

I LEVEL

กลุ่ม BRAIN_START_IT(&brain)

สมาชิกและหน้าที่การรับผิดชอบ

1. ณัชพล จิตตุรงค์ อภิกรณ์ 6330148721 UI/UX Designer and Development
2. พชร ลีมสิริรุ่งเรือง 6330337021 Develop code on the Temperature sensor
3. ศุภกร งานตีธีรา 6330518221 Develop code on the ultrasonic and float sensor
4. ศุภกรณ์ หิรัญวิภาส 6331345121 Develop code on NodeMCU

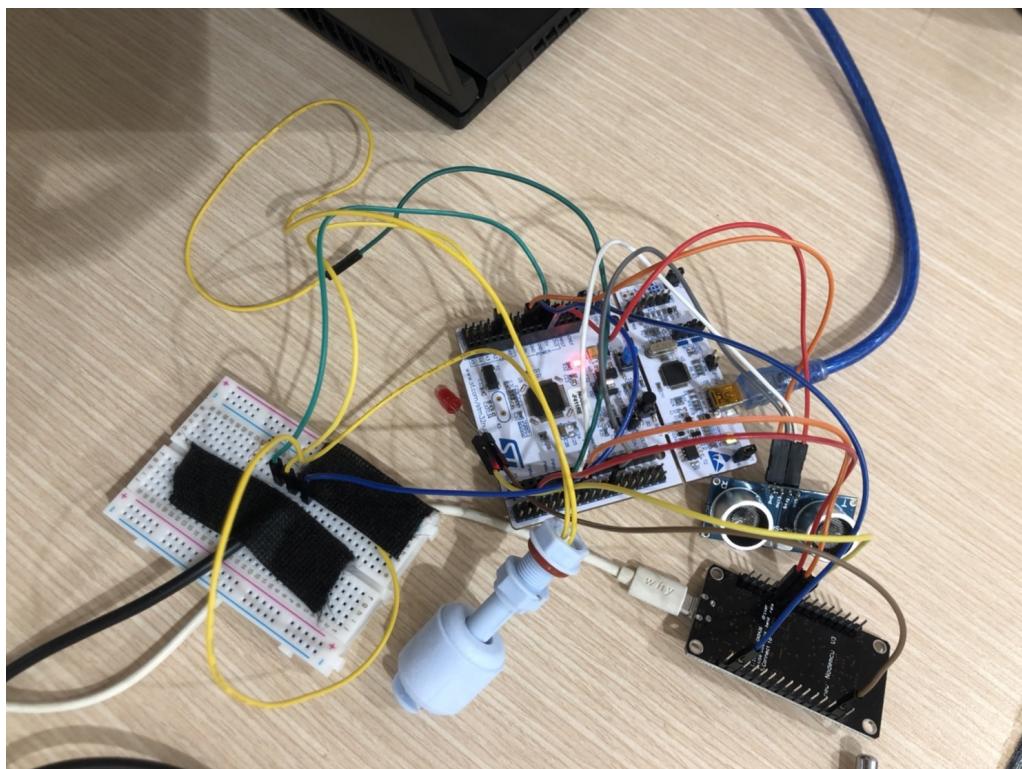
รายงานนี้เป็นส่วนหนึ่งของวิชา 2110366 ห้องปฏิบัติการระบบ
ฝังตัว (Embedded System Laboratory)
ภาคการศึกษา 2/2564
ภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

รายละเอียดโปรเจค

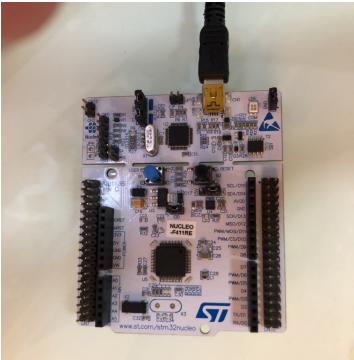
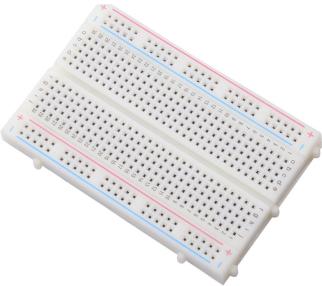
การเดินทางออกจากบ้านเพื่อไปหาที่เที่ยวนั้นอาจจะเป็นเรื่องที่สนุก แต่การเดินทางไปในที่ต่างๆไม่ว่าจะเดิน, นั่งรถไฟฟ้า หรือขับรถไปเองมันก็เหนื่อยไม่มากก็น้อย สถานการณ์ covid แบบนี้อีก การออกไปข้างนอกก็มีแต่จะสร้างความเสี่ยงถ้าเราสามารถอยู่บ้านได้อย่างสะดวกสบายมากขึ้นมันก็จะเป็นสิ่งที่ดี

