ระบบจัดการลานจอดรถอัจฉริยะ

Smart Parking Lot Management System

รายงานนี้เป็นส่วนหนึ่งของรายวิชา Internet of Things

CS423_327E

จัดทำโดย

นาย วรกิตติ์ เมืองจันทร์ 1660704790

นางสาว ภัชภิชา ธัชกรกุลวัฒน์ 1660704675

อาจารย์ผู้สอน อ.เลอศักดิ์ ลิมวิวัฒน์กุล

ภาคการศึกษาที่ 2 ปีการศึกษา 2569

บทคัดย่อ

รายงานฉบับนี้นำเสนอการออกแบบและพัฒนาระบบจัดการลานจอดรถอัจฉริยะ (Smart Parking Lot Management System) โดยใช้ไมโครคอนโทรลเลอร์ Doit ESP32 DevKit V1 เป็นหน่วยประมวลผลกลาง ระบบถูกออกแบบให้ทำงานผ่านการตรวจจับการเคลื่อนที่ของยานพาหนะด้วยเซนเซอร์อินฟราเรด (IR Sensor) ณ จุดทางเข้าและทางออก เพื่อทำการนับจำนวนรถยนต์ภายในลานจอดแบบตามเวลาจริง (Real-time) โดยรองรับความจุสูงสุดจำนวน 3 คัน สำหรับการควบคุมการเข้า-ออก ดำเนินการผ่านระบบไม้กั้นอัตโนมัติที่ขับเคลื่อนด้วยเซอร์โวมอเตอร์ (Servo Motor)

นอกจากนี้ ระบบยังได้บูรณาการฟังก์ชันช่วยจอด (Parking Assist) โดยใช้เซนเซอร์อัลตราโซนิก (HC-SR04 Ultrasonic Sensor) จำนวน 3 ตัว ติดตั้งประจำช่องจอด เพื่อทำหน้าที่แจ้งเตือนด้วยเสียงผ่านลำโพง (Buzzer) เมื่อยานพาหนะถอยเข้าใกล้สิ่งกีดขวางในระยะน้อยกว่า 5 เซนติเมตร ในส่วนของการแสดงผล สถานะของลานจอดจะถูกแสดงผ่านหน้าจอ OLED ขนาด 128x64 พิกเซล ร่วมกับโมดูลไฟจราจร 3 สี (Traffic Light Module) จากการทดสอบการทำงานของระบบพบว่า ฟังก์ชันทั้งหมดสามารถทำงานได้อย่างถูกต้องและครบถ้วนตามวัตถุประสงค์ที่ได้ออกแบบไว้

1. บทนำ

1.1 ที่มาและความสำคัญ

ปัญหาการค้นหาพื้นที่จอดรถเป็นปัญหาที่พบได้ทั่วไปในบริเวณที่มีปริมาณการจราจรหนาแน่น เช่น สถานศึกษา ห้างสรรพสินค้า หรืออาคารสำนักงาน
ผู้ขับขี่มักต้องสูญเสียเวลาในการขับวนหาพื้นที่ว่างเนื่องจากไม่ทราบสถานะที่แท้จริงของลานจอดรถ นอกจากนี้ การขาดระบบควบคุมพื้นที่ที่มีประสิทธิภาพยังอาจก่อให้เกิดความสับสน
และเสี่ยงต่อการเกิดอุบัติเหตุภายในบริเวณพื้นที่จอดรถได้

โครงงานนี้จึงถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาดังกล่าว โดยนำเทคโนโลยีและหลักการของอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things: IoT) มาประยุกต์ใช้ในการสร้างต้นแบบระบบจัดการลานจอดรถอัจฉริยะ
ระบบนี้ถูกออกแบบมาเพื่อให้สามารถแสดงสถานะความว่างของพื้นที่จอดรถได้แบบตามเวลาจริง (Real-time)
พร้อมทั้งมีระบบควบคุมการเข้า-ออกผ่านไม้กั้นอัตโนมัติ
เพื่อเพิ่มประสิทธิภาพและความปลอดภัยในการบริหารจัดการพื้นที่ลานจอดรถ

1.2 วัตถุประสงค์

1. เพื่อออกแบบและพัฒนาระบบตรวจจับและนับจำนวนยานพาหนะเข้า-ออกลานจอดรถแบบอัตโนมัติ
2. เพื่อพัฒนาระบบควบคุมไม้กั้นประตูลานจอดรถโดยใช้เซอร์โวมอเตอร์
3. เพื่อสร้างระบบช่วยจอด (Parking Assist) สำหรับแจ้งเตือนระยะปลอดภัยขณะถอยรถโดยใช้เซนเซอร์อัลตราโซนิก
4. เพื่อพัฒนาส่วนแสดงผลสถานะการทำงานของลานจอดรถผ่านหน้าจอ OLED และโมดูลไฟจราจร

