

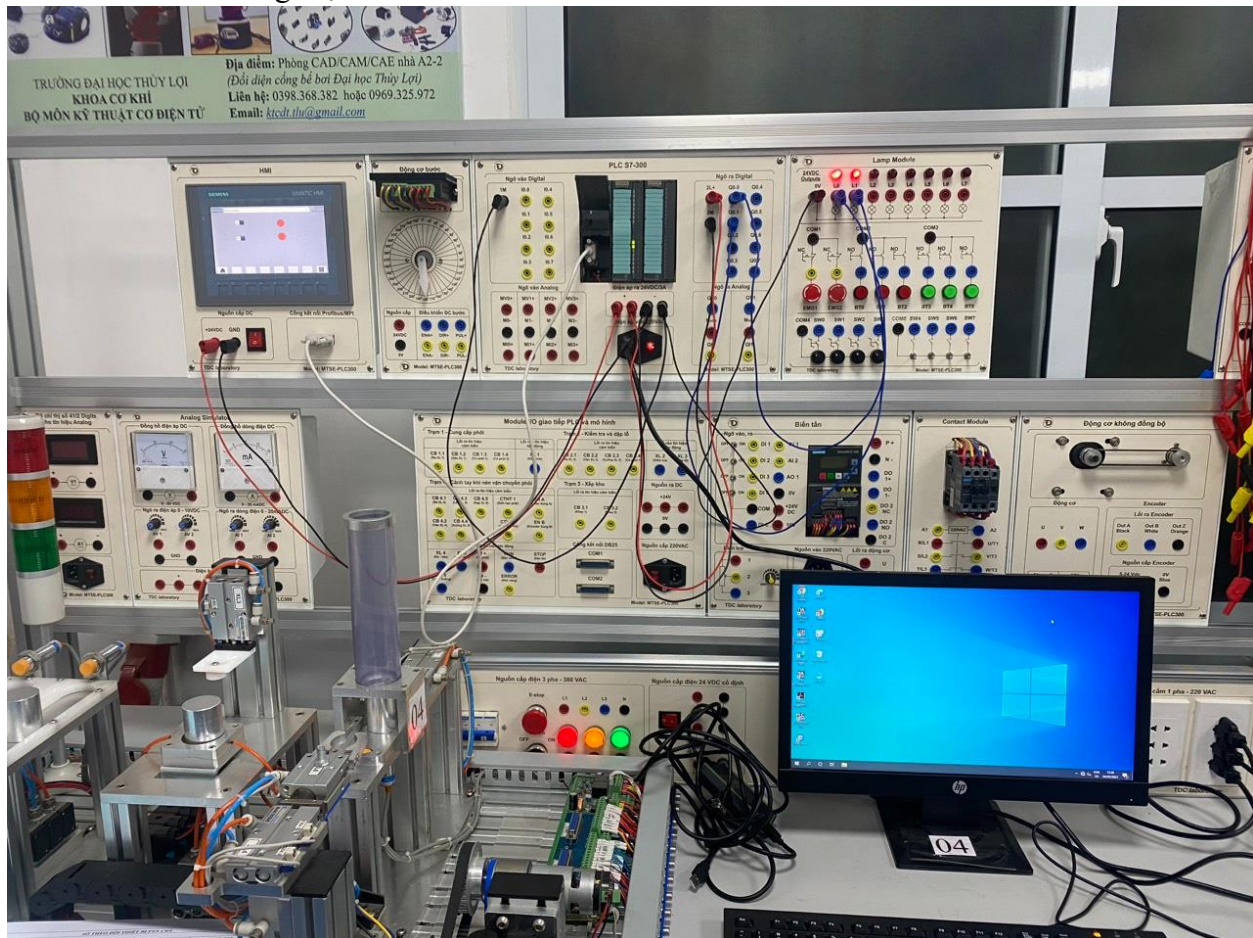
Ngày 4 tháng 5 năm 2023

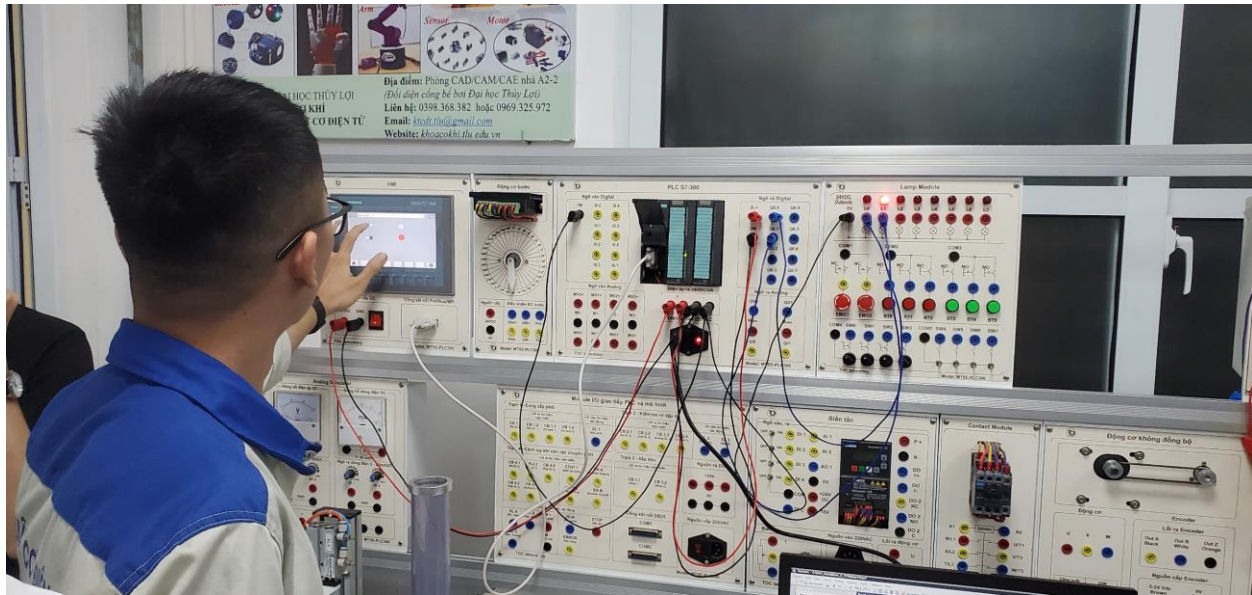
Buổi sáng

1. Hình ảnh công việc
2. Nội dung thực hiện ROBOT
 - Giới thiệu phần mềm KUKA.Sim Pro
