เรื่อง ระบบถังขยะอัตโนมัติ

Automatic Bin System

จัดทำโดย

ธนกฤต ศรีแก้ว รหัสนิสิต 65021666

ชูตระกูล แสนโซ้ง รหัสนิสิต 65024636

กฤตเมธ วันแรก รหัสนิสิต 65025637

วิลาวัลย์ เพ็ชรเอม รหัสนิสิต 65025783

เสนอ

อาจารย์ คมกริช มาเที่ยง

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชาระบบสมองกลฝังตัว (226242)

สาขาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร

มหาวิทยาลัยพะเยา

ภาคเรียนที่ 2 ปีการศึกษา 2566

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชาระบบสมองกลฝังตัว เพื่อให้ได้ศึกษาหาความรู้ในเรื่องการออกแบบระบบสมองกลฝังตัว เพื่อเป็นประโยชน์กับการเรียนในระดับอุดมศึกษาชั้นปีที่ 2

คณะผู้จัดทำได้เลือกหัวข้อระบบถังขยะอัตโนมัติในหารทำรายงาน เนื่องด้วยเป็นอุปกรณ์ที่ใกล้ตัวและมีการใช้งานในชีวิตประจำวันของทุกคน อีกทั้งทำให้ได้ศึกษาเกี่ยวกับการออกแบบระบบสมองกลฝังตัวที่มีระบบพื้นฐานต่างๆ อีกด้วย ทั้งนี้ทางคณะผู้จัดต้องขอขอบพระคุณ อาจารย์คมกริช มาเที่ยง เป็นอย่างสูง ที่ให้ความรู้ คำแนะนำ และแนวทางในการศึกษา ค้นคว้า

คณะผู้จำทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักศึกษา ที่กำลังหาข้อมูลในเรื่องการออกแบบระบบสมองกลฝังตัว หากมีข้อผิดพลาดประการใด คณะผู้จำขอน้อมรับไว้ และขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญภาพ

รูปที่ หน้า

สารบัญภาพ (ต่อ)

รูปที่ หน้า

บทที่ 1
ที่มาและความสำคัญ

1.1 ที่มาและความสำคัญ

เนื่องในปัจจุบันระบบสมองกลฝังตัวถูกพัฒนาและใช้งานกันอย่างแพร่หลายในงานระบบขนาดเล็กที่พบเจอได้ในชีวิตประจำวัน เช่น อุปกรณ์อินเทอร์เน็ตในทุกสรรพสิ่ง (Internet of Things) ไปถึงระบบขนาดใหญ่ หรือระบบที่เกี่ยวข้องกับเวลา และความปลอดภัยต่างๆ เช่น ระบบอุตสาหกรรม ระบบทางการแพทย์ต่างๆ ระบบยานพาหนะ เป็นต้น เพื่อช่วยอำนวยความสะดวก ความปลอดภัย และความน่าเชื่อถือในระบบงานต่างๆ

จากที่กล่าวข้างต้นนั้น ทางคณะผู้จัดทำจึงเล็งเห็นถึงความสำคัญของระบบสมองกลฝังตัวในปัจจุบัน และในอนาคต จึงมีความต้องการที่จะศึกษาค้นคว้า เกี่ยวกับการพัฒนา และการทำงานของระบบสมองกลฝังตัว โดยเลือกหัวข้อระบบ คือ ระบบถังขยะอัตโนมัติ

คณะผู้จัดทำได้เลือกทำระบบถังขยะอัตโนมัติ เนื่องด้วยเป็นสิ่งที่พบเห็น และใช้งานในชีวิตประจำวันของทุกคน เพื่อให้สามารถเข้าใจ มองเห็นถึงความสำคัญ และการนำไปใช้งานจริง อีกทั้ง มีองค์ประกอบการทำงานของระบบอยู่ในหัวข้อพื้นฐานของระบบสมองกลฝังตัวที่ควรศึกษา เช่น การอ่านค่าจากเซนเซอร์ (Sensor) การตรวจสอบค่าต่างๆ เพื่อประมวลผลการทำงาน การควบคุมเซอร์โวมอเตอร์ (Servo Motor) และการแสดงค่า เป็นต้น เพื่อให้ได้ความรู้ความเข้าใจในเรื่องของระบบสมองกลฝังตัว สามารถนำความรู้ไปต่อยอดพัฒนาเพื่อประกอบอาชีพที่เกี่ยวข้องได้ในอนาคต

1.2 วัตถุประสงค์

1. เพื่อศึกษาหาความรู้เกี่ยวกับการทำงานของระบบสมองกลฝังตัว
2. เพื่อศึกษาหาความรู้เกี่ยวกับการพัฒนาระบบสมองกลฝังตัว
3. เพื่อศึกษาหาความรู้เกี่ยวกับการทำงานระหว่างเซนเซอร์ต่างๆ กับระบบสมองกลฝังตัว
4. เพื่อศึกษาหาความรู้เกี่ยวกับการควบคุมอุปกรณ์ต่างๆ โดยระบบสมองกลฝังตัว
5. เพื่อสร้าง และพัฒนาระบบสมองกลฝังตัวที่ใช้งานได้ในชีวิตประจำวัน

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้เกี่ยวกับการทำงานของระบบสมองกลฝังตัว
2. ได้ความรู้ความรู้เกี่ยวกับการพัฒนาระบบสมองกลฝังตัว
3. ได้ความรู้เกี่ยวกับการทำงานระหว่างเซนเซอร์ต่างๆ กับระบบสมองกลฝังตัว
4. ได้ความรู้เกี่ยวกับการควบคุมอุปกรณ์ต่างๆ โดยระบบสมองกลฝังตัว
5. สามารถมีระบบสมองกลฝังตัวที่ใช้ได้ในชีวิตประจำวัน ช่วยประหยัดค่าใช้จ่ายได้
6. สามารถนำความรู้ที่ได้ ไปต่อยอดในอนาคต

บทที่ 2
เอกสารอ้างอิง

จากการศึกษาค้นคว้าเรื่อง ระบบสมองกลฝังตัว การเลือกหัวข้อระบบถังขยะอัตโนมัติ ทางคณะผู้จัดทำได้ศึกษาเอกสาร และงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