1.3 ขอบเขตของการศึกษา

1. โครงงานนี้เป็นการจัดทำในรูปแบบต้นแบบ (Prototype) สำหรับลานจอดรถขนาดเล็ก
   ซึ่งรองรับความจุยานพาหนะสูงสุดจำนวน 3 คัน
2. ระบบทำงานในโหมดเอกเทศ (Standalone) โดยไม่มีการเชื่อมต่อเครือข่ายอินเทอร์เน็ตในเวอร์ชันปัจจุบัน
3. การประมวลผล การแสดงผล และการควบคุมอุปกรณ์ทั้งหมด
   ดำเนินการผ่านฮาร์ดแวร์ที่เชื่อมต่อโดยตรงกับบอร์ดไมโครคอนโทรลเลอร์ ESP32

2. อุปกรณ์ฮาร์ดแวร์ที่ใช้ในการศึกษา

2.1 รายการอุปกรณ์

ในการพัฒนาต้นแบบระบบจัดการลานจอดรถอัจฉริยะ ได้มีการเลือกใช้อุปกรณ์ไมโครคอนโทรลเลอร์และเซนเซอร์ต่างๆ ดังรายละเอียดในตารางที่ 1

ตารางที่ 1 รายการอุปกรณ์ฮาร์ดแวร์ที่ใช้ในโครงงาน

ลำดับ	ชื่ออุปกรณ์	รายละเอียด	จำนวน	หน้าที่
1	Micricintroller	Doit ESP32 DevKit V1	1	ประมวลผลกลางและควบคุมระบบ
2	Ultrasonic Sensor	HC-SR04	3	ตรวจจับระยะห่างของยานพาหนะในช่องจอด
3	IR Sensor	Infared Obstacle Avoidance	2	ตรวจจับยานพาหนะบริเวณทางเข้า-ออก
4	Servo Motor	SG90	1	ขับเคลื่อนกลไกเปิด-ปิดไม้กั้น
5	OLED Display	SSD1306	1	แสดงสถานะลานจอดและจำนวนรถ
6	Traffic Light Module	LED 3 color	1	แสดงสถานะอนุญาตการเข้าลานจอด
7	Buzzer	Active Buzzer 5V	1	แจ้งเตือนถอยจอดและยืนยันคำสั่งปุ่มกด
8	Push Button	Tactile Switch	1	สั่งการเปิด-ปิดการทำงานของระบบ
9	LED Status	LED Green 5 mm + ตัวต้านทาน 220Ω	1	แสดงสถานะการทำงานของระบบ
10	แผงต่อวงจรและสายสัญญาณ	Breadboard + Jumper	1	เป็นฐานเชื่อมต่อวงจรและอุปกรณ์ฮาร์ดแวร์

2.2 การกำหนดขาเชื่อมต่อ (Pin Assignment)

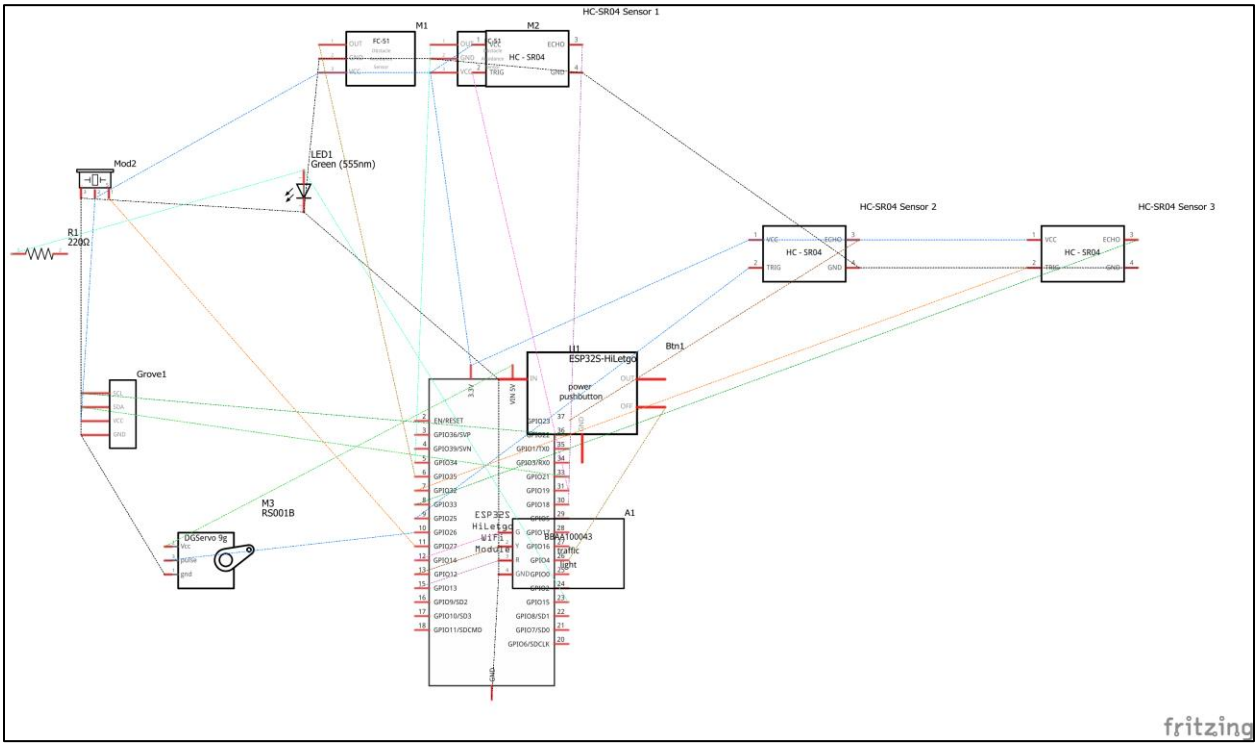
การสื่อสารระหว่างหน่วยประมวลผล ESP32 และอุปกรณ์ต่อพ่วงต่างๆ มีการกำหนดขาพิน (GPIO) และโหมดการทำงานดังรายละเอียดในตารางที่ 2

