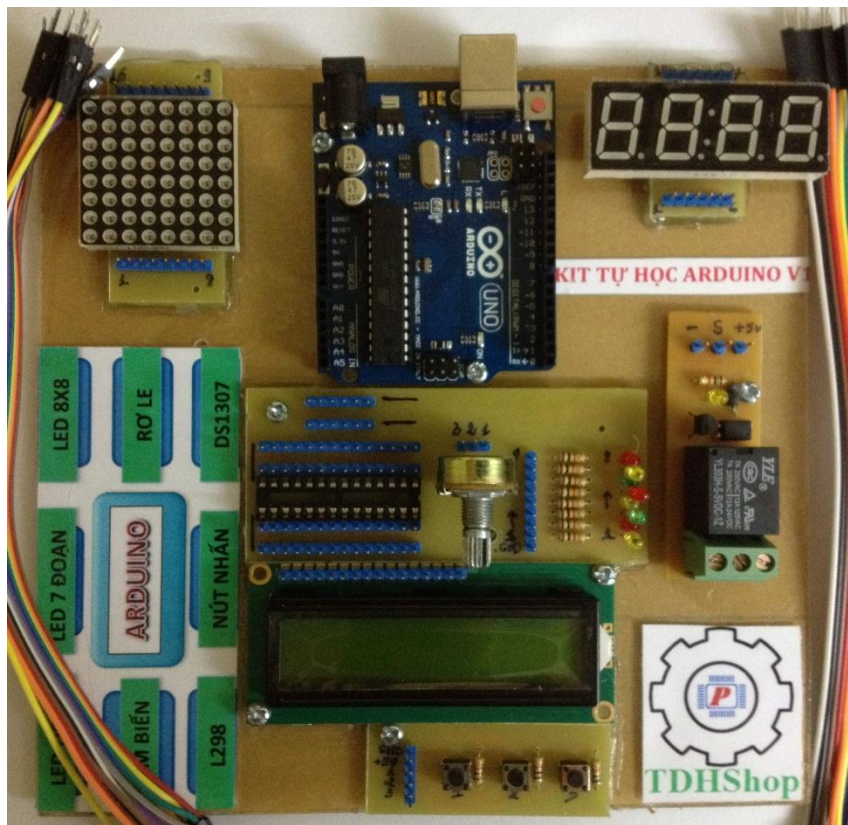


SỔ TAY ARDUINO



LẬP TRÌNH ĐIỀU KHIỂN ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU V2



SỔ TAY ARDUINO

HỌC VIẾT CHƯƠNG TRÌNH CHO ARDUINO THÔNG QUA CÁC LOẠI CẢM BIẾN CƠ BẢN

- 1. CẢM BIẾN ÁNH SÁNG DÙNG QUANG TRỞ.**
- 2. CẢM BIẾN BÁO CHÁY.**
- 3. CẢM BIẾN DÒ LINE.**
- 4. CẢM BIẾN MƯA.**
- 5. CẢM BIẾN SIÊU ÂM (KHOẢNG CÁCH SRF04).**
- 6. CẢM BIẾN ÂM THANH.**
- 7. CẢM BIẾN ĐỘ ẨM ĐẤT.**
- 8. CẢM BIẾN ĐO ĐỘ ẨM VÀ NHIỆT ĐỘ DTH11.**
- 9. CẢM BIẾN NHIỆT ĐỘ (LM35).**
- 10. CẢM BIẾN KHÍ GAS (MQ2).**
- 11. CẢM BIẾN NHIỆT ĐỘ (DS18B20).**
- 12. CẢM BIẾN CHUYỂN ĐỘNG PIR.**
- 13. CẢM BIẾN MÀU (TCS3200).**
- 14. CẢM BIẾN GIA TỐC (MPU 6050).**

SỔ TAY ARDUINO

**TRONG QUÁ TRÌNH VIẾT EBOOK CÓ GÌ MONG CÁC
BẠN THÔNG CẢM!!!! THAY MẶT CHO tdhshop.com.vn
CẢM ƠN ĐẾN TẤT CẢ MỌI NGƯỜI ĐÃ TIN TƯỞNG
SHOP.**

SỔ TAY ARDUINO

1. CẢM BIẾN ÁNH SÁNG DÙNG QUANG TRỞ.

Giới thiệu và nội dung cần nắm:

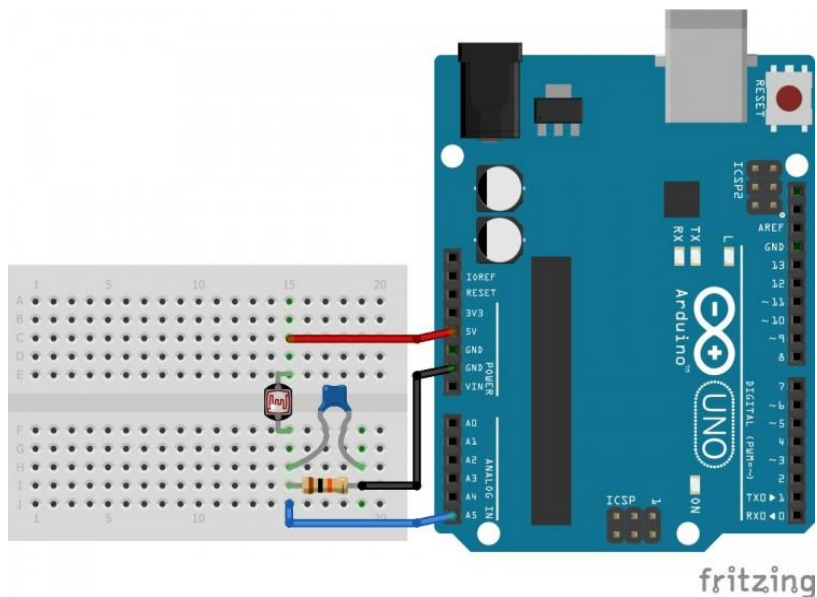
Quang trở là một loại "vật liệu" điện tử rất hay gặp và được sử dụng trong những mạch cảm biến ánh sáng. Có thể hiểu một cách dễ dàng rằng, quang trở là một loại ĐIỆN TRỞ có điện trở thay đổi theo cường độ ánh sáng. Nếu đặt ở môi trường có ít ánh sáng, có bóng râm hoặc tối thì điện trở của quang trở sẽ tăng cao còn nếu đặt ở ngoài nắng, hoặc nơi có ánh sáng thì điện trở sẽ giảm.

Phần cứng:

- Arduino Uno
- 1 điện trở 560 Ohm (hoặc 220 Ohm hoặc 1kOhm)
- 1 đèn LED siêu sáng
- 1 biến trở 10K hoặc 1K
- Breadboard
- Dây cắm breadboard
- 1 điện trở 10 kOhm
- 1 tụ điện 100nF

Làm thế nào để đọc tín hiệu từ quang trở?

Phương pháp có tên gọi là đọc hiệu điện thế từ cầu phân áp (gợi ý, quang trở bạn cứ xem là điện trở R1, vào phía trước mỗi chân Analog của Arduino có một điện trở, và xem đó là điện trở R2 = 10k). Bây giờ, chúng ta xem cách mắc thôi.



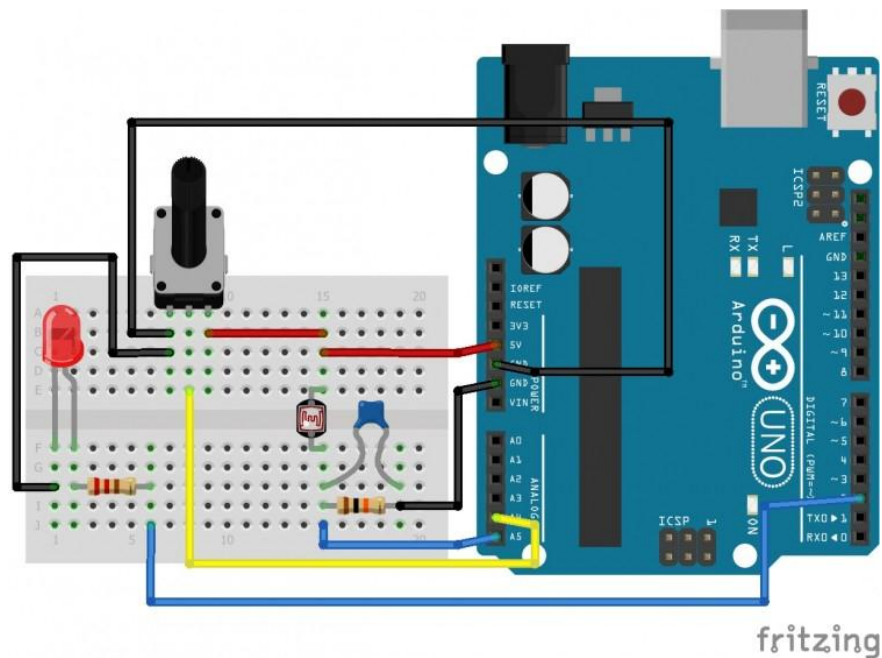
SỔ TAY ARDUINO

Bây giờ bạn upload code này lên và xem giá trị của quang trở, nhớ là hãy thay đổi độ sáng của môi trường để xem sự thay đổi của quang trở (chẳng hạn bạn lấy bàn tay che ánh sáng chiếu trực tiếp vào quang trở chẳng hạn).

```
int quangtro = A5; //Thiết đặt chân analog đọc quang trở
void setup() {
  // Khởi tạo cổng Serial 9600
  Serial.begin(9600);
}
void loop() {
  int giatriQuangtro = analogRead(quangtro); // đọc giá trị quang trở

  Serial.println(giatriQuangtro); // Xuất giá trị ra Serial Monitor
}
```

Bây giờ ta sẽ dùng biến trở để xây dựng một mức (ta có thể thay đổi được) và nếu giá trị của quang trở bé hơn mức này thì đèn sáng và ngược lại thì đèn tắt. Bạn hãy lắp mạch như sau: (trong quá trình lắp mạch không cần gắn tụ cũng được).



- Đèn led được nối với chân digital 2
- Biến trở được nối tới chân analog 4

SỔ TAY ARDUINO

Bây giờ bạn upload đoạn code sau và điều chỉnh biến trở để đèn LED sáng hoặc tắt:

```
int quangtro = A5; //Thiết đặt chân analog đọc quang trở
int biento = A4; // Thiết đặt chân analog đọc biến trở
int led = 2; // Thiết đặt chân digital là đèn led
void setup() {
  // Khởi tạo cổng Serial 9600
  Serial.begin(9600);
  //pinMode
  pinMode(led,OUTPUT);
}

void loop() {
  int giatriQuangtro = analogRead(quangtro); // đọc giá trị quang trở
  int giatriBiento = analogRead(biento); // đọc giá trị biến trở

  // Nếu giá trị quang trở <= giá trị biến trở thì bật đèn
  if (giatriQuangtro <= giatriBiento) {
    digitalWrite(led,HIGH);
  } else {
    digitalWrite(led,LOW);
  }
  // Xuất giá trị ra Serial Monitor
  Serial.print("Quang tro ");
  Serial.print(giatriQuangtro);
  Serial.print(", bien tro ");
  Serial.print(giatriBiento);
  Serial.print(" => ");
  // Dòng này tương đương đoạn code sau
  /*
  if (giatriQuangtro <= giatriBiento) {
    Serial.println("Sang");
  } else {
    Serial.println("Tat");
  }
  */
}
```

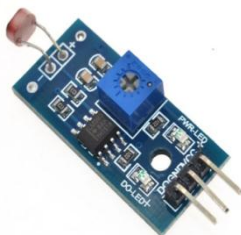

SỔ TAY ARDUINO

*/

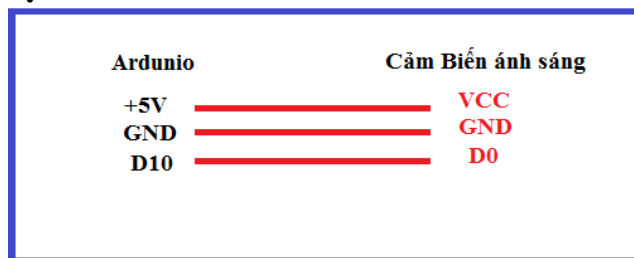
```
Serial.println((giatriQuangtro <= giatriBientro) ? "Sang" : "Tat");
```

```
}
```

Và trên thị trường hiện nay đã có sẵn module cảm biến ánh sáng:



Sơ đồ nối dây trên kit tự học:



Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO

Chương trình:

```
int cambien = 10; // khai báo chân digital 10 cho cảm biến

int Led = 8; //kháo báo chân digital 8 cho đèn LED

void setup () {

  pinMode(Led, OUTPUT); //pinMode xuất tín hiệu đầu ra cho led

  pinMode(cambien, INPUT); //pinMode nhận tín hiệu đầu vào cho cảm biê

}

void loop () {

  int value = digitalRead(cambien); //lưu giá trị cảm biến vào biến value

  digitalWrite(Led, value); //xuất giá trị ra đèn LED

}
```

2. CẢM BIẾN BÁO CHÁY.



Cảm biến phát hiện lửa

Đây là loại cảm biến chuyên dùng để phát hiện lửa, thường được sử dụng trong hệ thống báo cháy. Tầm phát hiện trong khoảng 80cm, góc quét là 60 độ. Có thể phát hiện lửa tốt nhất là loại có bước sóng từ 760nm - 1100nm. Mạch được tích hợp IC LM393 so sánh để tạo mức tín hiệu => ta có thể chỉnh độ nhạy bằng biến trở.