 - Chức năng phần mềm
 - Mô phỏng 3D
 - Lập trình cho robot KUKA

Buổi chiều

1. Hình ảnh công việc

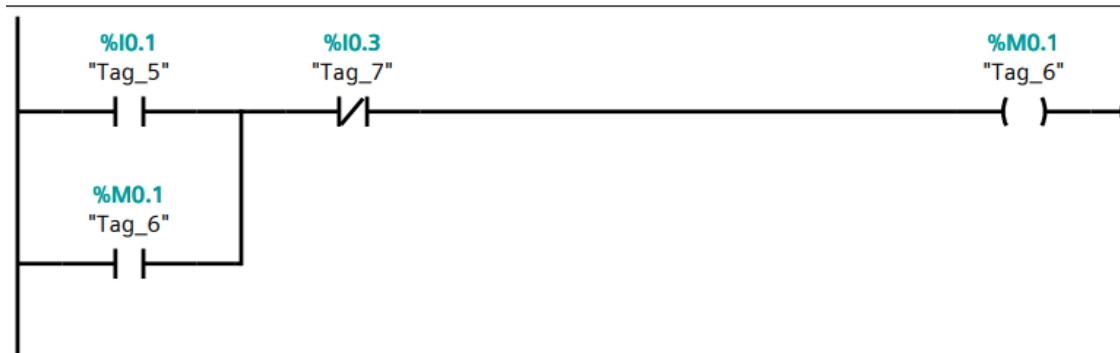
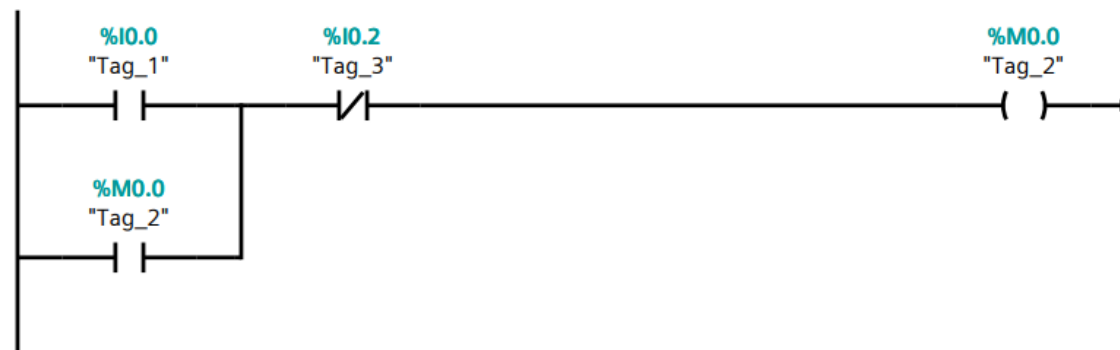




