

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ- VẬT LÝ KỸ THUẬT
BỘ MÔN VẬT LÝ ĐIỆN TỬ



BÁO CÁO ĐỀ TÀI MÔN HỌC
THIẾT KẾ HỆ THỐNG NHÚNG
(MHP: PHY10104)

Đề tài:

ĐIỀU KHIỂN XE ROBOT DI CHUYỂN TỰ ĐỘNG
TRÁNH VẬT CẢN

Nhóm thực hiện:

Phạm Hữu Thọ (NT)	1713131
Đinh Hoàng Ngân	1713086
Lê Nguyễn Thành Luân	1713074

Giáo viên hướng dẫn: TS NGUYỄN CHÍ NHÂN

.....

HỒ CHÍ MINH, December, 2019

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ- VẬT LÝ KỸ THUẬT
BỘ MÔN VẬT LÝ ĐIỆN TỬ



**BÁO CÁO ĐỀ TÀI MÔN HỌC
THIẾT KẾ HỆ THỐNG NHÚNG**

(MHP: PHY10104)

Đề tài:

**ĐIỀU KHIỂN XE ROBOT DI CHUYỂN TỰ ĐỘNG
TRÁNH VẬT CẢN**

Nhóm thực hiện:

Phạm Hữu Thọ (NT)	1713131
Đinh Hoàng Ngân	1713086
Lê Nguyễn Thành Luân	1713074

Giáo viên hướng dẫn: TS NGUYỄN CHÍ NHÂN

.....

LỜI MỞ ĐẦU

Các Robot tự hành được ứng dụng trong đời sống ngày càng nhiều như robot vận chuyển hàng hóa, robot kiểm tra nguy hiểm, robot xe lăn cho người khuyết tật. Robot phục vụ sinh hoạt gia đình... Điểm hạn chế của các robot tự hành hiện tại là tính thiếu linh hoạt và khả năng thích ứng khi làm việc ở những vị trí khác nhau. Từ những lý do đó nảy sinh vấn đề tránh vật cản cho robot tự hành nhằm nâng cao tính linh hoạt cho robot. Hầu hết các robot hiện đại đều có một kiểu tránh vật cản nào đó robot xác định vật cản và dừng lại ở khoảng cách gần so với vật cản để tránh va chạm, đến các thuật toán tinh tế hơn. Đề tài “XE ROBOT DI CHUYỂN TỰ ĐỘNG TRÁNH VẬT CẢN” nhằm góp phần giải quyết các vấn đề trên.

Nội dung của đề tài được phân thành các chương mục như sau:

Chương 1: Tổng quan đề tài

Chương 2: Thiết kế hệ thống

Chương 3: Kiểm tra hoạt động của hệ thống

Chương 4: Kết luận và hướng phát triển

Danh sách các thành viên thực hiện đề tài:

STT	MSSV	Họ và tên	Chữ ký
1	1713131	Phạm Hữu Thọ	
2	1713074	Lê Nguyễn Thành Luân	
3	1713086	Đinh Hoàng Ngân	

MỤC LỤC

LỜI MỞ ĐẦU	1
MỤC LỤC	2
MỤC LỤC HÌNH.....	4
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	6
1.1 Giới thiệu.	6
1.1.1 Tổng quan về xe cảm biến tự động tránh vật cản.	6
1.1.2 Yêu cầu hệ thống.....	6
1.1.3 Mô hình robot.....	7
1.2 Sơ đồ hệ thống.	7
1.3 Phần cứng cho hệ thống.....	8
1.3.1 Kit RF thu phát wifi ESP8266 NodeMCU.	8
1.3.2 Arduino Uno.	9
1.3.3 Module L298 v2.....	10
1.3.4 I2C LCD.	10
1.3.5 Servo.....	11
1.3.6 Cảm biến siêu âm SRF-05.....	12
1.3.7 Relay.....	13
1.3.8 Quạt tản nhiệt.....	13
1.3.9 DHT 22.....	14
1.3.10 Động cơ DC motor.....	14
1.3.11 Công suất mạch.....	15
CHƯƠNG 2: THIẾT KẾ HỆ THỐNG.....	16
2.1 Thiết kế sơ đồ mạch (schematic) và layout mạch.....	16
2.1.1 Thiết kế sơ đồ mạch (schamatic).	16

2.1.2	Mạch layout.	17
2.2	Mạch thực tế.....	17
2.3	Lưu đồ thuật toán và viết phần mềm nhúng điều khiển.....	18
2.3.1	Lưu đồ thuật toán	18
2.3.2	Viết phần mềm nhúng (firmware) điều khiển.	20
2.4	Giám sát điều khiển mô hình hệ thống thông qua điện thoại thông minh. .	28
2.4.1	Giới thiệu sơ nét về app blynk.	28
2.4.2	Thiết kế giao diện app blynk trên điện thoại.	29
2.5	Mô hình demo.	32
CHƯƠNG 3: KIỂM TRA HOẠT ĐỘNG CỦA HỆ THỐNG.....		33
3.1	Các bước kiểm tra hoạt động của hệ thống.	33
3.2	Đánh giá hoạt động của hệ thống.....	33
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		34
4.1	Kết quả đạt được.	34
4.2	Đề xuất hướng phát triển.	34
PHỤ LỤC		35
Source code Node MCU Esp8266.....		35
Source code Arduino Uno R3.....		39
TÀI LIỆU THAM KHẢO		48

MỤC LỤC HÌNH

Hình 1 Mô hình robot dò đường bằng sóng siêu âm.	7
Hình 2 Sơ đồ hệ thống.	7
Hình 3 Sơ đồ chân Node MCU ESP8266.	8
Hình 4 Board mạch Arduino.	9
Hình 5 Module điều khiển động cơ L298 v2.	10
Hình 6 Màn hiển thị LCD.	10
Hình 7 Động cơ Servo.....	11
Hình 8 Module cảm biến hồng ngoại.	12
Hình 9 Module relay.	13
Hình 10 Quạt tản nhiệt.	13
Hình 11 Module cảm biến nhiệt độ.	14
Hình 12 Động cơ DC.	14
Hình 13 Mạch thiết kế trên Proteus.	16
Hình 14 Mạch layout của nhóm.	17
Hình 15 Board thực tế.	17
Hình 16 Lưu đồ thuật toán.	18
Hình 17 Giới thiệu về app Blynk.....	28
Hình 18 Bảng cài đặt nút bấm.	29
Hình 19 Bảng cài đặt Gause.	29
Hình 20 Bảng cài đặt LCD.	30
Hình 21 Bảng cài đặt joystick.	30
Hình 22 Bảng cài đặt email.	31

Hình 23 Giao diện app Blynk thiết kế theo đề tài.	31
Hình 24 Mô hình demo xe tránh vật cản.....	32

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.

1.1 Giới thiệu.

Trong chương này, em giới thiệu sơ bộ về robot xe cảm biến, trong đó nội dung chính trình bày bao gồm: mô hình, sơ đồ khối và nguyên lý hoạt động của robot. Với chức năng điều khiển bằng tay, chúng ta điều khiển bằng sóng wifi 2.4GHz thông qua app blynk và NodeMCU ESP8266. Khi chuyển sang chế độ tự động, xe sẽ chuyển từ trạng thái điều khiển bằng tay sang trạng thái tự động dò đường tránh vật cản bằng cảm biến siêu âm SRF-05.

