

# Embedded System Laboratory

## Final Project Report

Topic: Prohibited Parking Alert System

จัดทำโดย

6532035021 นาย ชญาณิน คงเสรีกุล

6532040021 นาย ชนาธิป พัฒนาเพ็ญ

6532100021 นาย ธีภพ เล้าพรพิชยานุวัฒน์

6532142421 นาย ภูวิพัทธ์ สิงหาร

# Introduction

## Background Information

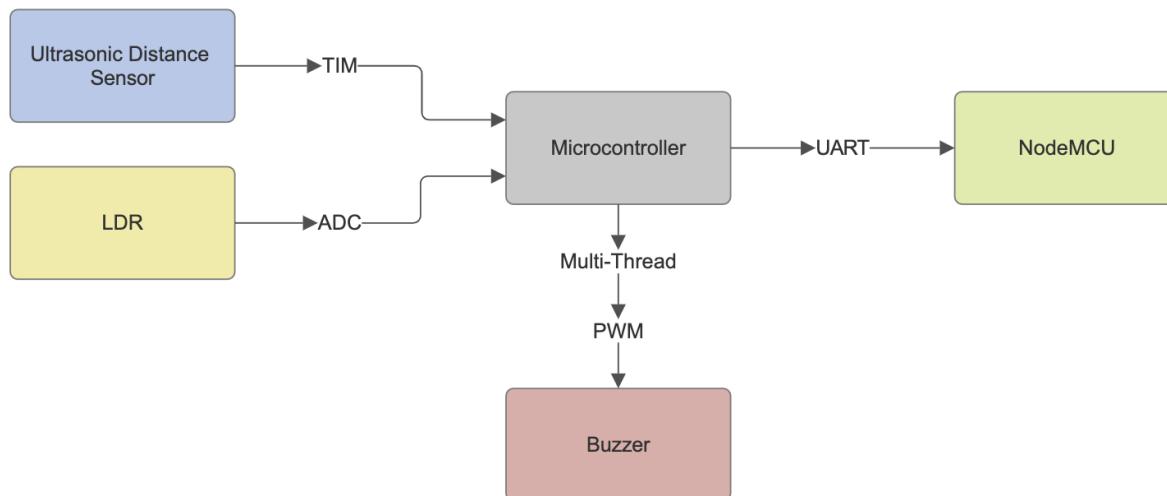
ปัญหาการจอดรถในที่ห้ามจอดหรือการจอดรถริมถนนที่ทำให้เสียช่องจราจรไป เป็นหนึ่งในปัญหาที่ทำให้การจราจรในภาพรวมติดชัด เช่น ในพื้นที่ถนน 2 เลน ที่มีเลนหนึ่งมีรถจอดริมทาง ทำให้เลนถนนที่รถเคลื่อนไหวได้เหลือเพียงเลนเดียว ซึ่งถ้าสามารถลดหรือแก้ปัญหานี้ได้ก็จะช่วยทำให้การจราจรในภาพรวมติดชัดน้อยลง ตามนโยบาย “บริหารจัดการจราจรด้วยระบบอัจฉริยะ (ITMS) เพื่อบริหารจัดการจราจรทั่วโลกข่ายและการดูแลนิยมจราจร”

โดยหนึ่งในวิธีที่จะช่วยแก้ปัญหานี้คือ การตรวจจับและแจ้งเตือนเมื่อมีการจอดรถในบริเวณที่ห้ามจอด พร้อมทั้งมีการรายงานต่อไปเพื่อตักเตือนหรือลงโทษผู้ฝ่าฝืนการจอดรถในที่ห้ามจอดต่อไป

## Project Scope

การตรวจจับรถที่จอดในที่ห้ามจอดสามารถทำได้โดยใช้เซ็นเซอร์วัดระยะทางควบคู่กับเซ็นเซอร์วัดความสว่างของแสง พร้อมทั้งมีการแจ้งเตือนด้วยเสียงผ่านลำโพง และทำการรายงานข้อมูลเพื่อแสดงผลบนเว็บไซต์

