|  |  |  |
| --- | --- | --- |
|  |  | Word Count: |
|  | File:Singapore Polytechnic logo.png - Wikipedia  EP0403: CREATING AN IOT PROJECT |  |
|  | Semester 2, 2021/2022 |  |

|  |  |  |
| --- | --- | --- |
|  | Prepared By: Yun Hong & Lutfi  Admin Number:1947557, 1946853 | Date: |

[Introduction 2](#_Toc1324627521)

[Persona and their problems 2](#_Toc1370053504)

[Problem Statement 2](#_Toc1444388042)

[Project Idea 2](#_Toc752496092)

[The Flow 2](#_Toc485859950)

[Connecting the devices 2](#_Toc1943829916)

[What is MQTT? 2](#_Toc2112732108)

[Why MQTT over serial connection? (Arduino to Node-RED) 2](#_Toc1516147739)

[The MQTT Network Diagram 2](#_Toc1730864820)

[Classification 2](#_Toc1856354062)

[Training 2](#_Toc1296883362)

[Connecting to the broker 2](#_Toc2058550970)

[Hardware Chosen 2](#_Toc1738546465)

[Bread Board 2](#_Toc651778577)

[Schematic 2](#_Toc859696287)

[PCB 2](#_Toc1388843938)

[Arduino 2](#_Toc1473765964)

[Connecting to the broker 2](#_Toc1822486669)

[The flow 2](#_Toc1045407489)

[The dashboard 2](#_Toc553160212)

[Notification 2](#_Toc891900630)

[What is InfluxDB? 2](#_Toc1379718207)

[Why InfluxDB over MySQL? 2](#_Toc723683607)

[The Dashboard 2](#_Toc1652144081)

[Security in IoT 2](#_Toc198999508)

[Evaluating the risks 2](#_Toc72529430)

[Steps to increase security 2](#_Toc1844626448)

[HTTP Ports 2](#_Toc1741123983)

[Authorization 2](#_Toc688812852)

[SSL(Certificates) 2](#_Toc1248482478)

[Token 2](#_Toc599597166)

[Our Security Feature 2](#_Toc1203441429)

[Personal Reflection 2](#_Toc1511131147)

[Yun Hong 2](#_Toc1619458919)

[Lutfi 2](#_Toc111996537)

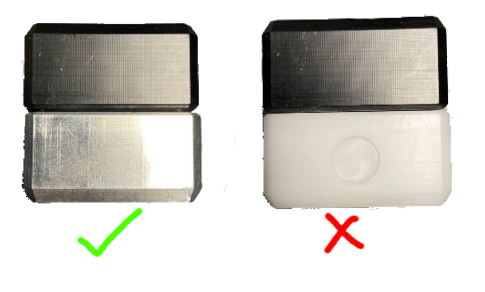
# Introduction

This report covers our mini project for the module, Creating an IoT (Internet of Things) project. We decided that our project will tackle the manufacturing industry and in particular the term i4.0.

## Persona and their problems

Our client, Mr. Tan Ah Beng, works in a company that produces cubes and he believes one of his machines can be improved.

This machine is supposed to produce a cube that combines a Black and Silver part to produce a cube. However, due to logistical error, a white part may be introduced into the system.



Currently there is a sensor on the machine to detect the cube color, however this process takes 30 seconds, and he believes an AI (Artificial Intelligence) can improve the detection speed.

