Mechatronics Project
Autonomous Guided Vehicle

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Supervised by - Prof. Dr.-Ing. Jörg Wollert, Victor Chávez-Bermúdez

06. April 2022

-Aadithya Ramamurthy, Michael Knoll, Rachana Kodilla, Sameer Tuteja &
Vedhashruthi Harinath

Outline

Motivation & Starting Point

System Design

> ROS
> PLC
> Power Supply
> Motor Controller

Conclusion & Outlook

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Motivation & Starting Point
Introduction

- Great demand to connect the workstations in the Industry 4.0 model factory at the AMA.

➢ Task to design, build and test an autonomously driving shop floor delivery vehicle.

  - Carry a load of 100kg
  - Costs a maximum of 4000€
  - Being industry compliant

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

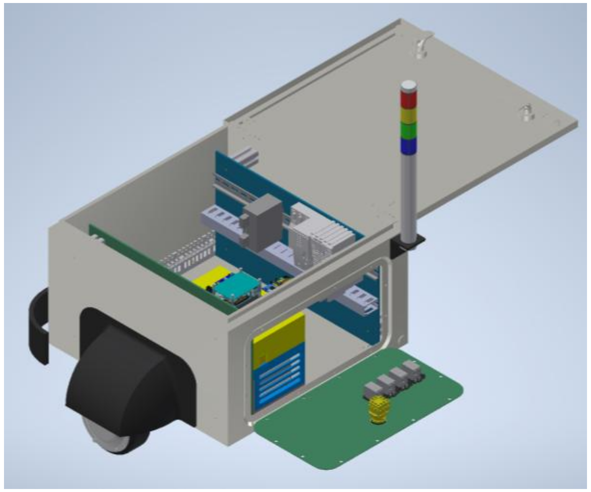
Motivation & Starting Point
Introduction

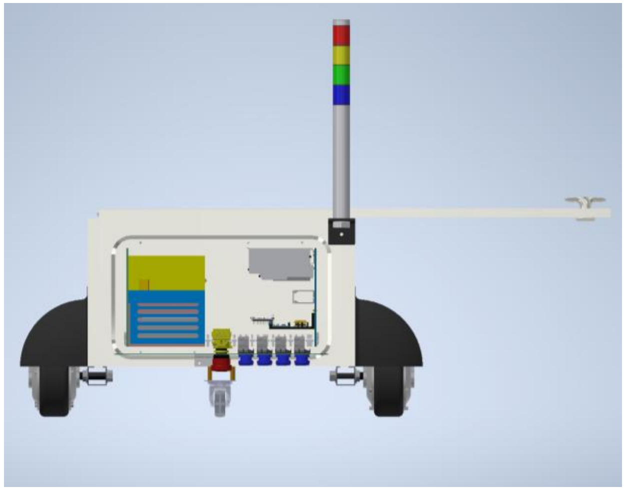
- Most Components already selected and purchased.

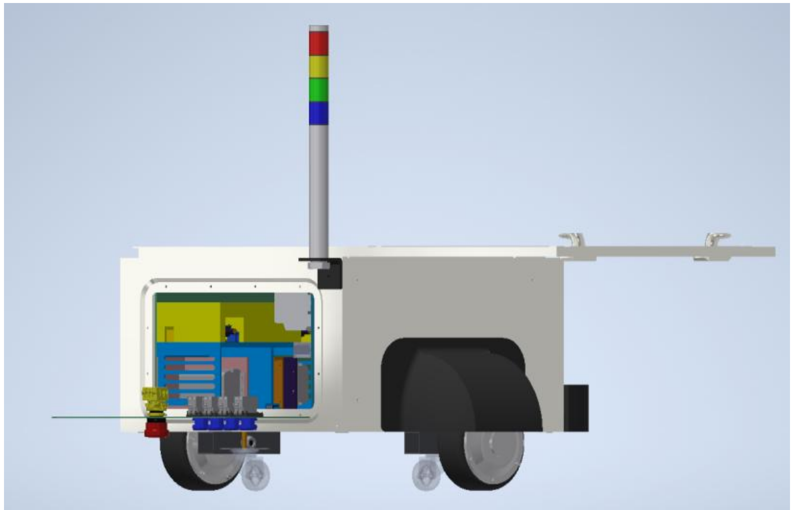
- System Design made.

➢ Requirements
- Obstacle detection and step detection
- Line following and station detection (Stage 1)
- Localisation and autonomous movement (Stage 2)
- ROS Communication

