# PHẦN 1: CƠ SỞ LÝ THUYẾT

# Sơ đồ khối chức năng

## Bo mạch Arduino

Arduino là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm.

## ****Arduino có thể kết nối với những gì ?****

## Một hệ thống Arduino có thể cung cấp cho bạn rất nhiều sự tương tác với môi trường xung quanh với:

## Hệ thống cảm biến đa dạng về chủng loại (đo đạc nhiệt độ, độ ẩm, gia tốc, vận tốc, cường độ ánh sáng, màu sắc vật thể, lưu lượng nước, phát hiện chuyển động, phát hiện kim loại, khí độc,…),…

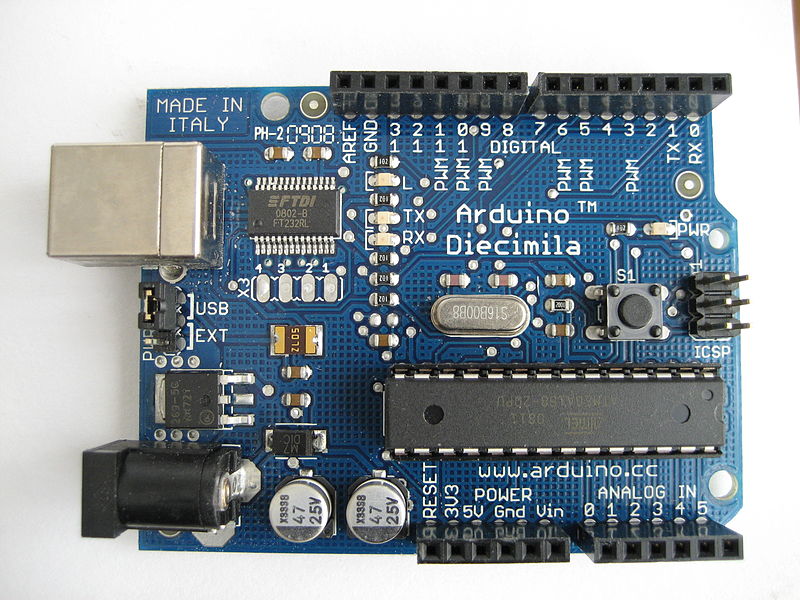
## Các thiết bị hiển thị (màn hình LCD, đèn LED,…).

## Các module chức năng (shield) hỗ trợ kêt nối có dây với các thiết bị khác hoặc các kết nối không dây thông dụng (3G, GPRS, Wifi, Bluetooth, 315/433Mhz, 2.4Ghz,…),...

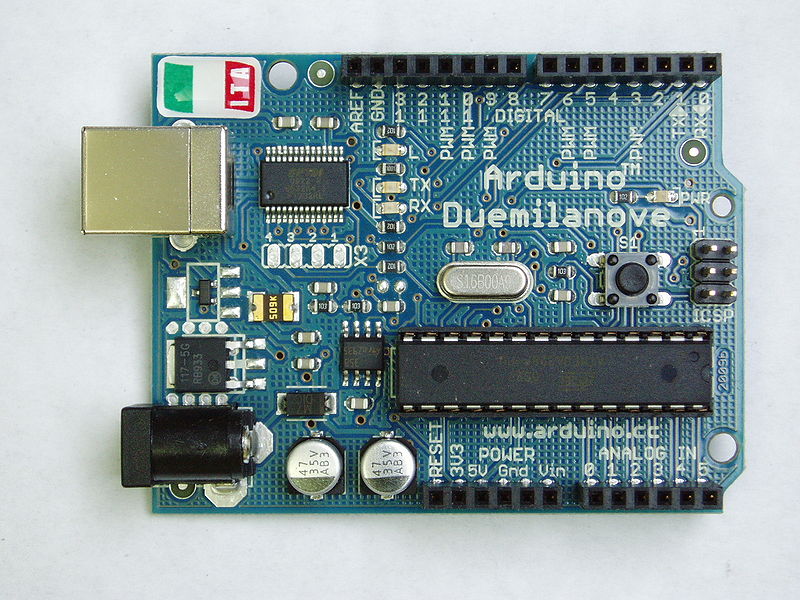
## Định vị GPS, nhắn tin SMS,…

Arduino hiện nay đã được biết đến một cách rộng rãi tại Việt Nam, Với Arduino bạn có thể ứng dụng vào những mạch đơn giản như mạch cảm biến ánh sáng bật tắt đèn, mạch điều khiển động cơ,... hoặc cao hơn nữa bạn có thể làm những sản phẩm như: máy in 3D, Robot, khinh khí cầu, máy bay không người lái,...

