Final Presentation

Oriol Alàs, Quim Picó, Sergi Simón

TABLE OF CONTENTS

01

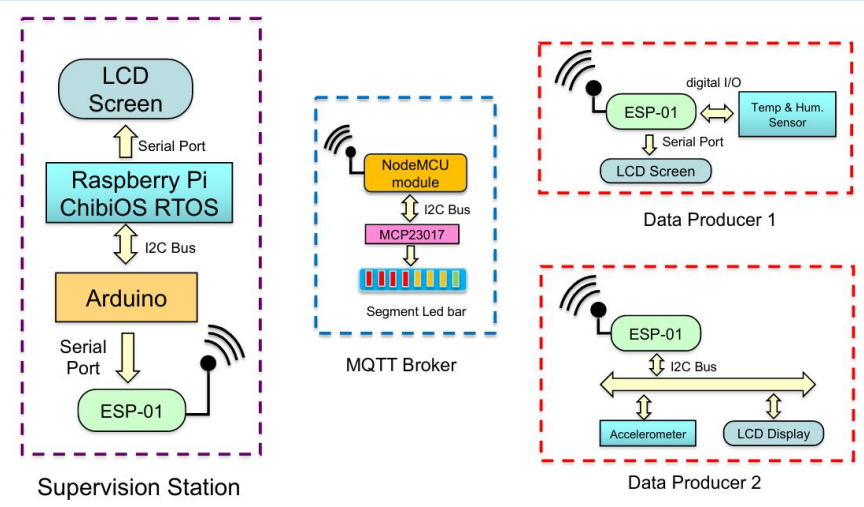
Introduction of the Project

Project Statement

- Manage and supervise wind turbines generator farms using embedded components.
- The prototype is composed by multiple devices:
  - Supervision station
  - Data providers
  - MQTT Broker

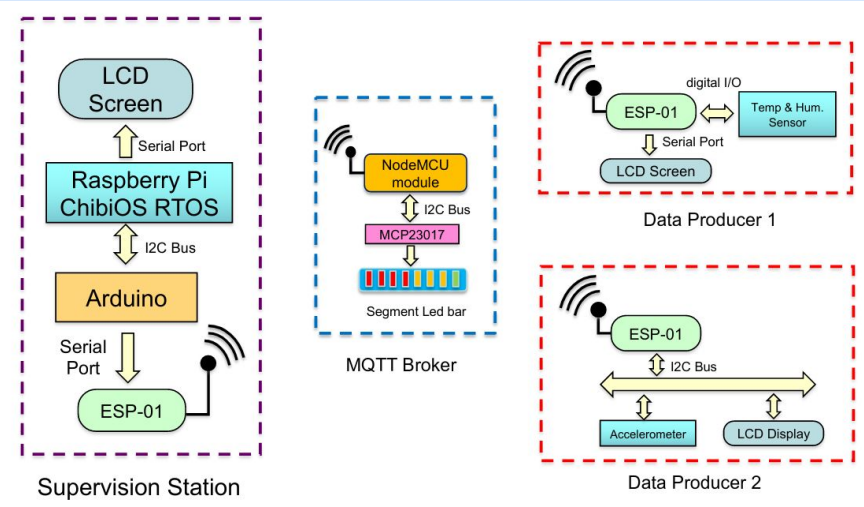
Project Requirements

- There is just only one Supervision Station for the whole WTGF.



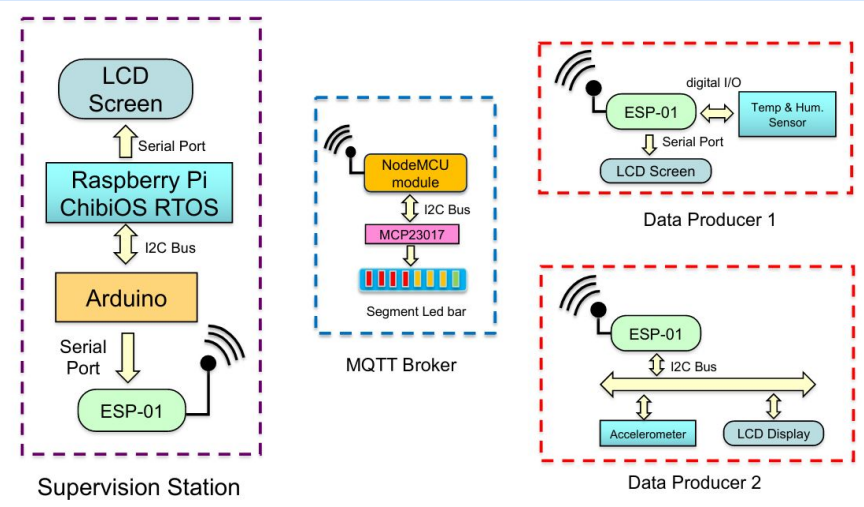
Project Requirements

- All sensors are placed far from the Supervision Station.



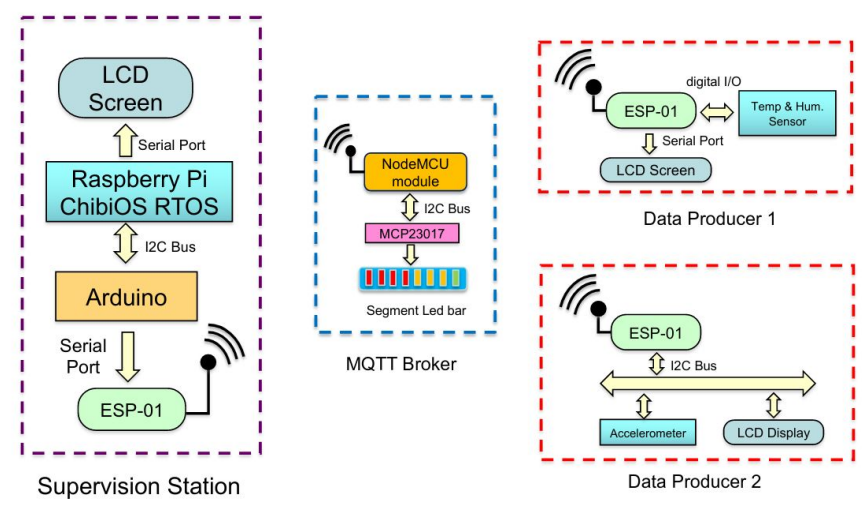
Project Requirements

- Sensor measurements must be obtained in Real-Time.



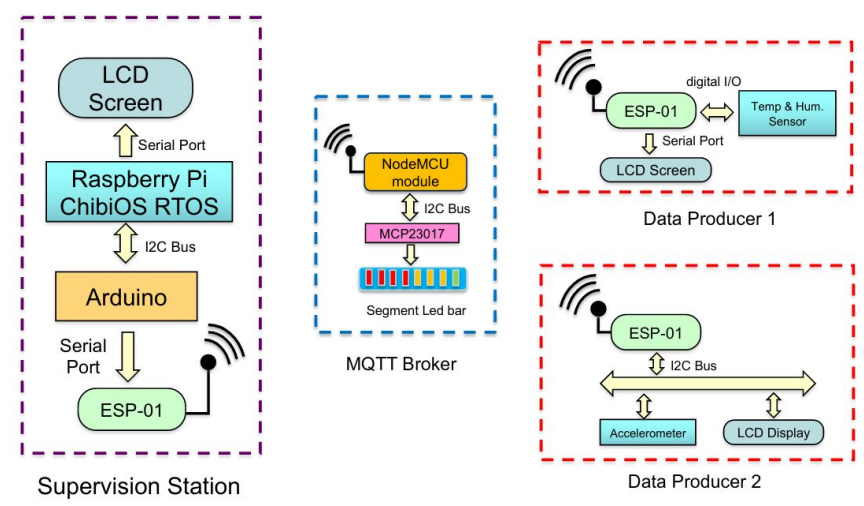
Project Requirements

- Sensors are controlled by the Data providers and they will show the current sensor readings.



