Schneider Electric

Intelligent Manufacturing Using industry 4.0

Intelligent Manufacturing Using industry 4.0

Otafy

Team leader

Motion

&

SCADA

Kareem

image processing

&

Code review

Ahmed

HMI

&

Leakage test

Amr

Mobil application

&

Packing system

Abd-El rheem

Filling system,

Motion

&

panel wiring

Mo'men

PLC programming

&

Node-red

Abdo

Rejection,

Dashboard

&

Documentation

Why Control Department ?

"Controls Engineer" vacancy

Responsibilities:

Design PLC programs according to the system description& functionality including safety programming, communication with the upper level SCADA system. Also, configuration of/ communication with the following: drives, scanners, barcode readers, weighing systems

HMI programing: using Allen Bradley factory talk view or WinCC flexible or Vijeo Designer to monitor& control different system components

•Commissioning and Startup includes:
- Download PLC& HMI programs
- Perform signal check from field to control panels
- Check Emergency Zones/ESTOP functionality
- Startup the system& check the product flow for Stand-alone PLC controlled areas and the fully integrated system

Qualifications:
• 2 years of experience
•Very Good knowledge of PLC Siemens or Allen Bradley, Schneider is a plus.
•knowledge of PLC programing languages (STL, FBD, LAD), Siemens SCL
•knowledge of industrial networks: Ethernet/IP,

•knowledge of PLC programing languages (STL, FBD, LAD), Siemens SCL
•knowledge of industrial networks: Ethernet/IP, Profinet, Profibus, Device Net, CAN and ASi
•Very good knowledge of Motor control techniques: Using soft-starters or variable drives (VFD)
•Very Good knowledge of HMI software packages [AB factory talk view, Siemens WinCC flexible]

Interested candidates, please send your CV to careers@advansys-esc.com, vacancy name in the title

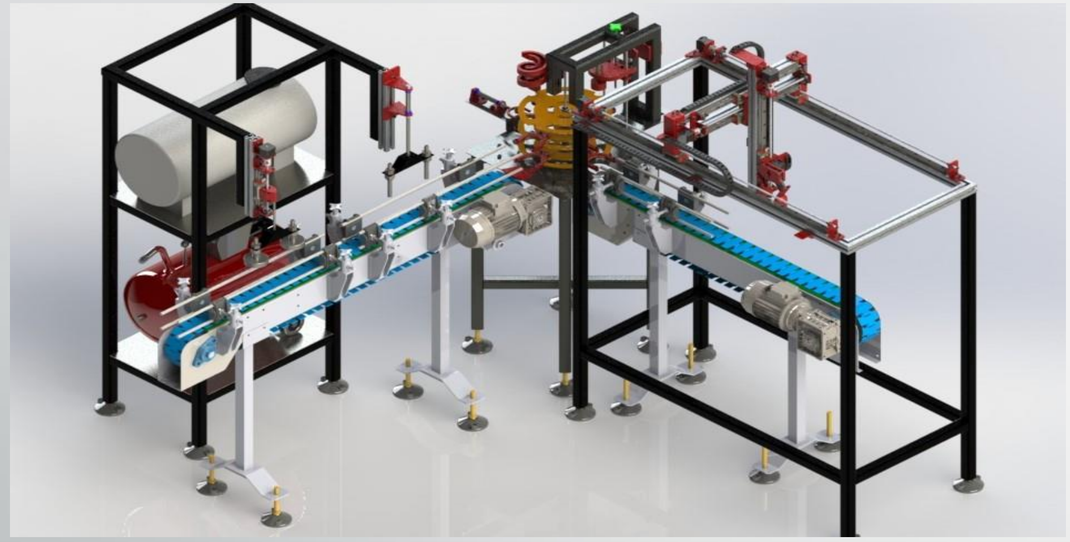
Project ILOS

PLC programing	→	EcoStruxure Machine Expert
HMI programing	→	Vijeo Designer Basic
SCADA system design	→	EcoStruxure Machine SCADA Expert
Industrial communication	→	Modbus Serial & Modbus TCP/IP
Variable speed drives	→	Altivar ATV320
Image processing	→	Using python
Visualization on mobile	→	Remote red App & Our mobile App

