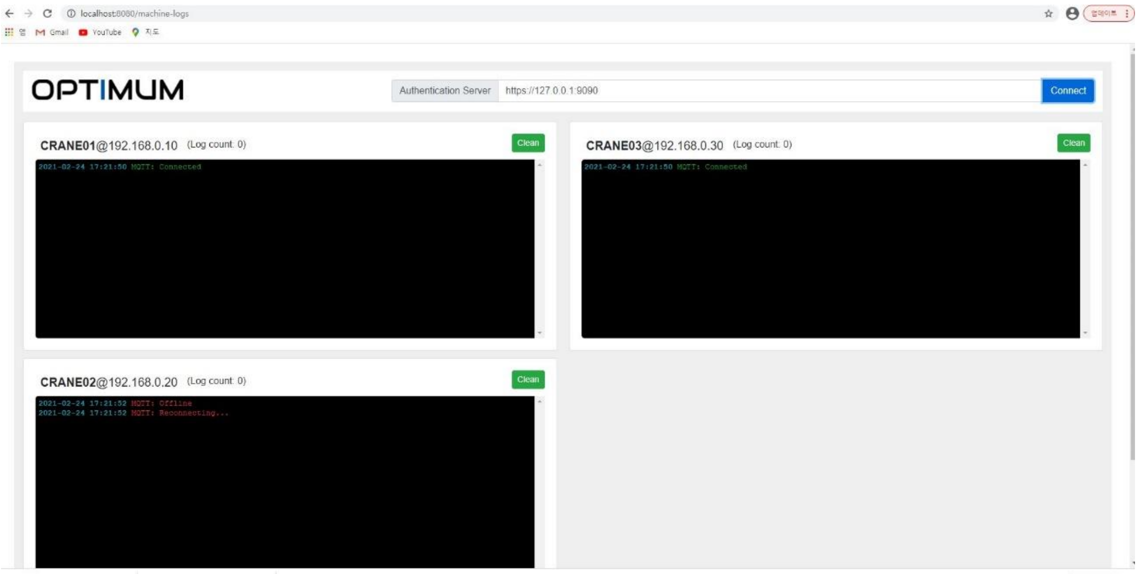


Figure 67368: MQTT Log UI

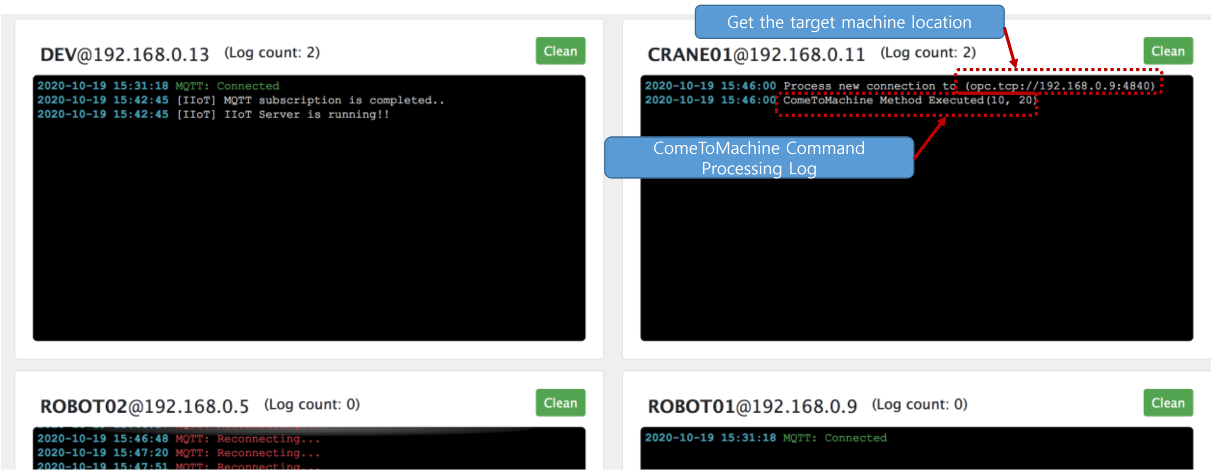


Figure 69470: Log for ComeToMachine Command

2.4.6 App Store

Cranes and robots must be operated by deploying IIoT, DCP and MQTT broker modules on Raspberry Pi III B+. Therefore, each development module IIoT, DCP is built as a Docker image and distributed to Docker Hub, and has been developed so that it can be installed and operated on each target machine.