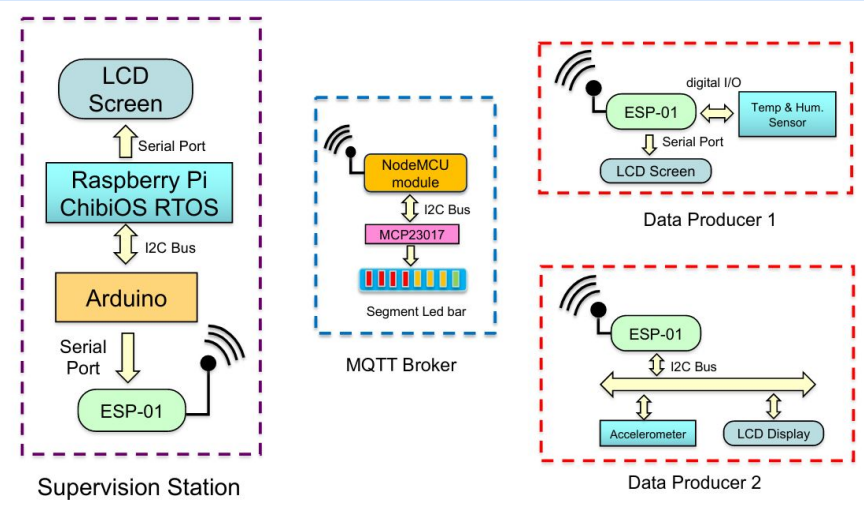
Project Requirements

- Whole WTGF data is transferred to the Supervision Station through a MQTT broker.



Project Requirements

- The MQTT broker will show graphically the current number of subscribers.



User Stories

1. It is necessary to obtain the temperature and humidity at the top of the WTG
2. It is necessary to determine the WTG tower movement
3.
4.
5.

User Stories

1. It is necessary to obtain the temperature and humidity at the top of the WTG
2. It is necessary to determine the WTG tower movement
3. Sensor data must be must be sent to a Supervision Station
4. MQTT broker must show the current number of connections
5.

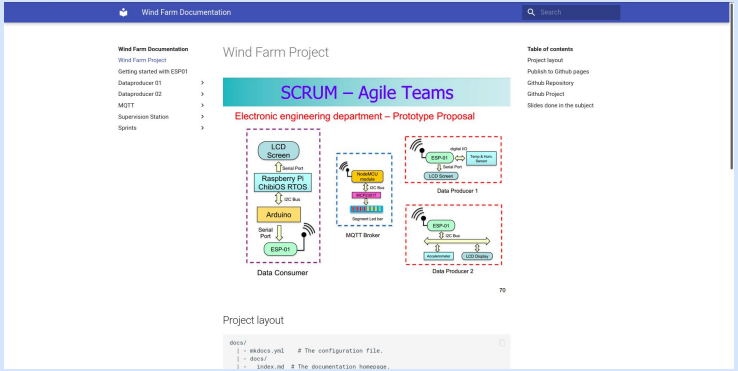
User Stories

1. It is necessary to obtain the temperature and humidity at the top of the WTG
2. It is necessary to determine the WTG tower movement
3. Sensor data must be must be sent to a Supervision Station
4. MQTT broker must show the current number of connections
5. Maintenance manager requires graphically representation of the obtained sensor data, within the last 24h, in the Supervision Station.

Project Management

GitHub Projects

- Scrum agile methodology.
- Project splitted into three sprints time period.
- MkDocs for documentation.



