

ROBOT TÌM ĐƯỜNG TRONG MÊ CUNG

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ.....	III
DANH MỤC CÁC TỪ VIẾT TẮT.....	IV
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1 GIỚI THIỆU	1
1.1.1 Lịch sử.....	1
1.1.2 Hướng thực hiện.....	1
CHƯƠNG 2. NGUYÊN LÝ CHUNG CỦA ĐỀ TÀI.....	3
2.1 SƠ ĐỒ KHỐI	3
2.2 SƠ ĐỒ NGUYÊN LÝ	3
2.3 SƠ ĐỒ GIẢI THUẬT	4
CHƯƠNG 3. TÌM HIỂU LINH KIỆN.....	9
3.1 VI ĐIỀU KHIỂN ATMEGA328P	9
3.2 CẢM BIẾN VẬT CẢN HỒNG NGOẠI SHARP GP2Y0A02YK0F	11
3.3 MẠCH ĐIỀU KHIỂN ĐỘNG CƠ L298N	11
3.4 HÌNH ẢNH THI CÔNG PHẦN CỨNG	13
3.4.1 Hình ảnh mạch in.....	13
3.4.2 Hình ảnh thực tế mô hình.....	16
CHƯƠNG 4. NHẬN XÉT.....	17
4.1 NHẬN XÉT	17
CHƯƠNG 5. KẾT LUẬN.....	18
5.1 KẾT LUẬN	18
5.2 HƯỚNG PHÁT TRIỂN	18
TÀI LIỆU THAM KHẢO.....	18
PHỤ LỤC A.....	19

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC TỪ VIẾT TẮT

UART	Universal Asynchronous Receiver – Transmitter
PWM	Pulse Width Modulation
MISO	Master Input / Slave Output
MOSI	Master Output / Slave Input
SCK	Serial Clock
IDE	Intergrated Development Environment

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu

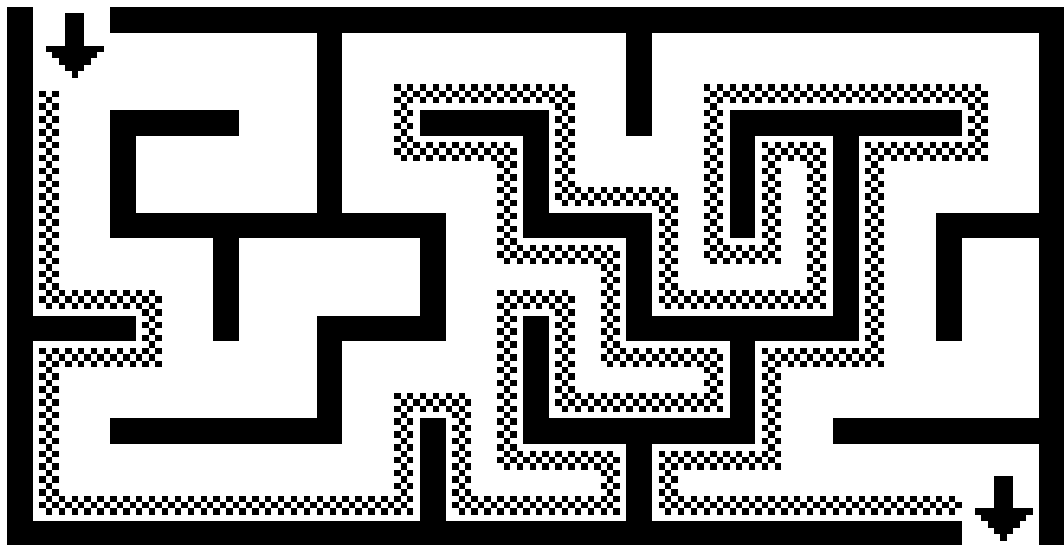
Trong thời đại công nghiệp ngày nay, Robot ngày càng được sử dụng phổ biến trong sản xuất cũng như trong cuộc sống hàng ngày của con người. Robot đã có một vai trò quan trọng khó có thể thay thế được, nó giúp con người việc trong điều kiện nguy hiểm, khó khăn. Ngoài ra, Robot còn được dùng trong các lĩnh vực nguy hiểm không gian, quân sự, giải trí... Lĩnh vực Robot di động ngày càng có được sự quan tâm của các nhà nghiên cứu và xã hội.

1.1.1 Lịch sử

Khái niệm Robot theo nghĩa chung thường được hiểu đồng nghĩa với khái niệm tự động hóa công nghiệp, điều này chỉ đúng một phần bởi vì: thứ nhất, Robot chỉ là một phần trong hệ thống tự động hóa, thứ hai là tự thân việc trình bày miêu tả Robot trong sinh hoạt hàng ngày là ít nhiều sự phóng đại.

1.1.2 Hướng thực hiện

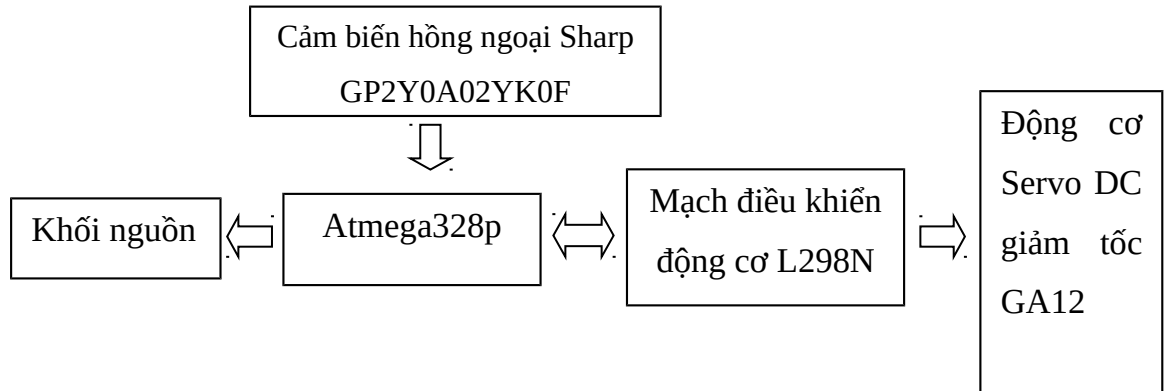
Trong đề tài thực hiện thuật toán bám tường (wall-following) là một quy tắc nổi tiếng để vượt qua mê cung, còn được gọi quy tắc tay trái hoặc quy tắc tay phải. Nếu mê cung chỉ liên thông đơn giản nghĩa là tất cả các bức tường của nó được kết nối với nhau hoặc kết nối với đường bao của mê cung, thì bằng cách dò một tay lên bức tường của nó được kết nối với nhạu hoặc kết nối với đường bao quanh mê cung, thì bằng cách dò một tay lên bức tường của mê cung thì người đi đảm bảo không bị lạc và tìm được lối ra nếu có một lối ra trên đường bao; hoặc nếu không có thì sẽ quay trở lại lối vào và sẽ đi qua tất cả các đường của mê cung ít nhất một lần.