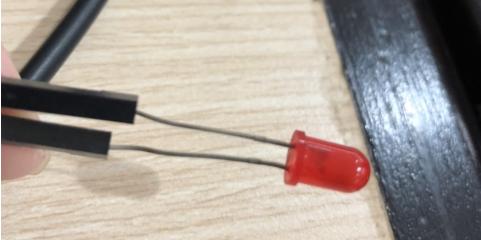
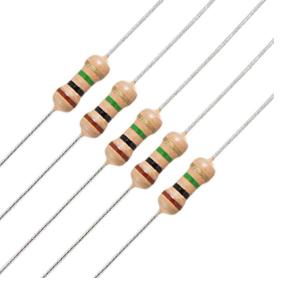
กลุ่มของเรามีจุดเด่น การอยู่บ้านแขวนมันก็เป็นเรื่องที่ดีที่จะทำให้ได้พักผ่อนอยู่บ้านได้ การแขวนน้ำที่สบายในทุกครั้งมันก็ดีต่อสุขภาพจิต จึงเกิด ilevel ขึ้น อุปกรณ์ที่สามารถวัดระดับน้ำ และอุณหภูมิของน้ำให้คงที่ตลอด ให้เราสบายใจในทุกๆ การแขวนน้ำ

อุปกรณ์จะส่งข้อมูลขึ้นไปบนเว็บไซต์อย่างต่อเนื่องเราไม่จำเป็นแต่ต้องหรือยุ่งกับอะไรก็สามารถรู้ได้มีระดับน้ำ และอุณหภูมิของน้ำเป็นไปตามที่เราต้องการ



อุปกรณ์

รูปภาพประกอบ	ชื่ออุปกรณ์
	STM32 Nucleo-F411RE
	NodeMCU ESP8266
	Ultrasonic Sensor Module (HC-SR04)
 Ref : Amazon	Breadboard

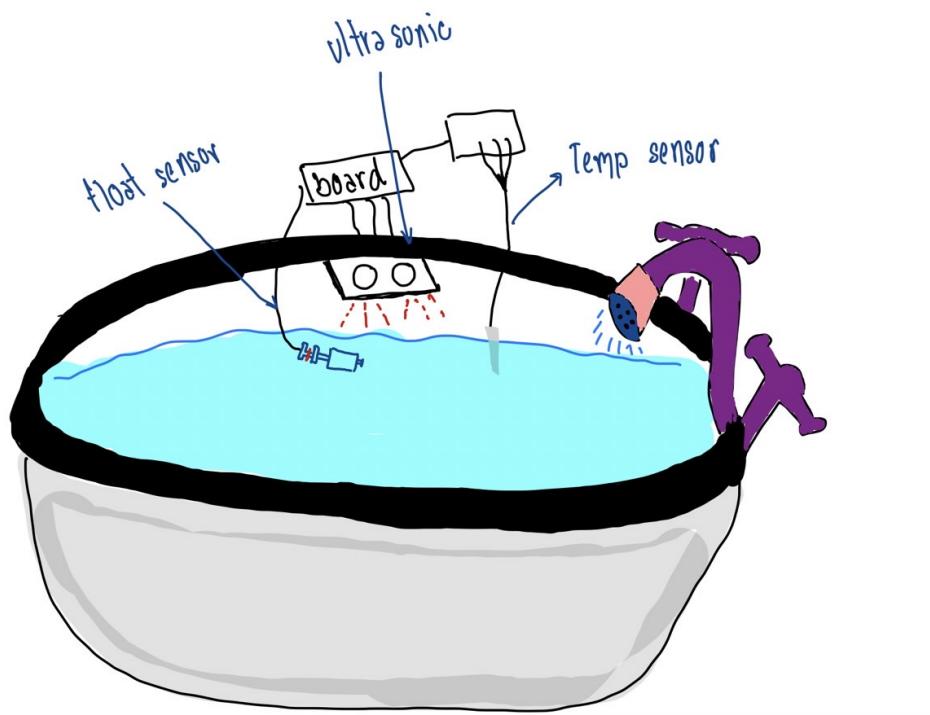
รูปภาพประกอบ	ชื่ออุปกรณ์
	LED
	Temperature sensor in water
	Float sensors
 <i>Ref: Amazon</i>	Resistor
 <i>Ref: Amazon</i>	Jumper Wire