Our sponsors

For Hardware

EGYROLL
ENGINEERING SOLUTIONS



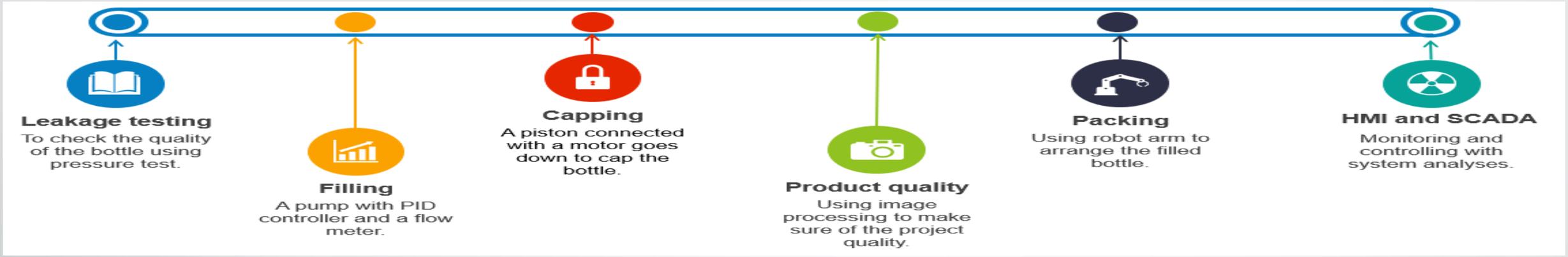
AI & IOT

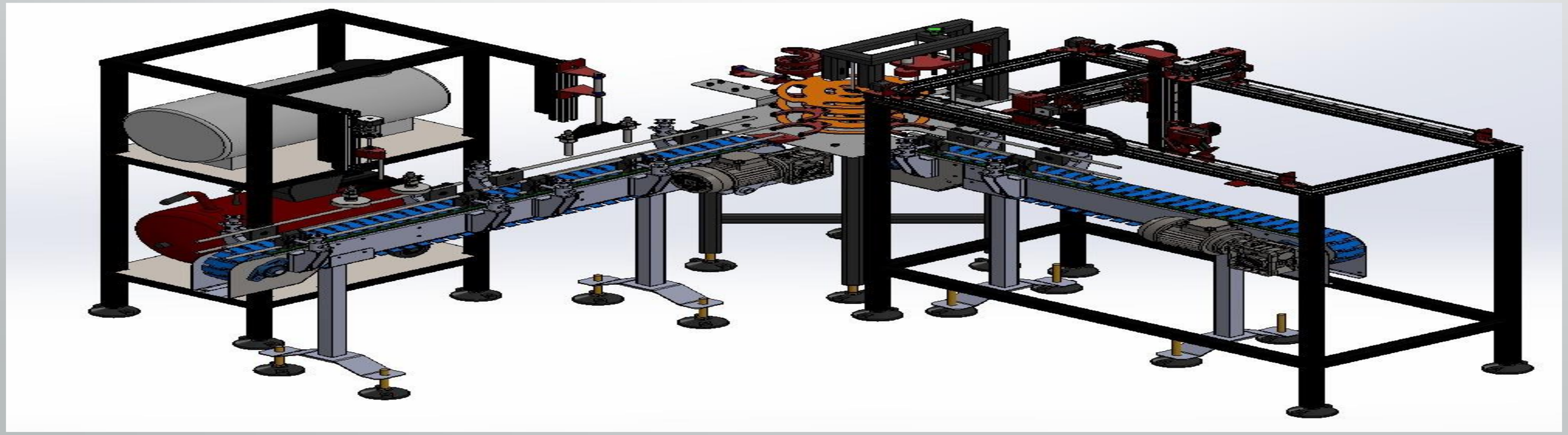
Egypt IoT & AI CHALLENGE

Control components

Schneider Electric

Project Explanation





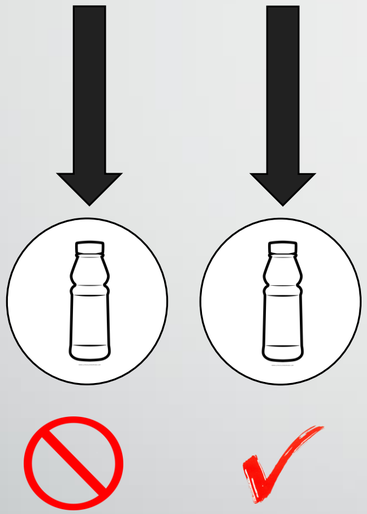
Process Flow

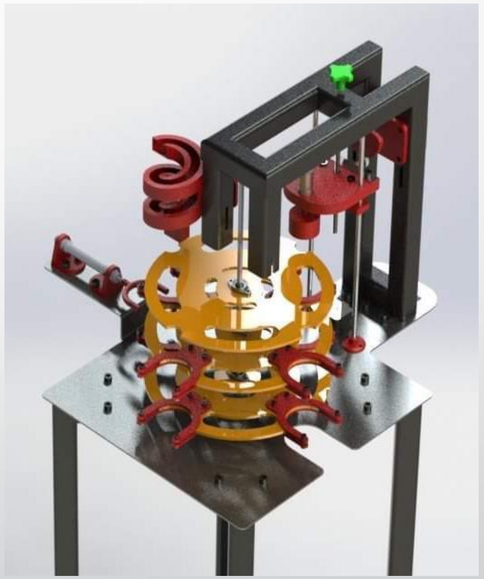
1 **Leakage System**

2 **Filling System**

3 **Capping System**

Rejected **Accepted**





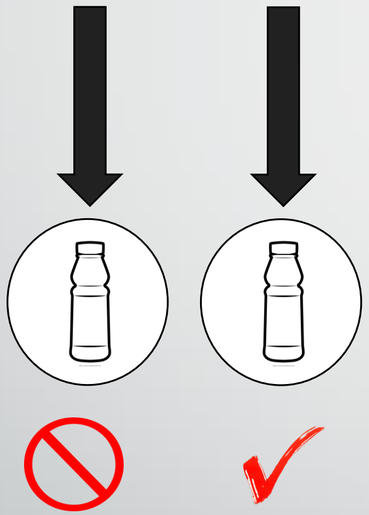
Wipe it out by Piston

Going to filling system

Process Flow

4 **Product Quality System**

Uncapped **Capped**

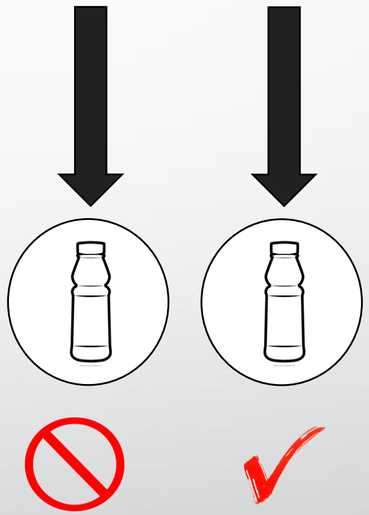


It sends to robot arm to go to unpacked side

It sends to robot arm to go to packed side

5 **Packing System**

Unpacked **Packed**



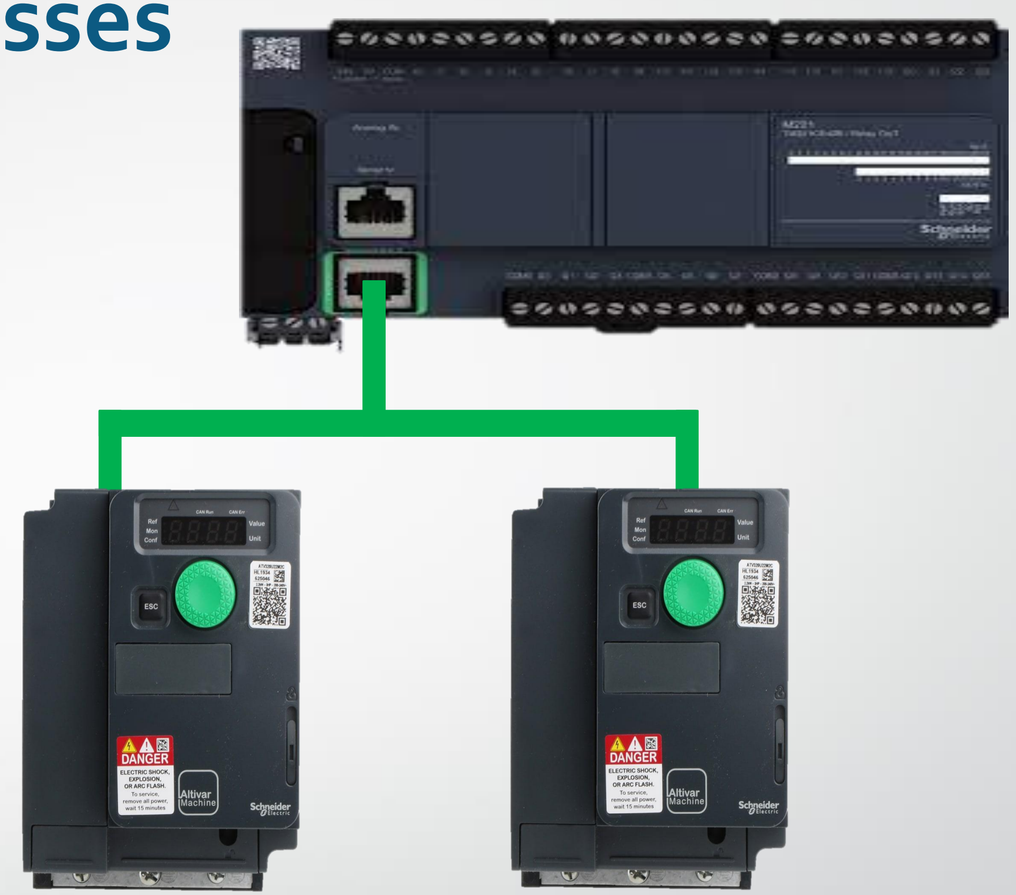
Goes to Rejected bottle box

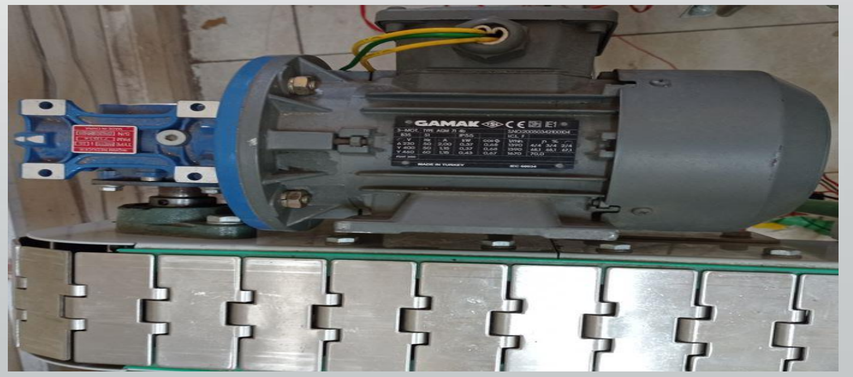
Goes to Packed bottle box (packing)

Project Processes

1. Motion System

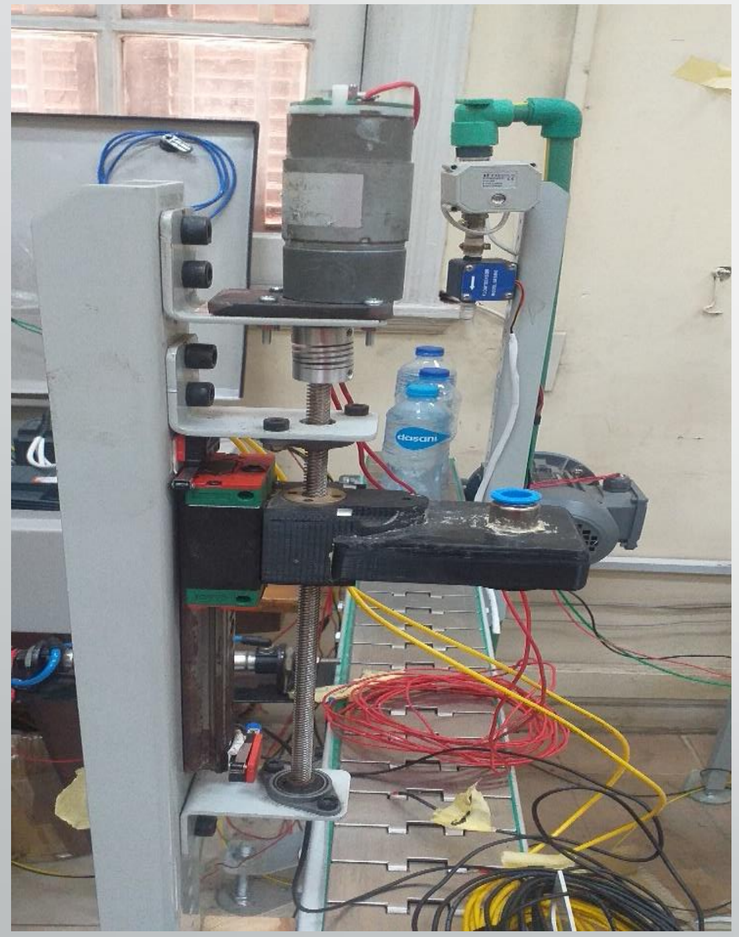
This system consists of two 3-phase induction motors driven by variable speed drive (ATV320)

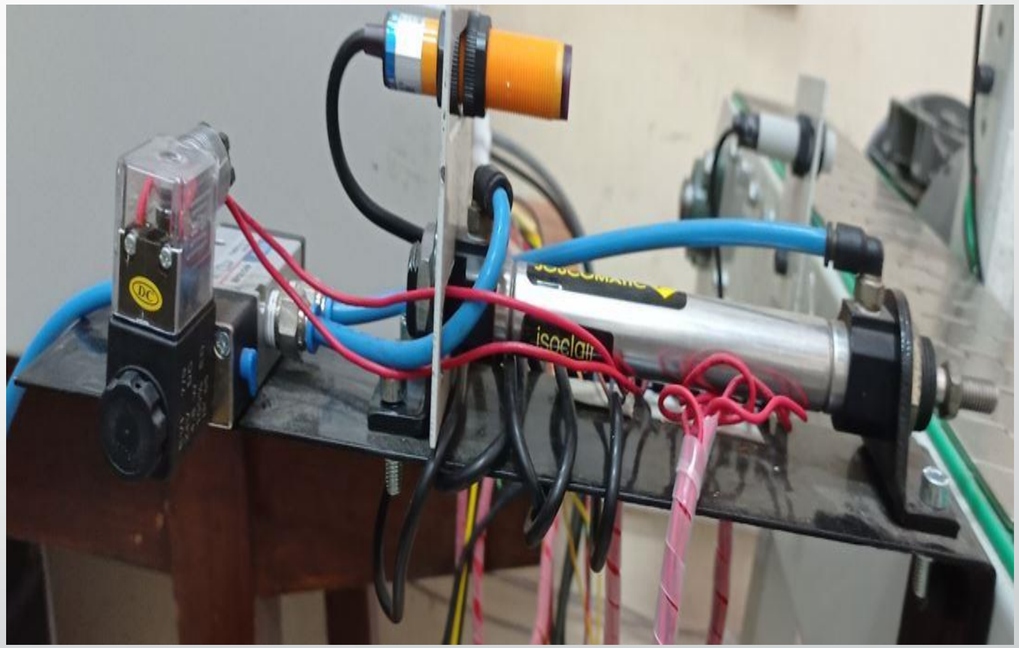




Project Processes

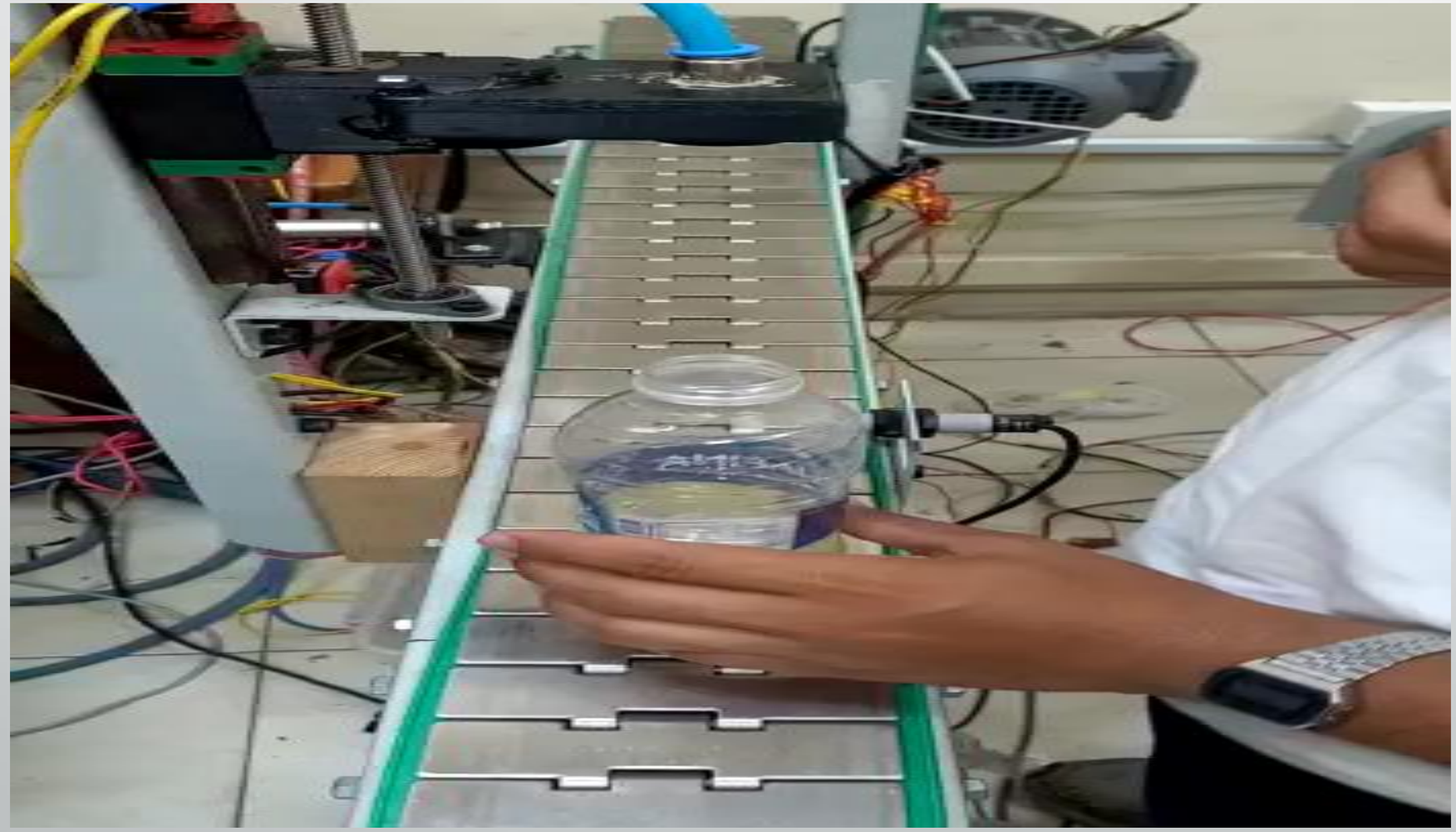
2. Leakage test System & Rejection





Project Processes

Rejected Bottle



Project Processes

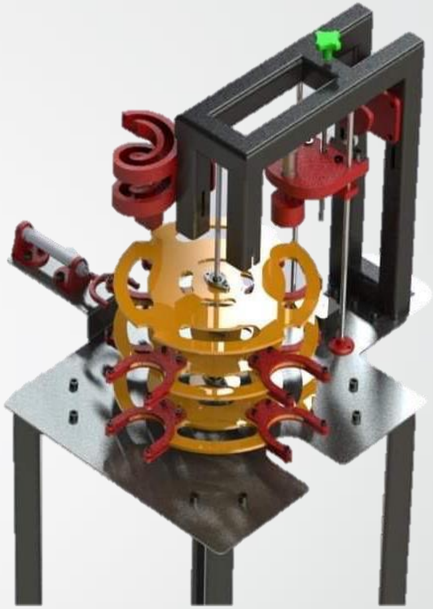
3. Filling System

Process description

- If the bottle reached the proximity sensor, stop the conveyor.
- Energize the water pump & motorized valve for 6 sec.
- Start the conveyor again.