Thông số kỹ thuật:

Web: tdhshop.com.vn - **Chuyên Kit TỰ HỌC ARDUINO**

SỔ TAY ARDUINO

Nguồn : 3.3V - 5V, 15mA

Điện áp ra : 3.3 - 5V, có cả analog và Digital.

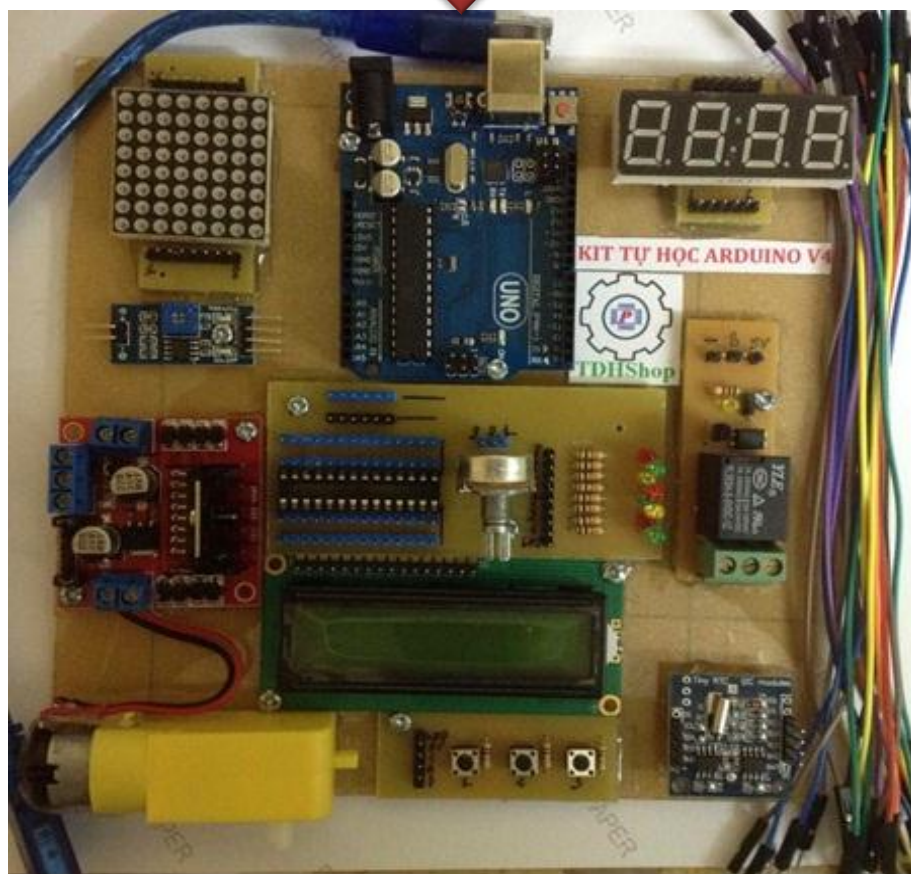
Khoảng cách : 80 cm

Góc quét : 60 độ

Kích thước : 3.2 x 1.4 cm

Sơ đồ nối dây trên kit tự học:

Arduino		CB báo cháy
+5V	—	VCC
GND	—	GND
A1	—	A0
D4	—	D0



SỔ TAY ARDUINO

Chương trình :

//CHƯƠNG TRÌNH CHO CẢM BIẾN BÁO CHÁY

//Cách nối dây: chân G nối GND, chân V+ nối nguồn 5V

//Chân AO (Analog Output-tín hiệu ra Analog) nối chân A1

//Chân DO (Digital Output - tín hiệu ra on/off) nối chân 4

#define ChanAO 1//Chân A1 là chân để đọc tín hiệu Analog AO

#define ChanDO 4//Chân 4 (bên digital) là chân để đọc tín hiệu Digital DO

#define baochay 12//Chân để điều khiển đèn led báo cháy

float giatriAO;//Biến chứa giá trị Analog AO

float giatriDO;//Biến chứa giá trị Digital DO

void setup()

{

Serial.begin(9600);//Giao tiếp với máy tính

pinMode(ChanDO, INPUT);//Định nghĩa chanDO (chân 4) là chân đọc tín hiệu vào

pinMode(baochay, OUTPUT);//Định nghĩa chân báo cháy (chân 12) là chân xuất tín hiệu ra cho đèn led

}

void loop()

{

giatriAO=analogRead(ChanAO);//Đọc giá trị Analog từ ChanAO

giatriDO=digitalRead(ChanDO);//Đọc giá trị digital từ ChanDO

Serial.print("Gia tri Chan Analog: ");//In giá trị Analog ra màn hình

Serial.print(giatriAO);

Serial.print(" Gia tri chan Digital: ");//In giá trị digital ra màn hình

Serial.println(giatriDO);

SỔ TAY ARDUINO

```
if(giatriAO<900)//Điều kiện để phát tín hiệu cháy (do người dùng tự đặt)
{
    Serial.println("Cháy nha");//Báo cháy
    digitalWrite(baochay,HIGH);//Bật đèn sáng
}
else//Ngược lại
{
    digitalWrite(baochay,LOW);//Tắt đèn
}
delay(1000);//Cập nhật giá trị sau 1s
}
```

SỔ TAY ARDUINO

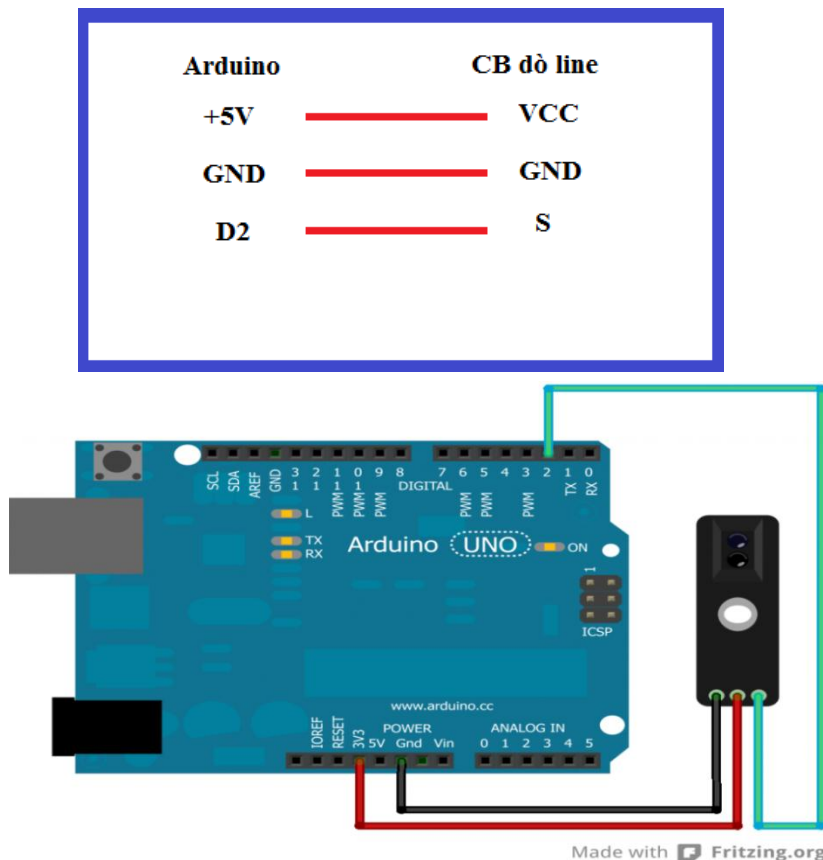
3. CẢM BIẾN DÒ LINE.



- Power supply: +5V
- Operating current: <10mA
- Operating temperature range: 0°C ~ + 50°C
- Output interface: 3-wire interface (1 - signal, 2 - power, 3 - power supply negative)
- Output Level: TTL level (black line of low effective, high efficient white line)
- Module Size: 10mm×35mm
- Module Weight: About 1g

Loại 3 chân : VCC, GND và D0 (S)

KẾT NỐI VỚI ARDUINO:



Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO

Mã code:

```
// CHƯƠNG TRÌNH THỬ ĐẶC TÍNH CỦA CẢM BIẾN DÒ ĐƯỜNG

#define den 13 // Sử dụng chân 13 để điều khiển đèn led

#define tinhieu 2 //Chân 5 của cảm biến nối với chân 2 trên mạch Arduino

int a;//Tên biến chứa giá trị của cảm biến dò đường

void setup()

{

    Serial.begin(9600);//Đưa dữ liệu lên máy tính để quan sát

    pinMode(den,OUTPUT);//Định nghĩa chân ra để điều khiển đèn led

    pinMode(tinhieu,INPUT);//Định nghĩa chân 2 là chân lấy tín hiệu vào của cảm biến

}

void loop()

{

    a=digitalRead(tinhieu);

    Serial.println(a);

    if(a==HIGH)//nếu a=1 (gặp màu đen) thì đèn tắt

    {

        digitalWrite(den,LOW);

    }

    if(a==LOW)//Nếu a=0 (gặp màu trắng) thì đèn sáng

    {

        digitalWrite(den,HIGH);

    }

    delay(1000);//Chờ 1 giây để hệ thống đáp ứng

}
```

SỔ TAY ARDUINO

4. CẢM BIẾN MƯA.

Chúng ta có thể dễ dàng nhận biết mưa bằng mắt thường hoặc cảm nhận ở da. Với tư tưởng ấy, các hệ thống điện tử phát hiện mưa cũng chia ra làm hai loại: thứ nhất là dùng camera để nhận biết và loại thứ hai là dùng cảm biến (tương tự da của con người). Trong môi trường Arduino, bạn có thể dùng cả 2 cách trên để phát hiện mưa. Tuy nhiên, để tiết kiệm chi phí và dễ dàng trong việc lập trình, cài đặt, người ta thường chọn phương pháp thứ hai.

Vấn đề về phát hiện mưa thì cực kì đơn giản, nhưng để truyền dữ liệu cảm biến mưa về trung tâm thì khó hơn đôi chút. Trong phạm vi bài viết này, mình chỉ đề cập đến phần làm thế nào để đọc cảm biến mưa thôi.



Mạch cảm biến mưa gồm 2 phần:

- Mạch cảm biến mưa được gắn ngoài trời
- Mạch điều chỉnh độ nhạy cần được che chắn

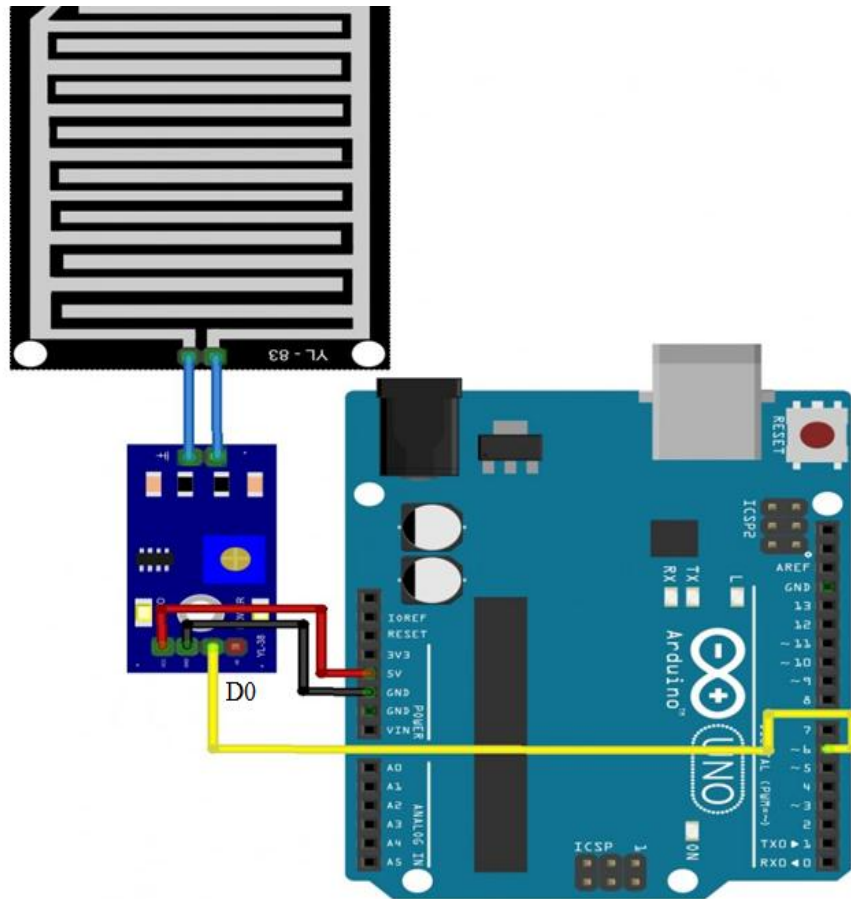
Mạch cảm biến mưa hoạt động bằng cách so sánh hiệu điện thế của mạch cảm biến nằm ngoài trời với giá trị định trước (giá trị này thay đổi được thông qua 1 biến trở màu xanh) từ đó phát ra tín hiệu đóng / ngắt rơ le qua chân D0(digital) hoặc AO(analog). Vì vậy, chúng ta dùng một chân digital hoặc analog để đọc tín hiệu từ cảm biến mưa.

