

UDLMINF 2020 - 21

# Ubiquitous & Embedded Systems

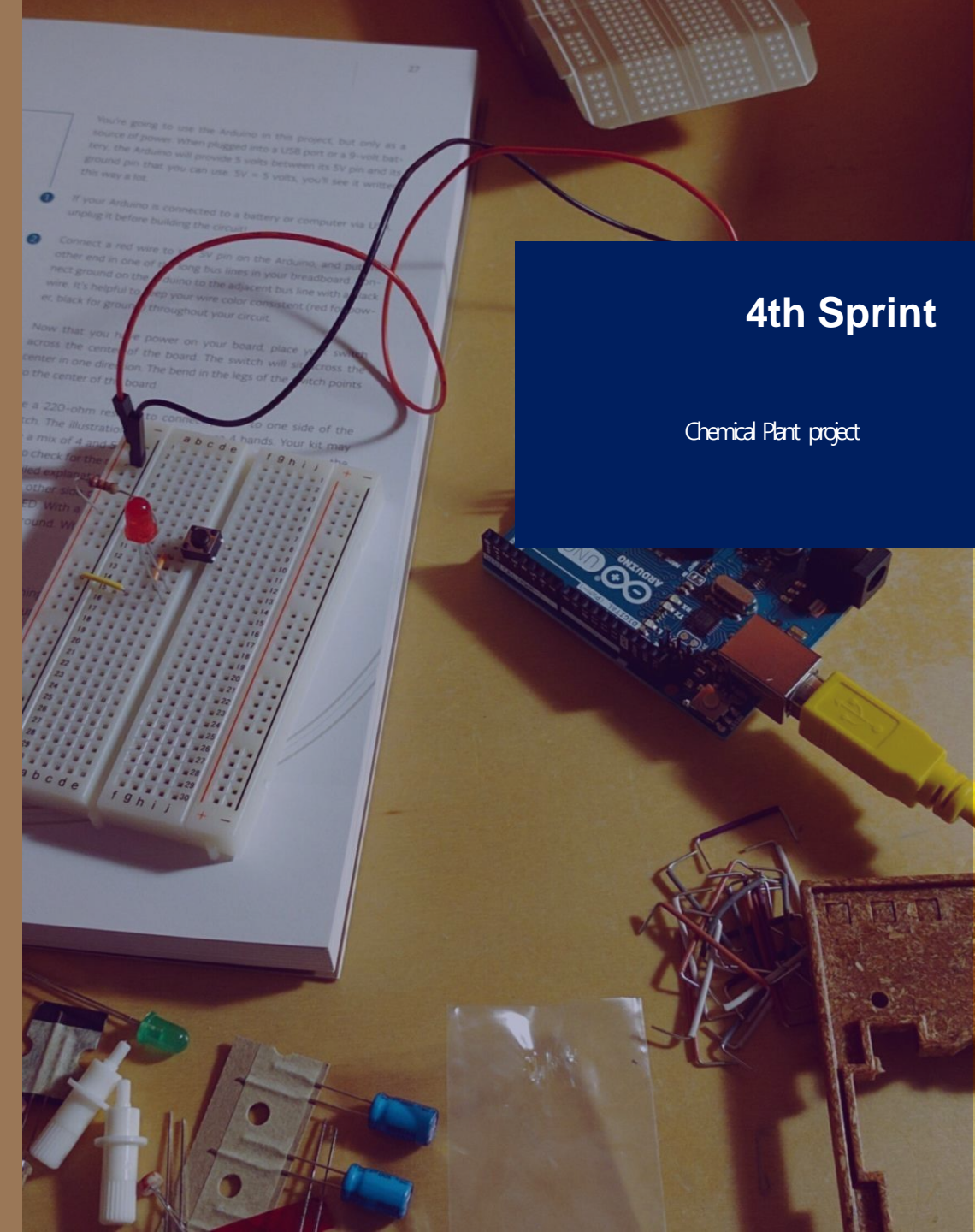
**Team 1 Danillo Lange, Jeongyun Lee, Ronnel Mathew**

Github : <https://github.com/jy-977/UBQ-> Danillo

Lange: <https://github.com/roxdan> Jeongyun Lee:

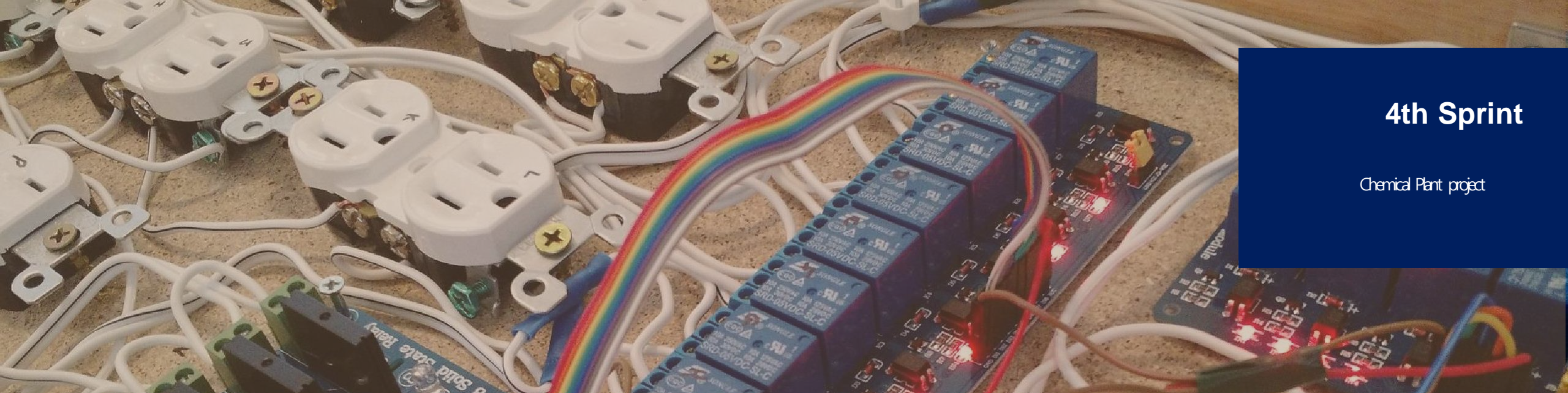
<https://github.com/jy-977> Ronnel Mathew:

<https://github.com/ron7858>



**4th Sprint**

Chemical Plant project



## 4th Sprint

Chemical Plant project

### ***PRESENTATION INDEX***

***Follow up***

**01**

***Briefing***

**02**

***4th Sprint***

**03**

***Q&A***

**04**

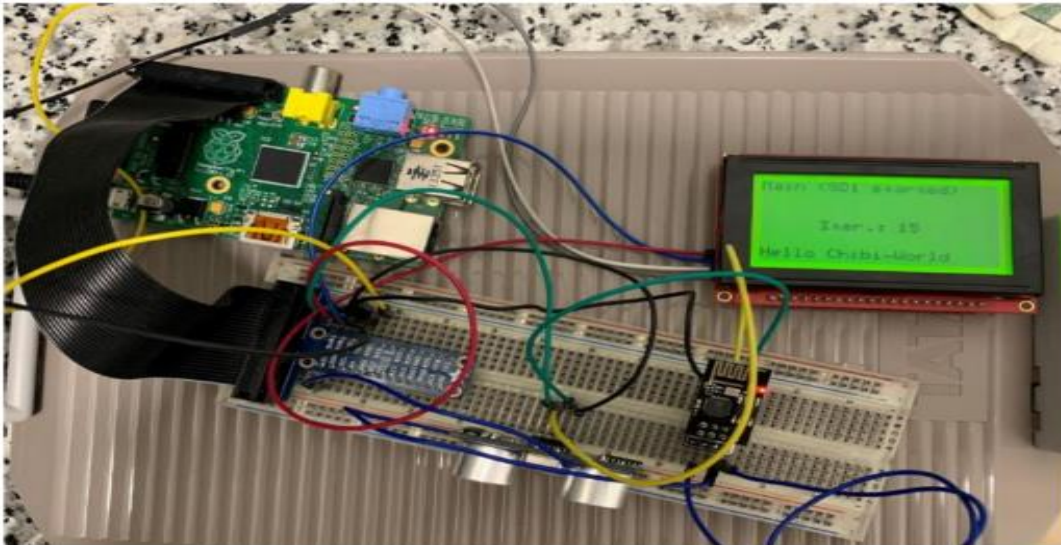
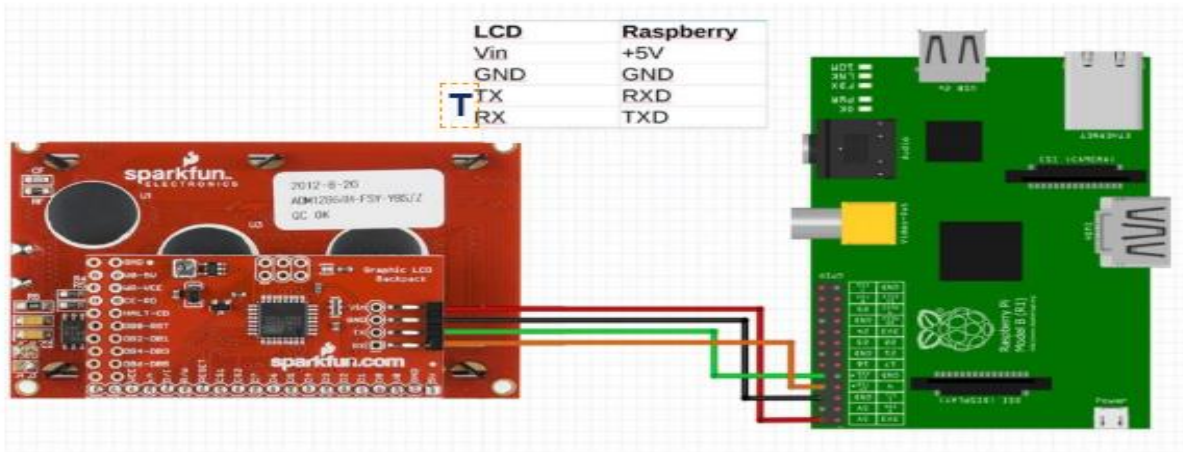