Hình 1-1: Mê cung bám tường trái

CHƯƠNG 2. NGUYÊN LÝ CHUNG CỦA ĐỀ TÀI

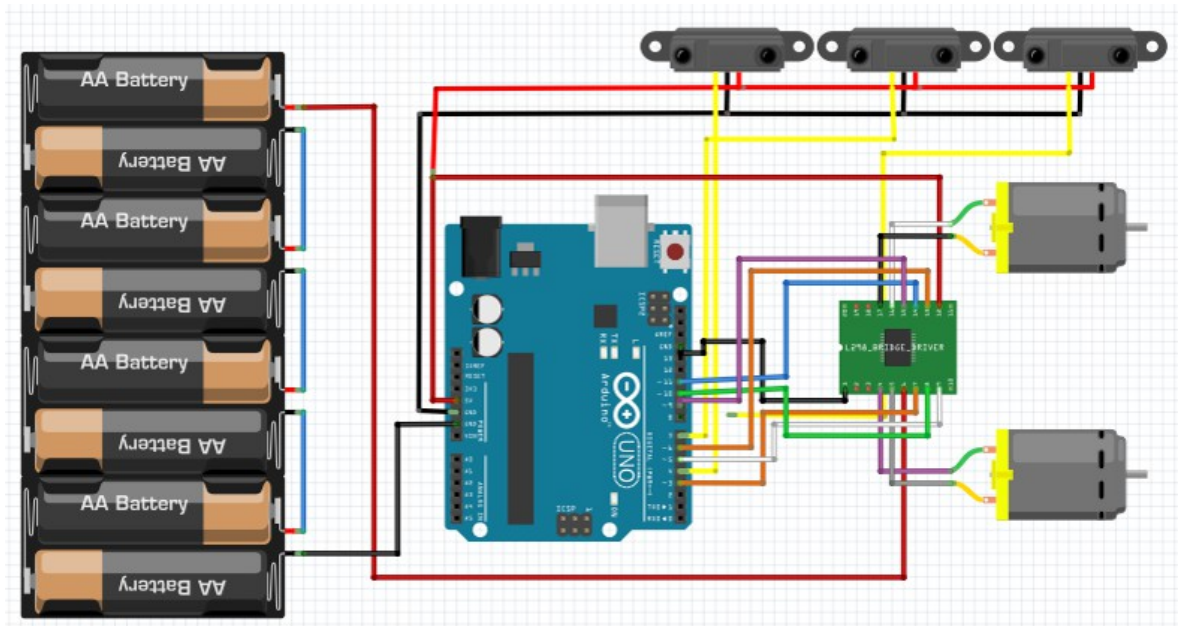
2.1 Sơ đồ khối



Hình 2-1: Sơ đồ khối toàn mạch

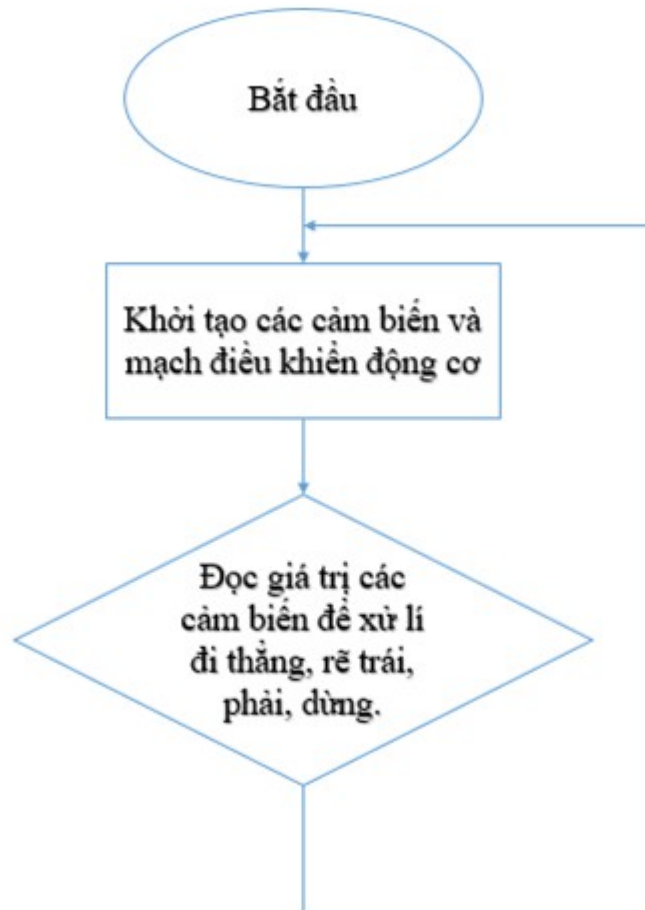
Khi cấp nguồn cho hệ thống, toàn bộ mạch hoạt động cảm biến hồng ngoại đo khoảng cách gửi về cho Atmega328p. Sau đó, Atmega328p xử lý mạch điều khiển động cơ đi thẳng, rẽ trái, rẽ phải, dừng hay tăng tốc độ.

