

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



# BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN: QUẢN TRỊ DỰ ÁN HỆ NHÚNG THEO  
CHUẨN KỸ NĂNG ITSS

ĐỀ TÀI:

## ROBOT DÒ LINE

Giảng viên hướng dẫn: TS. NGUYỄN ĐÌNH THUẬN

Lớp: AS – K62

Nhóm sinh viên thực hiện:

1. Lê Hoàng Anh Trung – 20176892
2. Mai Mạnh Thực – 20176885
3. Hoàng Minh Nguyệt – 20176839

*Hà Nội, tháng 06 năm 2021*

---

## MỤC LỤC

<b>PHÂN CÔNG CÔNG VIỆC.....</b>	<b>3</b>
<b>CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN.....</b>	<b>4</b>
1.1.    Mô tả yêu cầu bài toán .....	4
1.2.    Các vấn đề cần giải quyết.....	4
1.3.    Thiết bị và phần mềm yêu cầu.....	5
<b>CHƯƠNG 2. CHI PHÍ DỰ ÁN.....</b>	<b>6</b>
2.1.    Chi phí chung của dự án .....	6
<b>CHƯƠNG 3. CƠ SỞ LÝ THUYẾT.....</b>	<b>7</b>
3.1.    Cảm biến dò line .....	7
3.1.1.    Lý thuyết chung .....	7
3.1.2.    Ứng dụng trong bài toán robot dò line .....	7
3.2.    Lý thuyết PID.....	7
3.2.1.    Lý thuyết chung .....	7
3.2.2.    Lý thuyết điều khiển PID .....	8
3.2.3.    Ứng dụng trong bài toán dò line .....	10
<b>CHƯƠNG 4. PHÂN TÍCH THIẾT KẾ BÀI TOÁN.....</b>	<b>11</b>
4.1.    Thiết kế phần cứng .....	11
4.2.    Thiết kế phần mềm .....	12
<b>CHƯƠNG 5. XÂY DỰNG VÀ TRIỂN KHAI.....</b>	<b>12</b>
5.1.    Lắp ráp robot.....	12
5.2.    Xây dựng mã nguồn .....	13
5.3.    Quỹ đạo cho xe.....	15
<b>CHƯƠNG 6. KIỂM THỬ HỆ THỐNG.....</b>	<b>15</b>
6.1.    Kiểm thử hệ thống .....	15
<b>CHƯƠNG 7. KẾT LUẬN.....</b>	<b>16</b>
7.1.    Kiến thức thu được.....	16
7.2.    Hướng phát triển .....	16
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>17</b>

---

---

## PHÂN CÔNG CÔNG VIỆC

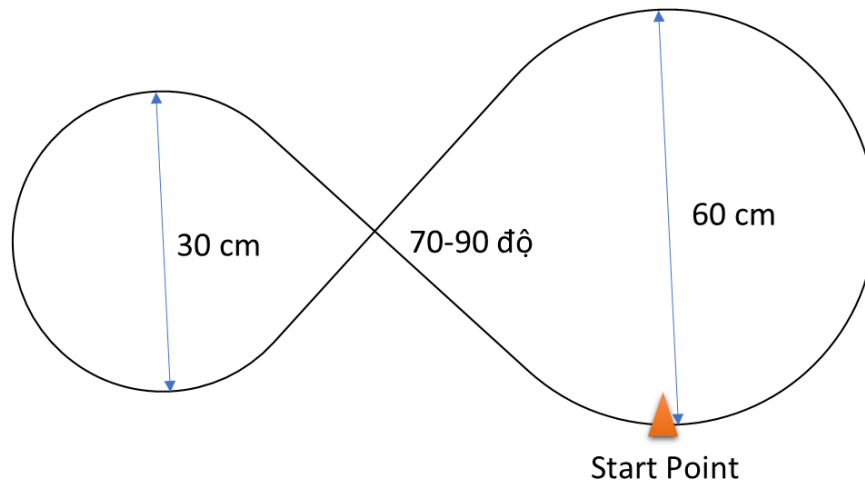
Thành viên	Phân công công việc	Đánh giá
Lê Hoàng Anh Trung MSSV: 20176892	[ Leader ] <ul style="list-style-type: none"> <li>- Phân công công việc các thành viên trong nhóm, theo dõi tiến độ làm việc</li> <li>- Quản lý mua sắm các thiết bị phần cứng</li> <li>- Tham gia lắp ráp robot</li> <li>- Tham gia xây dựng mã nguồn cho hệ thống</li> </ul>	Hoàn thành đúng tiến độ
Mai Mạnh Thục MSSV: 20176885	<ul style="list-style-type: none"> <li>- Tham gia lắp ráp robot</li> <li>- Tham gia xây dựng mã nguồn cho hệ thống</li> <li>- Sử dụng PID vào phần mềm</li> </ul>	Hoàn thành đúng tiến độ
Hoàng Minh Nguyệt MSSV: 20176839	<ul style="list-style-type: none"> <li>- Tham gia lắp ráp robot</li> <li>- Tham gia xây dựng mã nguồn cho hệ thống</li> <li>- Viết tài liệu cho dự án</li> </ul>	Hoàn thành đúng tiến độ