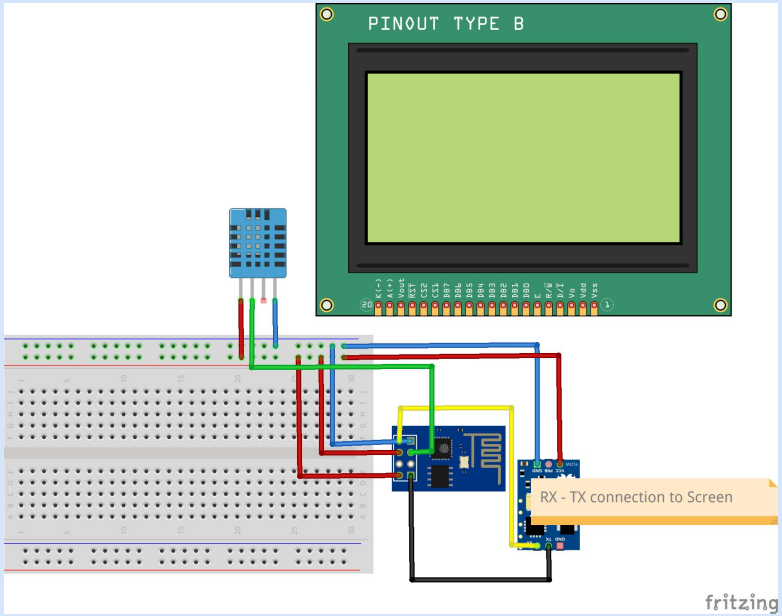
02

Schematics & Project development

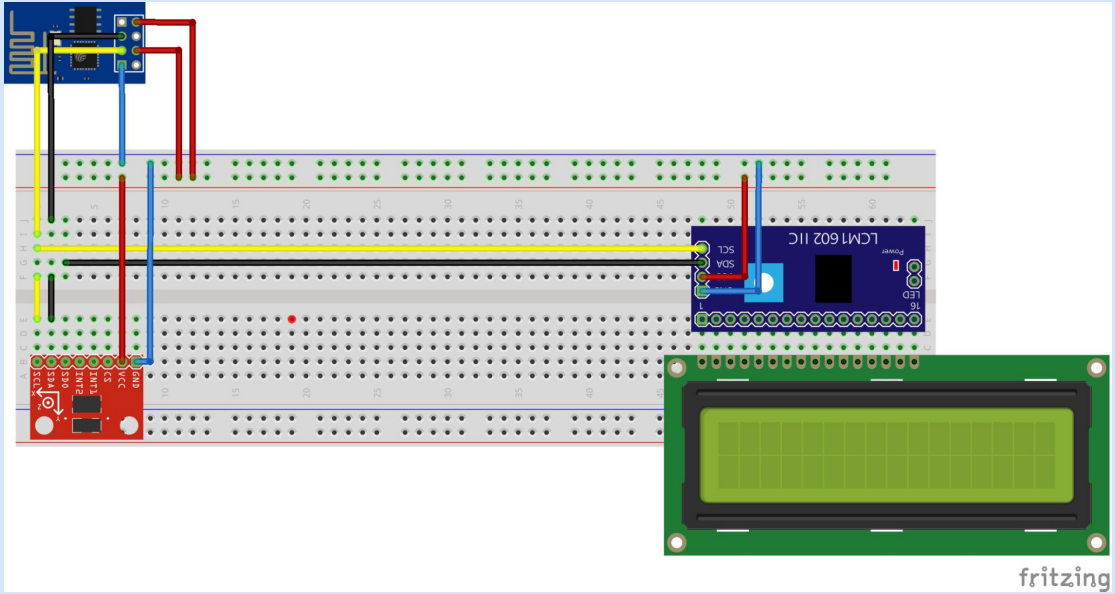
02.1

Schematics

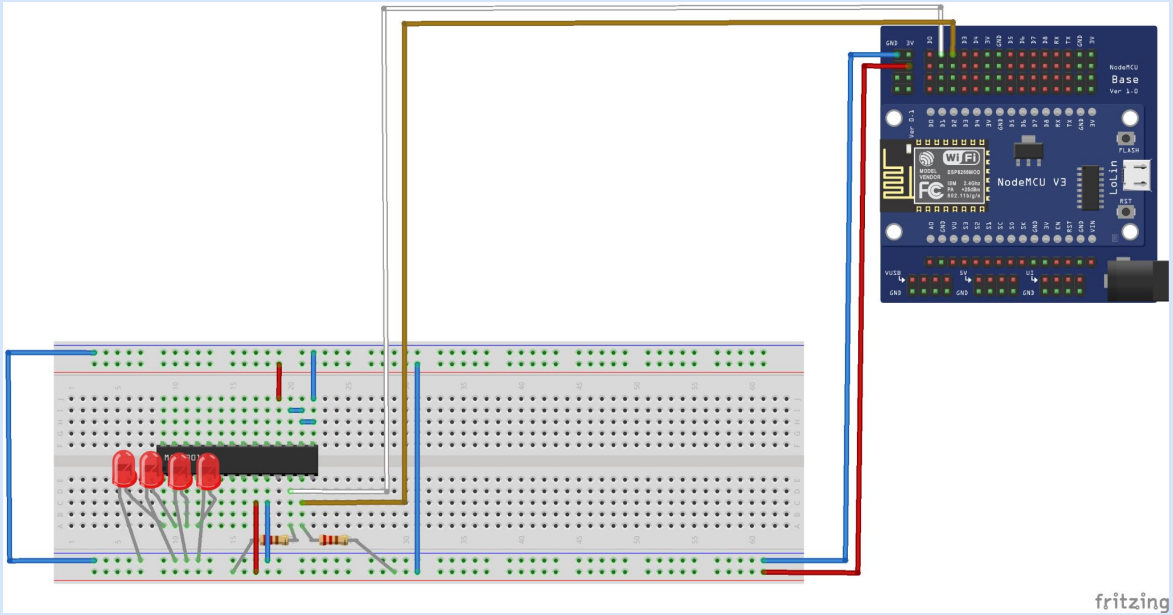
Frizzing DTP 1



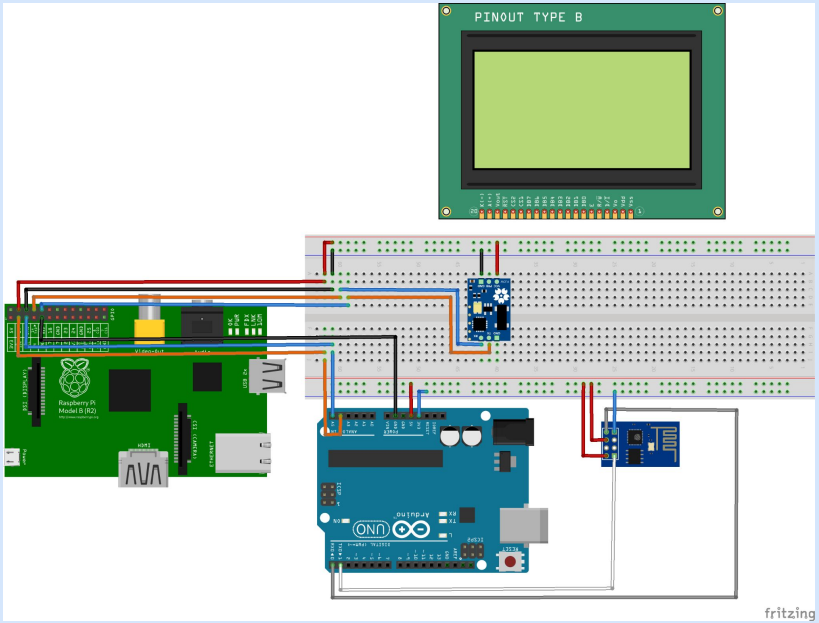
Frizzing DTP 2



Frizzing MQTT Broker



Frizzing Supervision Station



02.2

Project Development

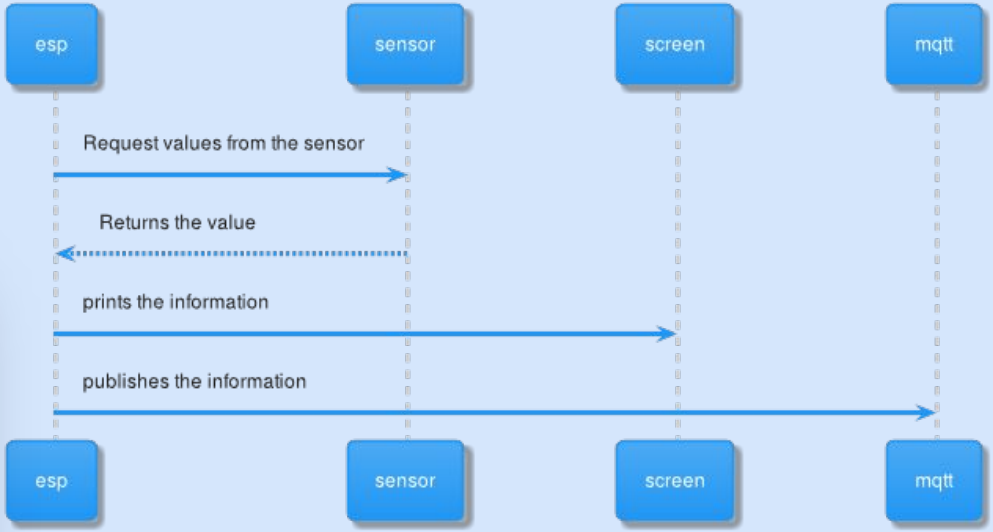
DTP 1 & MQTT

Libraries:

```
#include "DHT.h"
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
```