## Problem Statement

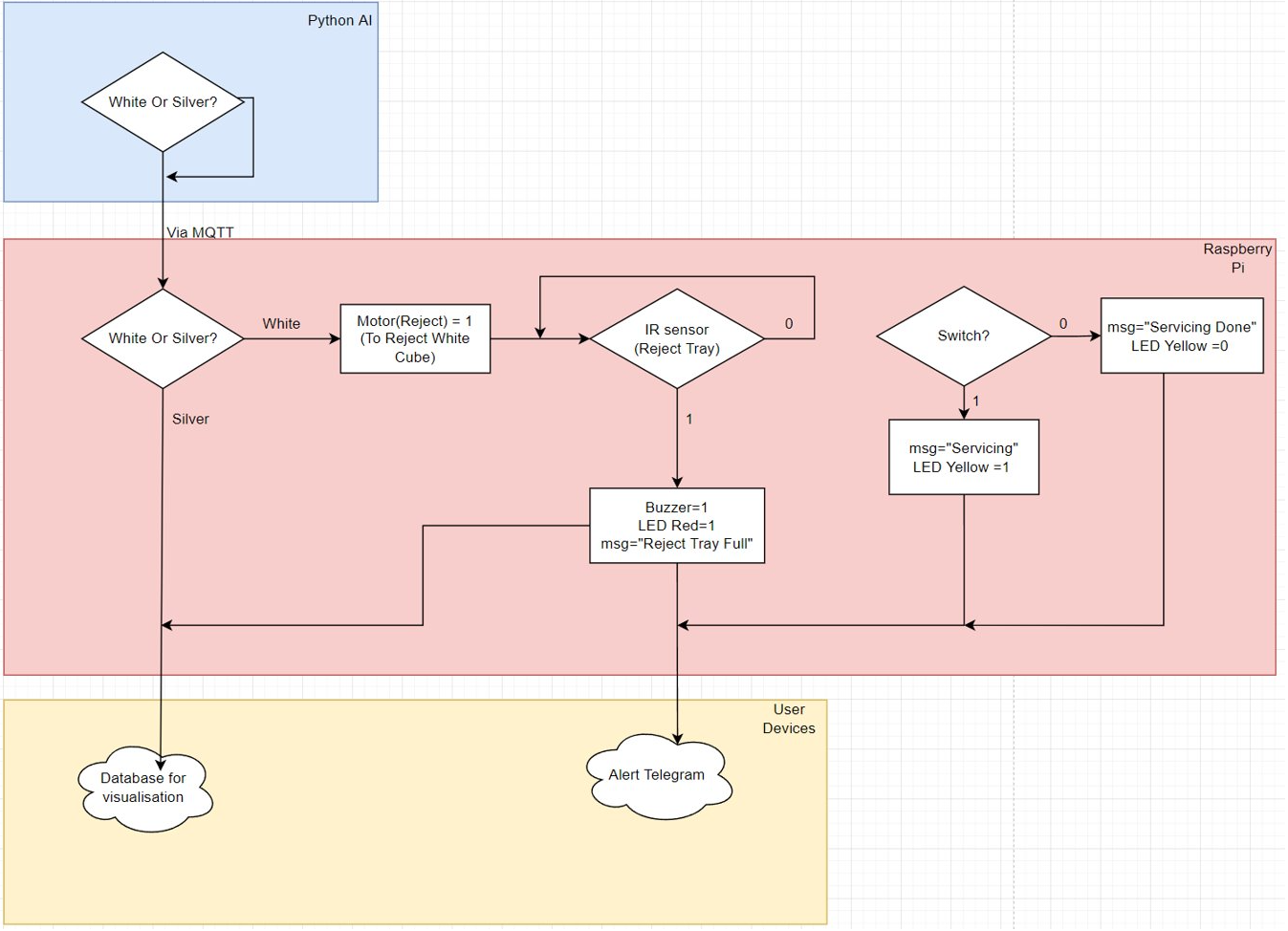
How to streamline process in i4.0 workflow?

# Project Idea

We have chosen to implement an IoT project to solve the problem.

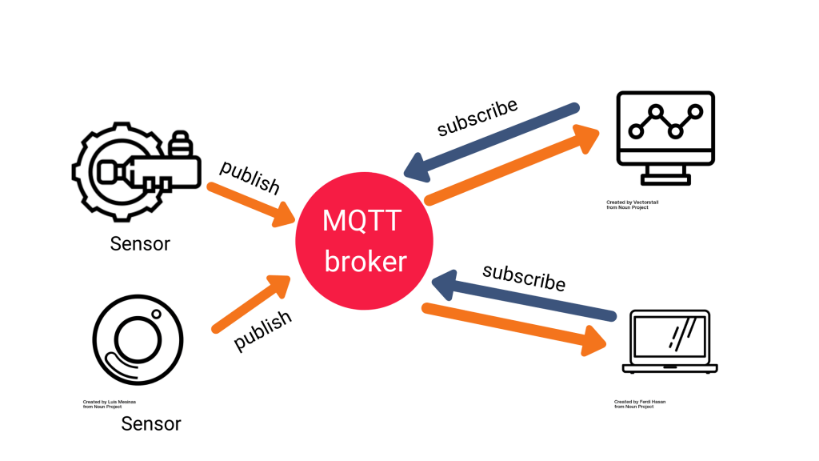
1. Camera is used to detect white and silver cubes; Ai is then used to classify between white and silver.
2. If the A.I (Artificial Intelligence) detects white cube, a motor will push the white cube into the reject tray.
3. The reject tray contains an Infared sensor that count the number of white cubes.
4. When there are too many white cubes, LED Red and Buzzer will turn on. It will also send a notification via Telegram.
5. A technician will then come by to remove the white cubes. He will also turn on a switch to show that the machine is being serviced. The servicing status will be sent via telegram.
6. The machine uptime/downtime and reject frequency will all be available via dashboard for anyone on the network.