1.1.1 Tổng quan về xe cảm biến tự động tránh vật cản.

Robot tránh chướng ngại vật là một thiết bị thông minh có thể tự động cảm nhận chướng ngại vật phía trước nó và tránh chúng bằng cách tự xoay theo hướng khác. Thiết kế này cho phép robot điều hướng trong môi trường không xác định bằng cách tránh va chạm, đây là yêu cầu chính đối với bất kỳ robot di động tự trị nào. Ứng dụng của Robot Tránh chướng ngại vật không bị giới hạn và nó được sử dụng trong hầu hết các tổ chức quân sự hiện nay, giúp thực hiện nhiều công việc rủi ro không thể thực hiện bởi bất kỳ người lính nào. Ở đây, một cảm biến siêu âm được sử dụng để cảm nhận các chướng ngại vật trên đường đi bằng cách tính khoảng cách giữa robot và chướng ngại vật. Nếu robot tìm thấy bất kỳ chướng ngại vật nào, nó sẽ thay đổi hướng và tiếp tục di chuyển.

1.1.2 Yêu cầu hệ thống.

a. Yêu cầu chức năng.

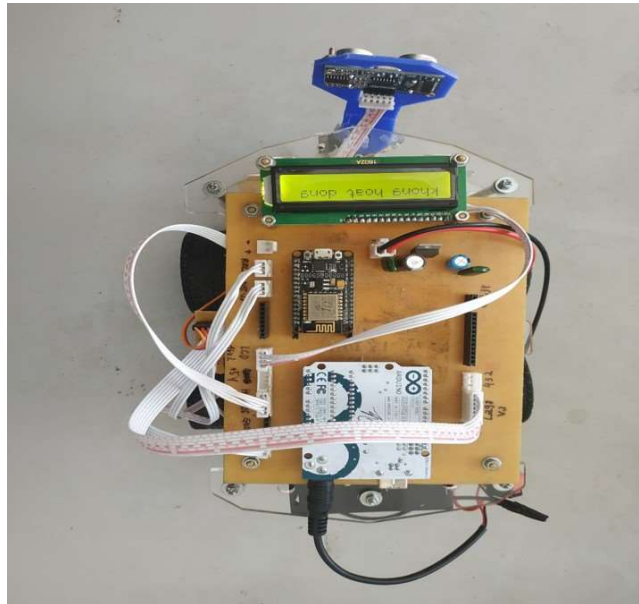
Robot có thể tránh vật cản.

b. Yêu cầu phi chức năng.

Robot hoạt động ổn định trong nhiều điều kiện khác nhau.

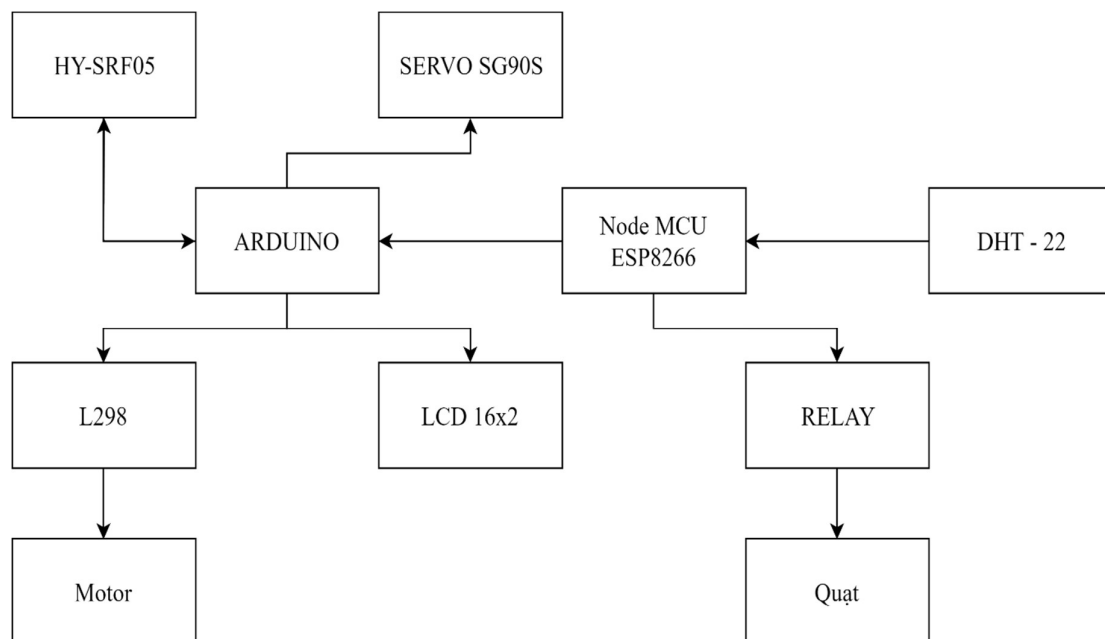
Dễ dàng tháo lắp, thay thế các module khi xảy ra các vấn đề.

1.1.3 Mô hình robot.



Hình 1 Mô hình robot dò đường bằng sóng siêu âm.

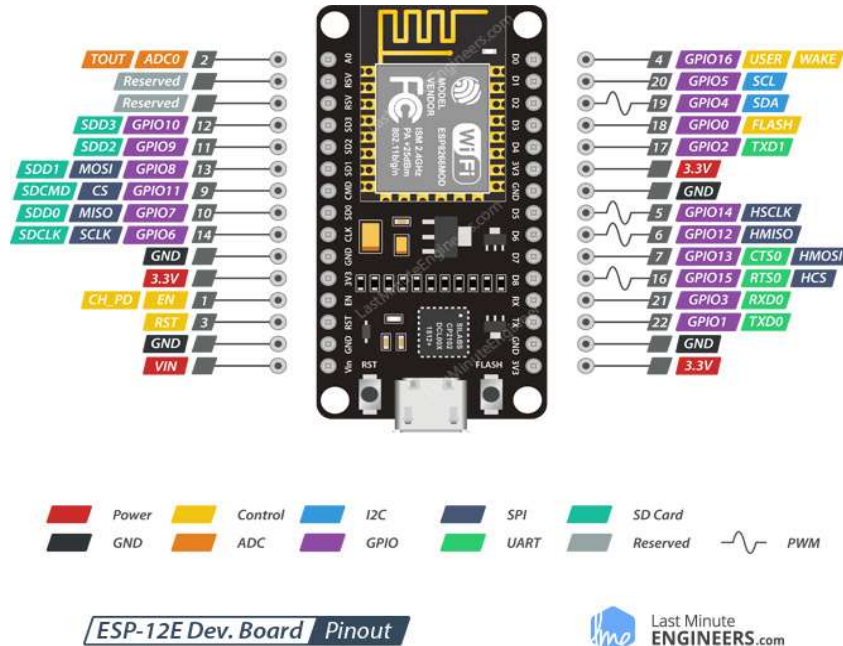
1.2 Sơ đồ hệ thống.



Hình 2 Sơ đồ hệ thống.

1.3 Phần cứng cho hệ thống.

1.3.1 Kit RF thu phát wifi ESP8266 NodeMCU.



Hình 3 Sơ đồ chân Node MCU ESP8266.