SỔ TAY ARDUINO

Khi cảm biến khô ráo (trời không mưa), chân D0 của module cảm biến sẽ được giữ ở mức cao (5V). Khi có nước trên bề mặt cảm biến (trời mưa), đèn **LED** màu đỏ sẽ sáng lên, chân D0 được kéo xuống thấp (0V).

Bài tập ví dụ kèm theo :

Cách đấu nối với mạch arduino:



Sơ đồ chân nối:

Cảm biến mưa	Arduino Uno
GND	GND
VCC	5V
D0	Digital 6

SỔ TAY ARDUINO

Lập trình và giải thích:

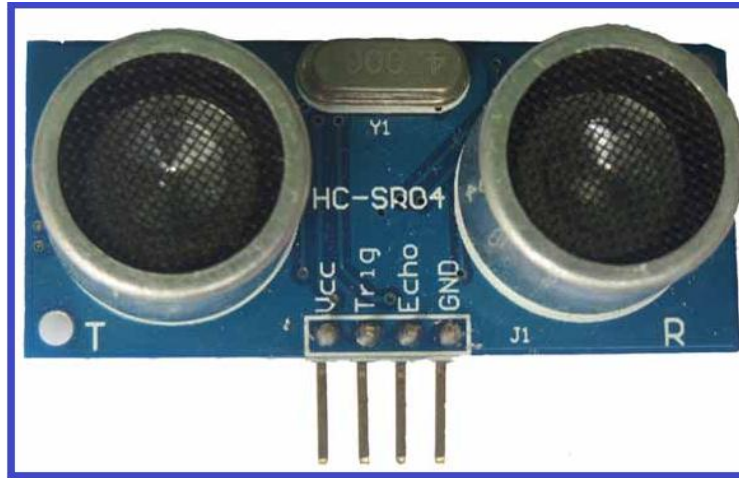
```
int rainSensor = 6; // Chân tín hiệu cảm biến mưa ở chân digital 6 (arduino)
void setup() {
  pinMode(rainSensor,INPUT);// Đặt chân cảm biến mưa là INPUT, vì tín hiệu sẽ được truyền
đến cho Arduino
  Serial.begin(9600);// Khởi động Serial ở baudrate 9600
  Serial.println("Da khoi dong xong");
}

void loop() {
  int value = digitalRead(rainSensor);//Đọc tín hiệu cảm biến mưa
  if (value == HIGH) { // Cảm biến đang không mưa
    Serial.println("Dang khong mua");
  }
  else
  {
    Serial.println("Dang mua");
  }
  delay(1000); // Đợi 1 tí cho lần kiểm tra tiếp theo.
}
```

SỔ TAY ARDUINO

5. CẢM BIẾN SIÊU ÂM.

Sử dụng cảm biến khoảng cách HC-SR04:



Tìm hiểu nguyên lý hoạt động của HC-SR04.
cách sử dụng với Arduino.

PHẦN CỨNG

- Kit tự học arduino (Arduino UNO)
- Breadboard
- Dây cắm breadboard
- 1 cảm biến siêu âm HC-SR04

GIỚI THIỆU

Cảm biến khoảng cách siêu âm HC-SR04 được sử dụng rất phổ biến để xác định khoảng cách vì RẺ và CHÍNH XÁC. Cảm biến sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 -> 300 cm, với độ chính xác gần như chỉ phụ thuộc vào cách lập trình.

Sơ đồ chân của HC-SR04 gồm: VCC, trig, echo, GND. Trong đó:

VCC --> pin 5V Arduino.

trig --> chân digital (OUTPUT), đây là chân sẽ phát tín hiệu từ cảm biến.

echo --> chân digital (INPUT), đây là chân sẽ nhận lại tín hiệu được phản xạ từ vật cản

GND ---> GND Arduino.

Nguyên lý hoạt động

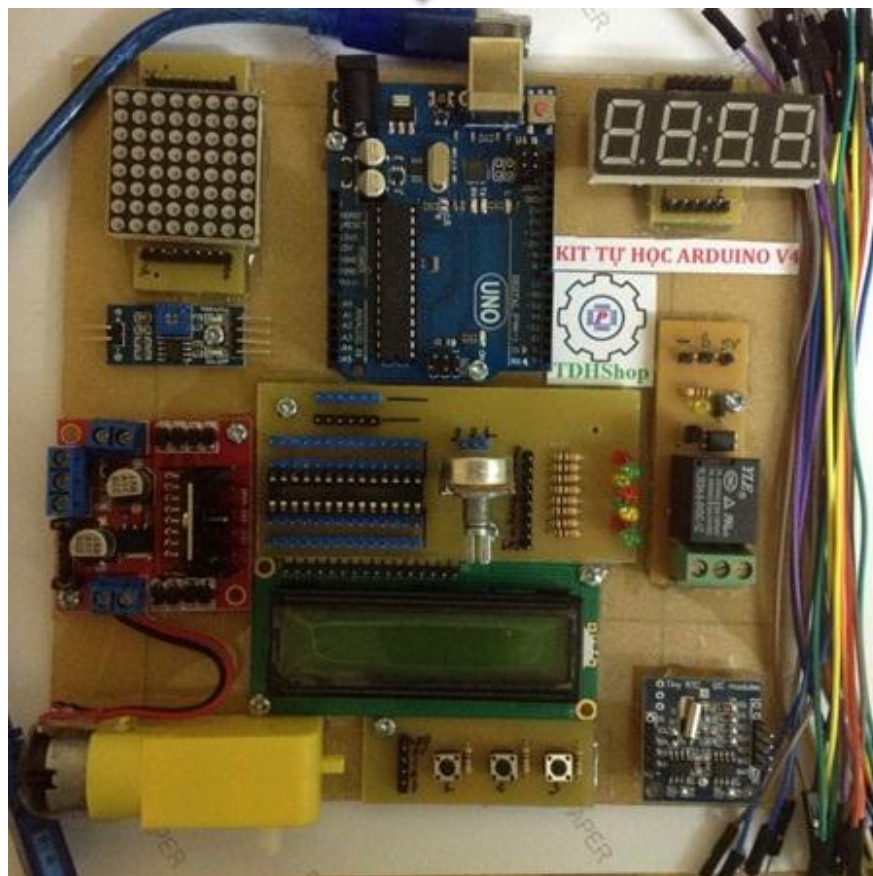
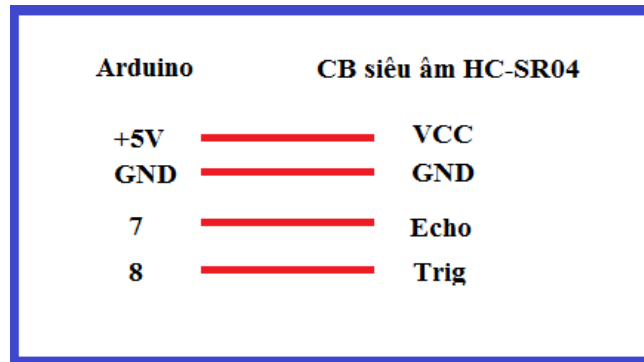
Để đo khoảng cách, ta sẽ phát 1 xung rất ngắn (5 microseconds - ú) từ chân **trig**. Sau đó, cảm biến sẽ tạo ra 1 xung HIGH ở chân **echo** cho đến khi nhận

SỞ TAY ARDUINO

lại được sóng phản xạ ở pin này. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và quay trở lại.

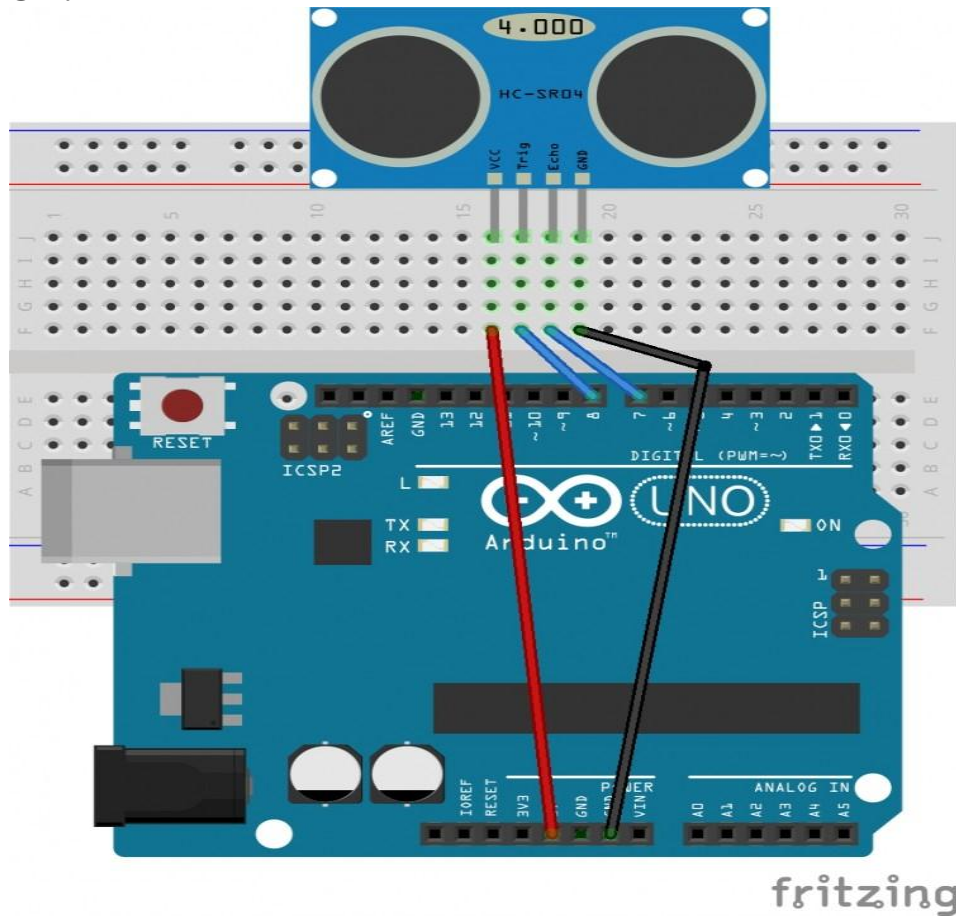
Tốc độ của âm thanh trong không khí là 340 m/s (hằng số vật lý), tương đương với 29,412 microSeconds/cm ($10^6 / (340 \cdot 100)$). Khi đã tính được thời gian, ta sẽ chia cho 29,412 để nhận được khoảng cách.

SƠ ĐỒ NỐI DÂY:



SỔ TAY ARDUINO

LẮP MẠCH:



LẬP TRÌNH

```
const int trig = 8;//chân trig của HC-SR04
const int echo = 7;//chân echo của HC-SR04
void setup()
{
  Serial.begin(9600);//giao tiếp Serial với baudrate 9600
  pinMode(trig,OUTPUT);//chân trig sẽ phát tín hiệu
  pinMode(echo,INPUT);//chân echo sẽ nhận tín hiệu
}
void loop()
{
  unsigned long duration;//biến đo thời gian
  int distance;//biến lưu khoảng cách

  /* phát xung từ chân trig */
  digitalWrite(trig,0);//tắt chân trig
```

SỔ TAY ARDUINO

```
delayMicroseconds(2);  
digitalWrite(trig,1);// phát xung từ chân trig  
delayMicroseconds(5);// xung có độ dài 5 microSeconds  
digitalWrite(trig,0);//tắt chân trig  
  
/*tính toán thời gian*/  
duration = pulseIn(echo,HIGH);//đo độ rộng xung HIGH ở chân echo.  
distance = int(duration/2/29.412);//tính khoảng cách đến vật.  
  
/*in kết quả ra Serial monitor*/  
Serial.print(distance);  
Serial.println("cm");  
delay(200);  
}
```

GIẢI THÍCH

- duration = pulseIn(echo,1);

hàm pulseIn() được dùng để đo độ rộng của xung, các bạn có thể xem thêm tại link

- distance = int(duration/2/29.412);

Thời gian sóng truyền từ cảm biến đến vật sẽ bằng duration/2, sau đó ta chia tiếp cho 29,412 để tính khoảng cách.