## The Flow



# Connecting the devices

For connecting all the devices and programs together, we used MQTT. Here is an example of how MQTT works:



The MQTT broker was hosted on Node-RED

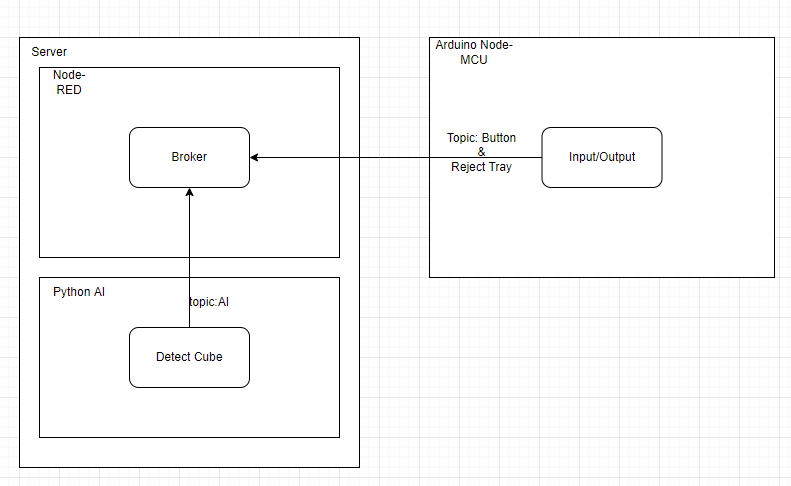
## What is MQTT?

MQTT is a lightweight, publish-subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP, however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT.

## Why MQTT over serial connection? (Arduino to Node-RED)

Arduino supports serial communication via the USB port however, it is a pain to work with. The serial connection is outdated and only supports one connection at a time.

## The MQTT Network Diagram



Artificial intelligence

Machines can learn from their mistakes, adapt to new inputs, and execute human like job thanks to artificial intelligence (AI). Deep learning is used in almost all AI applications. By processing enormous volumes of data and finding patterns in the data, computers may be trained to do specific tasks with these technologies.

## Classification

Classification is a type of supervised learning in which you teach a computer how to perform something with data that has previously been labeled by humans. This training set contains a defined number of labels of categories from which computers can learn. The system can categorize incoming data into pre- determined categories by identifying patterns in the training data, which is classification.

For our project the categories the computer must classify are white cube, silver cube and Unkown. Once the A.I recognize white cube, it will activate the motor pushing the cube into the tray.

Graphical user interface, text, application

Description automatically generated

## Training

To ensure that the classification is accurate, we use a total of almost 4000 photographs. We photographed the cube from several angles and in various lighting conditions; lighting was critical in the classification because the silver cube reflects light to the camera, causing the AI to misinterpret it as a white cube. We'll start testing the AI and troubleshooting based on the error once the training is complete and the accuracy is near to 1.

The first issue I encountered was that when there was no cube present in the camera, the AI mistook it for a white cube. To address this, we added a new category called Unkown, and because the project is in a control environment, we made our background red, placed red pictures in the unknown category, and trained the model.

A screenshot of a computer

Description automatically generated with low confidence

Diagram

Description automatically generated

Calendar

Description automatically generated with medium confidence

Text, letter

Description automatically generated

To increase the accuracy of the AI we change the epochs to 10 so the AI will train more.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The prediction number is group according to the file silver=0, Unkown=1 and white=2.

