

บทที่ 14

การเขียนโปรแกรมควบคุมหุ่นยนต์อัตโนมัติ

จุดประสงค์ทั่วไป

1. เพื่อให้ นักศึกษามีความรู้ความเข้าใจเกี่ยวกับงานเขียนโปรแกรมควบคุมหุ่นยนต์อัตโนมัติ

จุดประสงค์เชิงพฤติกรรม (เพื่อให้ผู้เรียนสามารถ)

1. เลือกใช้โปรแกรมสำหรับงานเขียนโปรแกรมควบคุมหุ่นยนต์อัตโนมัติได้อย่างถูกต้องตามความต้องการ
2. บอกขั้นตอนการติดตั้งโปรแกรมสำหรับงานเขียนโปรแกรมควบคุมหุ่นยนต์อัตโนมัติได้อย่างถูกต้องตามความต้องการ
3. อธิบายหลักการทำงานของหุ่นยนต์อัตโนมัติได้อย่างถูกต้อง
4. อธิบายขั้นตอนงานเขียนโปรแกรมควบคุมหุ่นยนต์อัตโนมัติได้อย่างถูกต้อง

ทฤษฎีการเรียนรู้งานเขียนโปรแกรมควบคุมหุ่นยนต์อัตโนมัติ

ชุดคำสั่งหรือที่นิยมเรียกว่าฟังก์ชัน ของภาษา C/C++ ของ Arduino มีมากมายให้เลือกใช้งานได้ตามความเหมาะสมกับการทำงาน แต่ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของหุ่นยนต์สามารถเลือกใช้งานบางฟังก์ชันตามความจำเป็น โดยการใช้งานทุกฟังก์ชันของภาษา C/C++ ของ Arduino สามารถศึกษาเพิ่มเติมจากเว็บไซต์อย่างเป็นทางการของ Arduino คือ www.arduino.cc หรือ www.arduino.org

ฟังก์ชันเพื่อการใช้งานควบคุมหุ่นยนต์ Arduino Wireless Robot สามารถแบ่งได้เป็น 6 ประเภท คือ

1. ฟังก์ชันอินพุตและเอาต์พุตดิจิทัล
2. ฟังก์ชันเอาต์พุตแอนะล็อก
3. ฟังก์ชันเกี่ยวกับเวลา
4. ฟังก์ชันเกี่ยวกับการสื่อสารผ่านพอร์ตอนุกรม
5. ฟังก์ชันเกี่ยวกับการควบคุมเซอร์โวมอเตอร์
6. ฟังก์ชันสำหรับการรับสัญญาณจากจอยสติ๊กแบบ PS2

14.1 ฟังก์ชันอินพุตและเอาต์พุตดิจิทัล

เป็นฟังก์ชันที่ใช้สำหรับการกำหนดประเภทของขาพอร์ต

การควบคุมทิศทางของมอเตอร์ตาม ขาพอร์ตที่ต้องการ

1. ฟังก์ชัน pinMode ใช้กำหนดขาพอร์ตที่ต้องการ ให้เป็นพอร์ตดิจิทัล

รูปแบบการใช้งาน `pinMode(pin,mode)`

พารามิเตอร์

pin คือ หมายเลขขาพอร์ตของ Arduino UNO (2-13 โดยขา 0, 1

สำหรับการสื่อสาร) mode คือ กำหนดโหมดการทำงานเป็น INPUT หรือ OUTPUT

(ใช้ตัวพิมพ์ใหญ่) ตัวอย่างเช่น

`pinMode(13,OUTPUT)` //กำหนดให้พอร์ต 13 เป็น Digital Output

`pinMode(5,INPUT)` //กำหนดให้พอร์ต 5 เป็น Digital Input

2. ฟังก์ชัน digitalWrite

ใช้กำหนดขาพอร์ตที่ต้องการ ให้เป็นพอร์ตดิจิทัล

รูปแบบการใช้งาน `digitalWrite(pin,value)`

พารามิเตอร์

pin คือ หมายเลขขาพอร์ตของ Arduino UNO (2-13 โดยขา 0, 1 สำหรับการสื่อสาร)

value คือ กำหนดค่า OUTPUT กำหนดได้ 2 ค่า คือ LOW หรือ HIGH

ตัวอย่างเช่น

`digitalWrite(13,LOW)` //กำหนดค่าออกพอร์ต 13 เป็น LOW

`digitalWrite(13,HIGH)` //กำหนดค่าออกพอร์ต 13 เป็น HIGH

14.2 ฟังก์ชันเอาต์พุตแอนะล็อก

เป็นฟังก์ชันที่ใช้สำหรับการกำหนดความเร็วของมอเตอร์ โดยใช้เทคนิค PWM ในการควบคุม ซึ่งจะควบคุมความเร็วของมอเตอร์ผ่านบอร์ดขับเคลื่อนมอเตอร์ L298N