Message:

```
{
“humidity”: %f,
“temperature”: %f,
“hic”: %f
}
```



DTP 2 & MQTT

Libraries:

```
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
```

Message:

```
{
“x” : %f,
“y” : %f,
“z” : %f
}
```

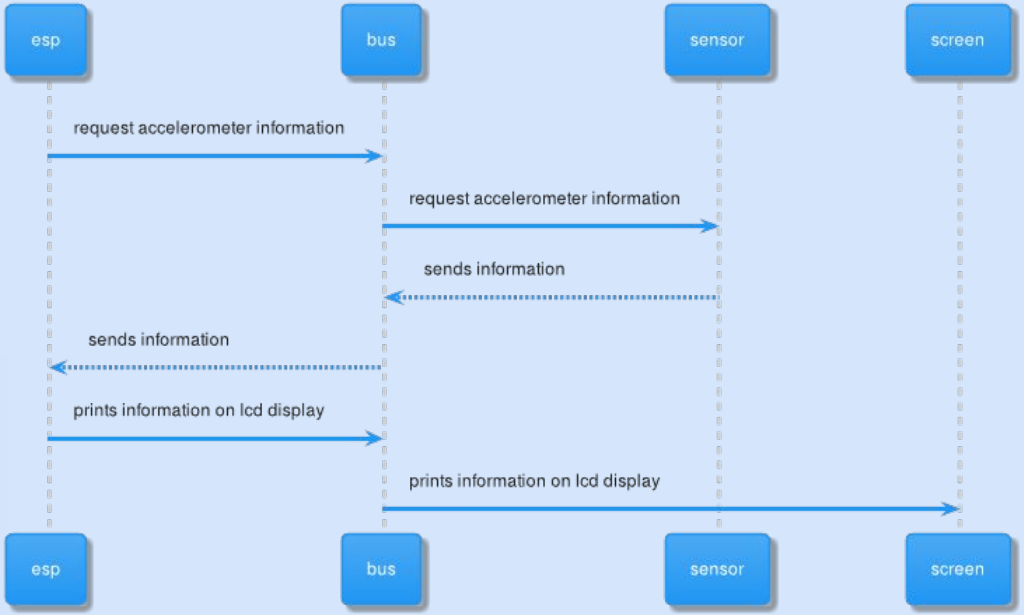


Diagram Flow MQTT

Libraries:

```
#include "Wire.h"
#include <ESP8266WiFi.h>
#include "uMQTTBroker.h"
```

```
int toLedValue(int num) {
  return int(pow(2, num) - 0.5);
}
```

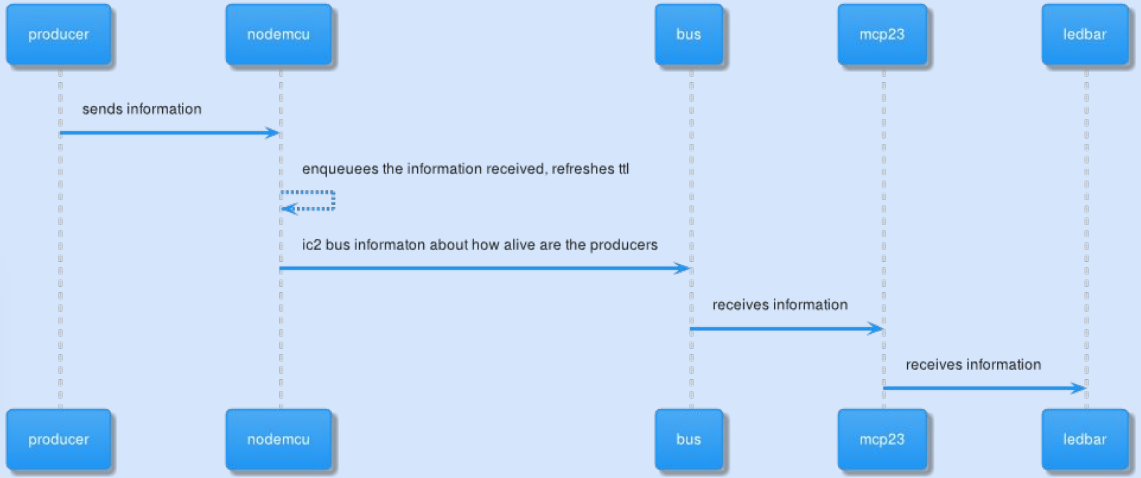
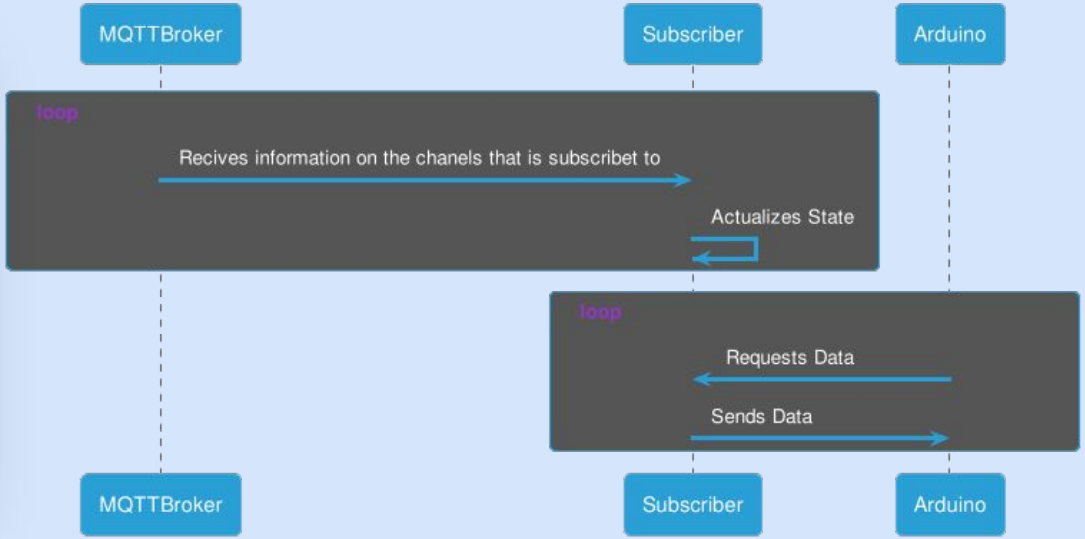


Diagram Flow
Supervision Station (1)

```
typedef struct
{
  float humidity;
  float temperature;
  float heatIndex;
} dataproducer1;

typedef struct
{
  float x;
  float y;
  float z;
} dataproducer2;
```

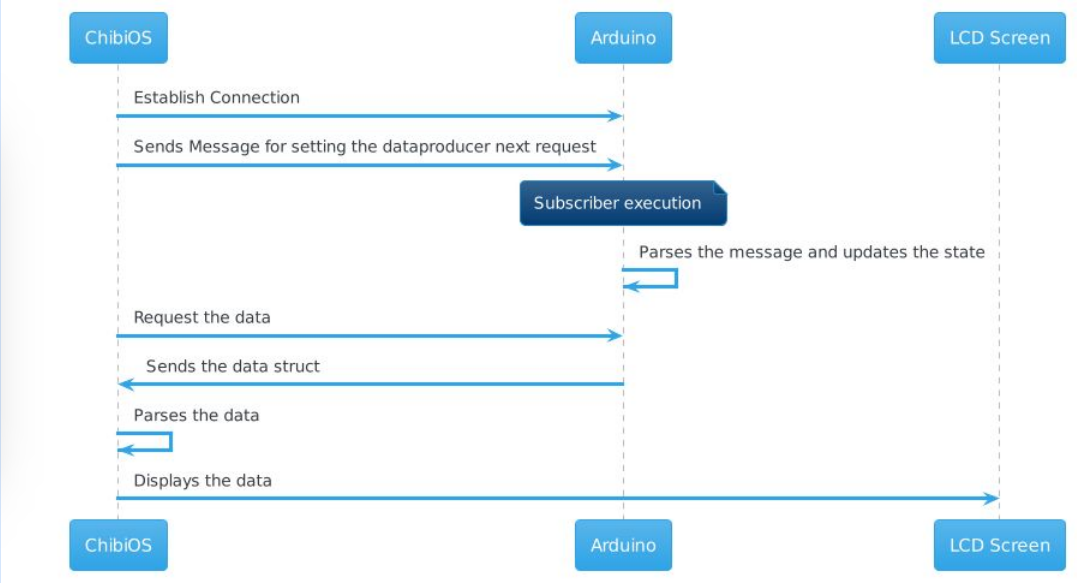


Protocol: Subscriber + Arduino

- ESP-01 actualizes the data every time a message is sent to the broker.
- Communication starts on Arduino:
  - **Arduino Sending:** “0” - **ESP-01 Returns:** “test”
  - **Arduino Sending:** “1” - **ESP-01 Returns:** { “temperature”: 23.00 , “humidity”:43.00, “hic”:25.00 }
  - **Arduino Sending:** “2” - **ESP-01 Returns:** { “X” : 0.12, “Y” : 3.56, “Z” : 13.32};”
  - **Arduino Sending:** “j” - **ESP-01 Returns:** Full json for debugging purposes.
- Arduino parses json using ArduinoJson and sends info to Reporting center

