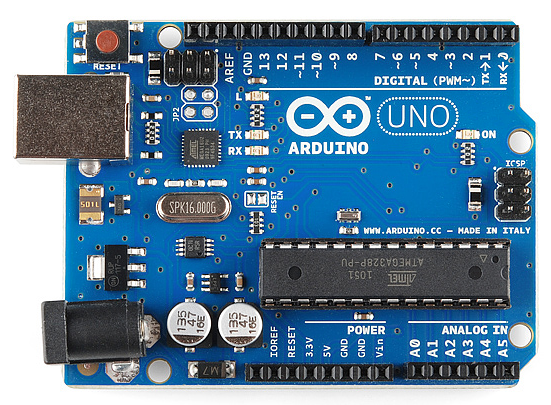
**Trường Đại Học Nha Trang**

Khoa Công Nghệ Thông Tin

****

**LẬP TRÌNH THIẾT BỊ NHÚNG ARDUINO**

****

**Học phần : Lập trình thiết bị nhúng**

**GVHD: Mai Cường Thọ**

**SVTH: Huỳnh Ngọc Hưng**

**Lớp: 61.CNTT-1**

**MSSV: 61133707**

**Khóa học 2021-2022**

**Mục lục**

[**Chương 1: Tổng quan về các linh kiện cơ bản** 6](#_Toc90847651)

[**1.** **Mạch Arduino Uno R3** 6](#_Toc90847652)

[**2.** **Đèn led** 9](#_Toc90847653)

[**3.** **Nút nhấn** 10](#_Toc90847654)

[**4.Đèn RBG** 12](#_Toc90847655)

[**5.Cảm biến nhiệt độ TMP36** 14](#_Toc90847656)

[**6.Led 7 đoạn** 15](#_Toc90847657)

[**7.Chiếc áp (Biến trở)** 16](#_Toc90847658)

[**8.IC 74HC595** 17](#_Toc90847659)

[**9.Led matrix 8x8** 18](#_Toc90847660)

[**10.Cảm biến siêu âm HC-SR04** 19](#_Toc90847661)

[**11.IC Điều khiển động cơ L293D** 20](#_Toc90847662)

[**12.Động cơ DC** 21](#_Toc90847663)

[**13.Động cơ Micro Servo** 22](#_Toc90847664)

[**14.Màn hình led LCD 16x2** 23](#_Toc90847665)

[**15.Cảm biến hồng ngoại Ir Sensor** 24](#_Toc90847666)

[**16.Remote** 25](#_Toc90847667)

[**17.NPN Transistor** 25](#_Toc90847668)

[**Chương 2: Lập trình ARDUINO** 26](#_Toc90847669)

[**Bài 1: Nháy Led** 26](#_Toc90847670)

[**Bài 2:Bật/Tắt đèn Led** 27](#_Toc90847671)

[**Bai 3: Cảm biến nhiệt độ** 28](#_Toc90847672)

[**Bai 4:Đèn Led sáng từ 1-10** 29](#_Toc90847673)

[**Bai 5 :Điều khiển động cơ** 30](#_Toc90847674)

[**Bai 6 :Điều khiển Led qua chiết áp** 34](#_Toc90847675)

[**Bai 7 :Led 7 đoạn** 35](#_Toc90847676)

[**Bai 8 :Led RGB** 36](#_Toc90847677)

[**Bai 9: Liên tục đọc nhiệt độ môi trường và hiện ra LCD mỗi dây,lập trình ngắt nếu có ngắt thì bật đèn và ngược lại** 40](#_Toc90847678)

[**Bài 10 : LED 7 đoạn hiện ra các số đếm 0-9 ,00-99** 42](#_Toc90847679)

[**Bài 11: Động cơ Điều khiển 3 bánh sử dụng vật cản HC-SR4** 46](#_Toc90847680)

[**Bài 12 : Player** 50](#_Toc90847681)

[**Bai 13 Cảm biến chuyển động** 51](#_Toc90847682)

[**Bài 14 : LCD-CAM BIEN AM THANH** 53](#_Toc90847683)

[**Bai 15: Remote điều khiển đèn** 54](#_Toc90847684)

[**Chương 3 : STM 32** 56](#_Toc90847685)

[**Bài 1 : Nháy 4 LED** 56](#_Toc90847686)

[**Bài 2 : STM32 led trái tim** 59](#_Toc90847687)

**Danh mục hình**

[Hình 1: Mạch Arduino Uno R3 6](#_Toc90847777)