2.2 Sơ đồ nguyên lý



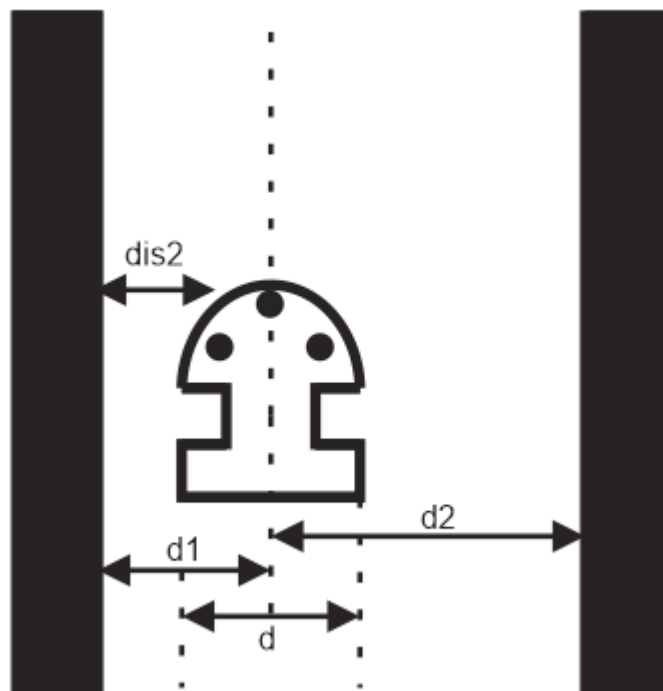
Hình 2-2: Sơ đồ nguyên lý mô phỏng

2.3 Sơ đồ giải thuật



Hình 2-3: Lựa đồ thuật toán của chương trình chính

Trường hợp 1: Robot song song với bức tường



Hình 2-4: Robot song song với tường trái

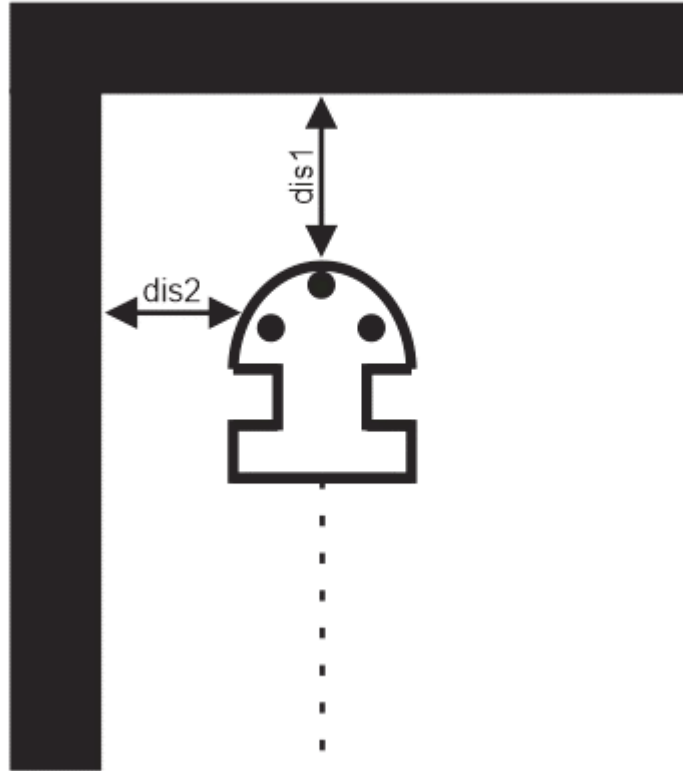
Robot bắt đầu đi thẳng và bám tường trái

Robot song song với 2 bức tường, sẽ đi thẳng đồng thời luôn luôn so sánh x giá trị khoảng cách cho trước để biết được:

Nếu $dis2$ nhỏ hơn x thì lệch trái, và Robot sẽ rẽ phải để cho $dis2 = x$

Nếu $dis2$ lớn hơn x thì lệch phải, và Robot sẽ rẽ trái để cho $dis2 = x$

Trường hợp 2: Robot bị chặn tường trái và phía trước

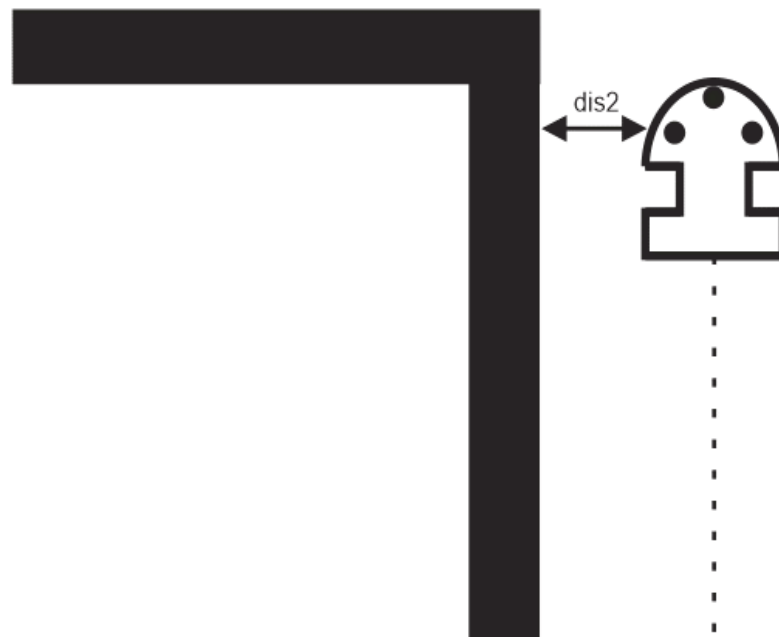


Hình 2-5: Robot bị chặn tường trái và phía trước

Robot sẽ đo được giá trị khoảng cách $dis2$ và $dis1$. Robot sẽ biết mình bị chặn trước và đồng thời đo được giá trị $dis2$ để biết bị chặn hay không.