## Methodology



Preliminary design

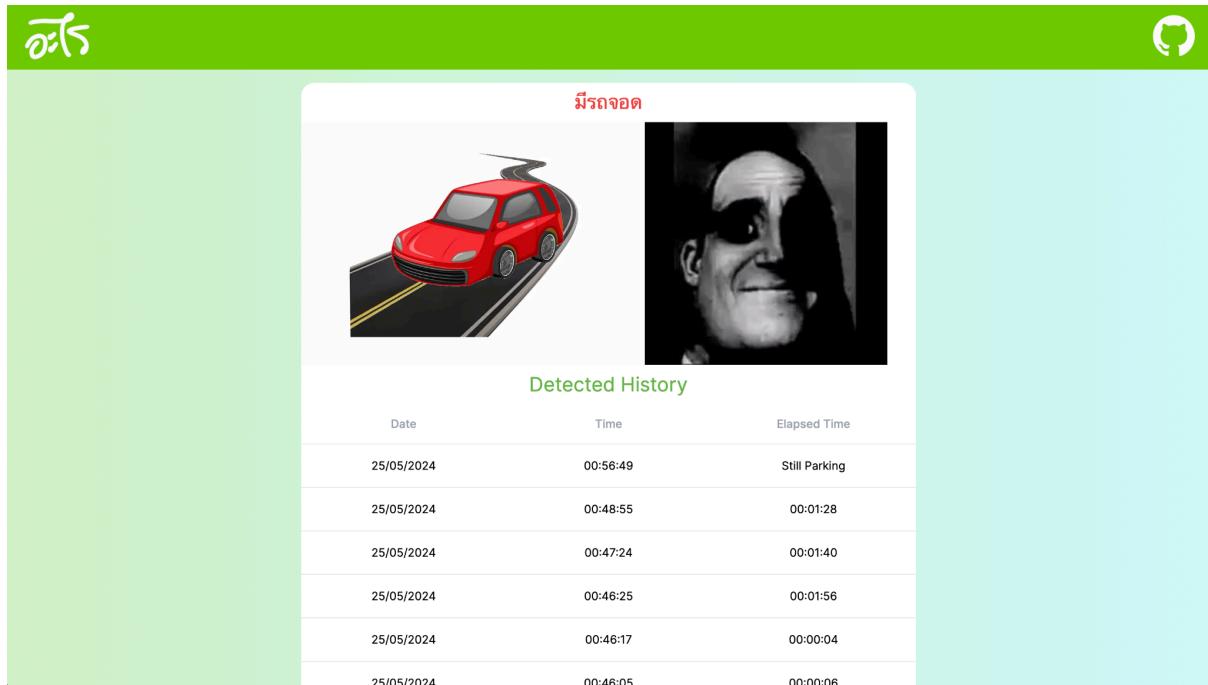
## Approach

เซ็นเซอร์ตรวจจับมี 2 ประเภท คือ Light Dependent Resistor (LDR) เพื่อตรวจด้วยความสว่างของแสง และ Ultrasonic Distance Sensor เพื่อตรวจด้วยระยะทาง โดยใช้อุปกรณ์ทั้งสองประเภทเป็นจำนวน 3 ตู้ หรือห้องหมุด 6 ห้อง อุปกรณ์แจ้งเตือน 1 ชิ้นคือ Buzzer (ลำโพง) โดยต่ออุปกรณ์ทั้งหมดกับไมโครคอนโทรลเลอร์ STM32 NUCLEO-F411RE