[Hình 2:Đèn led 9](#_Toc90847778)

[Hình 3:Nút nhấn 10](#_Toc90847779)

[Hình 4 Led RGB 12](#_Toc90847780)

[Hình 5 Cảm biến nhiệt độ STMP36 14](#_Toc90847781)

[Hình 6 Cấu tạo led 7 đoạn 15](#_Toc90847782)

[Hình 7 Cấu tạo chiếc áp 16](#_Toc90847783)

[Hình 8: IC74HC595 17](#_Toc90847784)

[Hình 9 Led matrix 8x8 18](#_Toc90847785)

[Hình 10 HC-SR04 19](#_Toc90847786)

[Hình 11 IC L293D 20](#_Toc90847787)

[Hình 12 Động cơ DC 21](#_Toc90847788)

[Hình 13 Cấu tạo Động cơ Servo 22](#_Toc90847789)

[Hình 14 Cấu tạo Led LCD 23](#_Toc90847790)

[Hình 15 Cảm biến hồng ngoại PIR 24](#_Toc90847791)

[Hình 16 Remote trên Tinkercard 25](#_Toc90847792)

[Hình 17 NPN Transistor 25](#_Toc90847793)

[Hình 1 : Nháy Led 26](#_Toc90847794)

[Hình 2: Bật/Tắt đèn Led 27](#_Toc90847795)

[*Hình 3: Cảm biến nhiệt độ* 28](#_Toc90847796)

[Hình 4 : Đèn Led sáng từ 1-10 29](#_Toc90847797)

[*Hình 5: Điều khiển động cơ* 31](#_Toc90847798)

[*Hình 6 : Điều khiển Led qua chiết áp* 34](#_Toc90847799)

[*Hình 7: Led 7 đoạn* 35](#_Toc90847800)

[*Hình 8 : Led RGB* 37](#_Toc90847801)

[Hình 9: Đọc nhiệt độ môi trường và hiện ra LCD 40](#_Toc90847802)

[Hình 10: LED 7 đoạn hiện ra các số đếm 0-9 ,00-99 43](#_Toc90847803)

[Hình 11: Động cơ Điều khiển 3 bánh sử dụng vật cản HC-SR4 47](#_Toc90847804)

[Hình 12: Player 50](#_Toc90847805)

[Hình 13: Cảm biến chuyển động 52](#_Toc90847806)

[Hình 14: LCD-CAM BIEN AM THANH 53](#_Toc90847807)

[Hình 15: Remote điều khiển 3 bóng đèn 55](#_Toc90847808)

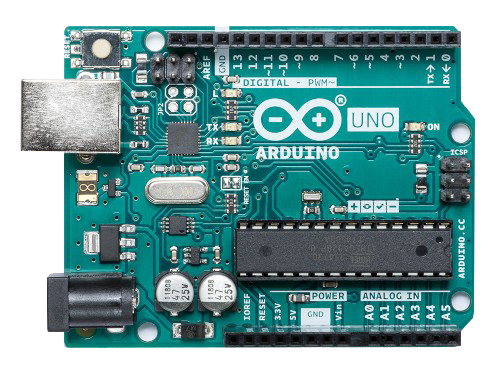
[Hình 16: Nháy 4 led với STM32 57](#_Toc90847809)

# **Chương 1: Tổng quan về các linh kiện cơ bản**

1. **Mạch Arduino Uno R3**

Arduino Uno là một bảng mạch vi điều khiển nguồn mở dựa trên vi điều khiển Microchip ATmega328 được phát triển bởi Arduino.cc. Bảng mạch được trang bị các bộ chân đầu vào/ đầu ra Digital và Analog có thể giao tiếp với các bảng mạch mở rộng khác nhau. Mạch Arduino Uno thích hợp cho những bạn mới tiếp cận và đam mê về điện tử, lập trình…Dựa trên nền tảng mở do Arduino.cc cung cấp các bạn dễ dàng xây dựng cho mình một dự án nhanh nhất ( lập trình Robot, xe tự hành, điều khiển bật tắt

led…)



.