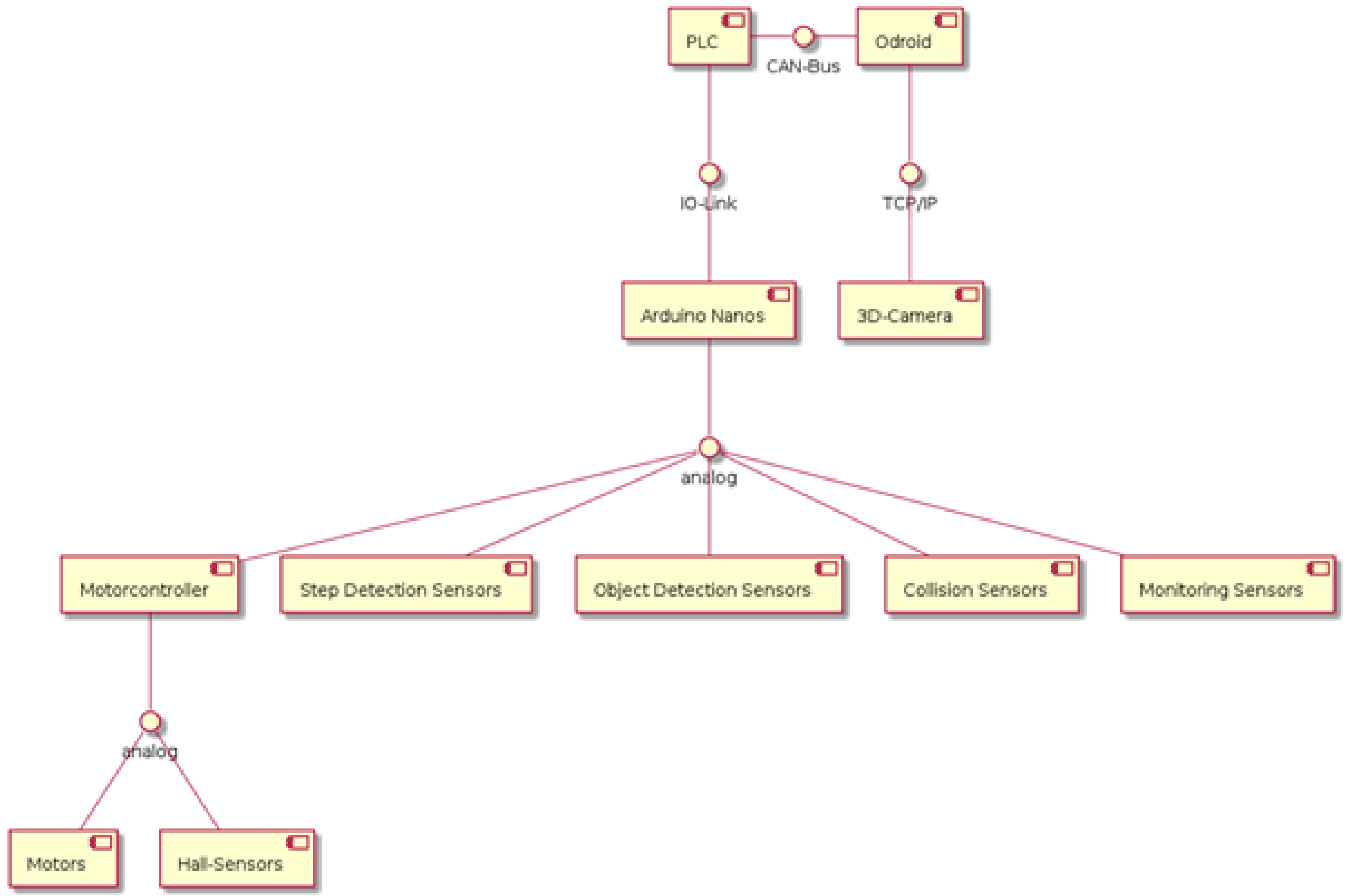
Motivation & Starting Point
3D Model



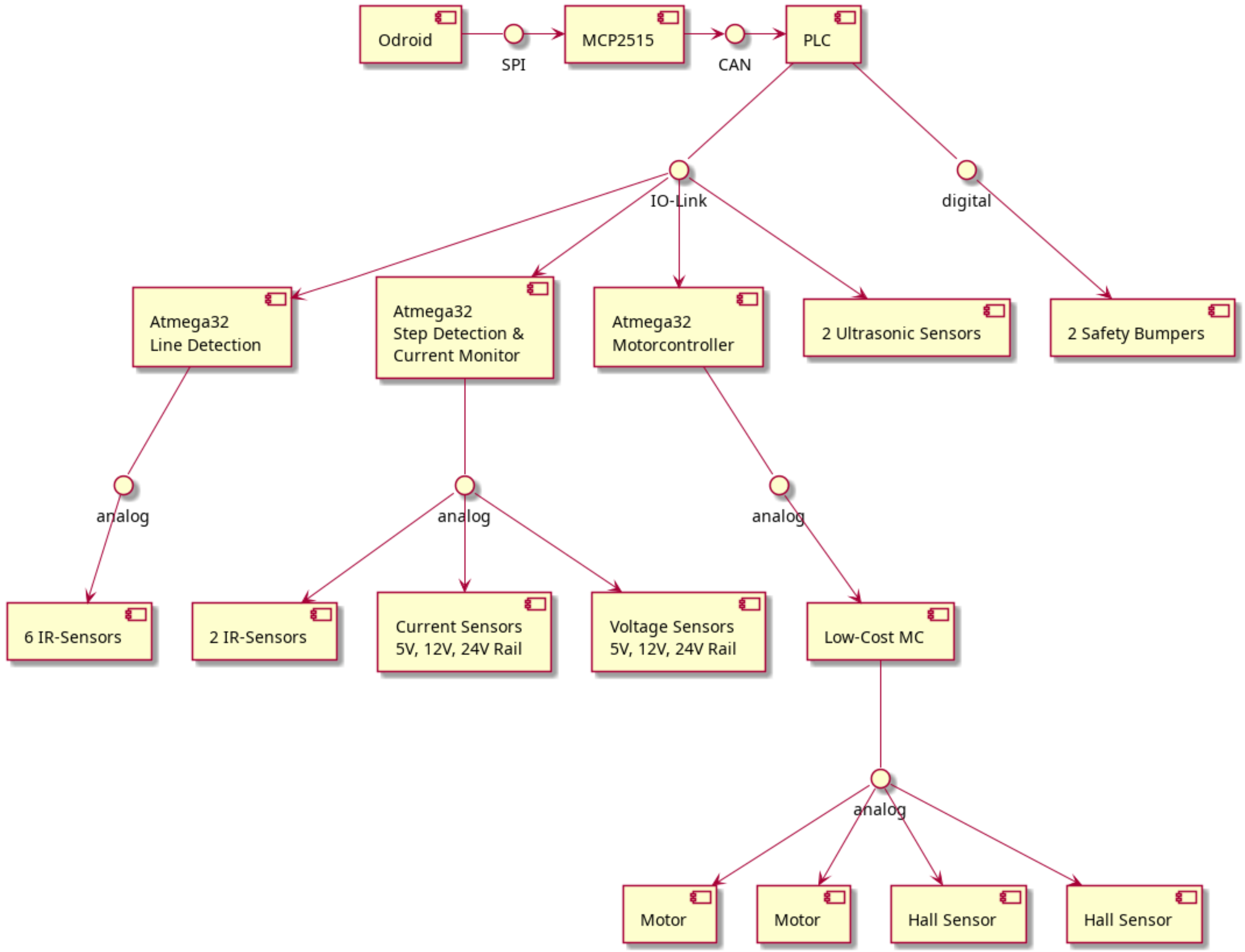




Motivation & Starting Point
Initial System Diagram

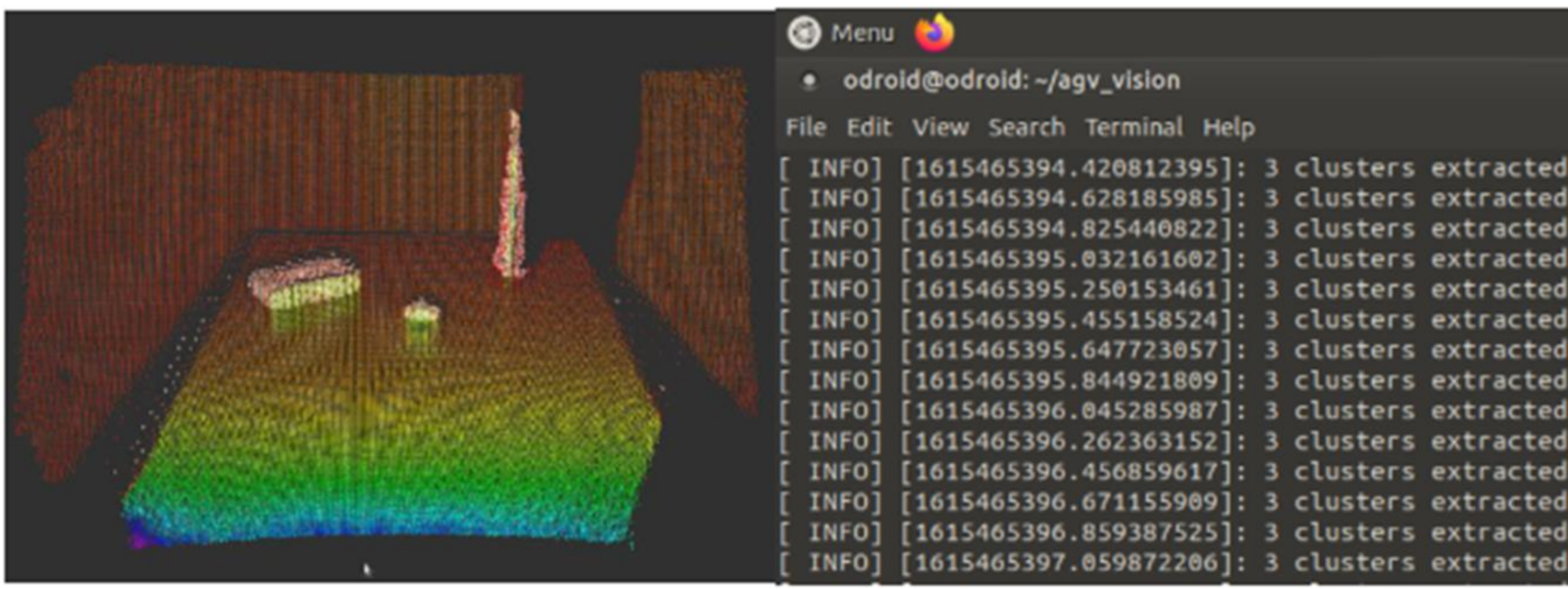


Motivation & Starting Point
System Diagram Reviewed

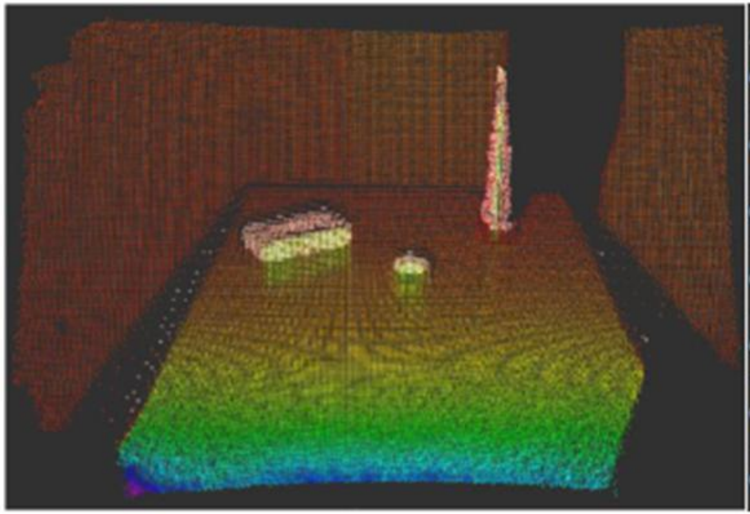


ROS
Obstacle detection

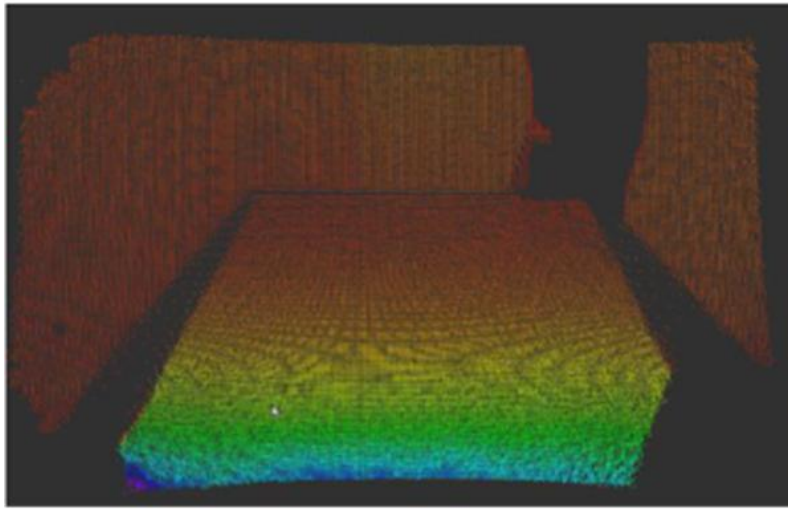
- Obstacle detection using IFM 3D Camera was tested which was done by the previous groups.
- Planar segmentation and Eucldianclustering was done on the point cloud image to detect the objects as clusters.



ROS
Obstacle detection



```
---
^Codroid@odroid:~$ rostopic echo /object
data: True
---
data: True
---
data: True
---
data: True
---
data: True
---
data: True
---
data: True
```



```
^Codroid@odroid:~$ rostopic echo /object
data: False
---
data: False
---
data: False
---
data: False
---
data: False
---
data: False
---
data: False
```

ROS
Obstacle detection

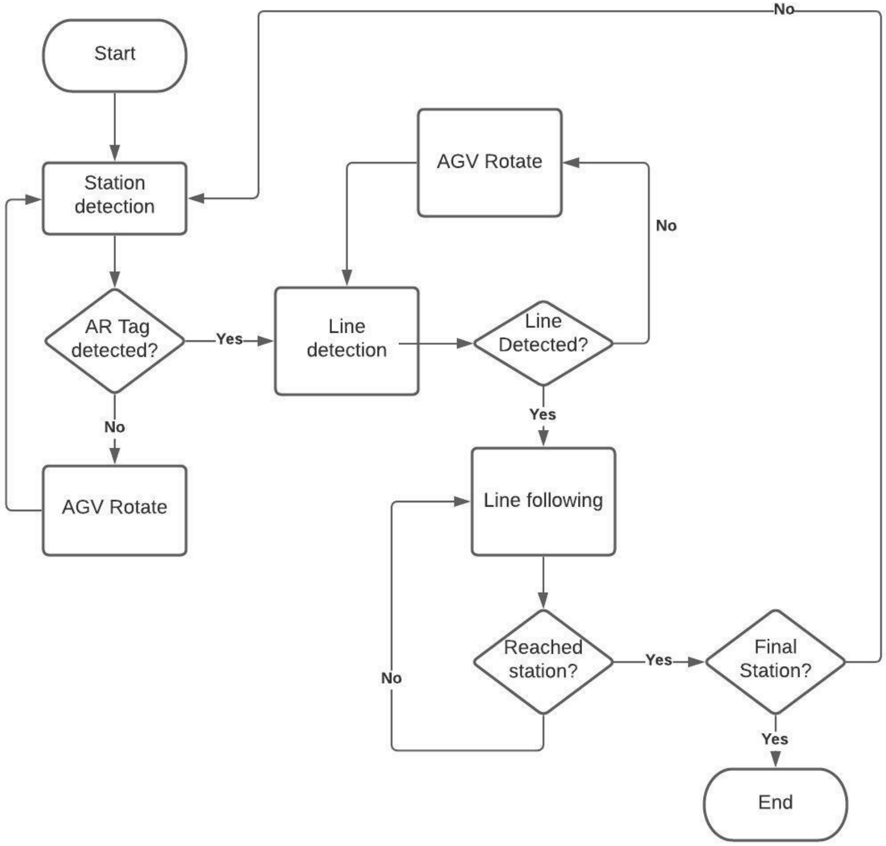
- A Boolean variable is used and set to false if number of clusters are zero and true if its greater than zero.

- This was published on a topic and a subscriber to this topic was used to send messages regarding the presence of obstacles

ROS
Station detection

AR Tags

FLOW CHART FOR STATION DETECTION USING AR TAGS AND LINE FOLLOWING OF AGV

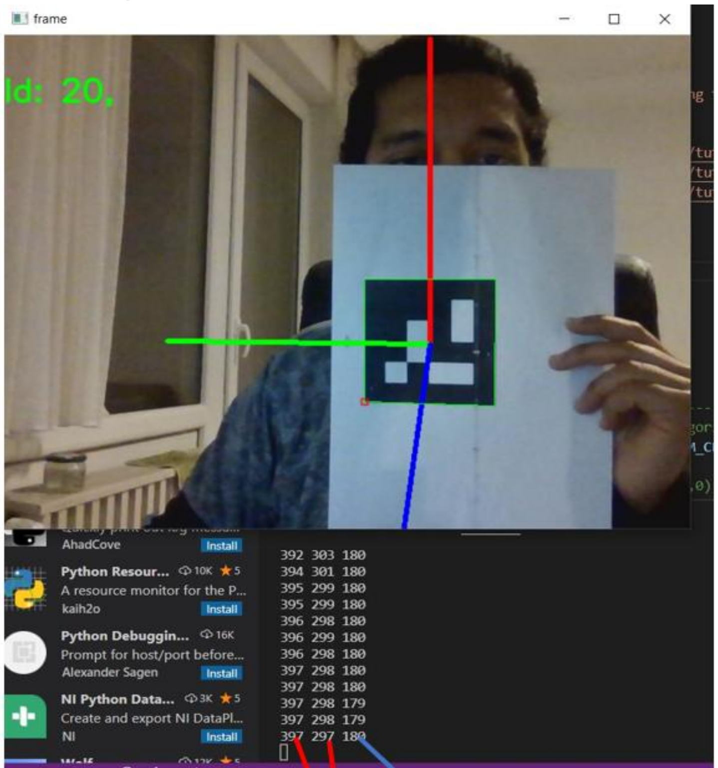


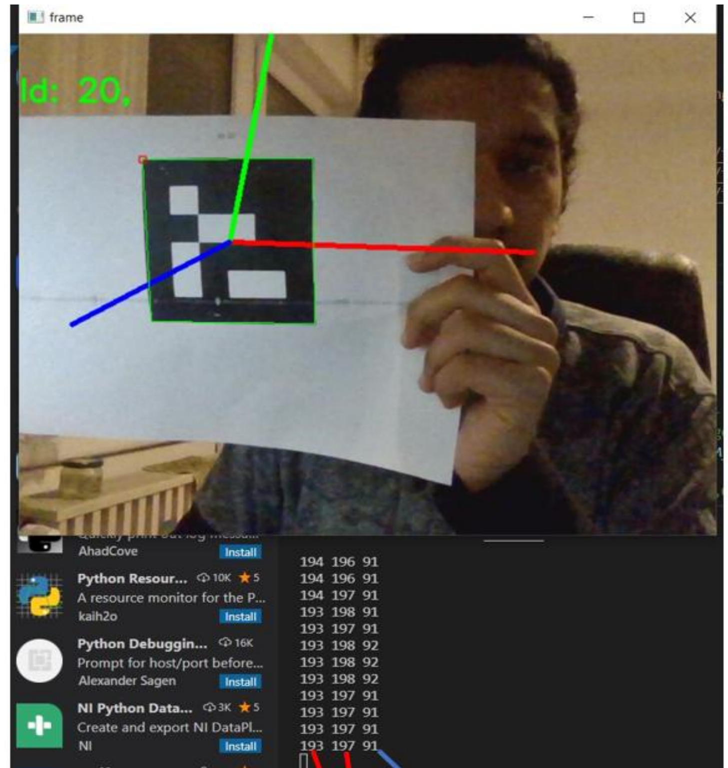
ROS
Station detection

AR Tags
- For the testing of AR Tag detection, ArUco markers were used.
- Once the code was run and executed, the marker ID was detected, the location of its center point was obtained in terms of X and Y coordinates and ots orientation was shown when the marker was placed in front of the camera.