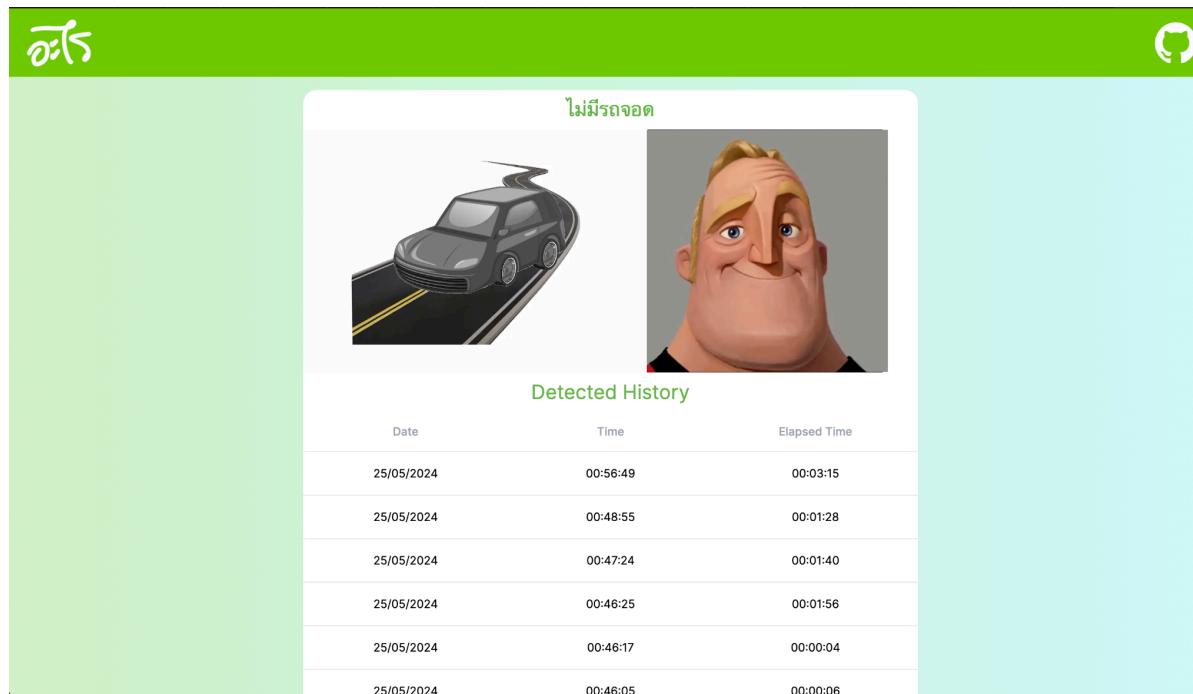
เงื่อนไขในการตรวจจับคือ ใน 1 ถูอุปกรณ์ (LDR และ Ultrasonic Sensor) มีรถเข้ามาใกล้เซนเซอร์วัดระยะทางที่กำหนดไว้และความส่วนที่เซนเซอร์วัดความส่วนของแสงตรวจจับได้มีการลดลง ซึ่งต้องเป็นไปตามทั้งสองเงื่อนไขจึงจะถือว่าเซนเซอร์คุณี้ตรวจจับได้ว่ามีรถอยู่จริง เพื่อป้องกันการตรวจจับที่ผิดพลาด เช่น มีคนมาขยับบัง จึงมีเซนเซอร์หงิด 3 ถู และต้องมีเซนเซอร์อย่างน้อย 2 ใน 3 ถูที่เข้าเงื่อนไขและต้องเข้าเงื่อนไขเป็นเวลาตานาตามที่กำหนดไว้ เพื่อป้องกันกรณีที่รถจอดเพียงไม่นาน เช่น รถติดไฟแดง จึงจะถือว่ามีรถมาจอดในที่ห้ามจอดอยู่จริงๆ

จากนั้นทำการส่งข้อมูลสถานะว่ามีรถจอดหรือไม่จาก microcontroller สู่ NodeMCU ผ่าน UART โดยจะส่งสัญญาณเฉพาะเวลาที่สถานะมีการเปลี่ยนเท่านั้นเพื่อการประหยัดทรัพยากร จากนั้น NodeMCU ส่งข้อมูลสถานะการจอดรถไป update ใน virtual database ของ Netpie

ตัวเว็บไซต์ที่ถูกเขียนด้วย NextJS จะทำการดึงข้อมูลจาก Database ของ Netpie โดยใช้ API ที่ Netpie จัดเตรียมไว้ให้ โดยจะแสดงผลข้อมูล วันที่/เวลา/ระยะเวลาในการจอด และ เปลี่ยนแปลงสถานะ “มีรถจอด” บนหน้าเว็บไซต์ และลง วันที่/เวลา/ที่เริ่มจอด และ “ระยะเวลาในการจอด” จะแสดงผลเป็น “Still Parking” ดังภาพ



- หากได้รับข้อมูล “มีรถจอด” เข้ามา จะทำการเปลี่ยนสถานะเป็น “มีรถจอด” บนหน้าเว็บไซต์ และลง วันที่/เวลา/ที่เริ่มจอด และ “ระยะเวลาในการจอด” จะแสดงผลเป็น “Still Parking” ดังภาพ
- หากได้รับข้อมูล “ไม่มีรถจอด” เข้ามา จะทำการเปลี่ยน จะทำการเปลี่ยนสถานะเป็น “ไม่มีรถจอด” บนหน้าเว็บไซต์ และแก้ไขสถานะ “ระยะเวลาในการจอด” เป็นระยะเวลาตั้งแต่รถคันล่าสุดเริ่มเข้ามาจอด จนถึง ระยะเวลาที่ไม่มีรถจอด ดังภาพ



เพื่อบันทึกข้อมูลช่วงเวลาที่มีรถเข้ามาจอด วันที่/เวลา/ระยะเวลาในการจอด จะถูกบันทึกลงบน Real-Time Database ของ Firebase ด้วย

