

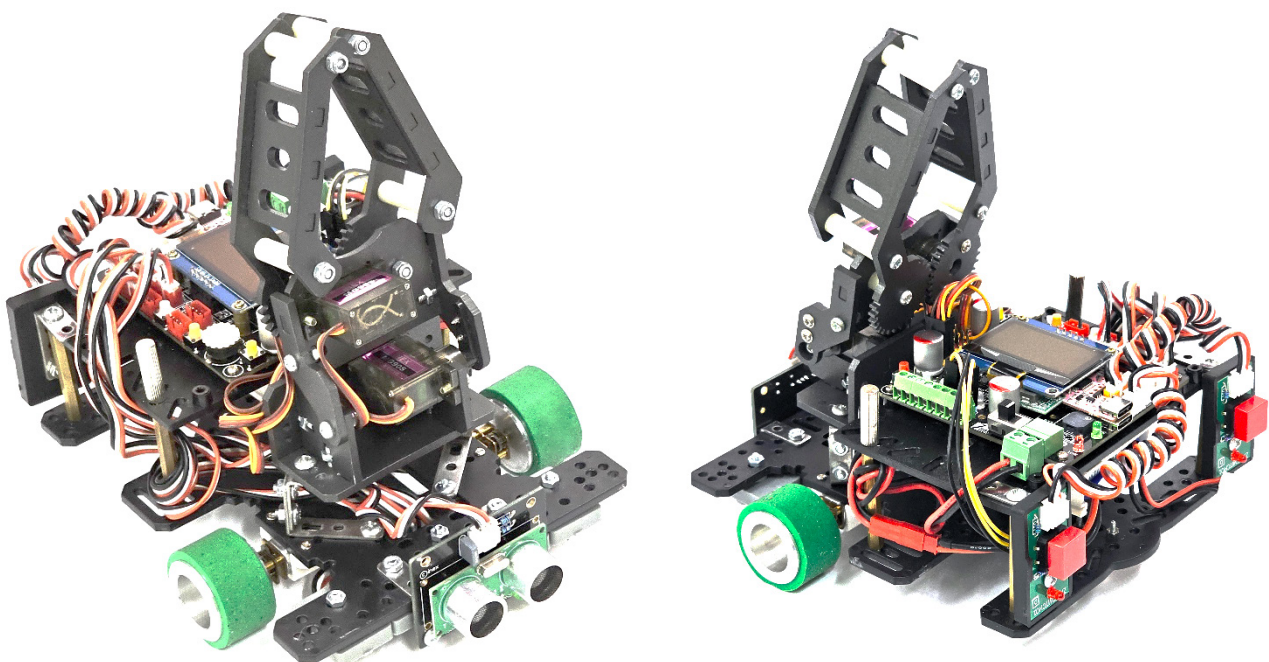
หุ่นยนต์กู้ภัย (ด้านเขาวงกต)

1 แนะนำ

ทางทีมงานออกแบบหุ่นยนต์ 2 ล้อให้เป็นตอนการใช้งานในเกมหุ่นยนต์กู้ภัยอัจฉริยะสำหรับงานมหกรรมแข่งขันเกมหุ่นยนต์ระดับโลก World Robot Games

รายการอุปกรณ์

- บอร์ดควบคุมหลัก POP-32i
- แผงวงจรตรวจจับแสงสีแดงสะท้อน ZX-03R × 5
- แผงวงจรตรวจจับระยะห่างด้วยคลื่นอัลตราโซนิก ZX-SONAR × 1
- แผงวงจรสวิตช์ ZX-Switch01 × 2
- แผ่นฐานหุ่นยนต์ iRC1 Chassis × 1
- มอเตอร์ N20 พร้อมล้อยาง × 2
- อุปกรณ์หนีบจับวัตถุ Gripper-X × 1
- แบตเตอรี่ Li-po 2 เซลล์ 7.4V × 1

