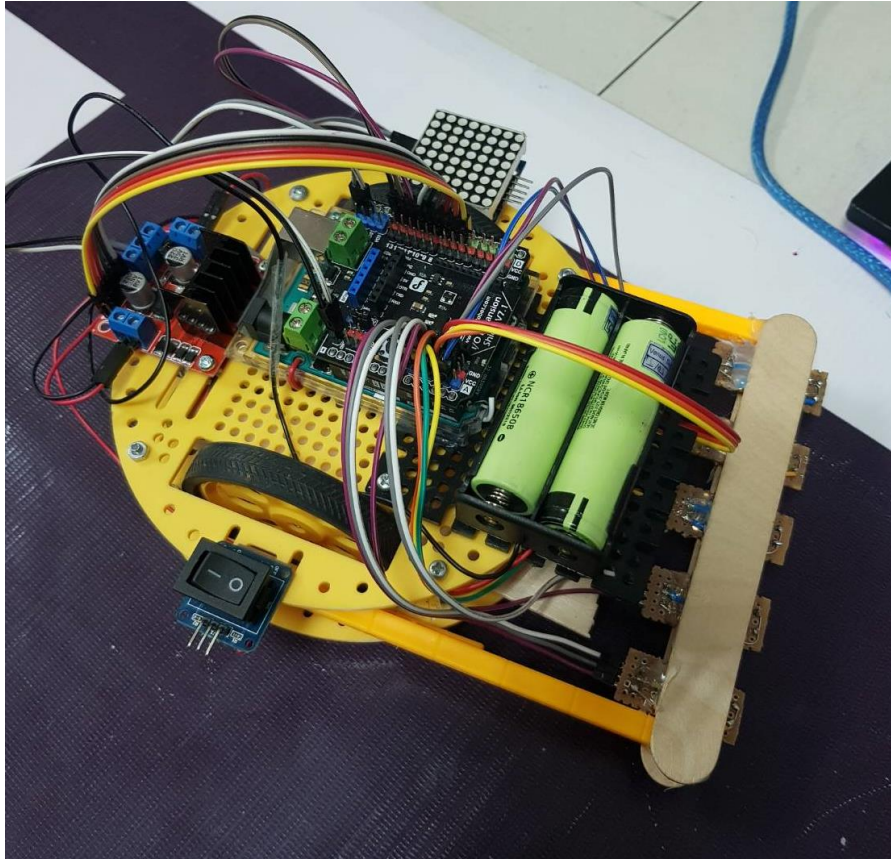
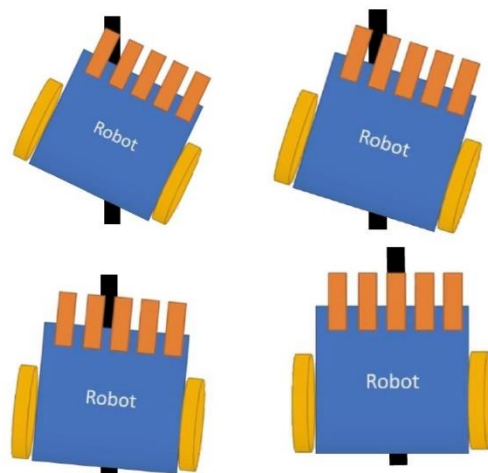