Sau đó, Robot sẽ dừng lại và rẽ phải, đi thẳng sẽ tiếp tục quay lại trường hợp 1

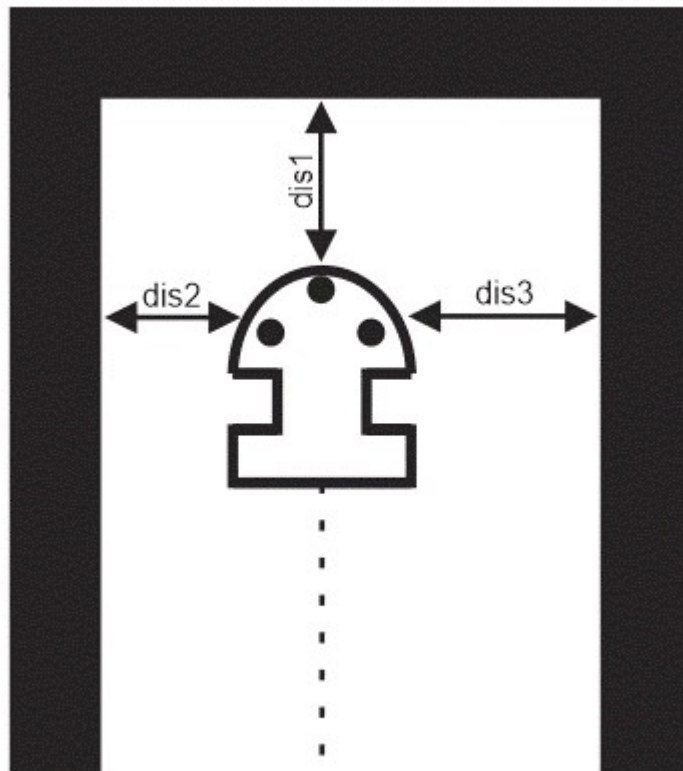
Trường hợp 3: Robot bị chặn tường trái



Hình 2-6: Robot bị chặn tường trái

Robot sẽ đo được khoảng cách trái, phải, trước. Do đó, sẽ biết được và bắt đầu rẽ trái và đi thẳng sẽ quay lại trường hợp 1.

Trường hợp 4: Robot bị chặn phía trước, trái và phải



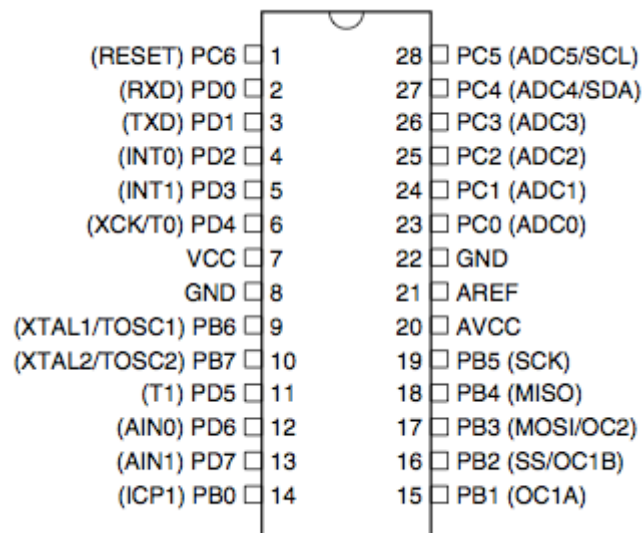
Hình 2-7: Robot bị chặn phía trước, trái và phải.

Robot tiếp tục đo được khoảng cách bằng ba cảm biến hồng ngoại và biết bị chặn 3 phía, nên sẽ rẽ phải 180 độ và đi thẳng tiếp tục quay lại trường hợp 1.