# Follow up on 3rd Sprint

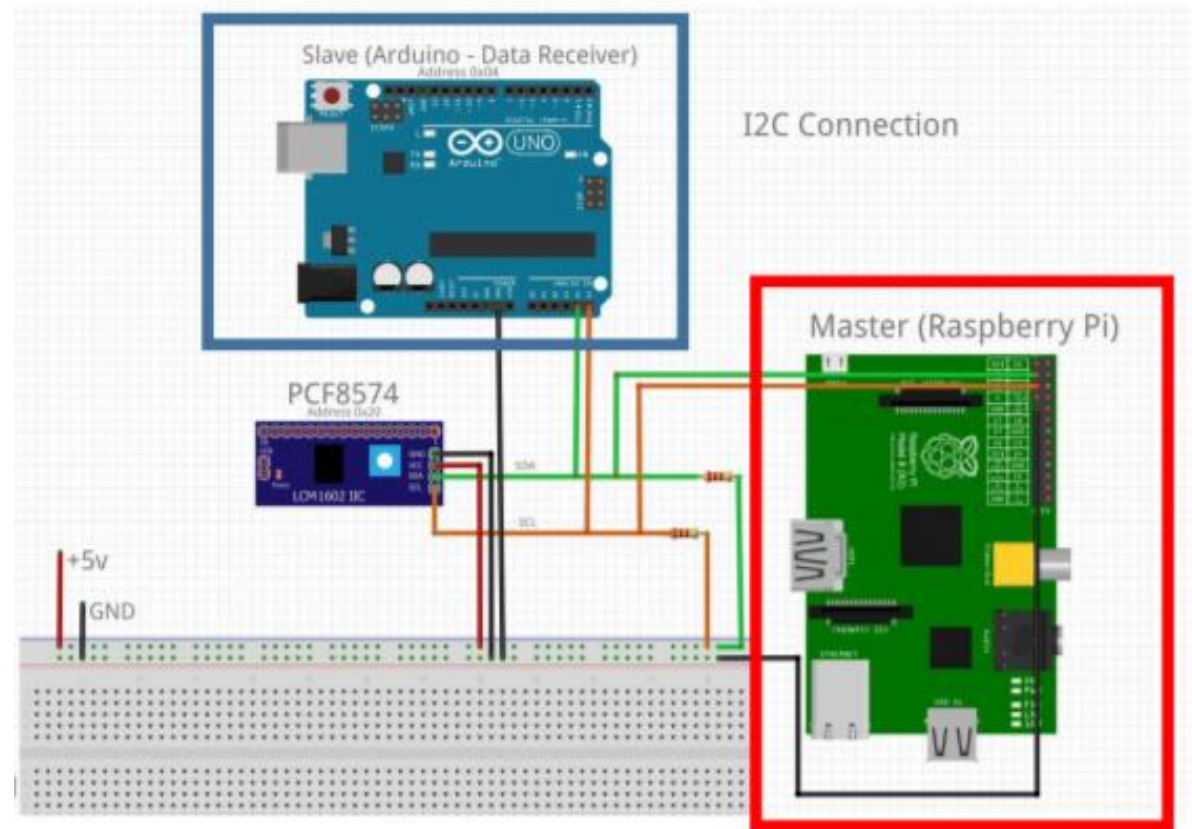
04

- How to -> LCD Screen: Raspberry + LCD
- R.Pi <-> Arduino interaction: Raspberry + Arduino + i2c protocol
- How to -> I2C protocol: i2c protocol behavior and connections
- How to -> PCF 8754: PCF connections with LED's + ChibiOS Code