62010711 - 62011019 - 62011044

Project#2 Robot Car Contest

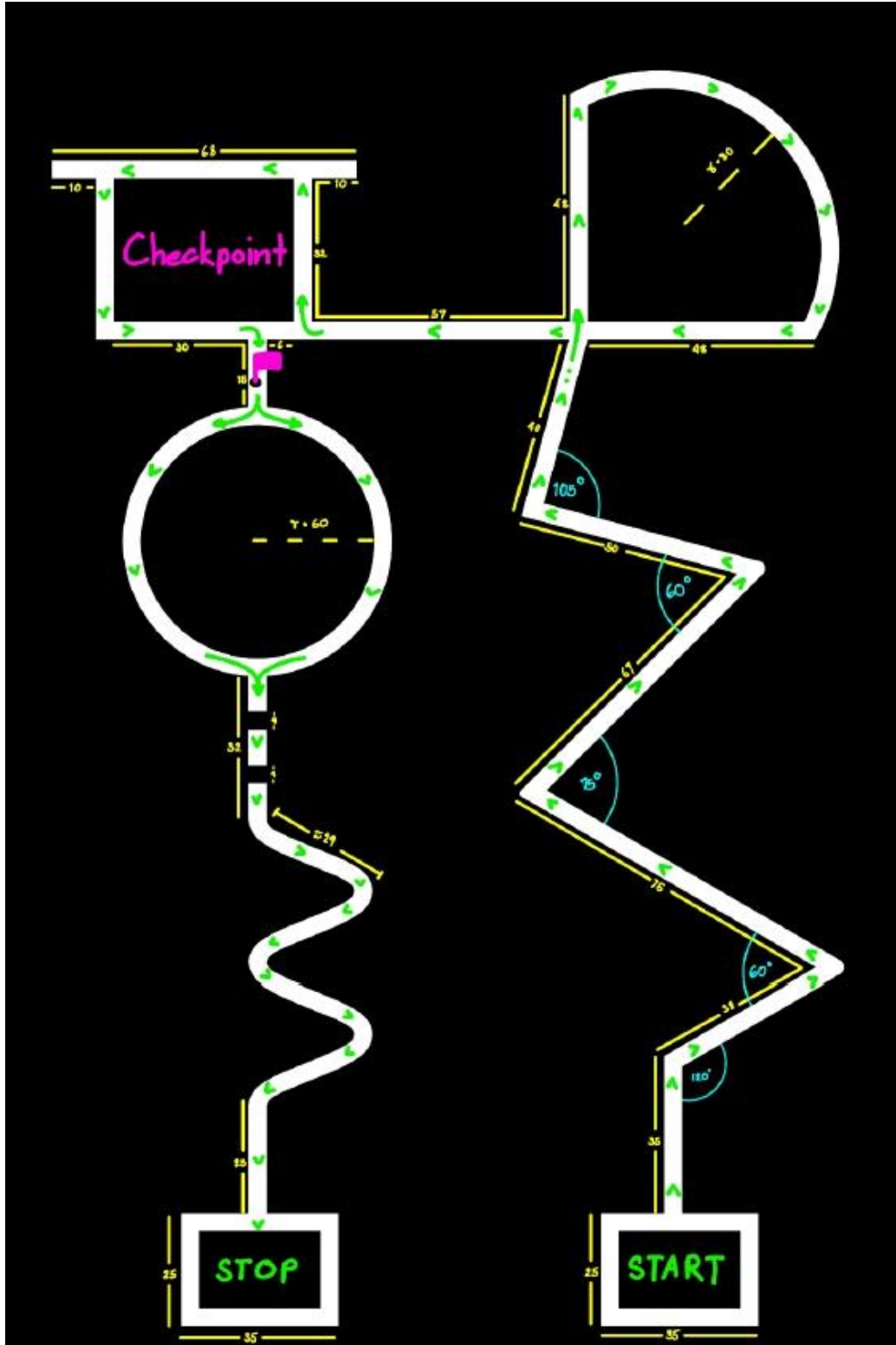


แนวคิด : ต้องการสร้างหุ่นยนต์เดินตามเส้น โดยใช้หลักการPID เข้ามาปรับใช้



62010711 - 62011019 - 62011044

โดยสนามที่ใช้ในการแข่งขันจะแบ่งเป็น2ช่วง คือตั้งแต่จุดStartจนถึง Checkpoint และ
Checkpoint จนถึงจุดหมาย



62010711 - 62011019 - 62011044

โครงสร้างของโค้ด

จะแบ่งเป็น3ส่วนคือ1.ส่วนฟังก์ชันการคำนวณค่าError โดยใช้หลักPIDแบ่งเป็น2ส่วนคือส่วนก่อนCheckpoint และหลังCheckpoint และเช็คค่าจากSensor ว่าอยู่Errorที่เท่าไร 2.ส่วนของฟังก์ชันการแสดงผลค่าError ออกทาง LED Dot Matrix 3.ฟังก์ชันการเคลื่อนที่ของรถที่แบ่งเป็น2ช่วง

โดยการตั้งค่าเริ่มต้นของรถจะแบ่งออกเป็น2แบบคือก่อนและหลังCheckpoint โดยมีค่าMeanV คือค่ากลางของSensorเพื่อเช็คค่าSensorอยู่เส้นขาวหรือเส้นดำ

```
int sensorsValue[5] ; // do not use 0 index
int meanV = 290;
int motorSpeed;
int baseSpeed = 90;
int rightSpeed, leftSpeed;
int maxSpeed = 200;
```

ส่วนก่อนCheckpoint จะมีค่าSpeedที่สูงกว่าตอนหลัง

```
int enA = 10;
int in1 = 9;
int in2 = 8;
// motor two
int enB = 5;
int in3 = 7;
int in4 = 6;
// PID
int Kp = 20;
```

เพราะต้องการให้ผ่านแยกและช่วงแรกจะมีโค้งหักศอกเยอะ

และจะมีค่าKPที่สูงกว่าหลังเพราะมีค่าSpeedที่สูงจึงต้องปรับให้เหมาะสม

ส่วนหลังCheckpoint จะมีค่าSpeedที่น้อย

เพราะช่วงหลังมีโค้งเยอะเพื่อให้รถเข้า

โค้งได้

```
int sensorsValue[5] ; // do not use 0 index
int meanV = 290;
int motorSpeed;
int baseSpeed = 80;
int rightSpeed, leftSpeed;
int maxSpeed = 200;
int enA = 10;
int in1 = 9;
int in2 = 8;
// motor two
int enB = 5;
int in3 = 7;
int in4 = 6;
// PID
int Kp = 15;
```