ROS
Station detection

AR Tags





Orientation of the marker
Position of the marker

Orientation of the marker
Position of the marker

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

AR Tags – Further system development

- AR Tag detection testing was done using the laptop web camera. It must be tested on a USB camera and must be implemented on ROS on Odroid.

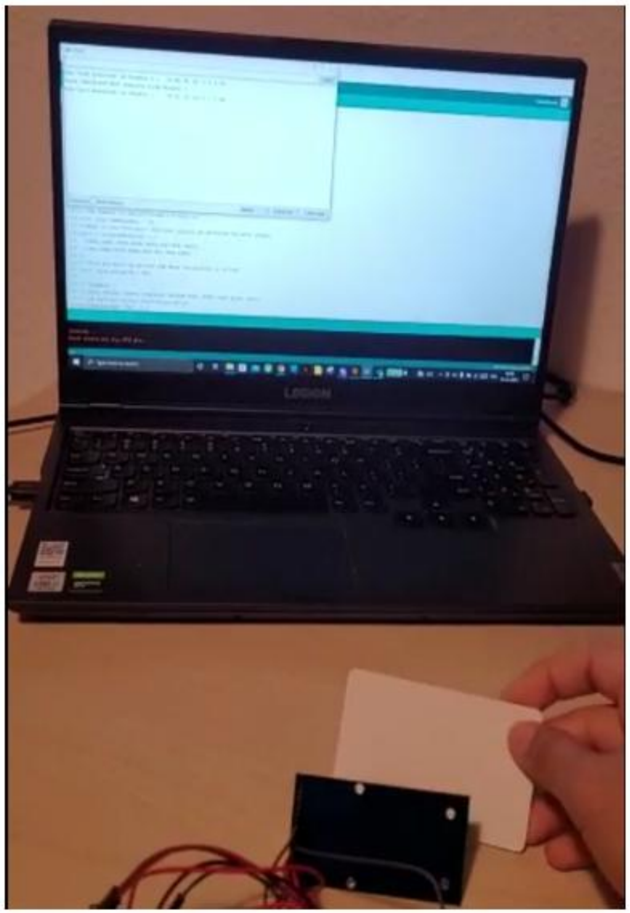
ROS
Station detection and localisation

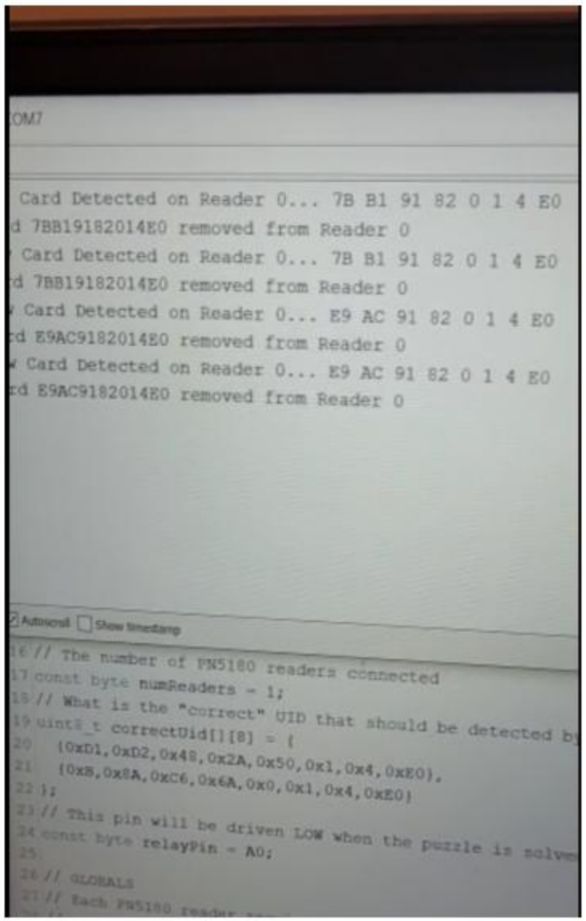
RFID Tags

- An alternate way to detect stations with line following, can also be as a redundant system with AR Tags. Further, it can be used for localisation of the bot.
- RFID Tag detection was done which were detected by PN5180 -NFC Reader.
- A NodeMCU ESP32 was used to test the PN5180 reader

ROS
Station detection and localisation

RFID Tags





ROS
Station detection and localisation

- The detection of RFID tags using the NFC Reader on nodeMCU ESP32 worked on SPI communication interface. To make it work on droid, the SPI interface had to be enabled on the Odroid.

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

SPI on Odroid

- Device Tree Overlay was used to enable the GPIO functions.
- Enabling the communication protocols SPI, I2C, UART and PWM using device tree overlay.
- Enabling the SPI feature on the board.
- To check the details of the GPIO of the Odroid, Wiringpi was installed and used.

ROS
Station detection and localisation

SPI on Odroid

- Yellow indicates that the SPI MOSI(19) and MISO(21) pins are connected using a jump cable and the SPI communication is taking place between them and red indicates no connection.

```
root@odroid:~# ./spidev_test -D /dev/spidev* -s 1000000 -b 8
spi mode: 0
bits per word: 8
max speed: 1000000 Hz (1000 KHz)

01 02 03 04
root@odroid:~# ./spidev_test -D /dev/spidev0.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF
```

ROS
Station detection and localisation

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Further system development

- RFID Tags detection was tested using the PN5180 NFC reader using a NodeMCU ESP32. It works on SPI communication.
- SPI communication on Odroid N2+ was enabled and further the code for RFID Tag detection has to be tested on Odroid N2+ using SPI Communication
- It has to be implemented on ROS for an alternative way to detect stations and further for localisation and autonomous movement of the bot.

System Design
ROS - Communication

- Execution of speed test with
  - CAN interface
  - OPCUA server/client configuration

resulted in faster data rates of the CAN interface

> Toggle Digital Output by exchanging 8-bit variable

CAN	OPCUA
10 Hz	0.5 Hz

System Design
ROS - Communication

- Definition of AGV CAN Messages

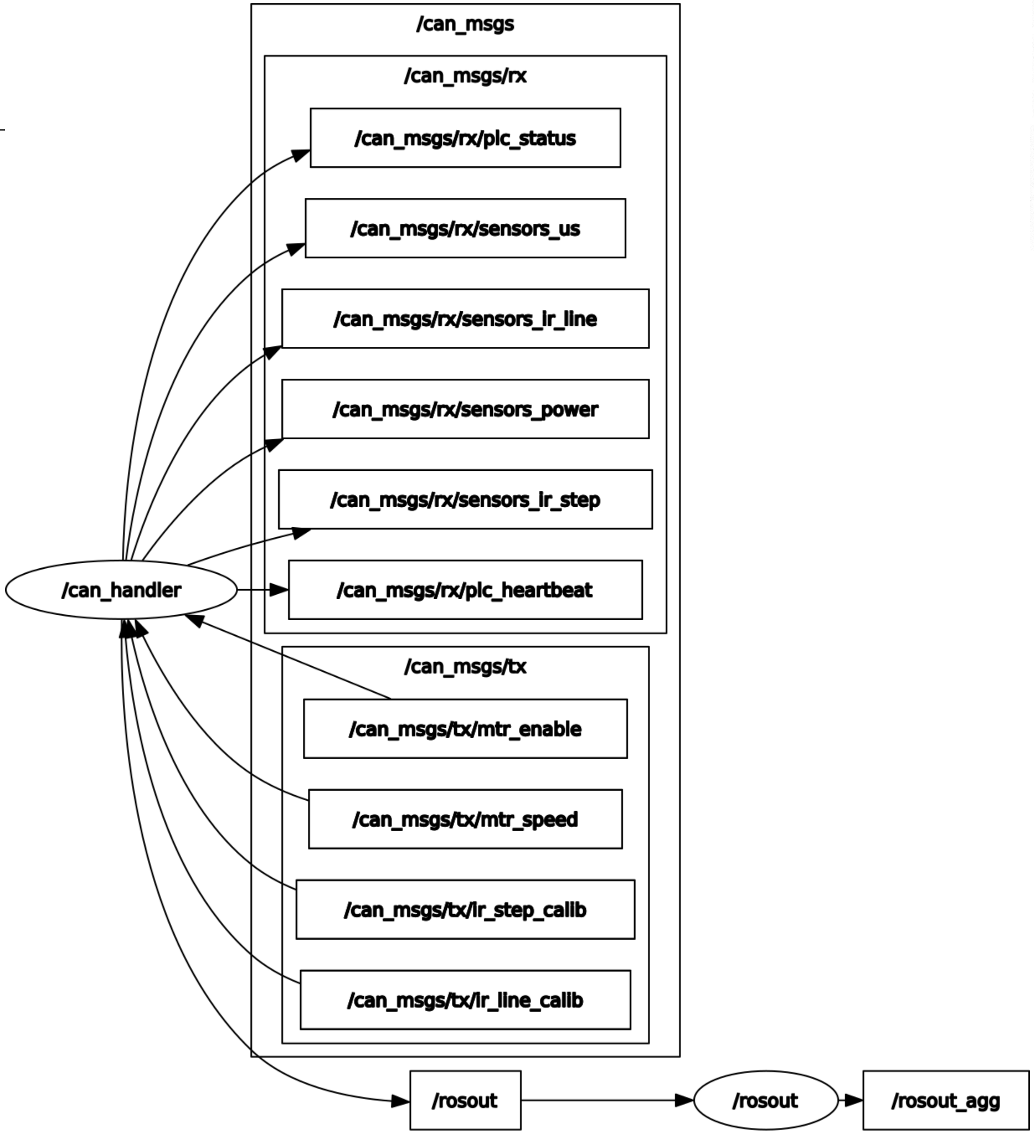
ID	Name	Sender	Signals	Start bit	Signal type	Bits	Factor	Offset	Min	Max	Unit
0x100	Heartbeat	PLC**	Heartbeat	0	Uint	8	1	0	1	1	-
0x101	IR Line Detection	Odroid*	IR_1_1	0	Uint	8	1	0	0	1	bool
			IR_1_2	8							
			IR_1_3	16							
			IR_1_4	24							
			IR_1_5	32							
			IR_1_6	40							
0x102	IR Step Detection	Odroid*	IR_2_1	0	Uint	8	1	0	0	1	bool
			IR_2_2	8							
0x103	US Object Detection	Odroid*	US_1	0	Uint	16	1	0	0	1	-
			US_2	16							
0x104	Power Monitor 24V	Odroid*	PWR_SENS_24V_V	0	Uint	16	1	0	0	65535	mV
			PWR_SENS_24V_I	16							mA