2.1 อุปกรณ์ที่ต้องใช้

2.1.1 บอร์ด ESP32

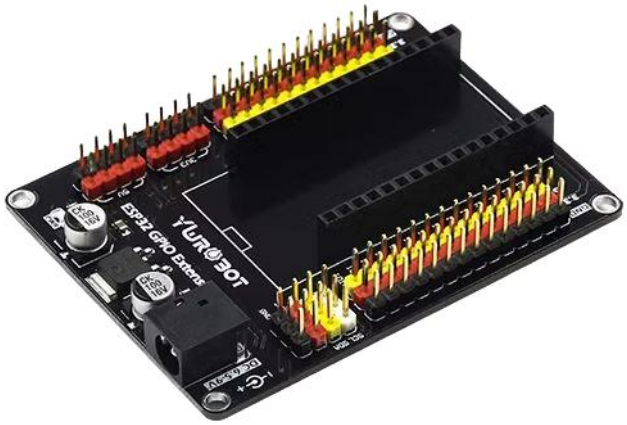
ESP32 เป็นไมโครคอนโทรลเลอร์ (Microcontroller) ที่รองรับการเชื่อมต่อ WiFi และ Bluetooth 4.2 BLE ในตัว ผลิตโดยบริษัท Espressif จากประเทศจีน

รูปที่ 1.1 บอร์ด ESP32

รูปที่ 1.2 แผนผังขาต่างๆ ของบอร์ด ESP32

2.1.2 บอร์ดฐานสำหรับ ESP32 (ESP32 Base Board)

เป็นบอร์ดขยายขาต่างๆ สำหรับบอร์ด ESP32 ให้สามารถต่อสายไฟได้ง่ายยิ่งขึ้น มีขาจ่ายไฟ 5V เพิ่มเติม และยังสามารถจ่ายไฟ 6.5V - 9V เข้าที่บอร์ดฐานเพื่อแปลงแรงดันให้สามารถจ่ายไฟที่เหมาะสมให้กับ ESP32 ได้อีกด้วย



รูปที่ 2.0 บอร์ดฐานสำหรับ ESP32

2.1.3 บอร์ดทดลองขนาดเล็ก (Breadboard)

เป็นอุปกรณ์ที่ใช้ในการต่อวงจรไฟฟ้าโดยไม่ต้องเชื่อมต่อวงจรแบบถาวร ช่วยให้สามารถทดลองวงจร และสร้าง ปรับเปลี่ยน แก้ไขวงจรไฟฟ้าได้อย่างรวดเร็ว สะดวก

รูปที่ 3.0 บอร์ดทดลองขนาดเล็ก

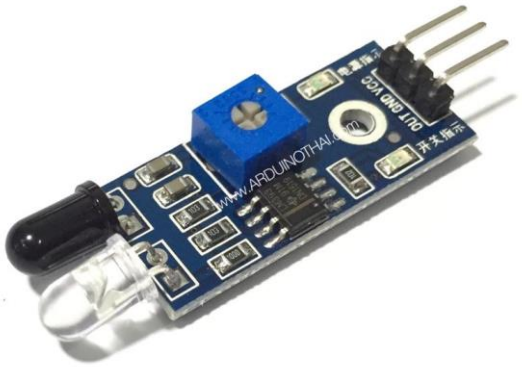
2.1.4 E18-D80NK Sensor

เป็นเซนเซอร์ตรวจจับวัตถุที่ใช้หลักการสะท้อนของคลื่นอินฟาเรด สามารถตรวจจับวัตถุได้ในระยะ และยังกำหนดระยะในการทำงานโดยปรับค่าที่ Potentiometer ได้อีกด้วย โดยจะใช้สำหรับตรวจจับวัตถุหน้าถังขยะ

รูปที่ 4.0 E18-D80NK Sensor

2.1.5 IR Infrared Obstacle Detection Sensor

เป็นเซ็นเซอร์ใช้ตรวจจับวัตถุโดยใช้หลักการสะท้อนของคลื่นอินฟาเรดเช่นเดียวกันกับ E18-D80NK Sensor ไม่สามารถปรับระยะการตรวจจับได้ แต่สามารถปรับความไวในการตรวจจับได้ โดยจะใช้สำหรับตรวจจับขยะภาพในถัง



รูปที่ 5.0 IR Infrared Sensor

2.1.6 เซอร์โวมอเตอร์ MG996

เป็นเซอร์โวมอเตอร์ที่สามารถหมุนได้ 0-180 องศา โดยใช้กระแสไฟที่ 5VDC ให้แรงบิดสูงสุดที่ 12kg/cm โดยจะใช้สำหรับเปิดฝาของถังขยะ

รูปที่ 6.0 MG996 Servo Motor

2.1.7 Module ไฟ LED 3 สี

เป็นแผงวงจรรวมไฟ 3 สี ที่ใช้แรงดันระหว่าง 3.3V – 5V ใช้สำหรับแสดงสถานะของขยะภายในถังขยะ

รูปที่ 7.0 Module ไฟ LED 3 สี

2.1.8 Active Buzzer

เป็นลำโพงแบบแม่เหล็ก เปียโซที่มีวงจรกำเนิดความถี่ภายในตัว ใช้ไฟเลี้ยง 3.3V – 5V สามารถสร้างเสียงเตือนได้ โดยจะใช้สำหรับสร้างเสียงเตือนการตรวจพบวัตถุหน้าถังขยะ เพื่อเปิดฝาถังขยะ

รูปที่ 8.0 Active Buzzer

2.1.9 สายไฟต่อวงจร

ใช้สำหรับเชื่อมต่ออุปกรณ์ต่างๆ เช่น ESP32 และเซ็นเซอร์ต่างๆ เข้าด้วยกัน

รูปที่ 9.0 สายไฟต่อวงจร

2.1.10 ถังขยะแบบฝาปิดบานพับ

ต้องใช้ถังขยะแบบฝาปิดบานพับที่สามารถปิดเปิดได้ เพื่อใช้สำหรับการเปิดปิดอัตโนมัติด้วยเซอร์โวมอเตอร์

รูปที่ 10.0 ถังขยะแบบฝาปิดบานพับ

2.2 เครื่องมือที่ต้องใช้

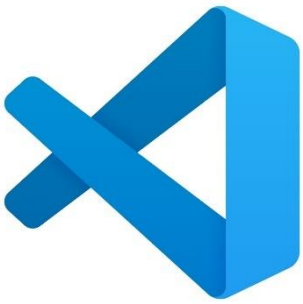
2.2.1 Docker

เป็นเครื่องมือแบบโอเพนซอร์ส (Open Source) ที่ช่วยจำลองสภาพแวดล้อม (environment) ในการรัน service หรือ server ตามหลักการสร้าง container เพื่อจัดการกับ library ต่างๆ อีกทั้งยังช่วยจัดการในเรื่องของ version control เพื่อง่ายต่อการจัดการกับปัญหาต่างๆ