#### 

#### Hình 1: Mạch Arduino Uno R3

|  |  |
| --- | --- |
| Chip điều khiển | [ATmega328P](http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf) |
| Điện áp hoạt động | 5V |
| Điện áp đầu vào(khuyên dùng) | 7-12V |
| Điện áp đầu vào (giới hạn) | 6-20V |
| Số chân Digital | 14 (of which 6 provide PWM output) |
| Số chân PWM Digital | 6 |
| Số chân Analog | 6 |
| Dòng điện DC trên mỗi chân I/O | 20 mA |
| Dòng điện DC trên chân 3.3V | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Tốc độ thạch anh | 16 MHz |
| LED\_BUILTIN | 13 |
| Chiều dài | 68.6 mm |
| Chiều rộng | 53.4 mm |
| Cân nặng | 25 g |

**Thông số kỹ thuật**

**Power**

* **LED:** Có 1 LED được tích hợp trên bảng mạch và được nối vào chân D13. Khi chân có giá trị mức cao (HIGH) thì LED sẽ sáng và LED tắt khi ở mức thấp (LOW).
* **VIN:** Chân này dùng để cấp nguồn ngoài (điện áp cấp từ 7-12VDC).
* **5V:** Điện áp ra 5V (dòng điện trên mỗi chân này tối đa là 500mA).
* **3V3:** Điện áp ra 3.3V (dòng điện trên mỗi chân này tối đa là 50mA).
* **GND:** Là chân mang điện cực âm trên board.
* **IOREF:** Điệp áp hoạt động của vi điều khiển trên **Arduino UNO** và có thể đọc điện áp trên chân IOREF. Chân IOREF không dùng để làm chân cấp nguồn.

**Bộ nhớ**

Vi điều khiển ATmega328:

**32 KB bộ nhớ Plash:** trong đó bootloader chiếm 0.5KB.

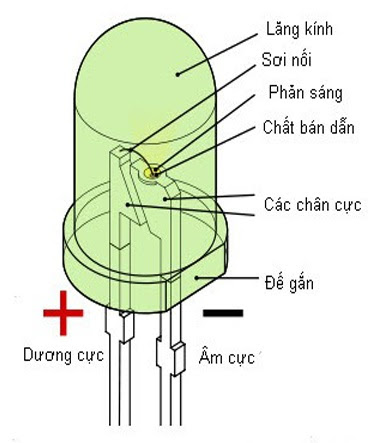
**2 KB cho SRAM:** (Static Random Access Menory): giá trị các biến khai báo sẽ được lưu ở đây. Khai báo càng nhiều biến thì càng tốn nhiều bộ nhớ RAM. Khi mất nguồn dữ liệu trên SRAM sẽ bị mất.

**1 KB cho EEPROM:** (Electrically Eraseble Programmable Read Only Memory): Là nơi có thể đọc và ghi dữ liệu vào đây và không bị mất dữ liệu khi mất nguồn.

**Các chân đầu vào và đầu ra**

Trên **Board Arduino Uno có 14 chân Digital** được sử dụng để làm chân đầu vào và đầu ra và chúng sử dụng các hàm pinMode(), digitalWrite(), digitalRead(). Giá trị điện áp trên mỗi chân là 5V, dòng trên mỗi chân là 20mA và bên trong có điện trở kéo lên là 20-50 ohm. Dòng tối đa trên mỗi chân I/O không vượt quá 40mA để tránh trường hợp gây hỏng board mạch.

1. **Đèn led**



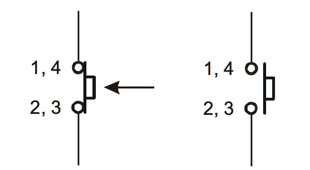
#### Hình 2:Đèn led

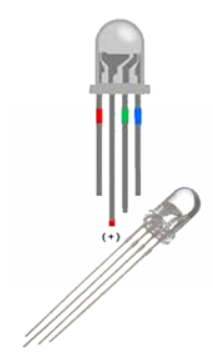
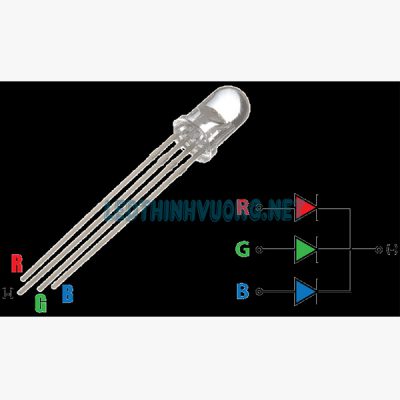
Đèn Led gồm hai chân, một chân nối vào cổng Digital của Arduino, chân còn lại nối vào cổng GND (cực âm của nguồn điện cấp cho Arduino UNO thường gọi là “Tiếp đất” nghĩa là bằng 0V)**.** Đảm bảo rằng dòng điện qua đèn Led không bao giờ lớn hơn 20mA.