Project Processes

3. Capping system

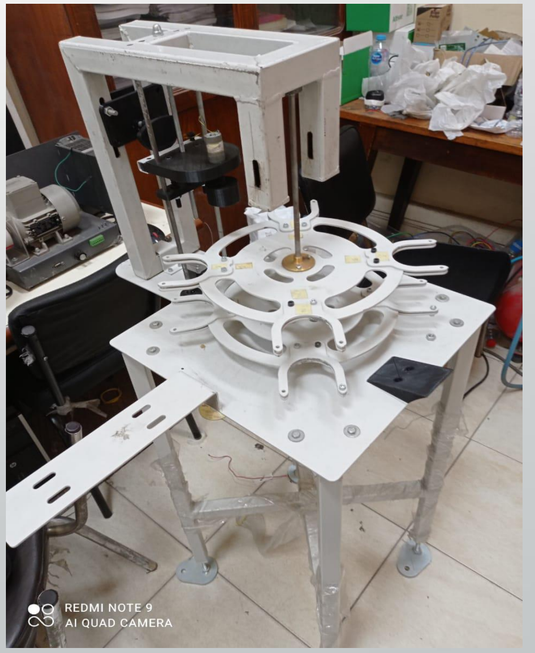


This system has 3 main functions:

1. Transfer the bottle from the first conveyor to the second conveyor.

2. put the cap on the bottle.

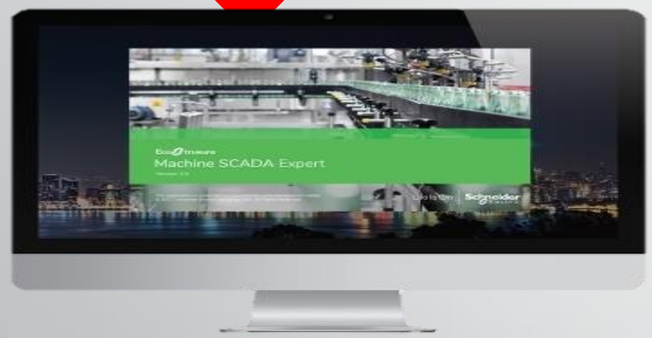
3. capping it.



this system has mechanical problems and needs to be designed again from scratch to overcome the high inertia of the moving parts.

Modbus TCP/IP

192.168.1.5



192.168.1.3

192.168.1.10

Modbus RTU

Node-Red

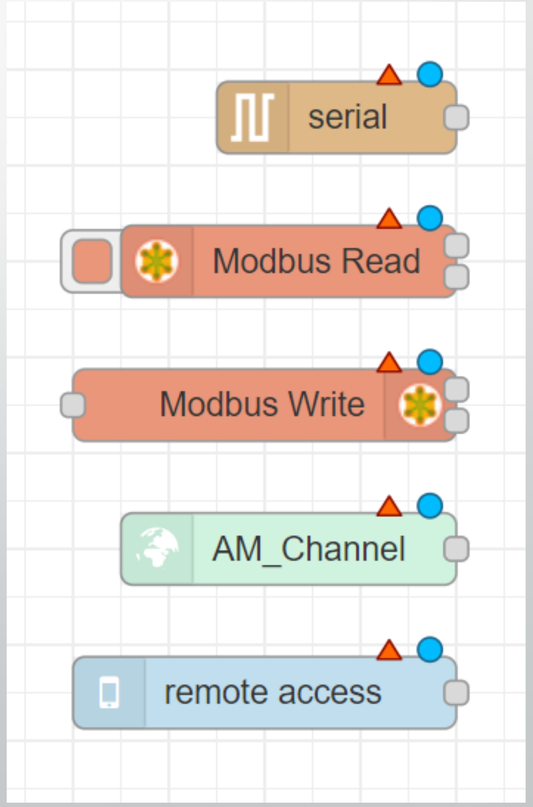
192.168.1.15

Node-Red

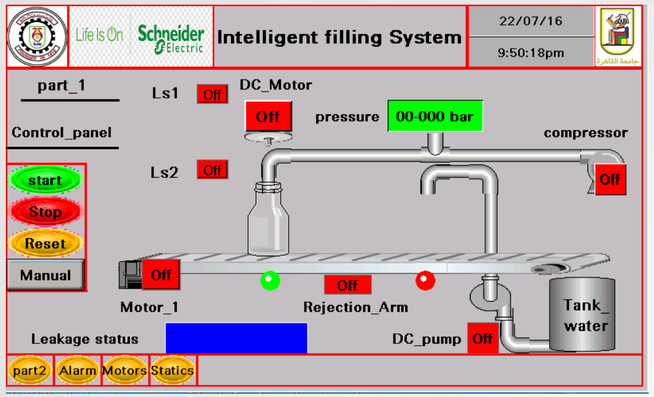
Definition

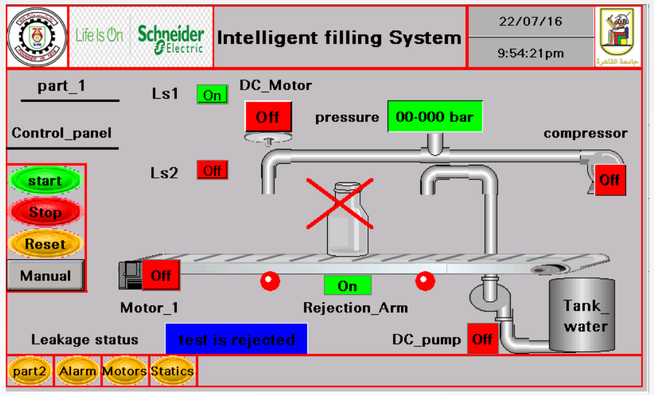
Free Application Enablement Platform (AEP)

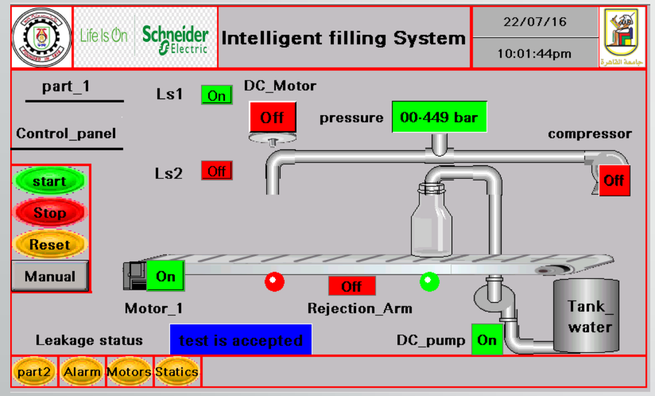
Usage

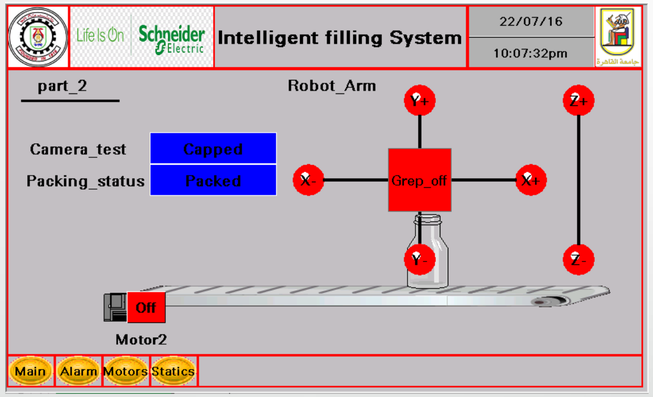


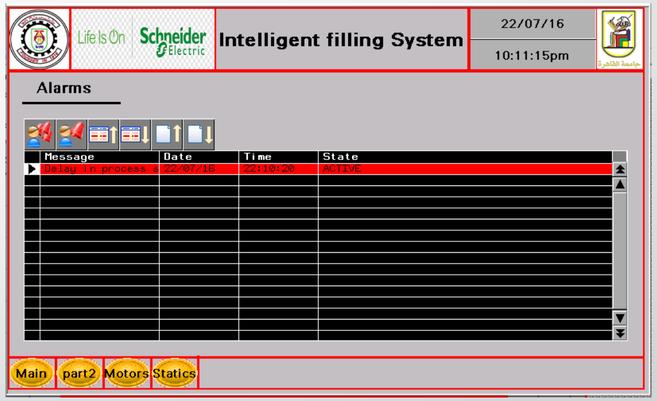
Visualization (HMI)