## Connecting to the broker

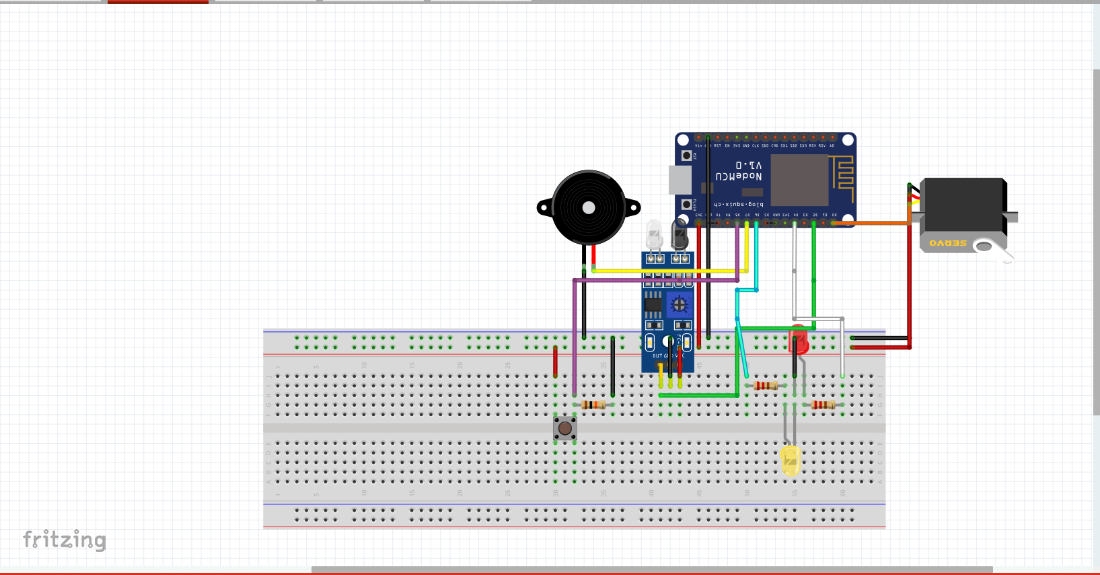
Instead of using the python script launcher available on Node-RED, we used MQTT to transmit data from the python script using the PubSub library.