CHƯƠNG 3. TÌM HIỂU LINH KIỆN

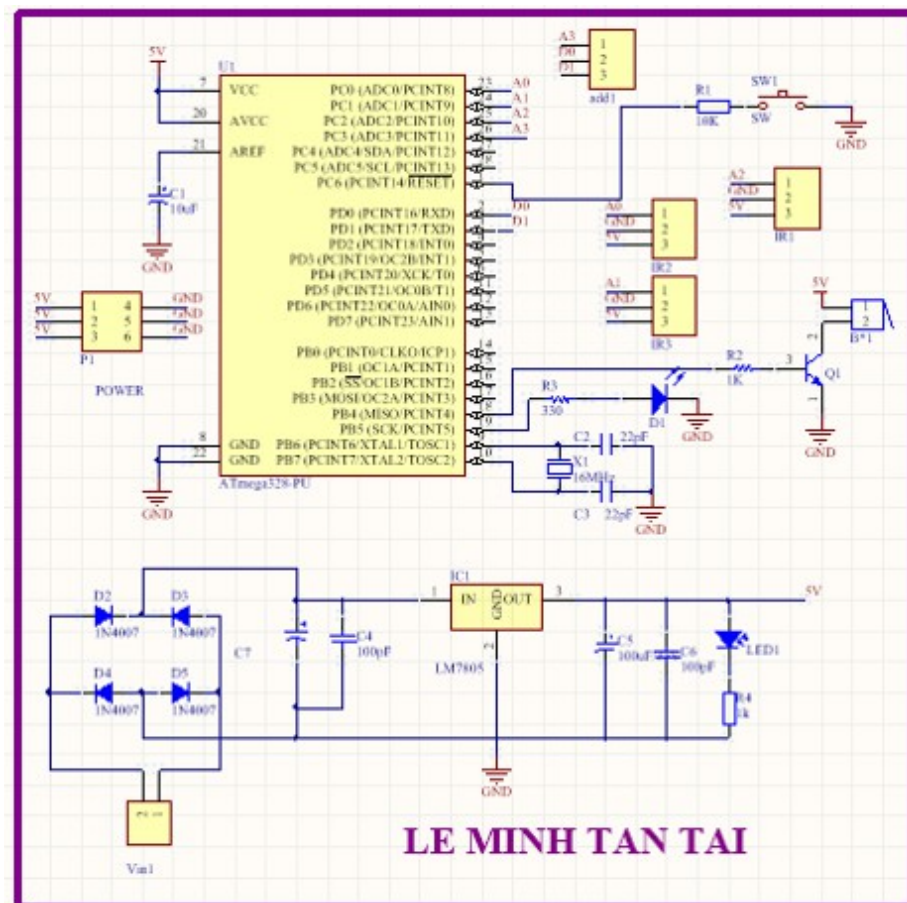
3.1 Vi điều khiển Atmega328p

Hình ảnh vi điều khiển Atmega328p



Hình 3.1: Vi điều khiển Atmega328p

- Kiến trúc: AVR 8bit
- Xung nhịp lớn nhất: 20Mhz
- Bộ nhớ chương trình (FLASH): 32KB
- Bộ nhớ EEPROM: 1KB
- Bộ nhớ RAM: 2KB
- Điện áp hoạt động rộng: 1.8V - 5.5V
- Số timer: 3 timer gồm 2 timer 8-bit và 1 timer 16-bit
- Số kênh xung PWM: 6 kênh (1timer 2 kênh)



Hình 3-2: Mạch nguyên lí vi điều khiển Atmega328p tích hợp mạch nguồn, chân các cảm biến

3.2 Cảm biến vật cản hồng ngoại Sharp GP2Y0A02YK0F



Hình 3-3: Cảm biến vật cản hồng ngoại Sharp GP2Y0A02YK0F

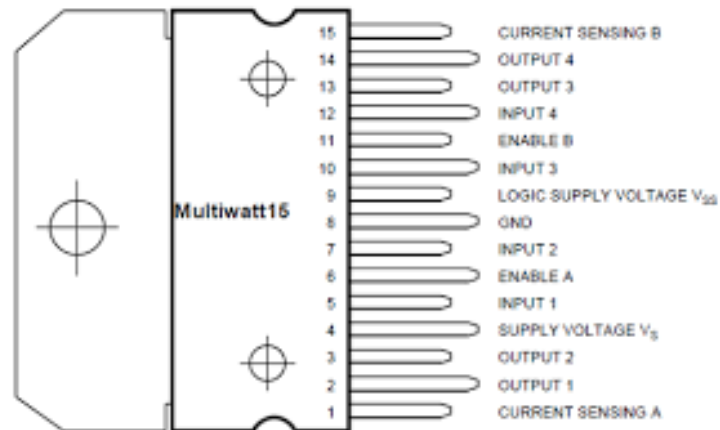
GP2Y0A02YK0F là cảm biến đo khoảng cách dùng tia hồng ngoại để đo khoảng cách tối đa 150cm. Với độ ổn định cao, chống nhiễu tốt và kích thước nhỏ gọn. Giá trị trả về là analog nên rất dễ sử dụng.

Thông số kỹ thuật:

- Phạm vi đo: 20-150cm
- Tín hiệu ra: Điện áp
- Kích thước: 29.5 x 13 x 21.6mm
- Điện áp: 4.5V - 5.5V

3.3 Mạch điều khiển động cơ L298N

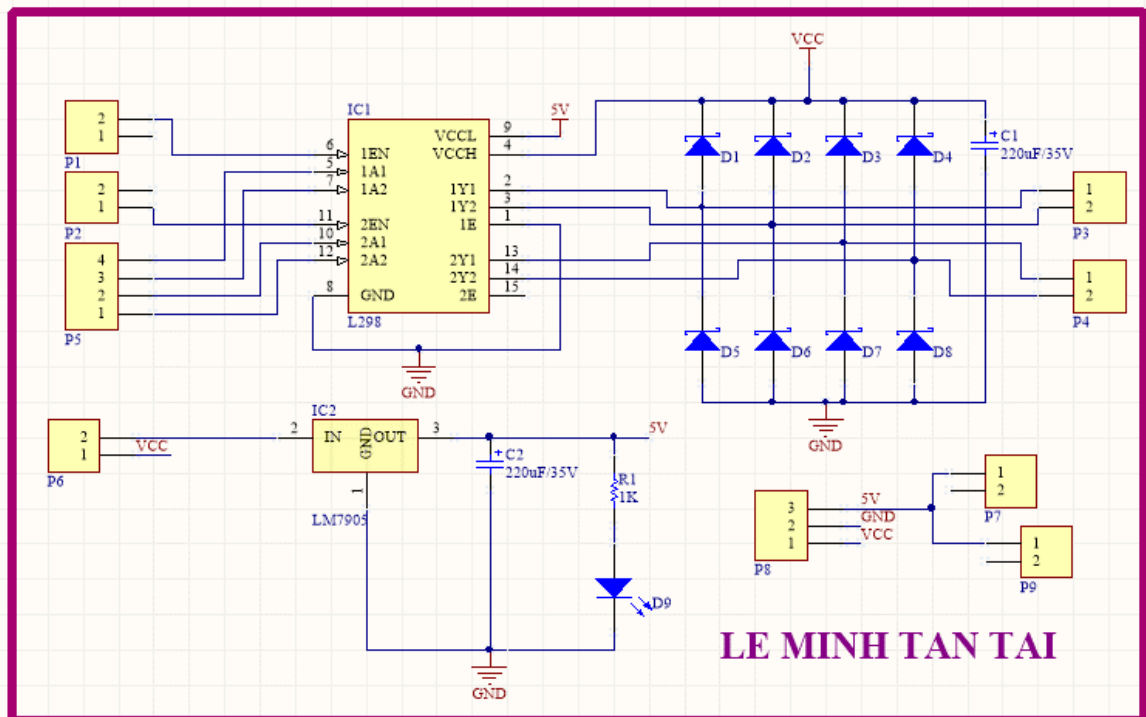
IC L298 là một IC tích hợp nguyên khối gồm 2 mạch cầu H bên trong. Với điện áp làm tăng công suất đầu ra từ 5V – 47V, dòng lên đến 4A, L298 thích hợp trong những ứng dụng công suất nhỏ như động cơ DC loại vừa



Hình 3-4: IC L298N

Bốn chân INPUT: IN1, IN2, IN3, IN4 được nối lần lượt với các chân 5, 7, 10, 12 của L298. Đây là các chân tín hiệu điều khiển.