การทำงาน

อุปกรณ์มี sensor ทั้งหมดสามตัว วัดว่า้น้ำถึงระดับที่ต้องการหรือยัง, เดิมไปแล้วเท่าไหร่ และ น้ำมีอุณหภูมิเท่าไหร่เมื่อวัดเสร็จจะส่งผ่าน STM32 ไปสู่ esp8266 และ ไปเชื่อมเข้ากับ realtime database เพื่อนำไปขึ้นสู่หน้าเว็บไซต์ ที่หน้าเว็บไซต์ ก่อนเป็นหน้าที่แสดงข้อมูลจะมีให้ใส่ชื่อและความสูงของถังเพื่อเก็บข้อมูลไว้ใช้ในครั้งอื่นได้ฯ

การติดตั้ง

เชื่อมบอร์ดเข้าคอมพิวเตอร์เพื่อรันโปรแกรมสำหรับตัว sensor และรันโปรแกรมใน Arduino เพื่อเชื่อมต่อ esp8266 เข้ากับ Wifi นำอุปกรณ์ไปติดไว้ดังนี้ float sensor ให้ทุนอยู่ในระดับที่เราถือว่าน้ำเต็มแล้ว , ultrasonic ติดไว้ติดขอบของถังหรืออ่างที่ต้องการจะวัด และ temperature sensor ให้จุ่มลงไปโดนน้ำที่เราต้องการจะวัด



Temperature sensor in water

```
uint8_t Temp_byte1, Temp_byte2;
uint16_t TEMP;

float Temperature = 0;
uint8_t Presence = 0;

void Set_Pin_Output(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin) {
    GPIO_InitTypeDef GPIO_InitStruct = { 0 };
    GPIO_InitStruct.Pin = GPIO_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOx, &GPIO_InitStruct);
}

void Set_Pin_Input(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin) {
    GPIO_InitTypeDef GPIO_InitStruct = { 0 };
    GPIO_InitStruct.Pin = GPIO_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_PULLUP; // GPIO_NOPULL / GPIO_PULLUP
    HAL_GPIO_Init(GPIOx, &GPIO_InitStruct);
}
```

ตัวแปรนอก main

- Temp_byte1 และ Temp_byte2 จะทำการเก็บค่าที่อ่านจาก data
- Temp จะทำการเก็บค่าที่ได้จากการคำนวณของ Temp_byte1 และ Temp_byte2
- Temperature ค่า float ที่แปลงจาก temp
- Presence เก็บสถานะว่าอุปกรณ์พร้อมทำงานไหม

function

- set_pin_input และ set_pin_output เป็นการ setup pin ตาม requirement ของ method ที่สั่งมา

```

#define DS18B20_PORT GPIOA
#define DS18B20_PIN GPIO_PIN_1

uint8_t DS18B20_Start(void) {
    uint8_t Response = 0;
    Set_Pin_Output(DS18B20_PORT, DS18B20_PIN); // set the pin as output
    HAL_GPIO_WritePin(DS18B20_PORT, DS18B20_PIN, 0); // pull the pin low
    usDelay(480); // usDelay according to datasheet

    Set_Pin_Input(DS18B20_PORT, DS18B20_PIN); // set the pin as input
    usDelay(80); // usDelay according to datasheet

    if (!(HAL_GPIO_ReadPin(DS18B20_PORT, DS18B20_PIN)))
        Response = 1; // if the pin is low i.e the presence pulse is detected
    else
        Response = -1;

    usDelay(400); // 480 us usDelay totally.

    return Response;
}