- Request & Response model

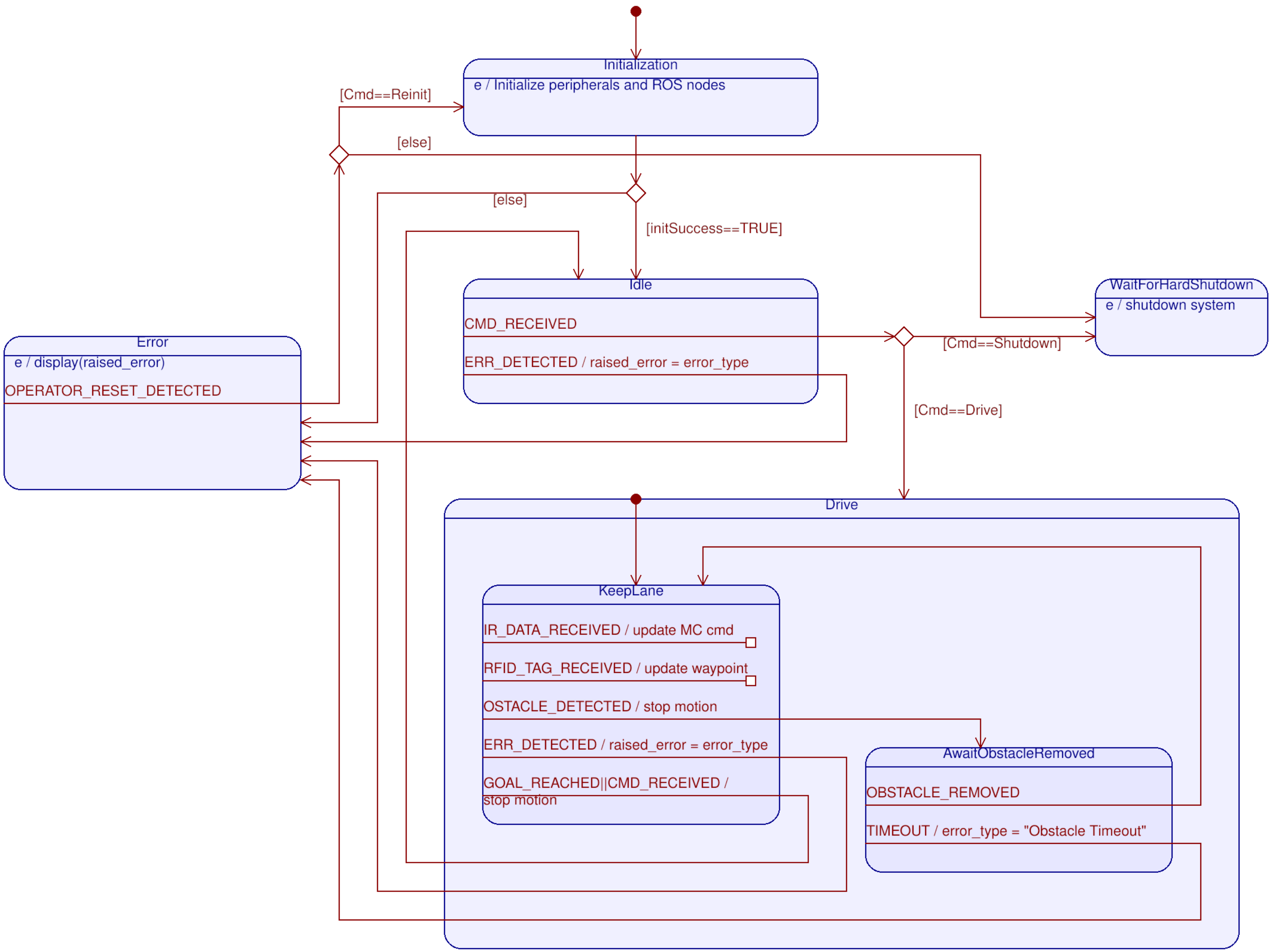
System Design
ROS - Communication

- Decoding / Encoding using Python Module cantools and CAN database file .dbc

- Nodes subscribe to rx & tx topics to receive and send CAN messages



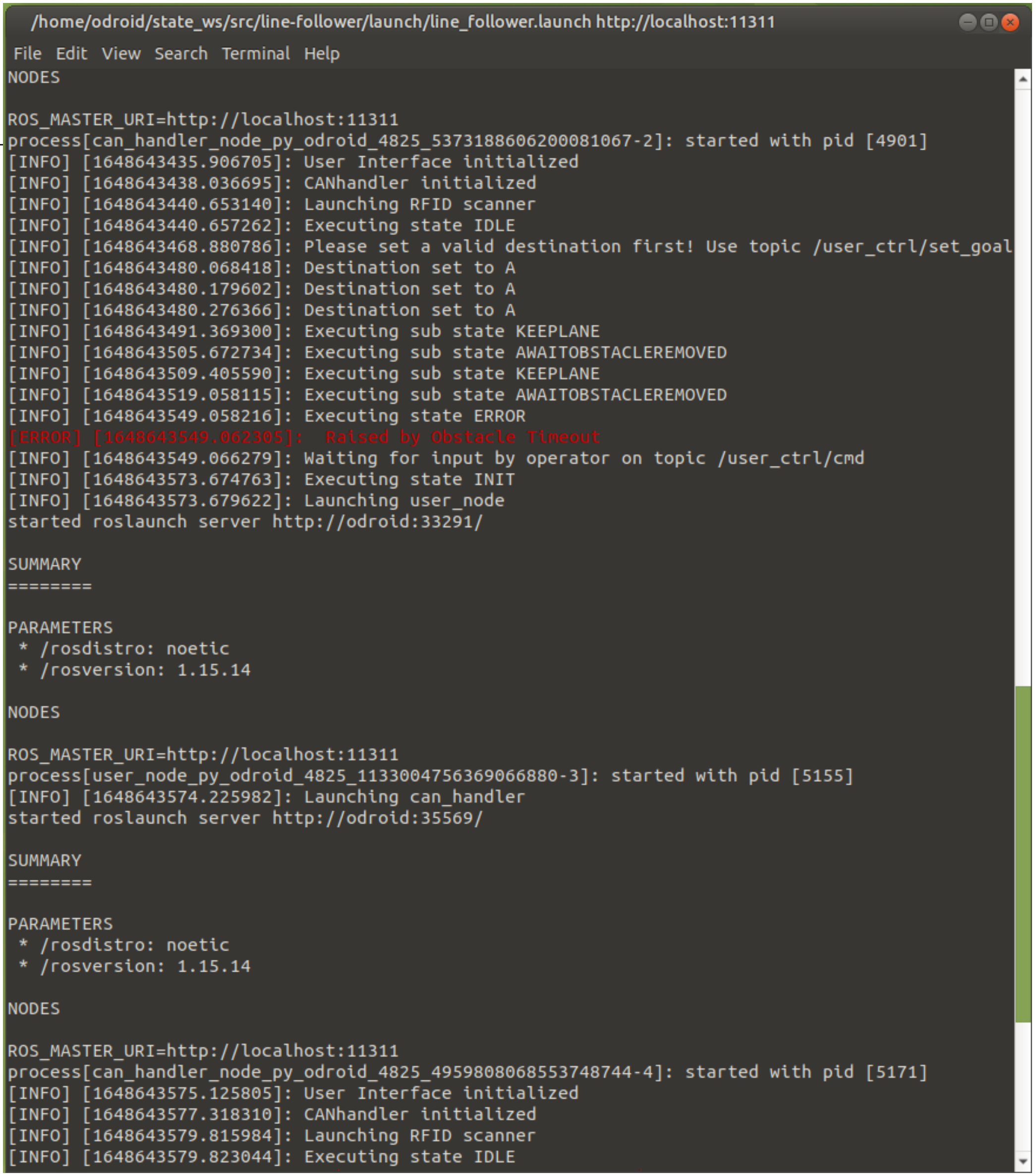
System Design
ROS – State Machine



System Design
ROS – State Machine

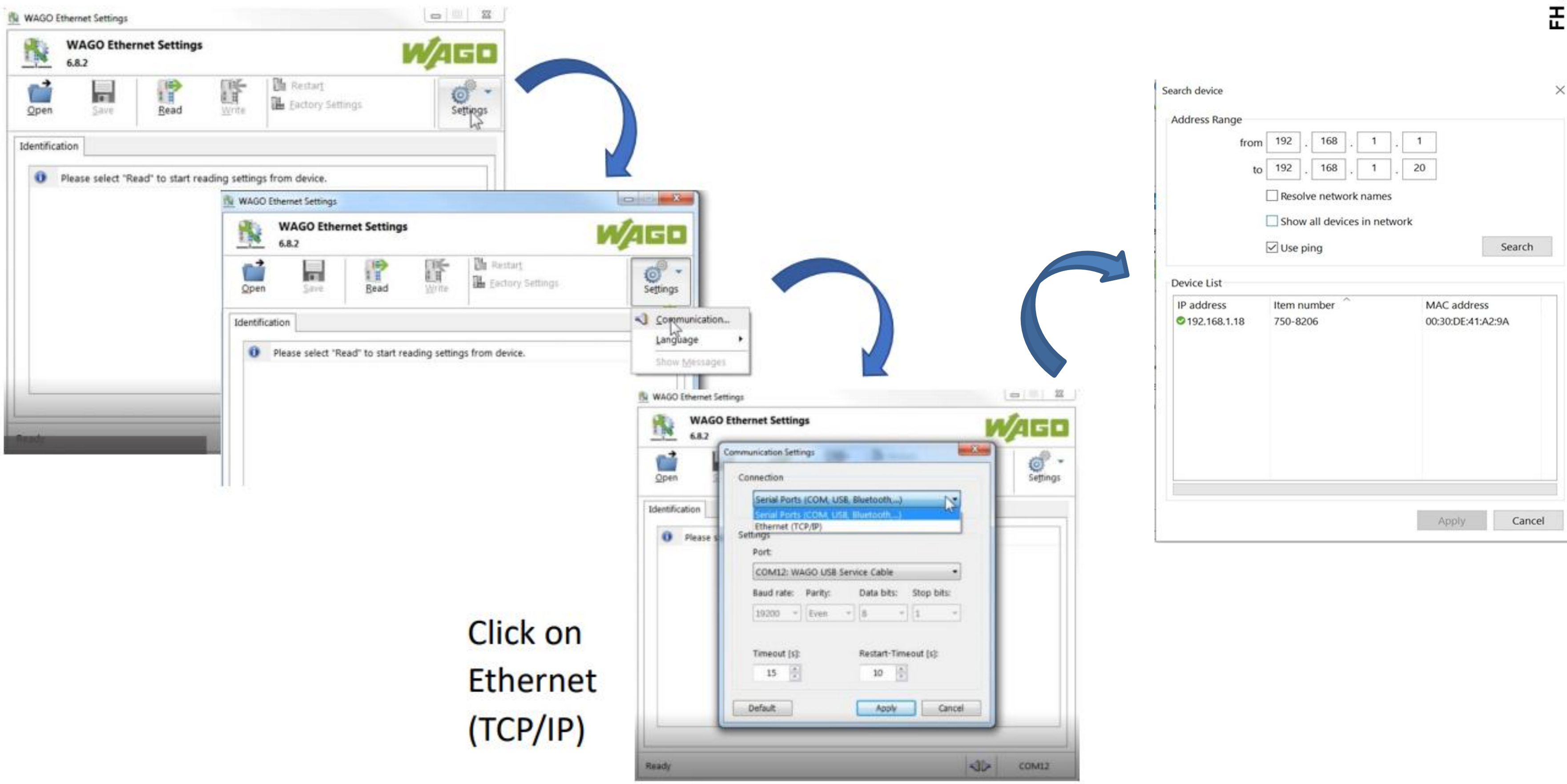
- User Control interface via ROS Topic

- Successful test with simulated PLC



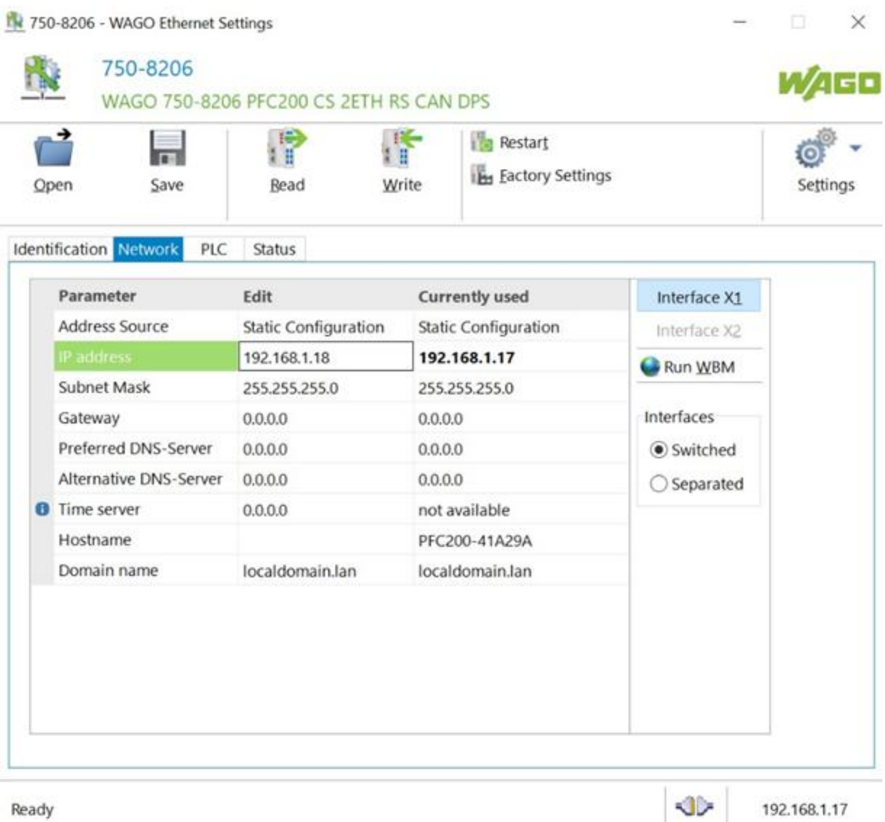
FH AACHEN UNIVERSITY OF APPLIED SCIENCES