The structure of development/distribution/installation and operation centring on the docker hub is as follows.

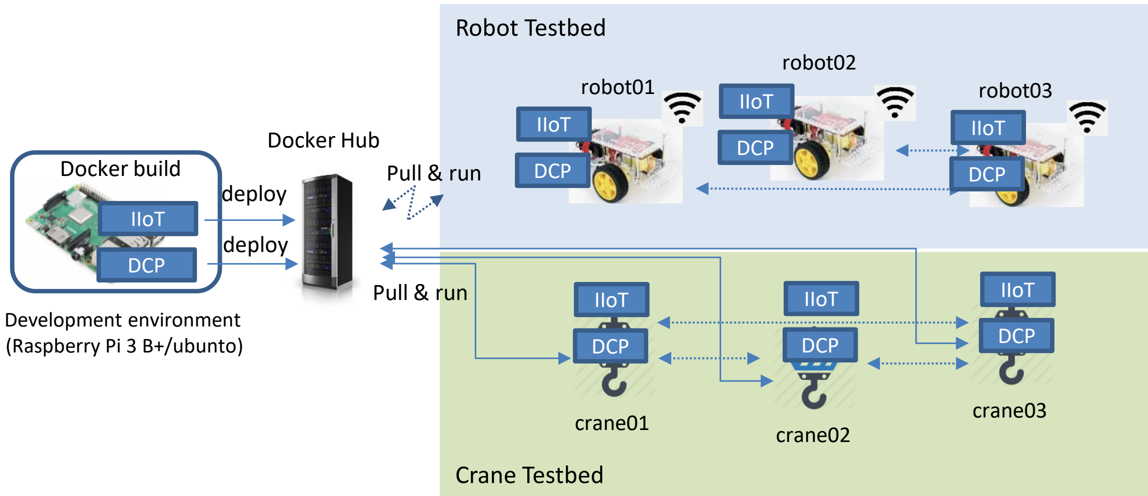


Figure 71: structure of docker hub

Each machine composes a docker-compose environment and can install and run docker images such as IIoT and DCP through the configuration file below.

- docker-compose.yml

```
version: '3'

services:

 iiot:
  image: optimumkr/iiot-pi:latest
  container_name: optimum-iiot
  hostname: optimum-iiot
  networks:
   optimum-network:
    aliases:
      - optimum-iiot
  ports:
   - "4840:4840"
  command: "/server/run.sh"
```

```
    depends_on:
      - mqtt

  mqtt:
    image: eclipse-mosquitto:latest
    hostname: optimum-mqtt
    container_name: optimum-mqtt
    restart: always
    ports:
      - "1883:1883"
    networks:
      optimum-network :
        aliases:
            - optimum-mqtt

  dcp:
    image: optimumkr/dcp-pi:latest
    container_name: optimum-dcp
    hostname: optimum-dcp
    networks:
      optimum-network:
        aliases:
            - optimum-dcp
    ports:
            - "20000:20000/udp"
            - "30000:30000"
    command: "/server/run.sh"
    depends_on:
      - mqtt

networks:
  optimum-network:
    driver: "bridge"
```

2.5 Process flow in the demonstrator line

Regarding HMI-based robot/crane control, six requirements were defined as follows, and HMI functions were developed.