---

## CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN

### 1.1. Mô tả yêu cầu bài toán

Lập trình arduino robot dò line với quỹ đạo sử dụng băng dính điện hình số 8 và điểm bắt đầu như hình sau



### 1.2. Các vấn đề cần giải quyết

- Thiết kế phần cứng robot
- Lập trình phần mềm điều khiển robot đi theo đúng quỹ đạo yêu cầu

---

### 1.3. Thiết bị và phần mềm yêu cầu

Danh sách thiết bị phần cứng

Thiết bị	Số lượng
Khung xe robot 3 bánh	1
Module điều khiển động cơ L298	1
Module dò đường 5 cặp phát hồng ngoại K11A3	1
Kit Arduino Uno R3	1
Module Bluetooth	1
Dây nối mạch 2 đầu	1
Pin và khay pin	2 pin + 1 khay pin
Dụng cụ lắp ráp ( tua vít, ốc )	1 tua vít + 1 túi ốc
Băng dính quỹ đạo	1 cuộn

Danh sách phần mềm yêu cầu

- Môi trường phát triển tích hợp Arduino ( Arduino IDE )
- Git để quản lý mã nguồn

## CHƯƠNG 2. CHI PHÍ DỰ ÁN

### 2.1. Chi phí chung của dự án

Chi phí		Số tiền
Chi phí phần cứng	Khung xe robot 3 bánh	119.000
	Module điều khiển động cơ L298	37.000
	Module dò đường 5 cặp phát hồng ngoại K11A3	95.000
	Kit Arduino Uno R3	125.000
	Module Bluetooth	115.000
	Dây nối mạch 2 đầu	50.000
	Pin và khay pin	95.000
	Dụng cụ lắp ráp ( tua vít, ốc )	50.000
	Băng dính quỹ đạo	10.000
Chi phí phần mềm		0
Chi phí phát triển		0
Chi phí kiểm thử và bảo trì		0
Chi phí quản lý		0
Phí phát sinh		0