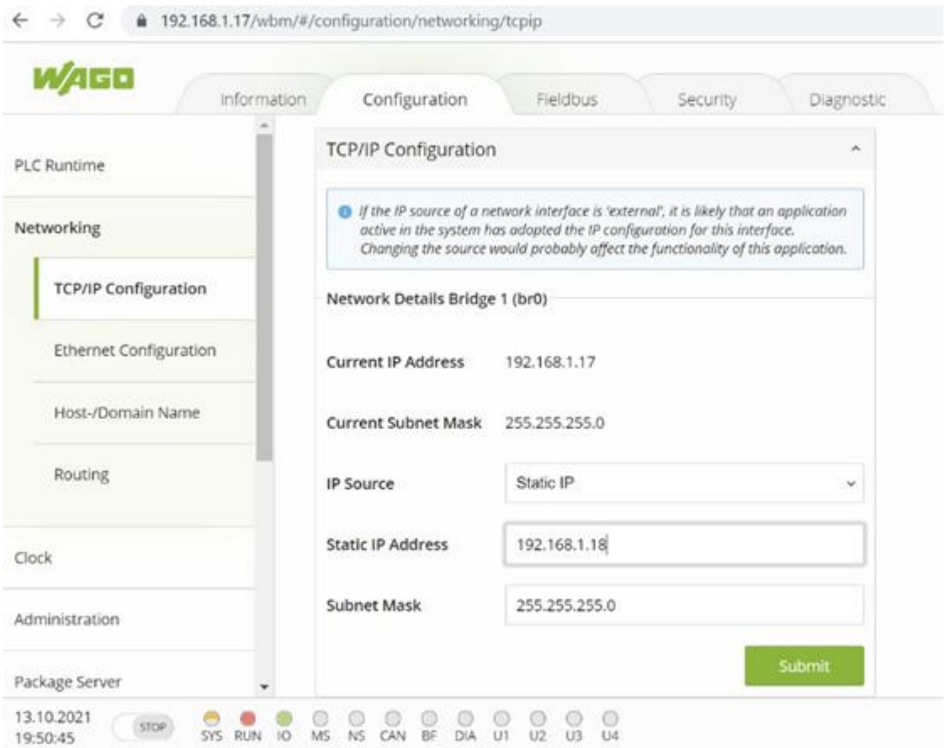
PLC
Connecting to the Wago Ethernet Settings



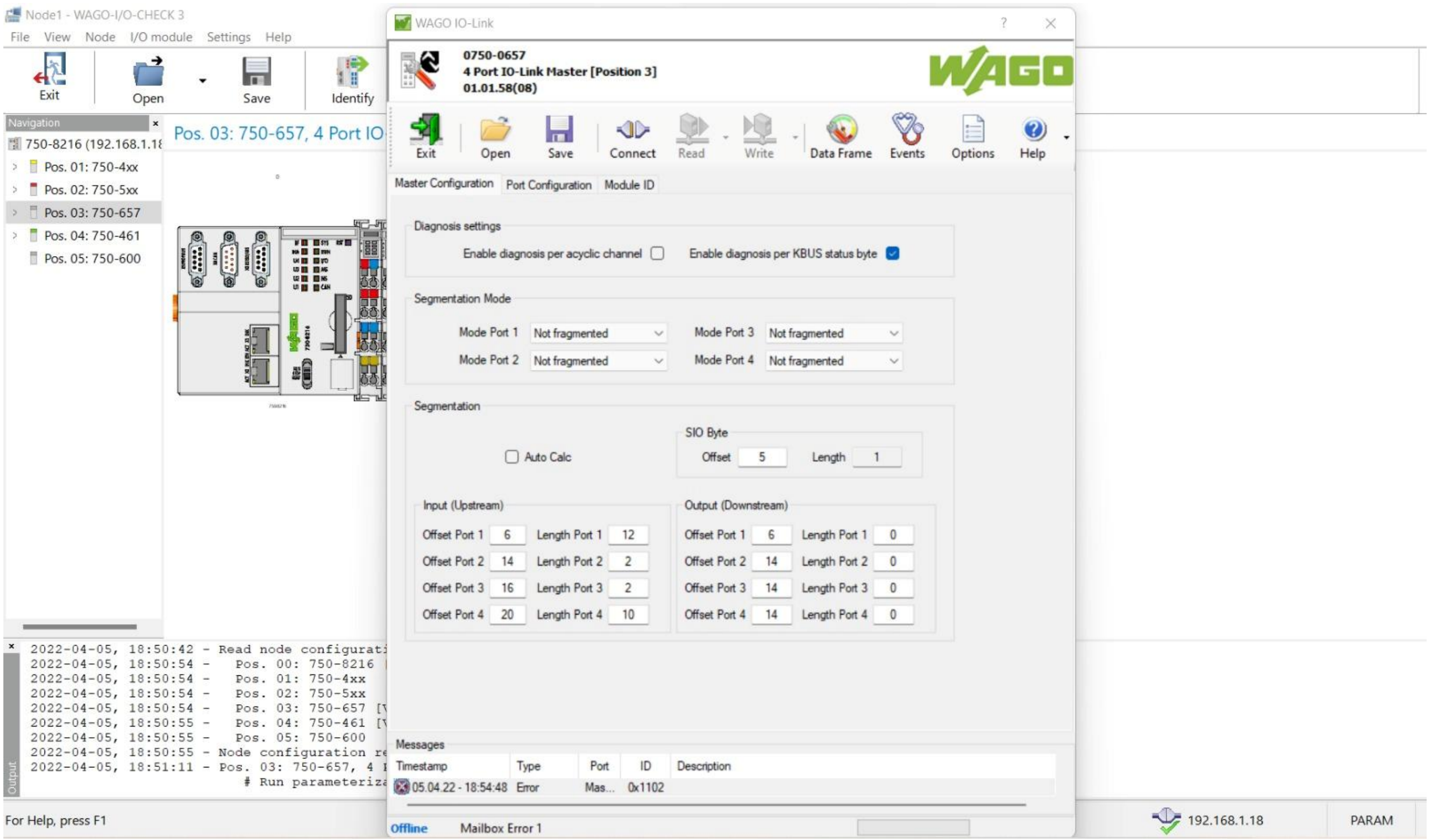
Click on
Ethernet
(TCP/IP)