1. **Nút nhấn**



#### Hình 3:Nút nhấn

* Nút ấn là một loại công tắc đơn giản điều khiển hoạt động của máy hoặc một số loại quá trình. Hầu hết, các nút nhấn là nhựa hoặc kim loại. Hình dạng của nút ấn có thể phù hợp với ngón tay hoặc bàn tay để sử dụng dễ dàng. Tất cả phụ thuộc vào thiết kế cá nhân. Nút ấn có 2 loại chính là nút nhấn thường mở hoặc nút nhấn thường đóng.
* **Nguyên lí làm việc của nút nhấn**
* Nút nhấn có ba phần: Bộ truyền động, các tiếp điểm cố định và các rãnh. Bộ truyền động sẽ đi qua toàn bộ công tắc và vào một xy lanh mỏng ở phía dưới. Bên trong là một tiếp điểm động và lò xo. Khi nhấn nút, nó chạm vào các tiếp điểm tĩnh làm thay đổi trạng thái của tiếp điểm. Trong một số trường hợp, người dùng cần giữ nút hoặc nhấn liên tục để thiết bị hoạt động. Với các nút nhấn khác, chốt sẽ giữ nút bật cho đến khi người dùng nhấn nút lần nữa.
* **Ứng dụng**
* Công tắc nút nhấn sử dụng nhiều trong các ứng dụng khác nhau như máy tính, điện thoại nút nhấn và nhiều thiết bị gia dụng. Bạn có thể nhìn thấy chúng trong nhà, văn phòng và trong các ứng dụng công nghiệp ngày nay. Chúng có thể bật, tắt máy hoặc làm cho thiết bị thực hiện các hoạt động cụ thể, như trường hợp với máy tính. Trong một số trường hợp, các nút nhấn có thể kết nối thông qua liên kết cơ học, điều khiển một nút nhấn khác hoạt động
* Đa số, các nút sẽ có màu sắc cụ thể để biểu thị mục đích của chúng. Ví dụ như nút nhất màu xanh thường được sử dụng để bật thiết bị hay nút nhấn màu đỏ để tắt thiết bị. Điều này tránh gây nên một sô nhầm lẫn. Nút dừng khẩn cấp thường là các nút ấn lớn, thường có màu đỏ và có đầu lớn hơn để sử dụng dễ dàng hơn.
* **Kí hiệu nút nhấn**
* 

**4.Đèn RBG**

#### Hình 4 Led RGB

Trong lĩnh vực điện tử thì bất kỳ màu sắc nào cũng được tạo ra từ 3 thành phần cơ bản. Đó là**màu đỏ (Red), màu xanh lá (Green), Màu xanh dương (Blue)**. Áp dụng trong kỹ thuật hiển thị màu sắc của điểm ảnh, người ta dùng cách phối hợp tỷ lệ về cường độ ánh sáng của 3 nguồn sáng cơ bản này để tạo ra các màu sắc khác nhau. Như các màu vàng, trắng ấm, hồng…

Led RGB là một dạng module duy nhất có thể cung cấp cả 3 màu đỏ, xanh lá, xanh dương. Đây chính là thành phần quan trọng để tạo ra đèn Led RGB. Với ánh sáng đổi màu được ứng dụng rất nhiều trong chiếu sáng, trang trí cảnh quan, công trình.

**Đặc điểm của đèn Led RGB**

Khác với các dòng Led thường, **Led RGB** có 4 chân, trong đó có 1 chân dương chung và 3 chân âm riêng cho từng màu (R – red – đỏ, G – Green – Xanh lá, B – Blue – Xanh dương). Có thể nói đây là sự kết hợp của 3 màu led và thay đổi màu sắc bằng việc điều khiển các chân của Led, thông thường ta sẽ dùng xung PWM.

**+Góc nhìn:**

Khi ta lại gần nguồn sáng thì góc nhìn sẽ rộng, do đó sẽ phân biệt được những chi tiết nhỏ, nên cần ghép 3 điểm lại gần với nhau thì việc phối hợp màu sẽ không bị phát hiện ra. Nhưng ở khoảng cách xa hay nguồn sáng hay bị che chắn bởi lớp vật liệu mờ (Mica, Kính bắn cát…) thì các điểm gần sẽ trở thành 1 điểm mà mắt không thể phân biệt được, nên ta có thể tạo ra sự phối hợp đa sắc màu bằng cách ghép những LED đơn có 3 màu cơ bản (R,G,B) để cho giá thành thấp hơn và linh hoạt hơn vào những ứng dụng đòi hỏi công suất sáng cao: Sàn nhảy, Hộp đèn đổi màu…