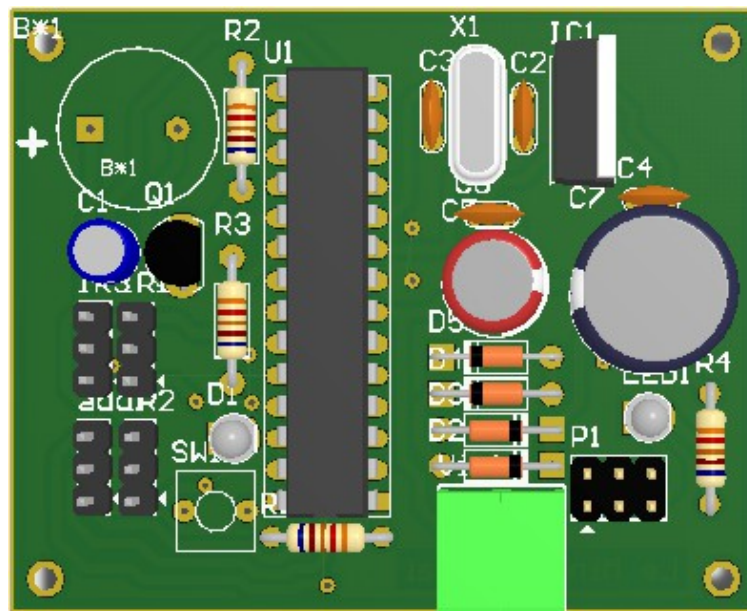
Bốn chân OUTPUT: OUT1, OUT2, OUT3, OUT4 (tương ứng với các chân INPUT) được nối với các chân 2, 3, 13, 14 của L298. Các chân này sẽ được nối với động cơ. Hai chân ENA và ENB dùng để điều khiển các mạch cầu H trong L298. Nếu ở mức logic “1” thì cho phép mạch cầu H hoạt động, nếu ở mức “0” thì mạch cầu H không hoạt động.