PLC
Reset the PLC

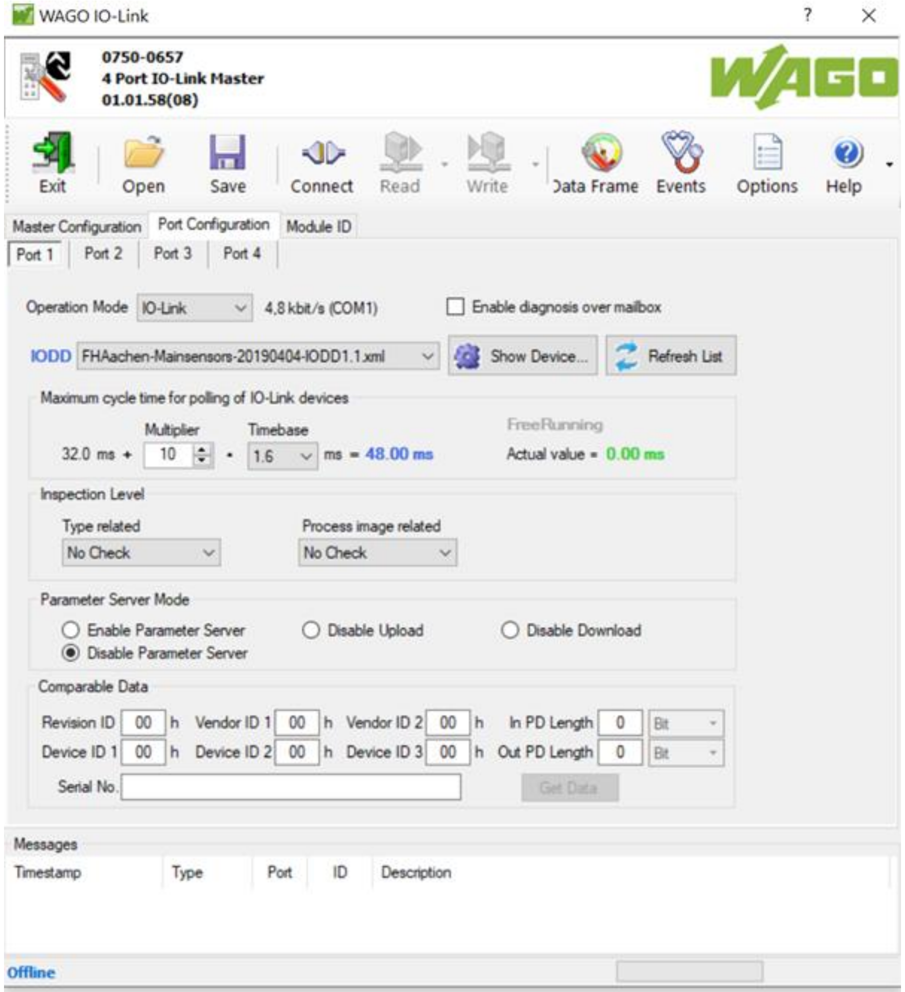




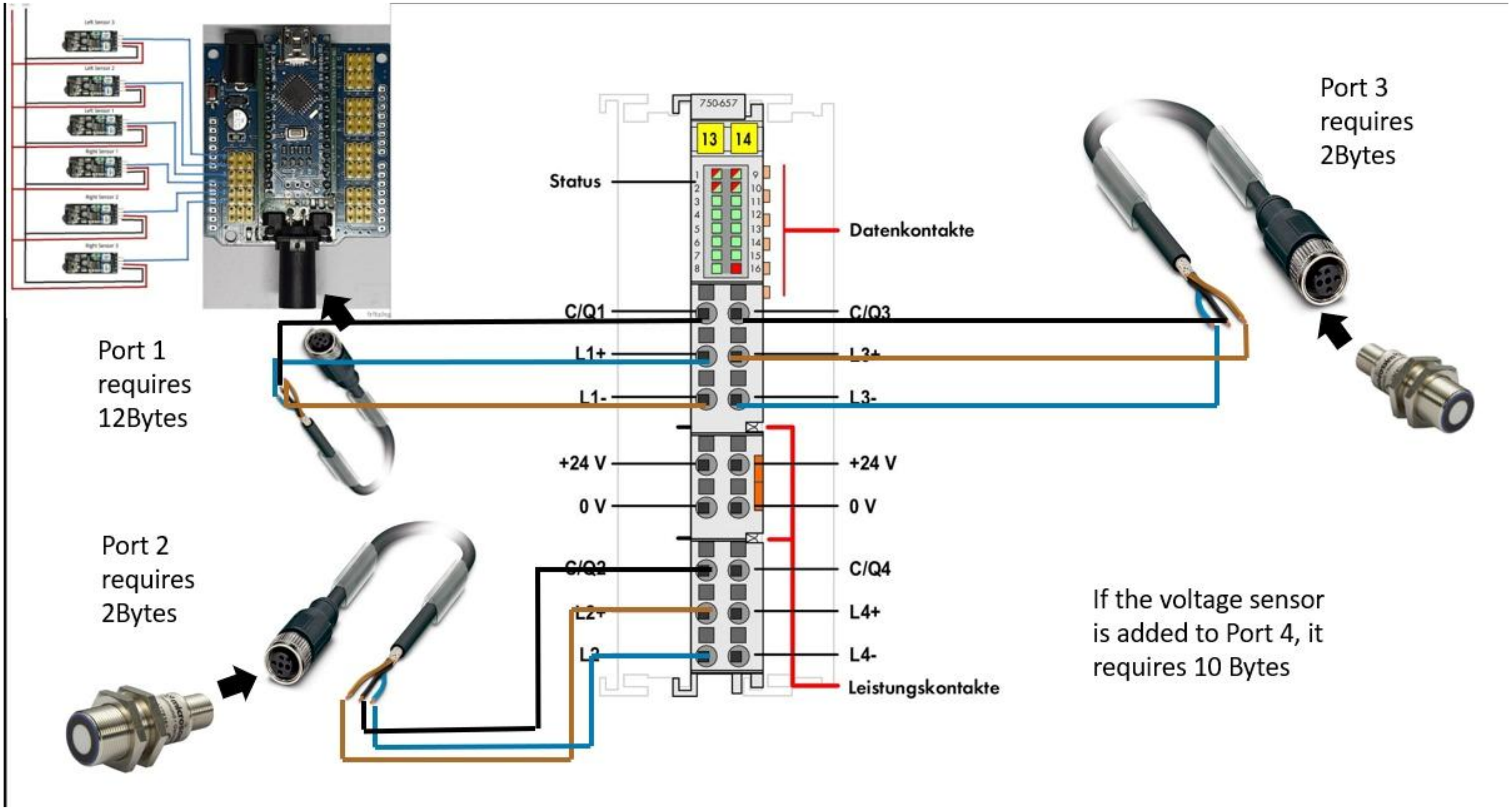
PLC
Configure in IO-Check 3



PLC
Configure in IO-Check 3



PLC
Connecting the sensors to the PLC



PLC
IO Link Port Function Blocks

Port 1: IO shield connecting to 6 IR sensors
Port 2: Ultrasonic Sensor
Port 3: Ultrasonic Sensor
Port 4: IO shield connecting to Voltage sensor

Function Block Library:WagoAppIOLink

Function Block: FbIOL_PortData

```
//Access the IOLink Process Data
IOLProcessData( xEnable:=bEnable,
            bIO_LinkPort:=byIOLinkPort,
            I_Port:=IO_Link_Master, //Name of the IO-Link Master in the "Device Structure" Tab of Ecockpit
            pTxBuffer:=ADR(typIOLink_VisuData.abyPDOut), //Process data output
            uDiTxnbytes:=SIZEOF(typIOLink_Device_Visu.abyPDOut), //Length of the data to send through PD Out
            pRxBuffer:=ADR(typIOLink_VisuData.abyPDIn),  //Process data input
            udirxbufferSize:=SIZEOF(typIOLink_VisuData.abyPDIn), //Size of the buffer used for recieving PDIn
            xTxTrigger:=PDOutTrigger, //Trigger to send proccess output data
            xCommunicationReset:=xCommReset, //Reset communication
            xError=>typIOLink_VisuData.bStatusInvalid,
            xValid=>typIOLink_VisuData.bStatusValid,
            udiRxNBytes=>typIOLink_VisuData.iPDInSize //Total recieved bytes (Process data input)
            );
IOLProcessData.oStatus.ShowResult(sDescription=>typIOLink_VisuData.sDeviceInfo); //Get a String error from the FB
```

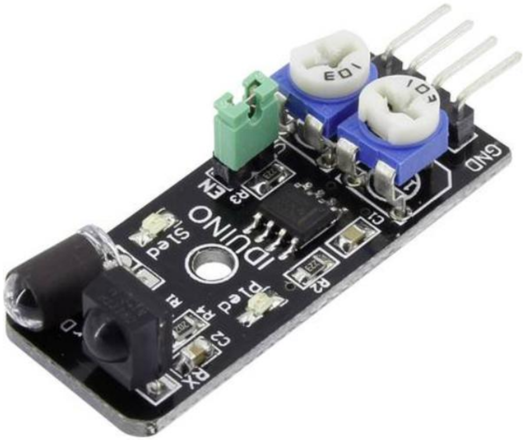
PLC
Why Microsonic Pico+35?

- Variant with 90° angled head
- IO-Link interface —> for support of the new industry standard
- Automatic synchronisation and multiplex operation —> for simultaneous operation of up to ten sensors in close quarters
- UL Listed to Canadian and US safety standards
- Improved temperature compensation —> adjustment to working conditions within 120 seconds
- Smart Sensor Profiles —> more transparency between IO-Link Devices

PLC
IR Sensor for Line following

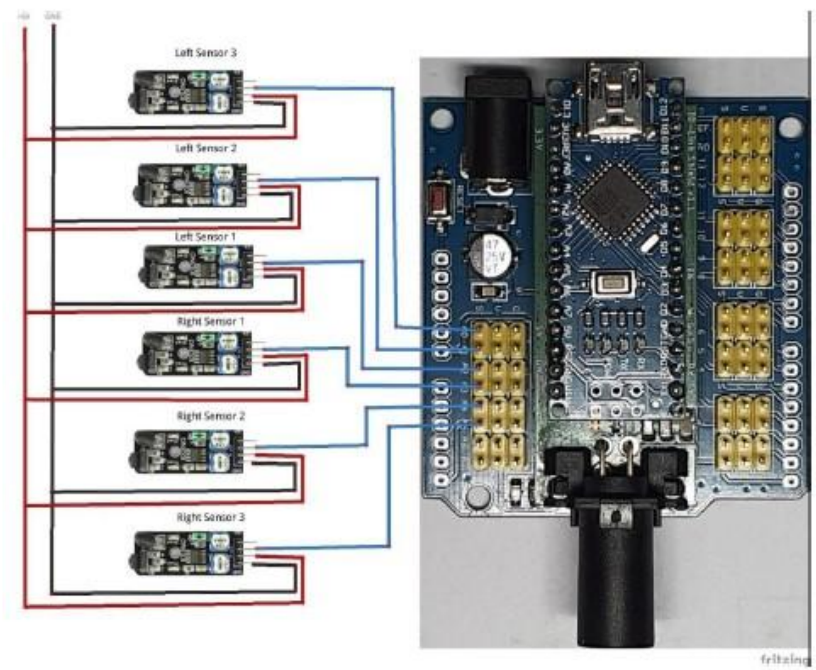
WHICH?
IDUINO IR SENSOR

WHY?
- detection distance :2-40cm
- Cost effective
- Working voltage: DC 3.3V-5V



HOW MANY?
6 IR Sensors

PLC
IR sensor to Arduino Nano schematic diagram



IR SENSORS → ARDUINO NANO → SHIELD → PLC

PLC
Softwares used

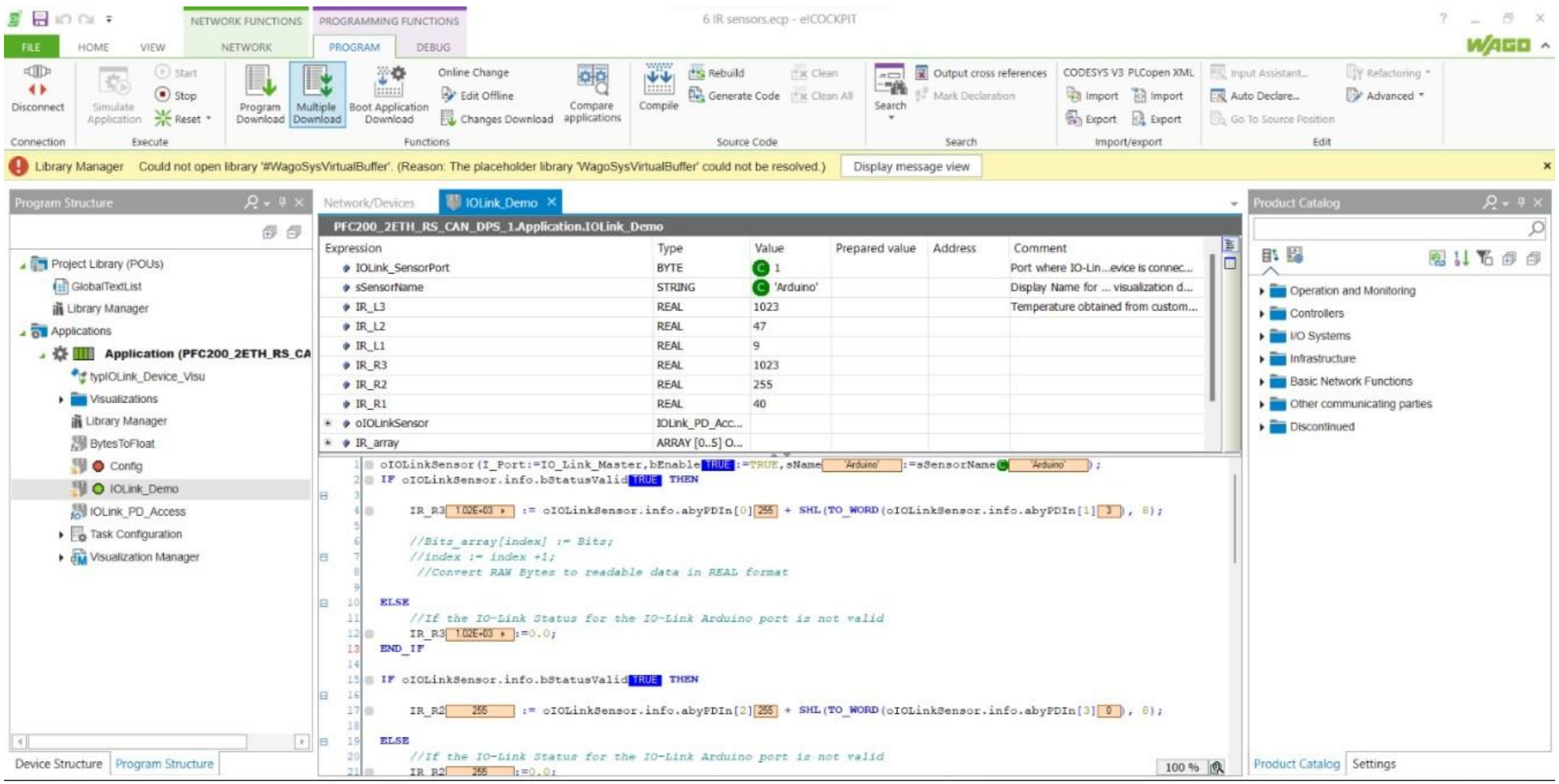
- WAGO IO CHECK 3
  To check the IO Link Communication

- WAGO e!COCKPIT
  To program the PLC

- ARDUINO IDE
  To program the Arduino

- PYTHON 3.10
  To communicate with the IFM IO Link Master

PLC
IR sensor values



Power Supply
Distribution

1. 24 V (without DC/DC converter) – wire for connection used 6mm2 (uninterrupted current flow)