Các board Arduino mẫu: Sáu phiên bản phần cứng của Arduino cũng đã được sản xuất thương mại tính đến thời điểm hiện tại:



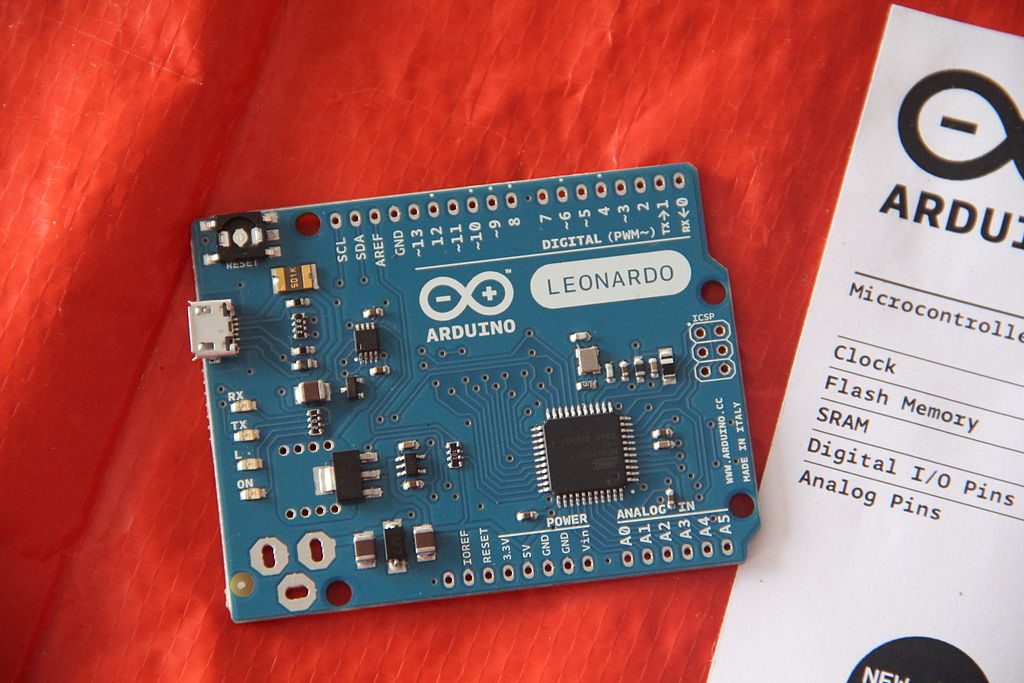
Hình 1.2 Arduino Diecimila in [Stoicheia](https://vi.wikipedia.org/w/index.php?title=Stoicheia&action=edit&redlink=1)



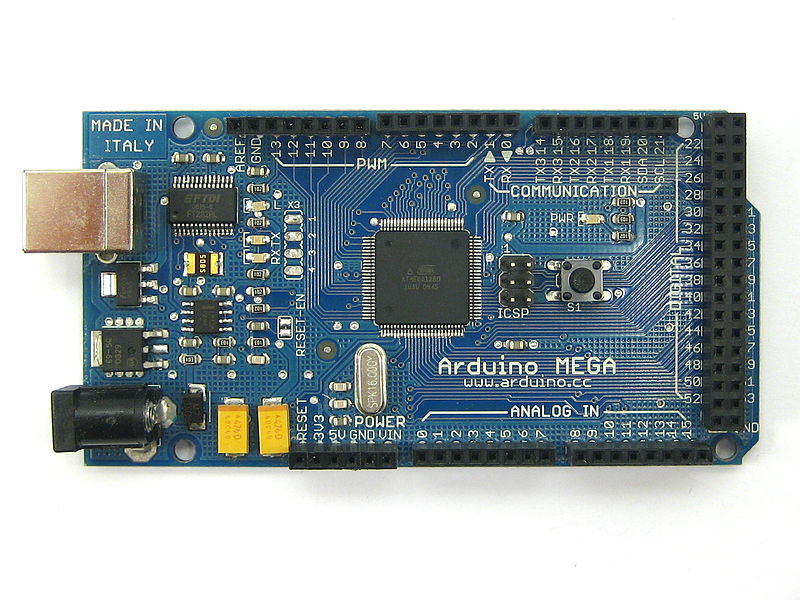
Hình 1.3Arduino Duemilanove (rev 2009b)