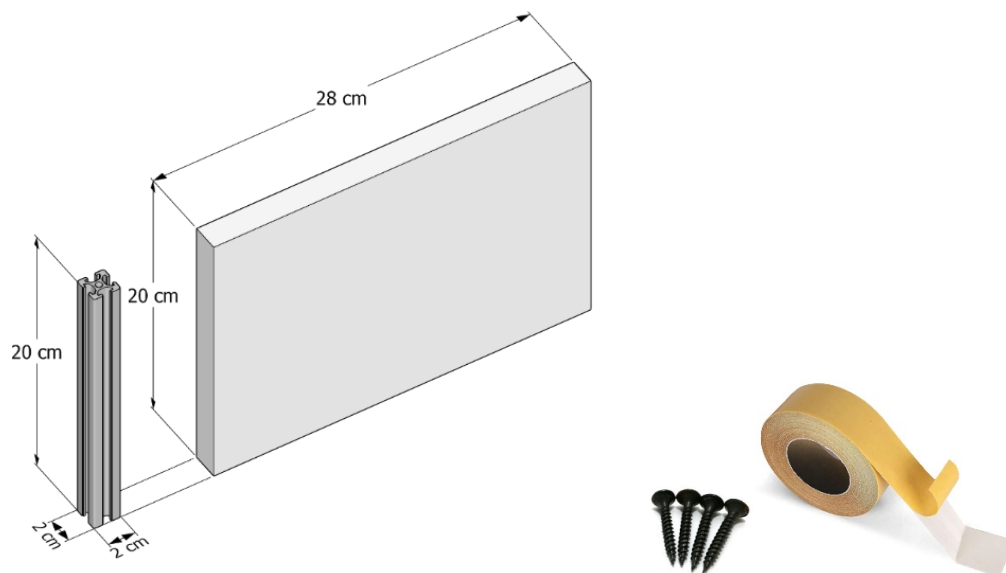


2 การสร้างสนามเขาวงกต

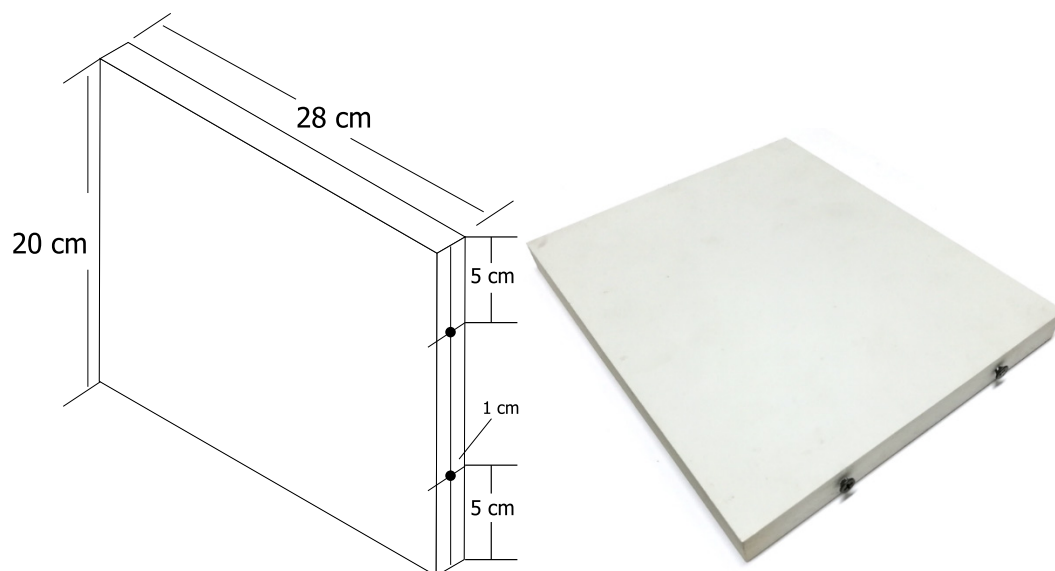
ตัวอย่างนี้การทำสนามให้เหมือนสนามแข่งขันจริง