```

ฟังก์ชัน start จะทำการสั่งให้ sensor ทำงาน โดยทำการตีงdata จากPin_low และอ่านPin ว่า pulse นั้นส่งโดย sensor หรือป่าว หากนั้นจะส่งค่าresponse ไปว่าได้เริ่มการทำงานของsensor หรือไม่

```

void DS18B20_Write(uint8_t data) {
    Set_Pin_Output(DS18B20_PORT, DS18B20_PIN); // set as output

    for (int i = 0; i < 8; i++) {
        if ((data & (1 << i)) != 0) // if the bit is high
        {
            // write 1

            Set_Pin_Output(DS18B20_PORT, DS18B20_PIN); // set as output
            HAL_GPIO_WritePin(DS18B20_PORT, DS18B20_PIN, 0); // pull the pin LOW
            usDelay(1); // wait for 1 us

            Set_Pin_Input(DS18B20_PORT, DS18B20_PIN); // set as input
            usDelay(50); // wait for 60 us
        }

        else // if the bit is Low
        {
            // write 0

            Set_Pin_Output(DS18B20_PORT, DS18B20_PIN);
            HAL_GPIO_WritePin(DS18B20_PORT, DS18B20_PIN, 0); // pull the pin LOW
            usDelay(50); // wait for 60 us

            Set_Pin_Input(DS18B20_PORT, DS18B20_PIN);
        }
    }
}

uint8_t DS18B20_Read(void) {
    uint8_t value = 0;

    Set_Pin_Input(DS18B20_PORT, DS18B20_PIN);

    for (int i = 0; i < 8; i++) {
        Set_Pin_Output(DS18B20_PORT, DS18B20_PIN); // set as output

        HAL_GPIO_WritePin(DS18B20_PORT, DS18B20_PIN, 0); // pull the data pin LOW
        usDelay(1); // wait for > 1us

        Set_Pin_Input(DS18B20_PORT, DS18B20_PIN); // set as input
        if (HAL_GPIO_ReadPin(DS18B20_PORT, DS18B20_PIN)) // if the pin is HIGH
        {
            value |= 1 << i; // read = 1
        }
        usDelay(50); // wait for 60 us
    }

    return value;
}

```

function

- write จะทำการเช็ค data ในแต่ละบิตก่อนว่าบิตนั้นเป็น high อยู่หรือไม่ ถ้า high จะทำการเขียน 1 และ low จะเขียน 0 และสั่งให้ pinlow
- read จะทำการอ่าน data ในแต่ละบิตและก่อนที่จะอ่านจะต้องทำการ setup pin ให้พร้อมอ่านก่อนเสมอ และทำการอ่านแต่ละบิตโดย high คือ 1 และ low คือ 0

```
Presence = DS18B20_Start();
HAL_Delay(1);
DS18B20_Write(0xCC); // skip ROM
DS18B20_Write(0x44); // convert t
HAL_Delay(800);

Presence = DS18B20_Start();
HAL_Delay(1);
DS18B20_Write(0xCC); // skip ROM
DS18B20_Write(0xBE); // Read Scratch-pad

Temp_byte1 = DS18B20_Read();
Temp_byte2 = DS18B20_Read();
TEMP = (Temp_byte2 << 8) | Temp_byte1;
Temperature = (float) TEMP / 16;

char temp[50];
sprintf(temp, "F:%d|S:%06.2f|T:%.2f\r\n", Float, distance, Temperature);
HAL_UART_Transmit(&huart2, (uint8_t*) temp, strlen(temp), HAL_MAX_DELAY);

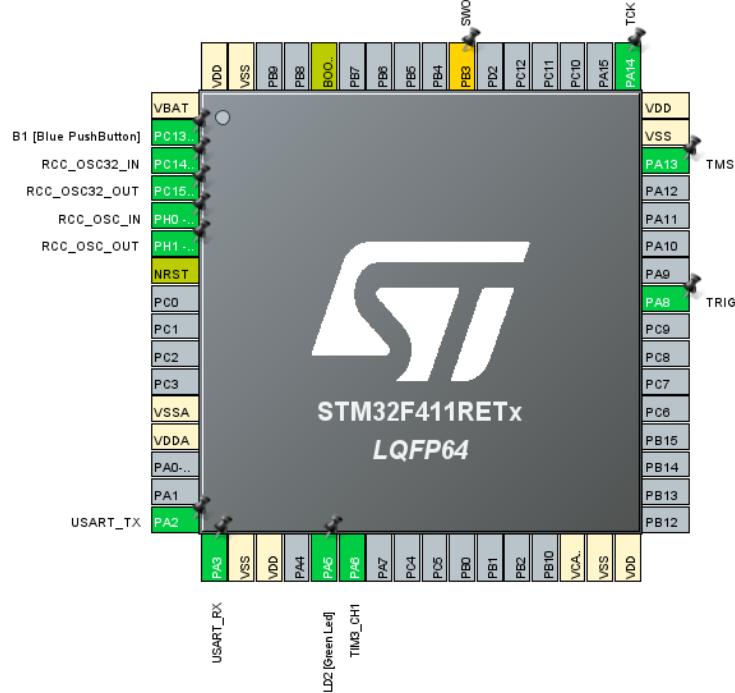
sprintf(buffer, "F:%d|S:%06.2f|T:%.2f", Float, distance, Temperature);
HAL_UART_Receive_IT(&huart1, &data, sizeof(data));
}

/* User code end */
```