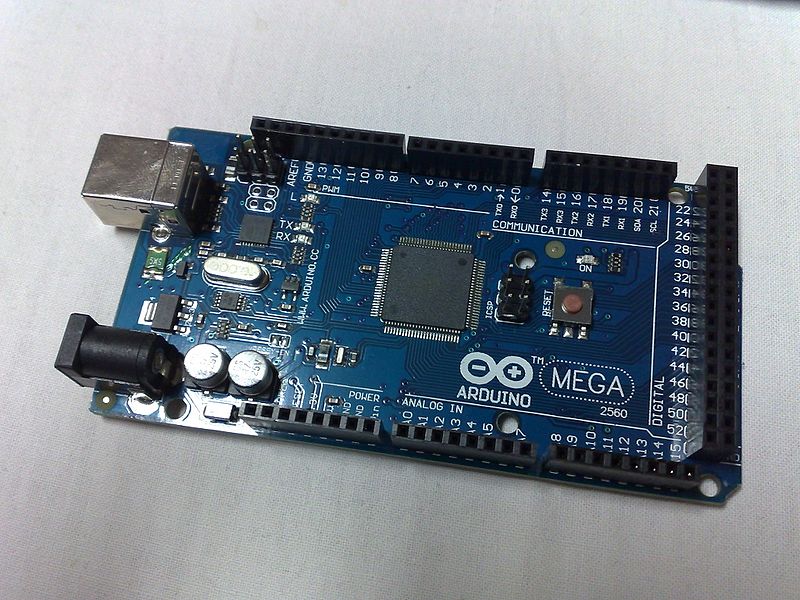
Hình 1.4 Arduino UNO



Hình 1.5 Arduino Leonardo



Hình 1.6 Arduino Mega



Hình 1.7 Arduino MEGA 2560 R3

**1.1.2. Arduino Uno R3**

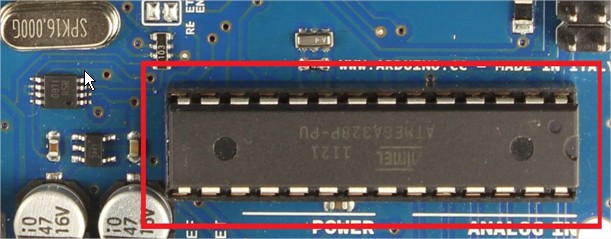
Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3).

Arduino UNO có những thông số cơ bản sau đây:

|  |  |
| --- | --- |
| Vi điều khiển | Atmega 328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| 5V DC (chỉ được cấp qua cổng USB) | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp vào khuyên dung | 7-12V DC |
| Điện áp vào giới hạn | 6-20V DC |
| Số chân Digital I/O | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3.3V) | 50 mA |
| Bộ nhớ flash | 32 KB (ATmega328) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328 |

**Bảng 1.1** Thông số cơ bản của Arduino Uno R3

## ****Vi điều khiển:****



Hình 1.8 Vi điều khiển ATMEGA328

Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lí những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lí tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD,…

## ****Bộ nhớ:****

## Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

* **32KB bộ nhớ Flash**: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng hiếm khi nào cần quá 20KB.
* **2KB cho SRAM** (**S**tatic **R**andom **A**ccess **M**emory): giá trị các biến khai báo khi lập trình sẽ lưu ở đây. Khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
* **1KB EEPROM**(**E**lectrically **E**raseble **P**rogrammable **R**ead **O**nly **M**emory): đây giống như một chiếc ổ cứng mini – nơi bạn có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