- Motor Driver
- Motors
- Voltage Sensor (voltage measurement)
- Fan

2. 24V (with DC/DC converter) – wire for connection 0.75 mm^2

- PLC
- IFM
- 24V Sensors

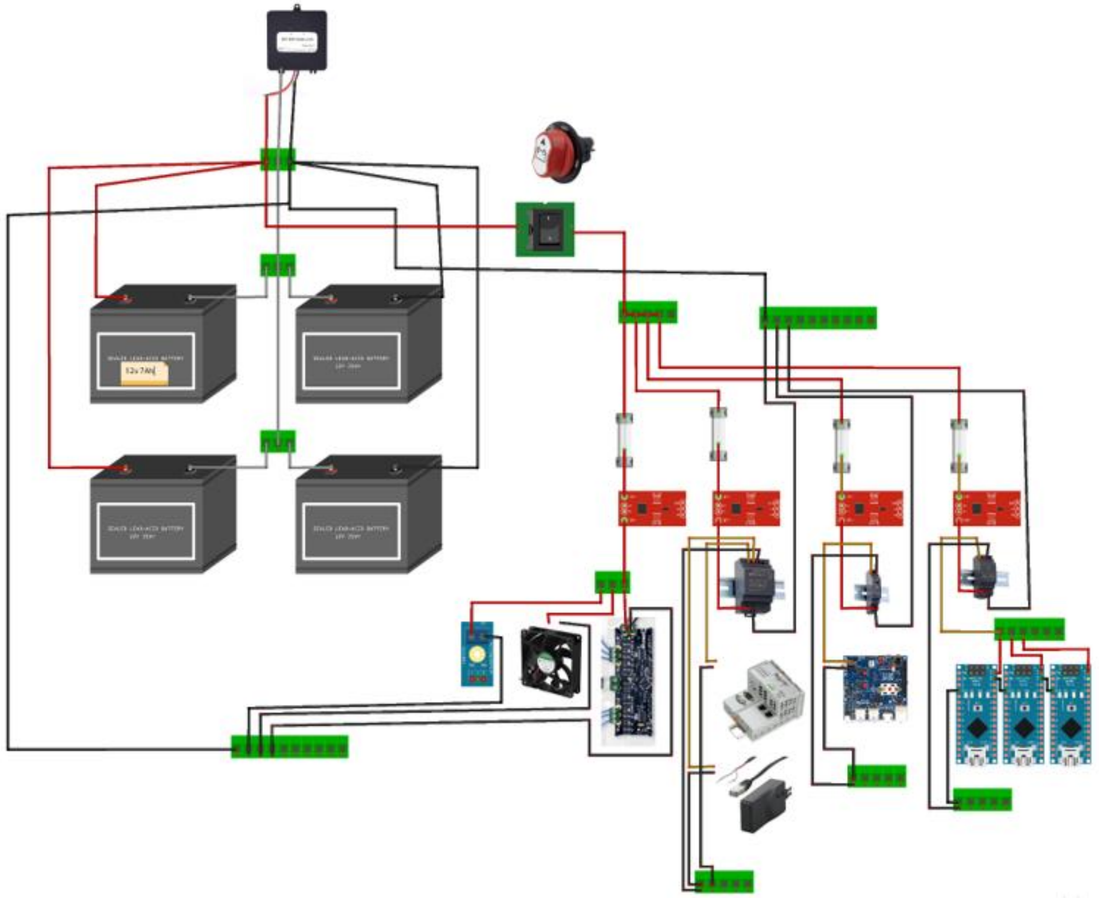
Power Supply
Distribution

3. 12V (with DC/DC converter) – wire for connection 0.75 mm^2
- Odroid N2+

4. 5V (with DC/DC converter) – wire for connection 0.75 mm^2
- Arduino Nano
- 5V Sensors

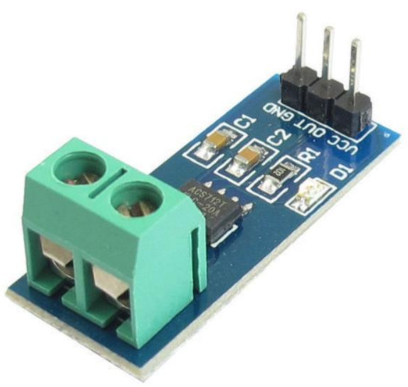
FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

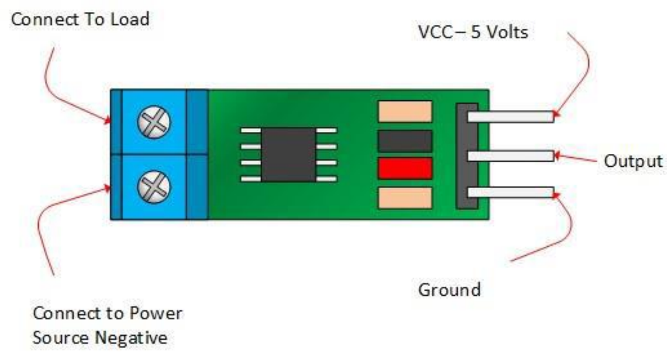
Power Supply
Distribution



Power Supply
Power monitoring sensors

- Current sensor

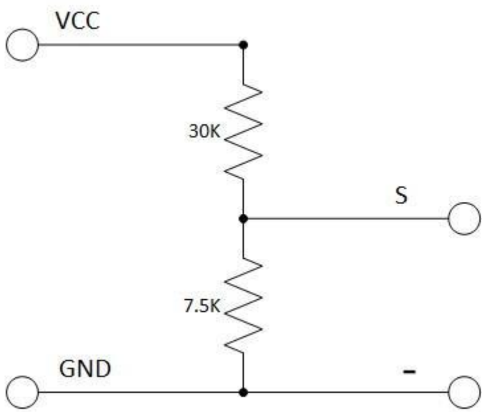


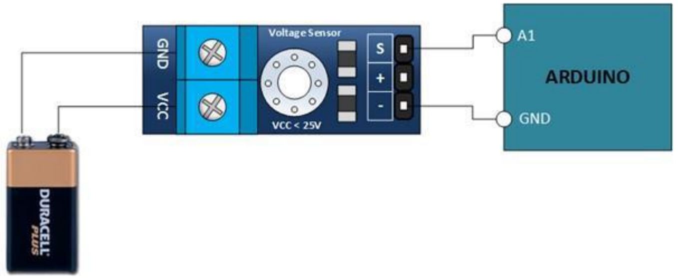


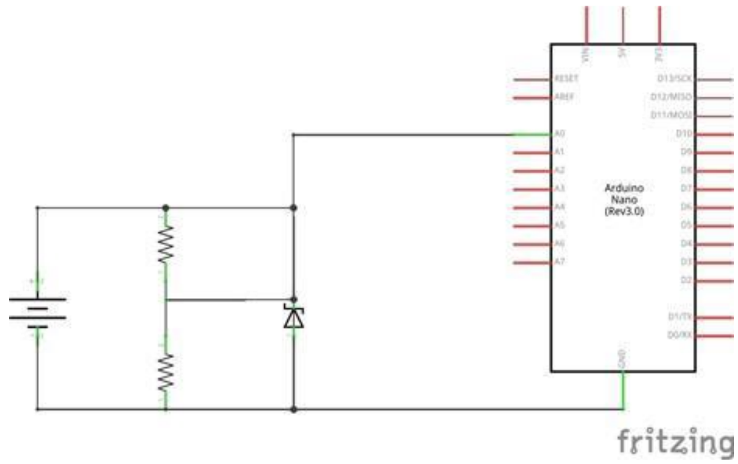
CurrentSensor
- uint8_t sensorPin - const int sensitivity - const int offsetVoltage - int adcValue - uint16_t adcVoltage - uint16_t currentValue - uint16_t adValue - int inputVoltageStandard
+ CurrentSensor(uint8_t pin) + uint16_t CurrentValue() + uint8_t CurrentMapValue() + outData CurrentMapValuesBytes()

Power Supply
Power monitoring sensors

- Voltage sensor







VoltageSensor

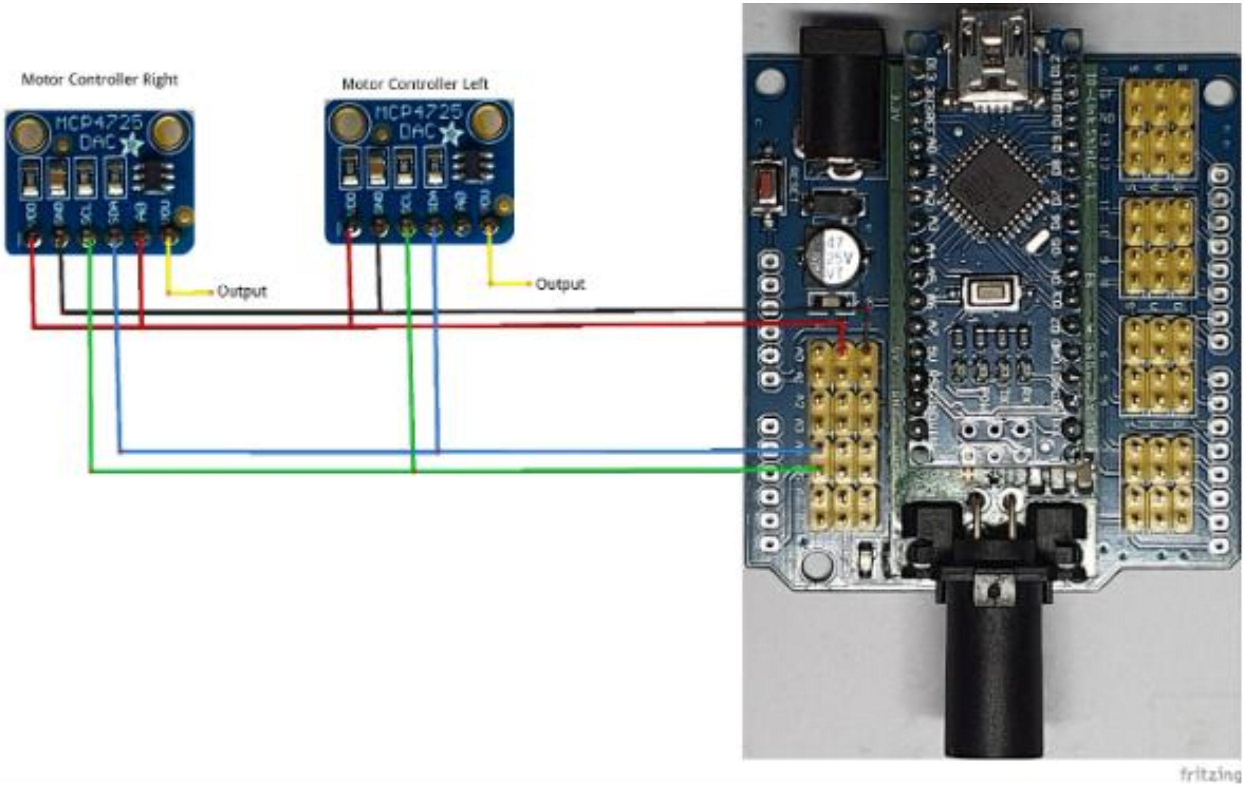
- uint8_t sensorPin
- uint16_t adcVoltage
- uint16_t inVoltage
- uint16_t R1
- uint16_t R2
- uint16_t refVoltage
- int adcValue
- uint16_t adValue

+ VoltageSensor(uint8_t pin)
+ uint16_t VolatgeValue()
+ uint8_t VoltageMapValue()
+ outData VoltageMapValuesBytes()

Motor controller
Ebike controller

- Ebike controller is controlled using analog signal. The controller responds to analog signal between the range 1.2v to 3.8v.
- To supply analog output, DAC MCP4725 is used. It is connected to Arduino nano using I2C communication.
- For the project’s application, the MCP4725 receives commands from PLC via IO-Link Shield. The shield mounts Arduino nano to which the MCP4725 module is connected. The output (VOU pin) is connected to the respective motor controller.