รายการอุปกรณ์

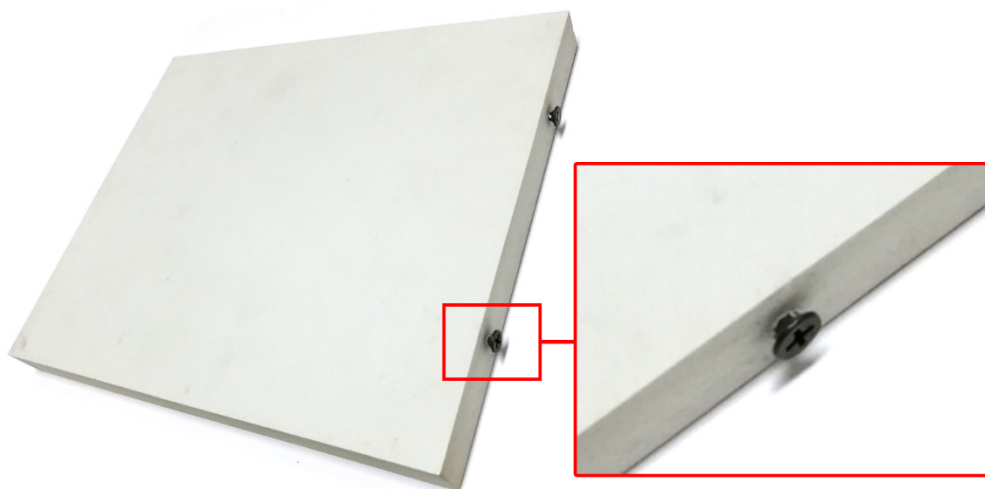
- อะลูมิเนียมโปรไฟล์สี่เหลี่ยม 2x2x20 เซนติเมตร
- แผ่นไม้พลาสติกสี่เหลี่ยม 2x28x20 เซนติเมตร (สร้างกำแพง).
- สกรูหัวตัดเกลียวปล่อย
- พลาสติก (ทำพื้นสนาม)
- เทปกาวติดพรมสองหน้า



เจาะรู 4 จุดเพื่อให้ชั้นสกรูเข้าไปยังแผ่นไม้พลาสติกดังภาพด้านล่าง



สำหรับกำแพงของสนามทำด้วยแผ่นพลาสติกขนาด 28cm x 20 cmหนา 2 cm ด้านข้างจะต้องยึดด้วยสกรูเกลียวป้อยเพื่อเสียบเข้ากับรางของอะลูมิเนียมโปรไฟล์ โดยตำแหน่งสกรูให้วัดจากก้านบนลงมา 5 cm อยู่กึ่งกลางของแผ่นพลาสติกพอดีดังรูป ทำทั้งหมด 4 จุด และร้อยสกรูเกลียวป้อยเข้าไป เหลือระยะห่างประมาณ 3 mm

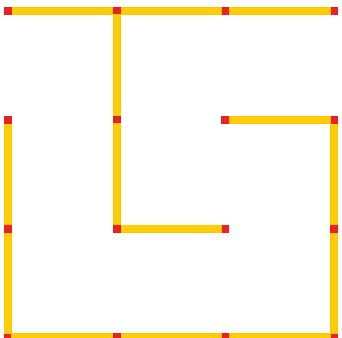
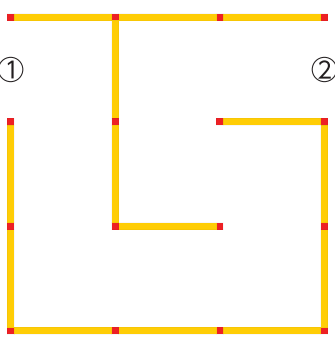
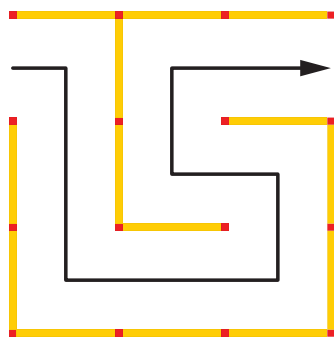


ใช้เทปกาวยึดพรมสองหน้าแปะใต้ขอบแผ่นไม้พลาสติกซึ่งตัดหรืออะลูมิเนียมโปรไฟล์ประกบกับพลาสติกพื้นสนามดังภาพด้านล่าง