**Tổng chi phí: 696.000VND**

Người xác nhận: Leader Lê Hoàng Anh Trung

## CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

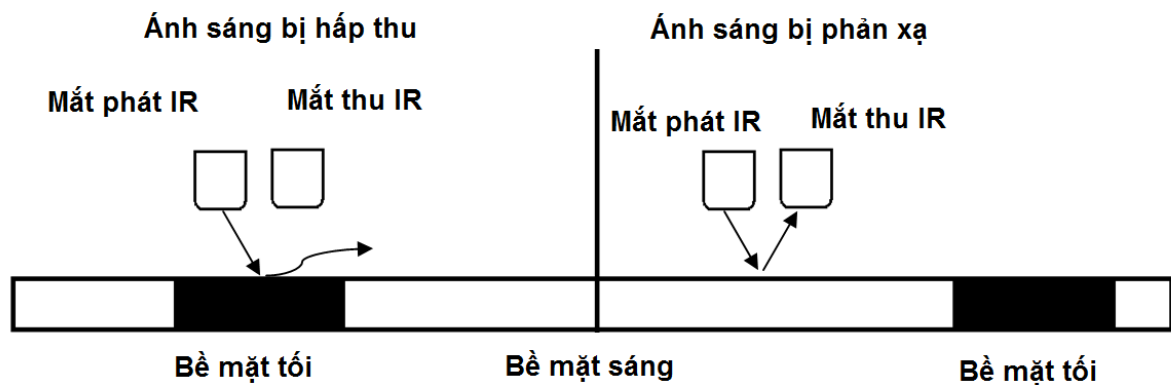
### 3.1. Cảm biến dò line

#### 3.1.1. Lý thuyết chung

Cảm biến dò line gồm có 2 mắt hồng ngoại (IR), một mắt phát và một mắt thu.

Cảm biến có thể sử dụng để phát hiện vật cản gần hoặc phát hiện màu hấp thụ hoặc phản xạ ánh sáng

Sơ đồ nguyên lý hoạt động của cảm biến



#### 3.1.2. Ứng dụng trong bài toán robot dò line

Thiết bị sử dụng trong bài: Module dò line 5 cặp phát hồng ngoại để phát hiện vùng màu đen và vùng màu sáng

Nguyên lý hoạt động:

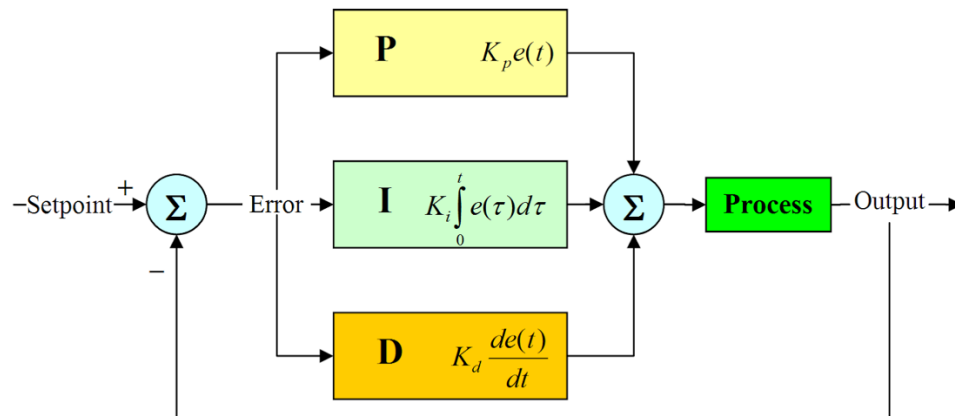
- Hoạt động dựa trên nguyên lý của tia hồng ngoại (bị hấp thụ khi chiếu vào bề mặt màu đen và phản xạ với các bề mặt màu sáng)
- 5 cảm biến hồng ngoại hướng xuống đất: cảm biến nào nằm trên vùng màu đen sẽ cho giá trị 0, ngược lại là 1

### 3.2. Lý thuyết PID

#### 3.2.1. Lý thuyết chung

**PID** ( *Proportional Integral Derivative Controller* ) là bộ điều khiển hồi tiếp vòng kín được sử dụng nhiều nhất trong doanh nghiệp. Bộ điều khiển PID sẽ tính toán giá trị "sai số" là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào.

PID là sự kết hợp của 3 bộ điều khiển: **tỉ lệ**, **tích phân** và **vi phân**, có khả năng điều chỉnh sai số thấp nhất có thể, tăng tốc độ đáp ứng, giảm độ vọt lố, hạn chế sự dao động.