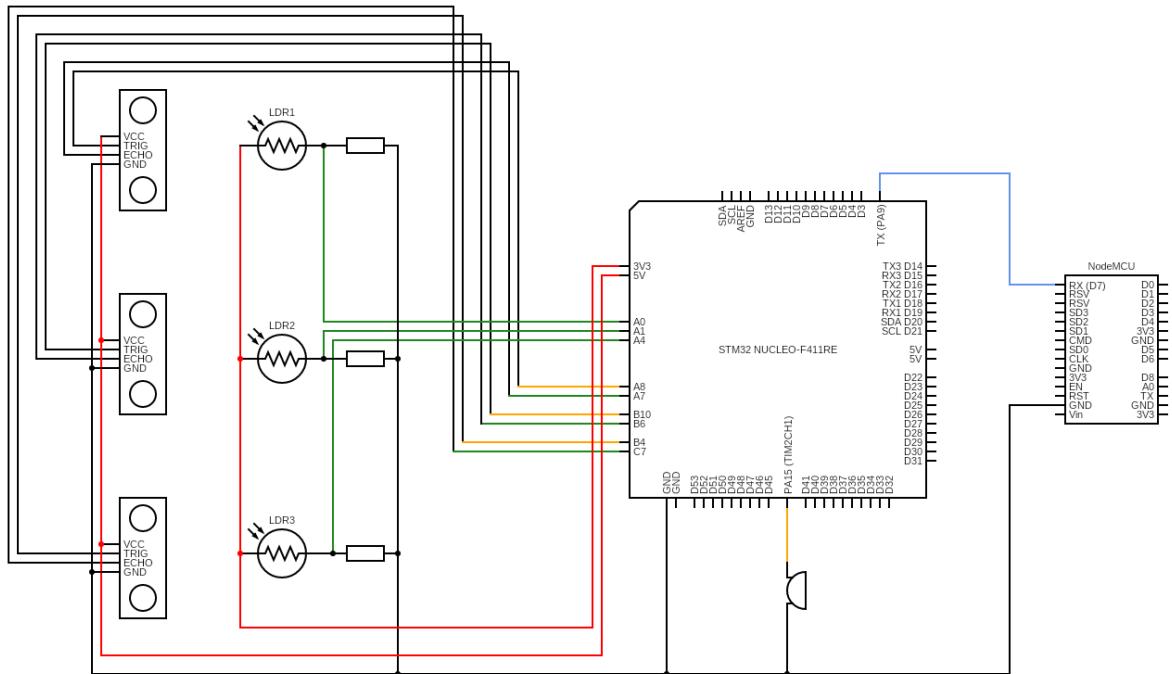
## Tools

1. STM32 NUCLEO-F411RE MCU
2. NodeMCU ESP8266 V3
3. Light Dependent Resistor (LDR) จำนวน 3 ชิ้น
4. Ultrasonic Distance Sensor HC-SR04 จำนวน 3 ชิ้น
5. Buzzer (ลำโพง)
6. Breadboard
7. Resistor
8. Wire

## Technologies

1. Netpie
2. Firebase
3. Next.js
4. React
5. Tailwind CSS

# Circuit Diagram



# Connection Table

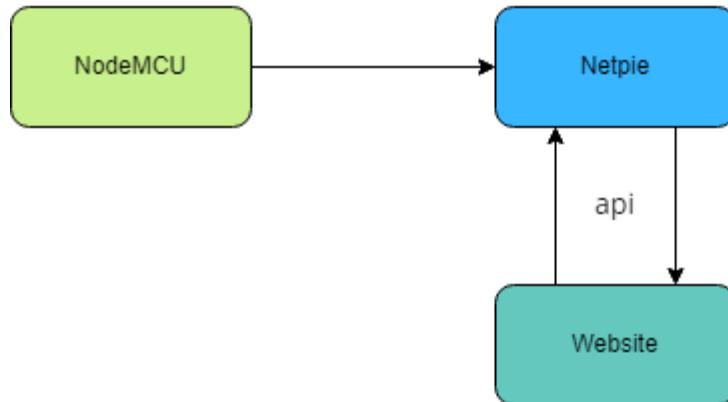
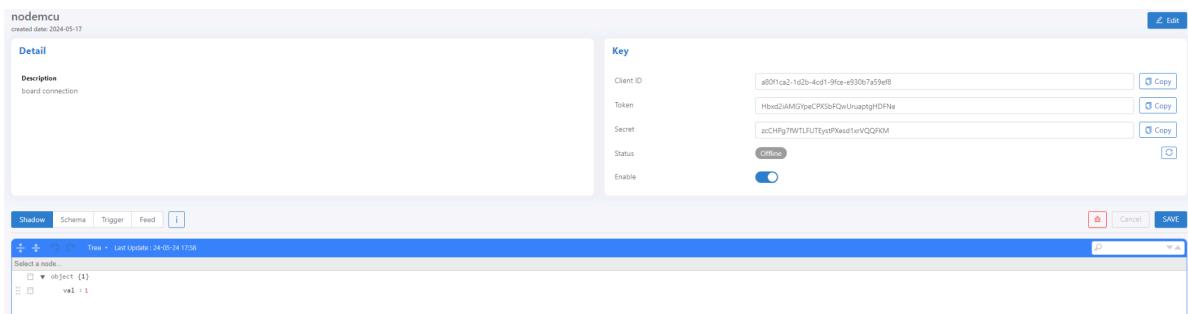
Name	Type	Port
Transmit to NodeMCU	UART1_TX	PA9
LDR 1	ADC1_IN0	PA0
LDR 2	ADC1_IN1	PA1
LDR 3	ADC1_IN4	PA4
Buzzer	TIM2_CH1	PA15
Trigger 1	GPIO_Output	PA8
Echo 1	GPIO_Input	PA7
Trigger 2	GPIO_Output	PB10
Echo 2	GPIO_Input	PB6
Trigger 3	GPIO_Output	PB4
Echo 3	GPIO_Input	PC7