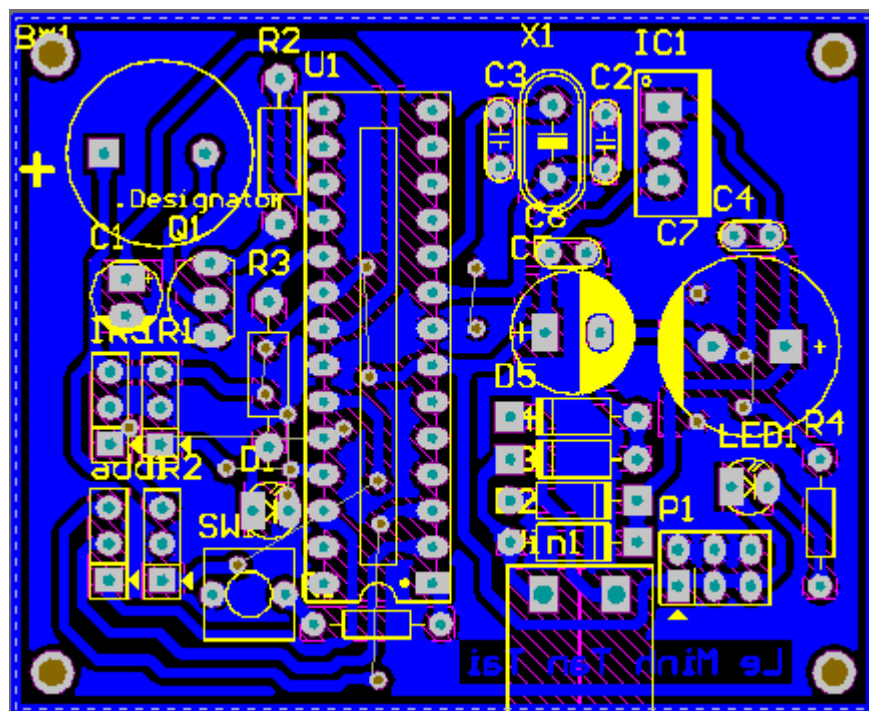
Hình 3-5: Nguyên lí module L298N

3.4 Hình ảnh thi công phần cứng

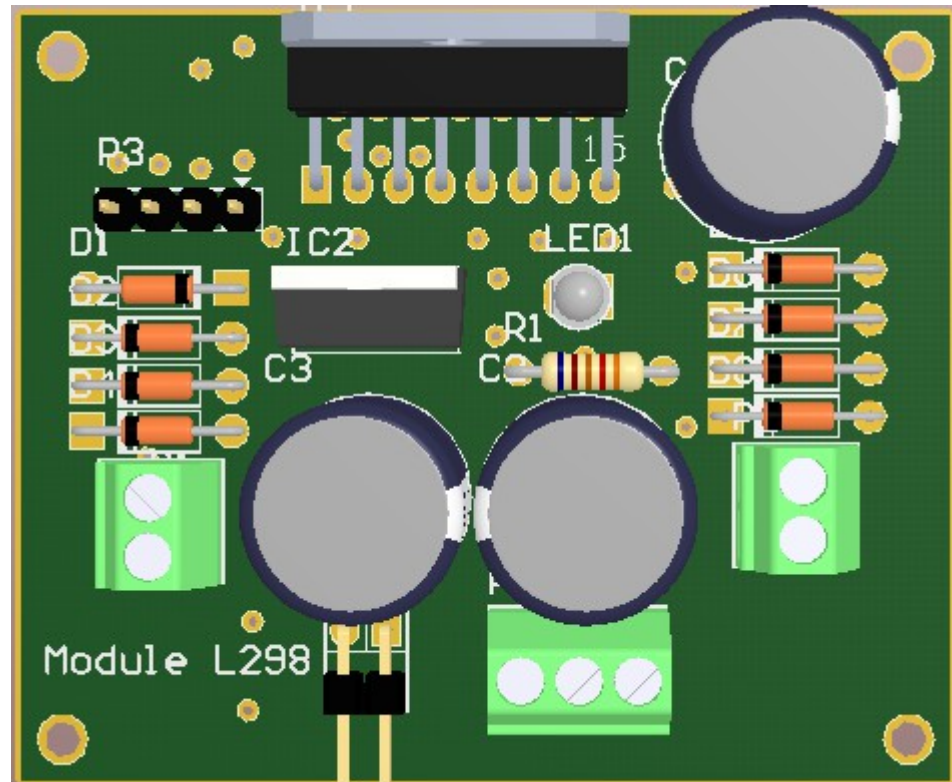
3.4.1 Hình ảnh mạch in



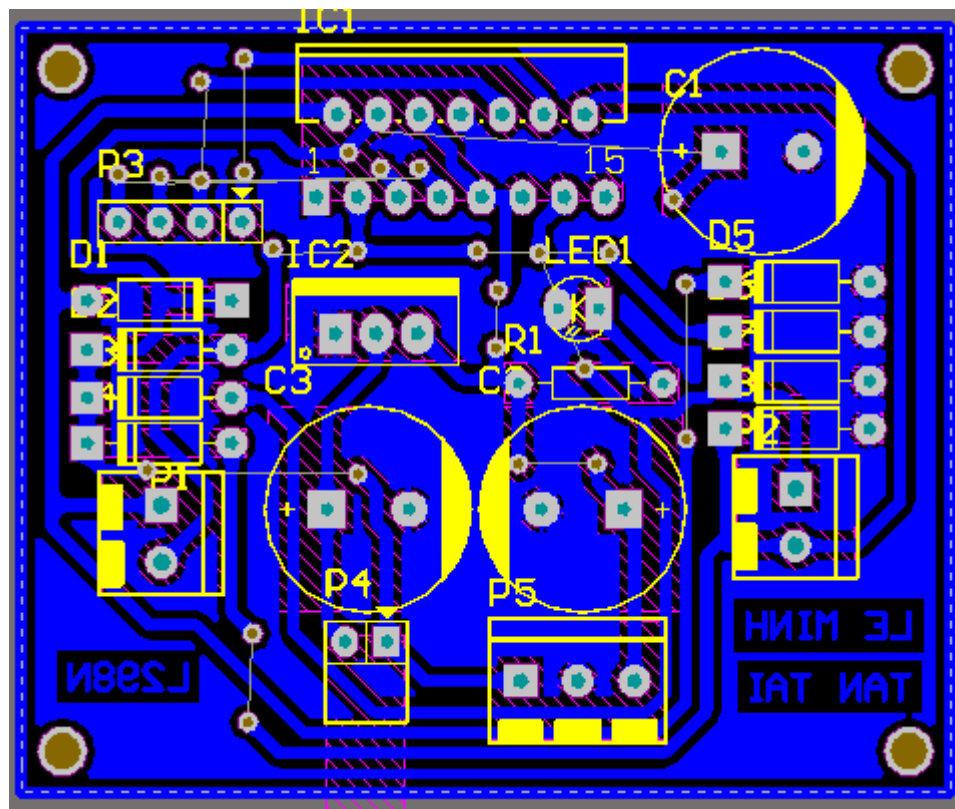
Hình 3-7: Hình ảnh 3D board mạch chính



Hình 3-8: Hình ảnh board mạch in chính

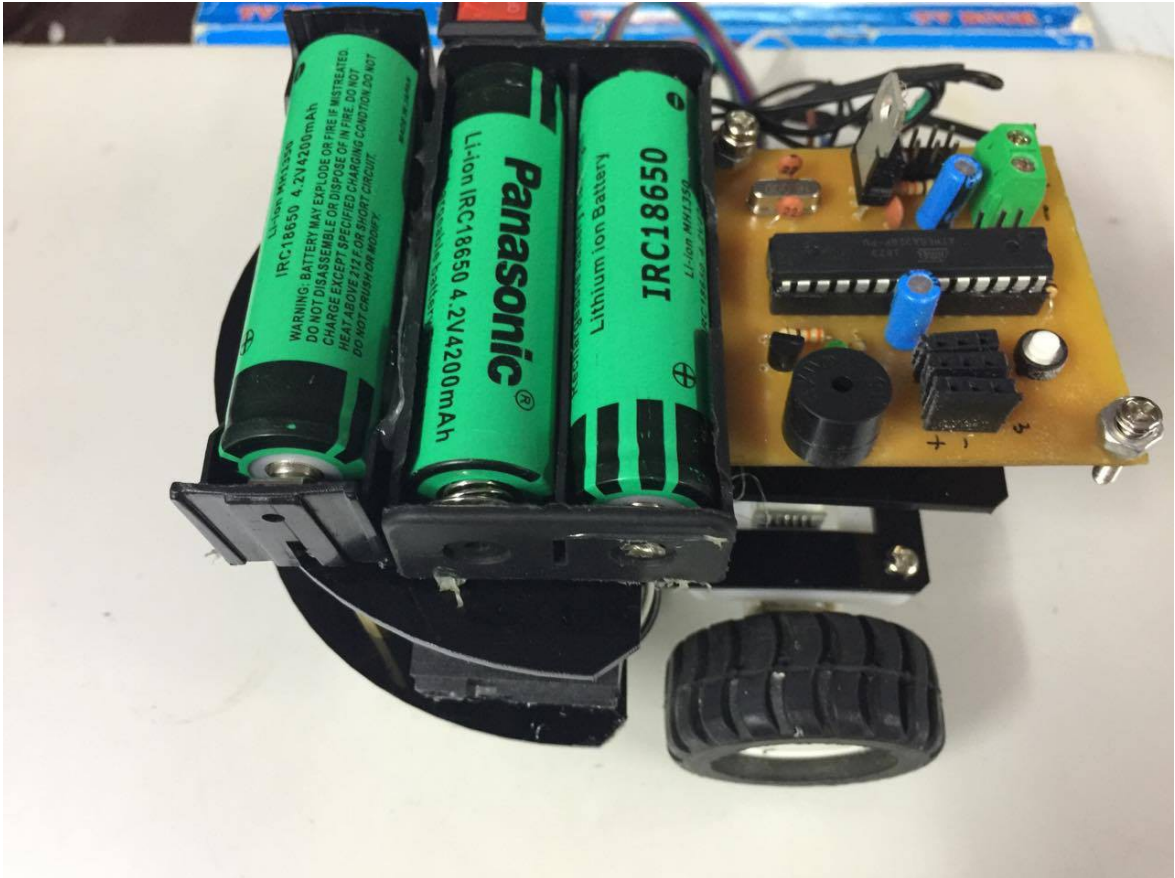


Hình 3-9: Hình ảnh 3D Module L298N



Hình 3-10: Hình ảnh board mạch in Module L298N

3.4.2 Hình ảnh thực tế mô hình



Hình 3-11: Hình ảnh board mạch thực tế

CHƯƠNG 4. NHẬN XÉT

4.1 Nhận xét

Mô hình Robot đã bám được tường trái với tốc độ chậm.

Nếu chạy nhanh robot xử lý không kịp, vì thuật toán chưa tối ưu.

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Sau thực hiện đồ án em rút ra được nhiều kinh nghiệm trong khi làm thực tế trên mô hình và hiểu rõ hơn. Gặp được nhiều sai sót và để rút kinh nghiệm tránh sai sót cho những lần sau.

5.2 Hướng phát triển

Cần xử lý tối ưu, phát triển góc quét để cho ổn định chạy tốc độ cao.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1] <http://arduino.vn/reference>

Tiếng Anh:

[2] <https://www.arduino.cc/en/Reference/>

[3] <https://wikipedia.org/wiki>

[4] http://umpippo.robotica.eng.br/Documentos/95_icar.pdf

[5] <http://www.cs.cmu.edu/~tyata/Abstract/ICRA98.pdf>

[6] https://www.cs.hmc.edu/~dodds/projects/RobS01/Assignment2/Fixed_HTML/follow.html

PHỤ LỤC A

```
//#include <MsTimer2.h>

int ir1= A0;
int val1;
//////////

int ir2 = A1;
int val2;
//////////

int ir3 = A2;
int val3;
//////////

#define IN1 5
#define IN2 4
#define IN3 6
#define IN4 7

void setup() {
    Serial.begin(9600);
    pinMode(IN1,OUTPUT);
    pinMode(IN2,OUTPUT);
    pinMode(IN3,OUTPUT);
    pinMode(IN4,OUTPUT);
    //////////
    MsTimer2::set(10, read_sensor);
    MsTimer2::start();
    //////////
}
```

```
}

```

```
void loop() {

```

```
  //read_sensor();

```

```
  analogWrite(IN1, 100);

```

```
  analogWrite(IN3, 100);

```

```
  analogWrite(IN2, 0);

```

```
  analogWrite(IN4, 0);

```

```