- The HMI equipment must form a network to control registered equipment.
- The HMI device must be able to operate the machine by receiving commands from the user through the GUI.
- The HMI device must be able to operate the machine by receiving commands from the user through the GUI.
- The HMI device must be able to automatically move the machine from an arbitrary location to a specified destination. (Go to X function)
- The HMI device must be able to move the selected machine to any other device location. (Come to Machine function)
- The HMI device must be able to move the machine B in the direction in which the machine A moves when the operator controls the machine A. (Follow Machine function)
- The HMI device must be able to move the machine B in the direction in which the machine A moves when the operator controls the device A. (Follow Machine function)

The flow of processing GoTo, ComeToMachie, and FollowMachine commands through the HMI device is as follows.

■ Procedure for GoTo command

Test Case	GoTo X command control function of the HMI device
Test Procedure	**[GoTo command configuration: Operator -> HMI]** 1. The operator selects the target machine and clicks the GoTo command button. 2. Set the destination location to be delivered by GoTo command and click the Execute button. **[Transfer of GoTo command: HMI -> Machine]** 3. MQTT Log UI is connected to the log of the target machine. 4. HMI transmits the request command to the IIoT module of the machine. 5. The IIoT module receives the command and sends it to the DCP.

■ Procedure for ComeToMachine command

Test Case	ComeToMachine command control function of HMI device
Test Procedure	**[ComeToMachine command configuration: Operator-> HMI]** 1. The operator selects the target machine and clicks the ComeToMachine command button. 2. Set the destination machine to be delivered with the ComeToMachine command and click the Run button. **[ComeToMachine command delivery: HMI -> Machine]** 3. MQTT Log UI is connected to the log of the target machine.

	4. HMI transmits the request command to the IIoT module of the machine. 5. The IIoT module receives the command and sends it to the DCP.

- Procedure for FollowMachine command

Test Case	ComeToMachine command control function of HMI device
Test Procedure	**[FollowMachine command configuration: Operator -> HMI]** 1. The operator selects the target machine and clicks the FollowMachine command button. 2. Set the destination machine to be delivered with the FollowMachine command and click the Execute button. **[Transfer FollowMachine command: HMI -> Machine]** 3. MQTT Log UI is connected to the log of the target machine. 4. HMI transmits the request command to the IIoT module of the machine. 5. The IIoT module receives the command and sends it to the DCP. **[FollowMachine command configuration: Operator -> HMI]** 6. The operator selects the target emergency stop button. **[Transfer FollowMachine command: HMI -> Machine]** 7. The IIoT module receives the command and sends it to the DCP.

2.6 Testing and verification

2.6.1 Functionalities of Robot

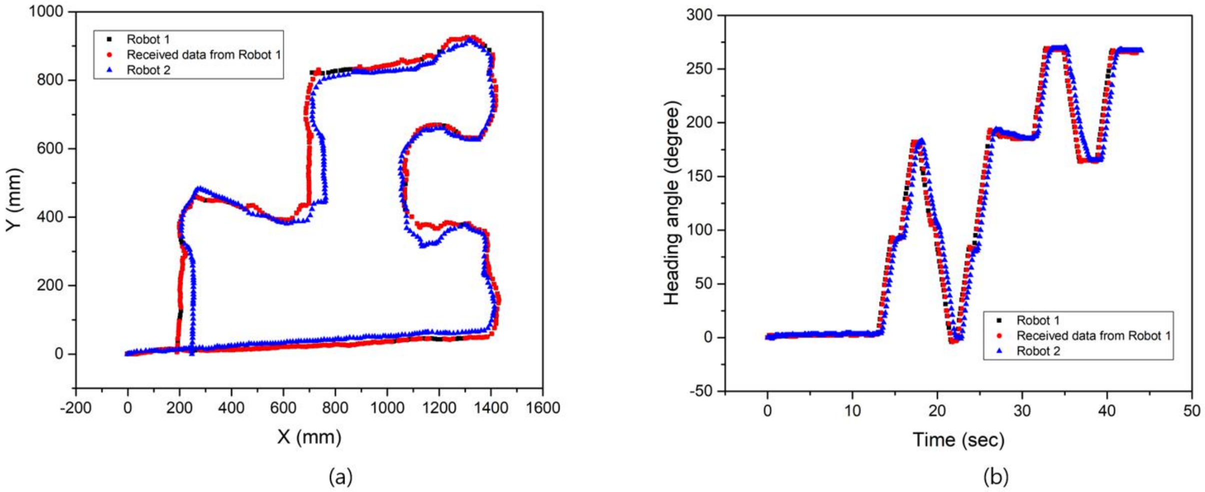


Figure 72: Traveling path between Robot 1 and Robot 2. (a) Position. (b) Heading angle.