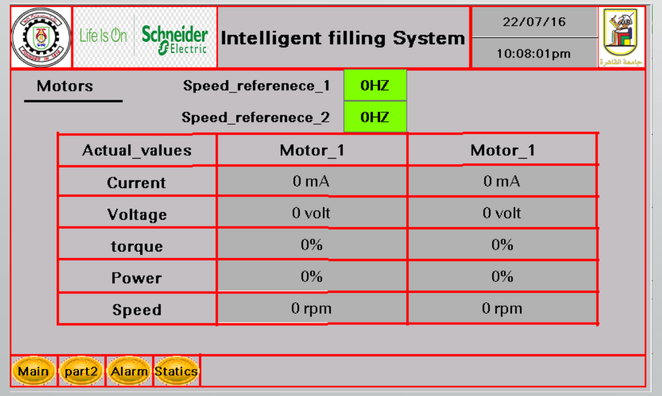


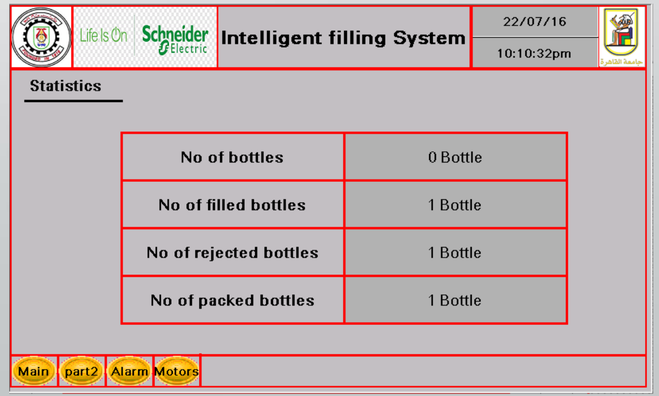




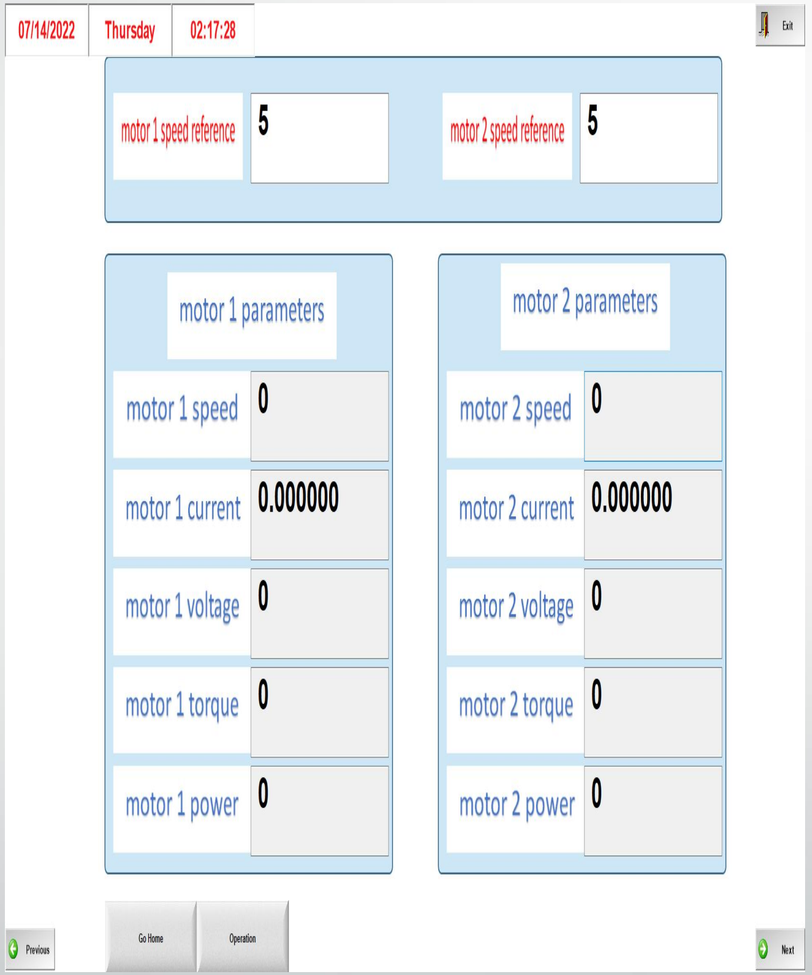


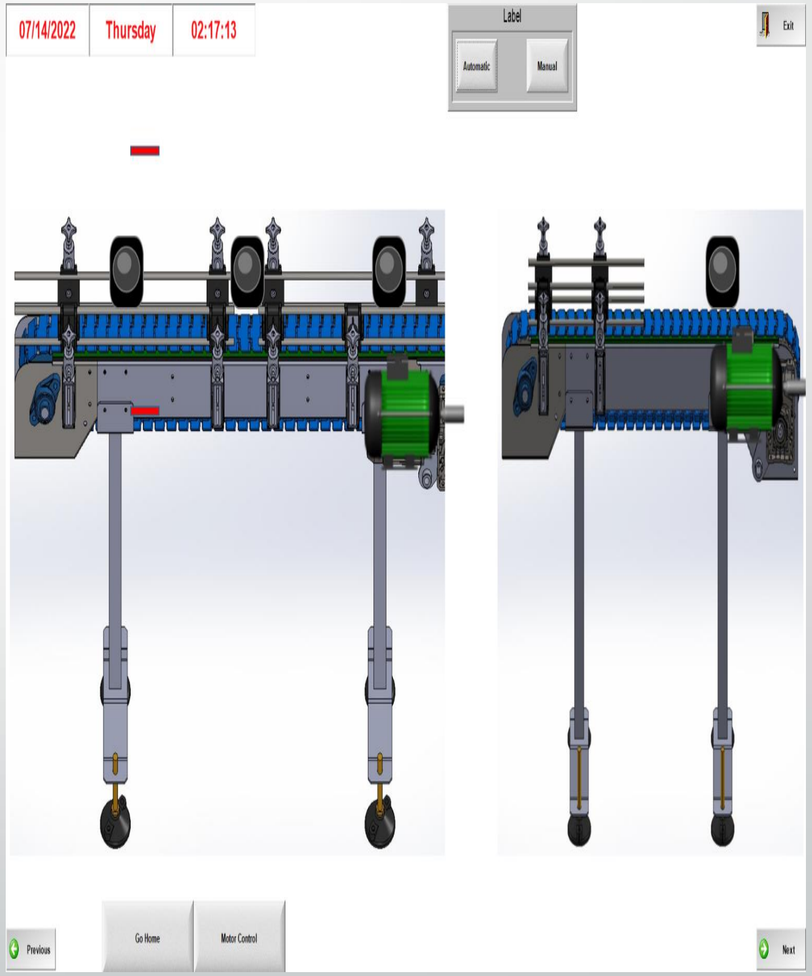




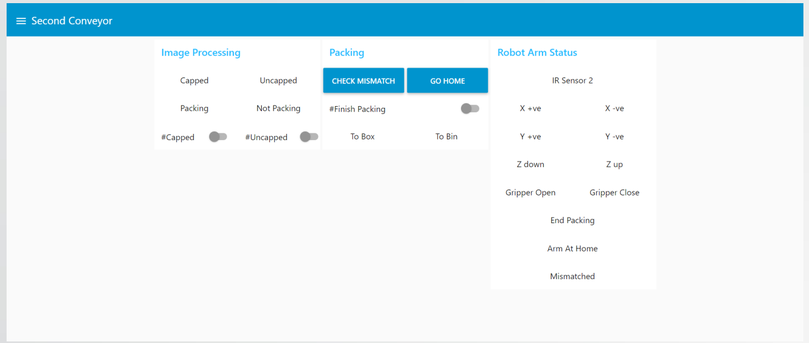


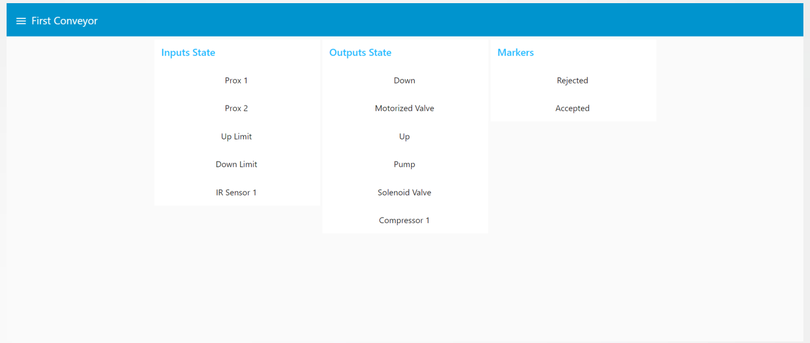
Visualization (SCADA)

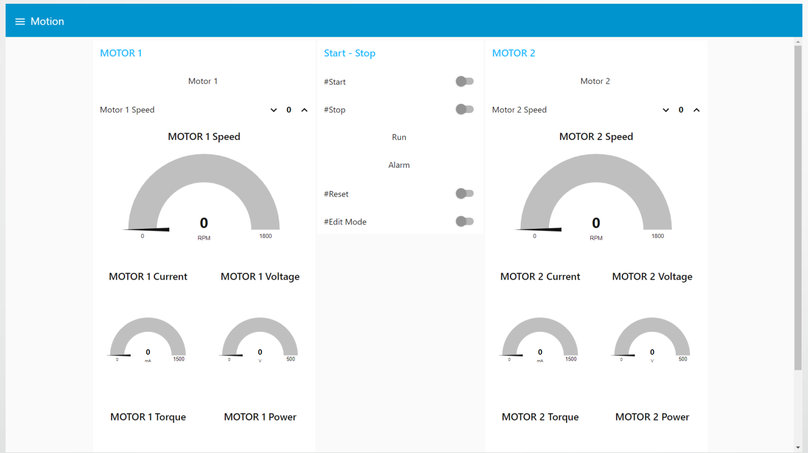


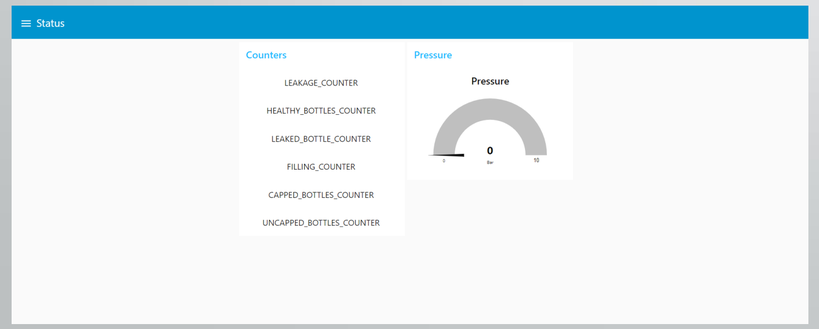


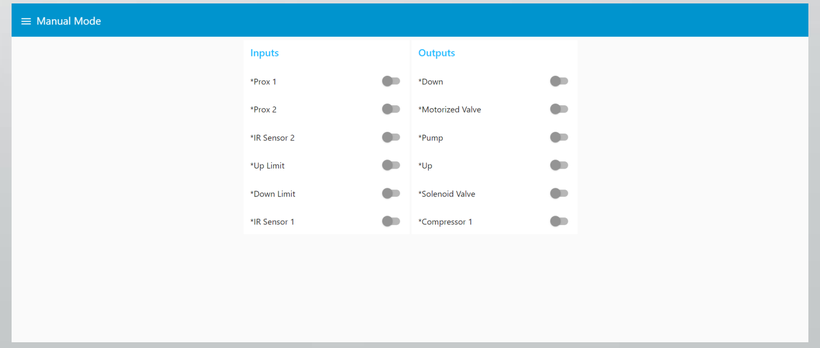
Visualization (Dashboard)