# Follow Up



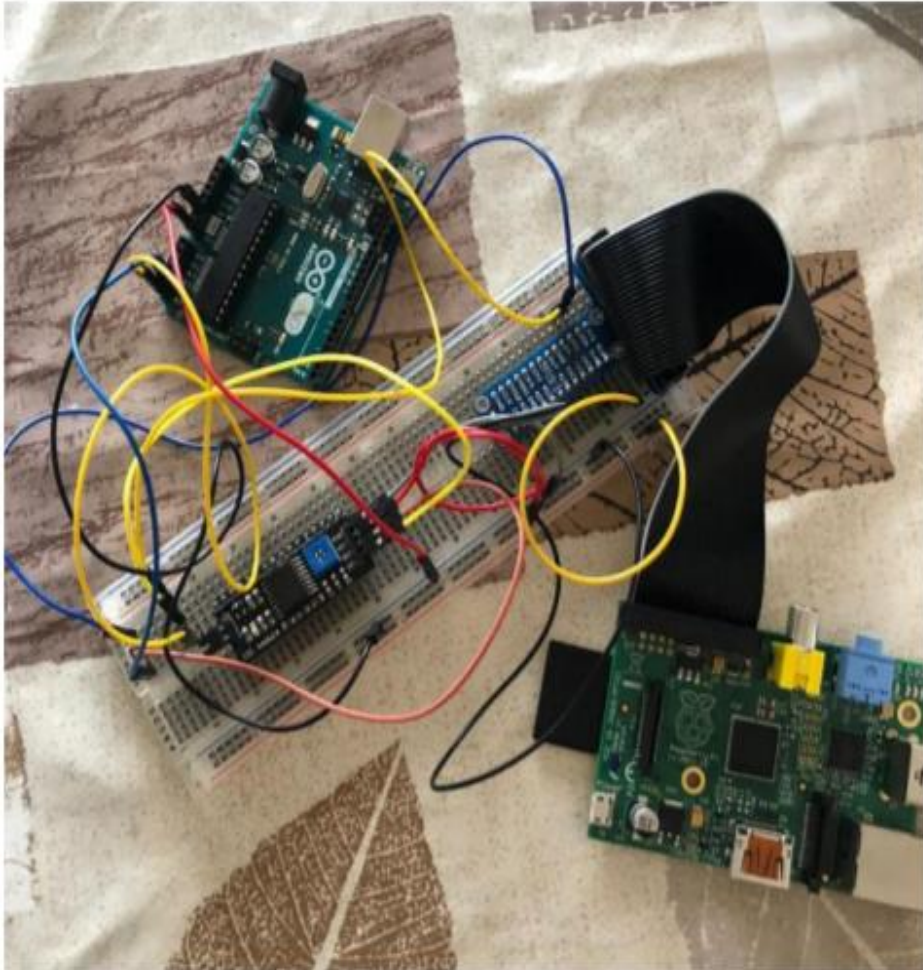
Raspberry+ LCD screen



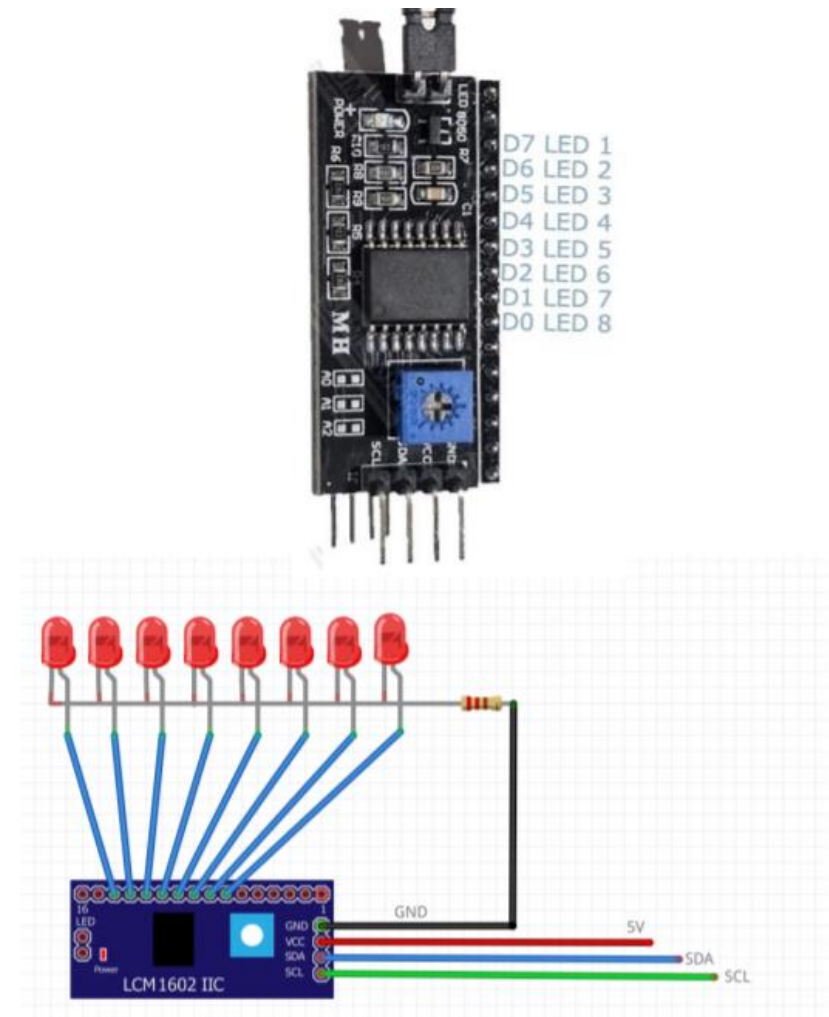
Raspberry+ Arduino  
interaction



# *Follow Up*



I2C Protocol



PCF8574 connection

# SCRUM SPRINT

Done

01

3 **Group**  
4 **Group**  
3 **Group**

Components Background knowledge  
Prepare the IDE – Arduino, EPS  
Prepare the IDE – R.Pi +Chibi os

8 Oct-  
22 OCT

02

5 **Danillo**  
3 **Danillo**  
3 **Yoon**  
4 **Ron**  
2 **Yoon**

Define the Checking/ Test Process  
ESP01 – Arduino interaction  
Data Producer 1 Development  
Data Producer 2 Development  
Presentation

22 Oct  
-5 Nov

03

2 **Ron**  
8 **Ron**  
4 **Y / D**  
9 **Y / D**  
2 **Danillo**

How to – LCD Screen  
How to – PCF 8754  
How to – I2C protocol  
R.Pi <-> Arduino interaction  
Presentation

5 Nov-  
19 Nov

04

3 **Yoon**  
5 **D / R**  
2 **Ron**  
8 **Group**

Data log screen representation  
Ultrasonic – Led bar representation  
Presentation  
Final Assembly

# 4th Sprint

04

## ■ Sprint tasks:

- Temperature/Humidity <-> Screen representation - Effort: 3 - (Yoon)
- Ultrasonic <-> Led bar representation - Effort: 5 - (Danillo)
- Final Assembling – Effort 8 - (Group)

# 4th Sprint Details

## 01. User Stories

Operator requires information of the temperature and humidity

- ✓ Requirement to make a graphical representation of the obtained temperature and humidity on the LCD screen.

→ Data log and Screen representation

Operator requires information of the current tank container level in real time and it must be shown in a led bar present in the controller

- ✓ Requirement to work with the PCF 8754 to turn on LEDs on an LED bar to represent the level of the tank.

→ Ultrasonic <-> LED bar representation

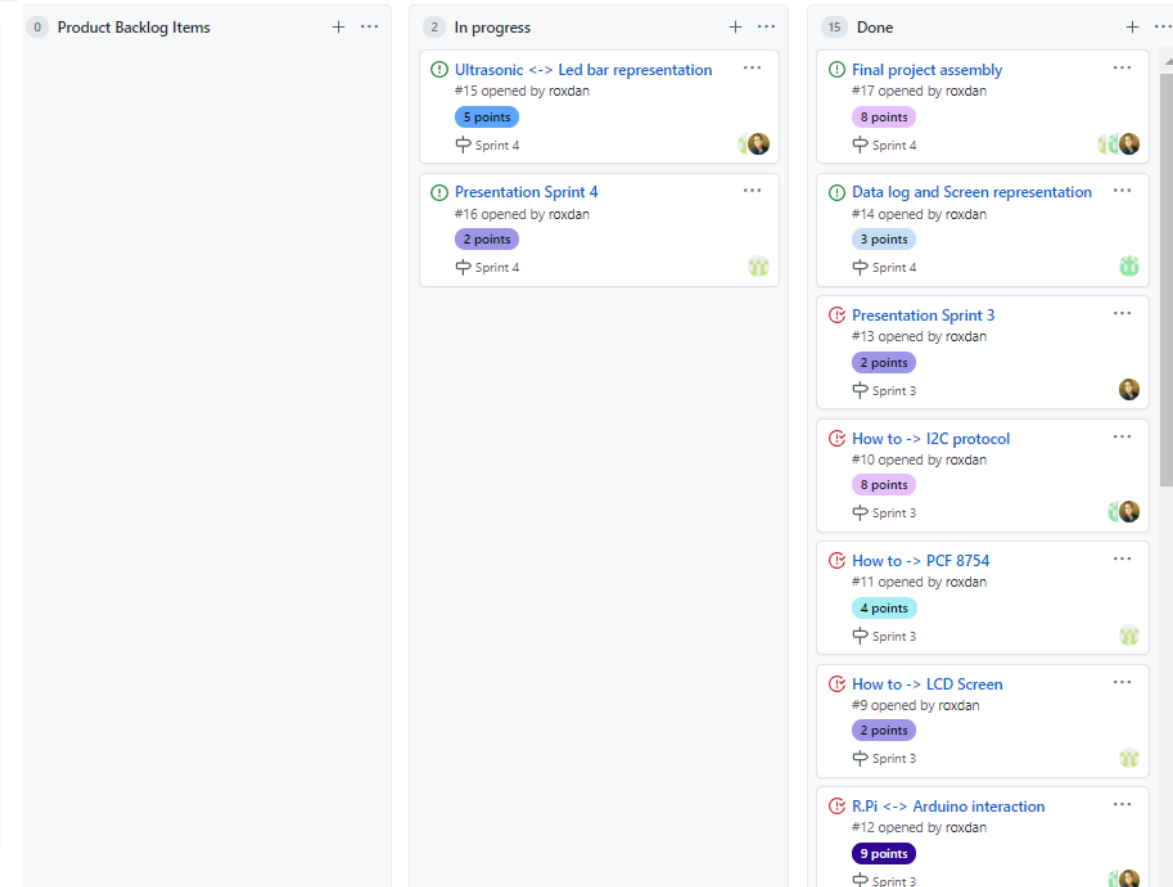


# 4th Sprint Scrum managing

04

## Scrum Sprint

Sprint	Date	Task	PIC	Effort Point	status
1		Acquire basic knowledge of the components and protoboard	All	3	Done
1		Prepare the IDE - Arduino & ESP-01	All	4	Done
1		Prepare the IDE - R.Pi + ChibiOS	All	3	Done
2	8 Oct - 22 Oct	Prepare the IDE - R.Pi + ChibiOS	All	3	Done
2	8 Oct - 22 Oct	Define the checking/test process	Danillo	5	Done
2	8 Oct - 22 Oct	ESP-01 <-> Arduino interaction	Danillo	3	Done
2	8 Oct - 22 Oct	Data Producer 1 development	Yoon	4	Done
2	8 Oct - 22 Oct	Data Producer 2 developmen	Ron	4	Done
2	8 Oct - 22 Oct	Presentation	Yoon	2	Done
3	22 Oct - 5 Nov	How to -> LCD Screen	Ron	2	Done
3	22 Oct - 5 Nov	How to -> I2C protocol	Yoon,Danillo	8	Done
3	22 Oct - 5 Nov	How to -> PCF 8754	Ron	4	Done
3	22 Oct - 5 Nov	R.Pi <-> Arduino interaction	Yoon,Danillo	9	Done
3	22 Oct - 5 Nov	Presentation	Danillo	2	Done
4	5 Nov - 19 Nov	Data log and Screen representation	Yoon	3	Done
4	5 Nov - 19 Nov	Ultrasonic <-> Led bar representation	Yoon	5	In progress
4	5 Nov - 19 Nov	Presentation	Ron	2	Done



The Scrum Board visualization shows the following columns and tasks:

- Product Backlog Items (0 items):**
  - 0 Product Backlog Items
- In progress (2 items):**
  - Ultrasonic <-> Led bar representation (#15 opened by roxdan, 5 points, Sprint 4)
  - Presentation Sprint 4 (#16 opened by roxdan, 2 points, Sprint 4)
- Done (15 items):**
  - Final project assembly (#17 opened by roxdan, 8 points, Sprint 4)
  - Data log and Screen representation (#14 opened by roxdan, 3 points, Sprint 4)
  - Presentation Sprint 3 (#13 opened by roxdan, 2 points, Sprint 3)
  - How to -> I2C protocol (#10 opened by roxdan, 8 points, Sprint 3)
  - How to -> PCF 8754 (#11 opened by roxdan, 4 points, Sprint 3)
  - How to -> LCD Screen (#9 opened by roxdan, 2 points, Sprint 3)
  - R.Pi <-> Arduino interaction (#12 opened by roxdan, 9 points, Sprint 3)



# Briefing Tasks 4th Sprint

04

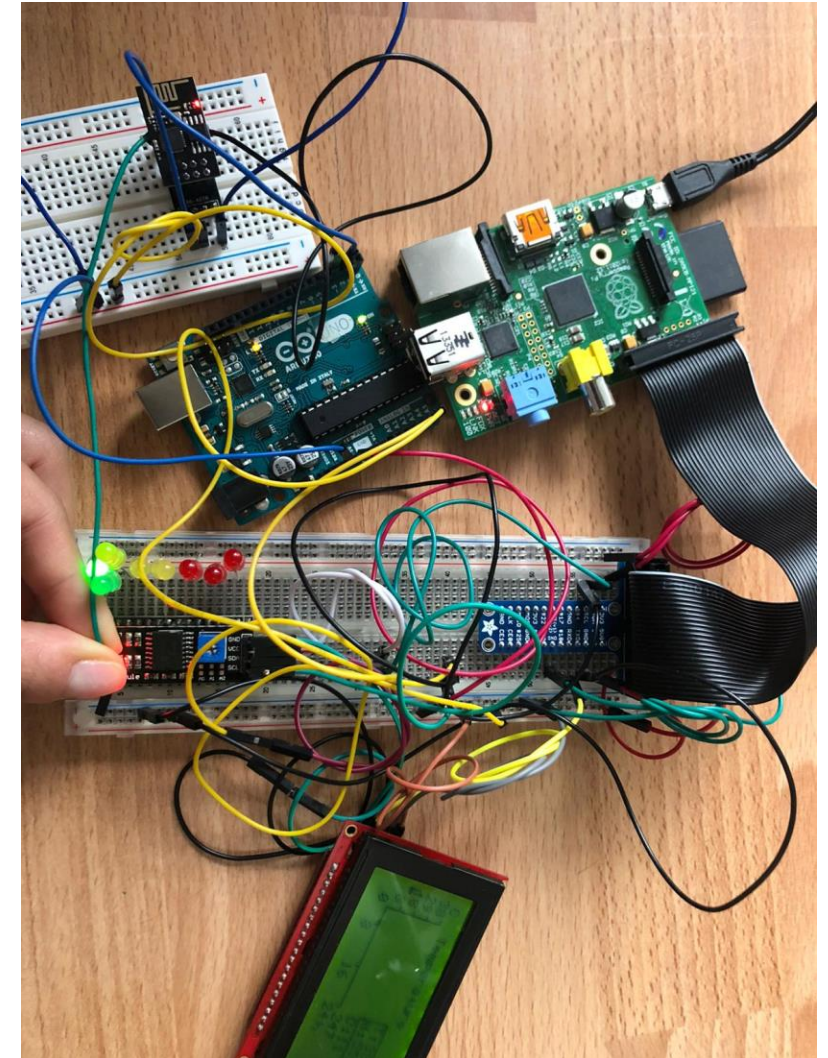
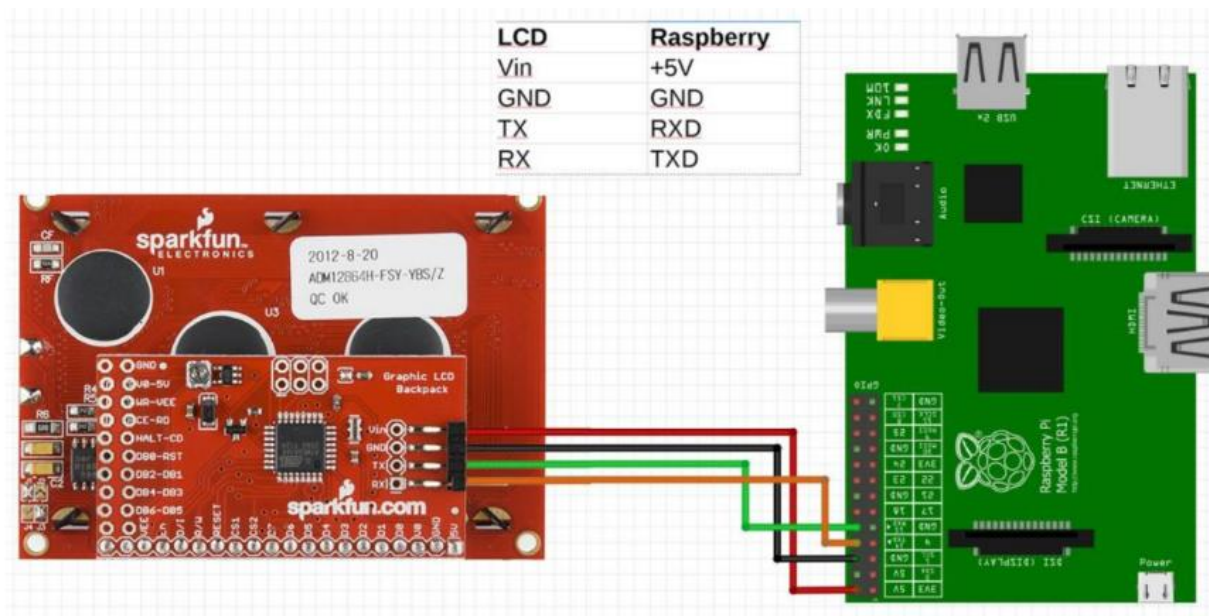
- Danillo Lange
- Jeongyun Lee
- Ronnel Mathew

Briefing the ongoing tasks

Jeongyun Lee

# 01 Datalog and Screen representation

The code is present in the GitHub repo and here the wiring connection can be seen as follows:



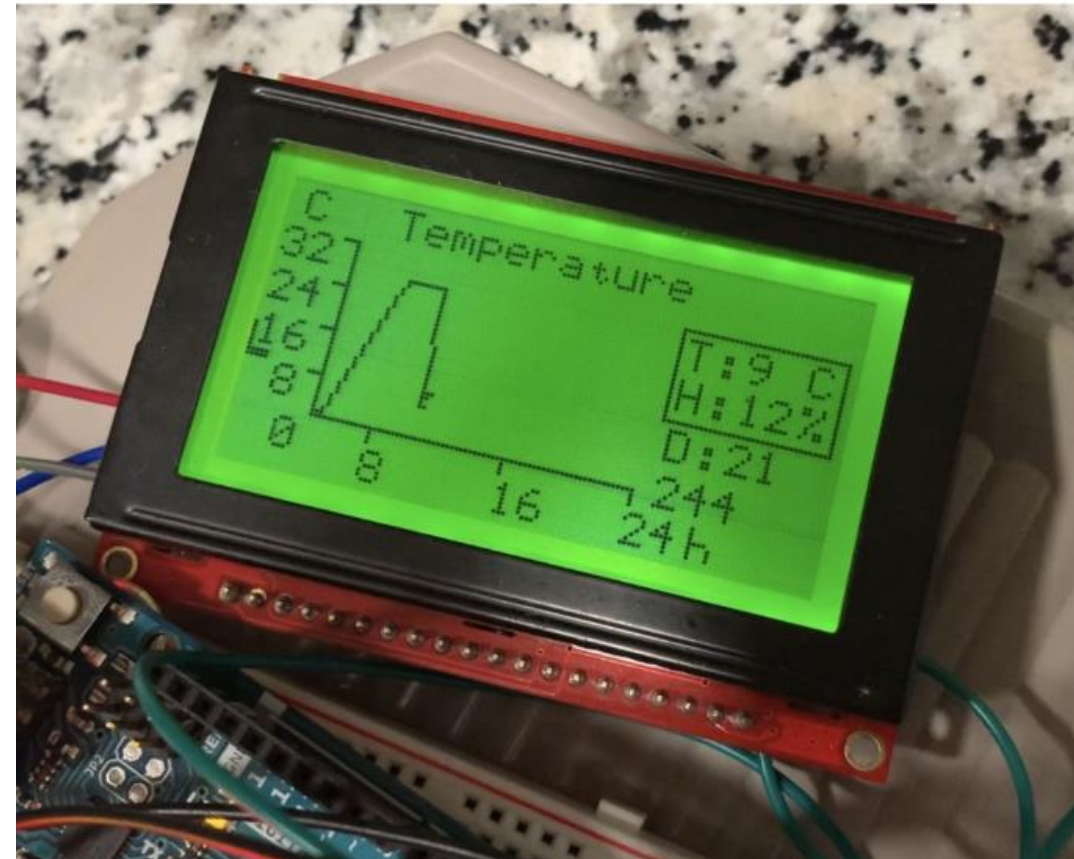


## Briefing the ongoing tasks

The objective is to make a graphical representation of the obtained temperature and humidity on an LCD screen being a relation of value x time (24h).

