

Embedded system project

Bangkokian's safe bus stop

6531326421 ปิยุषิ ไสวดี

651316121 ณัชชา หังสพฤกษ์

6531317821 ดาไล โฉติศาลิกก์ เชอร์เรนเจร์กากล

6531302321 ก้องภพ โภชนสมบูรณ์

15th May, 2024

ระบบขนส่งสาธารณะมีบทบาทสำคัญในชีวิตประจำวันของคนกรุงเทพ การเข้าถึงข้อมูลที่ถูกต้องและทันท่วงที่เกี่ยวกับสถานะของระบบขนส่งสาธารณะ เช่น จำนวนรถเมล์ว่าง จำนวนผู้โดยสาร และสภาพอากาศ เป็นสิ่งที่จำเป็นสำหรับการตัดสินใจในการเดินทาง

อย่างไรก็ตาม ปัญหาไฟตับที่เกิดขึ้นบ่อยครั้ง สร้างความไม่สะดวกและความเสียงต่อความปลอดภัยของผู้โดยสาร รายงานฉบับนี้ นำเสนอผลการวิจัยและพัฒนาระบบฟังสำหรับตรวจสอบไฟส่องสว่าง แสดงข้อมูลจำนวนรถเมล์ว่าง วัดค่าอุณหภูมิ ความชื้น ระบบนี้ถูกออกแบบมาเพื่อติดตั้งที่ป้ายรถเมล์ โดยสามารถตรวจสอบไฟส่องสว่าง หากไฟตับ หลอดไฟจะติดสว่างโดยอัตโนมัติ นอกจากนี้ ยังมีจอ LED แสดงข้อมูลจำนวนรถเมล์ว่างที่ฝาหน้าป้ายรถเมล์นี้

ระบบนี้ยังมีเห็นเชอร์วัดอุณหภูมิและความชื้น ซึ่งจะแสดงค่าบนจอ LED เช่นกัน ข้อมูลเหล่านี้ช่วยให้ผู้ใช้สามารถตัดสินใจได้ว่าควรขึ้นรถเมล์ทันทีใน หรือควรปรับเปลี่ยนแผนการเดินทางอย่างไร

โดย project นี้สร้างแก้ปัญหา **ป้ายรถเมล์มีข้อมูลและสว่างปลอดภัยทุกป้าย จาก chadchart.com/policy**

วัตถุประสงค์

วัตถุประสงค์ของโครงการวิจัยและพัฒนาครั้งนี้ ได้แก่

- พัฒนาระบบฟังสำหรับตรวจสอบไฟส่องสว่าง
- แสดงข้อมูลจำนวนรถเมล์ว่างบนจอ LED
- วัดค่าอุณหภูมิและความชื้น
- แสดงค่าอุณหภูมิและความชื้นบนจอ LED

ประโยชน์

- ผู้ใช้สามารถรับทราบข้อมูลจำนวนรถเมล์ว่างที่ฝาหน้าป้ายรถเมล์นี้
- ผู้ใช้สามารถรับทราบสภาพอากาศที่ป้ายรถเมล์
- ข้อมูลเหล่านี้ช่วยให้ผู้ใช้สามารถตัดสินใจได้ว่าควรขึ้นรถเมล์ทันทีใน หรือควรปรับเปลี่ยนแผนการเดินทางอย่างไร
- ระบบนี้ช่วยเพิ่มความปลอดภัยของผู้ใช้ โดยการติดตั้งหลอดไฟเมื่อไฟตับ