ตารางที่ 2 การกำหนดขาพินเชื่อมต่อระหว่าง ESP32 กับอุปกรณ์ต่อพ่วง

อุปกรณ์ต่อพ่วง	ขาพิน ESP32	โหมดการทำงาน	หน้าที่
Traffic Light - Red	GPIO 13	OUTPUT	แสดงไฟหยุดรอ (ไฟแดง)
Traffic Light - Yellow	GPIO 12	OUTPUT	แสดงไฟเตรียมพร้อม (ไฟเหลือง)
Traffic Light - Green	GPIO 14	OUTPUT	แสดงไฟอนุญาตให้ผ่าน (ไฟเขียว)
Servo Motor	GPIO 26	OUTPUT (PWM)	ควบคุมการหมุน 0-90 องศา
IR Sensor (IN)	GPIO 34	INPUT	ทำงานแบบ Active LOW
IR Sensor (OUT)	GPIO 35	INPUT	ทำงานแบบ Active LOW
Buzzer	GPIO 27	OUTPUT	เสียงเตือนถอยจอด / ยืนยันการกดปุ่ม
Push Button	GPIO 4	INPUT_PULLUP	สลับสถานะเปิด-ปิดระบบ (Toggle)
LED	GPIO 15	OUTPUT	สว่างเมื่อระบบทำงาน (System ON)
Ultrasonic #1	GPIO 25 (Trig) / 23 (Echo)	OUTPUT / INPUT	วัดระยะห่างช่องจอดที่ 1
Ultrasonic #2	GPIO 19 (Trig) / 18 (Echo)	OUTPUT / INPUT	วัดระยะห่างช่องจอดที่ 2
Ultrasonic #3	GPIO 19 (Trig) / 18 (Echo)	OUTPUT / INPUT	วัดระยะห่างช่องจอดที่ 3
OLED	GPIO 21 (SDA) /22 (SCL)	I2C	การสื่อสารข้อมูลแบบ I2C

3. แผนภาพวงจร (Schematic Diagram)

การออกแบบแผนภาพวงจร (Schematic Diagram) ของระบบจัดการลานจอดรถอัจฉริยะ แสดงให้เห็นถึงการเชื่อมต่อทางไฟฟ้าของอุปกรณ์ต่อพ่วงทั้งหมดเข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP32 DevKit V1 โดยใช้มาตรฐานการวาดวงจรอิเล็กทรอนิกส์สากล



รูปที่ 3.1 แผนภาพวงจร (Schematic Diagram) ของระบบจัดการลานจอดรถอัจฉริยะ

4. แผนภาพการเชื่อมต่ออุปกรณ์ (Wiring Diagram)

การออกแบบแผนภาพการเชื่อมต่ออุปกรณ์ (Wiring Diagram) ดำเนินการผ่านโปรแกรม Fritzing ในรูปแบบจำลองการต่อแผงวงจร (Breadboard View)
เพื่อแสดงการเดินสายไฟและตำแหน่งการจัดวางอุปกรณ์ฮาร์ดแวร์จริงให้เกิดความชัดเจน

4.1 การจัดวางอุปกรณ์บนแผงวงจร (Breadboard Layout)

โซนด้านบน: ติดตั้งเซนเซอร์อัลตราโซนิก (HC-SR04) จำนวน 3 ตัว สำหรับช่องจอดรถ

โซนด้านซ้าย: ติดตั้งบอร์ดไมโครคอนโทรลเลอร์ ESP32 DevKit V1

โซนตรงกลาง: ประกอบด้วยเซนเซอร์อินฟราเรด (ทางเข้า), หลอดไฟ LED แสดงสถานะระบบ และสวิตช์ปุ่มกดควบคุมระบบ