- IC chính: ESP8266 Wifi SoC.
- Phiên bản firmware: NodeMCU Lua.
- Chip nạp và giao tiếp UART: CP2102.
- GPIO tương thích hoàn toàn với firmware Node MCU.
- Cấp nguồn: 5VDC MicroUSB hoặc Vin.
- GPIO giao tiếp mức 3.3VDC.
- Tích hợp Led báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 25 x 50 mm.

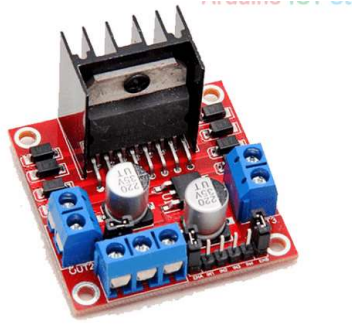
1.3.2 Arduino Uno.



Hình 4 Board mạch Arduino.

- Chip điều khiển chính: ATmega328P.
- Chip nạp và giao tiếp UART: ATmega16U2.
- Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ chân DC.
- Số chân Digital I/O: 14 (trong đó 6 chân có khả năng xuất xung PWM).
- Số chân PWM Digital I/O: 6.
- Số chân Analog Input: 6.
- Dòng điện DC Current trên mỗi chân I/O: 20 mA.
- Dòng điện DC Current chân 3.3V: 50 mA.
- Flash Memory: 32 KB (ATmega328P), 0.5 KB dùng cho bootloader.
- SRAM: 2 KB (ATmega328P).
- EEPROM: 1 KB (ATmega328P).
- Clock Speed: 16 MHz.
- Kích thước: 68.6 x 53.4 mm.

1.3.3 Module L298 v2.



Hình 5 Module điều khiển động cơ L298 v2.

- IC chính: L298 - Dual Full Bridge Driver.
- Điện áp đầu vào: 5~46VDC.
- Công suất tối đa: 25W 1 cầu.
- Dòng tối đa cho mỗi cầu H là: 2A.
- Mức điện áp logic: Low -0.3V~1.5V, High: 2.3V~Vss.

1.3.4 I2C LCD.



Hình 6 Màn hiển thị LCD.

- Điện áp hoạt động là 5 V.
- Chữ đen, nền xanh lá.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu.
- Có bộ ký tự được xây dựng hỗ trợ tiếng Anh và tiếng Nhật.

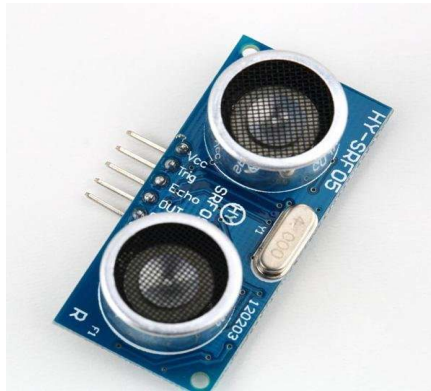
1.3.5 Servo.



Hình 7 Động cơ Servo.

- Model: MG90S servo.
- Điện áp hoạt động: 4.8 ~ 6VDC.
- Stall Torque: 1.8kg/cm(4.8V),2.2kg/cm(6V).
- Tốc độ quay: 0.1sec/60degree (4.8v), 0.08sec/60degree (6v).
- Bánh răng: Kim loại.
- Độ dài dây nối: 175mm.
- Trọng lượng:Khoảng 13.4g.
- Kích thước: 22.8 x 12.2 x 28.5mm.

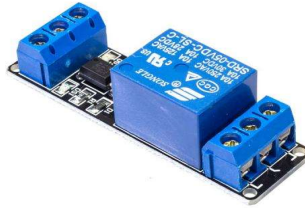
1.3.6 Cảm biến siêu âm SRF-05.



Hình 8 Module cảm biến hồng ngoại.

- Điện áp hoạt động: 5VDC.
- Dòng tiêu thụ: 10~40mA.
- Tín hiệu giao tiếp: TTL.
- Chân tín hiệu: Echo, Trigger (thường dùng) và Out (ít dùng).
- Góc quét: <15 độ.
- Tần số phát sóng: 40Khz.
- Khoảng cách đo được: 2~450cm (khoảng cách xa nhất đạt được ở điều kiện lý tưởng với không gian trống và bề mặt vật thể bằng phẳng, trong điều kiện bình thường cảm biến cho kết quả chính xác nhất ở khoảng cách <100cm).
- Sai số: 0.3cm (khoảng cách càng gần, bề mặt vật thể càng phẳng sai số càng nhỏ).
- Kích thước: 43mm x 20mm x 17mm.

1.3.7 Relay.



Hình 9 Module relay.

- Điện áp cung cấp: 5V DC (cuộn dây rơle).
- Tín hiệu điều khiển: TTL (3.3V hoặc 5V).
- Để kích hoạt đầu ra NO: 0 Volts.
- Số lượng rơle (kênh): 1 CH.
- Mã chuyển tiếp: Songle SRD-12VDC-SL-C.
- Công suất tối đa: 10A / 250VAC, 10A / 30VDC.
- Dòng tối đa: 10A (NO), 5A (NC).
- Thời gian hành động: 10 ms / 5 ms.
- Đèn LED kích hoạt.

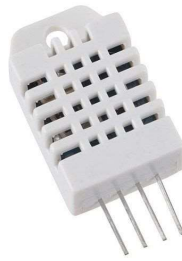
1.3.8 Quạt tản nhiệt.



Hình 10 Quạt tản nhiệt.

- Kích thước: 120 x 120 mm.
- Dòng điện: Xoay chiều.
- Điện áp: 5A-220V.

1.3.9 DHT 22.



Hình 11 Module cảm biến nhiệt độ.

- Dòng sử dụng max: 2.5mA (trong lúc yêu cầu data).
- Đọc tốt độ ẩm 0-100% với độ chính xác 2-5%.
- Đọc tốt Nguồn 3~5 VDC và I/O.
- Nhiệt độ -40 ~ 80°C độ chính xác $\pm 0.5^{\circ}\text{C}$.
- Tốc độ lấy mẫu không lớn hơn 0.5 Hz (2 giây 1 lần).
- Kích thước: 27 x 59 x 13.5mm.
- 4 pins, khoảng cách chân 2.54mm.

1.3.10 Động cơ DC motor.



Hình 12 Động cơ DC.

- Điện áp hoạt động: 3 – 6VDC.
- Dòng điện tiêu thụ: 110 – 140mA.
- Tỉ số truyền: 1:48.
- 125 vòng/ 1 phút tại 3VDC.
- 208 vòng/ 1 phút tại 6VDC.
- Moment: 0.8KG.CM.