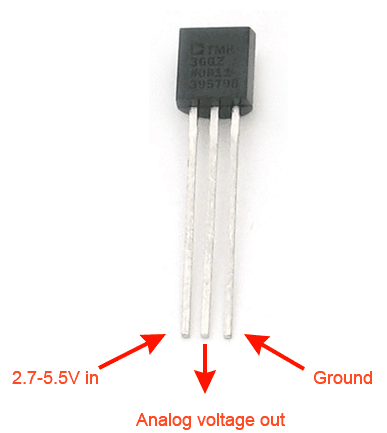
**+Thời Gian phối màu**:

Việc phối màu không nhất thiết ba màu cùng sáng vào một thời điểm, ta có thể cho sáng tuần tự các màu theo thời gian. Nên với đặc điểm này ta có thể dùng phương pháp thay đổi độ rộng xung (PMW) để biểu diễn cường độ sáng của từng màu. Điều này sẽ làm cho mạch điện trở nên đơn giản và chi phí rẻ.

**+Độ sáng**:

Những điểm sáng nhỏ, gần nhau thì công suất phát sáng nhỏ (màn hình TV có độ phân giải rất cao nhưng công suất không lớn). Còn ghép các đèn lại sẽ có công suất lớn đáp ứng nhu cầu sử dụng.

**5.Cảm biến nhiệt độ TMP36**



#### Hình 5 Cảm biến nhiệt độ STMP36

TMP36 là cảm biến nhiệt độ điện áp thấp, chính xác. Nó cung cấp một đầu ra điện áp tỷ lệ tuyến tính với nhiệt độ C. Nó cũng không yêu cầu bất kỳ hiệu chuẩn bên ngoài nào để cung cấp độ chính xác điển hình là ± 1 ° C ở + 25 ° C và ± 2 ° C trong phạm vi nhiệt độ −40 ° C đến + 125 ° C. Chúng tôi thích nó vì nó rất dễ sử dụng: Chỉ cần nối đất cho thiết bị và 2,7 đến 5,5 VDC và đọc điện áp trên chân Vout. Điện áp đầu ra có thể được chuyển đổi thành nhiệt độ một cách dễ dàng bằng cách sử dụng hệ số thang đo 10 mV / ° C.Arduino đọc được sau đó nhân với (5000/1024). Sau đó chúng ta phải trừ đi điện áp bù 400 và chia cho 10 sẽ ra kết quả là độ C.

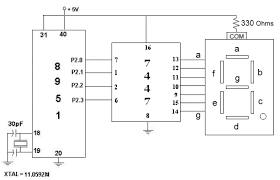
Đặc trưng

* Điện áp đầu vào: 2,7 V đến 5,5 VDC
* Hệ số thang đo 10 mV / ° C
* Độ chính xác ± 2 ° C theo nhiệt độ
* Độ tuyến tính ± 0,5 ° C
* Phạm vi hoạt động: −40 ° C đến + 125 ° C

*Cấu tạo:*

* Chân số 1 là chân cấp nguồn 5V (chân này có thể cắm vào nguồn 5V của Arduino).
* Chân thứ 2 là chân xuất tín hiệu tương tự (tín hiệu dạng xung).
* Chân thứ 3 là chân nối mát hay chân GND (khi sử dụng với Arduino các bạn có thể lấy từ chân GND từ Arduino).

**6.Led 7 đoạn**



#### Hình 6 Cấu tạo led 7 đoạn

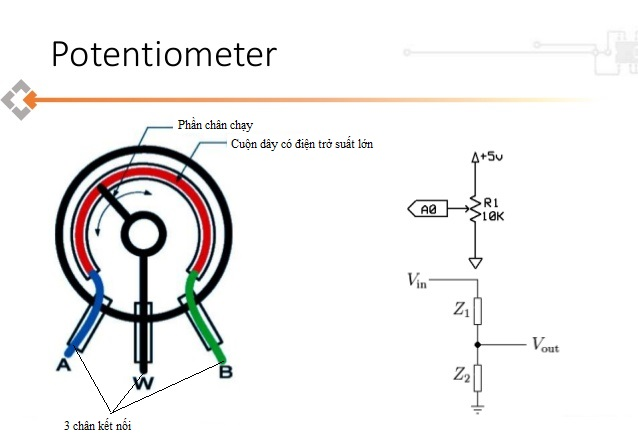
Led 7 đoạn thật chất chỉ là bảy đèn Led được kết nối song song. Mỗi đèn trong số bảy đèn Led được gọi là một đoạn vì khi được chiếu sáng, đoạn đó tạo thành một phần của chữ số (cả Thập phân và Hệ lục phân) sẽ được hiển thị. Đèn Led thứ 8 bổ sung bên cạnh vì đôi khi được sử dụng để chỉ báo dấu chấm thập phân. Mỗi đoạn led được đánh dấu từ a tới g. Đèn Led có bộ sung đèn thứ tám gọi là “chấm thập phân” (Decimal Point) ký hiệu DP được sử dụng khi hiển thị số không phải là số nguyên.