Diagram Flow
Supervision Station (2)

```c
struct data_unit_t
{
  float humidity;
  float temperature;
  float heat_index;
  float accelerometerX;
  float accelerometerY;
  float accelerometerZ;
};
```



Libraries

RPi

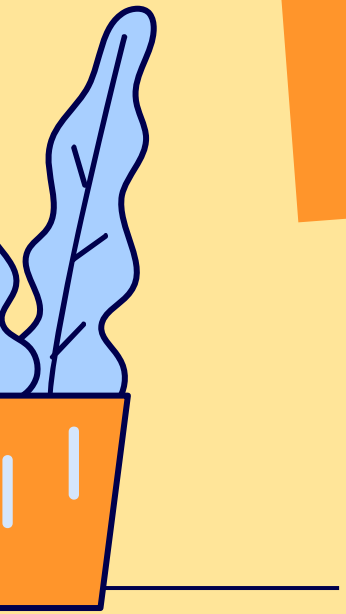
```
#include "ch.h"
#include "hal.h"
#include "chprintf.h"
#include "chvt.h"
#include <string.h>
```

ESP-01

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Dictionary.h>
```

Arduino

```
#include <ArduinoJson.h>;
#include "Wire.h"
```



03

Project Management

Initial Product Backlog

	Title	Assignees	Status	Milestone
4	Prepare the IDE for Arduino	Oriolac, quimpm, and sergisi	Done	Sprint 1
5	Dataproducer 1 temperature and humidity - analysis	Oriolac, quimpm, and sergisi	Done	Sprint 1
6	Prepare DataProducer 2 Board Fritzing with Arduino	Oriolac, quimpm, and sergisi	Done	Sprint 1
7	Dataproducer 2 accelerometer - analysis	Oriolac, quimpm, and sergisi	Done	Sprint 1
8	DHT11 Library for Humidity	Oriolac and sergisi	Done	Sprint 1
9	Testing ESP01 with Tinker	Oriolac	Done	Sprint 1
	Producer 1 Board Fritzing with Arduino	Oriolac	Done	
	taProducer 2 - I2c connection LCD Screer	sergisi	Done	
	Producer 1 - LCD Display Serial Backpa	quimpm and sergisi	Done	

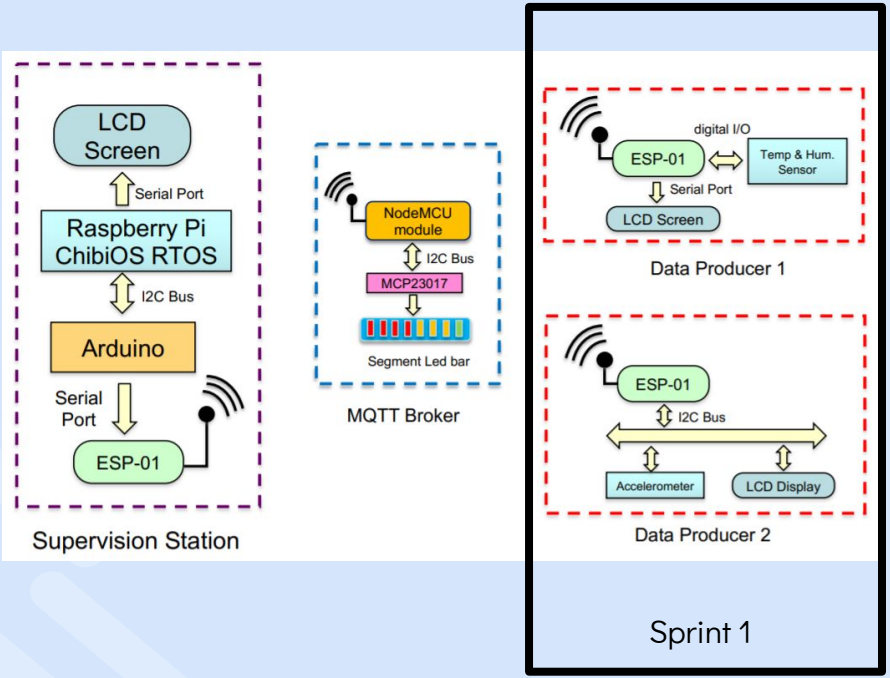
There was no weight neither a label in the issues.

The Backlog only had issues from Sprint 1

PROJECT PLANNING

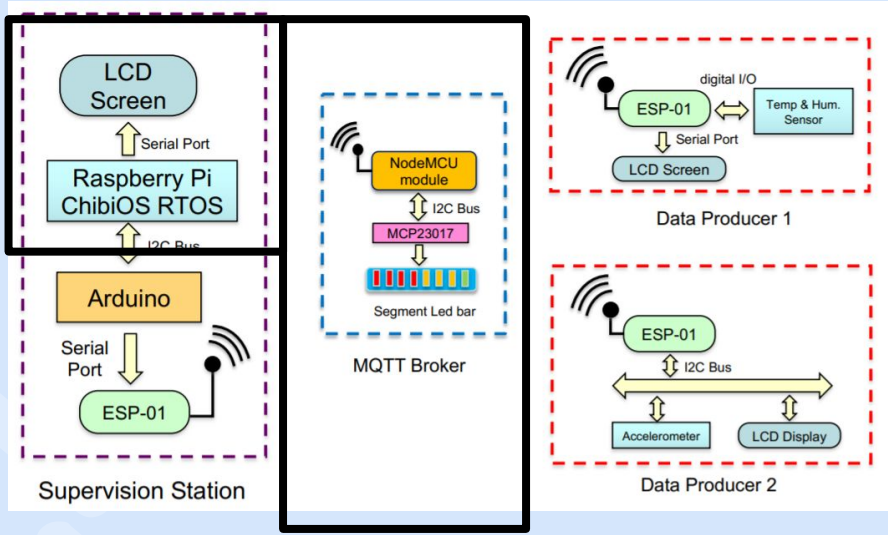
- 1st sprint
  - Data Producers
- 2nd sprint
  - MQTT Broker
- 3rd sprint
  - Supervision Station

PROJECT PLANNING



- 1st sprint
  - Data Producers
- 2nd sprint
  - MQTT Broker
- 3rd sprint
  - Supervision Station

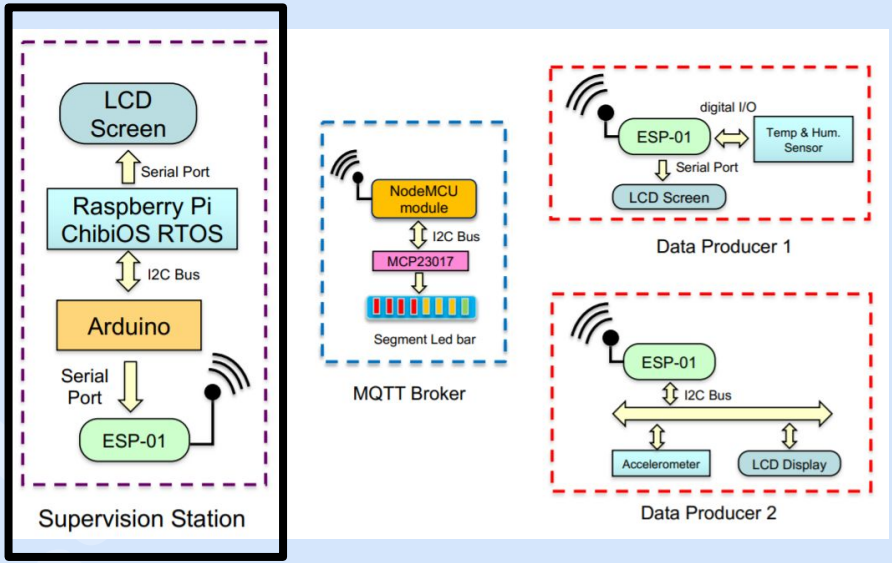
PROJECT PLANNING



Sprint 2

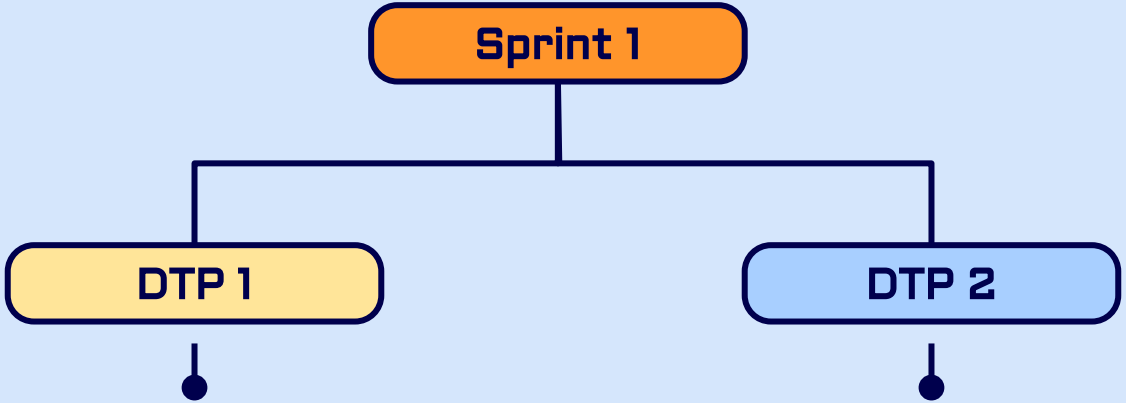
- 1st sprint
  - Data Producers
- 2nd sprint
  - MQTT Broker
- 3rd sprint