  ////////// Case 1: parallel //////////////////////////////////

```

```
  if(dis2>20){turn_left();delay(50);go_straight();delay(50);}

```

```
  if(dis2<8){ turn_right();delay(50);go_straight();delay(50);}

```

```
  if((dis2>=8)&&(dis2<=20)){go_straight();}

```

```
  //////////Case 2: turn right //////////////////////////////////

```

```
  if((dis1      <=      20)      &&      (      dis3>30      )      )

```

```
  { Stop();delay(300);turn_right();delay(800);go_straight();delay(50);}

```

```
  ////////// Case 3: turn left //////////////////////////////////

```

```
  if((dis1      >      20)      &&      (      dis2      >      30)      &&      (dis3      >30))

```

```
  { Stop();delay(300);turn_left();delay(800);go_straight();delay(50);}

```

```
  ////////// case 4: turn around //////////////////////////////////

```

```
  if((dis1      <20)      &&      (dis2      <      20)&&      (dis3      <20))

```

```
  { Stop();turn_right();delay(1800);go_straight();delay(50);}

```

```

}

```

```
void read_sensor()

```

```
{

```

```
  val1= digitalRead(ir1);

```



```
val2=digitalRead(ir2);
```

```
val3=digitalRead(ir3);
```

```
Serial.println("IR 1:");
```

```
Serial.println(val1);
```

```
Serial.println("IR 2:");
```

```
Serial.println(val2);
```

```
Serial.println("IR 3:");
```

```
Serial.println(val3);
```

```
}
```

```
void turn_left()
```

```
{
```

```
    digitalWrite(IN1,LOW);
```

```
    digitalWrite(IN2, LOW);
```

```
    analogWrite(IN3, 100);
```

```
    digitalWrite(IN4, LOW);
```

```
}
```

```
void turn_right()
```

```
{
```

```
    analogWrite(IN1, 100);
```

```
    digitalWrite(IN2, LOW);
```

```
    digitalWrite(IN3, LOW);
```

```
    digitalWrite(IN4, LOW);
```

```
}
```

```
void go_straight()
```

```
{  
  analogWrite(IN1, 100);  
  digitalWrite(IN2, LOW);  
  analogWrite(IN3, 100);  
  digitalWrite(IN4, LOW);  
}  
void Stop()  
{  
  digitalWrite(IN1, LOW);  
  digitalWrite(IN2, LOW);  
  digitalWrite(IN3, LOW);  
  digitalWrite(IN4, LOW);  
  
}
```