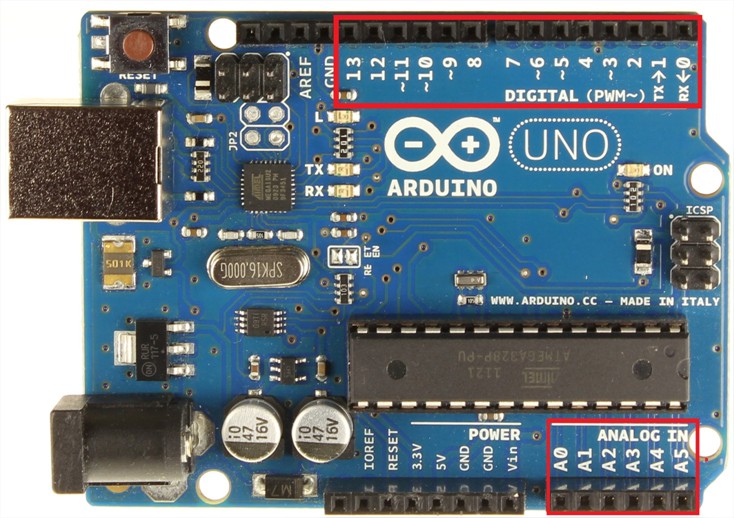
## ****Năng lượng:****

## Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyên dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lí nhất nếu không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, sẽ làm hỏng Arduino UNO.

## ****Các chân năng lượng:****

* **GND (Ground)**: cực âm của nguồn điện cấp cho Arduino UNO. Khi dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
* **5V**: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
* **3.3V**: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
* **Vin (Voltage Input)**: để cấp nguồn ngoài cho Arduino UNO, nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
* **IOREF**: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy, không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
* **RESET**: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

## ****Các cổng vào/ra****

[](http://k3.arduino.vn/img/2014/05/25/0/467_8121-1401018414-0--input.jpg)

Hình 1.9 Sơ đồ các chân vào ra của Arduino UNO

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

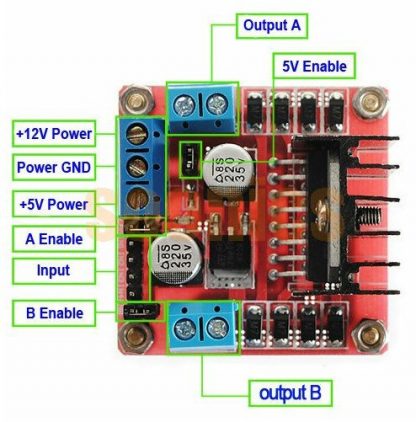
Một số chân digital có các chức năng đặc biệt như sau:

* **2 chân Serial**: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết
* **Chân PWM (~): 3, 5, 6, 9, 10, và 11**: cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 → 28-1 tương ứng với 0V → 5V) bằng hàm analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
* **Chân giao tiếp SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).  Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
* **LED 13**: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

Arduino UNO có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân **AREF** trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V  → 2.5V với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

**1.2. Mạch điều khiển động cơ DC L298**



Hình 1.10 Module điều khiển động cơ DC L298

Mạch điều khiển động cơ DC L298 ( mạch cầu H L298) có khả năng điều khiển 2 động cơ DC, dòng tối đa 2A mỗi động cơ, mạch tích hợp diode bảo vệ và IC nguồn 7805 giúp cấp nguồn 5VDC cho các module khác (chỉ sử dụng 5V này nếu nguồn cấp <12VDC).

Mạch điều khiển động cơ DC L298 dễ sử dụng, chi phí thấp, dễ lắp đặt, là sự lựa chọn tối ưu trong tầm giá.

**Thông số kỹ thuật:**

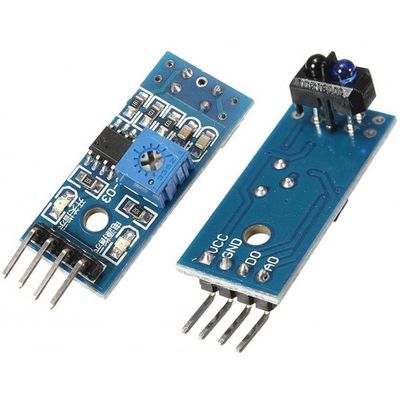
* IC chính: L298 - Dual Full Bridge Driver
* Điện áp đầu vào: 5~30VDC
* Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
* Công suất tối đa: 25W 1 cầu (lưu ý công suất = dòng điện x điện áp nên áp cấp vào càng cao, dòng càng nhỏ, công suất có định 25W).
* Dòng tối đa cho mỗi cầu H là: 2A
* Nhiệt độ bảo quản: -25 ℃ ~ +130 ℃
* Kích thước: 43x43x27mm