1. ฟังก์ชัน analogWrite

ใช้กำหนดขาพอร์ตที่ต้องการ ให้สร้างสัญญาณ PWM

รูปแบบการใช้งาน `analogWrite(pin,value)`

พารามิเตอร์

pin คือ หมายเลขขาพอร์ตของ Arduino UNO (3, 5, 6, 9, 10, 11)

mode คือ ค่า Duty cycle มีค่า 0-255 (ตัวแปรแบบ Byte)

หมายเหตุ

- ขาพอร์ตที่สามารถใช้ฟังก์ชัน analogWrite ได้ คือ ขา 3, 5, 6, 9, 10 และ 11
- ค่าความถี่ของสัญญาณ PWM มีค่า 490Hz

ตัวอย่างเช่น

- analogWrite(3,127)

//กำหนดให้พอร์ต 3 ส่งสัญญาณ PWM ที่มี Duty cycle ประมาณ 50%

//สามารถคำนวณได้จาก $(127/255) \times 100 = 49.80\%$

//กรณีที่มอเตอร์หมุนด้วยความเร็ว 3,600rpm (3,600 รอบ/นาที) สามารถคำนวณความเร็วต่อรอบของมอเตอร์ได้จาก $(50/100) \times 3,600 = 1,800\text{rpm}$

14.3 ฟังก์ชันเกี่ยวกับเวลา

เป็นฟังก์ชันที่ใช้สำหรับการใช้สำหรับหน่วยเวลาหรือคงสถานะของการท างาน เพื่อลดความผิดพลาดจากการกดปุ่มจอยสติ๊ก PS2 ซึ่งต้องก หนดค่าที่เหมาะสมจึงจะท างานได้ดี

1. ฟังก์ชัน delay

ใช้สำหรับหน่วยเวลาหรือคงสถานะของการท างานที่กระท ากผ่านมา

รูปแบบการใช้งาน delay(ms)

พารามิเตอร์

ms คือ ตัวเลขก หนดระยะเวลาที่ต้องการหน่วยเวลา ($1,000\text{ms} = 1\text{ s} = 1\text{ วินาที}$)

ตัวอย่างเช่น

Delay(1500) //หน่วยเวลาหรือคงสถานะการท างาน เป็นระยะเวลา 1.5 วินาที

14.4 ฟังก์ชันเกี่ยวกับการสื่อสารผ่านพอร์ตอนุกรม

เป็นฟังก์ชันที่ใช้สำหรับการสื่อสารระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino กับเครื่องคอมพิวเตอร์ เพื่อใช้สำหรับตรวจสอบความถูกต้องในการรับส่งสัญญาณระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino กับจอยสติ๊ก PS2 โดยจะแสดงผลทางหน้าจอของเครื่องคอมพิวเตอร์

1. ฟังก์ชัน Serial.begin

เป็นฟังก์ชันการสื่อสารผ่านพอร์ตอนุกรม ใช้ก หนดค่าอัตราการรับส่งข้อมูลแบบอนุกรม มีหน่วยเป็นบิตต่อวินาที(bits per second : bps)

รูปแบบการใช้งาน Serial.begin (int datarate);

พารามิเตอร์

Serial.begin คือ การก หนดค่าอัตราการรับส่งข้อมูลอนุกรม

Int datarate คือ อัตราการรับส่งข้อมูลอนุกรม กำหนดเป็นตัวแปรชนิด int สามารถ เลือกกำหนด ได้แก่ 300, 1200, 4800, 9600, 14400, 19200, 28800, 38400, 57600 หรือ 115200 bps

ตัวอย่างเช่น

- Serial.begin (9600); //กำหนดอัตราการรับส่งข้อมูลอนุกรม 9600 bps

2. ฟังก์ชัน Serial.print

ใช้พิมพ์ข้อมูลแบบแสดงข้อความต่อเนื่องในบรรทัดเดียวกัน ออกทางพอร์ตอนุกรม ใช้คู่กับ ฟังก์ชัน Serial.begin