To verify the function scenario and response time between robots, Robot 1 was manually controlled by the operator, and Robot 2 was experimented with "Follow machine". Robot 1 provided its own location information to Robot 2 every 100 msec, and Robot 2 continued to move using the location data received from Robot 1 as a destination. As shown in Figure 65, Robot 2 moves in a similar trajectory as Robot 1. The error about the position can be seen as an error due to the wheel control performance of the robot. It was confirmed that the heading angle between the two robots was similarly deformed. When Robot 1 transmits location information, it takes an average of 20 msec for Robot2 to receive it. Moreover, the time it takes for robot 2 to control the motor after receiving the location information was up to 80 msec.

2.6.2 Functionalities of HMI

Tests/validations were performed for the following six functions related to these requirements.

- Network connection setting function between HMI device and control target machine
- GoTo X command control function of the HMI device
- ComeToMachine command control function of HMI device
- FollowMachine command control function of the HMI device
- Machine monitoring function of the HMI device

Test Case	Network connection function based on HMI connection setting information
Test Criteria	If the registration machine list inquiry information is printed on the HMI and the connection is performed and the result is confirmed, it is successful.

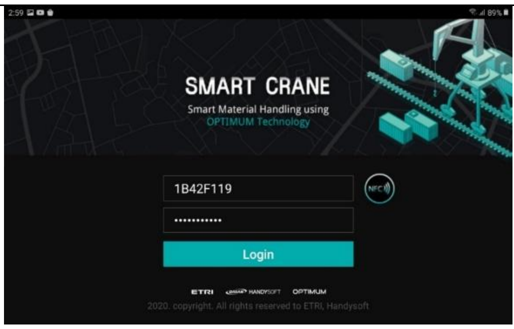
<table>
<tr><td>Test Result</td><td>


Figure 7374: HMI Login

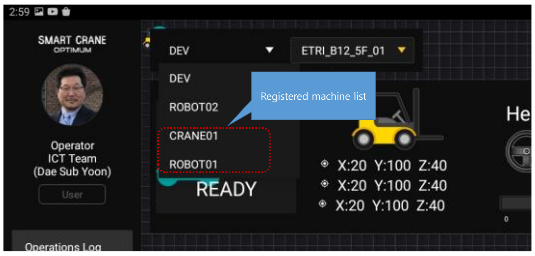


Figure 7576: Machine List Display

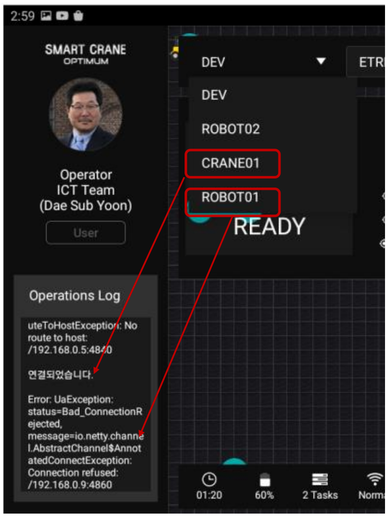


Figure 7778: Connection Results
</td></tr>
</table>

Test Case	GoTo X command control function of the HMI device
Test Criteria	The IIoT MQTT log is output as a result of receiving the command for the destination coordinates.