  - Supervision Station

PROJECT PLANNING



Sprint 3

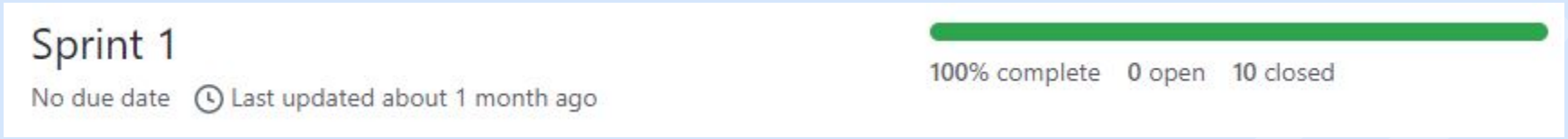
- 1st sprint
  - Data Producers
- 2nd sprint
  - MQTT Broker
- 3rd sprint
  - Supervision Station



34 h

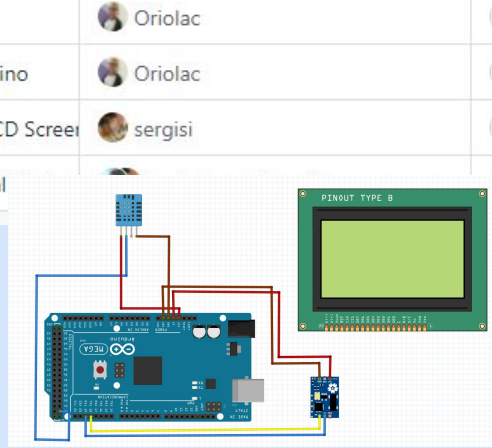
User Stories:

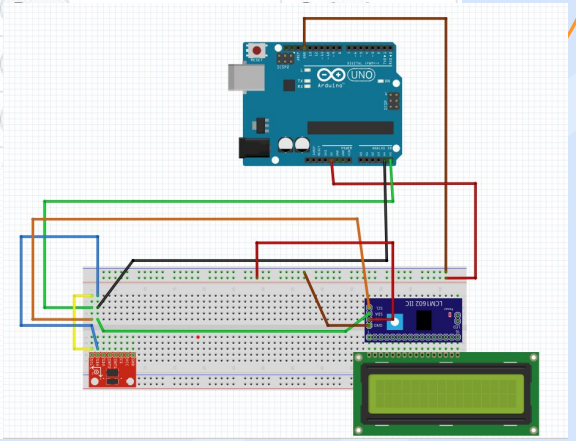
- It is necessary to obtain the temperature and humidity at the top of the WTG.
- It is necessary to determine the WTG tower movement.



Issues

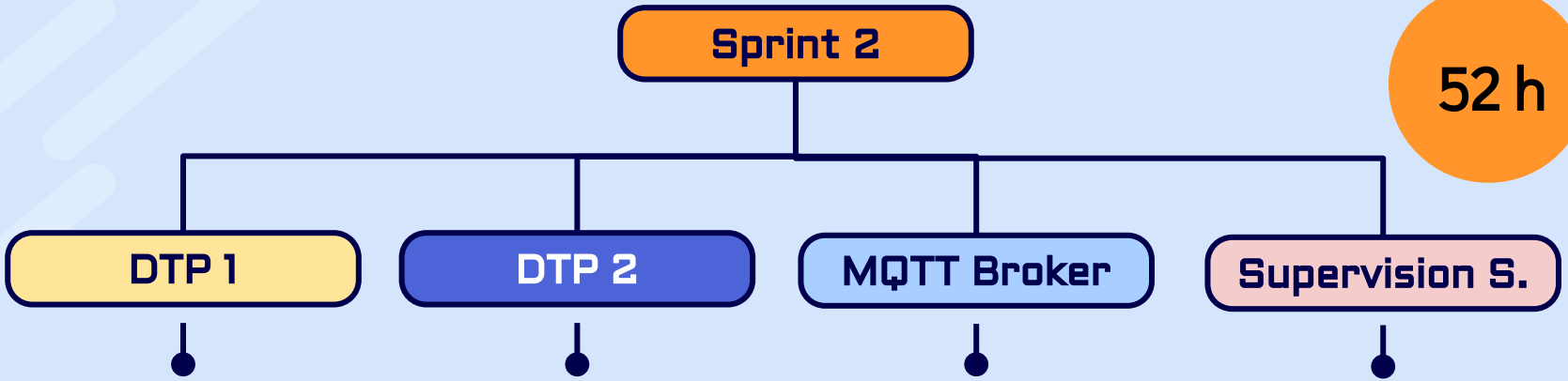
	Title	Assignees	Status	Milestone
4	Prepare the IDE for Arduino	Oriolac, quimpm, and sergisi	Done	Sprint 1
5	Dataproducer 1 temperature and humidity - analysis	Oriolac, quimpm, and sergisi	Done	Sprint 1
6	Prepare DataProducer 2 Board Fritzing with Arduino	Oriolac, quimpm, and sergisi	Done	Sprint 1
7	Dataproducer 2 accelerometer - analysis	Oriolac, quimpm, and sergisi	Done	Sprint 1
8	DHT11 Library for Humidity	Oriolac and sergisi	Done	Sprint 1
9	Testing ESP01 with Tinker	Oriolac		
10	Prepare DataProducer 1 Board Fritzing with Arduino	Oriolac		
11	Code sensors DataProducer 2 - I2c connection LCD Screer	sergisi		
12	Code sensors DataProducer 1 - LCD Display Serial			





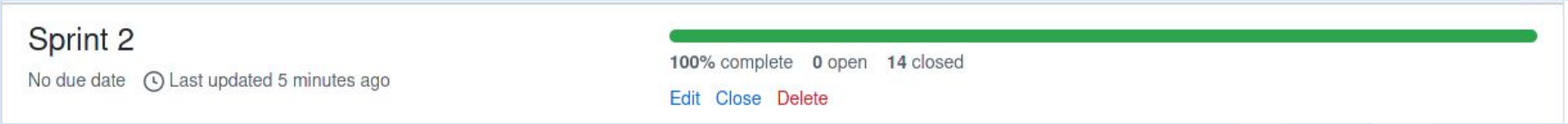
Troubles

- Understanding arduino environment for the first time
- Sparkfun Graphic LCD Serial Backpack library did not work correctly
  - Solution: Doing it with Serial manually
- Understanding I2C bus



User Stories:

- Sensor data must be sent to the MQTT broker using ESP-01
- MQTT broker must show the current number of connections
- Maintenance manager requires graphically representation of the obtained sensor data, within the last 24h, in the Supervision Station.

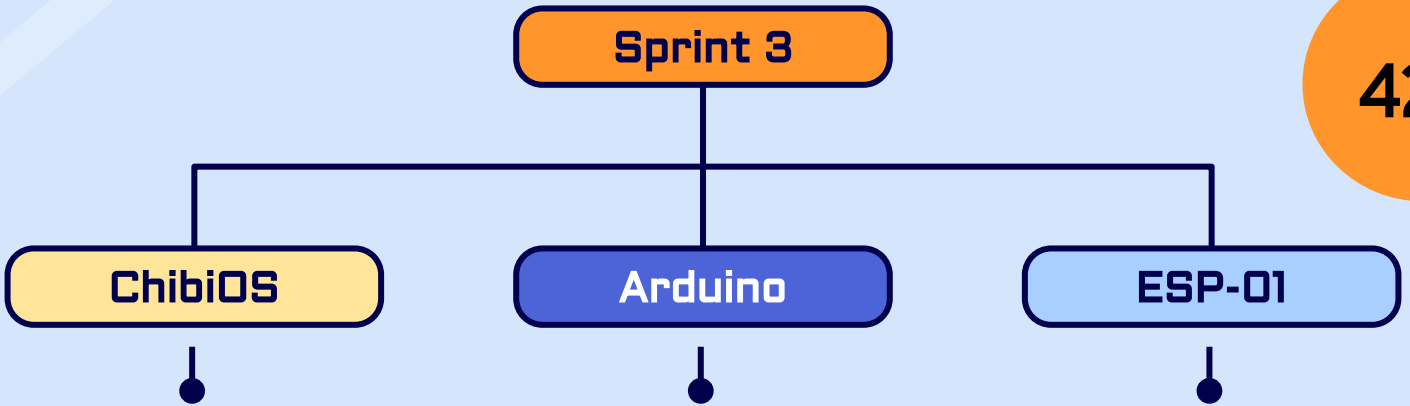


Issues

	Title	Assignees	Status	Milestone	Weight	Labels	User story
13	DTP1: MQTT connection	Oriolac, quimpm, and sergisi	Done	Sprint 2	2	development	1
