

ภาคผนวก

ภาคผนวก ก
โปรแกรมไมโครคอนโทรลเลอร์

โปรแกรมไมโครคอนโทรลเลอร์

```

int Solenoid_A=2 ; // ประกาศตัวแปรชื่อ Solenoid_A เป็นแบบint มีค่าเท่ากับ 2
int Solenoid_B=3 ; // ประกาศตัวแปรชื่อ Solenoid_B เป็นแบบ int มีค่าเท่ากับ 3
int Solenoid_A2=6 ; // ประกาศตัวแปรชื่อ Solenoid_A2 เป็นแบบ int มีค่าเท่ากับ 6
int Solenoid_B2=7 ; // ประกาศตัวแปรชื่อ Solenoid_B2 เป็นแบบ int มีค่าเท่ากับ 7

int a,b,c,d; // ประกาศตัวแปรชื่อ a,b,c,d เป็นแบบ int

void setup() // ฟังก์ชัน setup เป็นฟังก์ชันสำหรับการกำหนดค่าการทำงานสำหรับขาสัญญาณต่างๆ
{
    pinMode(Solenoid_A,OUTPUT); //ให้ Solenoid_A เป็นเอาต์พุต
    pinMode(Solenoid_B,OUTPUT); //ให้ Solenoid_B เป็นเอาต์พุต
    pinMode(Solenoid_A2,OUTPUT); //ให้ Solenoid_A2 เป็นเอาต์พุต
    pinMode(Solenoid_B2,OUTPUT); //ให้ Solenoid_B2 เป็นเอาต์พุต

    digitalWrite(Solenoid_A,LOW); //ให้ Solenoid_A ไม่มีสัญญาณออก
    digitalWrite(Solenoid_A2,LOW); //ให้ Solenoid_B ไม่มีสัญญาณออก
    digitalWrite(Solenoid_B,LOW); //ให้ Solenoid_A2 ไม่มีสัญญาณออก
    digitalWrite(Solenoid_B2,LOW); //ให้ Solenoid_B2 ไม่มีสัญญาณออก

    Serial.begin(19200); //กำหนดอัตราการรับส่งข้อมูลเป็น 19200 บิตต่อวินาที
}

void loop() // ฟังก์ชัน loop เป็นฟังก์ชันที่ทำงานแบบวนลูปทำงาน
{
    char command; //ประกาศตัวแปรชื่อ command เป็นแบบตัวหนังสือ
    command = Serial.read(); //ให้ command เท่ากับค่าที่รับจากพอร์ตอนุกรมออกมา
    if(command=='R') //ถ้าค่าที่รับเท่ากับ R มาทำเงื่อนไขนี้และให้ตัวแปร a=1
    {
        a=1;
    }
}

```

```

if(a==1) //ถ้า a=1มาทำเงื่อนไขนี้
{
    PWM('A'); //ไปฟังก์ชันย่อยชื่อ PWM('A')
    analogWrite(Solenoid_A,127); //กำหนดความถี่ให้เกิ้ล Solenoid_A เป็น 50 %
    analogWrite(Solenoid_B,0); //กำหนดความถี่ให้เกิ้ล Solenoid_B เป็น 0 %
    b=0;
}

if(command=='L') //ถ้าค่าที่รับเท่ากับ L มาทำเงื่อนไขนี้และให้ตัวแปร b=1
{
    b=1;
}

if(b==1) //ถ้า b=1มาทำเงื่อนไขนี้
{
    PWM('B'); //ไปฟังก์ชันย่อยชื่อ PWM('B')
    analogWrite(Solenoid_B,127); //กำหนดความถี่ให้เกิ้ล Solenoid_B เป็น 50 %
    analogWrite(Solenoid_A,0); //กำหนดความถี่ให้เกิ้ล Solenoid_A เป็น 0 %
    a=0;
}

if(command=='I') //ถ้าค่าที่รับเท่ากับ I มาทำเงื่อนไขนี้และให้ตัวแปร c=1
{
    c=1;
}

if(c==1) //ถ้า c=1มาทำเงื่อนไขนี้
{
    PWM('C'); //ไปฟังก์ชันย่อยชื่อ PWM('C')
    analogWrite(Solenoid_A2,127); //กำหนดความถี่ให้เกิ้ล Solenoid_A2 เป็น 50 %
    analogWrite(Solenoid_B2,0); //กำหนดความถี่ให้เกิ้ล Solenoid_B2 เป็น 0 %
    d=0;
}

```

```

if(command=='B') //ถ้าค่าที่รับเท่ากับ B มาทำเงื่อนไขและให้ตัวแปร d=1
{
    d=1;
}
if(d==1) //ถ้า d=1มาทำเงื่อนไข
{
    PWM('D'); //ไปฟังก์ชันย่อยชื่อ PWM('C')
    analogWrite(Solenoid_B2,127); //กำหนดคิวดิจิทัล Solenoid_B2 เป็น 50 %
    analogWrite(Solenoid_A2,0); //กำหนดคิวดิจิทัล Solenoid_A2 เป็น 0 %
    c=0;
}
if(command=='S') //ถ้าค่าที่รับเท่ากับ S มาทำเงื่อนไขและให้ไปฟังก์ชันย่อย Stop
{
    Stop();
}
if(command=='M') //ถ้าค่าที่รับเท่ากับ M มาทำเงื่อนไขและให้ไปฟังก์ชันย่อย Stop2
{
    Stop2();
}
}

void PWM(unsigned char pwm) //ฟังก์ชันย่อย PWM และประกาศตัวแปรในวงเล็บรับค่า
{
    if(pwm=='A') //ถ้า pwm เท่ากับ A ให้มาทำเงื่อนไข
    {
        TCCR3B = TCCR3B & 0b11111000|0x05; //ฟังก์ชันสร้าง PWM ของTime 3 จะสร้างที่
        30 เฮิรตซ์ออกขา 2,3,5
    }
    if(pwm=='B') //ถ้า pwm เท่ากับ B ให้มาทำเงื่อนไข
    {

```

```

        TCCR3B = TCCR3B & 0b11111000|0x05; //ฟังก์ชันสร้าง PWM ของTime 3 จะสร้างที่
30 เวิร์ทออกขา 2,3,5
    }
    if(pwm=='C') //ถ้า pwm เท่ากับ C ให้มาทำเงื่อนไขนี้
    {
        TCCR4B = TCCR4B & 0b11111000|0x05; //ฟังก์ชันสร้าง PWM ของTime 4 จะสร้างที่
30 เวิร์ทออกขา 6,7,8
    }
    if(pwm=='D') //ถ้า pwm เท่ากับ D ให้มาทำเงื่อนไขนี้
    {
        TCCR4B = TCCR4B & 0b11111000|0x05; ; //ฟังก์ชันสร้าง PWM ของTime 4 จะสร้างที่
30 เวิร์ทออกขา 6,7,8
    }
}

void Stop() //ฟังก์ชันย่อย Stop ให้ค่าตัวแปร a,b, Solenoid_A, Solenoid_B มีค่าเท่ากับ 0
{
    a=0;
    b=0;
    analogWrite(Solenoid_A,0);
    analogWrite(Solenoid_B,0);
}

void Stop2()//ฟังก์ชันย่อย Stop2 ให้ค่าตัวแปร c,d, Solenoid_A2, Solenoid_B2 มีค่าเท่ากับ 0
{
    c=0;
    d=0;
    analogWrite(Solenoid_A2,0);
    analogWrite(Solenoid_B2,0);
}

```