Sơ đồ khối bộ điều khiển PID

- $K_P$  – hằng số điều khiển tỷ lệ
- $K_I$  – hằng số điều khiển tích phân
- $K_d$  – hằng số điều khiển vi phân

\* Giải thuật tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt

- Tỷ lệ (**Proportional**): xác định tác động của sai số hiện tại
- Tích phân (**Integral**): xác định tác động của tổng các sai số quá khứ
- Đạo hàm (**Derivative**): xác định tác động của tốc độ biến đổi sai số
- Tổng chập của ba tác động này dùng để điều chỉnh quá trình thông qua một phân tử điều khiển như vị trí của van điều khiển hay bộ nguồn của phân tử gia nhiệt. Nhờ vậy, những giá trị này có thể làm sáng tỏ về quan hệ thời gian:  $P$  phụ thuộc vào sai số hiện tại,  $I$  phụ thuộc vào tích lũy các sai số quá khứ, và  $D$  dự đoán các sai số tương lai, dựa vào tốc độ thay đổi hiện tại.

### 3.2.2. Lý thuyết điều khiển PID

Điều khiển PID - Điều khiển 3 khâu

- **Khâu tỉ lệ (Proportional)**

Khâu tỉ lệ được cho bởi:

$$P_{out} = K_p e(t)$$

trong đó

$P_{out}$ : thừa số tỉ lệ của đầu ra

$K_p$ : Hệ số tỉ lệ, thông số điều chỉnh

$e$ : sai số =  $SP - PV$

$t$ : thời gian hay thời gian tức thời (hiện tại)

\*\* Nhận xét:

- $K_p$  càng lớn thì tốc độ đáp ứng càng nhanh.
- $K_p$  càng lớn thì sai số xác lập càng nhỏ (nhưng không thể triệt tiêu).
- $K_p$  càng lớn thì các cực của hệ thống có xu hướng di chuyển ra xa trục thực  
=> Hệ thống càng dao động và độ vọt lố càng cao.
- Nếu  $K_p$  tăng quá giá trị giới hạn thì hệ thống sẽ dao động không tắt dần  
=> mất ổn định.



- **Khâu tích phân ( Integral )**

Thừa số tích phân được cho bởi:

$$I_{\text{out}} = K_i \int_0^t e(\tau) d\tau$$

trong đó

$I_{\text{out}}$ : thừa số tích phân của đầu ra

$K_i$ : độ lợi tích phân, 1 thông số điều chỉnh

$e$ : sai số =  $SP - PV$

$t$ : thời gian hoặc thời gian tức thời (hiện tại)

$\tau$ : một biến tích phân trung gian

**\*\* Nhân xét:**

- Tín hiệu ra được xác định bởi sai số.
- $K_i$  càng lớn thì đáp ứng quá độ càng chậm.
- $K_i$  càng lớn thì sai số xác lập càng nhỏ. đặc biệt hệ số khuếch đại của khâu tích phân bằng vô cùng khi tần số bằng 0 => triệt tiêu sai số xác lập với hàm nấc.
- $K_i$  càng lớn thì độ vọt lố càng cao.

- **Khâu vi phân ( Derivative )**

Thừa số vi phân được cho bởi:

$$D_{\text{out}} = K_d \frac{d}{dt} e(t)$$

trong đó

$D_{\text{out}}$ : thừa số vi phân của đầu ra

$K_d$ : Độ lợi vi phân, một thông số điều chỉnh

$e$ : Sai số =  $SP - PV$

$t$ : thời gian hoặc thời gian tức thời (hiện tại)

**\*\* Nhân xét:**

- $K_d$  càng lớn thì đáp ứng quá độ càng nhanh.
- $K_d$  càng lớn thì độ vọt lố càng nhỏ.
- Hệ số khuếch đại tại tần số cao là vô cùng lớn nên khâu hiệu chỉnh D rất nhạy với nhiễu tần số cao. Khâu vi phân không thể sử dụng một mình mà phải dùng kết hợp với các khâu P hoặc I.

### **Tổng hợp**

Khâu tỉ lệ, tích phân, vi phân được cộng lại với nhau để tính toán đầu ra của bộ điều khiển PID

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Trong đó:

**Độ lợi tỉ lệ Kp**

giá trị càng lớn thì đáp ứng càng nhanh do đó sai số càng lớn, bù khâu tỉ lệ càng lớn. Một giá trị độ lợi tỉ lệ quá lớn sẽ dẫn đến quá trình mất ổn định và dao động.