**Các chân của L298:**

* 12V power, 5V power: là 2 chân cấp nguồn trực tiếp đến động cơ.  
  Bạn có thể cấp nguồn từ 9V đến 12V ở chân +12V power: như vậy bạn cũng sẽ có điện áp 5v, lợi dụng điều này ta có thể lấy điện áp 5v từ đây để dùng cho mạch Arduino ( và nhớ nối chân Power GND với Arduino)
* Power GND : là chân GND cấp nguồn cho động cơ.
* 2 Jump A enable và B enable, để như hình, đừng rút ra bạn nhé!
* IN1, IN2, IN3, IN4: Là 4 chân input , chức năng nhận tín hiệu từ vi điều khiển hoặc Arduino để điều khiển động cơ.
* Output A: nối với động cơ A. bạn chú ý chân +, -. Nếu bạn nối ngược thì động cơ sẽ chạy ngược.

**1.3. Module dò line**

Cảm biến dò line đơn TCRT5000



Hình 1.11 Cảm biến dò line đơn TCRT5000

Cảm biến dò line có thể dùng để phát hiện line trắng và đen. Mạch sử dụng cảm biến hồng ngoại TCRT5000 với khoảng cách phát hiện từ 1~25mm. Có thể điều chỉnh độ nhạy của cảm biến bằng cách điều chỉnh biến trở được tích hợp sẵn trên board.

Ứng dụng làm robot dò đường đi theo line, phát hiện màu sắc khác với màu mặc định từ trước,…