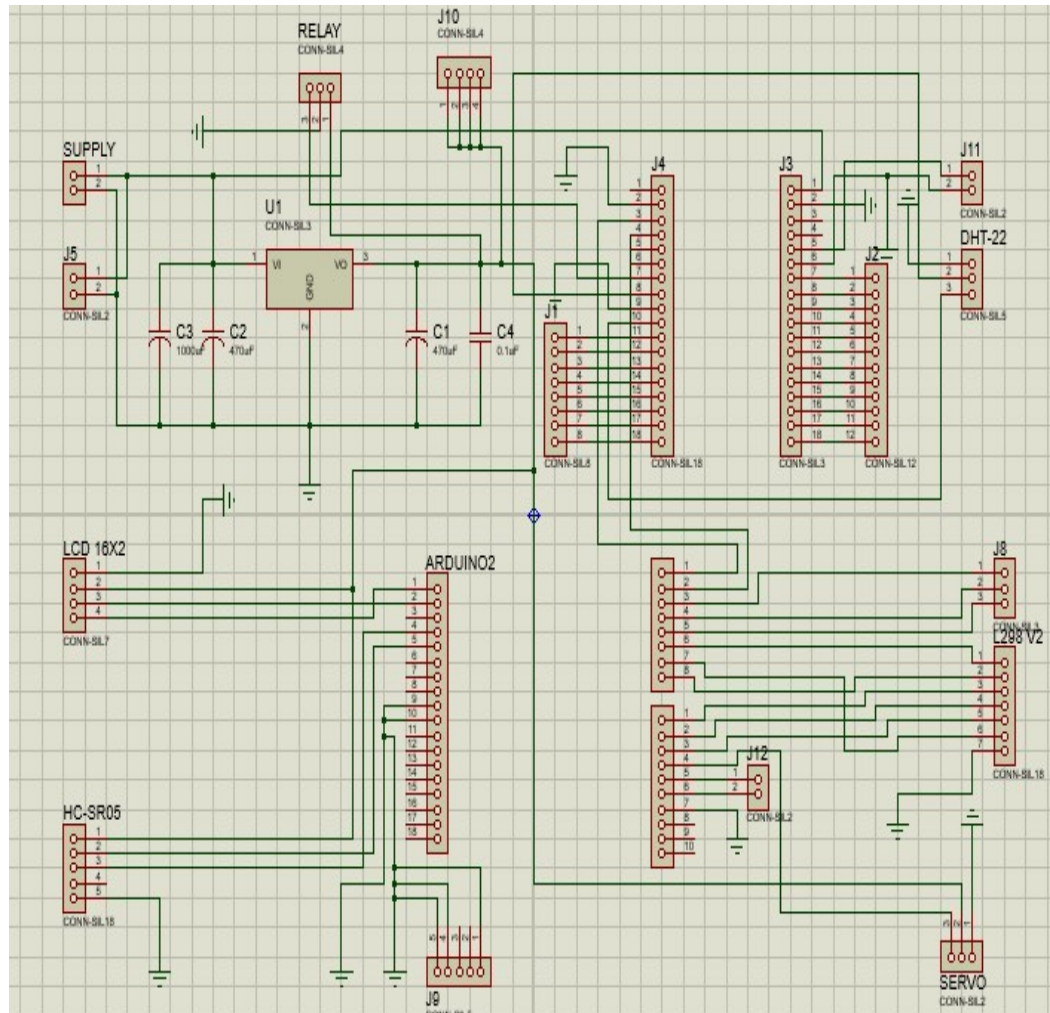
1.3.11 Công suất mạch.

Các thiết bị	Điện áp hoạt động		Công suất tiêu thụ trung bình
	Min	max	
ARDUINO UNO R3	+6VDC	+10VDC	100mW
NODE MCU ESP8266	+5VDC	+10VDC	66mW
MODULE L298 V2	+5VDC	+56VDC	25000mW
HY-SRF05		+5VDC	125mW
LCD 16x2		+5VDC	100mW
MOTOR	+3VDC	+12VDC	24000W
SERVO MG90S	+4,8VDC	+6VDC	100mW
Quạt tản nhiệt		+12V	18000mW
DHT 22	+3VDC	+5VDC	100mW
Tổng công suất tiêu thụ			67591mW ~ 68W

CHƯƠNG 2: THIẾT KẾ HỆ THỐNG.

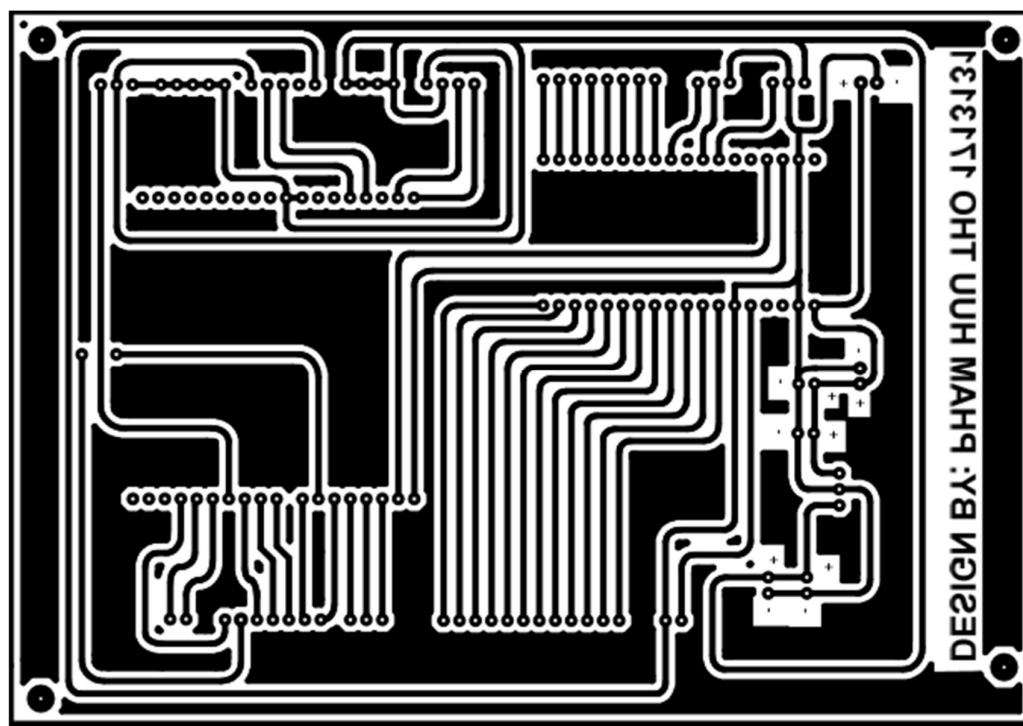
2.1 Thiết kế sơ đồ mạch (schematic) và layout mạch.

2.1.1 Thiết kế sơ đồ mạch (schamatic).



Hình 13 Mạch thiết kế trên Proteus.

2.1.2 Mạch layout.



Hình 14 Mạch layout của nhóm.