SỔ TAY ARDUINO

6. CẢM BIẾN ÂM THANH.



Bonggong

Cảm biến âm thanh

Cảm biến âm thanh sử dụng microphone và opamp để phát hiện âm thanh, khi cường độ âm thanh vượt qua 1 ngưỡng xác định (thay đổi được bằng biến trở) thì ngõ ra sẽ được kéo xuống mức thấp, đồng thời có led báo hiệu.

Thông số kỹ thuật:

- Nguồn: DC 4-6V
- Kích thước: 38mm * 16mm * 9mm.

VCC	Nguồn 4V đến 6V
GND	Mass
D0	Đầu ra tín hiệu số (mức cao hoặc mức thấp)
A0	Đầu ra tương tự (analog)

SƠ ĐỒ NỐI DÂY:

Arduino		CB âm thanh
+5V	_____	VCC
GND	_____	GND
A1	_____	A0
D4	_____	D0
D12	NỐI VỚI LED	

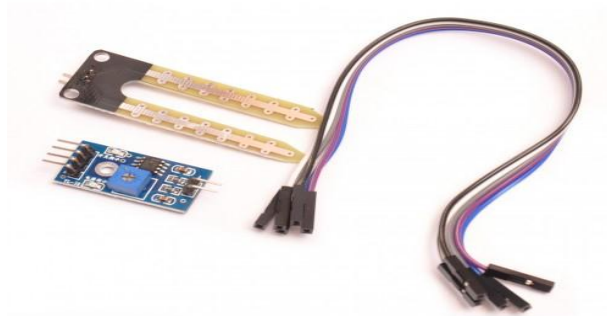
SỔ TAY ARDUINO

Chương trình mẫu:

```
//CHƯƠNG TRÌNH CẢM BIẾN ÂM THANH
//Dây G nối với chân GND, dây + nối với chân 5V
//Dây AO (dây tín hiệu Analog) nối với chân A1
//Dây DO (dây tín hiệu digital) nối với chân 4
#define Analog 1
#define Digital 4
int den=12;
float tinhieuAO;
float tinhieuDO;
void setup()
{
  Serial.begin(9600);
  pinMode(Digital,INPUT);
  pinMode(den, OUTPUT);
}
void loop()
{
  tinhieuAO=analogRead(Analog);
  tinhieuDO=digitalRead(Digital);
  Serial.print("Gia tri Analog: ");
  Serial.print(tinhieuAO);
  Serial.print("      Gia tri Digital: ");
  Serial.println(tinhieuDO);
  if(tinhieuAO>28)
  {
    digitalWrite(den,HIGH);
  }
  else
  {
    digitalWrite(den,LOW);
  }
  delay(50);
}
```


SỔ TAY ARDUINO

7. CẢM BIẾN ĐỘ ẨM ĐẤT



Cảm biến độ ẩm đất:

- Cảm biến phát hiện độ ẩm đất, bình thường đầu ra mức thấp, khi đất thiếu nước đầu ra sẽ mức cao. Module có thể sử dụng để tưới nước tự động
- Độ nhạy của cảm biến độ ẩm đất có thể điều chỉnh được (Bằng cách điều chỉnh biến trở màu xanh trên board mạch)
- Phần đầu dò được cắm vào đất để phát hiện độ ẩm, khi độ ẩm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao

Thông số kỹ thuật:

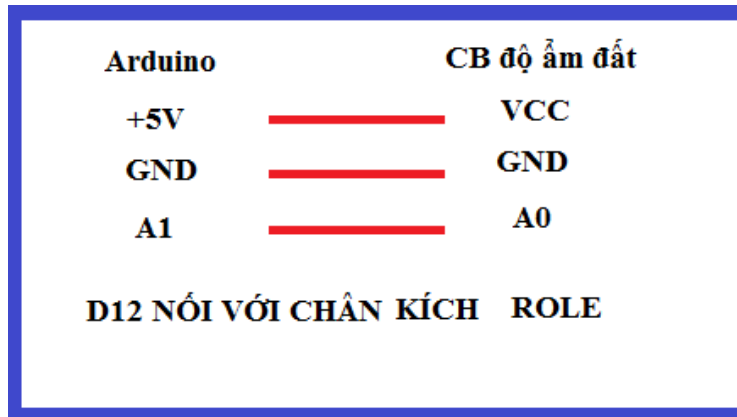
- Điện áp làm việc 3.3V ~ 5V
- Có lỗ cố định để lắp đặt thuận tiện
- PCB có kích thước nhỏ 3.2 x 1.4 cm
- Sử dụng chip LM393 để so sánh, ổn định làm việc

Đầu kết nối sử dụng 3 dây:

VCC	3.3V ~ 5V
GND	GND của nguồn ngoài
DO	Đầu ra tín hiệu số (mức cao hoặc mức thấp)
AO	Đầu ra tín hiệu tương tự (Analog)

SỔ TAY ARDUINO

SƠ ĐỒ NỐI DÂY:



Bạn có thể tham khảo về module role tại: <http://tdhshop.com.vn/module-ro-le-va-cach-su-dung-ro-le>

Code tham khảo:

```
//Chương trình ĐỌC DỮ LIỆU TỪ CẢM BIẾN ĐO ĐỘ ẨM ĐẤT
int CBDAm = 1; //Nối chân ra của cảm biến đo độ ẩm với chân A1
#define role 12 //Chân nối với role 12
float Doamdat; //Biến độ ẩm đất
void setup()
{
  Serial.begin(9600);
  //Khai báo các chân
  pinMode(role, OUTPUT);
}
void loop()
{
  Doamdat = analogRead(CBDAm);
  Serial.print("Do am la:"); //In ra màn hình
  Serial.print(Doamdat); //In ra độ ẩm đất
  //Bơm nước khi đất khô( thông qua role)
  if(Doamdat < 200)
  {
    digitalWrite(role, HIGH); //Bat role
    Serial.println("Tuoi cay");
  }
  //Ngưng bơm khi đất ướt
  else
  {
    digitalWrite(role, LOW);
    Serial.println("Cay du nuoc");
  }
  delay(1000);
}
```

SỔ TAY ARDUINO

8. CẢM BIẾN ĐO ĐỘ ẨM VÀ NHIỆT ĐỘ DHT11.

Phần cứng cần thiết

Màn hình LCD 16 x 2

Mạch điều khiển màn hình LCD sử dụng giao tiếp I2C

Cảm biến nhiệt độ - độ ẩm DHT11

Arduino UNO R3 (hoặc tương đương)

Breadboard

Dây cắm breadboard

Cảm biến DHT11



Cảm biến DHT11 đã được tích hợp trong một mạch duy nhất, bạn chỉ việc nối dây nguồn (Vcc, GND) và dây tín hiệu (Signal) vào mạch Arduino là xong.

Thông số kỹ thuật

- Điện áp hoạt động: 3-5.5V DC
- Ngưỡng độ ẩm: 20 - 90%
- Sai số độ ẩm: $\pm 5\%$
- Ngưỡng nhiệt độ: 0 - 55°C
- Sai số nhiệt độ: $\pm 2^{\circ}\text{C}$

Download và cài đặt thư viện hỗ trợ sử dụng DHT11:

<http://tdhshop.com.vn/tong-hop-cac-thu-vien-cho-arduino>

Sau khi download về các bạn làm theo các bước sau:

1. Copy file nén vào thư viện arduino
2. Từ ổ đĩa : C > Users > Admin > My Documents > Arduino > Libraries

Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO

3. Giải nén file .

4. Trên thanh công cụ của chương trình IDE :

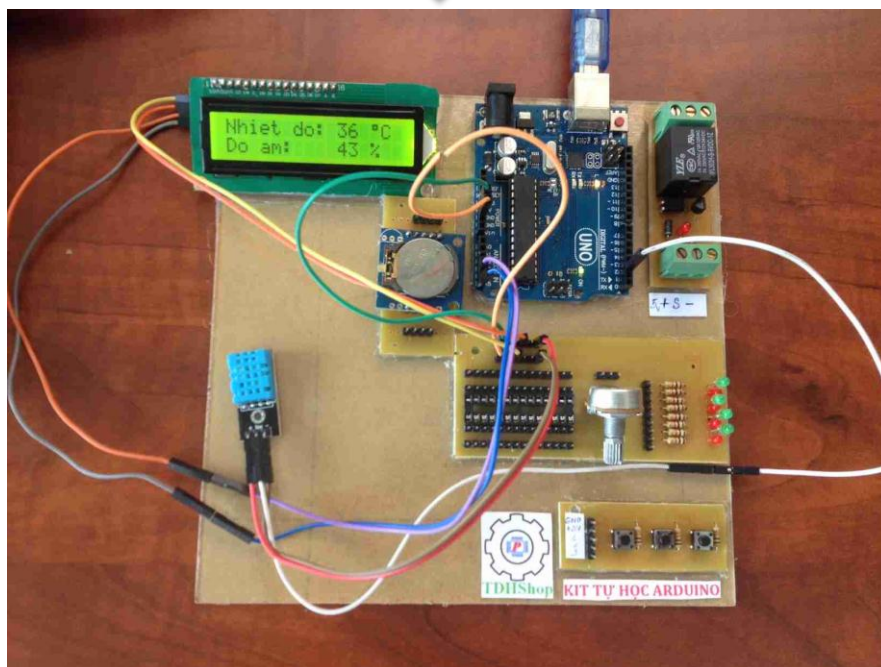
Sketch > Import Library > add Library (và dẫn đến đường dẫn ở trên) > open
(là xong)

Hoặc các bạn vào đây tham khảo nhé cách cài nhé:

<http://tdhshop.com.vn/huong-dan-cai-thu-vien-vao-arduino-ide-ho-tro-cho-viec-lap-trinh>

Kết nối cảm biến DHT11 với mạch Arduino

DHT11	Arduino UNO R3
GND	5V
Vcc	GND
Signal	D2



Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO

Lập trình điều khiển: (chương trình này không có sự tham gia của LCD)

Chúng ta xem dữ liệu trực tiếp trên *Serial port*

```
#include "DHT.h"

const int DHTPIN = 2;    //Đọc dữ liệu từ DHT11 ở chân 2 trên mạch Arduino
const int DHTTYPE = DHT11; //Khai báo loại cảm biến, có 2 loại là DHT11 và DHT22

DHT dht(DHTPIN, DHTTYPE);

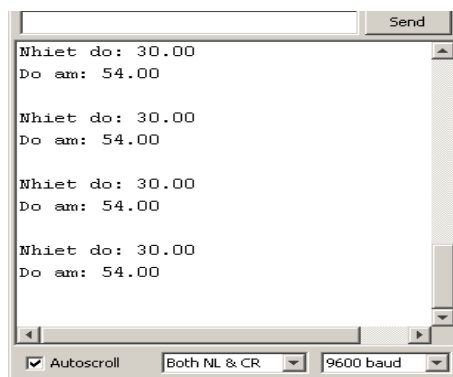
void setup() {
  Serial.begin(9600);
  dht.begin();    // Khởi động cảm biến
}

void loop() {
  float h = dht.readHumidity(); //Đọc độ ẩm
  float t = dht.readTemperature(); //Đọc nhiệt độ

  Serial.print("Nhiệt độ: ");
  Serial.println(t);    //Xuất nhiệt độ
  Serial.print("Độ ẩm: ");
  Serial.println(h);    //Xuất độ ẩm

  Serial.println();    //Xuống hàng
  delay(1000);    //Đợi 1 giây
}
```

Sau khi upload chương trình lên mạch Arduino, bạn hãy bấm Ctrl + Shift + M để mở cửa sổ Serial Monitor và xem kết quả.



Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO

Một lưu ý nho nhỏ cho những bạn lập trình nâng cao, đó là cảm biến DHT11 sẽ "treo" (delay) chương trình của bạn trong quá trình nó đọc nhiệt độ, độ ẩm!

Màn hình LCD



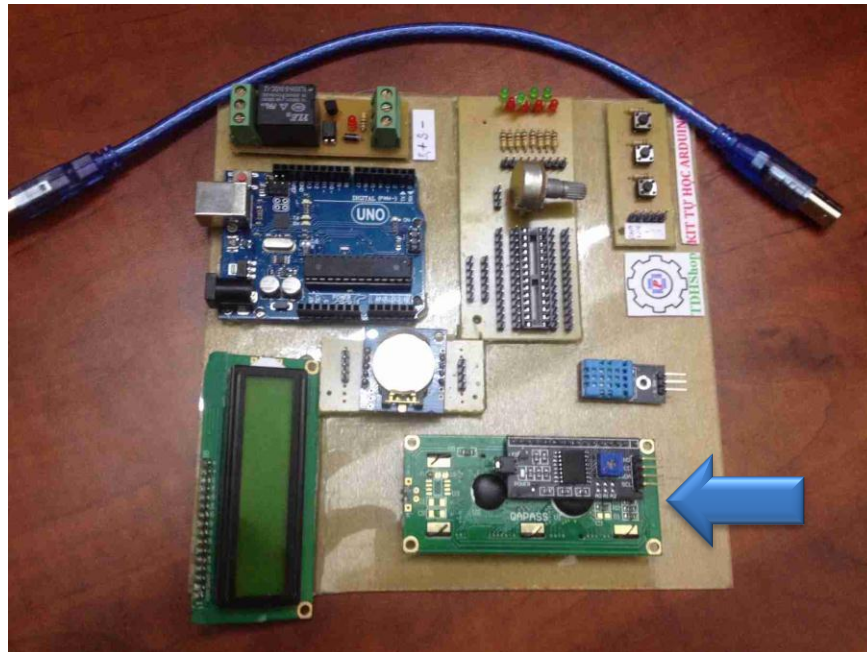
Thông thường, để sử dụng màn hình LCD, bạn sẽ phải mất rất nhiều chân trên Arduino để điều khiển.