Motor controller
Ebike controller

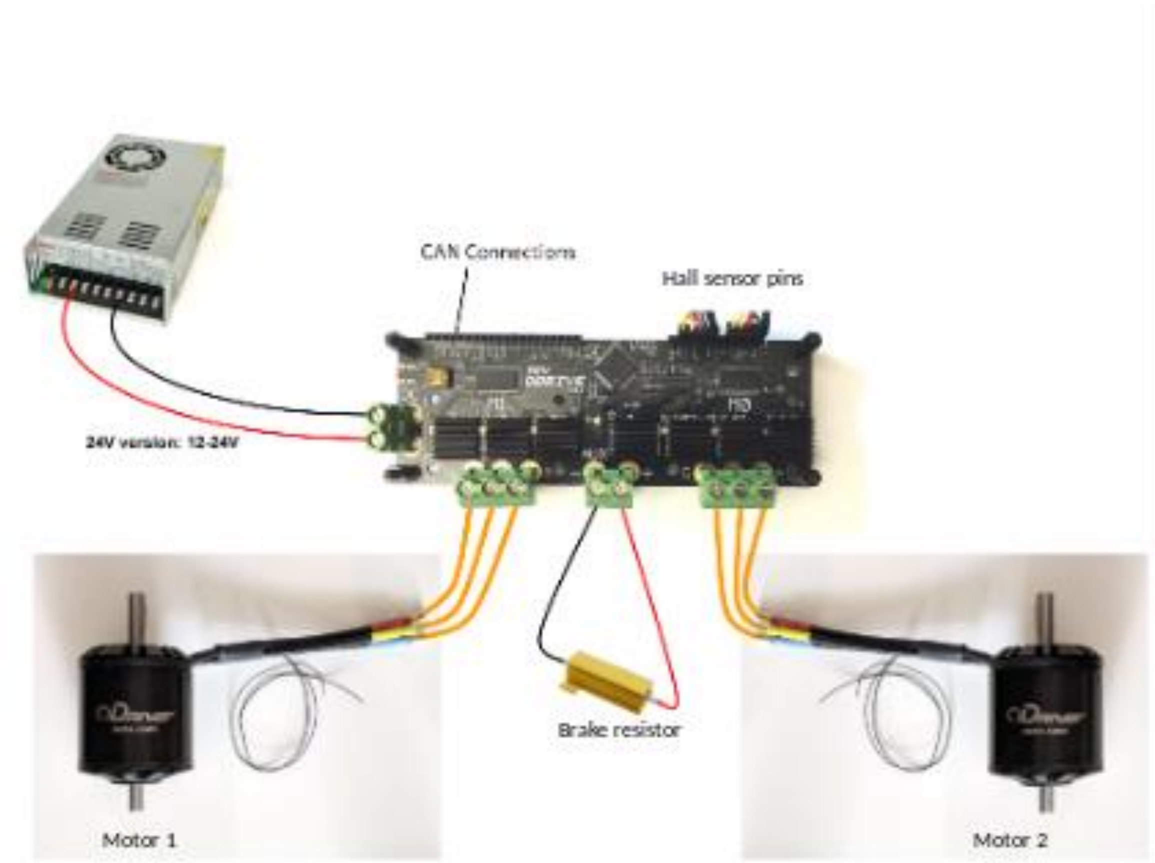


Motor Controller Right	Motor Controller Left	Arduino Nano
VCC	VCC	+5V
GND	GND	GND
SCL	SCL	A5
SDA	SDA	A4
A0	-	+5V

Motor controller
Odrive controller

- The project uses hoverboard BLDC motors. Odrive motor controller allows controlling two BLDC motors simultaneously using one board.
- The commands can ve sent via various communication protocols such as UART, CAN and USB. The motors will be connected in similar manner. BLDC motors consists of hall sensors which are connected to ODrive board.

Motor controller
Odrive controller



Motor controller
Current status

- The odrive motor controller had issues and the the controllers from the previous batch is being used now.

Motor controller

The data sent from the PLC is received by the IO-Link shield. The 7 bytes of data received is structured in the following order-

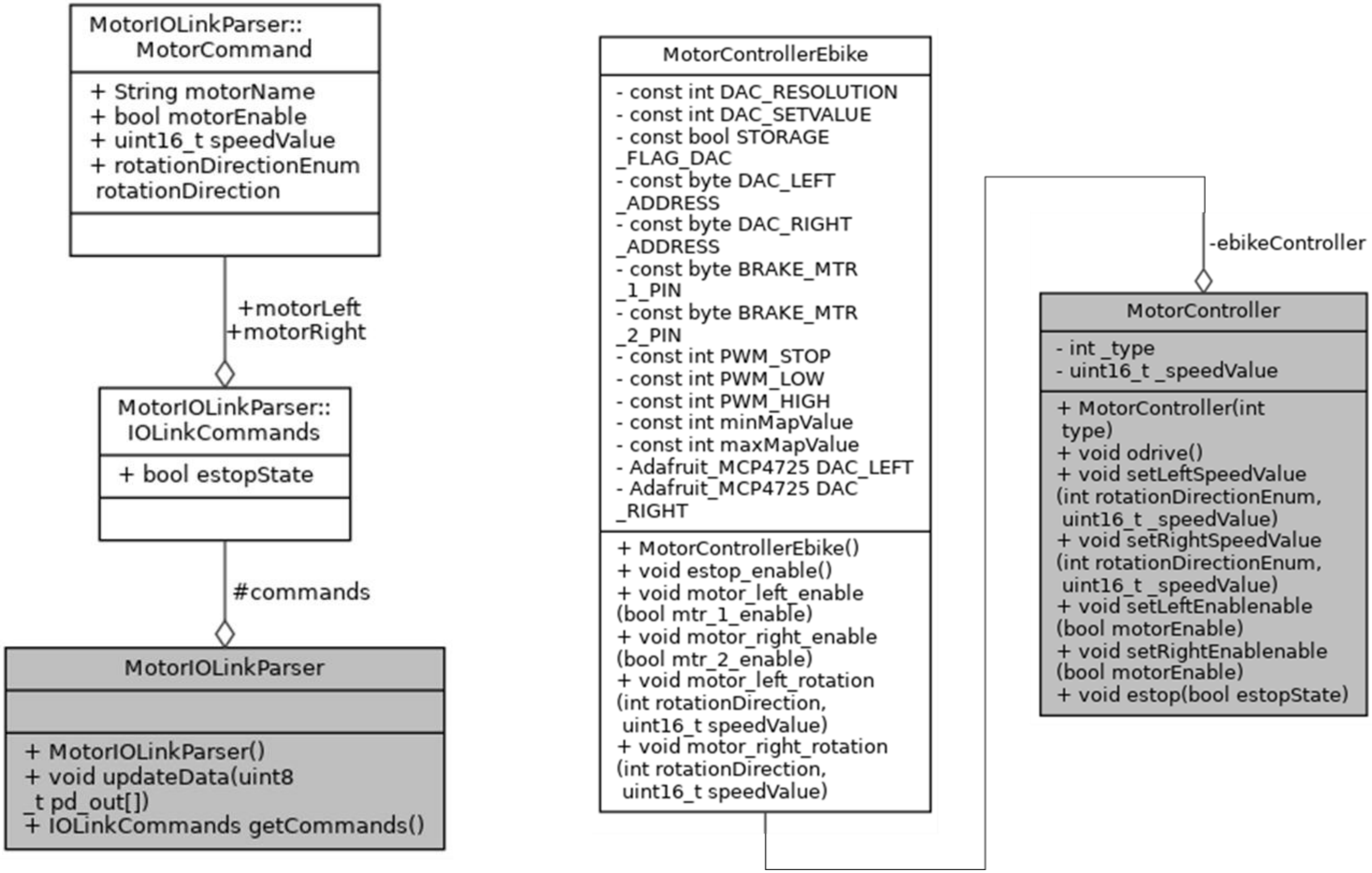
MTR_ESTOP, MTR_1_ENABLE, MTR_2_ENABLE, MTR_1_SPEED_H, MTR_1_SPEED_L, MTR_2_SPEED_H, MTR_2_SPEED_L.

Motor controller

The Arduino nano also sends feedback to the PLC via IO-Link communication. The data is of 6 bytes structured in the following order:

MTR_ESTOP_STATUS,MTR_ENABLE_STATUS,MTR_1_SPEED_STATUS_H,MTR_1_SPEED_STATUS_L, MTR_2_SPEED_STATUS_H, MTR_2_SPEED_STATUS_L.

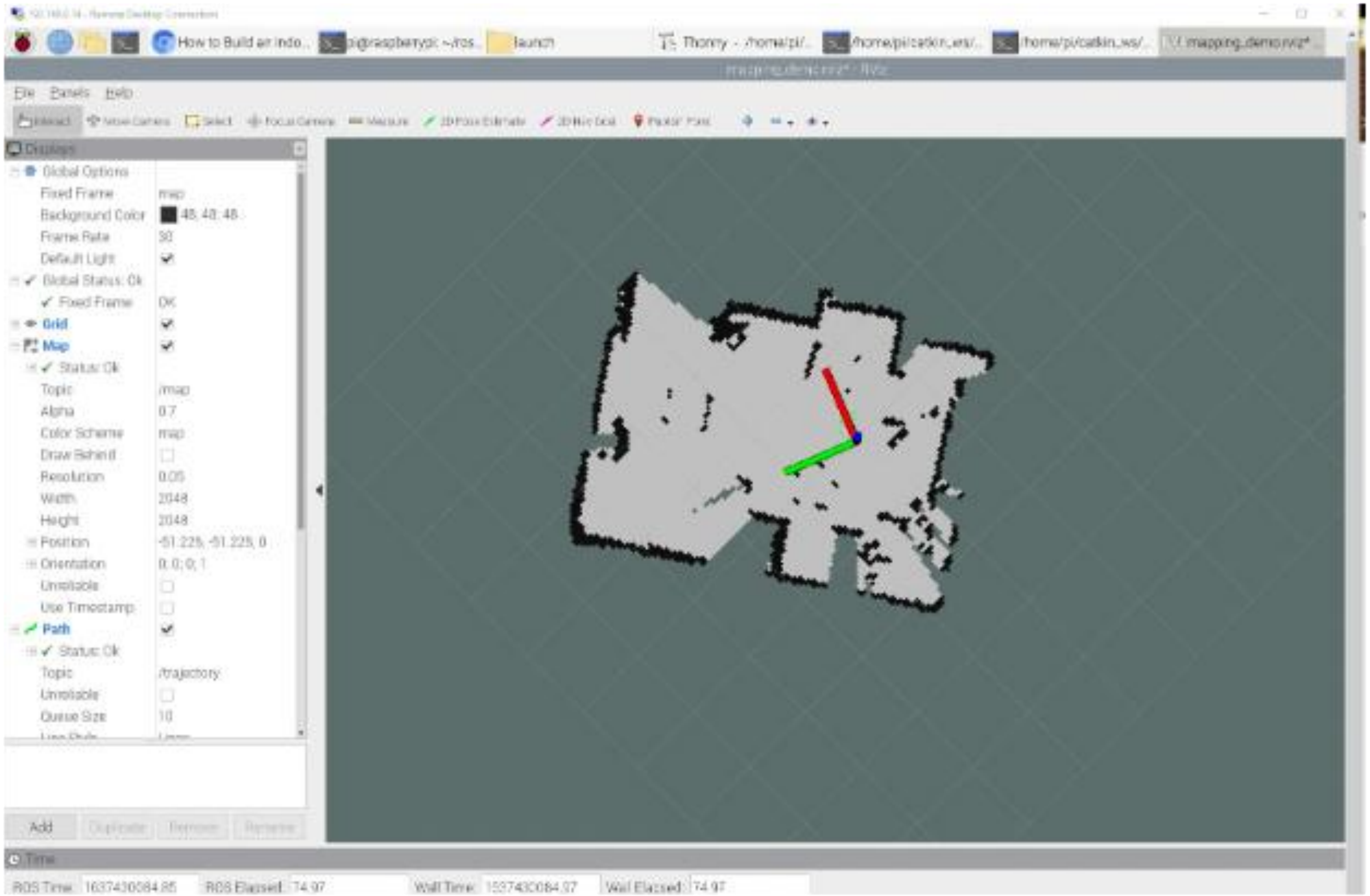
Motor controller



Lidar

- For the low-cost project, we considered using RP LIDAR a1 by Slamtec.
- The lidar can provide a variable frequency of 5Hz and 10Hz.
- It can detect objects within the circumference with 12 meters radius

Lidar
Future implementation for navigation with map generation



Conclusion & Outlook

- Calibration has to be done
- First Driving Tests to come
- Path Planning Concept needs to be implemented

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

FH Aachen
Faculty of Mechanical Engineering and Mechatronics
Mechatronics Project: Autonomous Guided Vehicle
Goethestr. 1
52064 Aachen