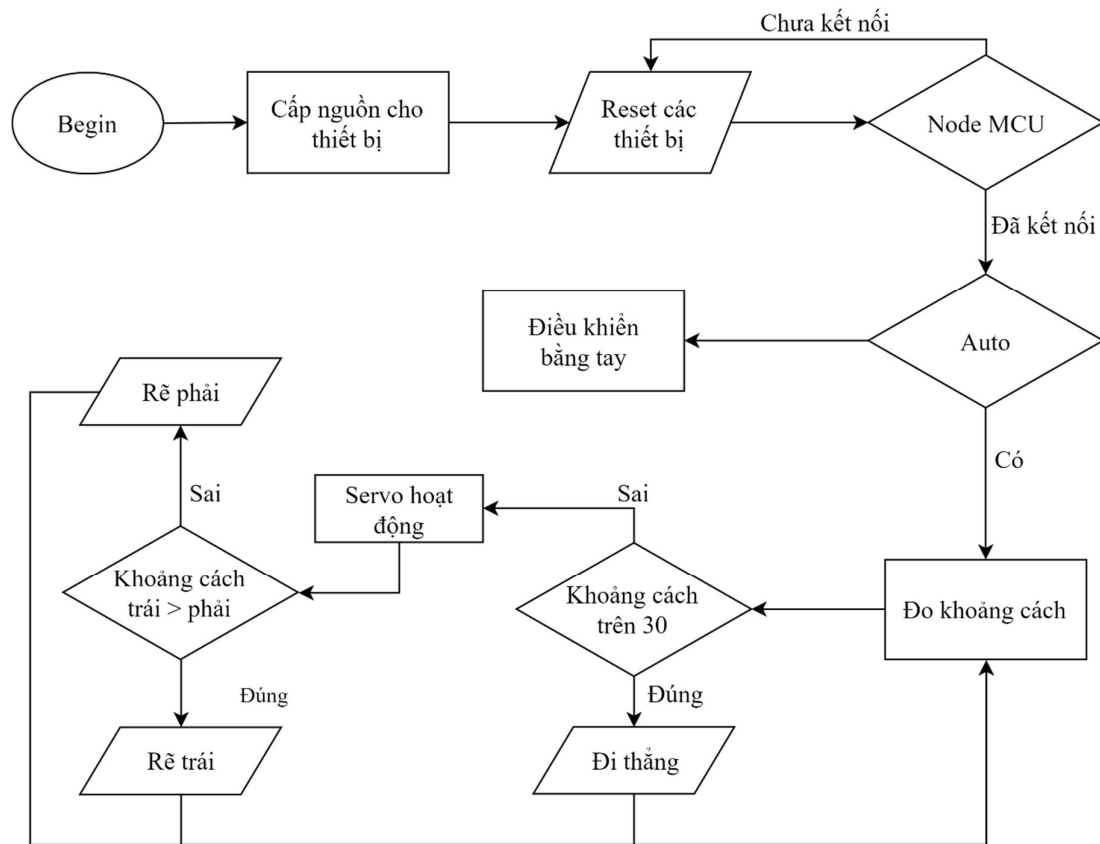
2.2 Mạch thực tế.



Hình 15 Board thực tế.

2.3 Lưu đồ thuật toán và viết phần mềm nhúng điều khiển.

2.3.1 Lưu đồ thuật toán



Hình 16 Lưu đồ thuật toán.

Khi cấp nguồn cho thiết bị, điện thoại sẽ reset và kết nối wifi đồng bộ với Node MCU ESP8266 của xe. Nếu chưa kết nối được thì hệ thống sẽ tiếp tục reset và tìm kiếm.

Sau khi kết nối thành công giữa Node MCU vs điện thoại. Ta sẽ chọn chế độ cho hệ thống hoạt động bằng tay hay chạy tự động.

- Nếu điều khiển bằng tay thì ta sử dụng trình điều khiển bằng tay đã setup trên app Blynk để điều khiển.
- Nếu sử dụng tự động:
 - Đầu tiên cảm biến SRF-05 sẽ đo dò vật cản bằng cảm biến tránh vật cản.
 - Sau đó:
 - Nếu trong khoảng cách trong 30 cm không có vật cản thì xe sẽ chạy thẳng.
 - Ngược lại trong khoảng cách dưới 30 cm có vật cản servo sẽ hoạt động quay trái, quay phải để tiếp tục dò và đo khoảng cách tới vật cản. Nếu khoảng cách đến vật cản bên nào lớn hơn hoặc không phát hiện vật cản xe sẽ rẽ về phía bên đó. Quá trình tự động lặp đi lặp lại cho tới khi ngắt tín hiệu tự động từ trình điều khiển trên thiết bị di động.

2.3.2 Viết phần mềm nhúng (firmware) điều khiển.

Source code Node MCU Esp8266.

```
void myTimerEvent()
{
  Blynk.virtualWrite(V1, millis() / 1000);
}
```

Hàm setup thời gian gửi giá trị lên chân ảo V1 ở khoảng thời gian cố định mỗi giây.

```
Blynk.run();
timer.run();
```

Câu lệnh để chạy chương trình đồng bộ với app Blynk

```
  toarduino();
}
  rdata = Serial.read();
  myString = myString + rdata;
```

Chúng ta khởi tạo giao tiếp UART giữa esp8266 và arduino

```
  if ( rdata == '\n')
  {
    String l = getValue(myString, ',', 0);
    String m = getValue(myString, ',', 1);
    String n = getValue(myString, ',', 2);
    firstVal = l.toInt();
    secondVal = m.toInt();
    thirdVal = n.toInt();
    myString = "";
  }
}
}
void sensorvalue1()
```

```

{
  int sdata = firstVal;
  Blynk.virtualWrite(V2, sdata);
}

```

Hàm gửi giá trị lưu vào chân ảo V2 trên app blynk

```

void sensorvalue2()
{
  int sdata = secondVal;
  Blynk.virtualWrite(V3, sdata);
}

```

Hàm gửi giá trị lưu vào chân ảo V3 trên app blynk

```

void sensorvalue3()
{
  int sdata = thirdVal;
  Blynk.virtualWrite(V4, sdata);
  float doam = dht.getHumidity();
  float nhietdo = dht.getTemperature();
  Serial.print("do am trong khong khi la: ");
  Serial.print(doam);
  Serial.println(" %");
  Serial.print("nhiet do moi truong la: ");
  Serial.print(nhietdo);
  Serial.println(" 'C'");
  Blynk.virtualWrite(V30, doam);
  Blynk.virtualWrite(V31, nhietdo);
  if(nhietdo < 32){
    Blynk.virtualWrite(V32, "Nhiet do BT");
    Blynk.virtualWrite(V33, "quat khong chay");
    digitalWrite(D6,LOW);
  }
}

```

```

    } else {
        Blynk.email("1713131@student.hcmus.edu.vn", "nhiet do dang qua cao. Can
kiem tra lai CPU");
        Blynk.virtualWrite(V32, "Nhiet do cao");
        Blynk.virtualWrite(V33, "quat dang chay");
        digitalWrite(D6,HIGH);
    }
}

```

Hàm gửi các giá trị nhớ lên chân V4 và giá đo được từ cảm biến lên các chân ảo V30 V31 V32 V33

```

String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

```

Hàm trên chuyển các số sang chữ để giao tiếp qua UART.

```

BLYNK_WRITE(V10)
{
    pinValue1 = param.asInt();
}

```


Hàm khởi tạo chân điều khiển từ chân ảo V10 điều khiển động cơ

```
BLYNK_WRITE(V11)
{
  pinValue2 = param.asInt();
}
```

Hàm khởi tạo chân điều khiển chân ảo V11 điều khiển động cơ

```
BLYNK_WRITE(V12)
{
  pinValue3 = param.asInt();
}
```

Hàm khởi tạo chân ảo V12 trên app blynk để điều khiển auto hay tự điều khiển

```
void toarduino()
{
  v2arduino = v2arduino + pinValue1 + "," + pinValue2 + "," + pinValue3;
  Serial.println(v2arduino);
  delay(100);
  v2arduino = "";
}
```

Hàm gửi các dữ liệu từ node mcu esp8266 sang arduino qua giao tiếp truyền thông UART.