Test Result

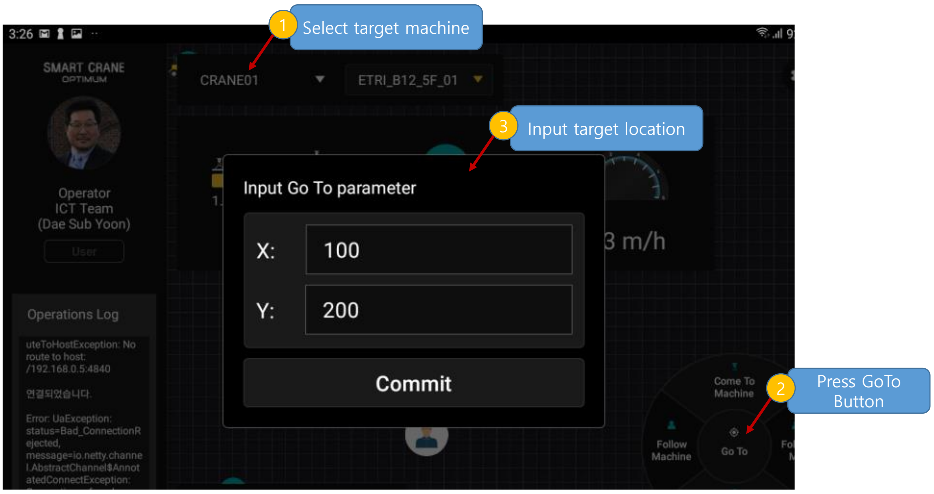


Figure 7980: GoTo command test

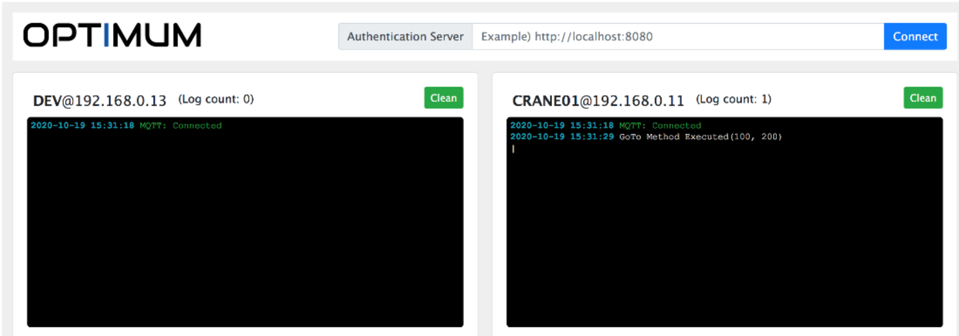
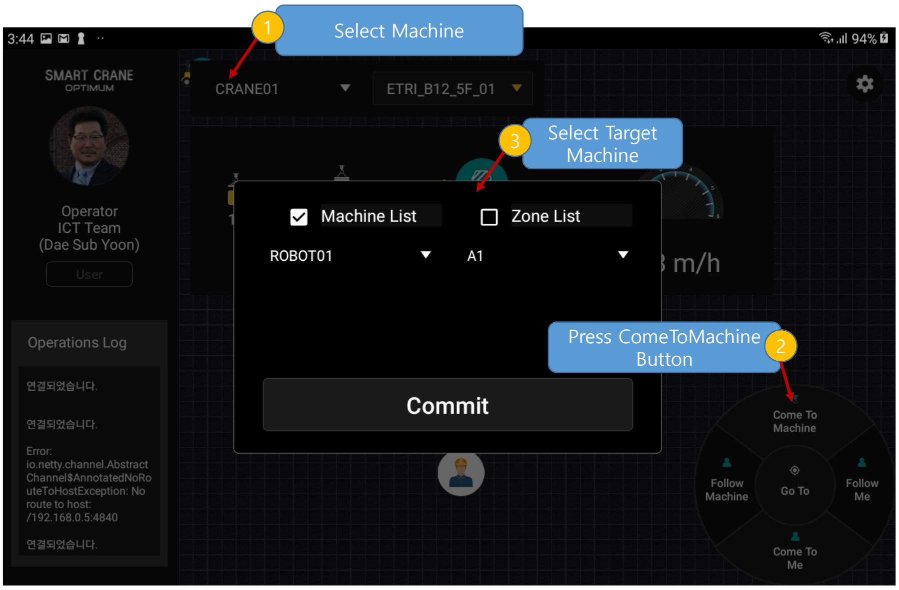
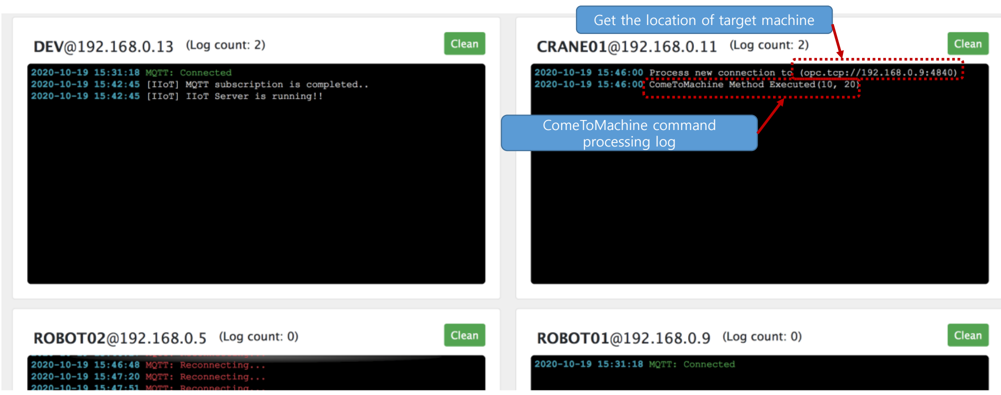