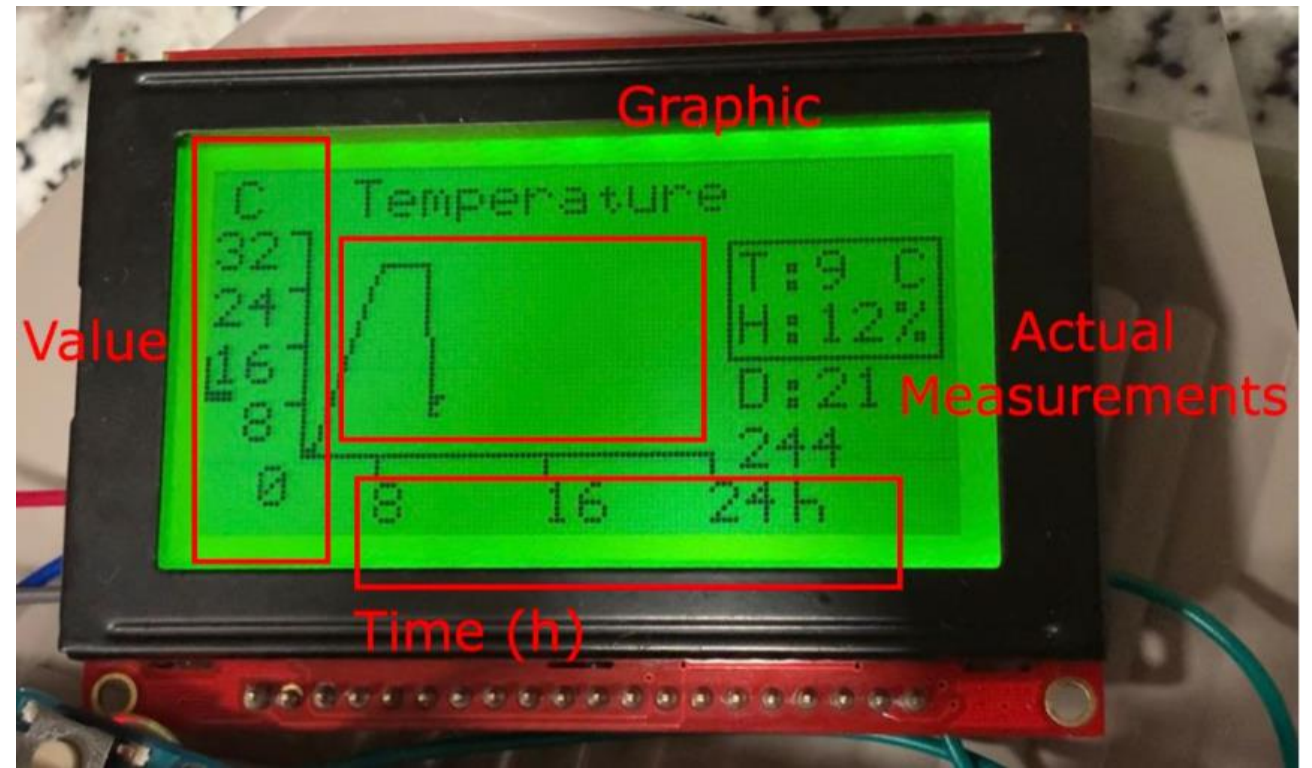
After every 5 iterations (seconds) the screen switches by displaying the temperature and the humidity.

The final result is obtained in the following figure:



## Final Result

- All values for the graph are saved on an array of co-ordinates so we see the graphics for 24h
- The y-axis represents the temperature in Celsius
- The x-axis represents the time in hours



## Code part for the temperature and for humidity (entire code is on GitHub):

Drawing  
structures  
of graphic:

```
void drawStructure()
{
    // info
    lcdPrintf(92, 47, "T:", 0);
    lcdPrintf(92, 38, "H:", 0);
    lcdPrintf(92, 27, "D:", 0);
    //
    lcdPrintf(10, 11, "%u", 0);
    lcdPrintf(118, 47, "C", 0);
    lcdPrintf(118, 38, "%%", 0);
    //mainframe
    drawLine(17, 13, 17, 52, 0);
    drawLine(18, 13, 87, 13, 0);
    //
    drawLine(14, 52, 17, 52, 0);
    drawLine(14, 42, 17, 42, 0);
    drawLine(14, 32, 17, 32, 0);
    drawLine(14, 22, 17, 22, 0);
    //
    drawLine(30, 12, 30, 10, 0);
    drawLine(59, 12, 59, 10, 0);
    drawLine(87, 12, 87, 10, 0);
    //
    drawBox(90, 49, 125, 29, 0);
    // legend
    lcdPrintf(30, 8, "%u", 8);
    lcdPrintf(59, 8, "%u", 16);
    lcdPrintf(87, 8, "%u", 24);
    lcdPrintf(101, 8, "h", 0);
    //
}
```

Drawing line:

```
void drawGraphLineHum(int value)
{
    if (value > 76)
        value = 76;

    // title
    lcdPrintf(25, 61, "Humidity", 0);
    lcdPrintf(4, 61, "%%", 0);

    // legend
    lcdPrintf(7, 22, "%u", 19);
    lcdPrintf(1, 32, "%u", 38);
    lcdPrintf(1, 42, "%u", 57);
    lcdPrintf(1, 52, "%u", 76);

    // values
    lcdPrintf(105, 47, "%u", temperature);
    lcdPrintf(105, 38, "%u", humidity);
    lcdPrintf(105, 27, "%u", aux_counter);

    int i = 0;

    for (i = 0; i < aux_counter; i++)
    {
        drawLine(stackLineHum[i][0],
                stackLineHum[i][1],
                stackLineHum[i][2],
                stackLineHum[i][3], 0); // draw all previous lines
    }
    drawLine(stackLineHum[aux_counter - 1][2],
            stackLineHum[aux_counter - 1][3],
            stackLineHum[aux_counter - 1][2] + 1,
            14 + roundNo(value / 2), 0); // draw current line
}
```



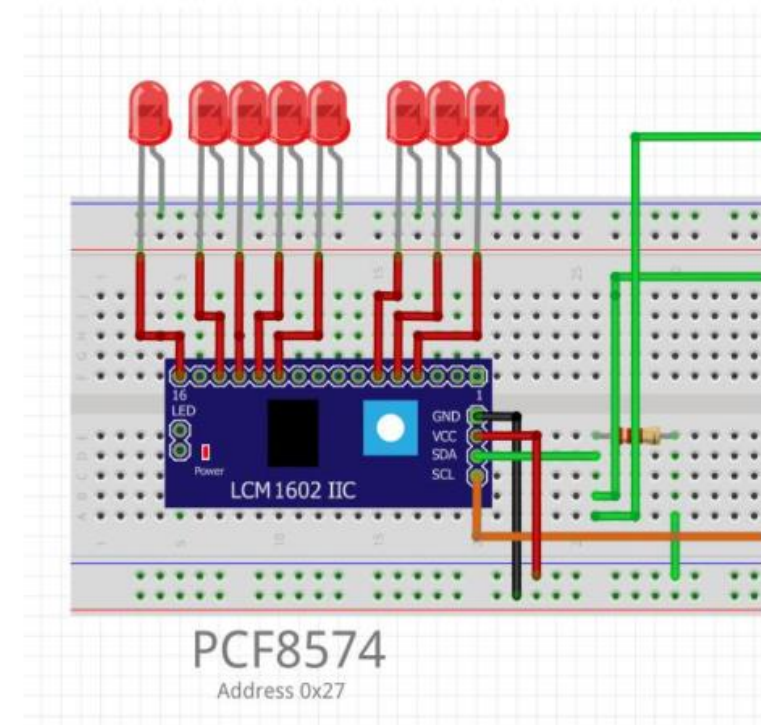
Briefing the ongoing tasks

Danillo / Ronnel

## 02 Ultrasonic LED bar representation

The PCF is connected to the Raspberry via the SDA and SCL lines (along with a resistor).