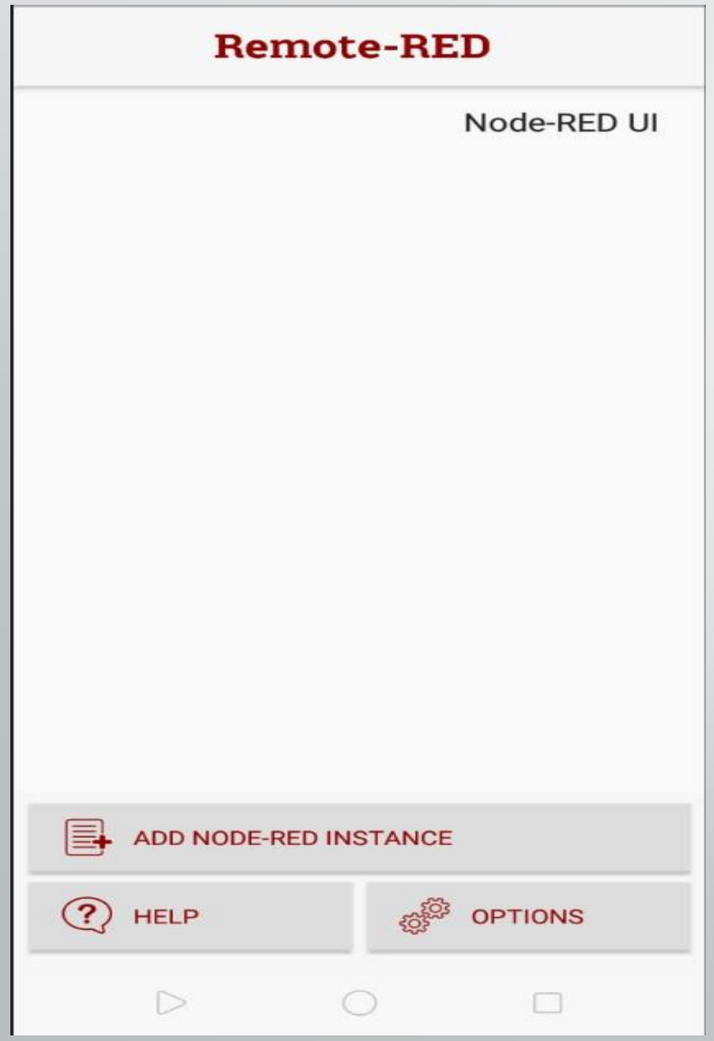


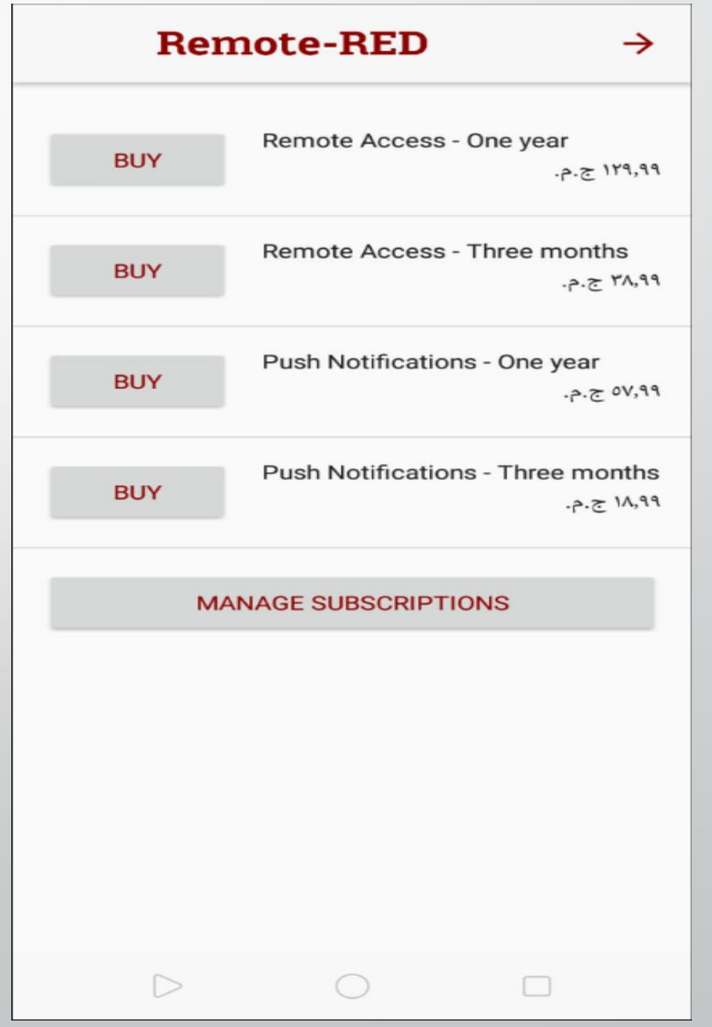




Remote Red App

To log in Dashboard you can use remote red mobile application (google play store) and remote red node





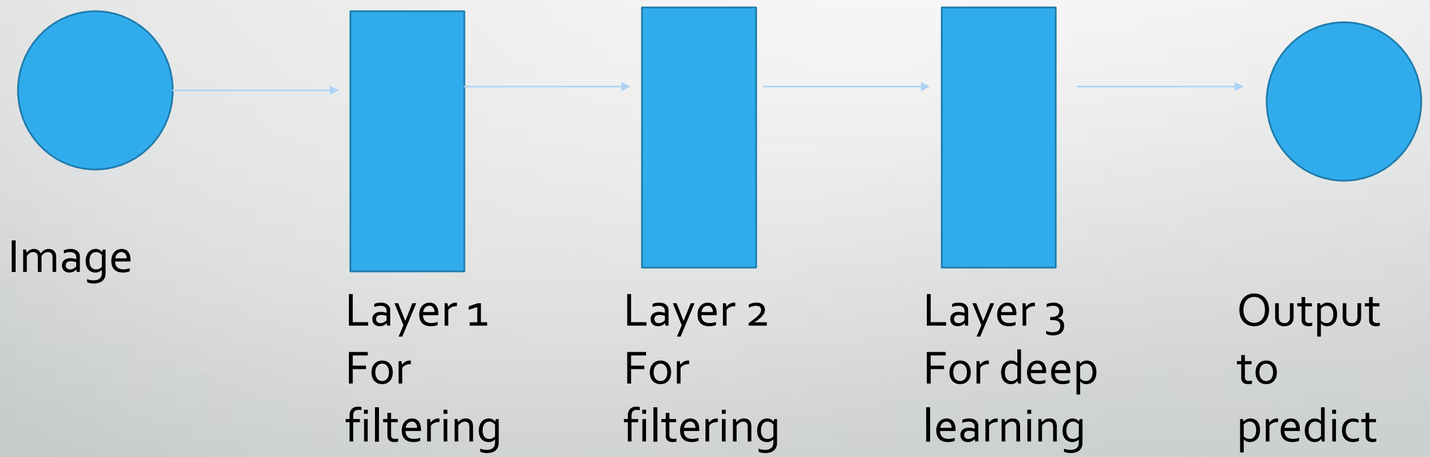
Project Processes

4. Product Quality System

- In this system , we check if the bottle is Capped or uncapped.
- TensorFlow , Keras and OpenCV libraries are used
- The whole Code is written By python
- YOLO library is used for object detection to detect bottle
- Model is build by Keras for deep learning

Building the Model

- We build the model by CNN (conventional neural networks)
- Process flow:



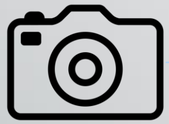
Building the Model

Layer (type)	Output Shape
conv2d (Conv2D)	(None, 148, 148, 32)
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)
conv2d_1 (Conv2D)	(None, 72, 72, 64)
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)
conv2d_2 (Conv2D)	(None, 34, 34, 64)
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 64)
flatten (Flatten)	(None, 18496)
dense (Dense)	(None, 128)
activation (Activation)	(None, 128)
dropout (Dropout)	(None, 128)
dense_1 (Dense)	(None, 1)
activation_1 (Activation)	(None, 1)

YOLO

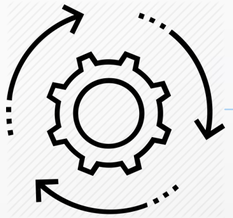
- It’s abbreviation you only look once
- It’s created by Google
- It contain the model that predicts the image according to the classes
- We used class bottle to detect our bottle

Code flow

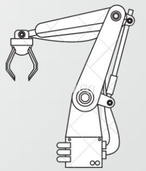


Open the camera

Bottle is detected by yolo And send the frame to model



Model is used to predict if bottle is capped or uncapped



Model sends to robot arm to go to packed box or unpacked box

- Python and Node-RED Connection
- Cartesian Robotic-Arm (Packing System)
- Mobile Application (Flutter)
- Public and Local Servers