Do vậy, để đơn giản hóa công việc, người ta đã tạo ra một loại mạch điều khiển màn hình LCD sử dụng giao tiếp I2C. Nói một cách đơn giản, bạn chỉ tốn ... 2 dây để điều khiển màn hình, thay vì 8 dây như cách thông thường.



SỔ TAY ARDUINO

Bạn chỉ việc hàn mạch vào như thế này là xong.



2 chân SDA và SCL là 2 chân tín hiệu dùng cho giao tiếp I2C.

Download và cài đặt thư viện hỗ trợ sử dụng màn hình LCD qua giao tiếp I2C: <http://tdhshop.com.vn/tong-hop-cac-thu-vien-cho-arduino>

Nếu bạn không có module hỗ trợ này, bạn vẫn có thể điều khiển màn hình theo cách thông thường. Tuy nhiên sẽ rất tốn thời gian và chưa chắc bạn sẽ làm được !

Kết nối module màn hình với Arduino

Module màn hình LCD (16 x 2)	Arduino
GND	GND
Vcc	5V

Web: tdhshop.com.vn - **Chuyên Kit TỰ HỌC ARDUINO**

SỔ TAY ARDUINO

SDA	A4
SCL	A5

Lập trình điều khiển:

Test màn hình LCD.

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

```
// lưu ý một số loại LCD lại sử dụng địa chỉ : 03F
```

```
//0x27 là địa chỉ màn hình trong bus I2C. Phần này chúng ta không cần phải quá  
bận tâm vì hầu hết màn hình (20x4,...) đều như thế này!
```

```
//16 là số cột của màn hình (nếu dùng loại màn hình 20x4) thì thay bằng 20
```

```
//2 là số dòng của màn hình (nếu dùng loại màn hình 20x4) thì thay bằng 4
```

```
void setup() {
```

```
    lcd.init();    //Khởi động màn hình. Bắt đầu cho phép Arduino sử dụng màn hình
```

```
    lcd.backlight(); //Bật đèn nền
```

```
    lcd.print("Hello"); //Xuất ra chữ Hello, mặc định sau khi init thì con trỏ tại cột 0  
hàng 0 (trong C, khác với quy ước của tiếng Việt, mọi chỉ số đều bắt đầu bằng số  
0, vì vậy bạn cần hiểu rằng, nếu ta kẻ một bảng có 2 hàng và 16 cột thì ô góc trên  
cùng bên trái là ô (0,0) tương tự với các ô khác, ta cứ tăng dần giá trị lên!
```

```
    lcd.setCursor(0,1); //Đưa con trỏ tới hàng 1, cột 0
```

```
    lcd.print("tdhshop !");
```

```
}
```

```
void loop() {
```

```
}
```

SỔ TAY ARDUINO

Kết hợp đọc nhiệt độ độ - độ ẩm và xuất ra màn hình: (sau khi đã nối LCD như trên tiếp tục kết nối với DHT11 với sơ đồ nối dây như hình trên)

Bây giờ bạn đã biết cách đọc nhiệt độ, độ ẩm rồi và cũng đã biết xuất ra màn hình. Vậy bây giờ chúng ta chỉ việc kết hợp cả 2 lại để được một đoạn code như thế này:

Nạp code trên vào arduino và cùng xem kết quả nhé:

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);

const int DHTPIN = 2;
const int DHTTYPE = DHT11;
DHT dht(DHTPIN, DHTTYPE);

byte degree[8] = {
  0B01110,
  0B01010,
  0B01110,
  0B00000,
  0B00000,
  0B00000,
  0B00000,
  0B00000
};

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.print("Nhiệt độ: ");
  lcd.setCursor(0,1);
  lcd.print("Độ ẩm: ");
  lcd.createChar(1, degree);
  dht.begin();
```

SỔ TAY ARDUINO

```
}  
void loop() {  
    float h = dht.readHumidity();  
    float t = dht.readTemperature();  
  
    if (isnan(t) || isnan(h)) { // Kiểm tra xem thử việc đọc giá trị có bị thất bại hay không. hàm isnan  
        bạn xem tại đây http://arduino.vn/reference/isnan  
    }  
    else {  
        lcd.setCursor(10,0);  
        lcd.print(round(t));  
        lcd.print(" ");  
        lcd.write(1);  
        lcd.print("C");  
        lcd.setCursor(10,1);  
        lcd.print(round(h));  
        lcd.print(" %");  
    }  
}
```

9. CẢM BIẾN NHIỆT ĐỘ (LM35).

Cảm biến nhiệt độ LM35 là một loại *cảm biến tương tự* rất hay được ứng dụng trong các ứng dụng đo nhiệt độ thời gian thực. Vì nó hoạt động khá chính xác với sai số nhỏ, đồng thời với kích thước nhỏ và giá thành rẻ là một trong những ưu điểm của nó. Vì đây là cảm biến tương tự (analog sensor) nên ta có thể dễ dàng đọc được giá trị của nó bằng hàm `analogRead()`. Nào, cùng nhau tìm hiểu thôi!

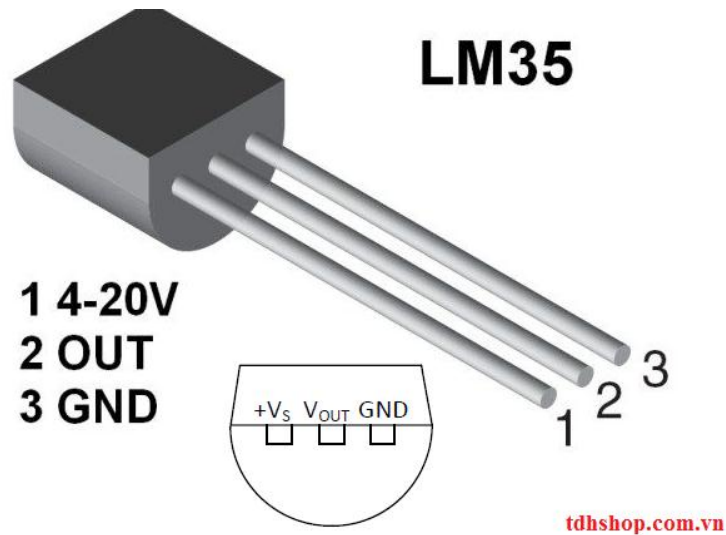
Qua bài viết này, mình hi vọng nó sẽ đem lại cho bạn một vài mẹo nhỏ để ứng dụng hàm `analogRead()` một cách nhuần nhuyễn, và bật mí cho bạn cách sử dụng cảm biến LM35 - một cảm biến nhiệt độ rất hay và dễ sử dụng!

Phần cứng

- Arduino UNO
- Breadboard
- Dây cắm breadboard
- 1 cảm biến LM35

SỔ TAY ARDUINO

Giới thiệu về cảm biến LM35:



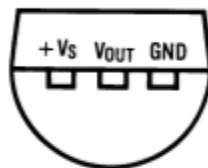
LM35 là một cảm biến nhiệt độ analog:

Nhiệt độ được xác định bằng cách đo hiệu điện thế ngõ ra của LM35.

→ Đơn vị nhiệt độ: °C.

→ Nhiệt độ thay đổi tuyến tính: 10mV/°C

**TO-92
Plastic Package**



BOTTOM VIEW
DS005516-2

**Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A**

Sơ đồ chân của LM35

LM35 không cần phải canh chỉnh nhiệt độ khi sử dụng.

SỔ TAY ARDUINO

Độ chính xác thực tế: $1/4^{\circ}\text{C}$ ở nhiệt độ phòng và $3/4^{\circ}\text{C}$ ngoài khoảng -55°C tới 150°C

LM35 có hiệu năng cao, công suất tiêu thụ là 60uA

Cảm biến LM35 hoạt động bằng cách cho ra một giá trị hiệu điện thế nhất định tại chân Vout (chân giữa) ứng với mỗi mức nhiệt độ.

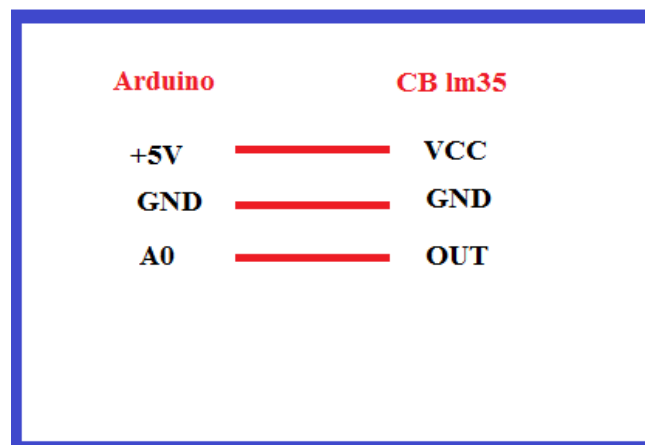
Như vậy, bằng cách đưa vào chân bên trái của cảm biến LM35 hiệu điện thế 5V, chân phải nối đất, đo hiệu điện thế ở chân giữa bằng các pin A0 trên arduino (giống y hệt cách đọc giá trị biến trở), bạn sẽ có được nhiệt độ ($0-100^{\circ}\text{C}$) bằng công thức:

```
float temperature = (5.0*analogRead(A0)*100.0/1024.0);
```

Với LM35, bạn có thể tự tạo cho mình mạch cảm biến nhiệt độ sử dụng LM35 và tự động ngắt điện khi nhiệt độ vượt ngưỡng tối đa, đóng điện khi nhiệt độ thấp hơn ngưỡng tối thiểu thông qua module rơ le...

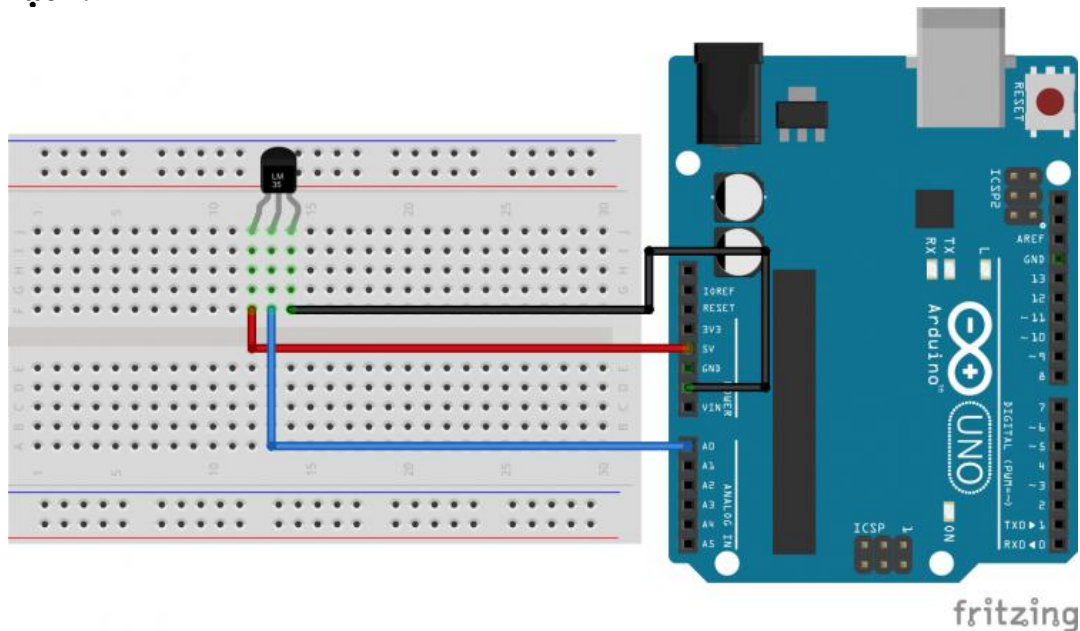
LM35 thay đổi nhiệt độ nhanh và chính xác.

Sơ đồ nối dây:



SỔ TAY ARDUINO

Lắp mạch:



Lập trình điều khiển:

```
int sensorPin = A0; // chân analog kết nối tới cảm biến LM35
void setup() {
  Serial.begin(9600); // Khởi động Serial ở mức baudrate 9600
  // Bạn không cần phải pinMode cho các chân analog trước khi dùng nó
}

void loop() {
  // đọc giá trị từ cảm biến LM35
  int reading = analogRead(sensorPin);
  // tính ra giá trị hiệu điện thế (đơn vị Volt) từ giá trị cảm biến
  float voltage = reading * 5.0 / 1024.0;
  // ở trên mình đã giới thiệu, cứ mỗi 10mV = 1 độ C.
  // Vì vậy nếu biến voltage là biến lưu hiệu điện thế (đơn vị Volt)
  // thì ta chỉ việc nhân voltage cho 100 là ra được nhiệt độ!
  float temp = voltage * 100.0;
  Serial.println(temp);
  /*Mẹo:
  Các bạn phải khai báo phân thực cho toàn bộ các số thực nhé!
```

SỔ TAY ARDUINO

*/

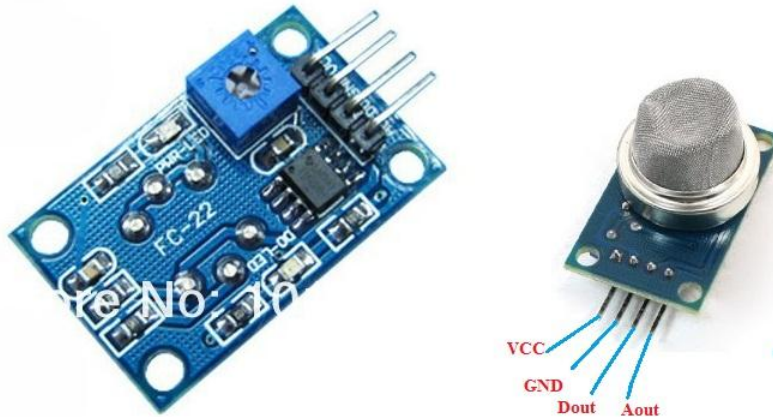
```
delay(1000); //đợi 1 giây cho lần đọc tiếp theo
```

```
}
```

10. LẬP TRÌNH ARDUINO VỚI CẢM BIẾN KHÍ GAS (MQ2).

Ở đây ta sẽ dùng Module cảm biến khí ga MQ2-GAS. Đây là cảm biến indoor nên bạn phải để nó trong nhà ở nhiệt độ phòng. Mức khí GAS nhận được bạn đọc về ở đầu ra dạng Analog của nó. Với Arduino thì ta dùng các chân Analog của nó để đọc.

Sơ đồ nối dây giữa arduino và CB khí gas (MQ2).



Sơ đồ nối dây theo bảng này:

Arduino	MQ2
A0	Aout
VCC	5V
GND	GND

SỔ TAY ARDUINO

Lập trình điều khiển:

```
void setup() {  
  
  Serial.begin(9600); //Mở cổng Serial để giao tiếp | tham khảo Serial  
  
}  
  
void loop() {  
  
  int value = analogRead(A0); //đọc giá trị điện áp ở chân A0 - chân cảm biến  
  
  //(value luôn nằm trong khoảng 0-1023)  
  
  Serial.println(value); //xuất ra giá trị vừa đọc  
  
  //-----  
  
  delay(1000); //đợi 1 giây để bạn kịp thấy serial - (optional)  
  
}
```

Để test, bạn có thể dùng hộp quẹt,... và xì hơi ga vào tấm lưới của nó.

SỔ TAY ARDUINO

11. CẢM BIẾN NHIỆT ĐỘ (DS18B20).

Hướng dẫn sử dụng cảm biến nhiệt độ DS18B20 (-55°C đến $+125^{\circ}\text{C}$) sai số $\pm 0.5^{\circ}\text{C}$. Nhiệt độ ít bị chênh lệch hơn.

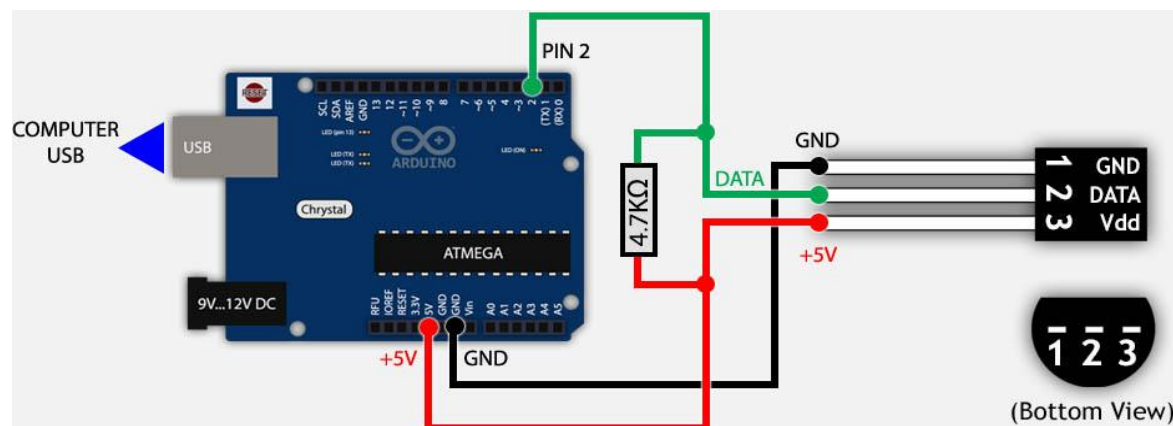
Giới thiệu sơ lược về DS18B20:



Về căn bản, con này cũng giống như LM35, các bạn có thể xem phần cảm biến LM35 ở phần trước. DS18B20 dùng cơ chế truyền tín hiệu 1-Wire. Nôm na với cái cơ chế này, chúng ta có thể đọc nhiều con DS18B20 cùng một lúc trên cùng 1 dây.

- Sử dụng một chân data với 64bit serial code cho phép chúng ta dùng nhiều con trên cùng 1 chân digital (cơ chế 1-Wire)
- Có thể cấp nguồn từ 3 - 5.5V
- Đo từ -55°C đến $+125^{\circ}\text{C}$ sai số $\pm 0.5^{\circ}\text{C}$ nếu đang trong khoảng -10°C đến $+85^{\circ}\text{C}$

Sơ đồ nối dây:



SỔ TAY ARDUINO

Lập trình Arduino cho cảm biến DS18B20:

Trước khi bắt tay vào lập trình ta cần phải tải hai thư viện sau và add vào thư viện cho arduino IDE:

Thư viện thứ 1: **OneWire.h**

http://www.mediafire.com/file/a9s0ww0l10dl25s/2689_123450-1468426301-0-onewire.zip

Thư viện thứ 2: **DallasTemperature.h**

http://www.mediafire.com/file/4r9ksj6z455d55j/2664_882450-1468426343-0-arduino-temperature-control-library-master.zip

Ta bắt tay vào lập trình sau khi add xong 2 thư viện trên:

```
//Include thư viện

#include <OneWire.h>

#include <DallasTemperature.h>

// Chân nối với Arduino

#define ONE_WIRE_BUS 2 //Thiết đặt thư viện onewire

OneWire oneWire(ONE_WIRE_BUS);

//Mình dùng thư viện DallasTemperature để đọc cho nhanh

DallasTemperature sensors(&oneWire);

void setup(void)

{

    Serial.begin(9600);

    sensors.begin();

}

void loop(void)
```

SỔ TAY ARDUINO

```
{  
    sensors.requestTemperatures();  
  
    Serial.print("Nhiệt độ");  
  
    Serial.println(sensors.getTempCByIndex(0)); // vì 1 ic nên dùng 0  
  
    //chờ 1 s rồi đọc để bạn kịp thấy sự thay đổi  
  
    delay(1000);  
}
```

Vậy là ta lập trình xong các bạn mở cổng *Serial port* xem kết quả nhé.

12. CẢM BIẾN CHUYỂN ĐỘNG PIR.

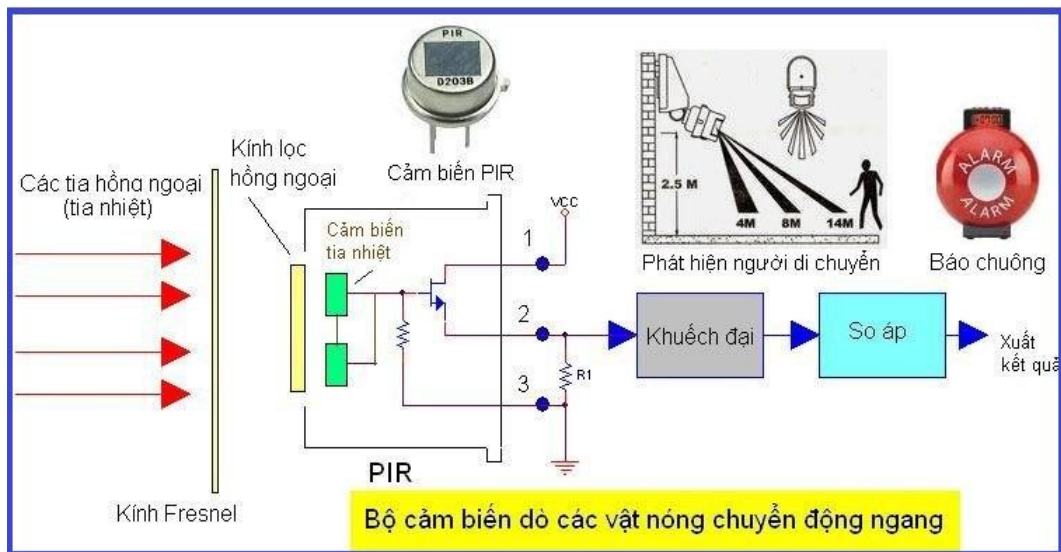
PIR là chữ viết tắt của **Passive InfraRed sensor** (PIR sensor), tức là bộ cảm biến thụ động dùng nguồn kích thích là tia hồng ngoại. Tia hồng ngoại (IR) chính là các tia nhiệt phát ra từ các vật thể nóng. Trong các cơ thể sống, trong chúng ta luôn có thân nhiệt (thông thường là ở 37 độ C), và từ cơ thể chúng ta sẽ luôn phát ra các tia nhiệt, hay còn gọi là các tia hồng ngoại, người ta sẽ dùng một tế bào điện để chuyển đổi tia nhiệt ra dạng tín hiệu điện và nhờ đó mà có thể làm ra cảm biến phát hiện các vật thể nóng đang chuyển động. Cảm biến này gọi là thụ động vì nó không dùng nguồn nhiệt tự phát (làm nguồn tích cực, hay chủ động) mà chỉ phụ thuộc vào các nguồn tha nhiệt, đó là thân nhiệt của các thực thể khác, như con người, con vật...



Trên đây là đầu dò PIR, loại bên trong gắn 2 cảm biến tia nhiệt, nó có 3 chân ra, một chân nổi masse, một chân nối với nguồn volt DC, mức áp làm việc có thể từ 3 đến 15V.

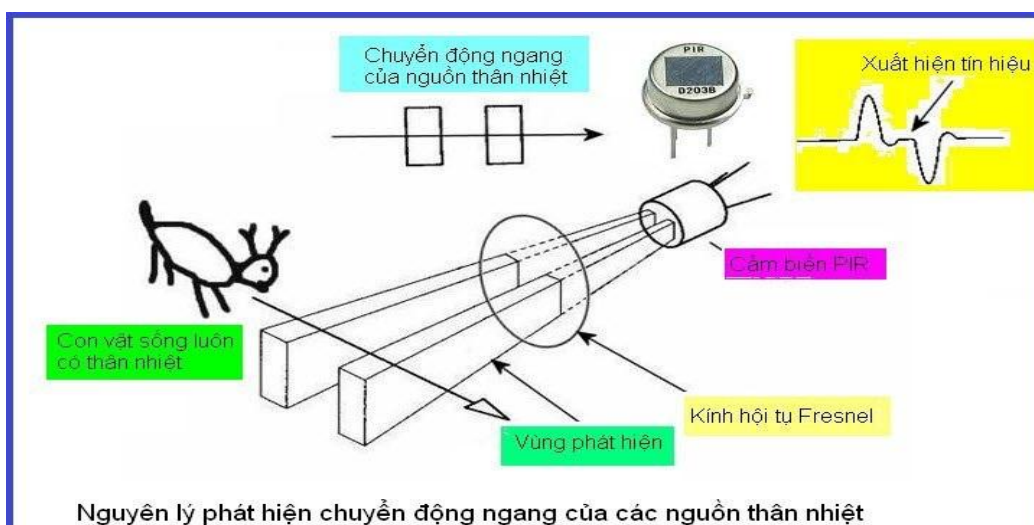
SỔ TAY ARDUINO

Góc dò lớn. Để tăng độ nhạy cho đầu dò, Bạn dùng kính Fresnel, nó được thiết kế cho loại đầu có 2 cảm biến, góc dò lớn, có tác dụng ngăn tia tử ngoại.



Nguyên lý làm việc của loại đầu dò PIR như hình sau:

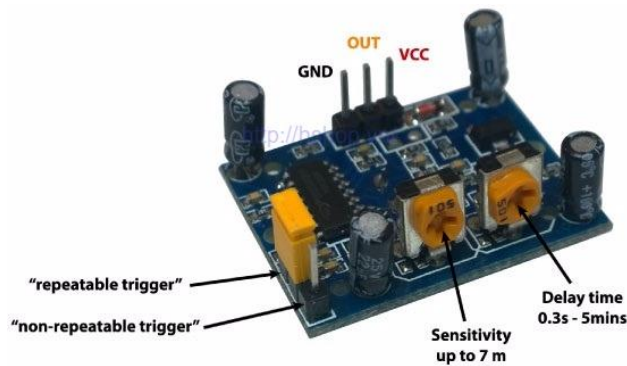
Các nguồn nhiệt (với người và con vật là nguồn thân nhiệt) đều phát ra tia hồng ngoại, qua kính Fresnel, qua kích lọc lấy tia hồng ngoại, nó được cho tiêu tụ trên 2 cảm biến hồng ngoại gắn trong đầu dò, và tạo ra điện áp được khuếch đại với transistor FET. Khi có một vật nóng đi ngang qua, từ 2 cảm biến này sẽ cho xuất hiện 2 tín hiệu và tín hiệu này sẽ được khuếch đại để có biên độ đủ cao và đưa vào mạch so áp để tác động vào một thiết bị điều khiển hay báo động.



Sơ đồ chân của cảm biến PIR

Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO



- GND – kết nối với mặt đất
- OUT – kết nối đến một chân digital của Arduino
- 5V – kết nối với 5V

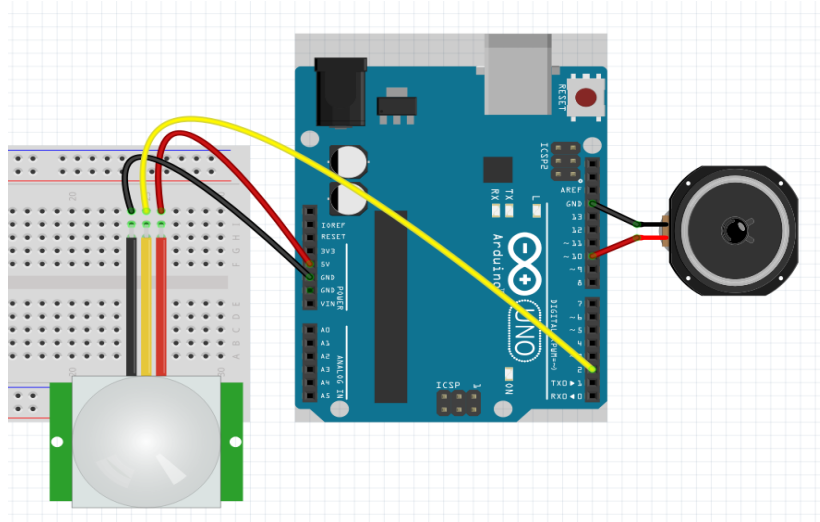
Linh kiện thí nghiệm:

- 1x PIR Motion Sensor
- 1x Arduino UNO (hoặc bất kỳ)
- 1x LED bất kỳ hoặc loa.

Sơ đồ kết nối giữa arduino và cảm biến:

Arduino	CB chuyển động PIR
+5V	VCC
GND	GND
D2	OUT
CHÂN D10 CỦA ARDUINO NỐI VỚI CHÂN + CỦA LOA HOẶC LED)	

SỔ TAY ARDUINO



Lập trình điều khiển:

```
int ledPin = 13;    // chọn chân 13 báo hiệu LED

int inputPin = 2;   // chọn ngõ tín hiệu vào cho PIR

int pirState = LOW; // Bắt đầu với không có báo động

int val = 0;

int pinSpeaker = 10; //chọn chân cho chuông khi có đột nhập

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);
  pinMode(pinSpeaker, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  val = digitalRead(inputPin); // đọc giá trị đầu vào.
```


SỔ TAY ARDUINO

```
if (val == HIGH)           // nếu giá trị ở mức cao.(1)
{
    digitalWrite(ledPin, HIGH); // LED On
    playTone(300, 160);        // thời gian chuông kêu
    delay(150);
    if (pirState == LOW)
    {
        Serial.println("Motion detected!");
        pirState = HIGH;
    }
}
else
{
    digitalWrite(ledPin, LOW);
    playTone(0, 0);
    delay(300);
    if (pirState == HIGH)
    {
        Serial.println("Motion ended!");
        pirState = LOW;
    }
}

void playTone(long duration, int freq)
{

```

SỔ TAY ARDUINO

```
duration *= 1000;

int period = (1.0 / freq) * 1000000;

long elapsed_time = 0;

while (elapsed_time < duration)
{
    digitalWrite(pinSpeaker,HIGH);

    delayMicroseconds(period / 2);

    digitalWrite(pinSpeaker, LOW);

    delayMicroseconds(period / 2);

    elapsed_time += (period);
}
}
```

13. CẢM BIẾN KHOẢNG CÁCH (SRF05). PHẦN NÀY BỔ XUNG

Đối với robot, khi ở chế độ tự động, nó phải lấy thông tin của môi trường xung quanh như: khoảng cách, nhiệt độ, ánh sáng ... Và tiến hành phân tích các dữ liệu đó, cuối cùng ra quyết định phù hợp. Ví dụ như robot tự tìm đường đi thì nó phải tính được xem phía trước có vật cản hay không, và tiến hành đi tới hay rẽ sang một hướng khác. Bài viết này sẽ hướng dẫn bạn đọc hiểu và sử dụng được một trong những cảm biến khoảng cách phổ biến - SRF05.

Thiết bị cần chuẩn bị

Các bạn cần chuẩn bị các thiết bị sau:

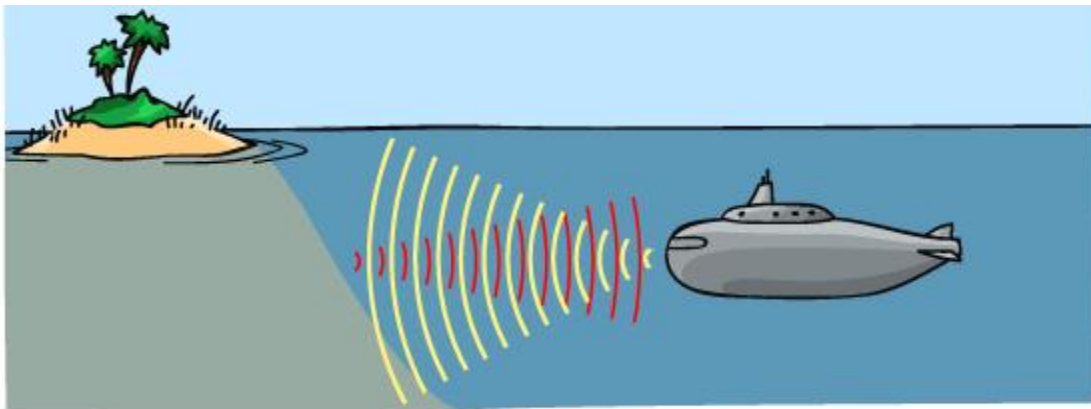
- 1 x Arduino Uno R3.
- 1 x cảm biến siêu âm SRF05.
- 1 x breadboard để cắm mạch test.
- Dây cắm các loại.

Cảm biến siêu âm SRF05

SỔ TAY ARDUINO

Sóng siêu âm (sonar) là một loại sóng cao tần mà con người không thể nghe thấy được. Tuy nhiên, ta có thể thấy được sự hiện diện của sóng siêu âm ở khắp mọi nơi trong tự nhiên. Ta có các loài động vật như dơi, cá heo ... dùng sóng siêu âm để liên lạc với nhau, để săn mồi hay định vị trong không gian.

Việc tính toán khoảng cách cũng còn phụ thuộc rất nhiều vào môi trường truyền dẫn, ví dụ như sóng âm truyền trong môi trường nước hay kim loại sẽ nhanh hơn rất nhiều so với sóng âm được truyền trong môi trường không khí. *Lưu ý là sóng âm không thể truyền được trong môi trường chân không.*



Cảm biến siêu âm SRF05 cũng hoạt động theo như nguyên tắc ở trên, thiết bị gồm có 2 loa - thu và phát - cùng với 5 chân để kết nối với Arduino, **tầm hoạt động tối đa của cảm biến này nằm trong khoảng 5m.**



Chức năng của các chân này như sau:

1. **Vcc:** cấp nguồn cho cảm biến.
2. **Trigger:** kích hoạt quá trình phát sóng âm. Quá trình kích hoạt khi một chu kì điện cao / thấp diễn ra.

SỔ TAY ARDUINO

3. **Echo**: bình thường sẽ ở trạng thái 0V, được kích hoạt lên 5V ngay khi có tín hiệu trả về, sau đó trở về 0V.
4. **Gnd**: nối với cực âm của mạch
5. **OUT**: không sử dụng

Ta tiến hành lắp đặt cảm biến khoảng cách SRF05 theo sơ đồ sau:

1. **Vcc**: nối với nguồn 5V của Arduino
2. **Gnd**: nối với PIN GND.
3. **Trigger**: nối với PIN 8
4. **Echo**: nối với PIN 7

Lập trình điều khiển

```
const int trig = 8; // chân trig của HC-SR05

const int echo = 7; // chân echo của HC-SR05

void setup()

{

  Serial.begin(9600); // giao tiếp Serial với baudrate 9600

  pinMode(trig,OUTPUT); // chân trig sẽ phát tín hiệu

  pinMode(echo,INPUT); // chân echo sẽ nhận tín hiệu

}

void loop()

{

  unsigned long duration; // biến đo thời gian

  int distance; // biến lưu khoảng cách

  /* Phát xung từ chân trig */

  digitalWrite(trig,0); // tắt chân trig
```

SỔ TAY ARDUINO

```
delayMicroseconds(2);

digitalWrite(trig,1); // phát xung từ chân trig

delayMicroseconds(5); // xung có độ dài 5 microseconds

digitalWrite(trig,0); // tắt chân trig

/* Tính toán thời gian */

// Đo độ rộng xung HIGH ở chân echo.

duration = pulseIn(echo,HIGH);

// Tính khoảng cách đến vật.

distance = int(duration/2/29.1);

/* In kết quả ra Serial Monitor */

Serial.print(distance);

Serial.println("cm");

delay(200);