การทำงานใน main

เริ่มต้นจะทำการ สั่ง sensor ให้เริ่มทำงาน ต่อมาจะ skip rom และ สั่งเขียน Scratchpad เพื่อเป็นการ setup ก่อนที่ทำการอ่านค่า จาก data และอ่าน data จากนั้นทำการคำนวนและส่งออกไป

Ultrasonic sensor



ขาทั้ง 4 ของ ultrasonic HC-SR04 ถูกต่อดังนี้

- ขา Vcc ถูกต่อเข้ากับ 5V ของบอร์ด
 - ขา Trig ถูกต่อเข้ากับ PA8 ของบอร์ด เป็น GPIO_Output
ตั้ง Label ไว้ว่า TRIG
 - ขา Echo ถูกต่อเข้ากับเข้ากับ PA6 ของบอร์ด ใช้ร่วมกับ
TIM3 ch1
 - ขา Gnd ถูกต่อเข้ากับเข้ากับ Ground ของบอร์ด

มีการใช้ timers ส่องตัว

- TIM3 ใช้สำหรับตรวจสอบสัญญาณเข้าลงของ Echo ของ ultrasonic ด้วยการ interrupt และ input capture direct mode
 - TIM4 ใช้สำหรับการสร้าง delay แบบ microsecond

```

    uint8_t icFlag = 0;
    uint8_t captureIdx = 0;
    uint32_t edge1Time = 0, edge2Time = 0;
    float distance;

```

ตัวแปรนอก main

- icFlag ตรวจสอบว่าสัญญาณ Echo ขึ้นและลงแล้ว
- captureIdx ตรวจสอบว่าสัญญาณ Echo ขึ้นหรือลง
- edge_Time เวลาที่ได้จาก TIM3 ณ ตอนขึ้นและลงของสัญญาณ Echo
- distance เก็บค่าระยะทางที่คำนวณได้

```

void usDelay(uint32_t uSec) {
    if (uSec < 2)
        uSec = 2;
    TIM4->ARR = uSec - 1;
    TIM4->EGR = 1;
    TIM4->SR &= ~1;
    TIM4->CR1 |= 1;
    while ((TIM4->SR & 0x0001) != 1)
        ;
    TIM4->SR &= ~(0x0001);
}

void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim) {
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);

    if (captureIdx == 0) {
        edge1Time = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1);
        captureIdx = 1;
    } else if (captureIdx == 1) {
        edge2Time = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1);
        captureIdx = 0;
        icFlag = 1;
    }
}

```

function นอก main

- usDelay delay เป็น microsecond ตามเลขที่รับมาโดยการใช้ TIM4
- HAL_TIM_IC_CaptureCallback เก็บค่าจาก TIM3 เมื่อเกิดการ callback แบบ IC

เริ่มการเขียนโค้ดใน while loop ของ main เพื่อทำให้ sensor ทำงานตามลำดับดังนี้

```
HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_RESET);  
usDelay(3);
```

1. สร้างสัญญาณ low ก่อนเริ่มการทำงาน

```
HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_SET);  
usDelay(10);  
HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_RESET);
```

2. สร้างสัญญาณ high 10 microsec และทำสัญญาณ low เพื่อเริ่มการทำงานของ sensor

```
HAL_TIM_IC_Start_IT(&htim3, TIM_CHANNEL_1);  
// wait for ic flag  
uint32_t startTick = HAL_GetTick();  
do {  
    if (icFlag)  
        break;  
} while ((HAL_GetTick() - startTick) < 500);  
icFlag = 0;  
HAL_TIM_IC_Stop_IT(&htim3, TIM_CHANNEL_1);
```

3. เริ่ม IT ของ TIM3 เพื่อจับสัญญาณของ Echo ทำงานกว่าจะครบสัญญาณ Echo หรือเกินเวลา หยุด IT

```
if (edge2Time > edge1Time) {  
    distance = ((edge2Time - edge1Time) + 0.0f) * 0.0171821;  
} else {  
    distance = 0.0f;  
}
```

4. คำนวณค่าที่วัดได้ออกเป็นเซนติเมตร

Float sensor

สายสองเส้นที่ออกมายจาก sensor เส้นหนึ่งต่อเข้ากับ ground
ของบอร์ด อีกเส้นต่อเข้ากับ PB10 label ไว้ว่า FLOAT ที่เป็น
GPIO_Input Pull-up

ส่วนของโค้ด

```
if (HAL_GPIO_ReadPin(FLOAT_GPIO_Port, FLOAT_Pin) == GPIO_PIN_RESET) {  
    Float = 0;  
} else {  
    Float = 1;  
}
```