The wiring connection can be seen as shown in the figure:



## Code (ChibiOS)

Initialize the PCF

1. We send a 0x00 to the PCF to activate.

```
i2cled_init(pcf_address, 0x00);
```

Next, we need to send 2 information to the PCF:

1. The own address + 0 (to indicate it is a write operation)
2. The data that will inform which pins will light up the LEDs

- a. The address we are using

```
#define pcf_address 0x27 //device  
address  
#define pcf_address_write 0x4E // pcf  
address + 0 bit write
```

- b. The data we are sending, its 8 bits

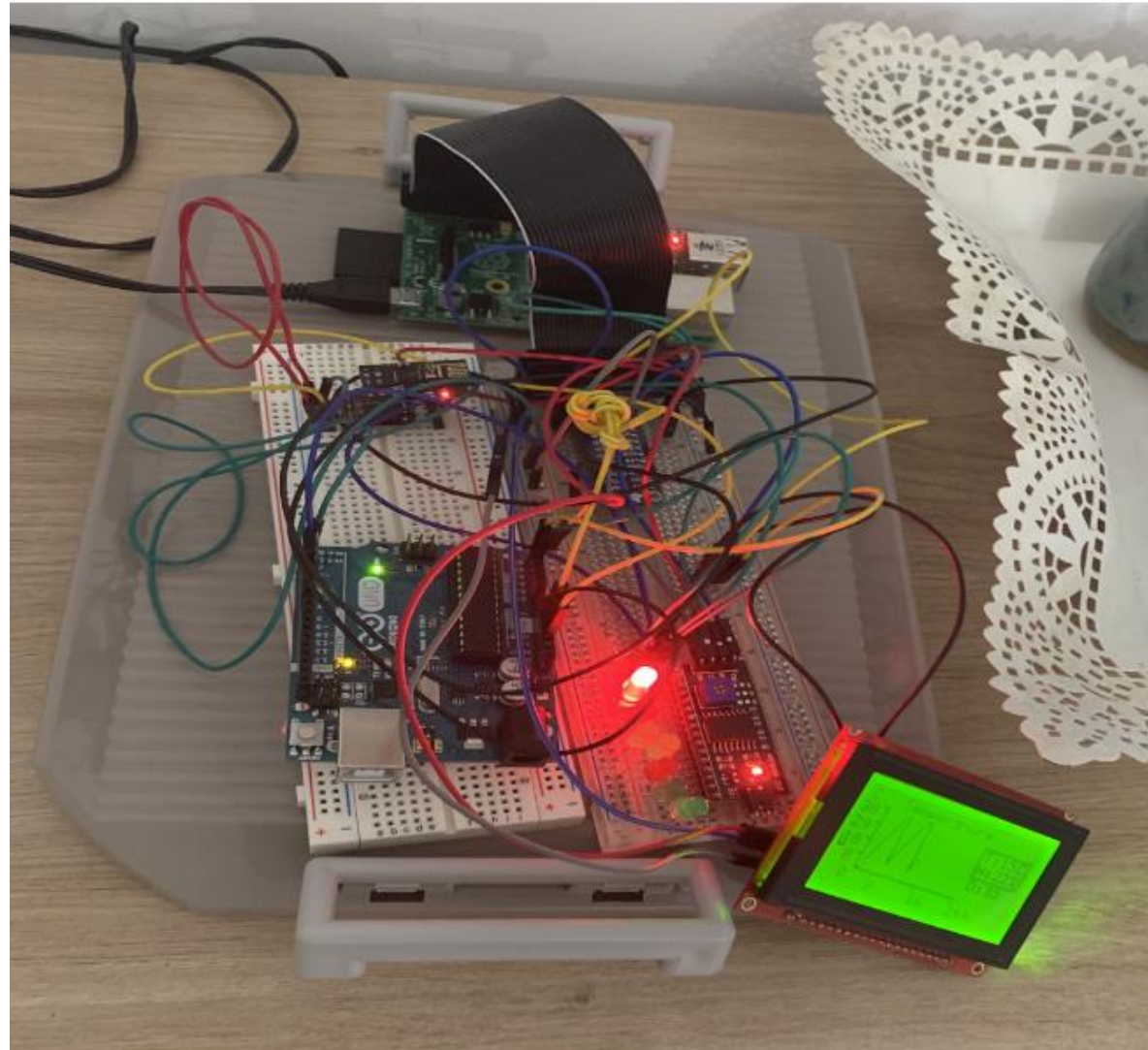
```
0b11110110
```

- c. Send both information in a array of size 2

```
request[0] = pcf_address_write;  
request[1] = data;  
  
msg_t status = i2cMasterTransmit(  
    &I2C0, device_address, request, 2,  
    NULL, 0);
```

## Final Result

The objective was to work with the PCF 8754 to turn on LEDs on an LED bar to represent the level of container and it receives the distance from the sensor and makes the calculation to display the correct amount of LEDs to switch on





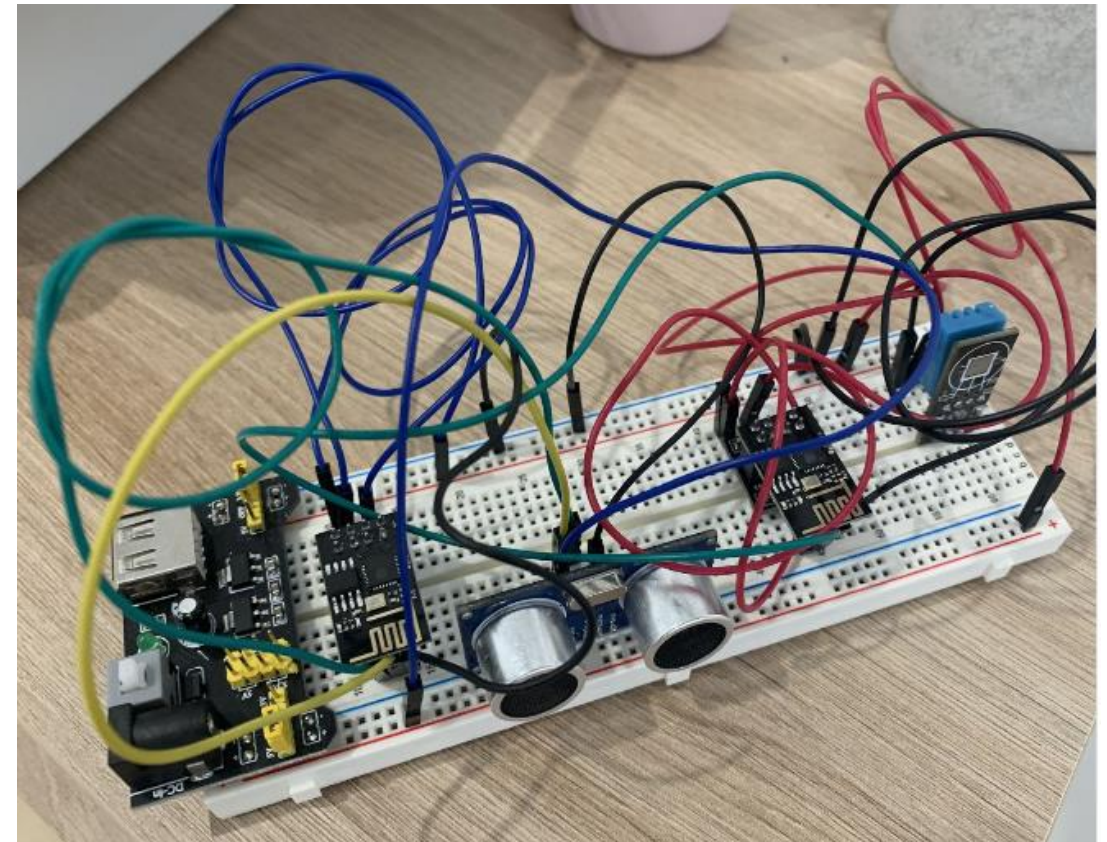
Briefing the ongoing tasks

Danillo Lange

## 03 Final Assembly

Now, we will move on to the final part of our project which is assembling all the components from the previous sprints and developing the project