14	Test NodeMCU + DTP1 + DTP2	Oriolac, quimpm, and sergisi	Done	Sprint 2	1	testing	4
15	Design fritzing NodeMCU board	sergisi	Done	Sprint 2	2	fritzing	4
16	DTP2 : MQTT connection	quimpm and sergisi	Done	Sprint 2	1	development	1
17	Redesign DTP2 board	quimpm	Done	Sprint 2	2	fritzing	1
18	Redesign DTP1 board	quimpm	Done	Sprint 2	1	fritzing	1
19	NodeMCU: LED bar	sergisi	Done	Sprint 2	5	development	4
20	NodeMCU MCP blink test	Oriolac and sergisi	Done	Sprint 2	5	testing	4
21	MQTT broker - analysis	Oriolac, quimpm, and sergisi	Done	Sprint 2	5	analysis	4
22	Prepare the IDE - Installation RPi + ChibiOS	Oriolac, quimpm, and sergisi	Done	Sprint 2	5	documentation	5
23	Development of RPi - LCDScreen connection	Oriolac	Done	Sprint 2	8	development	5
24	NodeMCU : MQTT broker	sergisi	Done	Sprint 2	8	development	4
25	Documentation chibiOS with LCD Screen	Oriolac	Done	Sprint 2	3	documentation	5

Troubles

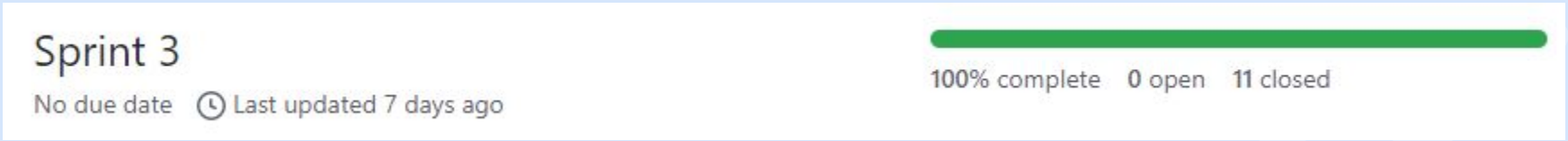
- Founded a bug in the library called *Backpack Sparkfun LCD Screen*
- Troubles with I2C with RPi



42 h

User Stories:

- Sensor data must be must be sent to a Supervision Station.
- Maintenance manager requires graphically representation of the obtained sensor data, within the last 24h, in the Supervision Station.



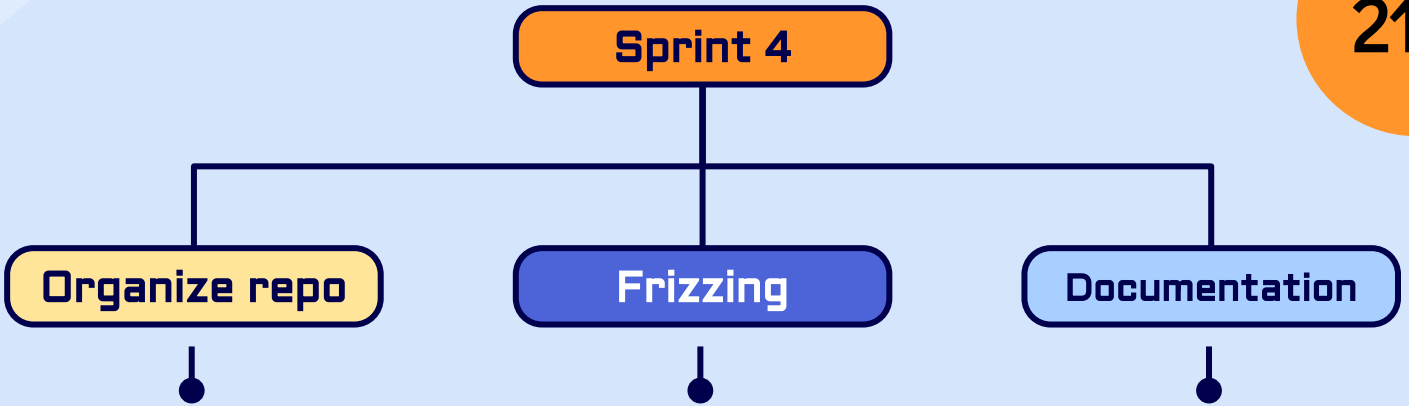
Issues

26	Documentation ChibiOS with I2C bus	sergisi	Done	Sprint 3	5	documentation	5
27	Analysis I2C bus of Arduino and RPi	sergisi	Done	Sprint 3	2	analysis	5
28	Code connection Arduino with ESP01	quimpm	Done	Sprint 3	3	development	5
29	Supervision Station Analysis	Oriolac, quimpm, and sergisi	Done	Sprint 3	5	analysis	5
30	Connection ESP1 of SupervisionStation ot MQTT Broker	quimpm	Done	Sprint 3	2	development	5
31	Development ChibiOS with LCD Screen	Oriolac	Done	Sprint 3	3	development	5