Source code Arduino Uno R3

```
rdata = nodemcu.read();
myString = myString+ rdata;
if( rdata == '\n')
{
Serial.println(myString);
String l = getValue(myString, ',', 0);
String m = getValue(myString, ',', 1);
String n = getValue(myString, ',', 2);
firstVal = l.toInt();
secondVal = m.toInt();
thirdVal = n.toInt();
Serial.println(thirdVal);
myString = "";
```

Hàm đọc các giá trị nhận được từ esp8266 qua giao tiếp UART

```
String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
}
```

```

    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

```

Hàm đồng bộ các giá trị thông qua giao tiếp truyền thông UART

```

void automatic()
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("xe dang auto");
    Serial.println("Xe dang chay auto");
    int distanceRight = 0;
    int distanceLeft = 0;
    delay(50);

```

Xe lúc đầu sẽ đi thẳng đến khi khoảng cách (distance) nhỏ hơn 30 thì lúc này xe sẽ đi lùi và thực hiện so sánh khoảng cách trái phải.

```

if (distance <= 30)
{
    moveStop();
    delay(300);
    moveBackward();
    delay(400);
    moveStop();
    delay(300);
    distanceRight = lookRight();
    delay(300);
    distanceLeft = lookLeft();
    delay(300);

    if (distanceRight >= distanceLeft)
    {

```

```

        turnRight();
        delay(300);
        moveStop();
    }
    else
    {
        turnLeft();
        delay(300);
        moveStop();
    }

}
else
{
    moveForward();
}

distance = readPing();
}

int lookRight()
{
    servo_motor.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
}

int lookLeft()

```

```
{  
    servo_motor.write(180);  
    delay(500);  
    int distance = readPing();  
    delay(100);  
    servo_motor.write(115);  
    return distance;  
}
```

```
int readPing()  
{  
    delay(100);  
    int cm = sonar.ping_cm();  
    if (cm==0)  
    {  
        cm=250;  
    }  
    return cm;  
}
```

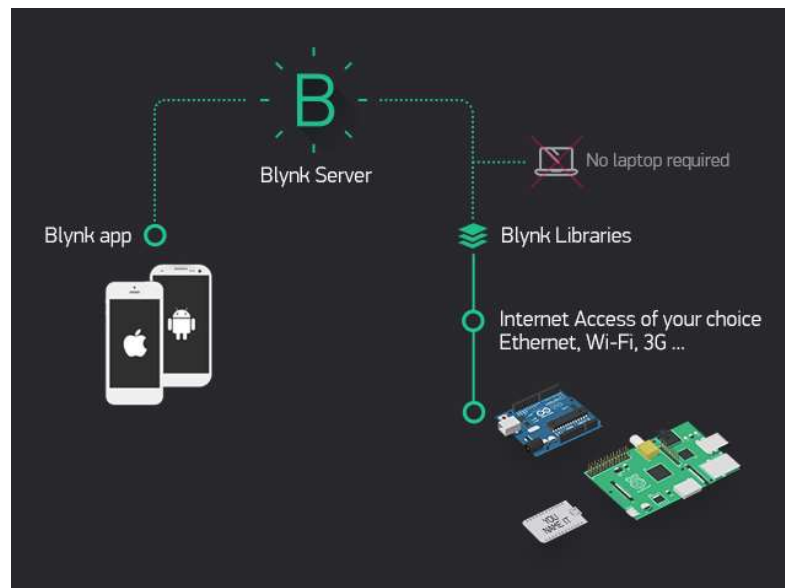
2.4 Giám sát điều khiển mô hình hệ thống thông qua điện thoại thông minh.

2.4.1 Giới thiệu sơ nét về app blynk.

Ứng dụng Blynk được thiết kế cho IOT (Internet Of Things). Nó có thể điều khiển phần cứng từ xa, hiển thị dữ liệu cảm biến, lưu trữ dữ liệu, trực quan hóa nó và làm nhiều việc hay ho khác.

Có 3 thành phần chính trong nền tảng:

- Ứng dụng Blynk cho phép nhà phát triển tạo giao diện cho các sự án của họ bằng các vật dụng khác nhau mà ứng dụng cung cấp.
- Blynk Server chịu trách nhiệm cho tất cả các giao tiếp giữa điện thoại thông minh và phần cứng. Bạn có thể sử dụng Blynk Cloud của nhà phát triển hoặc chạy máy chủ Blynk riêng của mình. Nó có nguồn mở, có thể dễ dàng xử lý hàng ngàn thiết bị và có thể khởi chạy trên Raspberry Pi.
- Thư viện Blynk sử dụng cho tất cả các nền tảng phần cứng phổ biến, cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và đi.

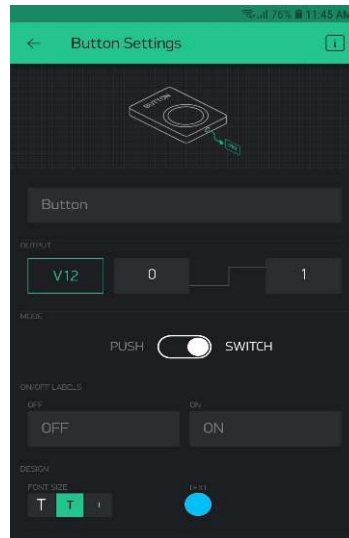


Hình 17 Giới thiệu về app Blynk

2.4.2 Thiết kế giao diện app blynk trên điện thoại.

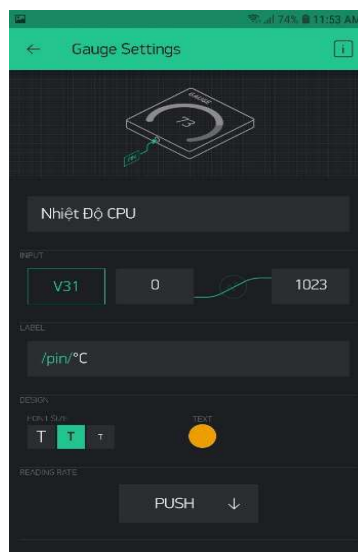
Các widgets được sử dụng trong dự án

- Nút bấm: hoạt động ở chế độ đẩy hoặc chuyển đổi. Cho phép gửi các giá trị BẬT/TẮT (CAO/THẤP). Nút gửi CAO khi nhấn và THẤP khi phát ra.



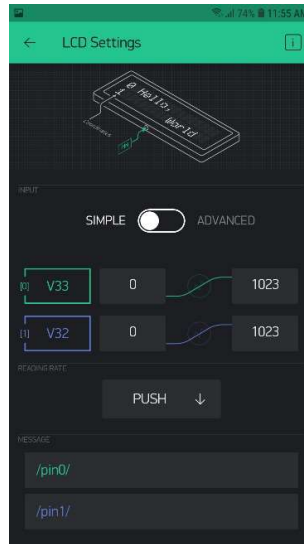
Hình 18 Bảng cài đặt nút bấm.

- Gauge: hiển thị trực quan các giá trị số đếm



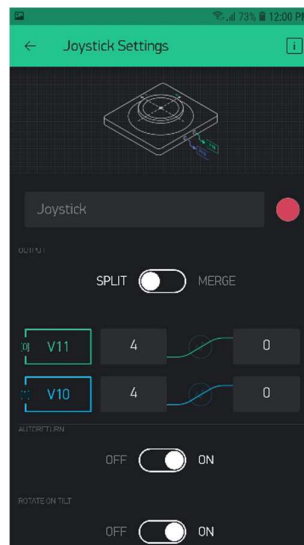
Hình 19 Bảng cài đặt Gauge.