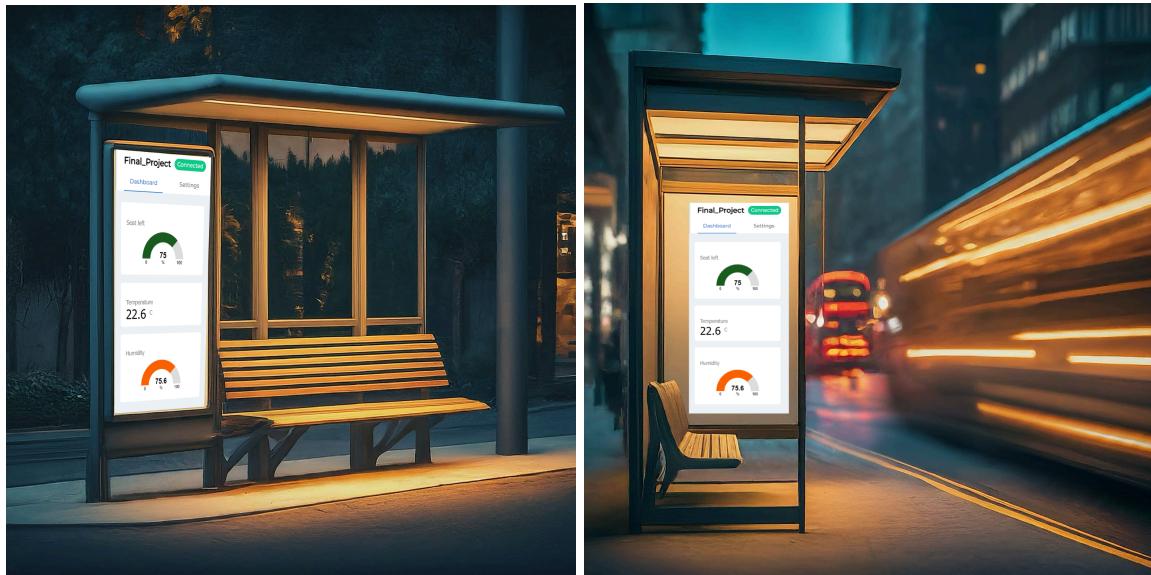
อุปกรณ์ที่ใช้

- Stm32 f411 nucleo-64
- nodeMCU esp8266
- Dht22 am2302 humidity and temperature sensor
- Light dependent resistor and 2xLED
- 4xButton

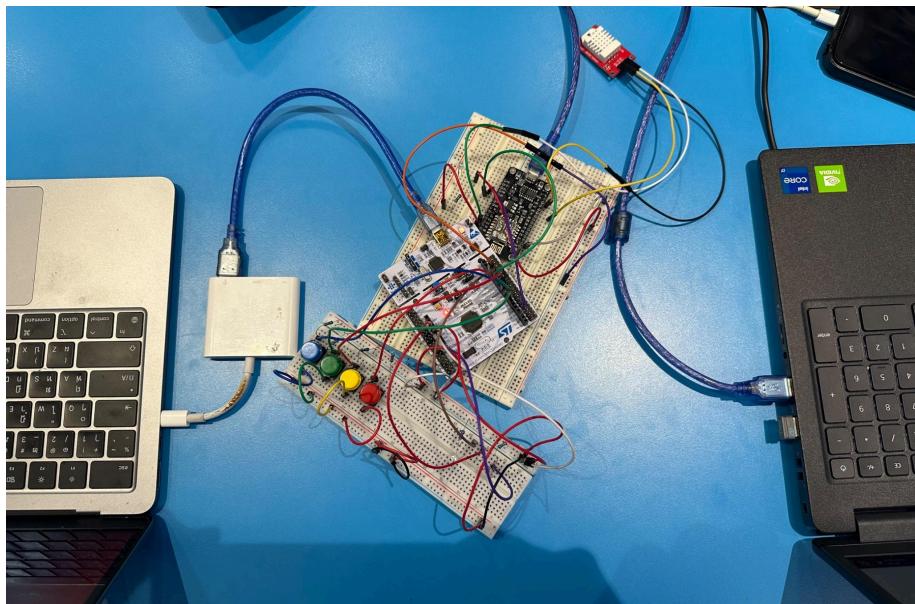
Overview of a project

“ป้ายรถเมล์มีข้อมูล และสว่างตลอดวันทุกปัจจัย”

- AI generated with real interface



- Actual circuit (including sensors, stm32, nodeMCU)