# Roles and Responsibilities

## System Architecture

6532040021 ธนาธิป พัฒนาเพ็ญ

ออกแบบ idea และ logic ของระบบการตรวจสอบการจับการจอดรถ, การส่งข้อมูลระหว่างบอร์ด, และ api สำหรับการส่งของมูลที่นั่นเว็บไซต์ นอกจากนี้ ได้ช่วย implement ในส่วน logic การเชื่อมต่อ internet, การส่งข้อมูลสถานะการจอดรถไปเก็บใน virtual database ของ netpie ของ NodeMCU, config schema api และไปช่วยเหลือฝ่ายเว็บไซต์บ้านเล็กน้อย ตามสถานการณ์



# Embedded System Development

6532142421 ภูวิพัทธ์ สิงห์

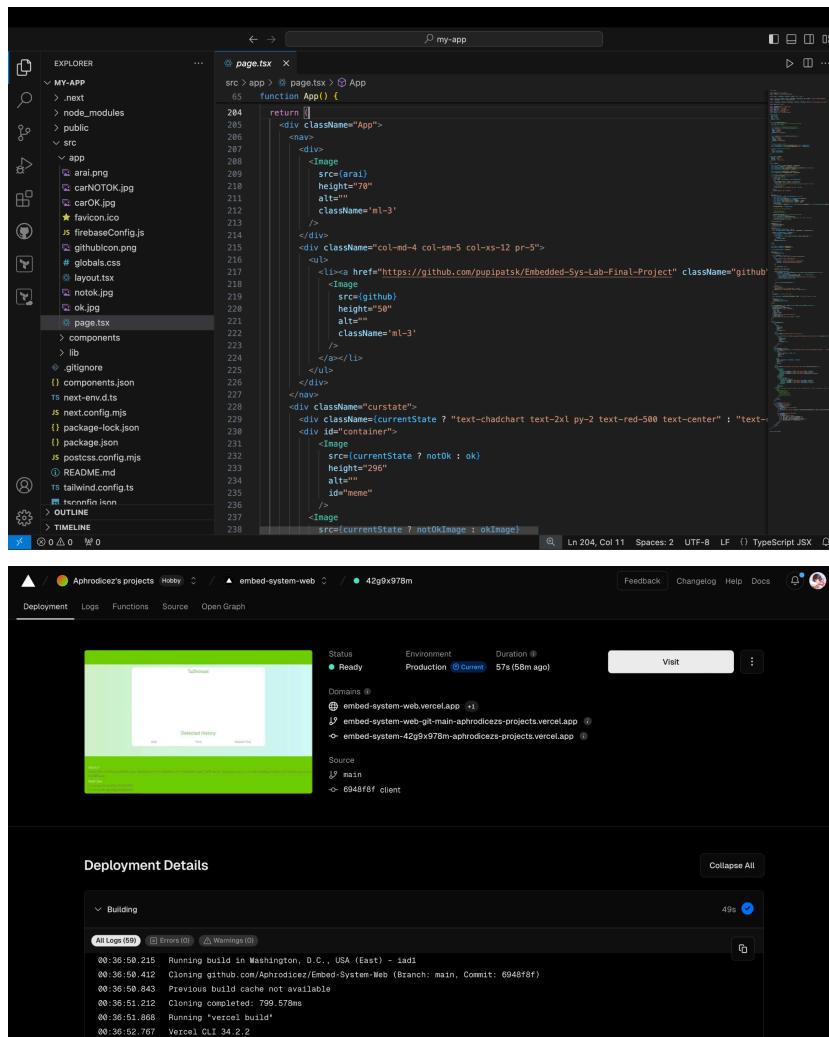
- ต่อ/ออกแบบวงจรและ การจัดสายไฟกับ port เพื่อมต่อของอุปกรณ์ต่อพ่วงทั้ง 8 ชิ้น (LDR x3, Ultrasonic x3, Buzzer, และ NodeMCU)
- ออกแบบ/วาด Preliminary design และ Circuit diagram
- Implement Code ในส่วนของ microcontroller กับ sensors และ NodeMCU
- Implement และแก้ปัญหา physics ในส่วนของการตรวจจับ เช่น Ultrasonic เป็นคลื่น มีการ รบกวนกันของสัญญาณ, LDR เป็น resistor ต้องจัดการ Voltage divider ให้เหมาะสม และการ เลือก Input Voltage / Control Voltage ของแต่ละอุปกรณ์
- ทดลองและวิเคราะห์หาค่า parameters ที่เหมาะสมของแต่ละ sensor

```
1 void StartDefaultTask(void const * argument)
2 {
3     /* USER CODE BEGIN 5 */
4     /* Infinite loop */
5     for(;;)
6     {
7         // LDR sensors
8         lux1 = readADCValue(ADC_CHANNEL_0);
9         lux2 = readADCValue(ADC_CHANNEL_1);
10        lux3 = readADCValue(ADC_CHANNEL_2);
11
12        // Ultrasonic sensors
13        distance1 = measureDistance(GPIOA, GPIO_PIN_8, GPIOA, GPIO_PIN_7);
14        osDelay(60);
15        distance2 = measureDistance(GPIOB, GPIO_PIN_10, GPIOB, GPIO_PIN_6);
16        osDelay(60);
17        distance3 = measureDistance(GPIOB, GPIO_PIN_4, GPIOC, GPIO_PIN_7);
18
19        // Logic
20        // LDR
21        lux_threshold = 3500;
22        lux1State = (lux1 <= lux_threshold) ? 1 : 0;
23        lux2State = (lux2 <= lux_threshold) ? 1 : 0;
24        lux3State = (lux3 <= lux_threshold) ? 1 : 0;
25
26        // Ultrasonic
27        dist_threshold = 10;
28        dist1State = (distance1 < dist_threshold) ? 1 : 0;
29        dist2State = (distance2 < dist_threshold) ? 1 : 0;
30        dist3State = (distance3 < dist_threshold) ? 1 : 0;