PCB (Printed Circuit Board) Design

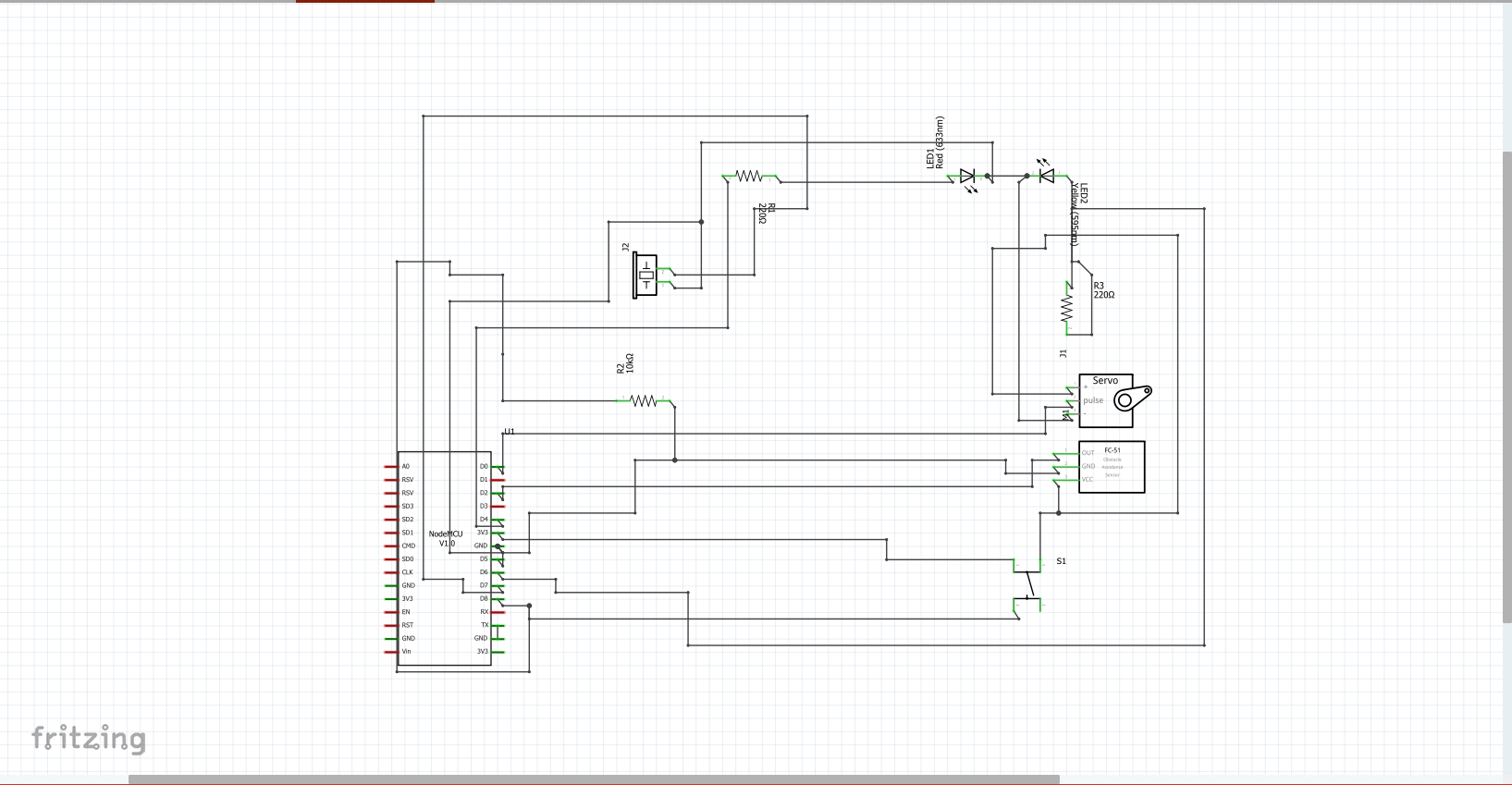
## Hardware Chosen

|  |  |
| --- | --- |
| Input/Output | Function |
| Button | The white cubes will subsequently be removed by a technician. In addition, he will turn on a button to indicate that the machine is being serviced. The status of the service will be communicated through telegram |
| Camera | A camera is used to detect white and silver cubes, and then A.I is used to classify them |
| Infrared sensor | To detect if the reject tray is full |
| Led Yellow | Will turn on when the button is pushed |
| Led Green | Will turn on when the infrared sensor detects the reject tray is full |
| Servo motor | Will push the white cube into the reject tray when the A.I detect white cube. |
| Buzzer | Will turn on when the infrared sensor detects the reject tray is full |