รูปที่ 11.0 Docker

2.2.2 Visual Studio Code

เป็น Text Editor และยังเป็น IDE อีกด้วย มีความนิยมมากในการนำมาใช้เขียนโปรแกรม และยังเป็นเครื่องมือฟรีที่ออกแบบมาให้ใช้งานได้ทั้งบน Windows, Linux และ MacOS



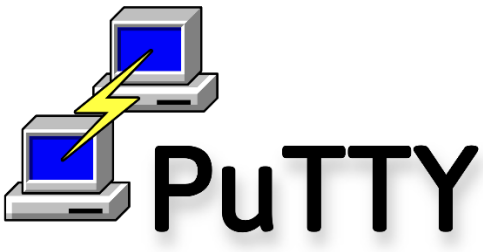
รูปที่ 12.0 Visual Studio Code

2.2.3 Flash Download Tool

พัฒนาโดย Espressif เป็นเครื่องมือที่ใช้ในการแฟลชโปรแกรมซอฟต์แวร์ลงบนชิป ESP8266 หรือ ESP32 โดยตรงผ่านทางการเชื่อมต่อ USB ระหว่างคอมพิวเตอร์ และบอร์ด ESP ด้วย ทำให้สามารถอัปโหลดโปรแกรม หรือแฟลชไฟล์ลงในบอร์ด ESP ได้อย่างง่ายดายโดยไม่จำเป็นต้องใช้ผ่าน IDE หรือเครื่องมืออื่นๆ ที่ซับซ้อน ช่วยให้การพัฒนาโปรแกรมสำหรับ ESP มีความสะดวกและรวดเร็วมาก

2.2.4 PuTTY

เป็นโปรแกรมโอเพนซอร์สที่ใช้ในการเชื่อมต่อกับเซิร์ฟเวอร์ หรืออุปกรณ์ที่ใช้โปรโตคอล SSH, Telnet, rlogin และระบบการเชื่อมต่อแบบพิเศษอื่นๆ ผ่านพอร์ต (Port)



รูปที่ 13.0 PuTTY

บทที่ 3
วิธีการดำเนินงาน

3.1 การออกแบบระบบ

3.1.1 แนวคิดพื้นฐานของระบบ

แนวคิดการทำงานของระบบถังขยะอัตโนมัตินี้ คือ เมื่อผู้ใช้อยู่บริเวณหน้าถังขยะ หรือภายในระยะการตรวจจับของเซ็นเซอร์ ถังขยะจะเปิดอัตโมนัติเป็นเวลา 3 วินาที และส่งเสียงแจ้งเตือน แต่หากผู้ใช้ยังอยู่บริเวณหน้าถังขยะ จะเปิดค้างไว้จนกว่าผู้ใช้จะเดินออกจากระยะตรวจจับ

อีกทั้งยังใช้เซ็นเซอร์เพื่อตรวจจับสถานะของขยะภายในถังว่าขยะเต็มหรือไม่ และจะแสดงสถานะผ่านหลอดไฟ LED ที่อยู่หน้าถังขยะ

3.1.2 การทำงานของระบบโดยละเอียด

ใช้เซ็นเซอร์ E18-D80NK ติดตั้งบริเวณด้านข้างของถังขยะ เพื่อตรวจจับผู้ใช้ หรือวัตถุ หากตรวจไม่พบจะส่งสัญญาณดิจิทัลสถานะ 1 และหากตรวจพบวัตถุส่งสัญญาณดิจิทัลสถานะ 0 เข้าสู่บอร์ด ESP32

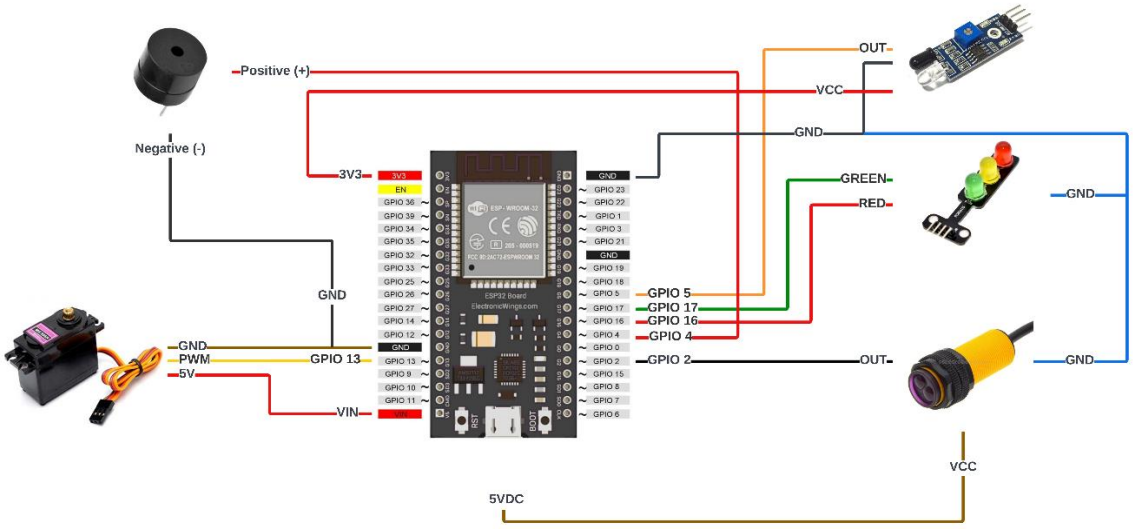
หากมีการตรวจพบวัตถุจากเซ็นเซอร์ E18-D80NK บอร์ด ESP32 จะส่งสัญญาณ PWM ไปควบคุมเซอร์โวมอเตอร์จะดึงฝาถังขยะขึ้นเป็นเวลา 3 วินาที พร้อมส่งสัญญาณให้แก่ Active Buzzer ทำให้ส่งเสียงเตือน และหากตรวจไม่พบวัตถุจากเซ็นเซอร์ E18-D80NK บอร์ด ESP32 จะส่งสัญญาณไปควบคุมเซอร์โวมอเตอร์ให้ปล่อยการดึงฝาถัง