*Cấu tạo:*

* Loại dương (Common Anode): nếu cực dương (anode) của tất cả 8 LED được nối với nhau và các cực âm (cathode) đứng riêng lẻ.
* Loại âm (Common Cathode): nếu cực âm (cathode) của tất cả 8 LED được nối với nhau và các cực dương (anode) đứng riêng lẻ.

*Nguyên lý hoạt động*: Led nào sáng thì Led đó phải được phân cực thuận. Do đó muốn tạo ra chữ số nào ta chỉ cần cho Led ở các vị trí tương ứng sáng lên. Bảng mô tả cách tạo ra các chữ số để hiển thị lên LED 7 đoạn

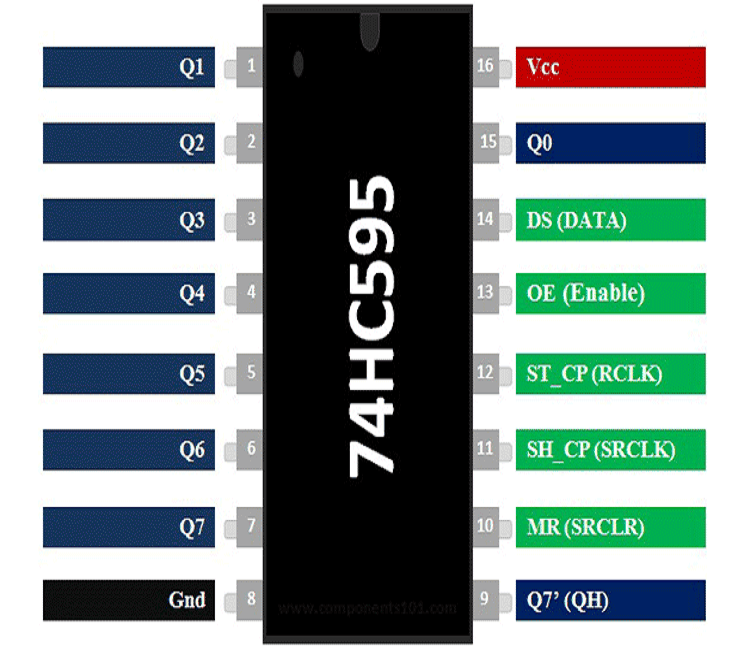
**7.Chiếc áp (Biến trở)**



#### Hình 7 Cấu tạo chiếc áp

Chiết áp là sự kết hợp với một biến trở tuyến tính có điện trở tối đa là 10KΩ. Khi bạn di chuyển thanh trượt từ bên này sang bên kia, điện áp đầu ra của nó sẽ nằm trong khoảng từ 0 V đến Vcc mà bạn áp dụng. Cấu tạo gồm ba chân OUT, GND, VCC như trong hình bên dưới.

**8.IC 74HC595**



#### Hình 8: IC74HC595

74HC595 là một thanh ghi dịch (shift register) hoạt động trên giao thức nối tiếp vào song song ra (Serial IN Parallel OUT).

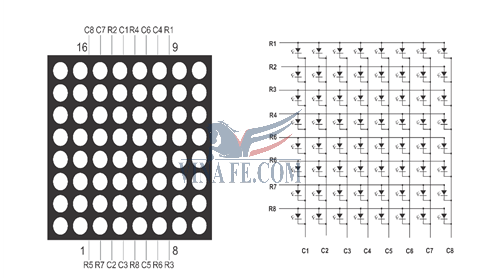
Nó nhận dữ liệu nối tiếp từ vi điều khiển và sau đó gửi dữ liệu này qua các chân song song.

Có thể tăng 8 chân đầu ra bằng cách sử dụng chip đơn.

Cũng có thể kết nối song song nhiều hơn 1 thanh ghi dịch.