31
32        // pair
33        pair1State = (lux1State && dist1State) ? 1 : 0;
34        pair2State = (lux2State && dist2State) ? 1 : 0;
35        pair3State = (lux3State && dist3State) ? 1 : 0;
36
37        // calculate
38        oldState = presentState;
39        uint8_t sumPairState = pair1State + pair2State + pair3State;
40
41        if (sumPairState == 2) {
42            presentState = 1;
43            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
44            state1CycleCount++;
45            if (state1CycleCount == 5) {
46                HAL_UART_Transmit_IT(shuart1, &presentState, sizeof(presentState));
47                RxState = presentState;
48            }
49            if (state1CycleCount == 5) {
50
51            }
52        }
53        else {
54            presentState = 0;
55            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
56            state1CycleCount = 0;
57
58            // Falling edge
59            if (oldState == 0 && presentState == 0) {
60                if (presentState != RxState) {
61                    HAL_UART_Transmit_IT(shuart1, &presentState, sizeof(presentState));
62                    RxState = presentState;
63                }
64            }
65        }
66
67        // print via UART
68        printUART2("lux1", lux1);
69        printUART2("lux2", lux2);
70        printUART2("lux3", lux3);
71        printUART2("distance1", distance1);
72        printUART2("distance2", distance2);
73        printUART2("distance3", distance3);
74
75        osDelay(1000);
76    }
77    /* USER CODE END 5 */
78 }
79
80
81 /* USER_CODE BEGIN Header_StartTask02 */
82 /**
83 * @brief Function implementing the myTask02 thread.
84 * @param argument: Not used
85 * @retval None
86 */
87 /* USER_CODE END Header_StartTask02 */
88 void StartTask02(void const * argument)
89 {
90     /* USER_CODE BEGIN StartTask02 */
91     /* Infinite loop */
92
93     for(;;)
94     {
95         if (presentState == 1 && state1CycleCount >= 5) {
96             TIM2->CCR1 = 50;
97         }
98         else TIM2->CCR1 = 0;
99     }
100    /* USER_CODE END StartTask02 */
101 }
```