6531326421 ปี่ยรุณิ ไสวี : Dht22 am2302 humidity and temperature sensor

Responsibility : implementing the software for connecting dht22 to nodeMCU (using Arduino IDE) and also physical connection (as shown below)



Software (code) can be found here : https://github.com/Ntchah/Embedded_Lab

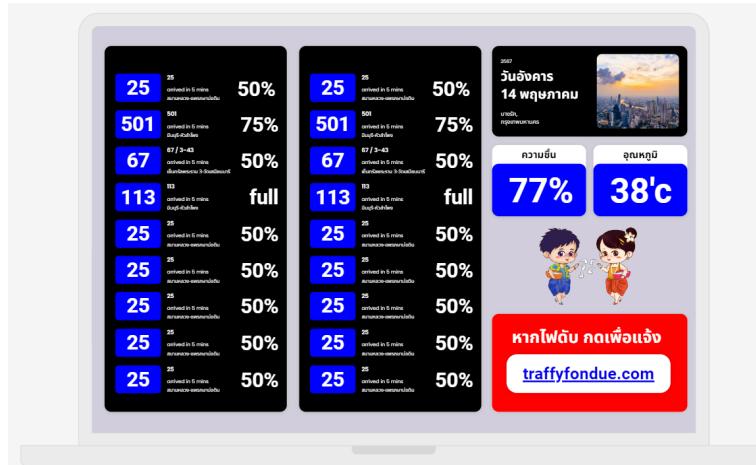
These are some example of the code implemented on nodeMCU using Arduino IDE along with provided dht22 library

```
32 void loop() {
33     // Wait a few seconds between measurements.
34     delay(2000);
35
36     // Reading temperature or humidity takes about 250 milliseconds!
37     // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
38     float h = dht.readHumidity();
39     // Read temperature as Celsius (the default)
40     float t = dht.readTemperature();
41     // Read temperature as Fahrenheit (isFahrenheit = true)
42     float f = dht.readTemperature(true);
43
44     // Check if any reads failed and exit early (to try again).
45     if (isnan(h) || isnan(t) || isnan(f)) {
46         Serial.println(F("Failed to read from DHT sensor!"));
47         return;
48     }
```

Responsibility : design user interface for the LED at the bus stop and implementing a software to connect stm32f411 (stm32 cube IDE) and nodeMCU esp8266 (Arduino IDE) using UART (Universal Asynchronous Receive-Transmit) as shown below

- UI part

The design can be accessed here : <https://app.uizard.io/p/3914231d>

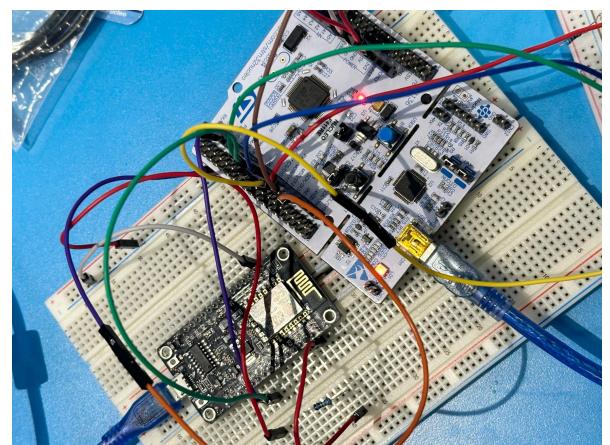


- UART part

The code can be accessed here : https://github.com/Ntchah/Embedded_Lab

Shown below is the physical connection between stm32 and nodeMCU the procedure are