2. Nội dung thực hiện PLC

Bật tắt 2 đèn LED bằng màn hình HMI

3. Chương trình điều khiển

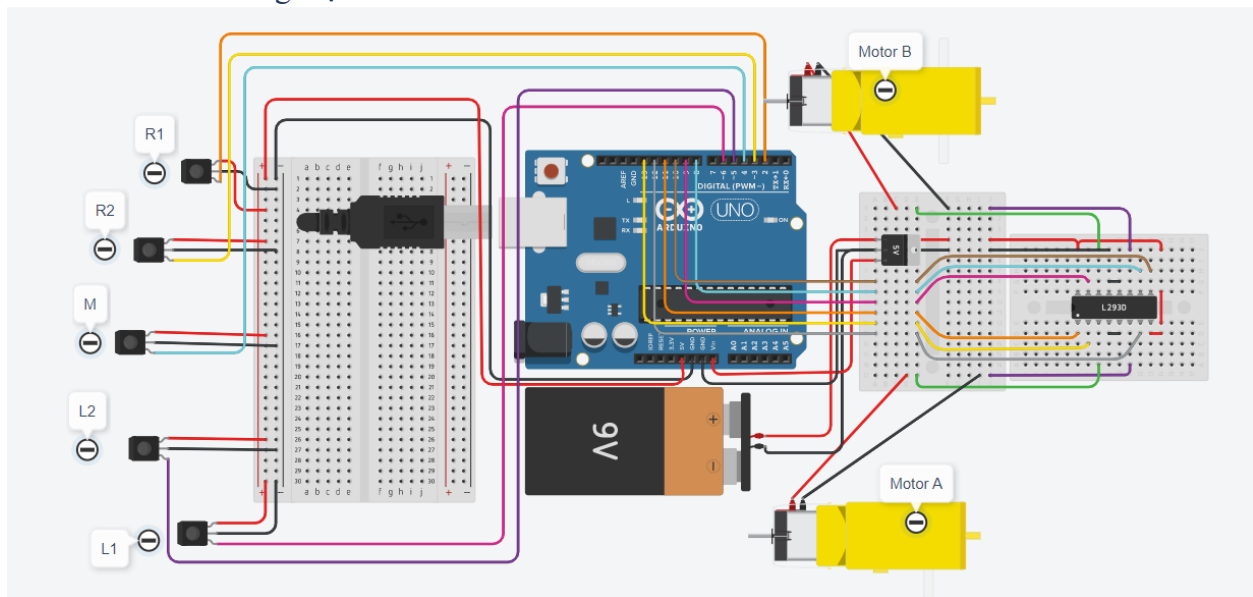




Ngày 5 tháng 5 năm 2023

Buổi sáng

1. Hình ảnh công việc



2. Nội dung thực hiện

Hãy tìm hiểu một số khái niệm cơ bản trước khi giải thích từng phần của code:

pinMode(pin, mode): Hàm này dùng để cấu hình chế độ của chân pin trên Arduino. pin là số chân pin và mode có thể là **INPUT** hoặc **OUTPUT**.

digitalRead(pin): Hàm này dùng để đọc giá trị (**HIGH** hoặc **LOW**) từ chân pin đã được cấu hình là **INPUT**.

digitalWrite(pin, value): Hàm này dùng để ghi giá trị (**HIGH** hoặc **LOW**) lên chân pin đã được cấu hình là **OUTPUT**.

analogWrite(pin, value): Hàm này dùng để ghi giá trị analog (PWM) lên chân pin đã được cấu hình là **OUTPUT**.

Các biến **MA1, MA2, MB3, MB4, E12, E34, L1, L2, M, R2, R1** là các số nguyên (int) và được gán giá trị tương ứng với số chân pin trên Arduino.

Trong hàm setup(), chúng ta cấu hình chế độ **OUTPUT** cho các chân pin **MA1, MA2, MB3, MB4, E12, E34** và chế độ **INPUT** cho các chân pin **L1, L2, M, R2, R1**.