ใช้เซ็นเซอร์ IR Infrared ติดตั้งบริเวณฝาถังด้านในเพื่อตรวจสถานะความเต็มของขยะภายใน ใช้หลักการคล้ายกับเซ็นเซอร์ E18-D80NK หากภายในมีปริมาณขยะน้อย จะไม่เข้าระยะของการตรวจจับวัตถุ ส่งสัญญาณดิจิทัลสถานะ 1 เข้าสู่บอร์ด ESP32 หากภายในมีปริมาณขยะจำนวนมาก จนเข้าใกล้ฝาถัง และเข้าสู่ระยะการตรวจจับของเซ็นเซอร์ IR Infrared จะส่งสัญญาณดิจิทัลสถานะ 0 เข้าสู่บอร์ด ESP32

หากภายในถังมีขยะน้อย หรือยังไม่เข้าระยะการตรวจจับของ IR Infrared บอร์ด ESP32 จะส่งสัญญาณเพื่อเปิดไฟ LED สีเขียว แต่หากภายในถังมีใกล้เต็ม หรือเข้าสู่ระยะการตรวจจับของ IR Infrared บอร์ด ESP32 จะส่งสัญญาณเพื่อเปิดไฟ LED สีแดง

3.1.3 การเชื่อมต่อระหว่าง ESP32 และอุปกรณ์ต่างๆ

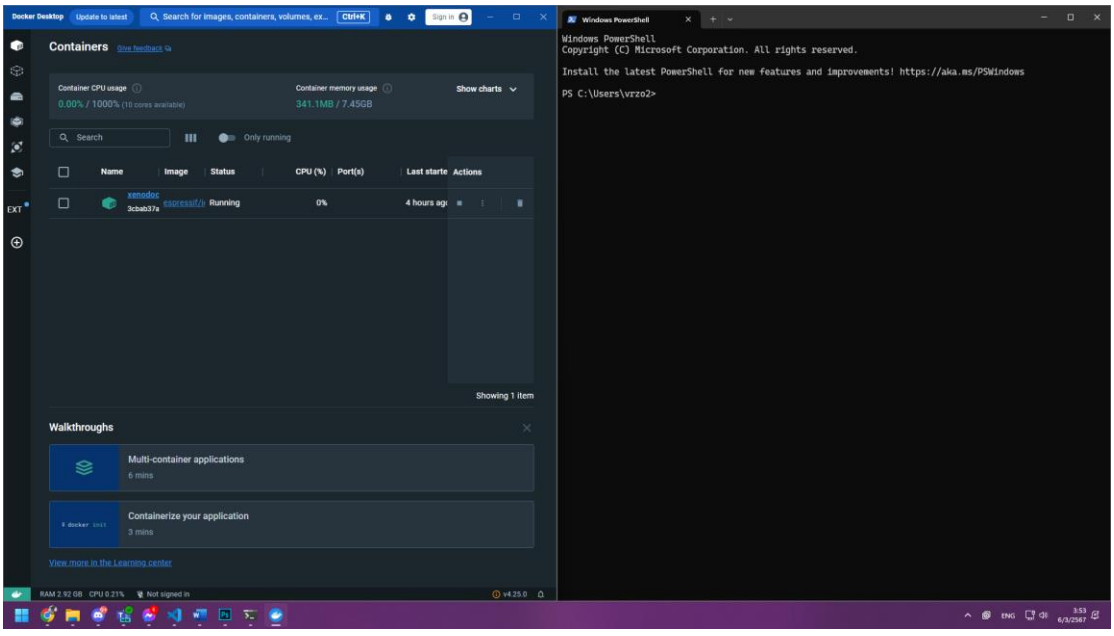
การเชื่อมต่อระหว่าง ESP32 และอุปกรณ์ต่างๆ เพื่อใช้ในระบบถังขยะอัตโนมัติ จะแสดงผ่านแผนผังจำลองดังนี้



รูปที่ 14.0 แผนผังการเชื่อมต่อระหว่าง ESP32 และอุปกรณ์ต่างๆ

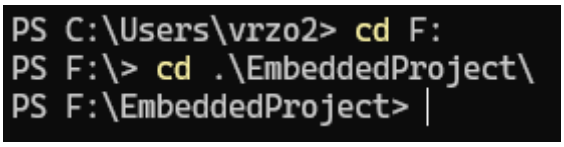
3.2 ขั้นตอนการเขียนโปรแกรมเพื่อลงบอร์ด ESP32

1. ทำการเปิดโปรแกรม Docker และ PowerShell หรือ Terminal ขึ้นมา



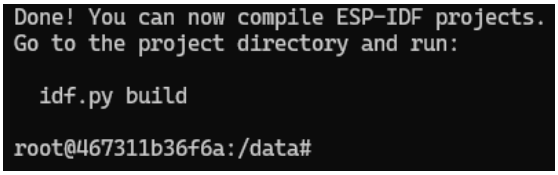
รูปที่ 15.0 โปรแกรม Docker และ PowerShell

2. เข้าไปที่ตำแหน่งที่ต้องการเก็บไฟล์ (File) ของโปรเจค (Project) ด้วยคำสั่ง cd



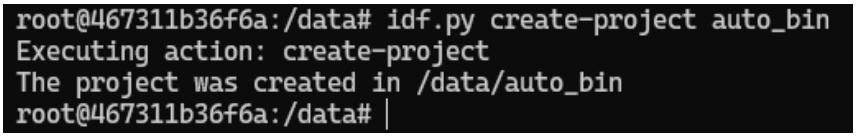
รูปที่ 16.0 การใช้คำสั่ง cd ใน PowerShell

3. ใช้คำสั่ง ***docker run --rm -it -v "${pwd}:/data" -w /data espressif/idf bash*** และรอจนกว่าจะเสร็จสิ้นดังรูปภาพ



รูปที่ 17.0 การใช้คำสั่ง docker run ใน PowerShell

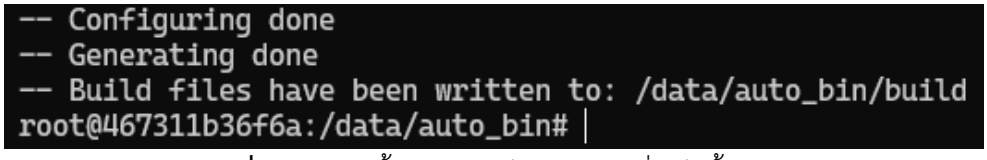
4. ใช้คำสั่ง ***idf.py create-project <ชื่อโปรเจค>*** เพื่อสร้างโฟลเดอร์ (Folder) ของโปรเจค หากสร้างสำเร็จจะได้ผลลัพธ์ดังรูป