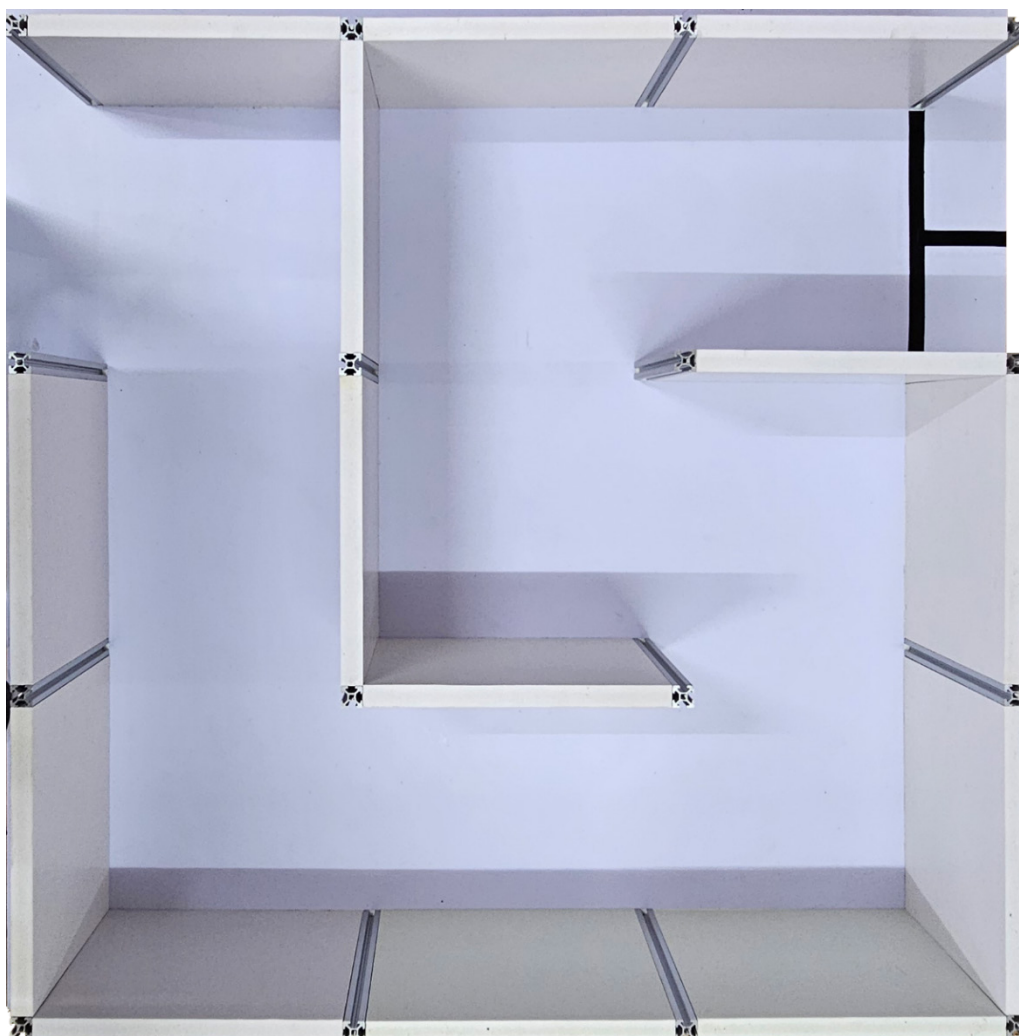


3 ตัวอย่างสนามเขากต

ดังขั้นตอนด้านล่างได้ทำการสร้างตัวอย่างสนามเขาวงกตสำหรับภารกิจเขาวงกต

		
<p>ขั้นตอนที่ 1: สร้างโครงข่ายเขาวงกต</p>	<p>ขั้นตอนที่ 2: กำหนดให้ 1 เป็นจุดเริ่มต้น และ 2 เป็นจุดสิ้นสุด</p>	<p>ขั้นตอนที่ 3: ให้นุ่นยนต์เคลื่อนที่ ทิศทางดังรูปด้านบน</p>

เห็นภาพของจริงดังด้านล่าง



4 การเคลื่อนที่ในเขาวงกต

เขาวงกตที่ถูกออกแบบมีพื้นว่างเปล่าหรือพื้นขาวบนพื้นแต่มีกำแพง ดังนั้นต้องควบคุมการเคลื่อนที่หุ่นยนต์ให้ตรงเท่าที่หุ่นยนต์สามารถทำได้ ในคู่มือใช้หุ่นยนต์ขับเคลื่อนด้วย 2 ล้อ ดังนั้นต้องกำหนดความเร็วมอเตอร์ด้านซ้ายและมอเตอร์ด้านขวาแยกอิสระจากกัน

5 การตรวจจับกำแพง

อย่างที่กล่าวไว้ เขาวงกตที่ถูกออกแบบไม่มีเส้น ดังนั้นต้องใช้เซนเซอร์ตัวอื่นเพื่อให้ตัวจับบางสิ่งบางอย่างในนั้นได้ สำหรับตัวอย่างใช้เซนเซอร์อัลตราโซนิกเพื่อตรวจจับกำแพงโดยวัดระยะห่างระหว่างตัวเซนเซอร์และกำแพง