Giả sử đã kết nối ba thanh ghi dịch với bộ vi điều khiển, các chân đầu ra được tăng lên 8 x 3 = 24.

**9.Led matrix 8x8**



#### Hình 9 Led matrix 8x8

Led matrix 8x8 dùng để tạo hiệu ứng hình ảnh trên Matrix, Vd: tạo chữ, số, tạo hiệu ứng chuyển động để làm biển chỉ dẫn, làm bảng quảng cáo,…

Led matrix 8x8 đơn màu thông thường có 16 chân, 8 cho mỗi hàng và 8 cho mỗi cột. Lý do cho tất cả các hàng và cột được nối dây với nhau là để giảm số lượng chân cần thiết.

*Phân loại:*

* Có 2 loại Matrix là Anode chung và Kathode chung. Điểm khác duy nhất của 2 loại là cách cấp nguồn cho 2 loại này là ngược nhau.
* Mỗi cột được kích hoạt trong một khoảng thời gian rất ngắn và đồng thời các đèn LED trên cột đó sáng lên bằng cách đặt địa chỉ cho hàng tương ứng. Các cột được chuyển đổi quá nhanh (hàng trăm hoặc hàng nghìn lần một giây) đến mức mắt người vẫn nhận thấy màn hình được chiếu sáng đầy đủ. Do đó, chỉ có tối đa tám đèn LED được thắp sáng cùng một lúc.

**10****.Cảm biến siêu âm HC-SR04**

#### Hình 10 HC-SR04

*Thông số kỹ thuật:*

* Điện áp làm việc: 5VDC
* Dòng điện: 15mA
* Tần số: 40 KHZ
* Khoảng cách phát hiện: 2cm – 4m
* Tín hiệu đầu ra: Xung mức cao 5V, mức thấp 0V
* Góc cảm biến: Không quá 15 độ.
* Độ chính xác cao: Lên đến 3mm
* Chế độ kết nối: VCC / Trig (T-Trigger) / Echo (R-Receive) / GND

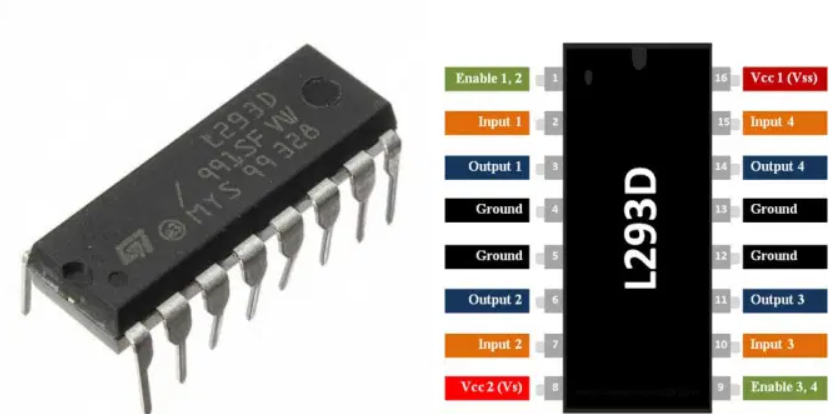
*Module cảm biến có 4 chân:*

* Chân VCC: Dùng để cấp nguồn 5v
* Chân Trig: Chân digital output
* Chân Echo: Chân digital input
* Chân GND: Chân 0v

*Nguyên lý hoạt động:*

Để đo khoảng cách bằng cảm biến siêu âm HC-SR04, ta sẽ phát 1 xung rất ngắn (5 microSeconds) từ chân Trig. Tiếp theo, một xung HIGH ở chân Echo sẽ được cảm biến tạo ra và phát đi cho đến khi nhận lại được sóng phản xạ ở chân này. Lúc này, độ rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và phản xạ lại. Trong không khí, tốc độ âm thanh đạt mức 340 m/s (hằng số), tương đương với 29,412 microSeconds/cm (106 / (340\*100)). Khi đã tính được thời gian, ta sẽ chia cho 29,412 để ra giá trị khoảng cách.

**11.IC Điều khiển động cơ L293D**



#### Hình 11 IC L293D

L293D là IC điều khiển động cơ 16 chân phổ biến. Một IC L293D có khả năng điều khiển hai động cơ DC cùng một lúc; cũng có thể điều khiển hướng của hai động cơ này một cách độc lập. Vì vậy, nếu bạn có các động cơ có điện áp hoạt động dưới 36V và dòng điện hoạt động dưới 600mA, được điều khiển bởi các mạch kỹ thuật số như Op-Amp, bộ định thời 555, cổng logic hoặc thậm chí các vi xử lý như Arduino, PIC, ARM, v.v. IC này sẽ là lựa chọn phù hợp với bạn.