62010711 - 62011019 - 62011044

1. ส่วนฟังก์ชันการคำนวณค่าError โดยใช้หลักPIDแบ่งเป็น2ส่วนคือส่วนก่อนCheckpoint และหลังCheckpoint และเช็คค่าจากSensor ว่าอยู่Errorที่เท่าไร

ส่วนก่อนCheckpoint จะให้พจน์ขอการเลี้ยวสำคัญเพราะต้องเข้าโค้งหักศอก

```
void PID() {  
  
    if (W(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4]))  
    {  
        Serial.print("All White");  
        error=-5;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -5;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -4;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -3;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = -2;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = -1;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && W(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 0;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 1;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && B(sensorsValue[3]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 2;  
    } else if ( W(sensorsValue[0]) && W(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 3;  
    } else if ( W(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 4;  
    } else if ( W(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 5;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        Serial.print("All Black");  
        error=99;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -5;  
    } else if ( W(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 5;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 5;  
    }  
  
    motorSpeed = Kp * error  
    leftSpeed = baseSpeed + motorSpeed;  
    rightSpeed = baseSpeed - motorSpeed;  
  
    if (leftSpeed > maxSpeed) leftSpeed = maxSpeed;  
    if (rightSpeed > maxSpeed) rightSpeed = maxSpeed;  
  
    if (leftSpeed < -maxSpeed) leftSpeed = -maxSpeed;  
    if (rightSpeed < -maxSpeed) rightSpeed = -maxSpeed;  
}
```


62010711 - 62011019 - 62011044

ส่วนหลัง Checkpoint จะให้รถช้าลงเพื่อให้เข้าค้ำได้เรียบเนียนไม่หลุดเส้นโดยความเร็วจะปรับอัตโนมัติจากค่าP แล้วค่าError

```
void PID() {  
  
    if (W(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4])) {  
        Serial.print("All White");  
        error = -5;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -4;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -4;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -3;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = -2;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = -1;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && W(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 0;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 1;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 2;  
    } else if ( W(sensorsValue[0]) && W(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 3;  
    } else if ( W(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 4;  
    } else if ( W(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 4;  
    } else if ( B(sensorsValue[0]) && B(sensorsValue[1]) && B(sensorsValue[2]) && B(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        Serial.print("All Black");  
        error = 99;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && W(sensorsValue[4]) ) {  
        error = -4;  
    } else if ( W(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 4;  
    } else if ( B(sensorsValue[0]) && W(sensorsValue[1]) && W(sensorsValue[2]) && W(sensorsValue[3]) && B(sensorsValue[4]) ) {  
        error = 99;  
    }  
  
    motorSpeed = Kp * error;  
    leftSpeed = baseSpeed + motorSpeed;  
    rightSpeed = baseSpeed - motorSpeed;  
    if (leftSpeed > maxSpeed) leftSpeed = maxSpeed;  
    if (rightSpeed > maxSpeed) rightSpeed = maxSpeed;  
    if (leftSpeed < -maxSpeed) leftSpeed = -maxSpeed;  
    if (rightSpeed < -maxSpeed) rightSpeed = -maxSpeed;  
}
```

62010711 - 62011019 - 62011044

ส่วนของการนำค่ามาเช็คเพื่อเปรียบเทียบว่าอยู่Errorเท่าไร โดยนำค่าSensor มาเปรียบเทียบกับค่า Mean

```
bool W(int n) {  
    if (n >= meanV) { // is black  
        return true;  
    } else {  
        return false;  
    }  
}
```

```
bool B(int n) {  
    if (n < meanV) { // is white  
        return true;  
    } else {  
        return false;  
    }  
}
```

62010711 - 62011019 - 62011044

2. ส่วนของการแสดงผลบน Dot Matrix โดยการใช้การ Plot จุด และใช้ ฟังก์ชันในการแสดงผล