Nếu điều kiện thỏa mãn, tức là giá trị đọc từ các chân pin **L1, L2, M, R2, R1** tương ứng là 1 (**HIGH**), thì chúng ta gọi các hàm tương ứng để thực hiện các hành động:

Nếu các chân pin đọc có giá trị là (1, 1, 0, 1, 1), chúng ta gọi hàm **forward()** để di chuyển về phía trước.

Nếu các chân pin đọc có giá trị là (1, 1, 1, 1, 1), chúng ta gọi hàm **backward()** để di chuyển lùi lại.

Nếu các chân pin đọc có giá trị là (1, 0, 0, 1, 1), chúng ta gọi hàm **left_s()** để quẹo trái một góc nhỏ.

Nếu các chân pin đọc có giá trị là (0, 0, 0, 1, 1), chúng ta gọi hàm **left_f()** để quẹo trái một góc lớn.

Nếu các chân pin đọc có giá trị là (1, 0, 1, 1, 1), chúng ta gọi hàm **left()** để quẹo trái.

Nếu các chân pin đọc có giá trị là (1, 1, 0, 0, 1), chúng ta gọi hàm **right_s()** để quẹo phải một góc nhỏ.

Nếu các chân pin đọc có giá trị là (1, 1, 0, 0, 0), chúng ta gọi hàm **right_f()** để quẹo phải một góc lớn.

Nếu các chân pin đọc có giá trị là (1, 1, 1, 0, 1), chúng ta gọi hàm **right()** để quẹo phải.

Nếu các chân pin đọc có giá trị là (0, 0, 0, 0, 0), chúng ta gọi hàm **stop()** để dừng lại.

forward(): Hàm này được gọi để di chuyển robot đi về phía trước. Các thao tác thực hiện trong hàm bao gồm:

Đặt chân pin **MA1 (Motor A +)** và **MB3 (Motor B -)** lên mức cao (**HIGH**) để quay motor theo chiều thuận.

Đặt chân pin **MA2 (Motor A -)** và **MB4 (Motor B +)** xuống mức thấp (**LOW**).

Sử dụng hàm **analogWrite()** để cung cấp điện áp PWM (Pulse Width Modulation) cho chân pin **E12** và **E34** để kích hoạt motor với mức công suất là 80 (giá trị từ 0-255).

backward(): Hàm này được gọi để di chuyển robot lùi lại. Các thao tác thực hiện trong hàm bao gồm:
Đặt chân pin **MA2 (Motor A -)** và **MB3 (Motor B -)** lên mức cao (**HIGH**) để quay motor theo chiều nghịch.

Đặt chân pin **MA1 (Motor A +)** và **MB4 (Motor B +)** xuống mức thấp (**LOW**).

Sử dụng hàm **analogWrite()** để cung cấp điện áp PWM cho chân pin **E12** và **E34** với mức công suất là 60.

left_s(): Hàm này được gọi để robot quẹo trái một góc nhỏ. Các thao tác thực hiện trong hàm bao gồm:
Đặt chân pin **MA1 (Motor A +)** và **MB3 (Motor B -)** lên mức cao (**HIGH**) để quay motor theo chiều thuận.

Đặt chân pin **MA2 (Motor A -)** và **MB4 (Motor B +)** xuống mức thấp (**LOW**).

Sử dụng hàm **analogWrite()** để cung cấp điện áp PWM cho chân pin **E12** và **E34** với mức công suất là 70.

left_f(): Hàm này được gọi để robot quẹo trái một góc lớn. Các thao tác thực hiện trong hàm tương tự như **left_s()** trên, chỉ khác là mức công suất cung cấp cho chân pin **E12** và **E34** là 60.