## UI/UX Designer and Development

6532035021 ឧណ្ណានិន គងសេរីកូល

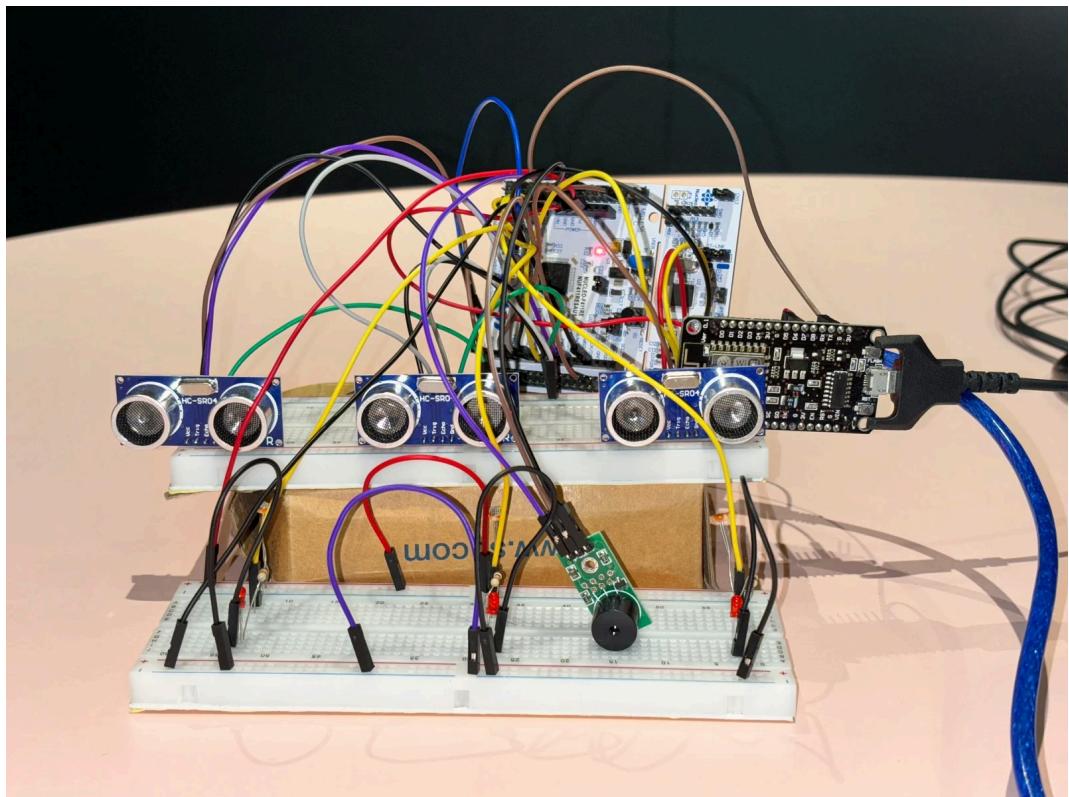
- ริเริ่มเสนอไอเดียในการพัฒนาเทคโนโลยีในการแก้ปัญหาการจอดในที่ห้ามจอด ตามนโยบาย “บริหารจัดการจราจรด้วยระบบอัจฉริยะ (ITMS) เพื่อนำริหารจัดการจราจรทั่วโลกง่ายและรวดเร็ว”
  - Implement Code หน้าเว็บไซต์ ด้วย NextJS
  - Front End: หลังจากได้หน้าเว็บไซต์แรก ได้มีส่วนในการตกแต่ง ตารางประวัติการจอดรถ ให้มีความ modern มากยิ่งขึ้น ด้วย Tailwind CSS
  - Back End: รับข้อมูลจาก API ของ Netpie เพื่อแสดงผลขึ้นหน้าเว็บ และเก็บข้อมูลไว้บน Firebase Realtime Database
  - Logic: Implement การคำนวนระยะเวลาในการจอด และแก้ไขข้อมูลใน Database หน้าเว็บไซต์ไป Host บน <https://vercel.com> เพื่อให้ผู้ใช้งานสามารถเข้าถึงหน้าเว็บไซต์



# Team Management

6532100021 ชีกพ เล้าพรพิชยานุรัตน์

ส่วนที่รับผิดชอบคือการจัดสรรเวลา แต่ก็ได้มีส่วนช่วยในการต่อวงจร เชื่อมต่ออุปกรณ์ต่างๆ(LDR x3, Ultrasonic x3, Buzzer, และ NodeMCU) และออกแบบ UX/UI website โดยใช้ภาษา HTML, CSS และ TypeScript ซึ่งมีการใช้ React ในการเขียน