เพื่อให้มั่นใจว่าหุ่นยนต์อยู่ใกล้ตรงกลางของหนึ่งช่องในขณะวัดซึ่งส่งผลต่อขั้นตอนต่อไป กรณีนี้ ถ้ากำหนดให้ระยะห่างระหว่างหุ่นยนต์และการตรวจจับกำแพงใกล้หรือไกลเกินไปหลังจากหมุนตัวไปทางด้านซ้ายหรือด้านขวาเรียบร้อยแล้ว หุ่นยนต์อาจจะชนหรือไกลจากกำแพงได้

6 เวลาหมุนซ้ายหรือขวา

เมื่อหุ่นยนต์ตรวจจับกำแพง ควรหมุนไปทางด้านซ้ายหรือด้านขวาเพื่อค้นหาเส้นทางที่สามารถไปได้

7 การถอยหลังสัมผัสกำแพง

หุ่นยนต์หมุนไปทางด้านซ้ายหรือด้านขวาโดยใช้ช่วงเวลาเพื่อหมุนไปยัง 90 องศาอย่างแม่นยำ มันยากที่เดาได้ มีสิ่งหนึ่งที่สามารถช่วยคุณได้คือสวิตช์ 2 ตัวติดตั้งหลังตัวหุ่นยนต์ หลังจากนั้นให้หุ่นยนต์ถอยหลังและสัมผัสกำแพง ในขณะที่หุ่นยนต์พยายามตั้งตัวให้ตรงเท่าที่หุ่นยนต์สามารถทำได้

8 อธิบายลำดับการเขียนโปรแกรม

1) ส่วนหัวของโปรแกรม

```
#define SONAR_READ_CM (analog(5) * 100) / 4095
#define SPD 45
#define kpsonar 7
#define kpspin_t 350
#define kpiffr 1500
#define kpfreq 500
#define kpstop_t 300
#include <POP32.h>
```

อธิบาย: ตัวแปรให้อยู่ในรูปมาโคร

- SONAR_READ_CM: ค่าจากการแปลงสูตรจากค่าข้อมูลดิบให้เป็นหน่วยเซนติเมตร
- SPD: ปรับค่าความเร็วมอเตอร์
- kpsonar: ปรับค่าเกนที่ระยะห่างระหว่างตัวเซนเซอร์อัลตราโซนิกและกำแพงในหน่วยเซนติเมตร
- kpspin_t: ปรับค่าเวลาหมุนตัวของหุ่นยนต์ในหน่วยมิลลิวินาที
- kpiffr: ปรับค่าเกนที่เซนเซอร์ตรวจจับแสงสีแดงสะท้อน
- kpfreq: ปรับค่าความถี่เสียงที่ต้องการเตือน
- kpstop_t: ปรับค่าเวลาหยุดเคลื่อนที่ของหุ่นยนต์ในหน่วยมิลลิวินาที

และผนวกไลบรารี POP-32

2) ฟังก์ชัน “เดินไปด้านหน้าจากนั้นหยุดชั่วคราวเมื่อเจอกำแพง”

```
void fdSONAR(int cm, int freq, int ao_t) {
    fd2(SPD - 2, SPD);
    while (SONAR_READ_CM > cm);
    ao();
    sound(freq, ao_t);
}
```

อธิบาย: ใช้ฟังก์ชันขับเคลื่อนหุ่นยนต์ 2 ล้อไปด้านหน้าโดยแยกอิสระของความเร็วมอเตอร์ด้านซ้ายและด้านขวาจากกันเพื่อเดินไปด้านหน้าให้ตรง (ในตัวอย่างนี้ได้ทำการทดสอบหุ่นยนต์เดินไปด้านหน้าเอียงไปทางด้านขวา ดังนั้นต้องลดความเร็วด้านซ้ายลงเพื่อให้หุ่นยนต์เดินตรง)

หุ่นยนต์เดินไปด้านหน้าต่อไปเมื่ออยู่ห่างไกลจากกำแพงจนกระทั่งอยู่ใกล้กำแพง จากนั้นหุ่นยนต์หยุดเคลื่อนที่ชั่วคราว

3) ฟังก์ชัน “หมุนซ้ายแล้วหยุดชั่วคราว”

```
void slMAZE(int sl_t, int ao_t) {  
    sl(SPD);  
    delay(sl_t);  
    ao();  
    delay(ao_t);  
}
```