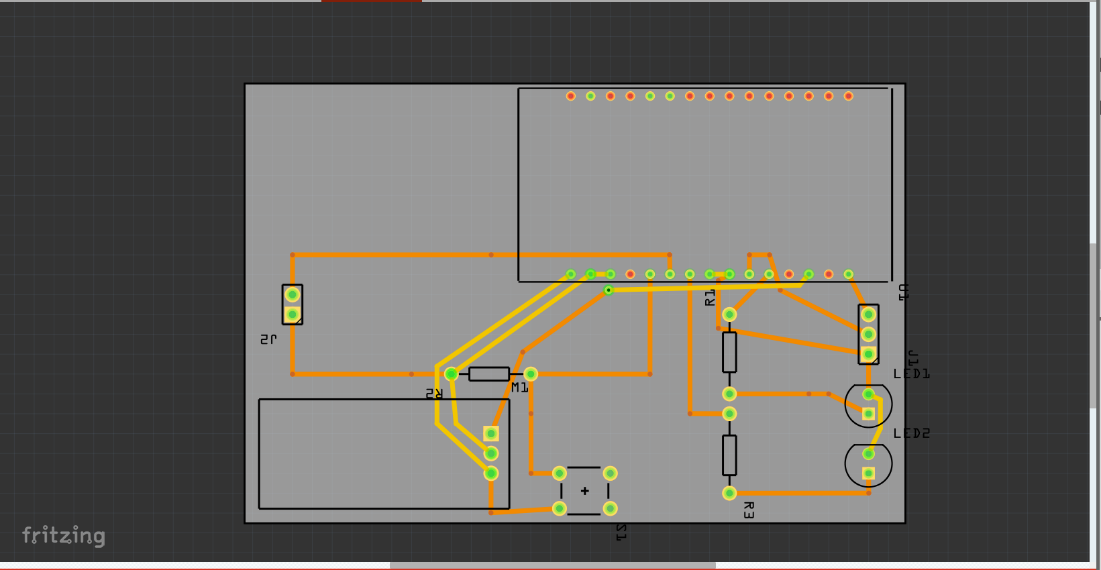
## Bread Board



## Schematic



## PCB



# Arduino

We chose to use a NoceMCU Arduino microcontroller instead of the traditional Arduino as it has an embedded ESP8266 on board. It also has all the I/O that we need, and it is inexpensive.

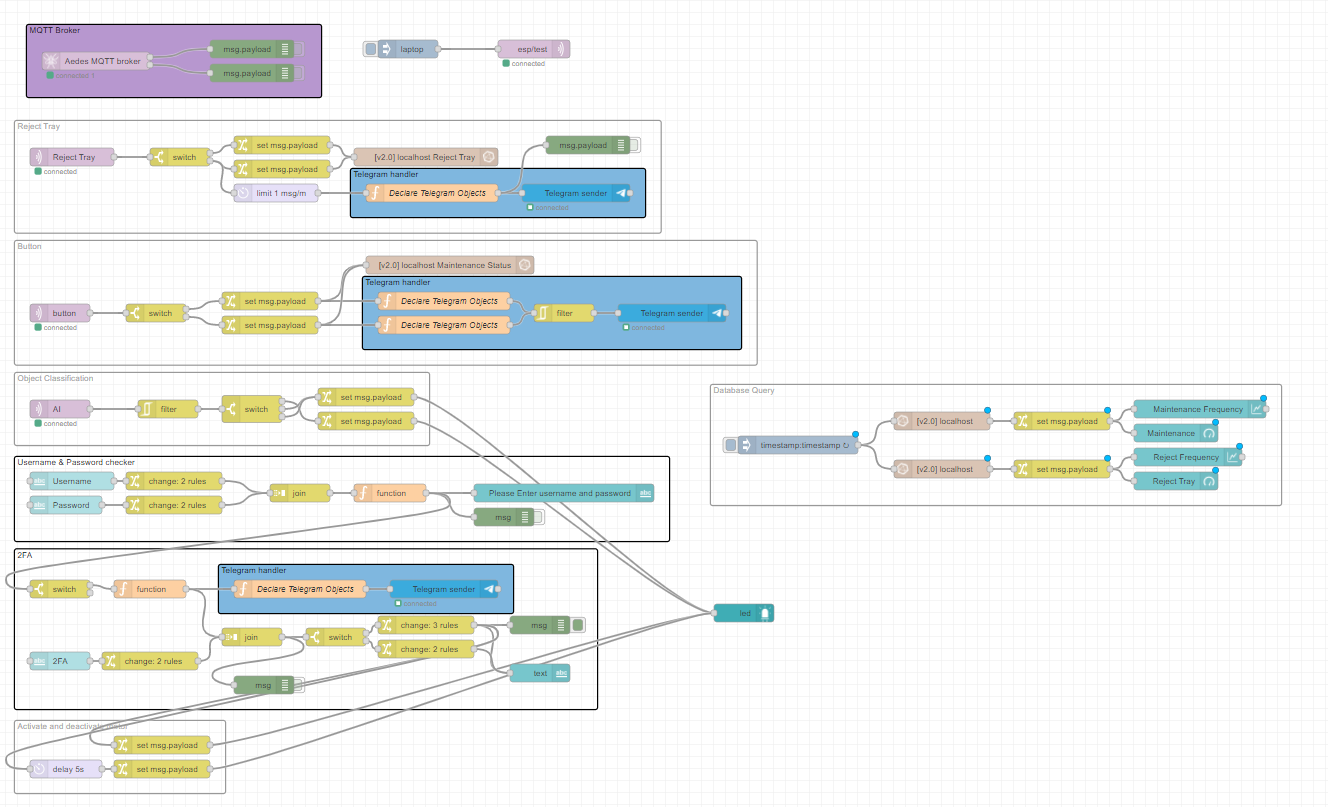
## Connecting to the broker

We used ESP8266WiFi library to connect ESP8266 to the Wi-Fi network. Then we used the PubSub library to connect to the broker