Figure 81: MQTT log for GoTo command

Test Case	CometoMachine command control function of HMI device
Test Criteria	As a result of receiving the command for the destination machine, the location of the destination machine is inquired and the IIoT MQTT log is output.
Test Result	The following is a screen to select CRANE01 machine through HMI and configure CometoMachine command.  **Figure 822: ComeToMachine command test**  **Figure 83384: MQTT log for ComeToMachine command**

<table>
<tr><td>Test Case</td><td>FollowMachine command control function of the HMI device</td></tr>
<tr><td>Test Criteria</td><td>After receiving the command for the destination machine, the location of the destination machine is inquired and the IIoT MQTT log is output while moving the machine.</td></tr>
<tr><td>Test Result</td><td>The following is a screen to select CRANE01 machine through HMI and configure FollowMachines command.

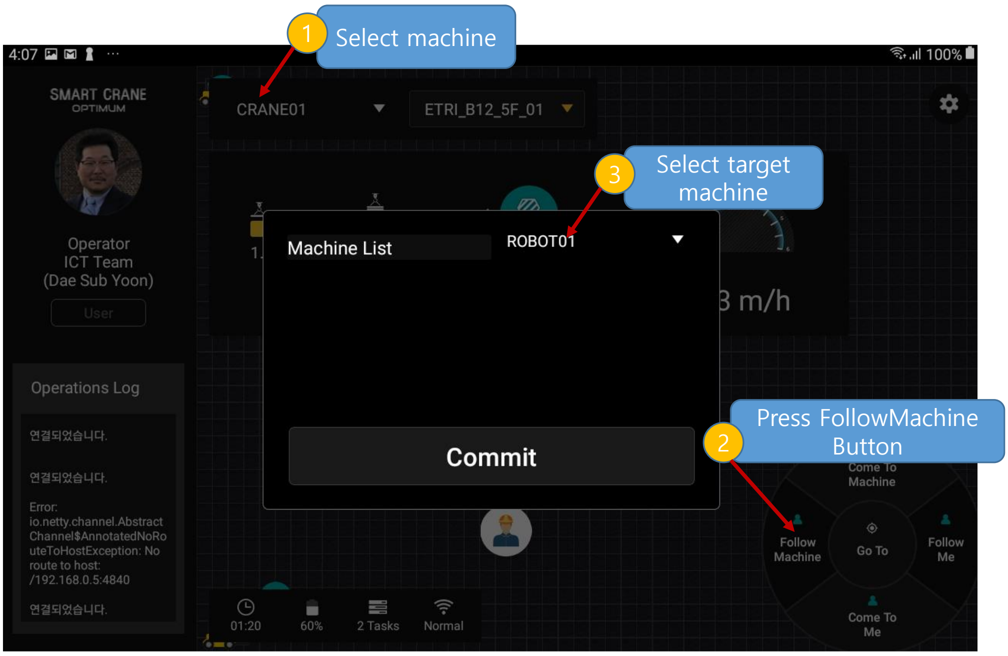


Figure 85486: FollowMachine command test

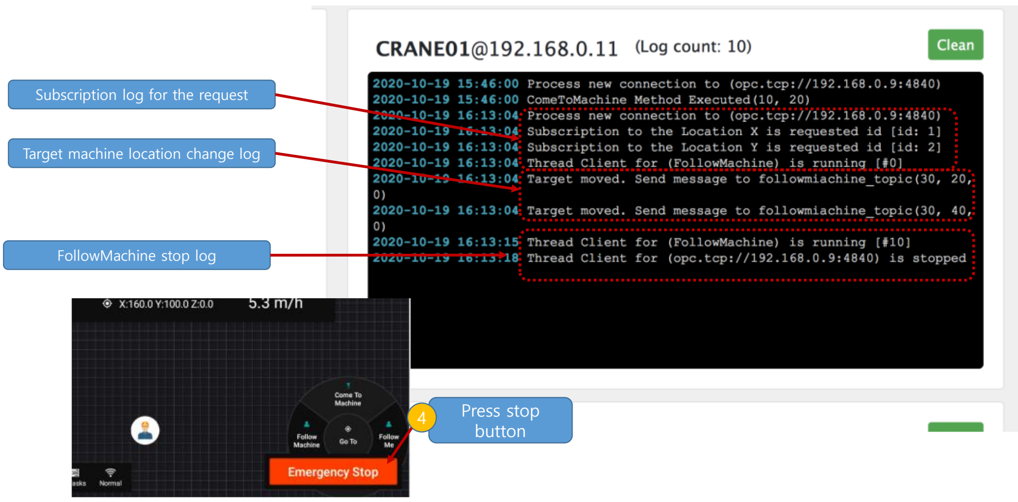


Figure 875: MQTT log for FollowMachine command</td></tr>
</table>

3 Abbreviations

MH	Material Handling
OPC UA	Open Platform Communications United Architecture