- LCD 16x2 hoạt động ở 3 chế độ:
 - Đơn giản: trong chế độ này tiện ích LCD hoạt động như tiện ích thông thường với tần số đọc.
 - Advanced: chế độ nâng cao cho người dùng có kinh nghiệm. Cho phép sử dụng các lệnh điều khiển LCD.
 - SuperChart: được sử dụng để trực quan hóa dữ liệu trực tiếp và lịch sử. Người dùng có thể sử dụng nó cho dữ liệu cảm biến.



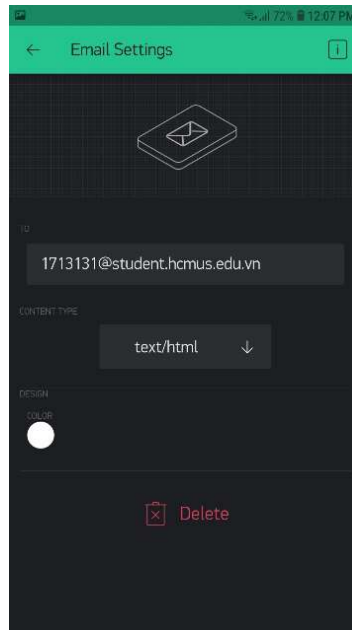
Hình 20 Bảng cài đặt LCD.

- Joystick: điều khiển chuyển động theo 4 hướng.



Hình 21 Bảng cài đặt joystick.

- Email: cho phép gửi email từ phần cứng đến mail của người dùng.

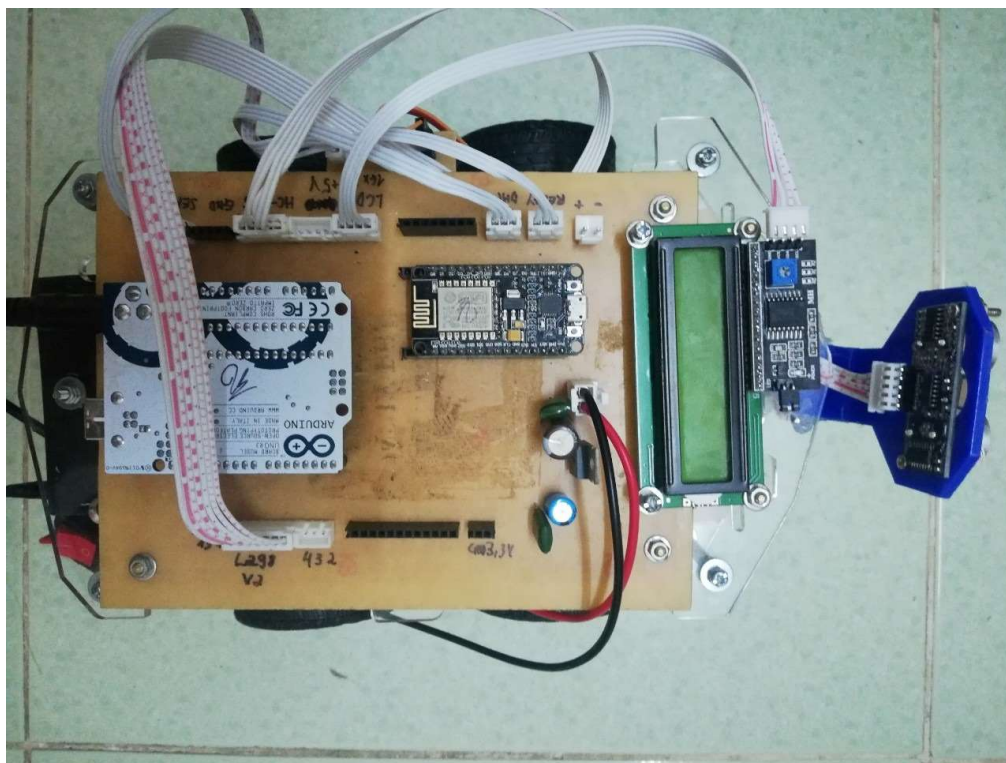


Hình 22 Bảng cài đặt email.



Hình 23 Giao diện app Blynk thiết kế theo đề tài.

2.5 Mô hình demo.



Hình 24 Mô hình demo xe tránh vật cản.

CHƯƠNG 3: KIỂM TRA HOẠT ĐỘNG CỦA HỆ THỐNG.

3.1 Các bước kiểm tra hoạt động của hệ thống.

Chế độ tự điều khiển:

Với chế độ tự điều khiển: khi ta bật nguồn điện cung cấp cho hệ thống, lúc này Node MCU ESP8266 sẽ giao tiếp với app Blynk. Ta chỉ cần di chuyển joystick theo các hướng khác nhau thì xe sẽ chạy theo các hướng chỉ định.

Chế độ auto:

Với chế độ auto: xe sẽ tự chạy khi có vật cản cách 30cm thì xe sẽ dừng lại và lùi lại 5cm. Sau đó, servo sẽ bắt đầu xoay các góc trái phải đồng thời cảm biến siêu âm cũng hoạt động để đo khoảng cách. Khi khoảng cách bên trái lớn hơn bên phải thì xe sẽ di chuyển qua bên trái và ngược lại.

Trong cả hai quá trình trên: cảm biến nhiệt độ luôn hoạt động để đo nhiệt độ của khối điều khiển động cơ và gửi dữ liệu lên app Blynk. Khi nhiệt độ lớn hơn 32 độ C, lúc này app Blynk sẽ gửi thông báo cho người dùng qua email đã cài đặt trước và kích relay điều khiển quạt tản nhiệt hoạt động.

3.2 Đánh giá hoạt động của hệ thống.

- Hệ thống đã hoạt động đúng theo yêu cầu.
- Nhược điểm của hệ thống:
- Trong quá trình chuyển đổi giữa 2 chế độ và quá trình thực hiện các chức năng trên app blynk: hệ thống sẽ bị delay.
- Nguyên nhân xuất hiện delay:
- Do tốc độ mạng của wifi.
- Do thời gian hoàn thành một chu trình của hệ thống.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.

4.1 Kết quả đạt được.

- Hoàn thành được xe tránh vật cản bằng 2 chế độ auto và điều khiển bằng tay qua app blynk.
- Hiểu về giao tiếp truyền thông UART, I2C và giao tiếp giữa node mcu esp8266 với app blynk và các cảm biến.
- Biết cách thiết kế mạch và layout mạch trên phần mềm Proteus và Altium.
- Hiểu về quy trình làm mạch in thủ công.

4.2 Đề xuất hướng phát triển.

- Mở rộng và phát triển xe thêm nhiều chức năng như: Tự động tránh vật cản, nhận biết được màu sắc, đi theo chuyển động của vật...
- Truyền dữ liệu từ arduino lên app blynk thông qua node mcu esp8266 bằng giao tiếp truyền thông UART.