Node-RED

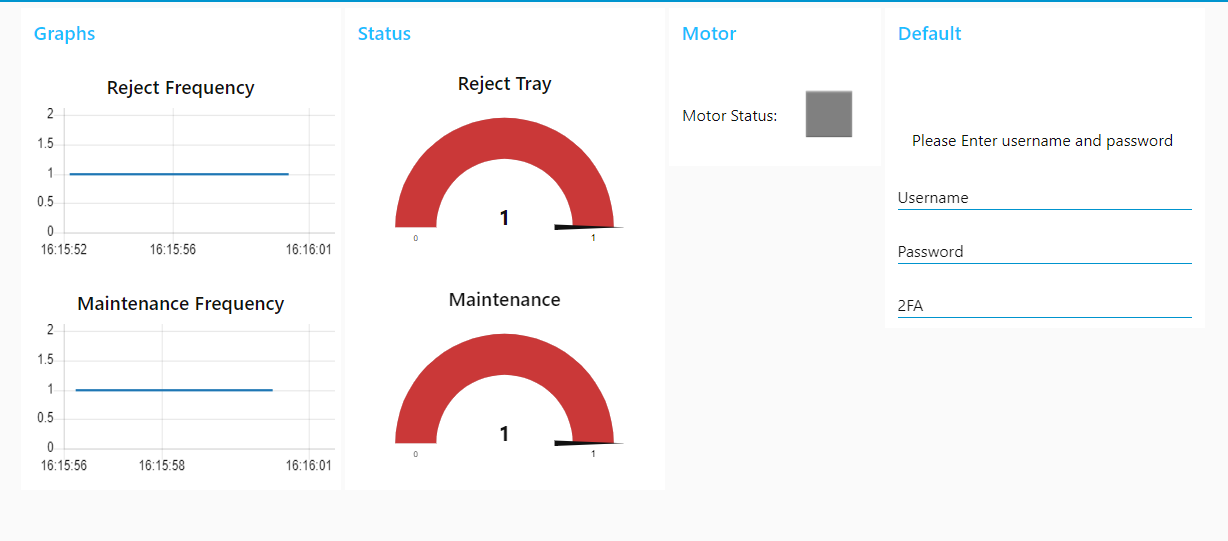
As most of the logic was processed on the Arduino, we only used Node-RED as a server to host the MQTT broker and send Data to InfluxDB.

## The flow



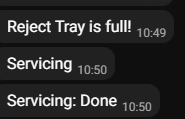
The flow has 6 features: The mqtt broker, The Reject tray Storage and notification, Servicing storage and notification, the AI Alert, and the security Function and the database query.

## The dashboard



## Notification

We have 3 notifications on telegram to alert the reject tray is full and the status of the machine servicing.



Client Server

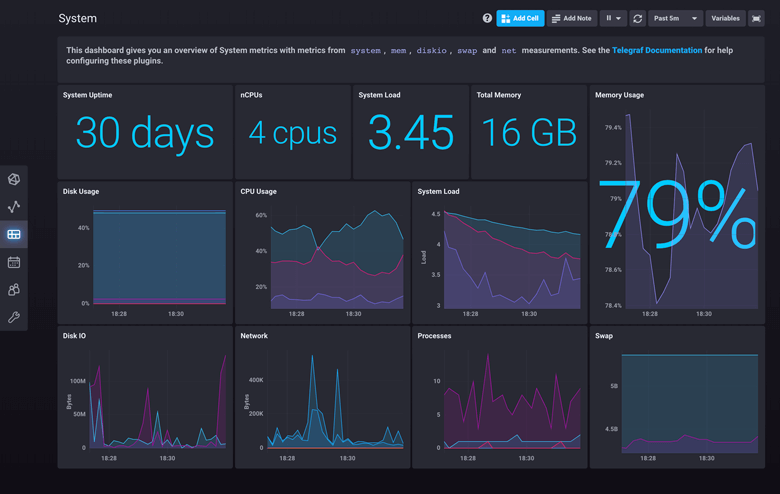
For the client server connections, we decided to use influxDB instead of MySQL.