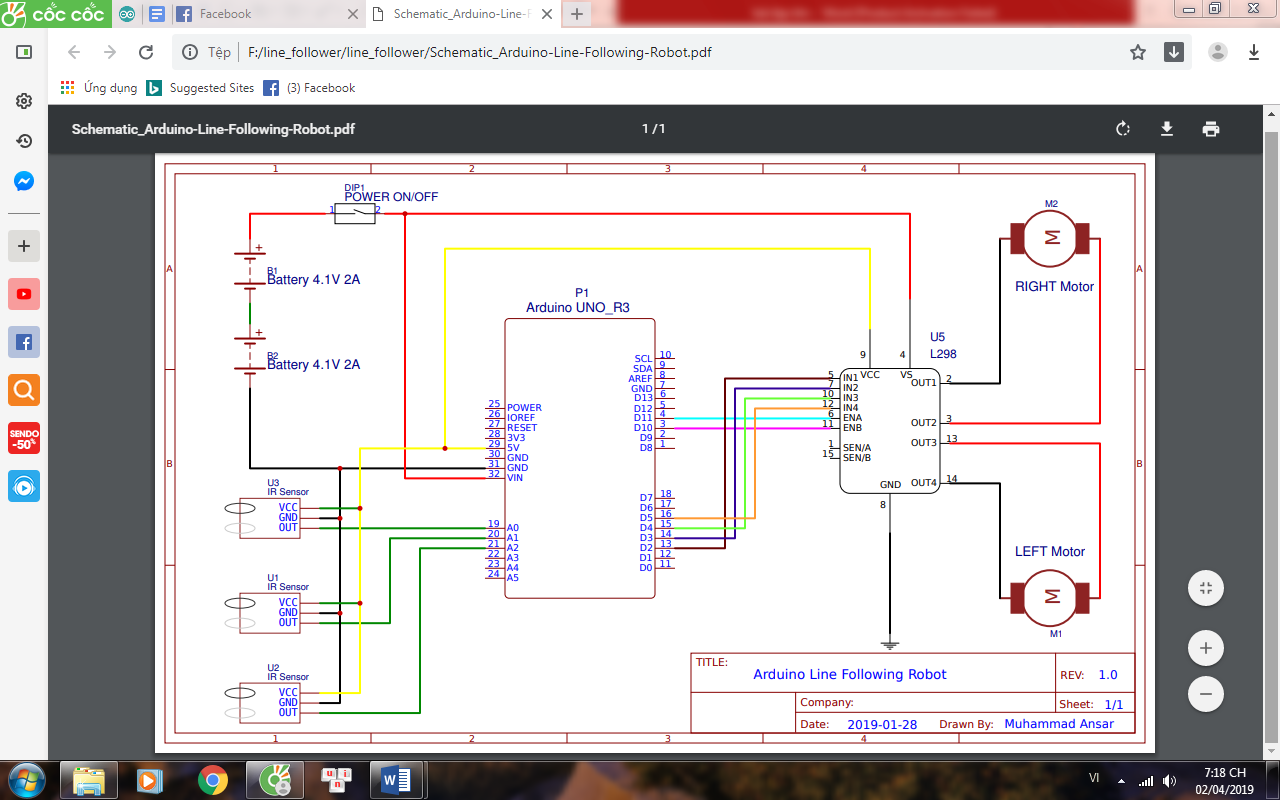
**Thông số cơ bản:**

* Nguồn cung cấp: 5V.
* Mạch sử dụng chip so sánh LM393
* Dòng điện tiêu thụ: <10mA.
* Dải nhiệt độ hoạt động: 0 độ ​C ~ 50 độ C
* Ngõ ra giao tiếp: 4 dây (VCC, GND, AO, DO)
* Kích thước: 4.2 cm x 1.0 cm x 0.3 cm

**Thứ tự chân:**

* Chân AO: tín hiệu Analog
* Chân DO: tín hiệu Digital
* Chân VCC: Nguồn
* Chân GND

**2. Sơ đồ mạch và cách nối dây**



**3. Sản phẩm**

**4. Mã nguồn**

int S\_A = 11; //speed motor a

int M\_A1 = 2; //motor a = +

int M\_A2 = 3; //motor a = -

int M\_B1 = 4; //motor b = -

int M\_B2 = 5; //motor b = +

int S\_B = 10; //speed motor b

int R\_S = A3; //sincer R

int S\_S = A4; //sincer S

int L\_S = A5 ; //sincer L

void setup()

{

pinMode(M\_B1, OUTPUT); //set chân M\_B1 của động cơ B là output

pinMode(M\_B2, OUTPUT); //set chân M\_B2 của động cơ B là output

pinMode(M\_A1, OUTPUT); //set chân M\_A1 của động cơ A là output

pinMode(M\_A2, OUTPUT); //set chân M\_A2 của động cơ A là output

pinMode(S\_B, OUTPUT); //set chân S\_B của động cơ B là output

pinMode(S\_A, OUTPUT); //set chân S\_A của động cơ A là output

pinMode(L\_S, INPUT); //set chân L\_S của cảm biến L là input

pinMode(S\_S, INPUT); //set chân S\_S của cảm biến S là input

pinMode(R\_S, INPUT); //set chân R\_S của cảm biến R là input

analogWrite(S\_A, 150); // ghi dữ liệu analog cho chân S\_A

analogWrite(S\_B, 150);

delay(200);

}

void loop()

{

if ((digitalRead(L\_S) == 0)&&(digitalRead(S\_S) == 1)&&(digitalRead(R\_S) == 0)){forward();} // đi thẳng

if ((digitalRead(L\_S) == 1)&&(digitalRead(S\_S) == 1)&&(digitalRead(R\_S) == 0)){turnLeft();} // rẽ trái

if ((digitalRead(L\_S) == 1)&&(digitalRead(S\_S) ==0)&&(digitalRead(R\_S) == 0)) {turnLeft();} // rẽ trái

if ((digitalRead(L\_S) == 0)&&(digitalRead(S\_S) == 1)&&(digitalRead(R\_S) == 1)){turnRight();} // rẽ phải

if ((digitalRead(L\_S) == 0)&&(digitalRead(S\_S) == 0)&&(digitalRead(R\_S) == 1)){turnRight();} // rẽ phải

if ((digitalRead(L\_S) == 1)&&(digitalRead(S\_S) ==

1)&&(digitalRead(R\_S) == 1)){Stop();} // dừng lại

}

void forward()

{

digitalWrite(M\_A1, LOW);

digitalWrite(M\_A2, HIGH);

digitalWrite(M\_B1, HIGH);

digitalWrite(M\_B2, LOW);

}

void turnRight()

{

digitalWrite(M\_A1, LOW);

digitalWrite(M\_A2, LOW);

digitalWrite(M\_B1, HIGH);

digitalWrite(M\_B2, LOW);

}

void turnLeft()

{

digitalWrite(M\_A1, LOW);

digitalWrite(M\_A2, HIGH);

digitalWrite(M\_B1, LOW);

digitalWrite(M\_B2, LOW);

}

void Stop()

{

digitalWrite(M\_A1, LOW);

digitalWrite(M\_A2, LOW);

digitalWrite(M\_B1, LOW);

digitalWrite(M\_B2, LOW);

}