**Độ lợi tích phân Ki**

giá trị càng lớn kéo theo sai số ổn định bị khử càng nhanh. Đổi lại là độ vọt lố càng lớn: bất kỳ sai số âm nào được tích phân trong suốt đáp ứng quá độ phải được triệt tiêu tích phân bằng sai số dương trước khi tiến tới trạng thái ổn định.

**Độ lợi vi phân Kd**

giá trị càng lớn càng giảm độ vọt lố, nhưng lại làm chậm đáp ứng quá độ và có thể dẫn đến mất ổn định do khuếch đại nhiễu tín hiệu trong phép vi phân sai số.

\* Bằng cách điều chỉnh 3 hằng số trong giải thuật của bộ điều khiển PID, bộ điều khiển có thể dùng trong những thiết kế có yêu cầu đặc biệt. Đáp ứng của bộ điều khiển có thể được mô tả dưới dạng độ nhảy sai số của bộ điều khiển, giá trị mà bộ điều khiển vọt lố điểm đặt và giá trị dao động của hệ thống. Lưu ý là công dụng của giải thuật PID trong điều khiển không đảm bảo tính tối ưu hoặc ổn định cho hệ thống.

### 3.2.3. Ứng dụng trong bài toán dò line

Khi xe hoạt động, sensor thu thập thông tin qua bộ chuyển đổi ADC chuyển thành tín hiệu điện và đưa về vi điều khiển. Giá trị nhận về sẽ được so sánh với 1 giá trị đặt trước (ngưỡng). Nếu giá trị  $ADC[i] < \text{giá trị ngưỡng}$  thì LED thứ  $i$  đang ở phần sâu, ngược lại LED thứ  $i$  đang ở phần vạch.

Mạch cảm biến dùng 10 cặp led thu phát tương ứng với 10 giá trị ADC gửi về vi điều khiển. Dựa trên 10 giá trị ADC thu về, xác định độ lệch tương đối giữa quỹ đạo của robot và quỹ đạo mong muốn, sau đó so sánh độ lệch đó thành các mức. Khi đó chúng ta sẽ áp dụng PID để tính toán để đưa ra giá trị điều khiển PWM đến mạch cầu H.

Các giá trị khi lệch line:

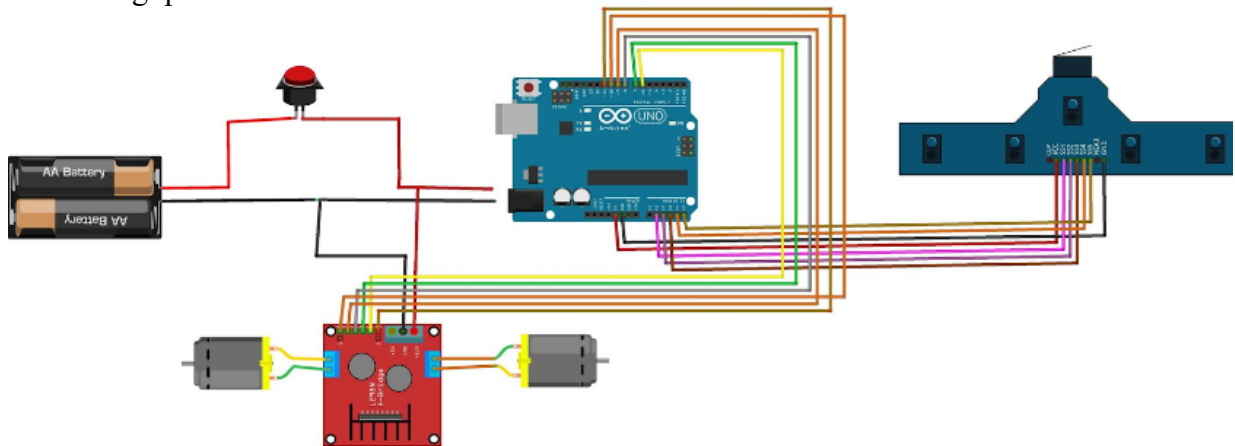
Giá trị sensor	Giá trị tương ứng	Giá trị sensor	Giá trị tương ứng
0 0 0 0 1	4	0 1 1 0 0	-1
0 0 0 1 1	3	0 1 0 0 0	-2
0 0 0 1 0	2	1 1 0 0 0	-3
0 0 1 1 0	1	1 0 0 0 0	-4
0 0 1 0 0	0	0 0 0 0 0	-5