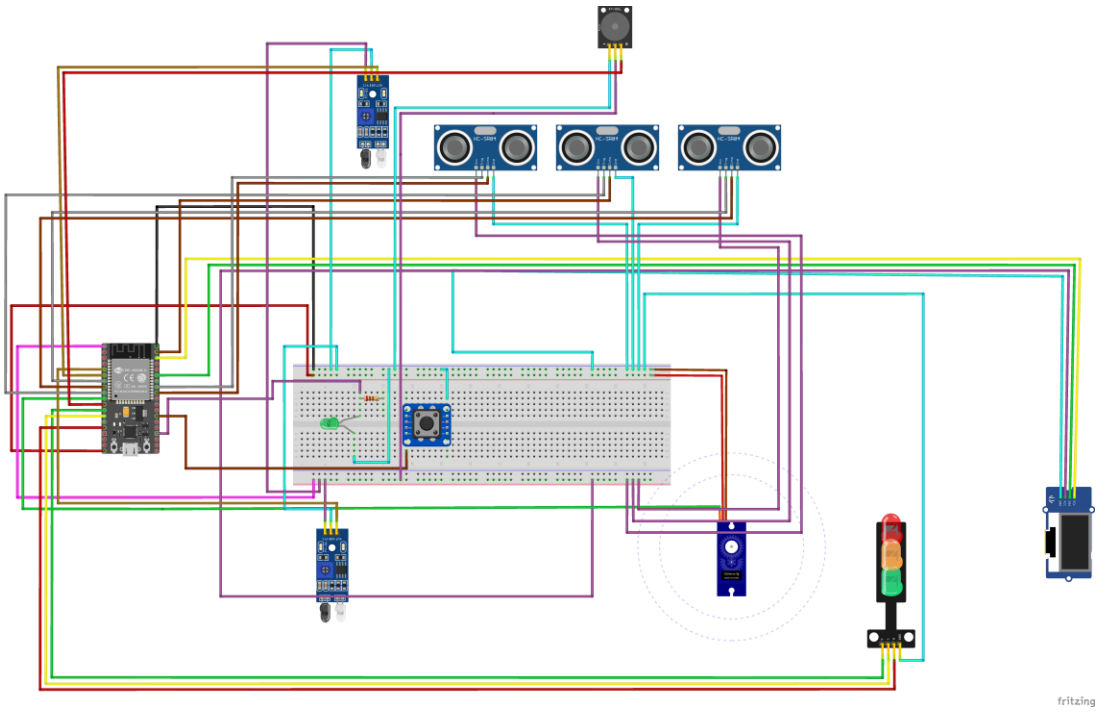
โซนด้านล่าง (ซ้าย-ขวา): ติดตั้งเซนเซอร์อินฟราเรด (ทางออก) และโมดูลไฟจราจร ตามลำดับ

โซนด้านขวา: ติดตั้งหน้าจอแสดงผล OLED ผ่านการสื่อสารแบบ I2C

โซนด้านบนสุด: ติดตั้งลำโพงแจ้งเตือน (Buzzer)

4.2 การเชื่อมต่อระบบไฟเลี้ยง (Power Supply Routing)

ระบบทั้งหมดใช้แหล่งจ่ายไฟ 5V ที่ดึงมาจากพอร์ต USB ของบอร์ด ESP32 โดยมีการเชื่อมต่อรางสายดิน (GND) บนแผงวงจรเข้าด้วยกันทั้งหมด (Common Ground) เพื่อให้ระบบไฟฟ้าเสถียร สำหรับไฟสถานะ LED สีเขียว ได้มีการต่อตัวต้านทานขนาด 220 โอห์ม แบบอนุกรมเพื่อจำกัดกระแสะไฟฟ้าให้อยู่ในระดับที่ปลอดภัยและป้องกันหลอดขาด



รูปที่ 4.2 แผนภาพการเชื่อมต่ออุปกรณ์ (Wiring Diagram) บนแผงวงจร

5. การทำงานของระบบ (System Operation)

5.1 ภาพรวมหารทำงาน (System Overview)

สถาปัตยกรรมซอฟต์แวร์ของระบบถูกออกแบบในลักษณะตอบสนองต่อเหตุการณ์ (Event-driven Architecture) โดยบอร์ด ESP32 จะทำงานในลูป หลักอย่างต่อเนื่อง เพื่อตรวจสอบสถานะของเซนเซอร์แต่ละตัว และสั่งการตอบสนองต่อเหตุการณ์ที่เกิดขึ้นในทันที เช่น การกดปุ่ม การตรวจพบยานพาหนะเข้า-ออก หรือการเปลี่ยนแปลงระยะห่างจากเซนเซอร์อัลตราโซนิก

5.2 ลำดับขั้นตอนการทำงาน (Operational Flow)

5.2.1. ระบบเปิด-ปิดและการเริ่มต้น (System Power Toggle)

ผู้ใช้งานสามารถกดปุ่ม (Push Button) หนึ่งครั้งเพื่อสลับสถานะการทำงานของระบบระหว่างเปิด (ON) และปิด (OFF) โดยลำโพง Buzzer จะส่งเสียงเตือนสั้นๆ 1 ครั้งทุกครั้งที่กดปุ่มเพื่อยืนยันคำสั่ง

- **สถานะ ON:** หลอดไฟ LED สีเขียวติดสว่าง, หน้าจอ OLED แสดงข้อความ "SYSTEM ON!" และโมดูลไฟจราจรแสดงไฟสีแดง