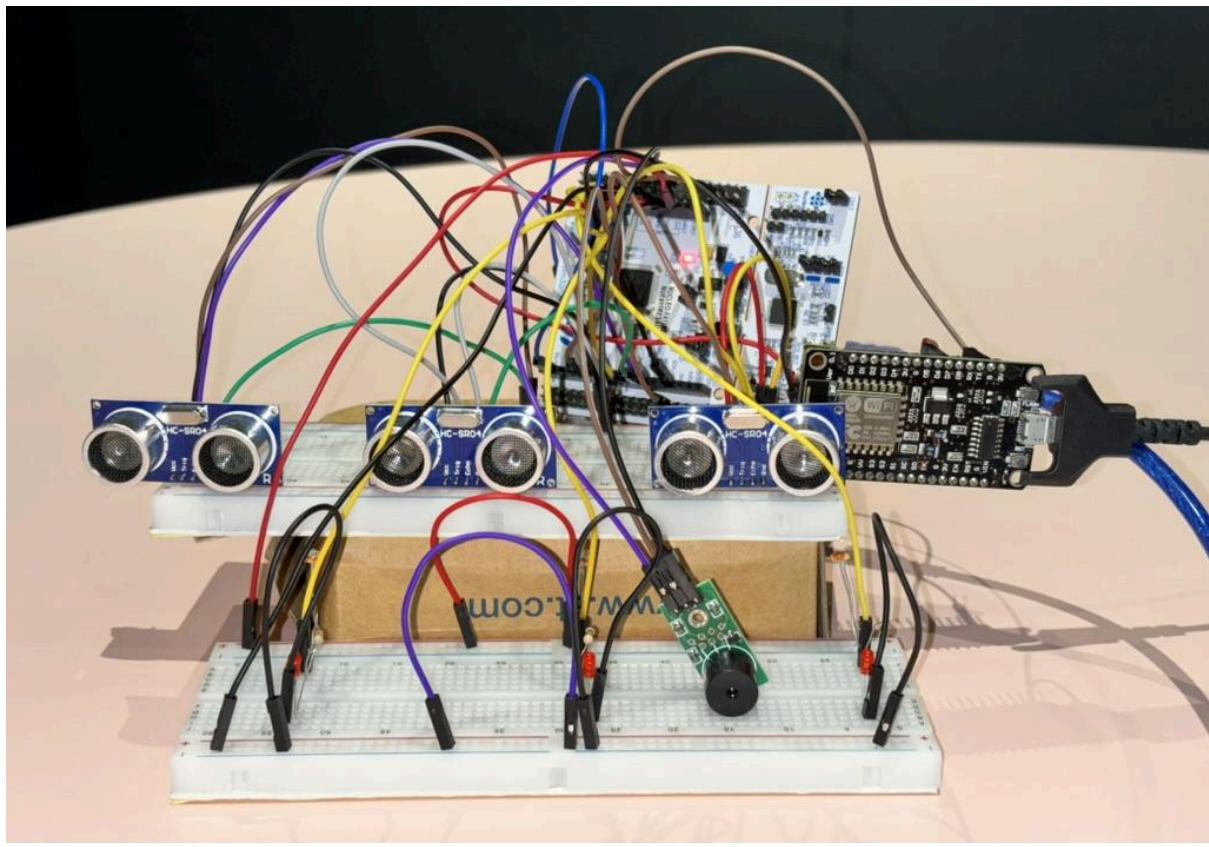


หน้าเว็บตอนแรก

date	time	status
15/1/66	15:00	มีรถจอด
15/1/66	15:00	ไม่มีรถจอด
15/1/66	15:00	มีรถจอด
15/1/66	15:00	ไม่มีรถจอด

toggle | update

## Outcome



หน้าเว็บสุดท้าย



ไม่มีรถจอด



## Detected History

Date	Time	Elapsed Time
25/05/2024	00:56:49	00:03:15
25/05/2024	00:48:55	00:01:28
25/05/2024	00:47:24	00:01:40
25/05/2024	00:46:25	00:01:56
25/05/2024	00:46:17	00:00:04
25/05/2024	00:46:05	00:00:06

Date	Start Time	End Time
25/05/2024	00:56:49	Still Parking
25/05/2024	00:48:55	00:01:28
25/05/2024	00:47:24	00:01:40
25/05/2024	00:46:25	00:01:56
25/05/2024	00:46:17	00:00:04
25/05/2024	00:46:05	00:00:06
25/05/2024	00:45:55	00:00:06
25/05/2024	00:45:45	00:00:06
25/05/2024	00:45:37	00:00:04
25/05/2024	00:45:25	00:00:06

จัดทำโดย

## Source Code

GitHub

<https://github.com/pupipatsk/Embedded-Sys-Lab-Final-Project>

Website

<https://embed-system-web.vercel.app/>

Main Source Code

<https://github.com/pupipatsk/Embedded-Sys-Lab-Final-Project/blob/f2f294e39511dcdea2ea39cc0b3223bc98d5d88f/Final%20Project/Core/Src/main.c>

Schematic

<https://github.com/pupipatsk/Embedded-Sys-Lab-Final-Project/blob/f2f294e39511dcdea2ea39cc0b3223bc98d5d88f/circuit.svg>