Dựa trên các mức lệch, điều chỉnh tốc độ 2 bánh trái – phải để đưa robot về quỹ đạo. Cụ thể, để rẽ trái tốc độ bánh xe bên phải robot nhanh hơn bên trái một giá trị tương ứng với các mức lệch (cần phải thử nghiệm nhiều lần), và ngược lại. Phương pháp này đơn giản, nhưng robot chạy không ổn định, lúc rất nhanh, lúc lại chậm, tính ổn định phụ thuộc nhiều vào động cơ và kết cấu cơ khí của robot. Để khắc phục nhược điểm này, bằng cách áp dụng bộ điều khiển PID điều khiển vị trí robot.

## CHƯƠNG 4. PHÂN TÍCH THIẾT KẾ BÀI TOÁN

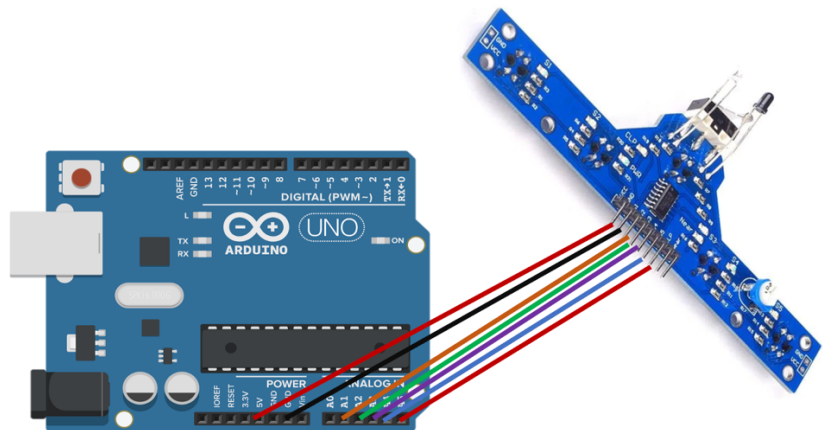
### 4.1. Thiết kế phần cứng

Sơ đồ tổng quát



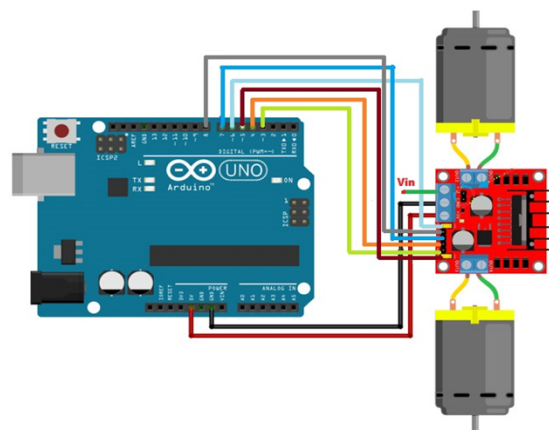
Sơ đồ chi tiết Arduino – Cảm biến dò line

Cảm biến dò đường	Mạch arduino
SS1	A0
SS2	A1
SS3	A2
SS4	A3
SS5	A4
VCC	5V
GND	GND



Sơ đồ chi tiết Động cơ – Arduino

Động cơ	Mạch khuếch đại
Động cơ trái	OUT1
	OUT2
Động cơ phải	OUT3
	OUT4
Mạch khuếch đại	Mạch Arduino
ENA	6
ENB	5
IN1	8
IN2	9
IN3	19
IN4	11