- **สถานะ OFF:** หลอดไฟ LED ดับลง, ไม้กั้นถูกสั่งให้อยู่ในตำแหน่งปิด (ลงอัตโนมัติ) และหน้าจอ OLED แสดงข้อความ "SYSTEM OFF"

5.2.2. ระบบควบคุมทางเข้าลานจอด (Auto Entry Gate)

เมื่อเซนเซอร์อินฟราเรดบริเวณทางเข้าตรวจพบยานพาหนะ ระบบจะทำการตรวจสอบจำนวนรถยนต์ที่จอดอยู่ภายในลานก่อนอนุญาตให้เข้า

กรณีมีพื้นที่ว่าง :

1. ไฟจราจรเปลี่ยนเป็นสีเหลือง และหน้าจอ OLED แสดงข้อความ "WAIT / CHECKING..." เป็นเวลา 2 วินาที
2. ไฟจราจรเปลี่ยนเป็นสีเขียว และหน้าจอ OLED แสดงข้อความ "WELCOME!"
3. เซอร์โวมอเตอร์สั่งยกไม้กั้นขึ้น (หมุนจาก 0 ถึง 90 องศา) และหน่วงเวลา 3 วินาทีเพื่อให้ยานพาหนะเคลื่อนผ่าน
4. ระบบนับจำนวนยานพาหนะ (parkedCars) เพิ่มขึ้น 1 คัน
5. เซอร์โวมอเตอร์สั่งปิดไม้กั้นลง (หมุนจาก 90 ถึง 0 องศา) และไฟจราจรกลับเป็นสีแดง

กรณีมีพื้นที่จอดเต็ม :

ไม้กั้นจะไม่ถูกเปิดออก และหน้าจอ OLED จะแสดงข้อความเตือน "NO SPACE / PARKING FULL"

5.2.3. ระบบตรวจจับทางออก (Exit Detection)

เมื่อเซนเซอร์อินฟราเรดบริเวณทางออกตรวจพบยานพาหนะเคลื่อนผ่าน ระบบจะคำนวณลดจำนวนยานพาหนะ (parkedCars) ลง 1 คัน และหน้าจอ OLED จะแสดงข้อความ "THANK YOU! / HAVE A GOOD DAY" เป็นเวลา 2 วินาที ทั้งนี้ ทางออกไม่ได้ติดตั้งระบบไม้กั้น จึงทำงานในลักษณะการนับจำนวน (Counter-based) เพียงอย่างเดียว

5.2.4. ระบบช่วยจอดและเตือนระยะชน (Parking Assist)

เซนเซอร์อัลตราโซนิกทั้ง 3 ตัวที่ประจำอยู่แต่ละช่องจอดจะทำงานอย่างต่อเนื่องตลอดเวลาที่เปิดระบบ หากระบบคำนวณพบว่าระยะห่างจากตัวรถถึงกำแพงมีค่าน้อยกว่าหรือเท่ากับ 5 เซนติเมตร ลำโพง Buzzer จะส่งเสียงเตือนยาวต่อเนื่อง จนกว่ายานพาหนะจะขยับออกจากระยะอันตราย

5.2.5. การแสดงผลตามเวลาจริง (Real-time Status Display)

หน้าจอ OLED: แสดงข้อมูลจำนวนพื้นที่จอดที่ถูกใช้ไปในรูปแบบ "PARK: X/3" พร้อมระบุสถานะภาพรวมของลานจอด เช่น "LOT IS AVAILABLE" หรือ "LOT IS FULL"

โมดูลไฟจราจร: ทำหน้าที่สื่อสารกับผู้ขับขี่บริเวณทางเข้า (ไฟสีแดง = หยุดรอระบบทำงาน, ไฟสีเหลือง = กำลังประมวลผลหรือขยับไม้กั้น, ไฟสีเขียว = อนุญาตให้เข้าได้)

6. ซอร์สโค้ด (Source Code)

6.1 ไลบรารีที่ใช้ในการทำงาน (Required Libraries)

การทำงานของระบบจำเป็นต้องพึ่งพาไลบรารี (Library) สำเร็จรูป เพื่อช่วยในการจัดการโมดูลต่างๆ ให้ทำงานร่วมกับ ESP32 ได้อย่างมีประสิทธิภาพ ดังรายละเอียดในตารางที่ 3

ตารางที่ 3 รายการไลบรารีที่ใช้ในชุดคำสั่ง (Firmware)

ชื่อไลบรารี	หน้าที่
Wire.h	จัดการการสื่อสารข้อมูลแบบ I2C สำหรับหน้าจอ OLED
Adafruit_GFX.h	ไลบรารีหลักสำหรับการสร้างกราฟิกและตัวอักษรบนหน้าจอ
Adafruit_SSD1306.h	ไดรเวอร์ (Driver) เฉพาะสำหรับควบคุมจอ OLED รุ่น SSD1306
ESP32Servo.h	ควบคุมการส่งสัญญาณ PWM ไปยังเซอร์โวมอเตอร์บนบอร์ด ESP32
soc/soc.h soc/rtc_cntl_reg.h	ปิดระบบเซนเซอร์วัดไฟตก (Brownout Detector) เพื่อป้องกันบอร์ดรีเซ็ตตัวเองเมื่อเซอร์โวมอเตอร์ดึงกระแสไฟสูงในระยะเวลาสั้นๆ