32	Development I2C bus Arduino with Raspberry Pi	sergisi	Done	Sprint 3	3	development	5
33	Development Arduino with ESP01	quimpm	Done	Sprint 3	3	development	5
34	Documentation Arduino ESP01	quimpm	Done	Sprint 3	1	documentation	5
35	Refactoring the wires of dataproducer 1, 2 and the mqtt b	Oriolac, quimpm, and sergisi	Done	Sprint 3	1	enhancement	-

Troubles

- **ESP01**: AT Commands Firmware
  - Implementing our own protocol

- **RPi**: Displaying the float
  - It has been modified the *chprintf.c* in order to display correctly the floats.
  - It has been modified the float precision to 2 decimals



21 h

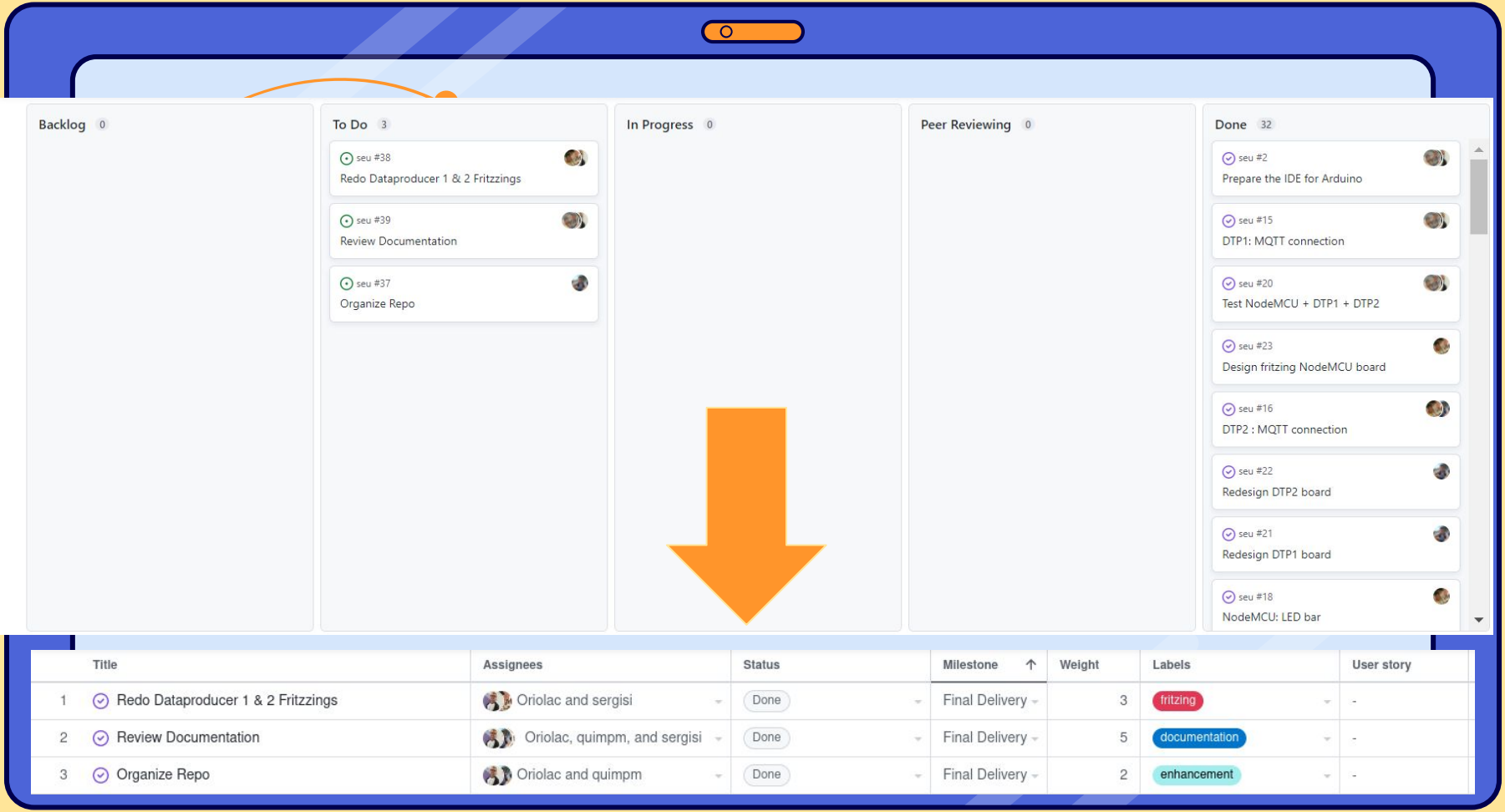
Final Delivery

No due date Last updated 5 days ago

Delivery of the project

100% complete 0 open 3 closed

Edit Close Delete



	Title	Assignees	Status	Milestone	Weight	Labels	User story
1	Redo Dataproducer 1 & 2 Fritzzings	Oriolac and sergisi	Done	Final Delivery	3	fritzing	-
2	Review Documentation	Oriolac, quimpm, and sergisi	Done	Final Delivery	5	documentation	-
3	Organize Repo	Oriolac and quimpm	Done	Final Delivery	2	enhancement	-

04
Project Conclusion

Total hours: 149 h

Possible new features to consider

- Show different data producers in each frame of Supervision Station's LCD Screen.
- Attempt a new design for the Supervision Station in order to be more scalable.
- ESP-01 maintains the last few messages to compute the median to have a more accurate reality to display in the supervision station.

- Good project design for getting in touch with embedded and real time systems.
  - Each task has its own difficulties.
  - In each sprint it has been possible to learn something new
  - Using agile methodology has allowed to make the project more structured and the progress has been stable among the semester.
- Some components are quite old and they malfunction occasionally.