---

## 4.2. Thiết kế phần mềm

Thiết kế mã nguồn gồm những hàm sau

1. Định nghĩa các chân điều khiển động cơ
2. Điều khiển xe cho chạy đúng line
  - Định nghĩa các giá trị lệch line
  - Tính giá trị PID
  - Điều khiển động cơ
3. Module lắp

# CHƯƠNG 5. XÂY DỰNG VÀ TRIỂN KHAI

## 5.1. Lắp ráp robot

Lắp ráp robot dựa trên sơ đồ như mô tả trên

## 5.2. Xây dựng mã nguồn

Định nghĩa các chân điều khiển động cơ

```

/// define sensor pinout
#define line_1      A0 // trái (hoặc ngược lại)
#define line_2      A1
#define line_3      A2 // giữa
#define line_4      A3
#define line_5      A4 // phải
const int InA = 8; // Động cơ trái
const int InB = 9;
const int InC = 10; // Động cơ phải
const int InD = 11;

void setup()
{
    pinMode(InA, OUTPUT);
    pinMode(InB, OUTPUT);
    pinMode(InC, OUTPUT);
    pinMode(InD, OUTPUT);
    pinMode(6,1); // chân bấm xung cho động cơ EnA
    pinMode(5,1); // chân bấm xung cho động cơ EnB
    pinMode(line_1,INPUT);
    pinMode(line_2,INPUT);
    pinMode(line_3,INPUT);
    pinMode(line_4,INPUT);
    pinMode(line_5,INPUT);
    digitalWrite(InA, LOW); // Bánh trái đi lên
    digitalWrite(InB, HIGH);
    digitalWrite(InC, LOW); // Bánh phải đi lên
    digitalWrite(InD, HIGH);
    Input = 0; // mặc định line giữa =0, luôn luôn điều chỉnh xe để input=0

    myPID.SetSampleTime(1); // thời gian lấy mẫu phụ thuộc tốc độ xe, lấy mẫu càng nhanh càng tốt (ms)
    myPID.SetMode(AUTOMATIC);
    myPID.SetOutputLimits(-speed_robot,speed_robot); // giá trị tốc độ, -speed tức bánh bên trái quay max, bên phải ngừng quay
}

```

Điều khiển xe cho chạy đúng line

- Định nghĩa các giá trị lệch line

---

```
//Đèn tắt là 0v, đèn sáng (vào line đen) là 5V, từ 5 cảm biến ta có 9 vị trí 4 3 2 1 0 -1 -2 -3 -4
void scan_sensor()
{
    if(digitalRead(line_5)==0) sensor=-4;
    else if((digitalRead(line_4)==0)&&(digitalRead(line_5)==0)) sensor=-3;
    else if(digitalRead(line_4)==0) sensor=-2;
    else if((digitalRead(line_3)==0)&&(digitalRead(line_4)==0)) sensor=-1;
    else if(digitalRead(line_3)==0) sensor=0; // vào line giữa
    else if((digitalRead(line_2)==0)&&(digitalRead(line_3)==0)) sensor=1;
    else if(digitalRead(line_2)==0) sensor=2;
    else if((digitalRead(line_1)==0)&&(digitalRead(line_2)==0)) sensor=3;
    else if(digitalRead(line_1)==0) sensor=4;
}

```

- Sử dụng PID vào bài toán: Sử dụng thư viện PID của Arduino IDE, điều chỉnh thông số Kp, Ki, Kd cho phù hợp với thiết bị

```
//double Kp=15, Ki=0.045, Kd=12;
double Kp=18.5, Ki=0.04, Kd=19.5;
```

```
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
```

- Điều khiển động cơ

```
// bấm xung điều khiển tốc độ 2 bánh xe, từ đó ta điều khiển được hướng rẽ, tốc độ..... thông qua 1 biến duy nhất
void motorControl(int16_t duty_value)
{
    int16_t speed_a, speed_b;
    int speed_zero;
    speed_zero = speed_robot/2;
    if(duty_value>1)
    {
        speed_b=-speed_zero;
        speed_a=duty_value;
    }
    else if(duty_value==0)
    {
        speed_a=speed_b=0;
    }
    else if(duty_value<-1)
    {
        speed_a=-speed_zero;
        speed_b=-duty_value;
    }
    analogWrite(5,speed_b+speed_zero); //trai EnB
    analogWrite(6,speed_a+speed_zero); //phai EnA
}

