

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO MÔN
THỰC HÀNH NHẬP MÔN ĐIỆN TOÁN

BÁO CÁO

IoT: ARDUINO LAB 4

GV hướng dẫn: Thầy HUỖNH TẤN LỘC

Sinh viên thực hiện:

<i>Họ và tên</i>	<i>MSSV</i>	<i>Mã lớp</i>
Lê Trọng Tín	2313452	L01
Trương Văn Thảo Nhi	2312514	L01
Lê Phương Vũ	2313954	L01

TP Hồ Chí Minh, 2023



Mục lục

1 Kiến thức và kết quả chuẩn bị	2
1.1 Tổng quan về DC motor	2
1.2 Tổng quan về driver L298N	2
2 Các bài toán áp dụng trên robot	3
2.1 Kiểm tra hoạt động của các left motor và right motor	3
2.2 Phân khai báo được dùng trong Arduino	4
2.3 Cho xe chạy thẳng về phía trước và lùi về phía sau	5
2.4 Cho xe rẽ phải và rẽ trái	5
2.5 Phần hàm thực thi của chương trình	6
3 Kết luận	6
Tài liệu	7

Bài báo cáo này trình bày kết quả thực hành trên mẫu robot ứng dụng Arduino. Bằng cách kết hợp các cú pháp và thư viện sẵn có của Arduino mà ta có thể thiết lập để cho xe chạy tới, chạy lùi hoặc rẽ sang trái, sang phải.

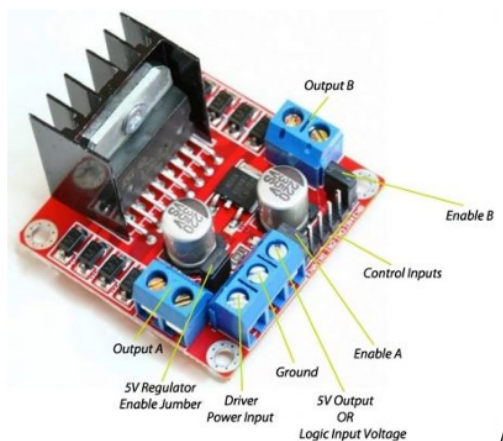
Trong phần đầu của bài báo cáo, chúng tôi xin được trình bày một số khái niệm cơ bản về DC motor, hay động cơ điện một chiều.

Đối với mỗi bài tập, đầu tiên chúng tôi sẽ phân tích đề bài một cách sơ lược, sau đó giải thích nguyên lý hoạt động và cuối cùng là trình bày mã nguồn cũng như kết quả hiện thực bằng hình ảnh.

1 Kiến thức và kết quả chuẩn bị

1.1 Tổng quan về DC motor

Mạch H-Bridge là một mạch điện tử có khả năng điều khiển động cơ điều hướng cả hai chiều. Nó thường được sử dụng trong nhiều ứng dụng khác nhau, phổ biến nhất là trong việc điều khiển động cơ của robot. Tên gọi "H-Bridge" xuất phát từ cách kết nối bốn transistor trong mạch, khiến cho sơ đồ mạch có dạng giống chữ "H".



Hình 1: Hình ảnh của driver L298N trên thị trường

1.2 Tổng quan về driver L298N

Serial Monitor là một công cụ quan trọng khi tạo dự án với Arduino. Nó có thể được dùng như một công cụ để gỡ lỗi, thử nghiệm các khái niệm hoặc giao tiếp trực tiếp với bo mạch chính của Arduino.

L298N Motor Driver là một bộ điều khiển sử dụng H-Bridge để dễ dàng kiểm soát hướng và tốc độ của tối đa 2 động cơ DC. Hướng dẫn này sẽ chỉ

cho bạn cách kiểm soát một động cơ sử dụng L298N.

Ta sử dụng Arduino để điều khiển hướng(tiến hoặc lùi) và tốc độ của động cơ.

2 Các bài toán áp dụng trên robot

2.1 Kiểm tra hoạt động của các left motor và right motor

Trước hết để có thể hiện thực các bài toán, ta cần phải đảm bảo rằng các động cơ bên trái và phải hoạt động đúng chức năng mà ta mong muốn như chạy tới, chạy lùi cho từng động cơ riêng lẻ, sau đó ta kết hợp lại với nhau để cả hệ thống hoạt động

- Kiểm tra động cơ bên trái (left motor)