อธิบาย: ใช้ฟังก์ชันขับเคลื่อนหุ่นยนต์ 2 ล้อหมุนไปทางด้านซ้าย จากนั้นหุ่นยนต์หยุดเคลื่อนที่ชั่วคราว

4) ฟังก์ชัน “หมุนขวาแล้วหยุดชั่วคราว”

```
void srMAZE(int sr_t, int ao_t) {  
    sr(SPD);  
    delay(sr_t);  
    ao();  
    delay(ao_t);  
}
```

อธิบาย: ใช้ฟังก์ชันขับเคลื่อนหุ่นยนต์ 2 ล้อหมุนไปทางด้านขวา จากนั้นหุ่นยนต์หยุดเคลื่อนที่ชั่วคราว

5) ฟังก์ชัน “ถอยไปด้านหลัง สัมผัสกำแพง และตั้งตัวให้ตรงโดยสวิทช์ 2 ตัว”

```
void bkSW(int freq, int ao_t) {  
    while (1) {  
        if (!in(22) && !in(23)) {  
            bk(SPD);  
            sound(freq, ao_t);  
            ao();  
            delay(ao_t);  
            break;  
        } else if (in(22) && !in(23)) {  
            sl(SPD);  
        } else if (!in(22) && in(23)) {  
            sr(SPD);  
        } else {  
            bk(SPD);  
        }  
    }  
}
```

อธิบาย: ถ้าสวิทช์ยังไม่ถูกสัมผัส หุ่นยนต์จะเดินถอยหลังตรง

ถ้าสวิทช์ด้านขวาถูกสัมผัส หุ่นยนต์พยายามหมุนไปทางด้านซ้าย

ถ้าสวิทช์ด้านซ้ายถูกสัมผัส หุ่นยนต์พยายามหมุนไปทางด้านขวา

ถ้าสวิทช์ทั้ง 2 ตัวถูกสัมผัส มอเตอร์จะเคลื่อนที่ถอยหลังในขณะที่สัมผัสกำแพงชั่วคราวเพื่อให้แน่ใจว่าหุ่นยนต์ตั้งตัวตรง จากนั้นหยุดและออกจากวนซ้ำโดยใช้คำสั่ง “break”

6) ฟังก์ชัน “ตรวจจับและหยุดที่เส้น”

```
void stopline(int ifr, int freq, int ao_t) {  
    fd2(SPD - 2, SPD);  
    while (analog(2) > ifr);  
    ao();  
    sound(freq, ao_t);  
}
```

อธิบาย: ฟังก์ชันนี้เหมือนฟังก์ชันข้อที่ 2 แต่ใช้เซนเซอร์แสงสีแดงสะท้อนเพื่อตรวจจับเส้นสีดำแทนที่เซนเซอร์อัลตราโซนิก

7) ฟังก์ชัน “ลำดับการทำงานหลักในภารกิจ”

```
void setup() {  
    // put your setup code here, to run once:  
}  
void loop() {  
    // put your main code here, to run repeatedly:  
    waitSW_OK_bmp();  
    fdSONAR(kpsonar, kpfreq, kpstop_t);  
    srMAZE(kpspin_t, kpstop_t);  
    bkSW(kpfreq, kpstop_t);  
    fdSONAR(kpsonar, kpfreq, kpstop_t);  
    slMAZE(kpspin_t, kpstop_t);  
    bkSW(kpfreq, kpstop_t);  
    fdSONAR(kpsonar, kpfreq, kpstop_t);  
    slMAZE(kpspin_t, kpstop_t);  
    bkSW(kpfreq, kpstop_t);  
    fdSONAR(kpsonar, kpfreq, kpstop_t);  
    srMAZE(kpspin_t, kpstop_t);  
    bkSW(kpfreq, kpstop_t);  
    fdSONAR(kpsonar, kpfreq, kpstop_t);  
    srMAZE(kpspin_t, kpstop_t);  
    bkSW(kpfreq, kpstop_t);  
    stopline(kpifr, kpfreq, kpstop_t);  
}
```

อธิบาย: รอให้กดปุ่ม OK แล้วปล่อยให้หุ่นยนต์ทำภารกิจตามลำดับ เมื่อทำภารกิจเสร็จสิ้นแล้ว เริ่มรอกดปุ่ม OK อีก

ครั้ง

9 Code Download



<https://pastebin.com/DwQiF8US>

10 Video Demo



<https://youtu.be/uzlP8nUF6zk>

THA Rev1.0 20240919