```

Module lắp

---

```

void loop() {
    Setpoint=0;
    scan_sensor(); // đọc dữ liệu từ cảm biến
    Input = sensor;
    myPID.Compute(); // tính toán
    motorControl(Output); // điều khiển bánh xe để robot đi đúng đường

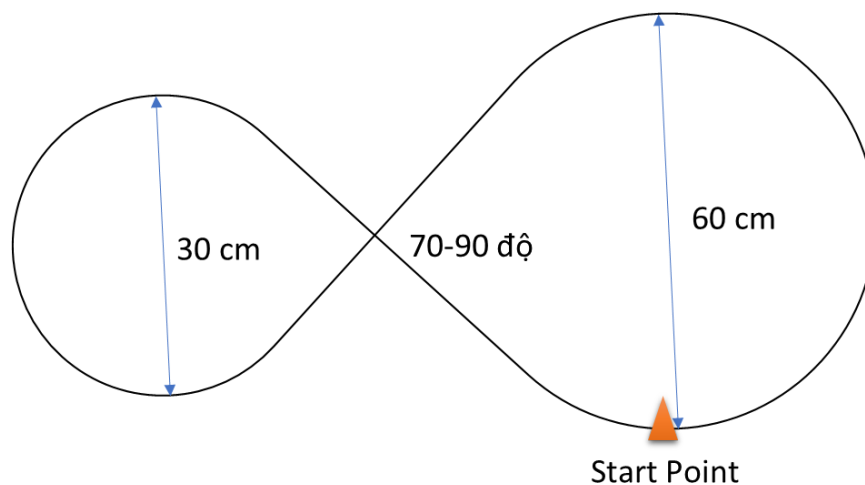
}

```

---

### 5.3. Quỹ đạo cho xe

Dán quỹ đạo bằng băng dính đen lên mặt sàn phẳng theo hình số 8 như yêu cầu



## CHƯƠNG 6. KIỂM THỬ HỆ THỐNG

### 6.1. Kiểm thử hệ thống

Input	Output	Kết quả
Bật nguồn và để xe tại vị trí xuất phát	Xe chạy bám line, đúng theo quỹ đạo số 8	OK

---

## CHƯƠNG 7. KẾT LUẬN

### 7.1. Kiến thức thu được

- Nâng cao kỹ năng làm việc nhóm
- Hiểu được quy trình quản lý và phát triển một dự án hệ thống nhúng từ công đoạn lên kế hoạch, thiết kế sản phẩm, xây dựng và kiểm thử hệ thống.
- Hiểu được cách sử dụng những module Arduino, L298N, cảm biến dò line 5 mắt
- Hiểu được thuật toán PID và sử dụng vào bài toán thực tế ( cụ thể là bài toán xe dò line )

### 7.2. Hướng phát triển

Dự án xe dò line nhóm chúng em thiết kế và xây dựng vẫn còn đơn giản và còn một vài thiếu sót. Trong tương lai chúng em dự định sẽ phát triển sản phẩm hoạt động chính xác hơn và thêm một vài tính năng như:

- Điều khiển từ xa thông qua kết nối bluetooth
- Xe tự động tránh vật cản



---

## TÀI LIỆU THAM KHẢO

- [ 1 ]. Slide bài giảng môn học về quản lý và phát triển dự án nhúng TS. NGUYỄN ĐÌNH THUẬN
- [ 2 ]. Lý thuyết điều khiển PID  
[https://vi.wikipedia.org/wiki/Bộ\\_điều\\_khiển\\_PID](https://vi.wikipedia.org/wiki/Bộ_điều_khiển_PID)
- [ 3 ]. Thư viện PID của Arduino  
<https://playground.arduino.cc/Code/PIDLibrary/>  
<https://github.com/br3ttb/Arduino-PID-Library>