Ta có phần mã nguồn như sau:

```
1 void setup(){
2     pinMode(9, OUTPUT);
3     pinMode(10, OUTPUT);
4
5 }
6 void loop(){
7     left_speed (180);
8 }
9 void left_speed (int speed){
10     if(speed > 0){
11         digitalWrite(10,LOW);
12         analogWrite(9,250);
13     }
14     else{
15         speed = 0 - speed;
16         digitalWrite(10,LOW);
17         analogWrite(9,0);
18     }
19 }
```

Hình 2: Mã nguồn kiểm tra hoạt động của động cơ trái

- Kiểm tra động cơ bên phải (right motor)
Ta có phần mã nguồn như sau:

```
1. void setup(){
2.   pinMode(12, OUTPUT);
3.   pinMode(11, OUTPUT);
4. }
5. void loop(){
6.   right_speed(250);
7. }
8. void right_speed (int speed){
9.   if(speed > 0){
10.    digitalWrite(12,LOW);
11.    analogWrite(11,speed);
12.  }
13.  else{
14.    speed = 0 - speed;
15.    digitalWrite(11,LOW);
16.    analogWrite(12,speed);
17.  }
18. }
```

Hình 3: Mã nguồn kiểm tra hoạt động của động cơ phải

2.2 Phần khai báo được dùng trong Arduino

Hàm void setup() : Dùng để khai báo các chân mắc được dùng trên xe mô phỏng với pin 9,10 là của động cơ bên trái, pin 11,12 là của động cơ bên phải.

Ta cần phải khai báo một số chức năng để có thể sử dụng Arduino

```
1. void setup(){
2.   pinMode(9, OUTPUT);
3.   pinMode(12, OUTPUT);
4.   pinMode(10, OUTPUT);
5.   pinMode(11, OUTPUT);
6. }
```

Hình 4: Khai báo các chân OUTPUT

2.3 Cho xe chạy thẳng về phía trước và lùi về phía sau

Hàm void forward (int speed) và backward (int speed): hàm dùng để cho xe mô phỏng tiến, lùi theo giá trị tốc độ ‘speed’.

```
1. void forward (int speed )
2. {
3.   digitalWrite(10, LOW);
4.   analogWrite(9, speed);
5.   digitalWrite(12, LOW);
6.   analogWrite(11, speed);
7. }
8. void backward (int speed)
9. {
10.  digitalWrite(10, HIGH);
11.  analogWrite(9, speed);
12.  digitalWrite(12, HIGH);
13.  analogWrite(11, speed);
14. }
```

Hình 5: Hàm forward và backward dùng để cho xe chạy tới và chạy lùi

2.4 Cho xe rẽ phải và rẽ trái

Hàm void turnright, turnleft: hàm dùng để cho xe mô phỏng quay phải, quay trái theo giá trị tốc độ ‘speed’.

```
1. void turnright (int speed )
2. {
3.   digitalWrite(10, LOW);
4.   analogWrite(9, speed);
5.   digitalWrite(12, LOW);
6.   analogWrite(11, 0);
7. }
8. void turnleft (int speed ) {
9.   digitalWrite(10, LOW);
10.  analogWrite(9, 0);
11.  digitalWrite(12, LOW);
12.  analogWrite(11, speed);
13. }
```

Hình 6: Hàm turnright và turnleft dùng để cho xe rẽ phải và rẽ trái

2.5 Phần hàm thực thi của chương trình

Sau khi đã thiết kế các hàm để có thể sử dụng điều khiển xe, ta sẽ khởi tạo một hàm thực thi chương trình một cách hoàn chỉnh.

Hàm void loop(): thực hiện các hàm đã khai báo.

```
1. void loop(){  
2.   forward(255);  
3.   delay(2000);  
4.   forward(0);  
5.   delay(2000);  
6.  
7.   turnright(255);  
8.   delay(2000);  
9.   forward(0);  
10.  delay(2000);  
11.  
12.  turnleft(255);  
13.  delay(2000);  
14.  forward(0);  
15.  delay(2000);  
16.  
17.  backward(50);  
18.  delay(2000);  
19.  forward(0);  
20.  delay(2000);  
21. }
```

Hình 7: Hàm thực thi

3 Kết luận

Thông qua các ví dụ về việc áp dụng các kiến thức đã được học phần Arduino, nhóm báo cáo đã trình bày và thực hiện các chương trình điều khiển cho xe chạy theo các hướng mong muốn. Cùng với việc kết hợp các kĩ năng về lập trình Arduino cũng như ngôn ngữ lập trình C++, bài thực hành lần này đã củng cố kiến thức nền tảng và mở rộng về các chức năng của Arduino.



Tài liệu

[Job] Jobit Joseph, *Beginners Guide to Arduino IDE and Arduino Programming* March 2022.