6.2 ชุดคำสั่งโปรแกรม (Firmware Code)

ชุดคำสั่งด้านล่างเป็นเฟิร์มแวร์ฉบับสมบูรณ์ที่ใช้ในการควบคุมการทำงานทั้งหมดของระบบจัดการลานจอดรถอัจฉริยะ

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ESP32Servo.h>
// --- โค้ดลับกันไฟตก บอร์ดดับ ---
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"

// --- ตั้งค่าจอ OLED ---
#define SCREEN_WIDTH 128 // ความกว้างจอ OLED
#define SCREEN_HEIGHT 64 // ความสูงจอ OLED
#define OLED_RESET    -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// --- กำหนดขาพินอุปกรณ์ ---
const int redPin = 13;
const int yellowPin = 12;
const int greenPin = 14;
const int servoPin = 26;
const int entrySensor = 34; // IR ทางเข้า
const int exitSensor = 35;  // IR ทางออก
const int buzzerPin = 27;   // ลำโพง
const int buttonPin = 4;    // ปุ่มกด เปิด-ปิดระบบ

//  เพิ่มขาพินสำหรับ "ไฟสถานะระบบ" (ไฟลานจอด)
```

```
const int systemLightPin = 15;

// --- กำหนดขาพิน Ultrasonic 3 ตัว ---
const int trig1 = 32; const int echo1 = 33;
const int trig2 = 25; const int echo2 = 23;
const int trig3 = 19; const int echo3 = 18;

// --- ตัวแปรจัดการลานจอดรถ ---
int parkedCars = 0;
const int MAX_SPACE = 3;
int servoSpeed = 15;

// --- ตัวแปรสำหรับปุ่มเปิด-ปิด ---
bool isSystemActive = false;
bool lastButtonState = HIGH;
bool isBuzzerActive = false;

Servo barrierServo;

long getDistance(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW); delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH, 30000);
  if (duration == 0) return 999;
  return duration * 0.034 / 2;
}

void setup() {
```

```
WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // ปิดระบบรีเซ็ตตอนไฟตก

Serial.begin(115200);
Serial.println("\n[SYSTEM] ESP32 Ready! Waiting for start button...");

// เริ่มการทำงานจอ OLED
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println(F("SSD1306 allocation failed"));
}
display.clearDisplay();
display.setTextColor(WHITE);

pinMode(redPin, OUTPUT); pinMode(yellowPin, OUTPUT); pinMode(greenPin, OUTPUT);
pinMode(buzzerPin, OUTPUT);
pinMode(entrySensor, INPUT); pinMode(exitSensor, INPUT);
pinMode(buttonPin, INPUT_PULLUP);

//  ตั้งค่าให้ไฟสถานะระบบเป็น OUTPUT และสั่งให้ดับไว้ก่อนตอนเริ่ม
pinMode(systemLightPin, OUTPUT);
digitalWrite(systemLightPin, LOW);

pinMode(trig1, OUTPUT); pinMode(echo1, INPUT);
pinMode(trig2, OUTPUT); pinMode(echo2, INPUT);
pinMode(trig3, OUTPUT); pinMode(echo3, INPUT);

digitalWrite(buzzerPin, LOW);
barrierServo.attach(servoPin);
barrierServo.write(0);
```

```
  // หน้าจอเริ่มต้นก่อนเปิดระบบ
  display.clearDisplay();
  display.setTextSize(2);
  display.setCursor(5, 15); display.print("SYSTEM OFF");
  display.setTextSize(1);
  display.setCursor(20, 40); display.print("PRESS TO START");
  display.display();
}

void loop() {
  // ========================================
  // เช็กการกดปุ่ม (เปิด/ปิด ระบบ)
  // ========================================
  bool currentButtonState = digitalRead(buttonPin);
  if (lastButtonState == HIGH && currentButtonState == LOW) {
    isSystemActive = !isSystemActive;
    digitalWrite(buzzerPin, HIGH); delay(150); digitalWrite(buzzerPin, LOW);

    if (isSystemActive) {
      Serial.println("\n>> SYSTEM: ON");
      display.clearDisplay(); display.setTextSize(2); display.setCursor(10, 25); display.print("SYSTEM ON!");
display.display();

      //  สั่งเปิดไฟสถานะระบบ เมื่อเปิดเครื่อง
      digitalWrite(systemLightPin, HIGH);

      delay(1500);
    } else {
      Serial.println("\n>> SYSTEM: OFF");
```

```
    display.clearDisplay();
    display.setTextSize(2); display.setCursor(5, 15); display.print("SYSTEM OFF");
    display.setTextSize(1); display.setCursor(20, 40); display.print("PRESS TO START");
    display.display();
    barrierServo.write(0);
    digitalWrite(redPin, HIGH); digitalWrite(yellowPin, LOW); digitalWrite(greenPin, LOW);