## What is InfluxDB?

InfluxDB is an open-source time series database developed by the company InfluxData. It is written in the Go programming language for storage and retrieval of time series data in fields such as operations monitoring, application metrics, Internet of Things sensor data, and real-time analytics.

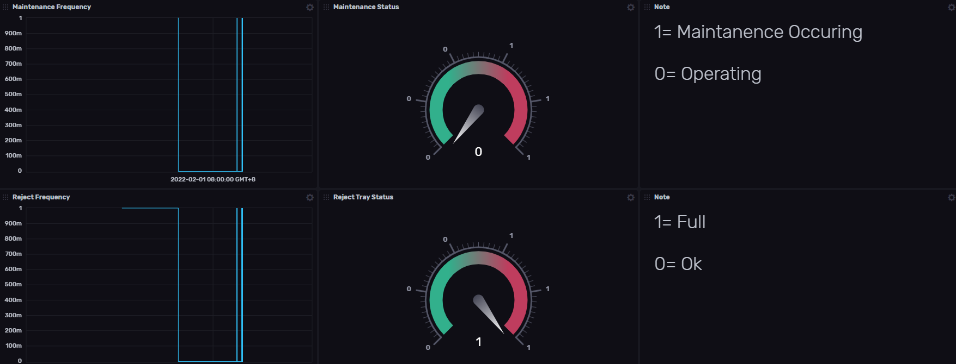
## Why InfluxDB over MySQL?

InfluxDB is by far the superior choice for our application as we are storing time-based data (e.g., “Temperature: 36° Time: 14:42”) Yes, MySQL is also able to store these values, however InfluxDB is optimized for our use case. It also has a visualization function built in.



The security is also stronger on influxDB as if a user would like to read/write data, they would need username, password, and an API (Application Program Interface) token, unlike MySQL which only requires username and password.

## The Dashboard



# Security in IoT

As the project uses network connection, different software and different communication protocols, there are many vulnerabilities.

## Evaluating the risks

OWASP is a foundation founded to increase the security of software and IoT devices. We can follow their [Threat Modeling Process](https://owasp.org/www-community/Threat_Modeling_Process#step-2-determine-and-rank-threats) to evaluate our project risk and respond to them accordingly.

## Steps to increase security

### HTTP Ports

The ports used in the project were 1880 for Node-RED and 8086 for InfluxDB. To increase security, one should block all the ports except for these few ports. They can also opt for enabling HTTPS as HTTPS can verify the authenticity of the server client connection

### Authorization

MQTT and influxDB has an Authorization system where you would need to have a username to access the data. Without the username and password, one would be unable to publish/subscribe to the broker.

Node-RED can also be configured to have an authorization system/

### SSL(Certificates)

MQTT uses TLS or SSL (Secure Sockets Layer) for its communication. For a more secure connection, one should use the SSL which requires a certificate to be generated and used for connection. Without it, one would be unable to publish/subscribe to the broker.

Node-RED can also be configured to use HTTPS instead of HTTP but requires two certificates to be generated

### Token

InfluxDB needs a token to access the information. One should keep this safe to deter attackers.

## Our Security Feature

We have a security feature that activates the motor only when the correct username and password is entered with a correct 2FA key that is sent to telegram.



# Personal Reflection

## Yun Hong

This module was fun and enjoyable, I learned a lot about artificial intelligence, and I have a deeper understanding of how artificial intelligence works with classification of cube and how IOT security plays a huge part in protecting our information on the internet.

I would like to thank the lecturer and my partner Lutfi for helping me out when I face some problems.

## Lutfi

I came into this module thinking that it was easy as I had taken all the previous IoT modules. However, I was taken aback by the density of this module. It really covers the IoT aspect of using multiple devices to communicate unlike the previous modules, which just used one device to read, output and interact with the user.

I have learnt a lot about what IoT and AI can do for us and now I am more certain about what career path to embark on in the future.