}
```

Giải tích code:

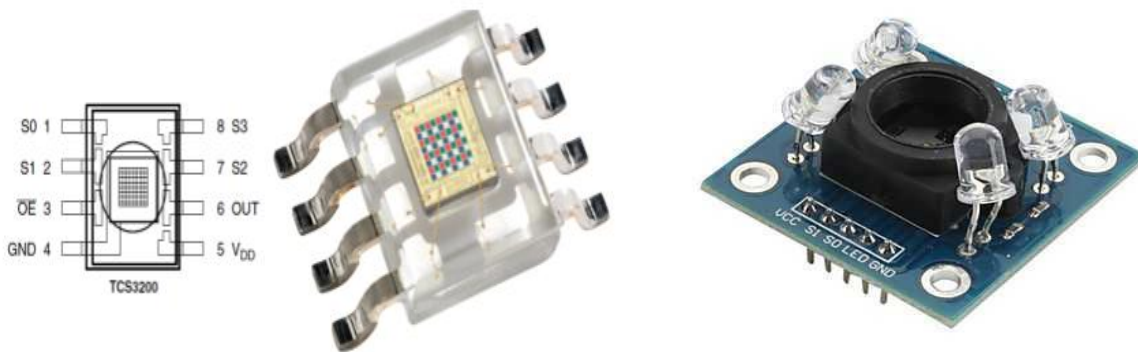
Ta biết thời gian âm thanh truyền trong không khí ở 20°C là **344 m/s**. Bằng quy tắc tam suất đơn giản ta có thể dễ dàng tính được sóng âm di chuyển 1 cm trong không khí sẽ mất **$1000 / 344 * 100 \approx 29.1$** .

Arduino cung cấp cho ta hàm **pulseIn** có tác dụng trả về thời gian (tính bằng milisec) kể từ khi hàm này được gọi cho đến khi có tín hiệu tại PIN chỉ định trước, hay trả về 0 nếu không nhận được tín hiệu / quá thời gian timeout.

SỔ TAY ARDUINO

14. CẢM BIẾN MÀU (TCS3200).

Module cảm biến màu TCS3200 là một module cảm biến phát hiện đầy đủ màu sắc, bao gồm cả cảm biến màu TCS3200 với khả năng nhận biết 3 màu cơ bản RGB và 4 đèn LED trắng. Các TCS3200 có thể phát hiện và đo lường gần như tất cả màu sắc có thể nhìn thấy. Ứng dụng bao gồm kiểm tra đọc dải, phân loại theo màu sắc, cảm biến ánh sáng xung quanh và hiệu chuẩn, và kết hợp màu sắc, đó chỉ là một vài ứng dụng. TCS3200 có các bộ tách sóng quang, có 2 bộ lọc màu sắc là bộ lọc màu đỏ, xanh, hoặc màu xanh, hoặc không có bộ lọc (rõ ràng). Các bộ lọc của mỗi màu được phân bố đều khắp các mảng để loại bỏ sai lệch vị trí giữa các điểm màu. Bên trong là một bộ dao động tạo ra đầu ra sóng vuông có tần số là tỷ lệ thuận với cường độ của màu sắc lựa chọn.



Thông số kỹ thuật:

Điện áp cung cấp: (2.7V đến 5.5V)

Chuyển đổi từ cường độ ánh sáng sang tần số với độ phân giải cao.

Có khả năng lập trình để nhận biết đầy đủ các màu sắc.

Điện năng tiêu thụ thấp.

Giao tiếp trực tiếp với vi điều khiển.

S0 ~ S1: Dùng để lựa chọn tỉ lệ tần số đầu ra

S2 ~ S3: Dùng để lựa chọn kiểu photodiode

OUT Pin: Đầu ra tần số.

OE Pin: Tần số đầu ra cho phép hoạt động (hoạt động ở mức thấp).

Hỗ trợ đèn LED bổ sung và kiểm soát ánh sáng.

SỔ TAY ARDUINO

Kích thước: 28.4x28.4mm

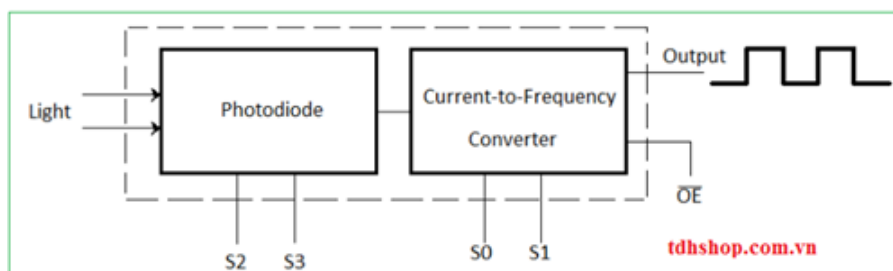
Nguyên tắc nhận biết màu:

TCS3200 có bộ lọc màu, khi lựa chọn một bộ lọc màu nó sẽ cho phép chỉ nhận biết 1 màu và các màu khác sẽ bị chặn. Ví dụ, khi lựa chọn các bộ lọc màu đỏ, Chỉ có ánh sáng tới màu đỏ có thể được thông qua, màu xanh và màu xanh lá cây sẽ được ngăn chặn. Vì vậy chúng ta có thể nhận được cường độ ánh sáng màu đỏ. Tương tự như vậy, khi lựa chọn các bộ lọc khác mà chúng ta có thể nhận được ánh sáng màu xanh hoặc màu xanh lá cây.

Cách thiết lập màu sắc cho TCS3200:

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Tần số đầu ra của TCS3200 trong khoảng 2HZ~500KHZ. Tần số đầu ra có dạng xung vuông với tần số khác nhau khi mà màu sắc khác nhau và cường độ sáng là khác nhau. Chúng ta có thể lựa chọn tỉ lệ giữa cường độ sáng và màu sắc theo bảng sau. Trên module này cũng có chân LED EN đây là chân điều khiển đèn LED để tắt hoặc bật.



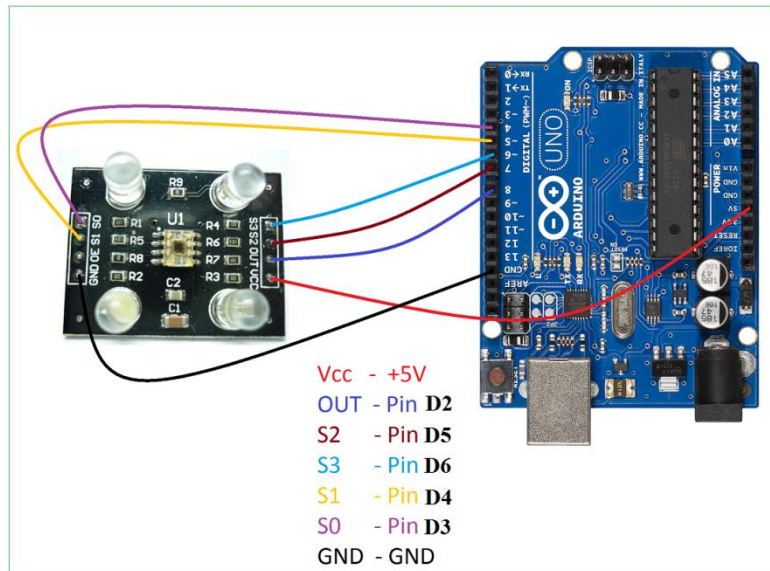
Cách thiết lập tần số cho TCS3200:

S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO

Sơ đồ nối dây:

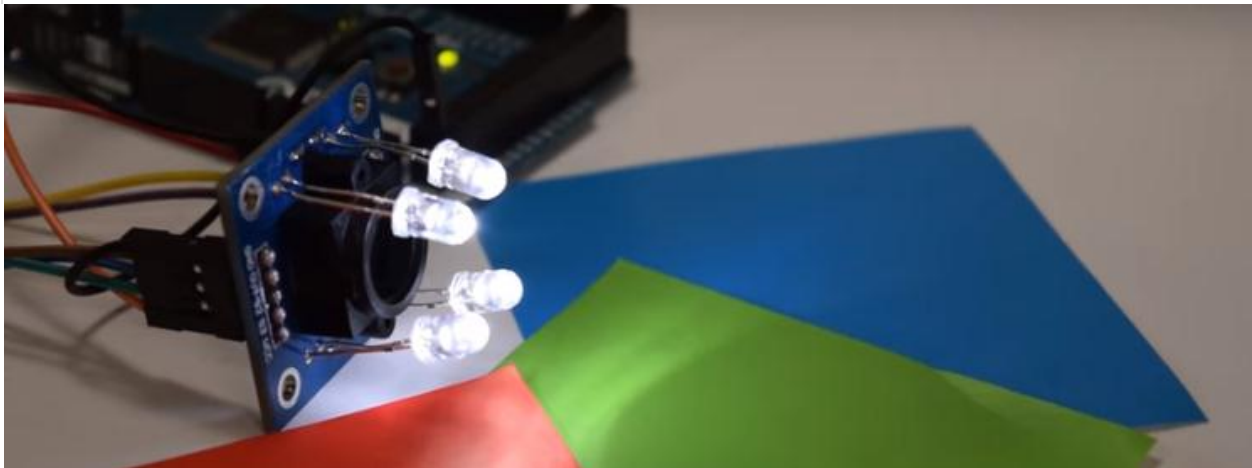


Lập trình điều khiển:

```
1. #define S0 4
2. #define S1 5
3. #define S2 6
4. #define S3 7
5. #define sensorOut 8
6.
7. int frequency = 0;
8.
9. void setup() {
10. pinMode(S0, OUTPUT);
11. pinMode(S1, OUTPUT);
12. pinMode(S2, OUTPUT);
13. pinMode(S3, OUTPUT);
14. pinMode(sensorOut, INPUT);
15.
16. // Setting frequency-scaling to 20%
17. digitalWrite(S0,HIGH);
18. digitalWrite(S1,LOW);
19.
20. Serial.begin(9600);
21. }
22.
23. void loop() {
24. // Setting red filtered photodiodes to be read
25. digitalWrite(S2,LOW);
26. digitalWrite(S3,LOW);
```

SỔ TAY ARDUINO

```
27. frequency = pulseIn(sensorOut, LOW);
28. Serial.print("R= "); //printing name
29. Serial.print(frequency); //printing RED color frequency
30. Serial.print(" ");
31. delay(100);
32. digitalWrite(S2,HIGH);
33. digitalWrite(S3,HIGH);
34. frequency = pulseIn(sensorOut, LOW);
35. Serial.print("G= "); //printing name
36. Serial.print(frequency); //printing RED color frequency
37. Serial.print(" ");
38. delay(100);
39. digitalWrite(S2,LOW);
40. digitalWrite(S3,HIGH);
41. frequency = pulseIn(sensorOut, LOW);
42. Serial.print("B= "); //printing name
43. Serial.print(frequency); //printing RED color frequency
44. Serial.println(" ");
45. delay(100);
46. }
```



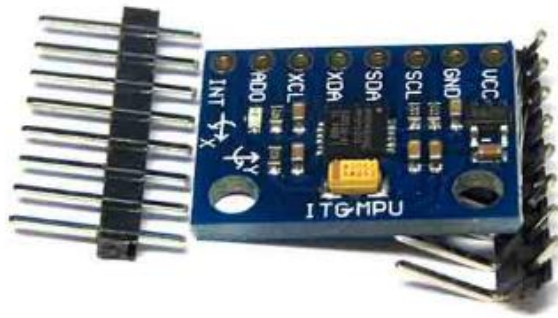
Sau khi upload chương trình và kết nối xong chúng ta lấy 3 mẫu giấy màu như trên đưa sát vào 4 led màu trắng của cảm biến, sau đó mở serial port lên xem cái giá trị trả về của màu đó nhé.

SỔ TAY ARDUINO

15. CẢM BIẾN GIA TỐC (MPU 6050).

Cảm biến gia tốc MPU 6050 là cảm biến có 6 trục được thiết kế cho các ứng dụng nguồn nuôi thấp, giá rẻ, và có hiệu suất cao, được ứng dụng trong các điện thoại máy tính bảng, và các thiết bị di động khác.

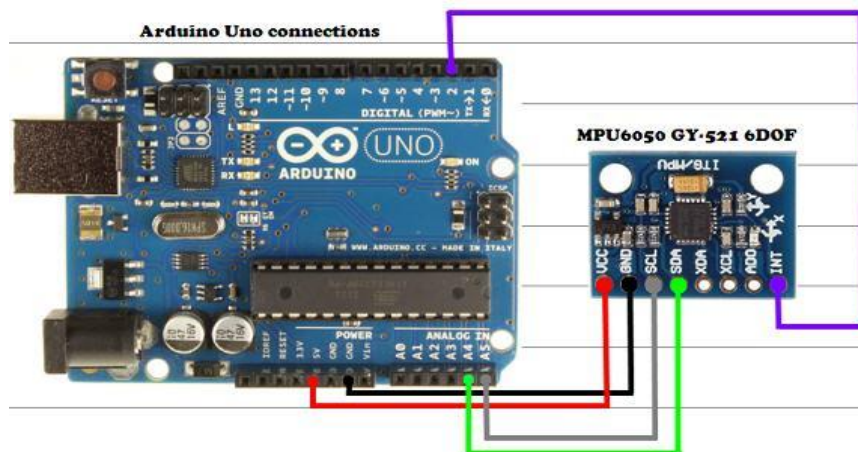
MPU-6000/6050 gồm có 3 trục con quay hồi chuyển (gyroscope) và 3 trục gia tốc (accelerometer) được thiết kế cùng nhau trong nhân chip Digital Motion Processor (DMP) sử dụng thuật toán MotionFusion. thuật toán MotionFusion có khả năng giao tiếp với 1 số cảm biến khác qua các chân I2C phụ trợ (auxiliary master I2C bus), giúp cho thiết bị có thể trang bị đầy đủ các loại cảm biến trong 1 hệ thống điều khiển. Board MPU6050 đã được thiết kế với IC nguồn 3.3V trên board và trở kéo trên bus I2C.



Thông số kĩ thuật cảm biến gia tốc MPU 6050 :

- Điện áp cấp: 3~5v.
- 3 góc con quay hồi chuyển với độ nhạy lên đến 131 LSBs/sps và đầy đủ các độ ± 250 , ± 500 , ± 1000 , and ± 2000 dps.
- 3 góc gia tốc kế với đầy đủ khả năng lập trình với $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$
- Kích thước: 14 x 21mm.

Sơ đồ nối dây:



SỔ TAY ARDUINO

Lập trình điều khiển:

Trước hết ta phải cài thư viện MPU6050 và I2C vào arduino IDE.

```
#include<Wire.h>

const int MPU_addr=0x68; // I2C address of the MPU-6050

int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

void setup(){

  Wire.begin();

  Wire.beginTransmission(MPU_addr);

  Wire.write(0x6B); // PWR_MGMT_1 register

  Wire.write(0); // set to zero (wakes up the MPU-6050)

  Wire.endTransmission(true);

  Serial.begin(9600);

}

void loop(){

  Wire.beginTransmission(MPU_addr);

  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)

  Wire.endTransmission(false);

  Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers

  AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)

  AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)

  AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)

  Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)

  GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)

  GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)

  GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
```

Web: tdhshop.com.vn - Chuyên Kit TỰ HỌC ARDUINO

SỔ TAY ARDUINO

```
Serial.print("AcX = "); Serial.print(AcX);  
  
Serial.print(" / AcY = "); Serial.print(AcY);  
  
Serial.print(" / AcZ = "); Serial.print(AcZ);  
  
Serial.print(" / Tmp = "); Serial.print(Tmp/340.00+36.53); //equation for temperature in  
degrees C from datasheet  
  
Serial.print(" / GyX = "); Serial.print(GyX);  
  
Serial.print(" / GyY = "); Serial.print(GyY);  
  
Serial.print(" / GyZ = "); Serial.println(GyZ);  
  
delay(333);  
  
}
```