    //  สั่งปิดไฟสถานะระบบ เมื่อปิดเครื่อง
    digitalWrite(systemLightPin, LOW);
  }
  delay(200);
}
lastButtonState = currentButtonState;

if (isSystemActive == false) return;

// ========================================
// 1. ระบบเตือนถอยจอด 5 ซม.
// ========================================
long dist1 = getDistance(trig1, echo1);
long dist2 = getDistance(trig2, echo2);
long dist3 = getDistance(trig3, echo3);

if ((dist1 > 0 && dist1 <= 5) || (dist2 > 0 && dist2 <= 5) || (dist3 > 0 && dist3 <= 5)) {
  digitalWrite(buzzerPin, HIGH);
  if (!isBuzzerActive) { Serial.println("- WARNING: รถถอยใกล้กำแพงเกิน 5 ซม."); isBuzzerActive = true; }
} else {
  digitalWrite(buzzerPin, LOW);
  if (isBuzzerActive) { Serial.println("- OK: ระยะปลอดภัย"); isBuzzerActive = false; }
```

```
}

// ========================================
// 2. หน้าจอหลัก และ ไฟแดง
// ========================================
digitalWrite(redPin, HIGH); digitalWrite(yellowPin, LOW); digitalWrite(greenPin, LOW);

display.clearDisplay();

// บรรทัดบน (จำนวนรถ)
display.setTextSize(2);
display.setCursor(0, 5);
display.print("PARK: "); display.print(parkedCars); display.print("/"); display.print(MAX_SPACE);

// บรรทัดล่าง (สถานะ)
display.setTextSize(1);
display.setCursor(0, 40);
if (parkedCars >= MAX_SPACE) {
  display.print(">> LOT IS FULL <<");
} else {
  display.print("LOT IS AVAILABLE");
}
display.display(); // สั่งให้จอแสดงผล

// ========================================
// 3. เมื่อมีรถเข้า (IR 1 ทางเข้าทำงาน)
// ========================================
if (digitalRead(entrySensor) == LOW) {
  if (parkedCars < MAX_SPACE) {
```

```
  digitalWrite(redPin, LOW); digitalWrite(yellowPin, HIGH);

  display.clearDisplay(); display.setTextSize(2);
  display.setCursor(0, 10); display.print("WAIT");
  display.setTextSize(1); display.setCursor(30, 40); display.print("CHECKING...");
  display.display();
  delay(2000);

  digitalWrite(yellowPin, LOW); digitalWrite(greenPin, HIGH);

  display.clearDisplay(); display.setTextSize(2);
  display.setCursor(15, 25); display.print("WELCOME!"); display.display();

  for (int pos = 0; pos <= 90; pos += 1) { barrierServo.write(pos); delay(servoSpeed); }
  delay(3000);

  parkedCars++;

  digitalWrite(greenPin, LOW); digitalWrite(yellowPin, HIGH);
  display.clearDisplay(); display.setTextSize(2);
  display.setCursor(5, 10); display.print("CLOSING..."); display.display();

  for (int pos = 90; pos >= 0; pos -= 1) { barrierServo.write(pos); delay(servoSpeed); }

} else {
  display.clearDisplay(); display.setTextSize(2);
  display.setCursor(10, 10); display.print("NO SPACE");
  display.setTextSize(1); display.setCursor(20, 40); display.print(">> PARKING FULL"); display.display();
  delay(3000);
```