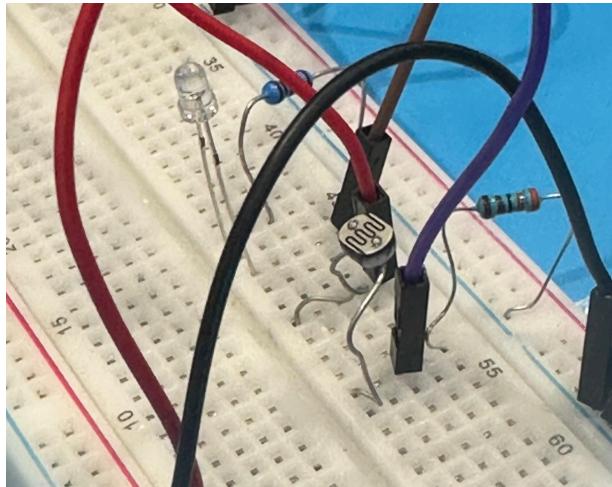
1. Download library and implement a program
to use Software Serial on Arduino IDE
2. Implement HAL_UART_Transmit to send data
Received from sensors to Node MCU
3. Physical connection
 - ESP8266 Ground to Stm32 Ground
 - ESP8266-RX (D7) to STM32-TX (PA9)
 - ESP8266-TX (D8) to STM32-RX (PA10)



6531317821 ดาไล โชคศิลป์กร เข้อเรนจาร์ก้าล : LDR sensor, button ,NETPIE assistant

Responsibility : implementing a software on STM32 cube IDE to receive a data from LDR sensor and turn on LED light if it's dark (resemble a situation at a bus stop) and to receive a data from buttons.If pressed,it means the occupied seat (resemble a situation in the bus). Moreover,every physical connection between the sensors and stm32 board is what I am responsible for.

- LDR sensor

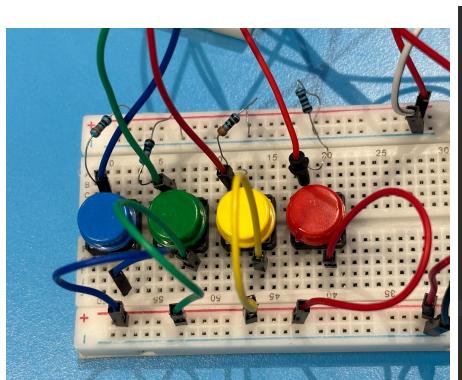


```

HAL_ADC_Start(&hadc1);
if (HAL_ADC_PollForConversion(&hadc1, 1000) == HAL_OK) {
    adcval = HAL_ADC_GetValue(&hadc1);
    sprintf(buf, "sd\r\n" adcval);
    HAL_UART_Transmit(&huart2, buf, strlen(buf), 100);
}
if (adcval > 1000)
    TIM2->CCR1 = 0;
else
    TIM2->CCR1 = 100;
    
```

- Button sensor

To illustrate,each button represent each seat on the bus.Since we have 4 buttons.The LED board at the bus stop will show 0%-25%-50%-100% respectively



```

TIM2->CCR1 = 100;
if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_6) == GPIO_PIN_SET)
    val1 = 1;
if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_7) == GPIO_PIN_SET)
    val2 = 1;
if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == GPIO_PIN_SET)
    val3 = 1;
if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_7) == GPIO_PIN_SET)
    val4 = 1;
if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_6) == GPIO_PIN_RESET)
    val1 = 0;
if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_7) == GPIO_PIN_RESET)
    val2 = 0;
if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == GPIO_PIN_RESET)
    val3 = 0;
if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_7) == GPIO_PIN_RESET)
    val4 = 0;

int tot = 100 - (val1 + val2 + val3 + val4) * 25;
sprintf(buf, "%d\r\n", tot);
HAL_UART_Transmit(&huart2, buf, strlen(buf), 1000);
HAL_UART_Transmit(&huart1, buf, strlen(buf), 1000);
HAL_Delay(500);
    
```

- NETPIE assistance : implement MQTT protocol , JSON file, and every step supervised

6531302321 ก้องภพ โภชนสมบูรณ์ : Project Manager, NETPIE assistant

Responsibility : Establish communication between NETPIE and nodeMCU. By implementing wifi connection software, nodeMCU is able to establish the connection leading to the ability to create a real-time interface. Moreover, I have delegated tasks to everyone in my group, as they have reported above, and made sure that everything is going correctly as we planned.

The link of the interface : <https://portal.netpie.io/p/P151903426209/dash/DH869881414853>

The interface below is implemented using NETPIE