รูปแบบการใช้งาน **Serial.print (data);**

พารามิเตอร์

Serial.print คือ พิมพ์ข้อมูลออกพอร์ตอนุกรม แบบต่อเนื่อง

data คือ ข้อมูลที่ต้องการพิมพ์ออก แสดงผลหน้าจคอมพิวเตอร์

ตัวอย่างเช่น

- Serial.print ("Hello ROBOT"); //พิมพ์ข้อมูลข้อความ Hello ROBOT ออกทางพอร์ตอนุกรม

3. ฟังก์ชัน Serial.println

ใช้พิมพ์ข้อมูลแบบขึ้นบรรทัดใหม่หลังจบข้อความ ออกทางพอร์ตอนุกรม ใช้คู่กับฟังก์ชัน Serial.begin

รูปแบบการใช้งาน **Serial.println (data);**

พารามิเตอร์

Serial.println คือ พิมพ์ข้อมูลออกพอร์ตอนุกรม แบบขึ้นบรรทัดใหม่

data คือ ข้อมูลที่ต้องการพิมพ์ออก แสดงผลหน้าจคอมพิวเตอร์

ตัวอย่างเช่น

- Serial.println ("Hello ROBOT"); //พิมพ์ข้อมูลข้อความ Hello ROBOT ออกทางพอร์ตอนุกรม แบบขึ้นบรรทัดใหม่หลังจบข้อความ

14.5 ฟังก์ชันเกี่ยวกับการควบคุมเซอร์โวมอเตอร์

เป็นฟังก์ชันที่ใช้สำหรับการควบคุมเซอร์โวมอเตอร์ของชุดปากคิบ เพื่อให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการ ทั้งการยกชิ้นงานและหนีบจับชิ้นงาน

1. ฟังก์ชัน Servo class

ใช้ควบคุมเซอร์โวมอเตอร์ที่ต่อกับ Arduino UNO โดยใช้ความสามารถของไลบรารี Servo.h เพื่อให้เรียกใช้งานในโปรแกรมได้ง่าย

รูปแบบการใช้งาน Servo class;

พารามิเตอร์

Servo คือ การเรียกใช้งานเซอร์โวมอเตอร์

class คือ ชื่อตัวแปรของ Servo class ซึ่งสามารถใช้ชื่ออื่นได้ เช่น myservo หรือ servomotor หรือ servo1 หรืออื่นๆ ที่ไม่ตรงกับคำสงวน

ตัวอย่างเช่น

- Servo myservo; //การใช้งานเซอร์โวมอเตอร์ในโปรแกรมจะต้องเรียกใช้ด้วยคำว่า myservo

2. ฟังก์ชัน attach()

ใช้กำหนดขาพอร์ตที่ต้องการ เพื่อสร้างสัญญาณพัลส์ไปควบคุมเซอร์โวมอเตอร์

รูปแบบการใช้งาน myservo.attach(pin);

พารามิเตอร์

myservo คือ การเรียกใช้งานเซอร์โวมอเตอร์จากการสร้างตัวแปรในฟังก์ชัน Servo class;

attach คือ การกำหนดขาพอร์ตที่ต้องการ

pin คือ หมายเลขขาพอร์ตของ Arduino UNO แนะนำ ให้ใช้พอร์ต 2-13

ตัวอย่างเช่น

- myservo.attach(12); //กำหนดให้พอร์ต 12 ใช้ในการสร้างสัญญาณควบคุมเซอร์โวมอเตอร์

3. ฟังก์ชัน write()

ใช้ส่งสัญญาณพัลส์ไปควบคุมเซอร์โวมอเตอร์ โดยการกำหนดเป็นองศา

รูปแบบการใช้งาน myservo.write(degrees);