Error ออกมา

```
void printbyte(byte character[]) {  
    int i = 0;  
    for (i = 0; i < 8; i++)  
    {  
        lc.setRow(0, i, character[i] );  
    }  
}
```

```
byte zero[8] = {0x00, 0x0e, 0x11, 0x11, 0x11, 0x11, 0x11, 0x0e};  
byte one[8] = {0x00, 0x04, 0x0c, 0x14, 0x04, 0x04, 0x04, 0x1f};  
byte two[8] = {0x00, 0x00, 0x0e, 0x11, 0x02, 0x04, 0x08, 0x1f};  
byte three[8] = {0x00, 0x0c, 0x12, 0x02, 0x0e, 0x02, 0x12, 0x0c};  
byte four[8] = {0x00, 0x00, 0x02, 0x06, 0x0a, 0x1f, 0x02, 0x02};  
byte five[8] = {0x00, 0x1f, 0x10, 0x10, 0x1f, 0x01, 0x01, 0x1f};  
byte minone[8] = {0x00, 0x04, 0x0c, 0x14, 0xc4, 0x04, 0x04, 0x1f};  
byte mintwo[8] = {0x00, 0x00, 0x0e, 0x11, 0xc2, 0x04, 0x08, 0x1f};  
byte minthree[8] = {0x00, 0x0c, 0x12, 0x02, 0xce, 0x02, 0x12, 0x0c};  
byte minfour[8] = {0x00, 0x00, 0x02, 0x06, 0xca, 0x1f, 0x02, 0x02};  
byte minfive[8] = {0x00, 0x1f, 0x10, 0x10, 0xdf, 0x01, 0x01, 0x1f};
```

62010711 - 62011019 - 62011044

3.ฟังก์ชันการเคลื่อนที่ของรถที่แบ่งเป็น2ช่วง

ฟังก์ชันที่เหมือนกันทั้ง2ช่วงคือเดินหน้า ถอยหลัง หยุดรถ

```
void movestraight() {  
    analogWrite(enA, leftSpeed);  
    analogWrite(enB, rightSpeed);  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
}
```

การเดินหน้าของรถ ความเร็ว
ขึ้นกับการตั้งค่าเริ่มต้นและ
การคำนวณค่าError

การถอยหลังของรถ เมื่อเจอ
Sensorอยู่จุดดำทั้งหมดรถจะ
ถอยหลังมาตั้งหลัก

```
void backward() {  
    analogWrite(enA, 60);  
    analogWrite(enB, 60);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, HIGH);  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, HIGH);  
}
```

```
void stopcar() {  
    analogWrite(enA, 0);  
    analogWrite(enB, 0);  
    digitalWrite(in1, LOW);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, LOW);  
}
```

การหยุดรถก่อนการเลี้ยวเพื่อ
ความแม่นยำในการเลี้ยว

62010711 - 62011019 - 62011044

ฟังก์ชันที่ต่างกันคือการเลี้ยวซ้ายและเลี้ยวขวา

ส่วนก่อนCheckpoint การเลี้ยวจะเลี้ยวด้วยความเร็วที่เร็วกว่าช่วงหลังเพราะมีไค้งหักศอก

```
void turnleft() {  
    analogWrite(enA, 0);  
    analogWrite(enB, 150);  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
  
}  
void turnright() {  
    analogWrite(enA, 150);  
    analogWrite(enB, 0);  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
  
}
```

62010711 - 62011019 - 62011044

ส่วนหลังCheckpoint การเลี้ยวจะเลี้ยวด้วยความเร็วที่ช้าเพราะช่วงหลังต้องเลี้ยวตามโค้งให้
รถวิ่งช้าจะได้ไม่หลุดเส้น

```
void turnleft() {  
    analogWrite(enA, 0);  
    analogWrite(enB, 130);  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
  
}  
void turnright() {  
    analogWrite(enA, 130);  
    analogWrite(enB, 0);  
    digitalWrite(in1, HIGH);  
    digitalWrite(in2, LOW);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
  
}
```

62010711 - 62011019 - 62011044

โดยการทำงานจะเช็คค่าอยู่Errorที่เท่าไหร่ก็จะทำตามฟังก์ชันนั้นๆ โดยความเร็วขึ้นอยู่กับค่า Error และค่า Kp

```
if (error == 0)
{
    movestraight();
    printbyte(zero);
}
else if (error == 1)
{
    movestraight();
    printbyte(one);
}
else if (error == 2)
{
    movestraight();
    printbyte(two);
}
else if (error == 3)
{
    movestraight();
    printbyte(three);
}
else if (error == 4)
{
    movestraight();
    printbyte(four);
}
else if (error == 5)
{
    stopcar();
    delay(200);
    turnright();
    printbyte(five);
    delay(400);
}
else if (error == -1)
{
    movestraight();
    printbyte(minone);
}
else if (error == -2)
{
    movestraight();
    printbyte(mintwo);
}
else if (error == -3)
{
    movestraight();
    printbyte(minthree);
}
```

62010711 - 62011019 - 62011044

```
else if (error == -4)
{
    movestraight();
    printbyte(minfour);
}
else if (error == -5)
{
    stopcar();
    delay(200);
    turnleft();
    printbyte(minfive);
    delay(400);

}
else if (error == 99)
{
    backward();
    printbyte(zero);
}
```