รูปที่ 18.0 การสร้างโฟลเดอร์โปรเจค

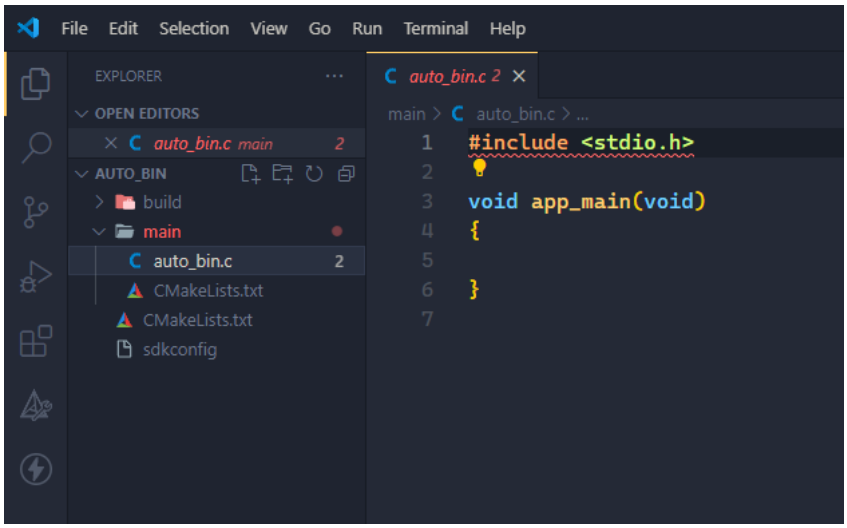
และเข้าไปที่โฟลเดอร์ของโปรเจคที่สร้างขึ้น

5. ใช้คำสั่ง ***idf.py set-target esp32*** เพื่อตั้งให้เป็นรุ่นของ ESP32 และรอจนกว่าจะเสร็จสิ้นได้ผลลัพธ์ดังรูป



รูปที่ 19.0 การตั้งค่ารุ่นให้เป็น ESP32 ที่เสร็จสิ้นแล้ว

6. เปิด Visual Studio Code และเข้าไปที่โฟลเดอร์ของโปรเจค พร้อมเปิดไฟล์ <ชื่อโปรเจค.c> ที่อยู่ในโฟลเดอร์ main



รูปที่ 20.1 การเปิดไฟล์โปรเจคใน Visual Studio Code

7. ทำการเขียนโค้ด (Code) ของโปรแกรมดังนี้

```
main > auto_bin.c > app_main(void)
1  #include <stdio.h>
2  #include "driver/gpio.h" //GPIO
3  #include "freertos/FreeRTOS.h" //FreeRTOS for vTaskDelay
4  #include "freertos/task.h"
5  #include "esp_log.h"
6  #include "driver/mcpwm_prelude.h" // PWM
```

รูปที่ 20.2 ส่วน include header file ที่เกี่ยวข้อง

```
8  // IR PIN CONFIG
9  #define IR_PIN GPIO_NUM_2
10 int val_ir = 0;
11
12 // BUZZER PIN CONFIG
13 #define BUZZER_PIN GPIO_NUM_4
```

รูปที่ 20.3 ส่วนการกำหนดขา IR และขา Active Buzzer

```
// SERVO CONFIG
static const char *TAG = "MCPWM";
#define SERVO_MIN_PULSEWIDTH_US 500
#define SERVO_MAX_PULSEWIDTH_US 2500
#define SERVO_MIN_DEGREE        -90
#define SERVO_MAX_DEGREE        90

#define SERVO_PULSE_GPIO             13

#define SERVO_TIMEBASE_RESOLUTION_HZ 1000000
#define SERVO_TIMEBASE_PERIOD        20000
```

รูปที่ 20.4 ส่วนการกำหนดขา Servo และ ค่าต่างๆ ที่ใช้

```
// SERVO COMPARE
static inline uint32_t example_angle_to_compare(int angle)
{
    return ((angle - SERVO_MIN_DEGREE) * (SERVO_MAX_PULSEWIDTH_US - SERVO_MIN_PULSEWIDTH_US) / (SERVO_MAX_DEGREE - SERVO_MIN_DEGREE) + SERVO_MIN_PULSEWIDTH_US);
}
int angle = 0;
```

รูปที่ 20.5 ฟังก์ชันการแปลงมุมเซอร์โวมอเตอร์เป็นความกว้าง PWM

```
// IR_IN_PIN CONFIG
#define IR_IN_PIN GPIO_NUM_5
#define GREEN_LED_PIN GPIO_NUM_17
#define RED_LED_PIN GPIO_NUM_16
int val_ir_inBin = 0;
```

รูปที่ 20.6 ส่วนการกำหนดขา IR ภายในถัง และขา LED

```
// INPUT PIN
#define GPIO_INPUT_PIN_SET ((1ULL<<IR_PIN) | (1ULL<<IR_IN_PIN))

// OUTPUT PIN
#define GPIO_OUTPUT_PIN_SET ((1ULL<<BUZZER_PIN) | (1ULL<<GREEN_LED_PIN) | (1ULL<<RED_LED_PIN))
```

รูปที่ 20.7 การกำหนดกลุ่มของขา input และ output

โค้ดที่กำลังแสดงทั้งหมดต่อไปนี้อยู่ในฟังก์ชัน app_main()

```
void app_main(void){
    // GPIO INPUT CONFIG
    gpio_config_t i_conf = {};
    i_conf.intr_type = GPIO_INTR_DISABLE;
    i_conf.mode = GPIO_MODE_INPUT;
    i_conf.pin_bit_mask = GPIO_INPUT_PIN_SET;
    i_conf.pull_down_en = 0;
    i_conf.pull_up_en = 1;
    gpio_config(&i_conf);

    // GPIO OUTPUT CONFIG
    gpio_config_t o_conf = {};
    o_conf.intr_type = GPIO_INTR_DISABLE;
    o_conf.mode = GPIO_MODE_OUTPUT;
    o_conf.pin_bit_mask = GPIO_OUTPUT_PIN_SET;
    o_conf.pull_down_en = 0;
    o_conf.pull_up_en = 1;
    gpio_config(&o_conf);
```