พารามิเตอร์

myservo คือ การเรียกใช้งานเซอร์โวมอเตอร์จากการสร้างตัวแปรในฟังก์ชัน Servo class;

write คือ การส่งสัญญาณพัลส์ไปควบคุมเซอร์โวมอเตอร์

degrees คือ องศาที่ต้องการให้เคลื่อนที่ไป มีค่าตั้งแต่ 0-180

ตัวอย่างเช่น

- myservo.write(90); //ส่งสัญญาณไปควบคุมเซอร์โวมอเตอร์ชื่อ myservo ให้ไปที่

ตำแหน่ง 90 องศา

14.6 การควบคุมความเร็วของมอเตอร์ด้วยวิธีการ PWM

การควบคุมความเร็วของมอเตอร์สามารถท

าได้ด้วยวิธีการลดแรงดันไฟฟ้าที่ป้อนเข้ามอเตอร์ แต่วิธีการที่นิยมใช้งานและท

าได้ง่ายจากฟังก์ชันการท างานของไมโครคอนโทรลเลอร์ทุกตระกูล คือ วิธีการ PWM (Pulse Wide Modulator) ซึ่งก หนดความเร็วของมอเตอร์ด้วยการก หนดความ กว้างของสัญญาณพัลส์

(ค่าเปอร์เซ็นต์ดีวตี้ไซเคิล : Duty Cycle) มีค่าตั้งแต่ 0% - 100% Duty Cycle สามารถ

คำนวณค่าแรงดันเฉลี่ยได้ดังตัวอย่างต่อไปนี้

- Duty Cycle 100% = $(100/100) \times 5V = 5V$

แรงดันเฉลี่ย Duty Cycle 100% คือ 5V

- Duty Cycle 75% = $(75/100) \times 5V = 3.75V$

แรงดันเฉลี่ย Duty Cycle 75% คือ 3.75V

- Duty Cycle 50% = $(50/100) \times 5V = 2.5V$

แรงดันเฉลี่ย Duty Cycle 50% คือ 2.5V

- Duty Cycle 25% = $(25/100) \times 5V = 1.25V$

แรงดันเฉลี่ย Duty Cycle 25% คือ 1.25V

ที่มา : พิลิฐ พางาม. (2562). หุ่นยนต์เบื้องต้น(Introduction to Robotics). นนทบุรี : บริษัท
ศูนย์หนังสือ เมืองไทย จ ากัด

14.7 ตัวอย่างการเขียนโปรแกรมควบคุมหุ่นยนต์เดินตามเส้น 2 เซ็นเซอร์ Arduino +

L298P เปิดโปรแกรม Arduino (IDE) และ Upload โค้ดนี้ ไปยัง บอร์ด Arduino UNO

```
R3 /*-----defining Inputs-----*/
```

```
int LS = 2; // left sensor
```

```
int RS = 3; // right sensor
```

```
/*-----defining Outputs-----*/
```

```
int LM1 = 4; // left motor
```

```
int LM2 = 5; // left motor
```

```
int RM1 = 6; // right motor
```

```
int RM2 = 7; // right motor
```

```
/*-----PWM ควบคุมความเร็วมอเตอร์-----*/
```

```
int PWM1 = 10; // PWM left motor
```

```
int PWM2 = 11; // PWM right motor
```

```
int SPEED = 90; // Speed PWM สามารถปรับความเร็วได้ถึง 255
```

```
void setup()
```

```
{
```

```
    pinMode(LS, INPUT);
```

```
    pinMode(RS, INPUT);
```

```
    pinMode(LM1, OUTPUT);
```

```
    pinMode(LM2, OUTPUT);
```

```
    pinMode(RM1, OUTPUT);
```

```
    pinMode(RM2, OUTPUT);
```

```
    pinMode(PWM1, OUTPUT);
```

```
    pinMode(PWM2, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
    if (!(digitalRead(LS)) && !(digitalRead(RS))) // Move Forward
```

```
    {
```

```
        analogWrite(PWM1, SPEED);
```

```
        digitalWrite(LM1, HIGH);
```

```
        digitalWrite(LM2, LOW);
```

```
        analogWrite(PWM2, SPEED);
```

```
        digitalWrite(RM1, HIGH);
```

```
        digitalWrite(RM2, LOW);
```

```
    }
```

```
    if (!(digitalRead(LS)) && digitalRead(RS)) // Turn right
```

```
    {
```

```
        digitalWrite(LM1, LOW);
```

```

digitalWrite(LM2, LOW);
analogWrite(PWM2, SPEED);

digitalWrite(RM1, HIGH);
digitalWrite(RM2, LOW);
}
if (digitalRead(LS) && !(digitalRead(RS))) // turn left
{
    analogWrite(PWM1, SPEED);
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
}
if (digitalRead(LS) && digitalRead(RS)) // stop
{
    digitalWrite(LM1, LOW);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
}
}

```

ที่มา : <http://www.robotsiam.com/article/12/>