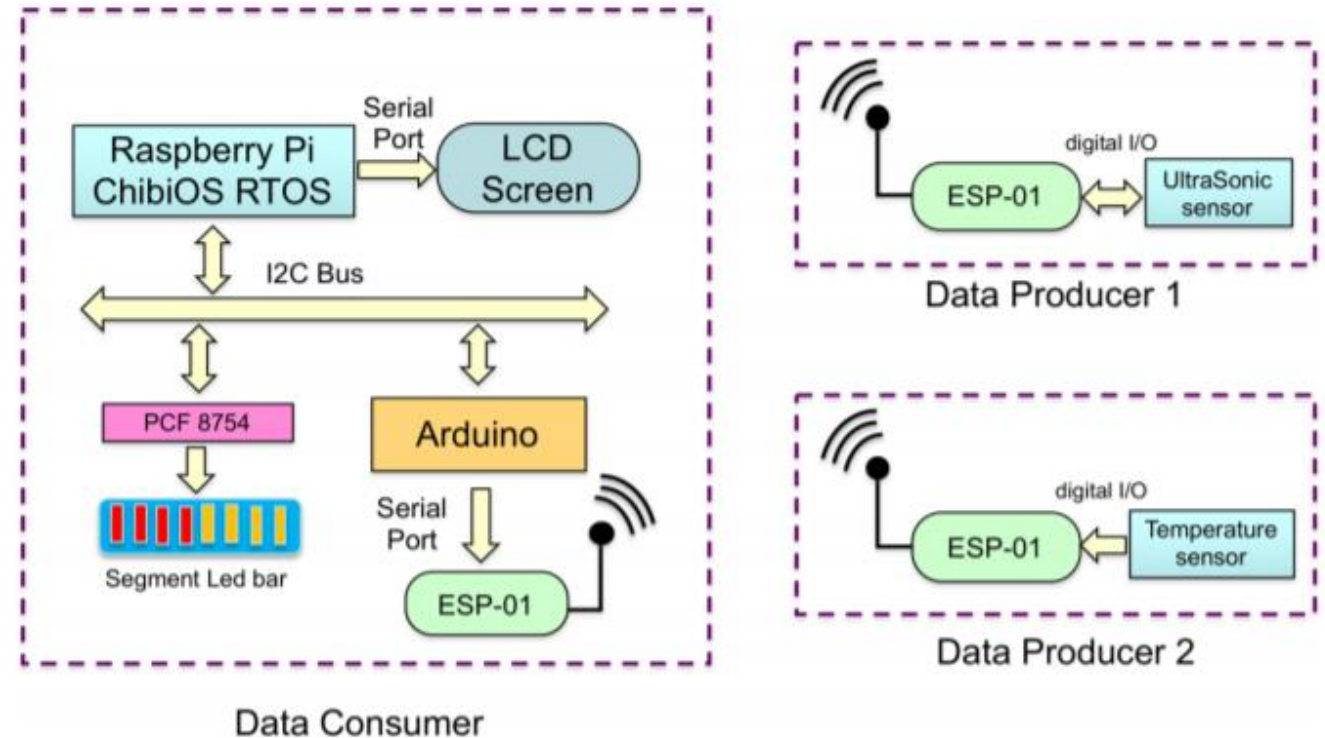
1. Power considerations: The Data Producers 1 and 2 are connected to the shield of power supply



## Briefing the ongoing tasks

2. Preparation: We have developed and built the project using the codes in the following GitHub folders:

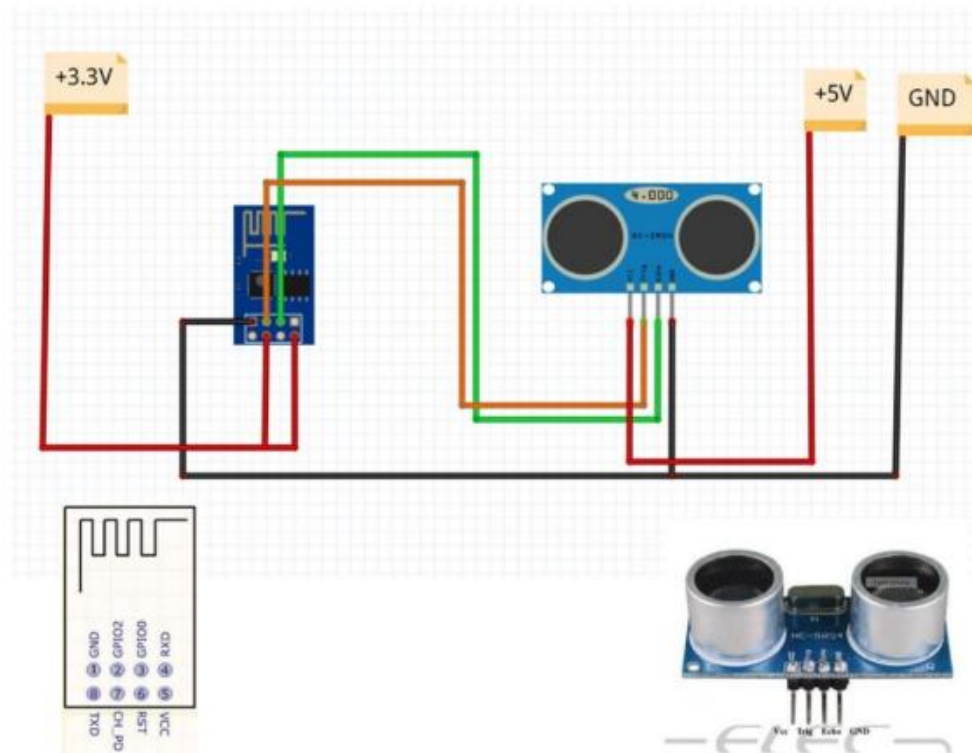
- i. Data Producers: <https://github.com/jy-977/UBQ-/tree/main/code>
- ii. Arduino (Data Receiver): [https://github.com/jy-977/UBQ-/tree/main/code/Receiver\\_ARDUINO](https://github.com/jy-977/UBQ-/tree/main/code/Receiver_ARDUINO)
- iii. ChibiOS: [https://github.com/jy-977/UBQ-/tree/main/code/ChibiOS\\_i2c\\_Raspberr](https://github.com/jy-977/UBQ-/tree/main/code/ChibiOS_i2c_Raspberr)  
[y](https://github.com/jy-977/UBQ-/tree/main/code/ChibiOS_i2c_Raspberr)