รูปที่ 20.8 การตั้งค่า GPIO

```
65      // TIMER FOR SERVO
66      ESP_LOGI(TAG, "Create timer and operator");
67      mcpwm_timer_handle_t timer = NULL;
68      mcpwm_timer_config_t timer_config = {
69          .group_id = 0,
70          .clk_src = MCPWM_TIMER_CLK_SRC_DEFAULT,
71          .resolution_hz = SERVO_TIMEBASE_RESOLUTION_HZ,
72          .period_ticks = SERVO_TIMEBASE_PERIOD,
73          .count_mode = MCPWM_TIMER_COUNT_MODE_UP,
74      };
75      ESP_ERROR_CHECK(mcpwm_new_timer(&timer_config, &timer));
76      mcpwm_oper_handle_t oper = NULL;
77      mcpwm_operator_config_t operator_config = {
78          .group_id = 0,
79      };
80      ESP_ERROR_CHECK(mcpwm_new_operator(&operator_config, &oper));
81      ESP_LOGI(TAG, "Connect timer and operator");
82      ESP_ERROR_CHECK(mcpwm_operator_connect_timer(oper, timer));
83      ESP_LOGI(TAG, "Create comparator and generator from the operator");
84      mcpwm_cmpr_handle_t comparator = NULL;
85      mcpwm_comparator_config_t comparator_config = {
86          .flags.update_cmp_on_tez = true,
87      };
88      ESP_ERROR_CHECK(mcpwm_new_comparator(oper, &comparator_config, &comparator));
```

รูปที่ 20.9 การสร้าง Timer สำหรับเซอร์โวมอเตอร์

```
90      mcpwm_gen_handle_t generator = NULL;
91      mcpwm_generator_config_t generator_config = {
92          .gen_gpio_num = SERVO_PULSE_GPIO,
93      };
94      ESP_ERROR_CHECK(mcpwm_new_generator(oper, &generator_config, &generator));
95      ESP_ERROR_CHECK(mcpwm_comparator_set_compare_value(comparator, example_angle_to_compare(0)));
96      ESP_LOGI(TAG, "Set generator action on timer and compare event");
97
98      ESP_ERROR_CHECK(mcpwm_generator_set_action_on_timer_event(generator,
99                      MCPWM_GEN_TIMER_EVENT_ACTION(MCPWM_TIMER_DIRECTION_UP, MCPWM_TIMER_EVENT_EMPTY, MCPWM_GEN_ACTION_HIGH)));
100
101     ESP_ERROR_CHECK(mcpwm_generator_set_action_on_compare_event(generator,
102                     MCPWM_GEN_COMPARE_EVENT_ACTION(MCPWM_TIMER_DIRECTION_UP, comparator, MCPWM_GEN_ACTION_LOW)));
103
104     ESP_LOGI(TAG, "Enable and start timer");
105     ESP_ERROR_CHECK(mcpwm_timer_enable(timer));
106     ESP_ERROR_CHECK(mcpwm_timer_start_stop(timer, MCPWM_TIMER_START_NO_STOP));
107
108     int angle = 0;
```

รูปที่ 20.10 การสร้าง MCPWM Generator สำหรับควบคุมเซอร์โวมอเตอร์

```
110     // LOOP APP
111     while(1){
112         // IR IN
113         val_ir_inBin = gpio_get_level(IR_IN_PIN);
114         printf("-- -IN BIN Value : %d ---\n", val_ir_inBin);
115         if(val_ir_inBin == 0){
116             printf("-> Bin status : RED!!\n");
117             gpio_set_level(RED_LED_PIN, 1);
118             gpio_set_level(GREEN_LED_PIN, 0);
119         }else{
120             printf("-> Bin status : GREEN\n");
121             gpio_set_level(RED_LED_PIN, 0);
122             gpio_set_level(GREEN_LED_PIN, 1);
123         }
```

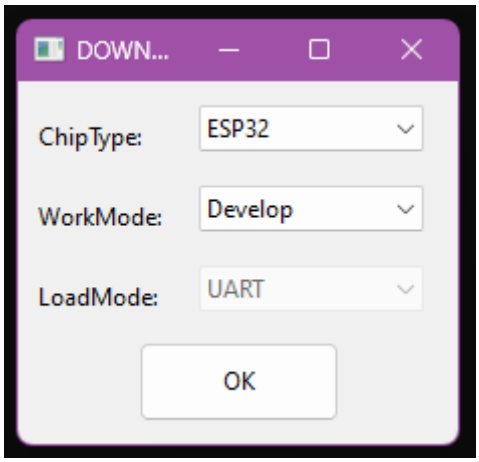
รูปที่ 20.11 โปรแกรมการตรวจสอบขยะภายในถัง

```
        // IR DETECTION
        val_ir = gpio_get_level(IR_PIN);
        printf("--- IR Value : %d ---\n", val_ir);
        if(val_ir == 0){
            printf("-> Bin open!!\n");
            angle = 90;
            ESP_ERROR_CHECK(mcpwm_comparator_set_compare_value(comparator, example_angle_to_compare(angle)));
            gpio_set_level(BUZZER_PIN, 1);
            vTaskDelay(100 / portTICK_PERIOD_MS);
            gpio_set_level(BUZZER_PIN, 0);
            vTaskDelay(3000 / portTICK_PERIOD_MS);
        }else{
            printf("-> Bin close\n");
            angle = -90;
            ESP_ERROR_CHECK(mcpwm_comparator_set_compare_value(comparator, example_angle_to_compare(angle)));
        }
        printf("\n");

            vTaskDelay(100 / portTICK_PERIOD_MS);
        }

}
```