Python and Node-RED Connection

Our Solution

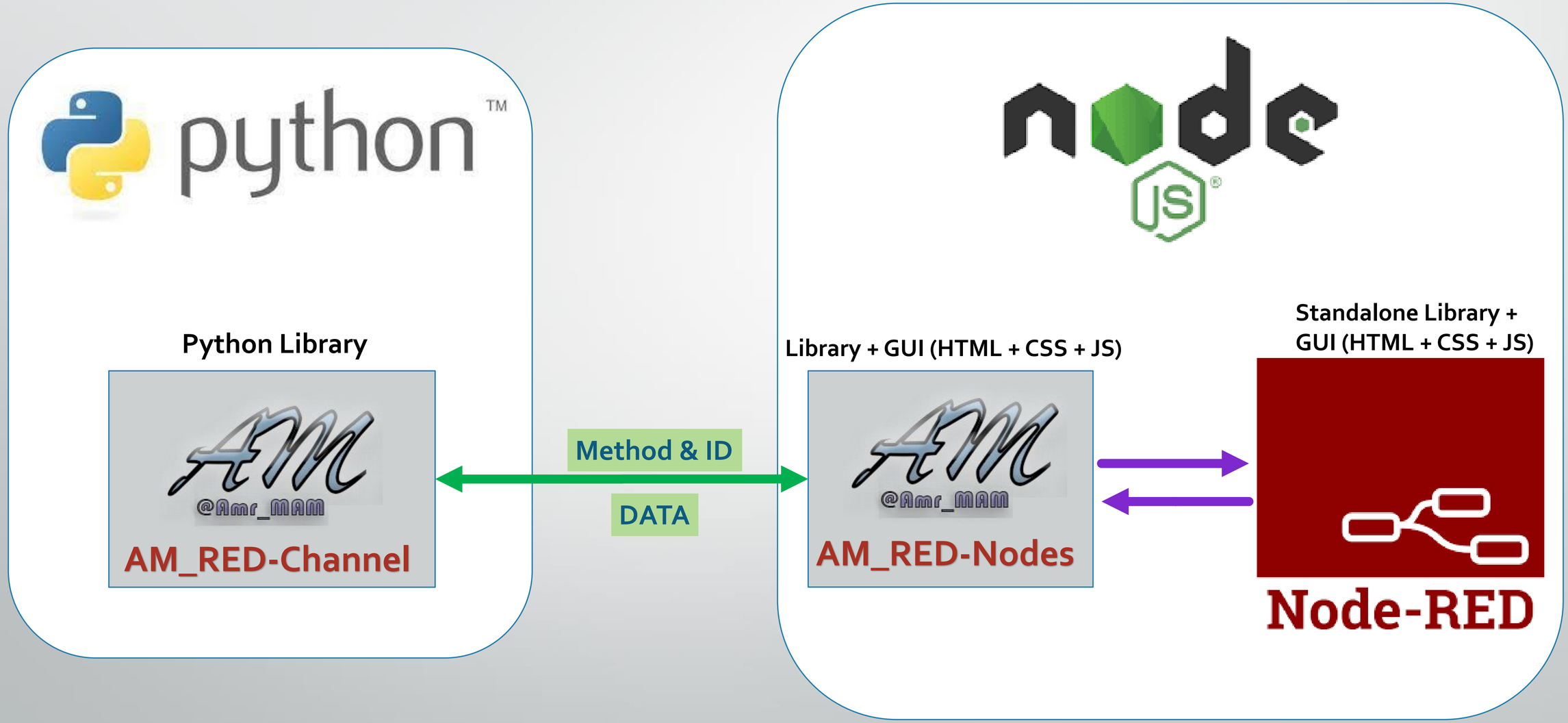
Python Library

am-red-channel

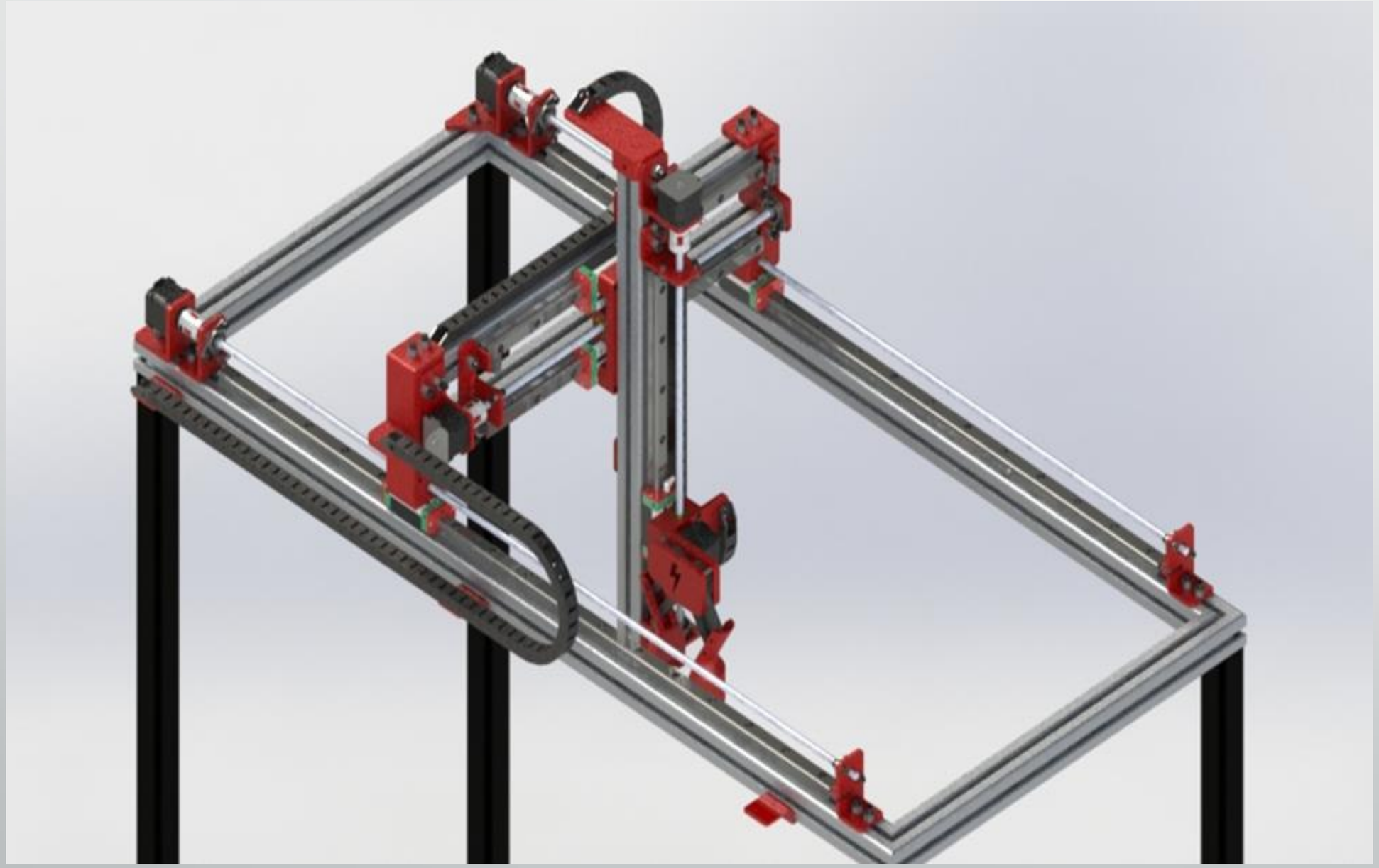
Node Library

am-red_nodes

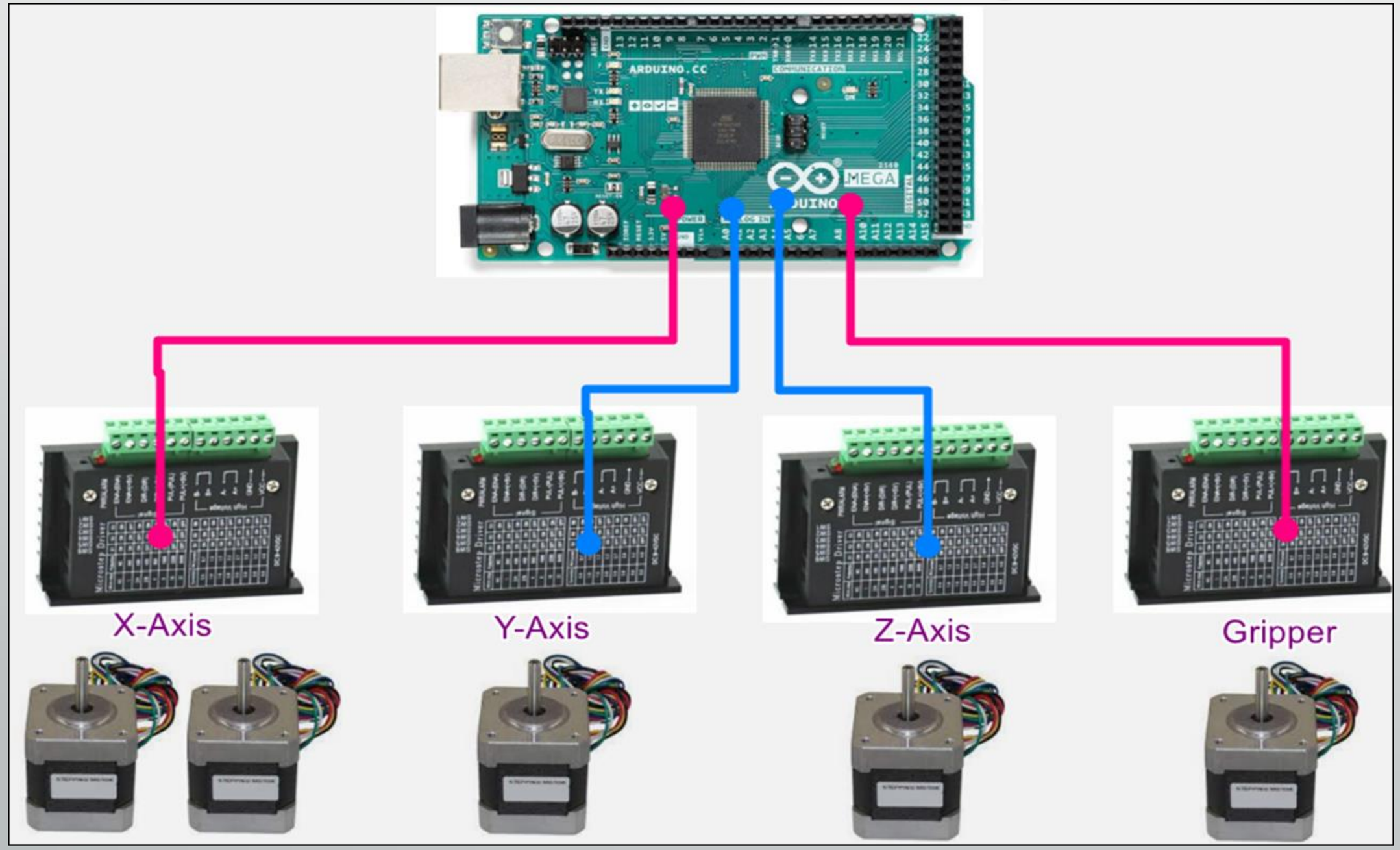
Python and Node-RED Connection



Cartesian Robotic-Arm (Packing System)



Cartesian Robotic-Arm (Packing System)



Cartesian Robotic-Arm (Packing System)