PHỤ LỤC

Source code Node MCU Esp8266

```
void myTimerEvent()
{
    Blynk.virtualWrite(V1, millis() / 1000);
}

void loop()
{
    if (Serial.available() == 0 )
    {
        Blynk.run();
        timer.run(); // Initiates BlynkTimer
        toarduino();
    }
    if (Serial.available() > 0 )
    {
        rdata = Serial.read();
        myString = myString + rdata;
        if ( rdata == '\n')
        {
            String l = getValue(myString, ',', 0);
            String m = getValue(myString, ',', 1);
            String n = getValue(myString, ',', 2);
            firstVal = l.toInt();
            secondVal = m.toInt();
            thirdVal = n.toInt();
            myString = "";
        }
    }
}
```

```

void sensorvalue1()
{
    int sdata = firstVal;
    Blynk.virtualWrite(V2, sdata);
}

void sensorvalue2()
{
    int sdata = secondVal;
    Blynk.virtualWrite(V3, sdata);
}

void sensorvalue3()
{
    int sdata = thirdVal;
    Blynk.virtualWrite(V4, sdata);
    float doam = dht.getHumidity();
    float nhietdo = dht.getTemperature();
    Serial.print("do am trong khong khi la: ");
    Serial.print(doam);
    Serial.println(" %");
    Serial.print("nhiet do moi truong la: ");
    Serial.print(nhietdo);
    Serial.println(" 'C");
    Blynk.virtualWrite(V30, doam);
    Blynk.virtualWrite(V31, nhietdo);
    if(nhietdo < 32){
        Blynk.virtualWrite(V32, "Nhiet do BT");
        Blynk.virtualWrite(V33, "quat khong chay");
        digitalWrite(D6,LOW);
    } else {
        Blynk.email("1713131@student.hcmus.edu.vn", "nhiet do dang qua cao. Can
kiem tra lai CPU");
    }
}

```

```

    Blynk.virtualWrite(V32, "Nhiet do cao");
    Blynk.virtualWrite(V33, "quat dang chay");
    digitalWrite(D6,HIGH);
  }
}

String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

BLYNK_WRITE(V10)
{
    pinValue1 = param.asInt();
}

BLYNK_WRITE(V11)
{
    pinValue2 = param.asInt();
}

BLYNK_WRITE(V12)
{
    pinValue3 = param.asInt();
}

```

```
void toarduino()
{
    v2arduino = v2arduino + pinValue1 + "," + pinValue2 + "," + pinValue3;
    Serial.println(v2arduino);
    delay(100);
    v2arduino = "";
}
```


Source code Arduino Uno R3

```
void loop()
{

if(nodemcu.available() == 0 )
{
    delay(100);

}

if ( nodemcu.available() > 0 )
{

    rdata = nodemcu.read();
    myString = myString+ rdata;
    //Serial.print(rdata);
    if( rdata == '\n')
    {
        Serial.println(myString);
        String l = getValue(myString, ',', 0);
        String m = getValue(myString, ',', 1);
        String n = getValue(myString, ',', 2);
        firstVal = l.toInt();
        secondVal = m.toInt();
        thirdVal = n.toInt();
        Serial.println(thirdVal);
        myString = "";

        if ( (firstVal == 2) && (secondVal == 2)&&(thirdVal==0))
        {
            analogWrite(ena,0);
```

```

    analogWrite(enb,0);
    digitalWrite(in1,LOW);
    digitalWrite(in2,LOW);
    digitalWrite(in3,LOW);
    digitalWrite(in3,LOW);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("khong hoat dong");
    Serial.println("khong hoat dong");

} else

if ( (firstVal < 2) && (secondVal == 2)&& (thirdVal==0))
{
    analogWrite(ena,120);
    analogWrite(enb,120);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("xe di thang");
    Serial.println("xe dang di thang");
} else

if ( (firstVal > 2) && (secondVal == 2)&&(thirdVal==0) )
{
    analogWrite(ena,120);
    analogWrite(enb,120);
    digitalWrite(in1,LOW);

```

```

        digitalWrite(in2,HIGH);
        digitalWrite(in3,LOW);
        digitalWrite(in4,HIGH);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("xe dang lui");
        Serial.println("xe dang di lui");
    }else

if ( (firstVal == 2) && (secondVal < 2)&&(thirdVal==0) )
{
    analogWrite(ena,100);
    analogWrite(enb,80);
    digitalWrite(in1,HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("xe re trai");
    Serial.println("xe dang re trai");
} else

if ( (firstVal == 2) && (secondVal >2)&&(thirdVal==0) )
{
    analogWrite(ena,80);
    analogWrite(enb,100);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,HIGH);

```

```

        digitalWrite(in4,LOW);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("xe re phai");
        Serial.println("xe dang re phai");
    }
    else if(thirdVal==1) {
        automatic();
    }
}
}
}

```

```

String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }

    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void automatic()

```

```

{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("xe dang auto");
  Serial.println("Xe dang chay auto");
  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  if (distance <= 30)
  {
    moveStop();
    delay(300);
    moveBackward();
    delay(400);
    moveStop();
    delay(300);
    distanceRight = lookRight();
    delay(300);
    distanceLeft = lookLeft();
    delay(300);

    if (distanceRight >= distanceLeft)
    {
      turnRight();
      delay(300);
      moveStop();
    }
    else
    {
      turnLeft();
    }
  }
}

```

```

        delay(300);
        moveStop();
    }

}
else
{
    moveForward();
}

    distance = readPing();
}

int lookRight()
{
    servo_motor.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
}

int lookLeft()
{
    servo_motor.write(180);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
}

```

```

}

int readPing()
{
    delay(100);
    int cm = sonar.ping_cm();
    if (cm==0)
    {
        cm=250;
    }
    return cm;
}

void moveStop()
{
    analogWrite(ena,0);
    analogWrite(enb,0);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("xe dang auto");
    lcd.setCursor(0,1);
    lcd.print("khong hoat dong");
    Serial.println("Xe dang chay auto");
    Serial.println("Xe dang ngung hoat dong");

}

```

```

void moveForward()
{
    analogWrite(ena,90);
    analogWrite(enb,90);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("xe dang auto");
    lcd.setCursor(0,1);
    lcd.print("xe di thang");
    Serial.println("Xe dang chay auto");
    Serial.println("Xe dang di thang");
}

```

```

void moveBackward()
{
    analogWrite(ena,90);
    analogWrite(enb,90);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("xe dang auto");
    lcd.setCursor(0,1);
    lcd.print("xe re trai");
    Serial.println("Xe dang chay auto");
}

```



```

    Serial.println("Xe dang re trai");
}

void turnRight()
{
    analogWrite(ena,90);
    analogWrite(enb,90);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("xe dang auto");
    lcd.setCursor(0,1);
    lcd.print("xe re phai");
    Serial.println("Xe dang chay auto");
    Serial.println("Xe dang re phai");
}

void turnLeft()
{
    analogWrite(ena,90);
    analogWrite(enb,90);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
}

```

TÀI LIỆU THAM KHẢO

- [1] <https://www.arduino.cc/>
- [2] <http://www.circuitbasics.com/basics-uart-communication/>
- [3] <https://ns-electric.com/knowledge-base/intro-to-arduino-uart-serial-communication/>
- [4] [https://www.researchgate.net/publication/308988751 Universal Asynchronous Receiver and Transmitter UART](https://www.researchgate.net/publication/308988751_Universal_Asynchronous_Receiver_and_Transmitter_UART)