รูปที่ 20.12 โปรแกรมการตรวจสอบวัตถุหน้าถัง และเปิดปิดฝาถัง

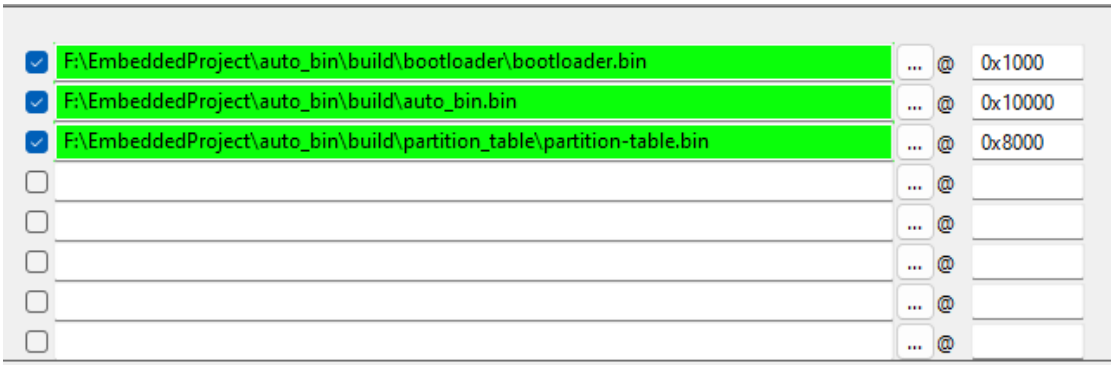
8. ใช้คำสั่ง idf.py build เพื่อแปลงโค้ดภาษา C ในไฟล์โปรเจคให้เป็นภาษาเครื่องที่ ESP32 เข้าใจได้ และเมื่อเสร็จสิ้นแล้วจะได้ผลลัพธ์ดังรูป

9. ใช้โปรแกรม Flash Download Tool เพื่อแฟลชโปรแกรมลงบอร์ด โดยเลือก ChipType เป็น ESP32 และ WorkMode เป็น Develop



รูปที่ 21.0 การเลือกโหมดการทำงาน Flash Download Tool

เลือกไฟล์ที่ต้องการลง ได้แก่ bootloader.bin ลงในตำแหน่ง 0x1000, ชื่อโปรเจค.bin ลงในตำแหน่ง 0x10000, partition-table.bin ลงในตำแหน่ง 0x8000 ดังรูป และทำการแฟลชลงบอร์ด



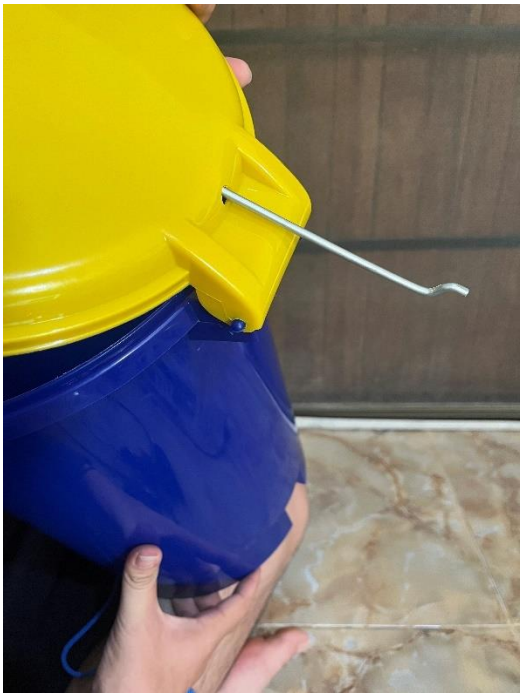
รูปที่ 22.0 เลือกไฟล์ที่ใช้แฟลชลงบอร์ด

3.3 ขั้นตอนการติดตั้งอุปกรณ์ต่างๆ เข้ากับถังขยะ

1. ทำการเจาะด้านหลังของฝาถัง เพื่อติดตั้งแท่งเหล็กสำหรับใช้เป็นก้านดึงในการเปิดปิดด้วยเซอร์โวมอเตอร์

รูปที่ 23.0 การเจาะรูเพื่อติดตั้งก้านดึง

รูปที่ 24.0 การติดตั้งก้านดึง

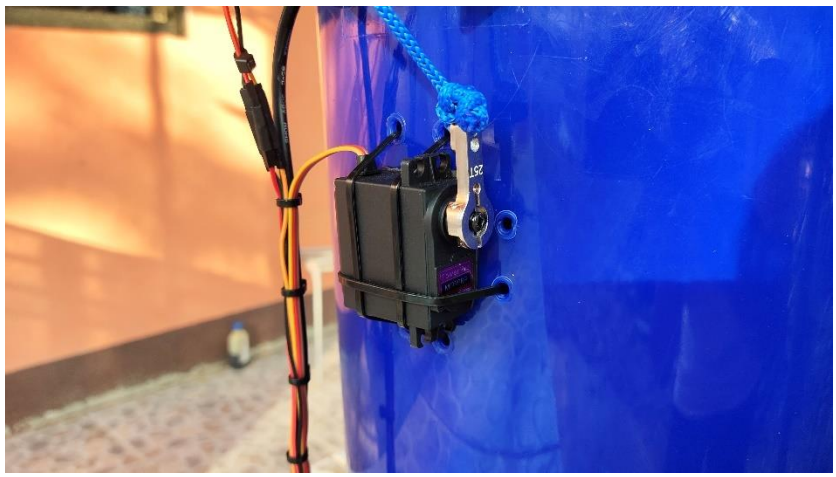


รูปที่ 25.0 การติดตั้งก้านดึงที่เสร็จสมบูรณ์

2. ติดตั้งเซ็นเซอร์ E18-D80NK เข้าที่บริเวณด้านข้างของถัง โดยหันด้านหน้าของเซ็นเซอร์ออกสู่ด้านหน้าของถัง

รูปที่ 26.0 การติดตั้งเซ็นเซอร์ E18-D80NK

3. ติดตั้งเซอร์โวมอเตอร์เข้าที่บริเวณด้านหลังของถังขยะ โดยให้ตรงกับบริเวณก้านดึงของฝาถัง