PLC

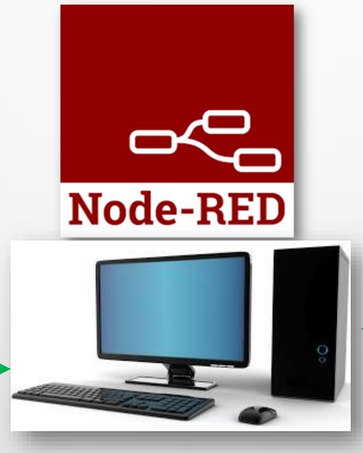
Indirect Connection

TCP/IP

HMI

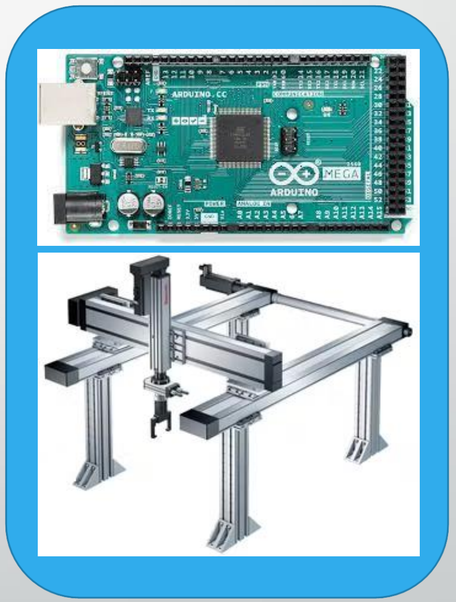
TCP/IP

TCP/IP



Serial COM

Packing System

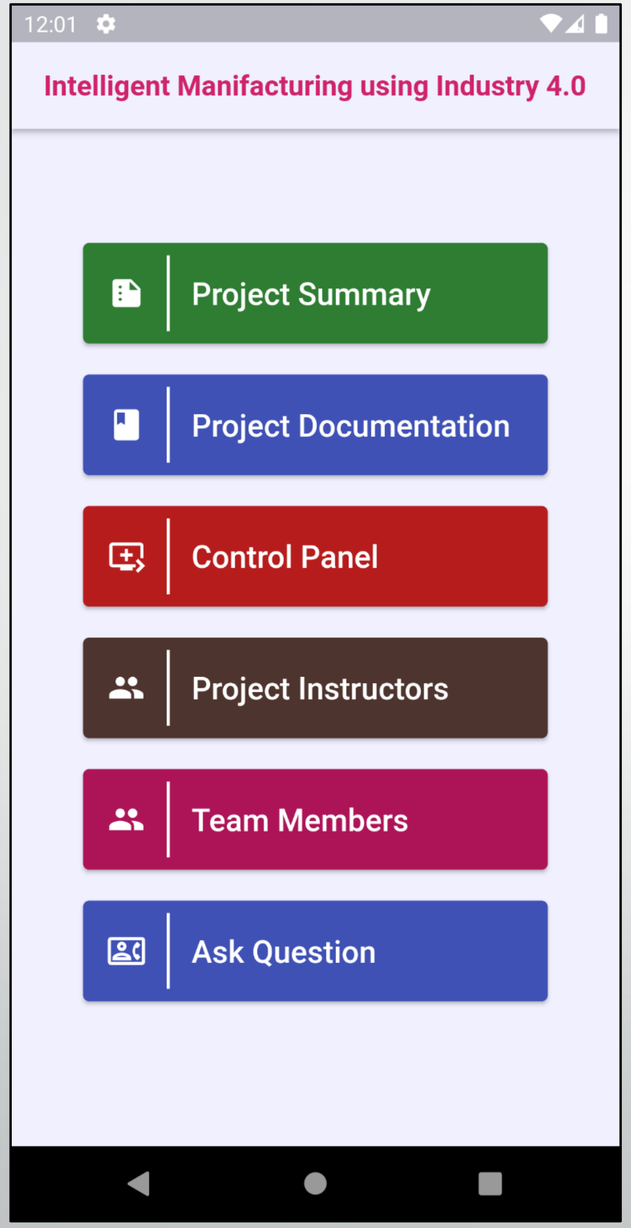


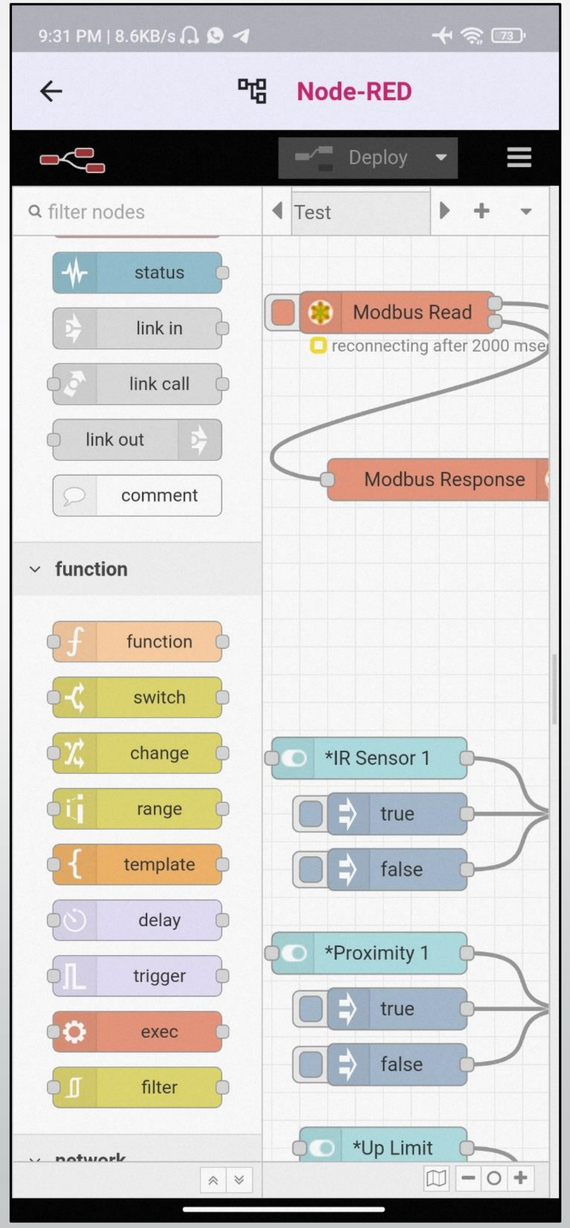
SCADA

TCP/IP

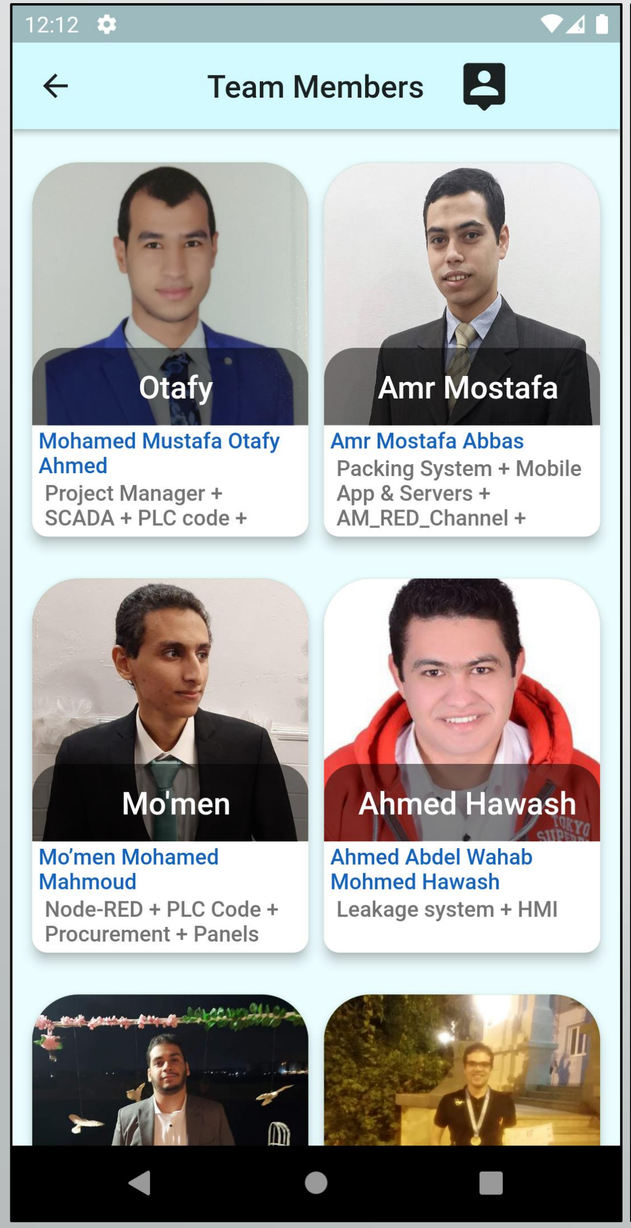
➢ Mobile Application (Flutter)

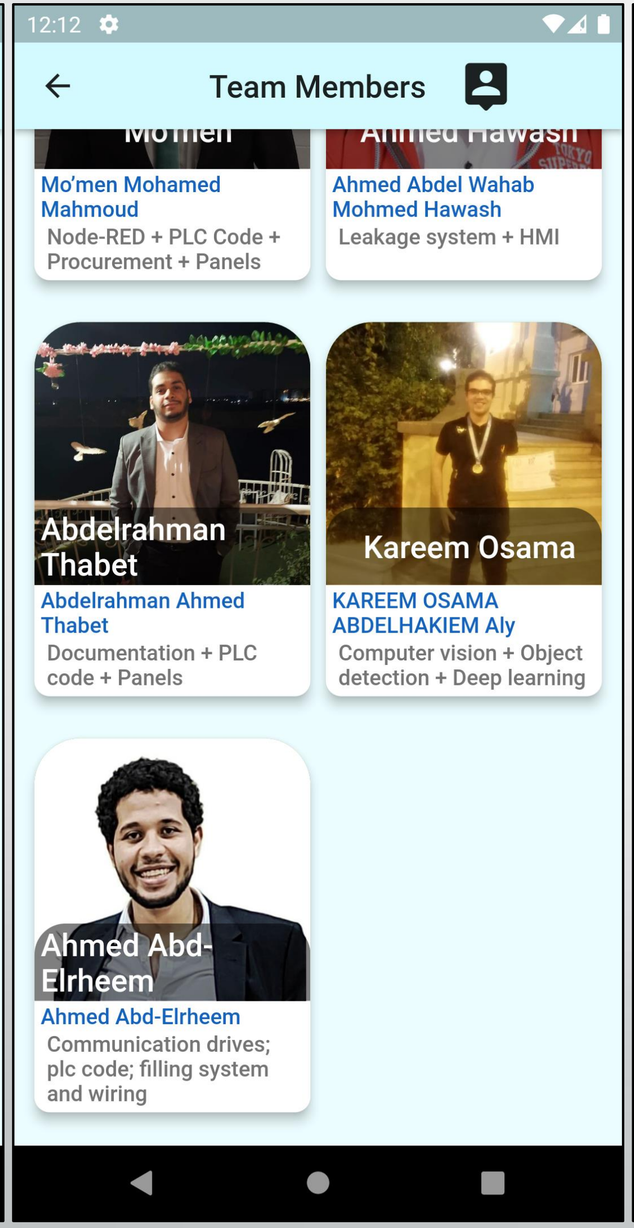
Mobile Application (Flutter)

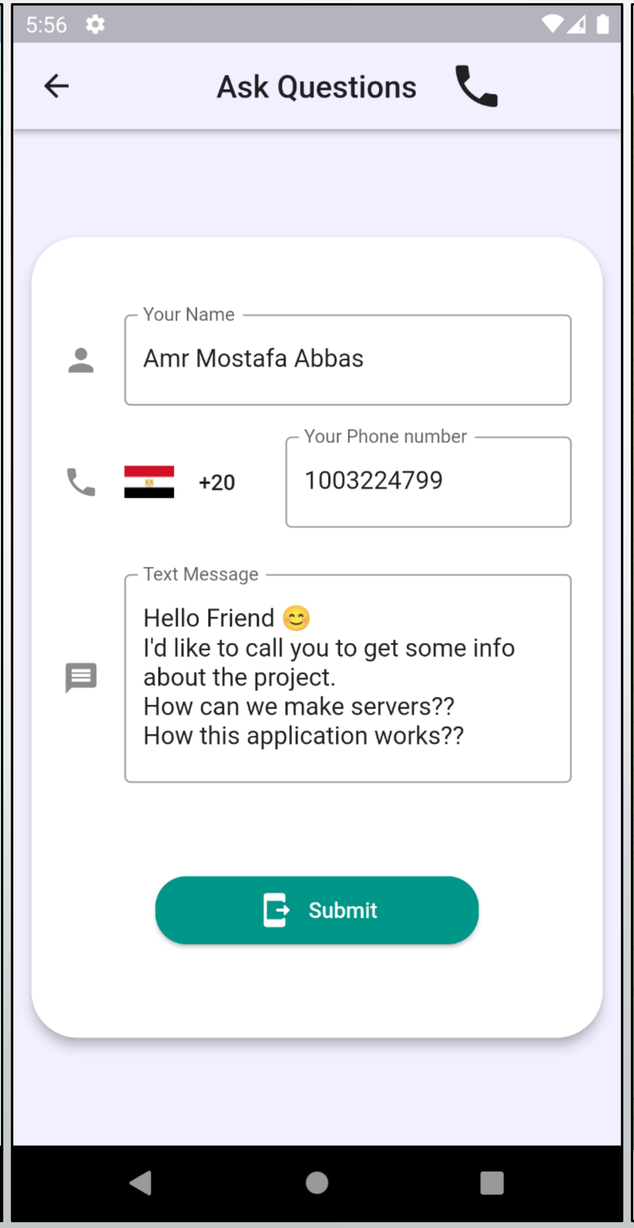


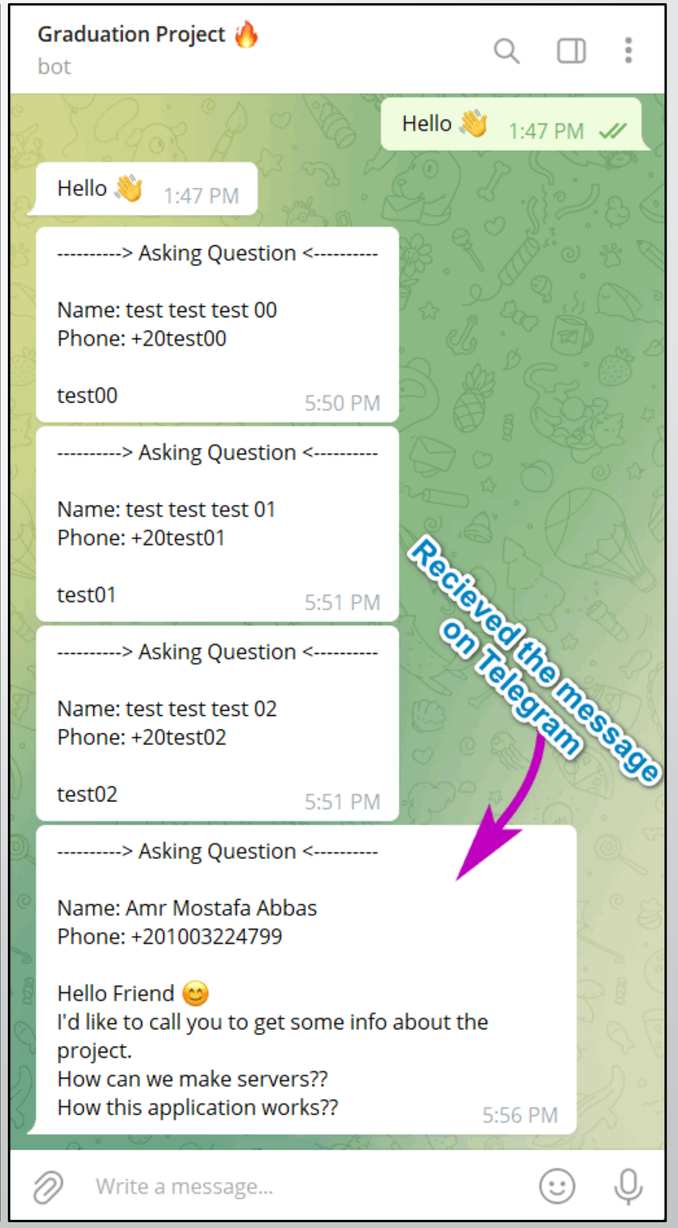


Mobile Application (Flutter)