เป็นการตรวจสอบว่า port อยู่ในรูปแบบไหน RESET หมายถึง น้ำ
ไม่ถึงระดับ SET ก็ตรงข้าม

การต่อ STM กับ esp8266

การส่งข้อมูลจาก sensor ที่ต่ออยู่กับบอร์ดเข้า data base ผ่าน esp8266 มีการทำงานในส่วนของการเชื่อมอุปกรณ์ดังนี้

การเชื่อม esp8266 เข้ากับบอร์ด

- D7 (RX ของ esp8266) ต่อเข้ากับ PA9 (TX ของ USART1 ของบอร์ด)
- D8(TX ของ esp8266) ต่อเข้ากับ PA10 (RX ของ USART1 ของบอร์ด)
- GND ของ esp8266 ต่อเข้ากับ Ground ของบอร์ด (ground ที่ตำแหน่งไหนก็ได้)
- D6 ต่อเข้ากับ Ground ของ esp8266 (ground ที่ตำแหน่งไหนก็ได้) ใช้ในการทำงานของ LED

การส่งข้อมูลใช้ USART1 ด้วย baud rate 9600 Bits/s และมีการเปิด interrupt ของ USART1

ส่วนของโค้ด

```
sprintf(buffer, "F:%d|S:%06.2f|T:%.2f", Float, distance, Temperature);  
HAL_UART_Receive_IT(&huart1, &data, sizeof(data));
```

ส่วนการจัด format ของข้อมูลให้อยู่ในรูป F:x|S:xxx.xx|T:xx.xx เมื่อ x คือตัวเลข ใช้ sprintf ใส่เข้าไปในตัวแปร buffer ที่เป็น array ของ char พร้อมเรียก function HAL_UART_Receive_IT เพื่อไปเรียก HAL_UART_RxCpltCallback ต่อ

```
char data[4] = "";
char buffer[21] = "F:0|S:123.45|T:67.89"; // F:x|S:xxx.xx|T:xx.xx
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) {
    if (huart == &huart1) {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
        HAL_UART_Transmit(&huart2, data, sizeof(data), 1000);
        HAL_UART_Transmit(&huart1, buffer, sizeof(buffer), 1000);
    }
    HAL_UART_Receive_IT(&huart1, &data, sizeof(data));
}
```

โดยทำงานก็ต่อเมื่อได้รับคำสั่ง ready to send จาก NodeMCU

NodeMCU

```
void setup() {  
    pinMode(rxPin, INPUT);  
    pinMode(txPin, OUTPUT);  
    pinMode(LED, OUTPUT);  
  
    Serial.begin(115200);  
    NodeSerial.begin(9600);  
    Serial.println();  
    Serial.println("NodeMCU/ESP8266 Run");  
    connectWifi();  
    Firebase.begin(FIREBASE_HOST, FIREBASE_KEY);  
}
```

setup สำหรับการรับข้อมูลที่ส่งจาก board nucleo โดย set RX pin และ TX pin สำหรับการรับข้อมูลและส่งข้อมูลตามลำดับโดยใช้ buad rate เป็น 9600 พร้อมเชื่อมต่อ wifi และเชื่อมต่อกับ firebase

```
void connectWifi() {  
    Serial.begin(115200);  
    Serial.println(WiFi.localIP());  
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);  
    Serial.print("connecting");  
    while (WiFi.status() != WL_CONNECTED) {  
        Serial.print(".");  
        delay(500);  
    }  
    Serial.println();  
    Serial.print("connected: ");  
    Serial.println(WiFi.localIP());  
}
```

เชื่อมต่อ wifi โดยใช้ SSID กับ password

```

    bool is_full = false;
    float water_level = 0.0;
    float temperature = 0.0;

    void loop() {
        setDataValue();
        updateFirebaseData();
        delay(1500); // delay
    }

```

กำหนดค่าต่างๆของสำหรับการรับข้อมูลจาก board Nucleo และส่งไปยัง firebase พร้อมสร้าง loop ในการรับค่าจาก board Nucleo และ update ค่าไปยัง firebase (delay 1.5 seconds)

```

void setDataValue() {
    String readyToSend = "OK";
    NodeSerial.println(readyToSend); // len is 4
    delay(100);

    String recieveText = "Recieve: "; // len is 10. Reciving -> F:x|S:xxxx.xx|T:xx.xx
    while (NodeSerial.available())
    {
        recieveText += (char)NodeSerial.read();
    }

    is_full = recieveText.substring(11, 12) == "1";
    water_level = recieveText.substring(15, 21).toFloat();
    temperature = recieveText.substring(24, 29).toFloat();

    if(is_full) digitalWrite(LED, HIGH); // TODO change it to is_full
    else digitalWrite(LED, LOW);

    Serial.println(
        "IsFull: " + String(is_full) +
        ", WaterLevel: " + String(water_level) +
        ", Temperature: " + String(temperature)
    );
}