Buổi chiều

1. Nội dung công việc

```
int
MA1 = 13,
MA2 = 12,
MB3 = 8,
MB4 = 9,
E12 = 11,
E34 = 10,
L1 = 6,
L2 = 5,
M = 4,
R2 = 3,
R1 = 2;
void setup()
{
  pinMode(MA1, OUTPUT); // Motor A +
  pinMode(MA2, OUTPUT); // Motor A-
  pinMode(MB3, OUTPUT); // Motor B -
  pinMode(MB4, OUTPUT); // Motor B +
  pinMode(E12, OUTPUT); // Enable 1&2
  pinMode(E34, OUTPUT); // Enable 3&4
  pinMode(L1, INPUT); // L1
  pinMode(L2, INPUT); // L2
  pinMode(M, INPUT); // M
```

```

    pinMode(R2, INPUT); // R2
    pinMode(R1, INPUT); // R1
}

void loop()
{
    if((digitalRead(L1)== 1) && (digitalRead(L2) == 1) && (digitalRead(M) == 0) && (digitalRead(R2)== 1)
    && (digitalRead(R1) == 1));
    {forward();} // calling functions
    if((digitalRead(L1)== 1) && (digitalRead(L2) == 1) && (digitalRead(M) == 1) && (digitalRead(R2)== 1)
    && (digitalRead(R1) == 1));
    {backward();}
    if((digitalRead(L1)== 1) && (digitalRead(L2) == 0) && (digitalRead(M) == 0) && (digitalRead(R2)== 1)
    && (digitalRead(R1) == 1));
    {left_s();}
    if((digitalRead(L1)== 0) && (digitalRead(L2) == 0) && (digitalRead(M) == 0) && (digitalRead(R2)== 1)
    && (digitalRead(R1) == 1));
    {left_f();}
    if((digitalRead(L1)== 1) && (digitalRead(L2) == 0) && (digitalRead(M) == 1) && (digitalRead(R2)== 1)
    && (digitalRead(R1) == 1));
    {left();}
    if((digitalRead(L1)== 1) && (digitalRead(L2) == 1) && (digitalRead(M) == 0) && (digitalRead(R2)== 0)
    && (digitalRead(R1) == 1));
    {right_s();}
    if((digitalRead(L1)== 1) && (digitalRead(L2) == 1) && (digitalRead(M) == 0) && (digitalRead(R2)== 0)
    && (digitalRead(R1) == 0));
    {right_f();}
    if((digitalRead(L1)== 1) && (digitalRead(L2) == 1) && (digitalRead(M) == 1) && (digitalRead(R2)== 0)
    && (digitalRead(R1) == 1));
    {right();}
    if((digitalRead(L1)== 0) && (digitalRead(L2) == 0) && (digitalRead(M) == 0) && (digitalRead(R2)== 0)
    && (digitalRead(R1) == 0));
    {stop();}
}

void forward()
{
    digitalWrite(MA1, HIGH);
    digitalWrite(MA2, LOW);
    digitalWrite(MB4, LOW);
    digitalWrite(MB3, HIGH);
    analogWrite(E12, 80);
    analogWrite(E34, 80);

}

void backward()
{
    digitalWrite(MA1, LOW);
    digitalWrite(MA2, HIGH);
    digitalWrite(MB4, LOW);

```



```
    digitalWrite(MB3, HIGH);
    analogWrite(E12, 60);
    analogWrite(E34, 60);
}
void left_s()
{
    digitalWrite(MA1, HIGH);
    digitalWrite(MA2, LOW);
    digitalWrite(MB4, LOW);
    digitalWrite(MB3, HIGH);
    analogWrite(E12, 70);
    analogWrite(E34, 60);
}
void left_f()
{
    digitalWrite(MA1, LOW);
    digitalWrite(MA2, HIGH);
    digitalWrite(MB4, LOW);
    digitalWrite(MB3, HIGH);
    analogWrite(E12, 60);
    analogWrite(E34, 60);
}
void left()
{
    digitalWrite(MA1, HIGH);
    digitalWrite(MA2, LOW);
    digitalWrite(MB4, LOW);
    digitalWrite(MB3, HIGH);
    analogWrite(E12, 70);
    analogWrite(E34, 50);
}
void right_s()
{
    digitalWrite(MA1, HIGH);
    digitalWrite(MA2, LOW);
    digitalWrite(MB4, LOW);
    digitalWrite(MB3, HIGH);
    analogWrite(E12, 60);
    analogWrite(E34, 70);
}
void right_f()
{
    digitalWrite(MA1, HIGH);
    digitalWrite(MA2, LOW);
    digitalWrite(MB4, HIGH);
    digitalWrite(MB3, LOW);
    analogWrite(E12, 60);
    analogWrite(E34, 60);
}
```

```
void right()
{
    digitalWrite(MA1, HIGH);
    digitalWrite(MA2, LOW);
    digitalWrite(MB4, LOW);
    digitalWrite(MB3, HIGH);
    analogWrite(E12, 50);
    analogWrite(E34, 70);
}
void stop()
{
    digitalWrite(MA1, LOW);
    digitalWrite(MA2, LOW);
    digitalWrite(MB4, LOW);
    digitalWrite(MB3, LOW);
    analogWrite(E12, 0);
    analogWrite(E34, 0);
}
```