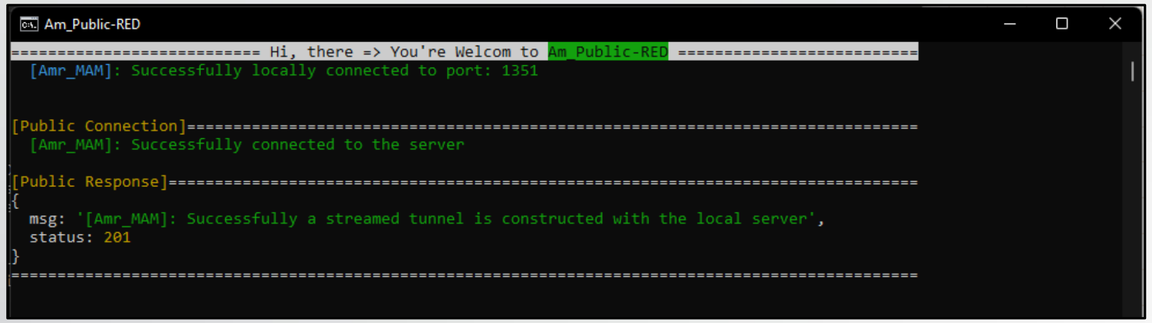




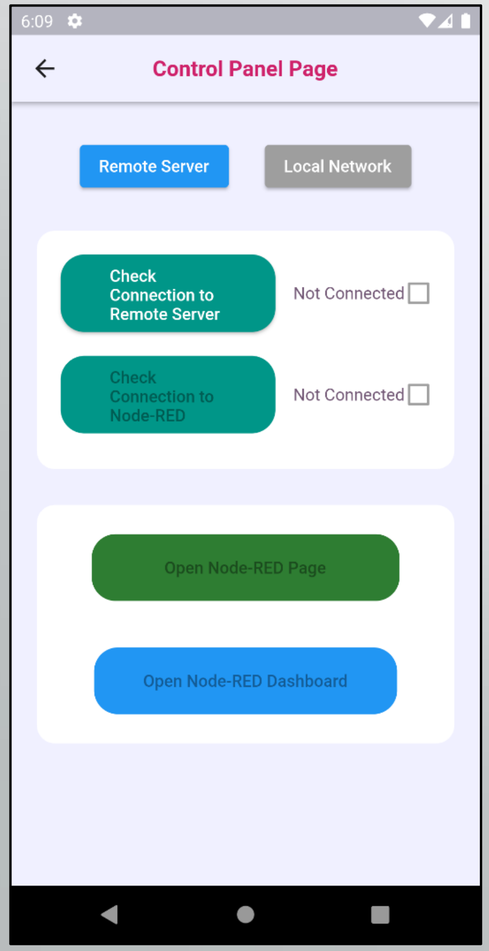


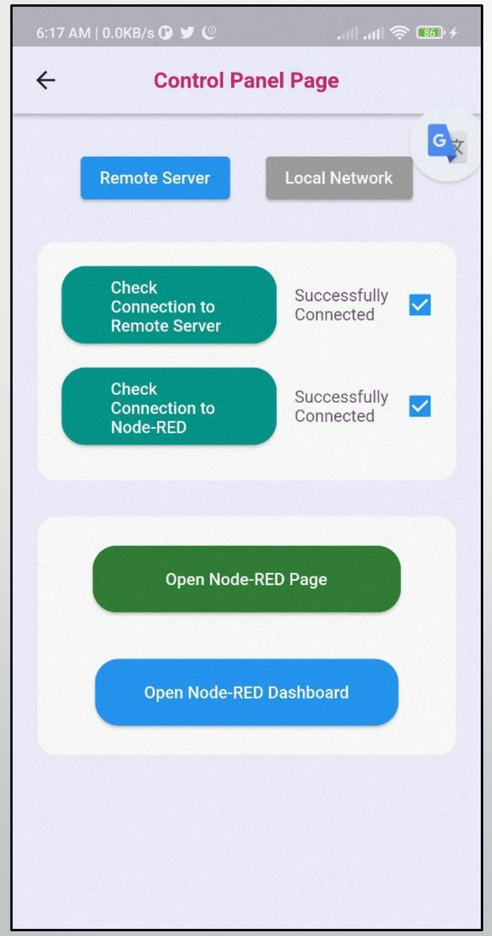


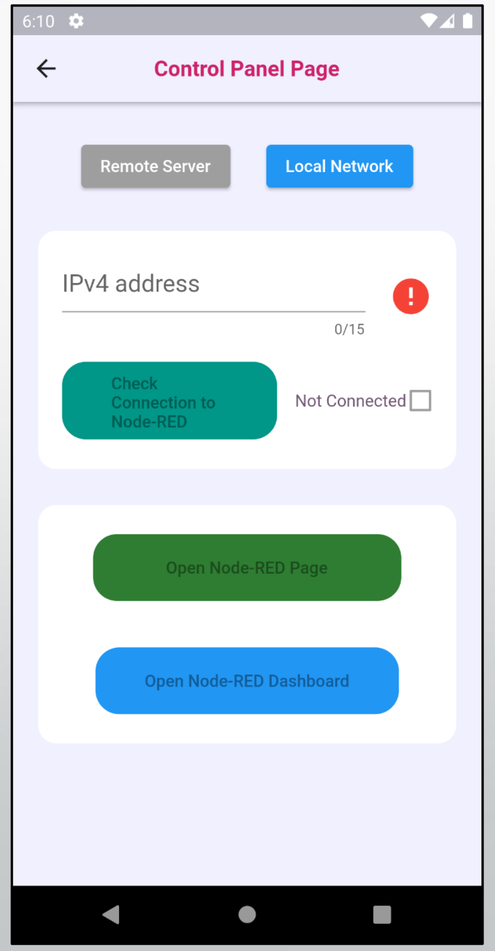
Mobile Application (Flutter)

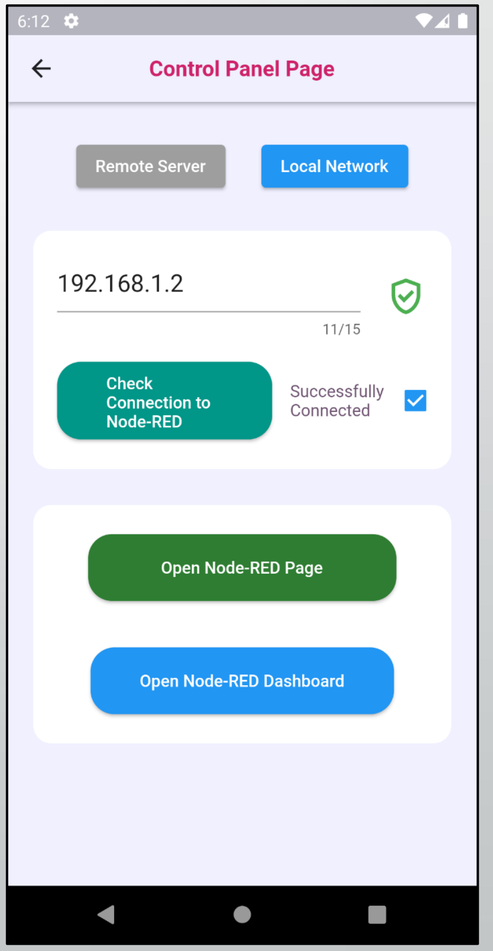


Local Connection









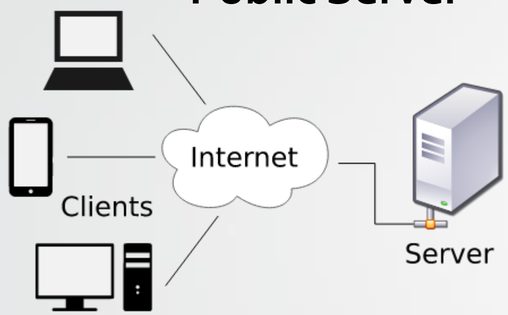
Mobile Application (Flutter)

Remote connection

Public Server

Public host in our case
https://am-graduation-apis.herokuapp.com/

Req, Res



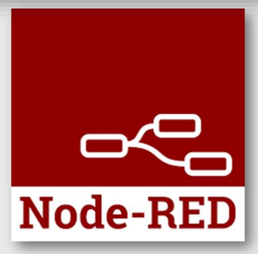
PC , Laptop or Raspberry

Local Server

Network Tunnel + Port Forwarding

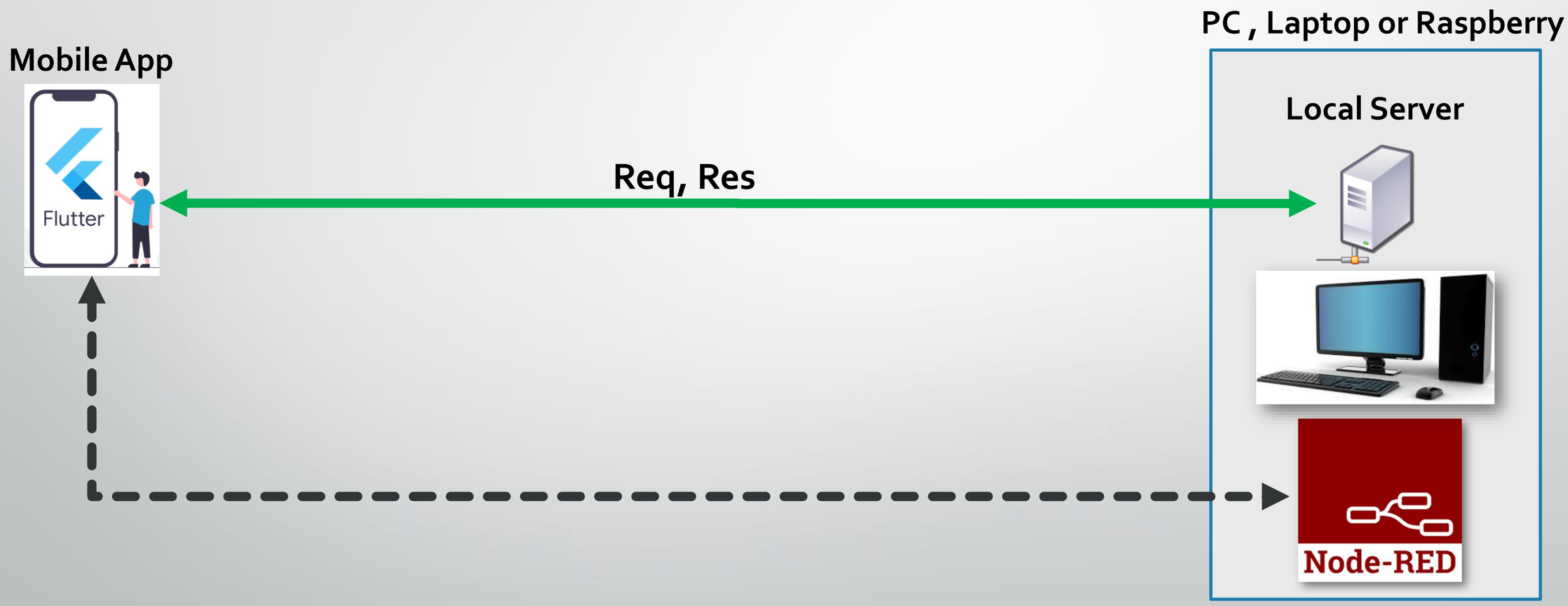
Mobile App

Indirect Req, Res



➢ Mobile Application (Flutter)

Local connection



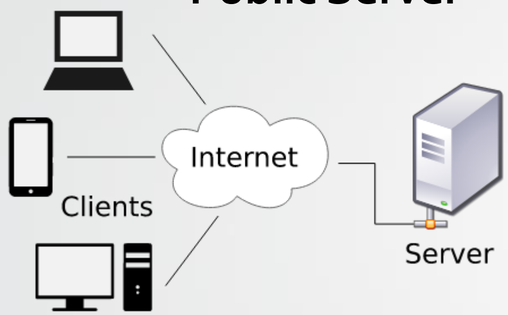
Mobile Application (Flutter)

Remote connection – Telegram

Public host in our case
https://am-graduation-apis.herokuapp.com/

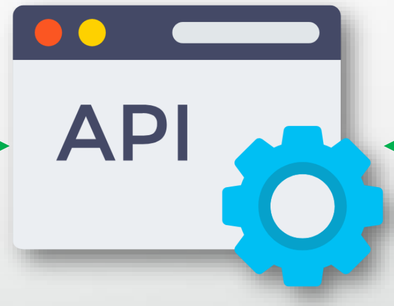
Public Server

Req, Res



Telegram Server APIs

Req, Res



Req, Res

Telegram App

Mobile App

Indirect Req, Res

Future improvements

- Replace control panel by a larger one
- Installing a mechanical track
- Show capping system to mechanical experts to assess the suitability of the design to perform the task
- Buying a New capper
- Transform the machine from semi-automatic state to a fully automatic state


ANY
QUESTIONS