```
    }
    while(digitalRead(entrySensor) == LOW) { delay(100); }
  }

  // ==========================================
  // 4. เมื่อมีรถออก (IR 2 ทางออกทำงาน)
  // ==========================================
  if (digitalRead(exitSensor) == LOW) {
    if (parkedCars > 0) {
      parkedCars--;
      display.clearDisplay(); display.setTextSize(2);
      display.setCursor(5, 10); display.print("THANK YOU!");
      display.setTextSize(1); display.setCursor(15, 40); display.print("HAVE A GOOD DAY"); display.display();
      delay(2000);
    }
    while(digitalRead(exitSensor) == LOW) { delay(100); }
  }
}
```

7. ผลการทดสอบการทำงานของระบบ (System Test Results)

การทดสอบประสิทธิภาพของระบบได้ดำเนินการบนฮาร์ดแวร์จริง โดยตรวจสอบการทำงานของแต่ละฟังก์ชันตามเงื่อนไขที่ออกแบบไว้ ผลการทดสอบสามารถสรุปได้ดังตารางที่ 4

ตารางที่ 4 สรุปผลการทดสอบฟังก์ชันการทำงานของระบบ

ระบบที่ทดสอบ	เงื่อนไขการทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลการทดสอบ
Power Toggle - on	สั่งการกดปุ่มเพื่อเปิดระบบ	LED สถานะติดสว่าง, จอ OLED แสดง "SYSTEM ON!"	ผ่านเกณฑ์
Power Toggle - off	สั่งการกดปุ่มเพื่อปิดระบบ	LED สถานะดับลง, ไม้กั้นถูกลดระดับลงสู่สถานะปิด	ผ่านเกณฑ์
Entry Gate	ยานพาหนะเข้า (กรณีมีที่ว่าง)	ไม้กั้นเปิดอัตโนมัติ, จอ OLED แสดงข้อความ "WELCOME!"	ผ่านเกณฑ์
Entry Gate	ยานพาหนะเข้า (กรณีที่จอดเต็ม)	ไม้กั้นไม่ตอบสนอง, จอ OLED แสดง "NO SPACE"	ผ่านเกณฑ์
Exit Detection	ยานพาหนะเคลื่อนผ่านทางออก	จำนวน parkedCars ลดลง 1, แสดงข้อความ "THANK YOU!"	ผ่านเกณฑ์
Parking Assist	ยานพาหนะถอยใกล้กำแพง (ระยะ $\leq$ 5 ซม.)	ลำโพง Buzzer แจ้งเตือนด้วยเสียงอย่างต่อเนื่อง	ผ่านเกณฑ์
Parking Assist	ยานพาหนะอยู่ในระยะปลอดภัย (ระยะ > 5 ซม.)	ลำโพง Buzzer หยุดส่งเสียงเตือน	ผ่านเกณฑ์
Status Display	ตรวจสอบการนับจำนวนรถสะสม	จอ OLED แสดงผล "PARK: X/3" อัปเดตตัวเลขได้ถูกต้อง	ผ่านเกณฑ์

8. บทสรุปและข้อเสนอแนะ (Conclusion and Future Work)

8.1 สรุปผลการดำเนินงาน

โครงงานการพัฒนาระบบจัดการลานจอดรถอัจฉริยะ (Smart Parking Lot Management System) สามารถดำเนินการออกแบบ พัฒนา และทดสอบประสิทธิภาพได้สำเร็จครบถ้วนตามวัตถุประสงค์ทุกประการ โดยไมโครคอนโทรลเลอร์ ESP32
สามารถทำหน้าที่เป็นศูนย์กลางในการประมวลผลและควบคุมอุปกรณ์ฮาร์ดแวร์ทั้งหมดได้อย่างมีเสถียรภาพ
ระบบสามารถนับจำนวนยานพาหนะเข้า-ออกแบบตามเวลาจริง (Real-time)
ตลอดจนควบคุมการทำงานของไม้กั้นอัตโนมัติด้วยเซอร์โวมอเตอร์ได้อย่างแม่นยำ
พร้อมทั้งสามารถสื่อสารสถานะลานจอดกับผู้ใช้งานผ่านหน้าจอ OLED และไฟจราจรได้อย่างชัดเจน นอกจากนี้ ระบบช่วยจอด (Parking Assist) ยังสามารถแจ้งเตือนสิ่งกีดขวางเพื่อเพิ่มความปลอดภัยได้อย่างมีประสิทธิภาพ

8.2 ข้อจำกัดของระบบ (System Limitations)

ในเวอร์ชันปัจจุบัน ระบบยังใช้กระบวนการนับจำนวนรถผ่านเซนเซอร์อินฟราเรดแบบตรวจจับการเคลื่อนที่ผ่านจุด (Counter-based Detection) ซึ่งอาจก่อให้เกิดความคลาดเคลื่อนทางสถิติได้
ในกรณีที่ยานพาหนะขับผ่านเซนเซอร์ทางเข้าแล้วทำการถอยรถออกทางเดิม นอกจากนี้
ตัวระบบยังถูกจำกัดการทำงานอยู่ในรูปแบบเอกเทศ (Standalone) จึงยังไม่สามารถตรวจสอบสถานะการทำงานจากระยะไกลได้

8.3 แนวทางการพัฒนาในอนาคต (Future Work)

เพื่อเพิ่มขีดความสามารถของระบบให้สมบูรณ์ยิ่งขึ้นในอนาคต สามารถบูรณาการระบบเชื่อมต่อเครือข่ายไร้สาย (Wi-Fi Connectivity) ที่มีอยู่ในบอร์ด ESP32 เพื่อส่งข้อมูลขึ้นสู่ระบบคลาวด์ (Cloud) และแสดงผลสถิติแบบออนไลน์ผ่าน Web Dashboard รวมไปถึงการประยุกต์ใช้เทคโนโลยีการประมวลผลภาพ (Computer Vision) เช่น กล้องตรวจจับป้ายทะเบียน (ALPR) แทนการใช้เซนเซอร์อินฟราเรดแบบดั้งเดิม เพื่อลดความผิดพลาดในการนับจำนวนยานพาหนะ

เอกสารอ้างอิง (References)

ESP32 Technical Reference Manual

สืบค้นจาก https://www.espressif.com/en/support/documents/technical-documents

Adafruit SSD1306 OLED Library

สืบค้นจาก https://github.com/adafruit/Adafruit_SSD1306

Arduino Language Reference

สืบค้นจาก https://www.arduino.cc/reference/en/

ESP32Servo Library

สืบค้นจาก https://github.com/madhephaestus/ESP32Servo

Fritzing Electronics made easy

สืบค้นจาก https://fritzing.org