```

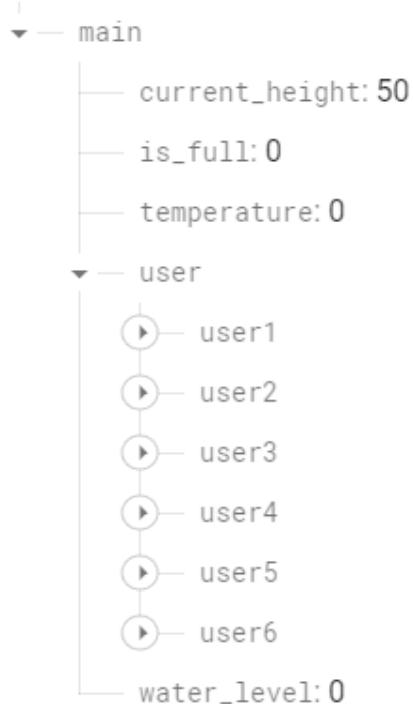
นำค่าที่รับมาจาก board Nucleo มาใส่ในตัวแปรที่กำหนดไว้ข้างต้น (temperature, water_level, is_full) โดยมีคำสั่ง readyToSend ส่งไปยัง board Nucleo เพื่อเป็นสัญญาณว่าให้ส่งค่ามาที่ esp8266 ได้แล้ว (ในกรณีที่น้ำเต็มแล้วก็จะมีสัญญาณเตือนเป็น LED ไว้)

```
void updateFirebaseData() {  
    Firebase.setFloat(firebaseData, "/main/temperature", temperature);  
    Firebase.setFloat(firebaseData, "/main/water_level", water_level);  
    Firebase.setFloat(firebaseData, "/main/is_full", is_full);  
}
```

update ค่าต่างๆ ใน firebase ตามตัวแปรที่กำหนดไว้ข้างต้น
(temperature, water_level, is_full)

Web Application

<https://test-95a70-default-rtdb.firebaseio.com/main.json>



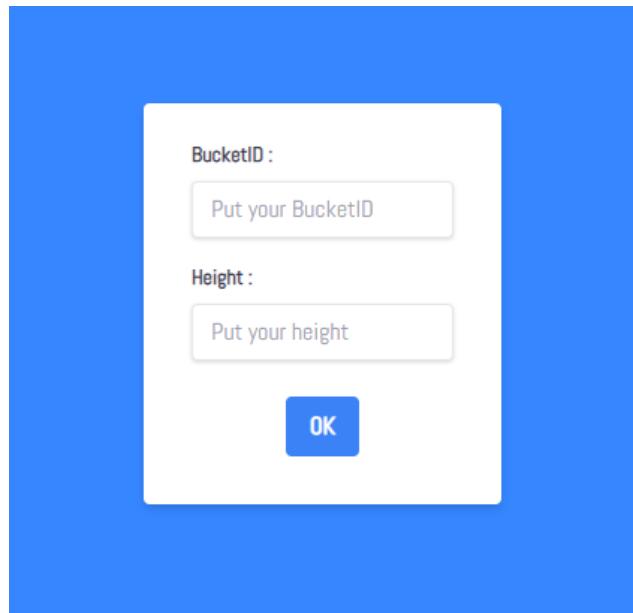
ส่วนของ Database ใช้เป็น Realtime Database Firebase เก็บข้อมูล เมื่อข้อมูลมีการเปลี่ยนแปลงหน้าเว็บไซต์จะเปลี่ยนแปลงไปด้วย

```
async function writeUserData(userId, height) {
  const reference = ref(db, 'main/user/' + userId);
  await set(reference, {
    level: Number(height)
  });
}

async function updateCurrentHeight(height){
  const reference = ref(db, 'main');
  await update(reference, {
    current_height: Number(height)
  })
}
```

มีฟังก์ชัน writeUserData เพื่อเพิ่มชื่อและความสูงของถังน้ำที่ต้องการจะวัดและ และ updateCurrentHeight เพื่อ update ค่าความสูง ณ ปัจจุบันของถังที่กำลังวัดอยู่