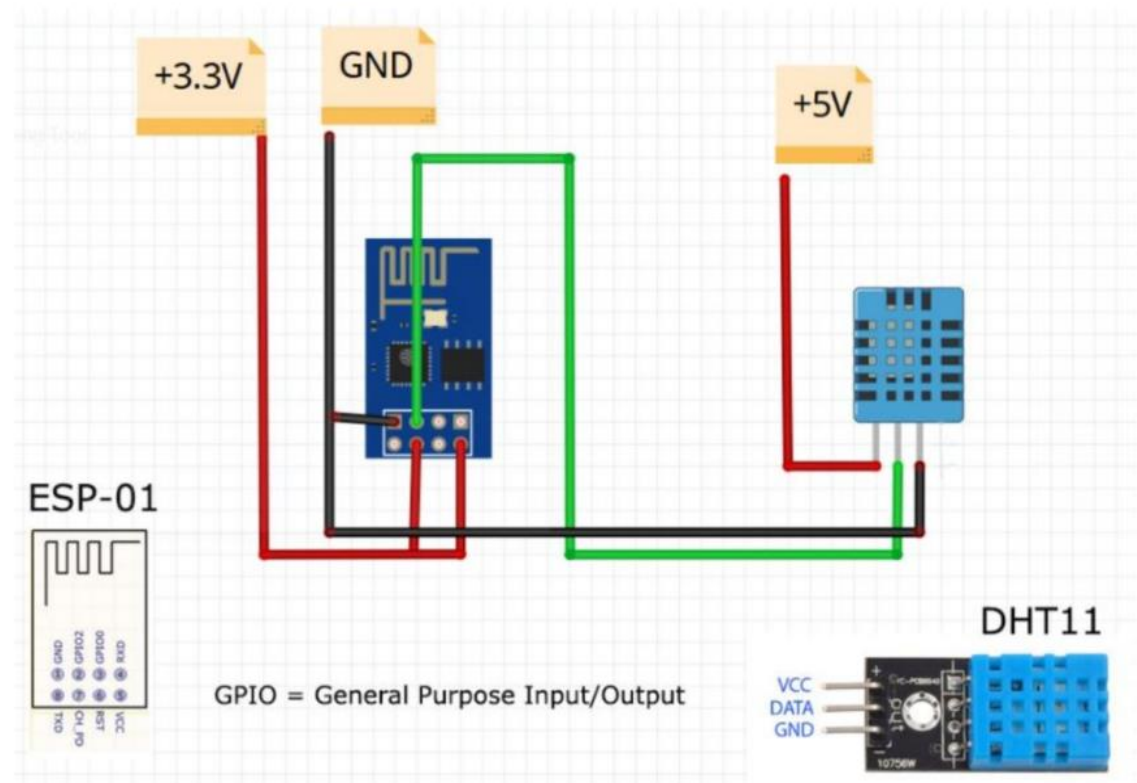
# Briefing the ongoing tasks

## 3. Wiring Connection

**Data Producer 1 schema:**



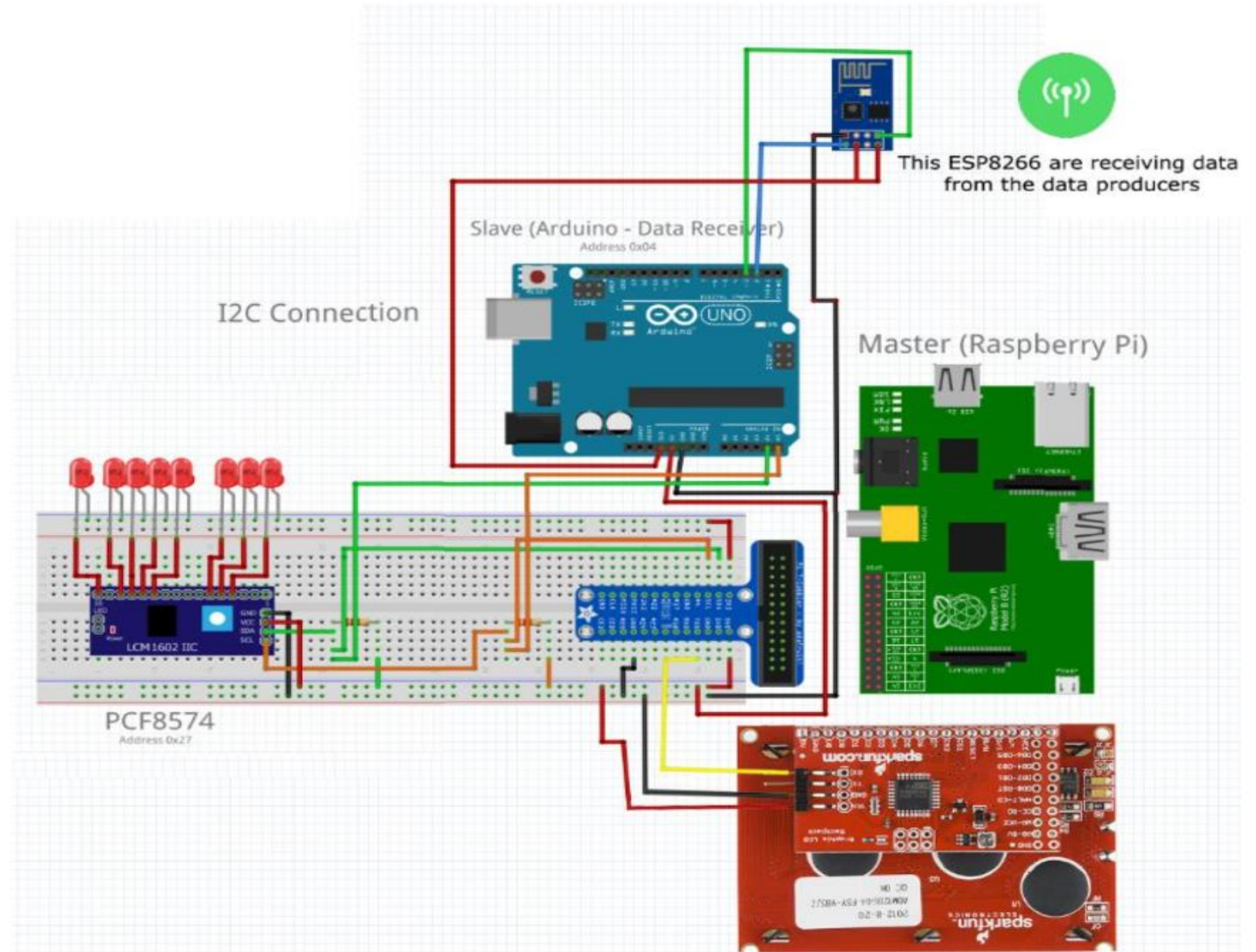
**Data Producer 2 schema:**





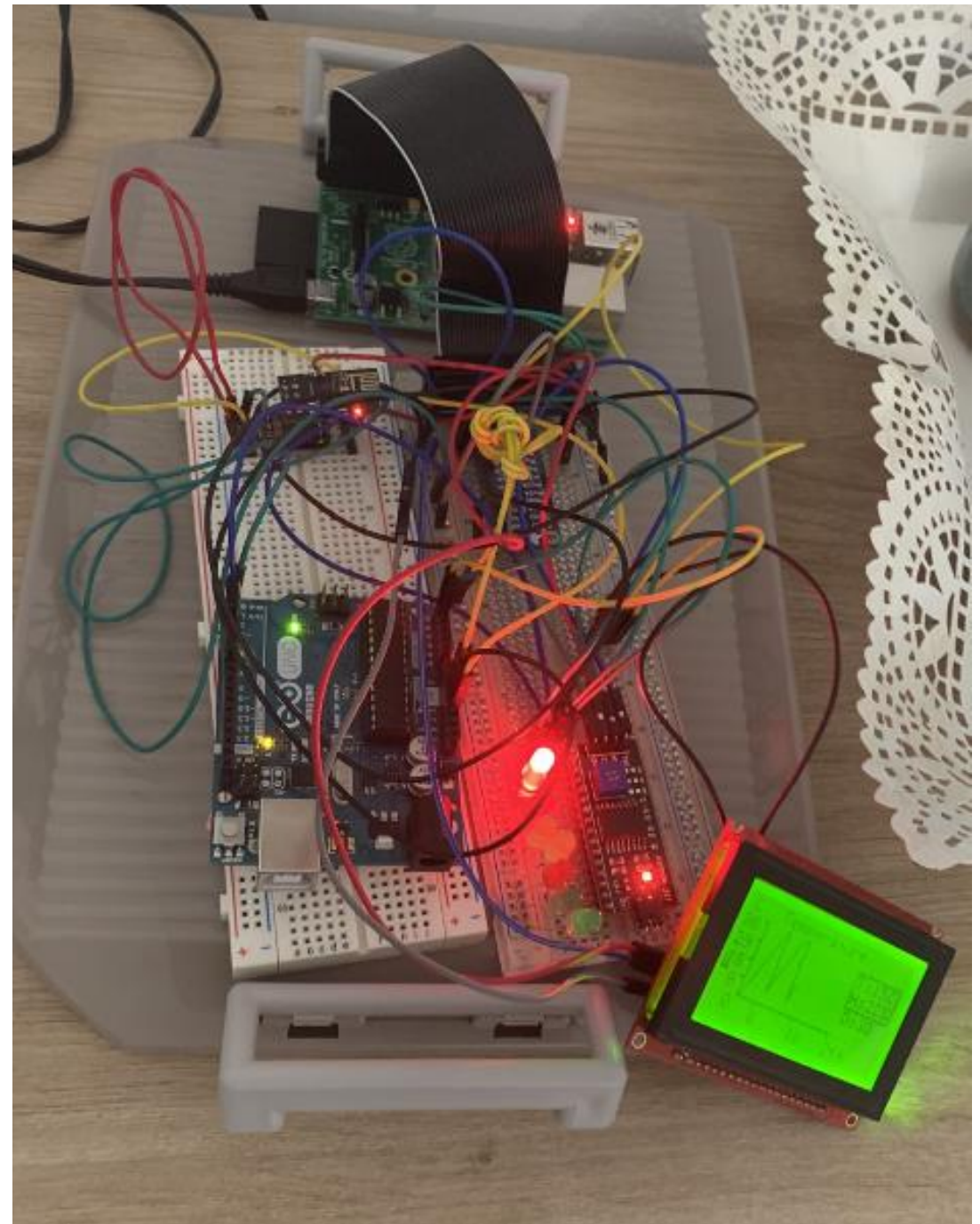
## Briefing the ongoing tasks

This is the wiring connection for the Data Receiver schema wherein, the Arduino is the Slave and Raspberry acts as the Master



## Briefing the ongoing tasks

The built Data Receiver using 2 breadboards:





# 4th Sprint

04

- Documentation related to this sprint (Sprint 4) on GitHub

## Sprint goal:

The goal of this sprint was to finally implement the last layer of the project which is the graphical representation to the final users.

Design an easy to read screen with information is also in the priorities, for the LED segment bar we are considering it to reflect the actual state of measurements made by the distance sensor

## Problems faced:

- Difficulties in programming in ChibiOS, sometimes the program would just ignore all the thread sleeps
- Hard to figure it out how to communicate with the PCF8574

Datalog_Screen_representation_documentation.pdf	Adding documentation and updated co
Final_Assembly.pdf	Adding documentation and updated co
I2C_Arduino_documentation.pdf	Added raspberry code
LCD_ChibiOS_Raspberry.pdf	Added doc and code about the LCD usin
PCF8574_LEDBAR.pdf	Added i2c and PCF Documentation
Sprint2_Delivery.pdf	Added Arduino and ESP-01 (Receiver) co
Sprint3_Delivery.pdf	Updated sprint 3 delivery
Ultrasonic_Led_bar_representation_documentation.pdf	Adding documentation and updated co



UDL  
MINF 20-21

# Q&A

Thank you for listening