รูปที่ 27.0 การติดตั้งเซอร์โวมอเตอร์เข้าที่ด้านหลังของถังขยะ

4. ใช้เชือกผูกระหว่างแขนของโซเวอร์มอเตอร์ กับก้านดึงของฝาถังขยะ

รูปที่ 28.0 ผูกเชือกระหว่างแขนของโซเวอร์มอเตอร์ กับก้านดึง

5. ติดตั้งเซ็นเซอร์ IR Infrared เข้าที่บริเวณใต้ฝาถัง หรือภายในถังขยะ

รูปที่ 29.0 ติดตั้งเซ็นเซอร์ IR Infrared

6. ติดตั้ง Module LED เข้าที่บริเวณหน้าถัง

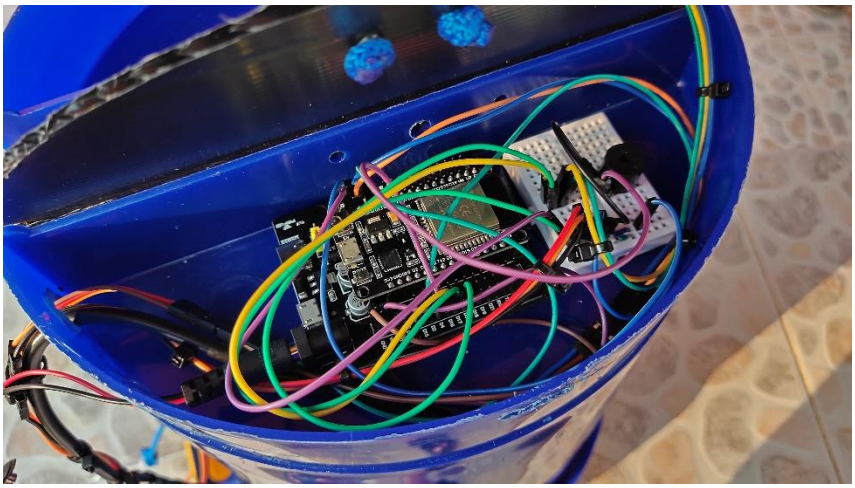
รูปที่ 30.0 ติดตั้ง Module LED

7. รวบรวมสายไฟต่างๆ ของเซ็นเซอร์ เข้าไปสู่ส่วนที่ติดตั้งบอร์ด ESP32

รูปที่ 31.0 การรวบรวมสายไฟต่างๆ ของเซ็นเซอร์

รูปที่ 32.0 การรวบรวมสายไฟต่างๆ เข้าส่วนที่ติดตั้งบอร์ด ESP32

8. ติดตั้งบอร์ด ESP32 และเชื่อมต่อสายไฟต่างๆ ตามแผนผังจำลองในรูปที่ 14



รูปที่ 33.0 การติดตั้งบอร์ด ESP32 และเชื่อมต่อสายไฟต่างๆ

9. ติดฝาเปิดปิดส่วนที่ติดตั้งบอร์ด ESP32

รูปที่ 34.0 ฝาเปิดปิดส่วนที่ติดตั้งบอร์ด ESP32 ขณะเปิด

รูปที่ 35.0 ฝาเปิดปิดส่วนที่ติดตั้งบอร์ด ESP32 ขณะปิด

10. สามารถปรับระยะการตรวจจับของเซ็นเซอร์ E18-D80NK โดยใช้ไขควงได้ตามต้องการ หากหมุนตามเข็มนาฬิกาจะเป็นการเพิ่มระยะการตรวจจับ และหากหมุนทวนเข็มนาฬิกาจะเป็นการลดระยะการตรวจจับ

รูปที่ 36.0 ปรับระยะของเซ็นเซอร์ E18-D80NK

3.4 การทดสอบการใช้งาน

3.4.1 การทดสอบการตรวจจับเพื่อเปิดปิดฝาถัง

เมื่อมีไม่มีวัตถุใดๆ บริเวณหน้าถังขยะ ฝาถังก็จะปิดเป็นปกติ

รูปที่ 37.0 ถังขยะเมื่อไม่มีวัตถุใดๆ อยู่บริเวณด้านหน้า

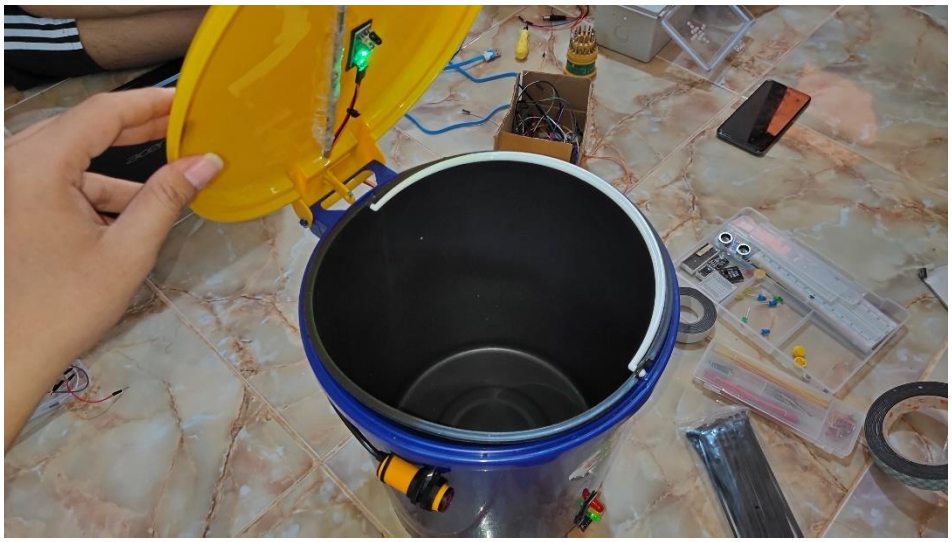
เมื่อมีวัตถุ หรือผู้ใช้อยู่บริเวณหน้าถังในระยะการตรวจจับที่ปรับค่าไว้ จะส่งเสียงแจ้งเตือน และฝาถังจะเปิดตลอด จนกว่าวัตถุนั้น ออกจากระยะการตรวจจับ

รูปที่ 38.1 ถังขยะเมื่อตรวจพบวัตถุด้านหน้าในระยะ 1

รูปที่ 38.1 ถังขยะเมื่อตรวจพบวัตถุด้านหน้าในระยะ 2

3.4.2 การทดสอบการตรวจสอบสถานะความเต็มของขยะภายในถัง

เมื่อถังขยะว่าง หรือไม่เต็ม จะแสดงไฟสถานะสีเขียวที่หน้าถัง



รูปที่ 39.0 ถังขยะเมื่อภายในวางเปล่า

รูปที่ 40.0 ไฟแสดงสถานที่เมื่อภายในถังขยะไม่เต็ม

เมื่อขยะที่อยู่ภายในถังใกล้เต็ม หรือเต็มแล้ว จะแสดงไฟสถานะสีแดงที่บริเวณหน้าถัง

รูปที่ 41.0 ถังขยะเมื่อภายในเต็ม

รูปที่ 42.0 ไฟแสดงสถานที่เมื่อภายในถังขยะเต็ม