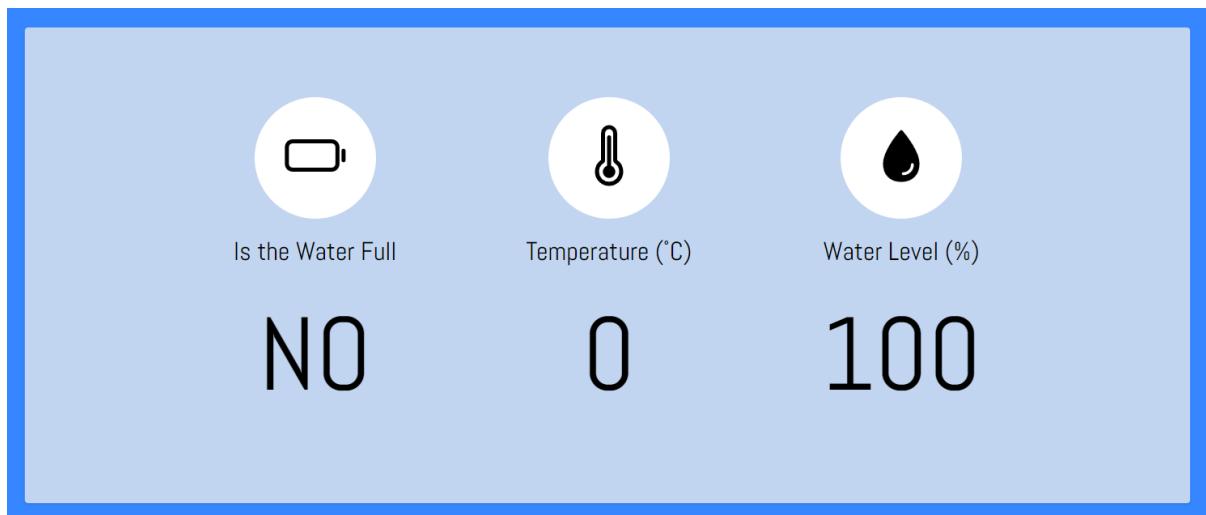
```
async function addbucket(event) {
  event.preventDefault();
  const reference = ref(db, 'main/user');
  const userId = document.getElementById('userId').value;
  const height = document.getElementById('height').value;
  get(child(reference, userId)).then(snapshot => {
    if (snapshot.exists()) {
      if(height > 0){
        writeUserData(userId, height);
        updateCurrentHeight(height);
        window.location.href = "index.html";
      }else if(height == 0){
        updateCurrentHeight(snapshot.val().level);
        window.location.href = "index.html";
      }else{
        window.location.href = "home.html";
        alert("Height must be a positive number.");
      }
    } else {
      if(height > 0 ){
        writeUserData(userId, height);
        updateCurrentHeight(height);
        window.location.href = "index.html";
      }else if(height == 0){
        window.location.href = "home.html";
        alert("You need to set the height.");
      }else{
        window.location.href = "home.html";
        alert("Height must be a positive number.");
      }
    }
  }).catch(error => {
    console.error(error);
  });
}
```



โดยในหน้าแรกจะเป็นหน้าฟอร์มสำหรับใส่ชื่อถังน้ำและความสูง
ซึ่งจำเป็นที่จะต้องใส่ BucketID ซึ่งหากเรามี BucketID อยู่แล้ว เรา
สามารถที่จะสามารถไปยังหน้าต่อไปได้แล้ว (เนื่องจากมีค่าที่ set ไว้
แล้ว) โดยหากเราใส่ค่า Height เข้าไปจะเป็นการ set ค่าของถังน้ำใหม่
และเราสามารถที่จะเพิ่ม BucketID ใหม่ได้โดยที่ต้องใส่ค่า Height ด้วย

```
onValue(reference, (snapshot) => {
  const data = snapshot.val();
  if (data.is_full == 0) {
    isFull.innerText = 'NO';
    icon.name = 'battery-dead-outline';
  } else {
    isFull.innerText = 'YES';
    icon.name = 'battery-full-outline';
  }
  temp.innerText = data.temperature;
  level.innerText = (data.current_height - data.water_level)/data.current_height * 100 ;
});
```

ในการดึงข้อมูลมาเราจะใช้การ snapshot ออกมาย ซึ่งเมื่อข้อมูลใน
database เกิดการเปลี่ยนแปลง จะเปลี่ยนค่าบน website ในทันที



โดยหน้าเว็บจะแสดงผล 3 อย่าง

- แสดงน้ำ ณ ขณะนั้นเต็มหรือยัง
- แสดงอุณหภูมิ ณ ขณะนั้นเท่ากับกีองศา
- แสดงระดับน้ำ ณ ขณะนั้นเป็นกีเบอร์เซ็นต์ของถัง