MQTT	Message Queuing Telemetry Transport
DCP	Decentralized control platform
I/O	Input / Output
IIoT	Industrial Internet of Things
SLAM	Simultaneous Localization and Mapping
VPU	Visual Processing Unit
IMU	Inertial Measurement Unit
DRC	Demag Radio Control
GUI	Graphical User interface
HMI	Human Machine Interface

4 References

[1] LINDE Homepage:
Plattform-Hochhubwagen L14 – L20 AP von Linde Material Handling (linde-mh.de)

[2] Operating Manual Demag Smart-Tandem 2-2 (Id.no.: 21134444)

[3] Operating Instructions SafeControl (Id-no. 21180944)

[4] Operating Manual DRC-JS D3 radio control (Id-no. 21127144)

[5] NXP Homepage: i.MX Evaluation and Development Boards | NXP Semiconductors

[6] SIERRA WIRELESS Homepage: Sierra wireless AirLink RV55 Manuals | ManualsLib

[7] GUI Specification for Material Handling Use Case, OPTIMUM_PPR6_Attach_5_GUI_v1.9.docx

[8] https://www.st.com/en/microcontrollers-microprocessors/stm32l452re.html

[9] ESP32-WROOM-32U – Wireless connection [8] https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf

[10] BNO055 - Absolute Orientation Sensor [9] https://www.bosch-sensortec.com/products/smart-sensors/bno055.html#documents