*Thông số kỹ thuật:*

* Có thể được sử dụng để điều khiển 2 động cơ DC với cùng một lúc.
* Có thể điều khiển được tốc độ và chiều quay của động cơ.
* Điện áp động cơ Vcc2 (Vs): 4.5V đến 36V
* Dòng điện động cơ cực đại: 1.2A
* Dòng điện động cơ liên tực cực đại: 600mA
* Điện áp cung cấp cho Vcc1(vss): 4.5V đến 7V
* Thời gian chuyển tiếp: 300ns (ở 5V và 24V)

**12.Động cơ DC**



#### Hình 12 Động cơ DC

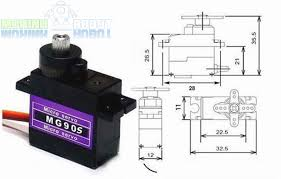
Động cơ DC hay còn gọi là động cơ điện một chiều chính là động cơ được điều khiển bằng dòng có hướng xác định. Cũng có thể nói dễ hiểu hơn thì đây là một loại động cơ chạy bằng nguồn điện áp DC – điện áp 1 chiều (Khác với những điện áp AC xoay chiều). Đầu dây ra của động cơ này thường gồm có hai dây (dây nguồn – VCC và dây tiếp đất – GND). DC motor là một loại động cơ một chiều với động cơ quay liên tục.

*Cấu tạo:*

Gồm có 3 phần chính stator( phần cảm), rotor ( phần ứng), và phần cổ góp- chỉnh lưu.

* Stator của động cơ điện 1 chiều thường sẽ là 1 hoặc nhiều những cặp nam châm vĩnh cửu, hoặc là nam châm điện.
* Rotor có những cuộn dây quấn và được nối với những nguồn điện một chiều.
* Bộ phận chỉnh lưu có nhiệm vụ là đổi chiều của dòng điện trong khi chuyển động quay của roto là liên tục. Thông thường thì bộ phận này gồm có một bộ cổ góp và một bộ chổi than để tiếp xúc với cổ góp.

**13.Động cơ Micro Servo**



#### Hình 13 Cấu tạo Động cơ Servo

Động cơ có nhiều loại cũng như có rất nhiều biến thể khác nhau và một trong số các biến thể đó là loại động cơ cho phép ta điều khiển tốc độ, góc quay, ... hay nói khác đi cho phép ta ra lệnh điều khiển và thực thi lệnh đó một cách cực kì chính xác - đó chính là động cơ servo.

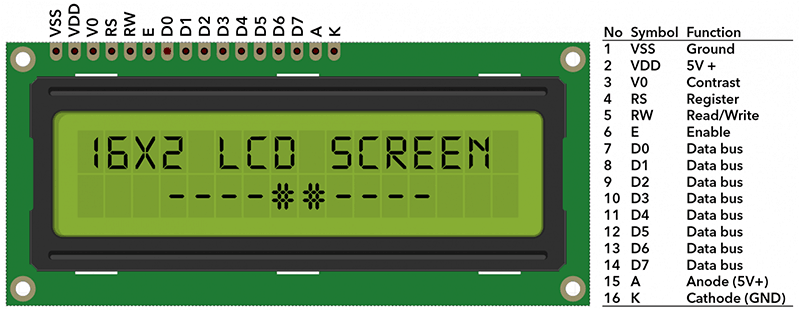
Động cơ servo cũng được chia làm nhiều loại, phụ thuộc vào góc quay tối đa của chúng. 2 loại phổ biến được sử dụng là:

* Động cơ Servo quay 180°: Futaba S3003, MG90[S],...
* Động cơ Servo quay 360°: MG995, MG996R,...

Động cơ servo là loại động cơ cho phép ta điều khiển một cách cực kì chính xác. Vì vậy, khác với động cơ thông thường ta chỉ cần cấp nguồn cho động cơ là có thể vận hành được. Động cơ servo yêu cầu ta phải cấp nguồn (2 dây) và nhận điều khiển từ mạch chính (1 dây), mỗi dây thường được đánh màu như sau:

* Đỏ: nối với nguồn
* Đen: nối với cực âm của mạch
* Vàng: nhận tín hiệu từ mạch điều khiển

**14.Màn hình led LCD 16x2**



#### Hình 14 Cấu tạo Led LCD

*Sơ đồ các chân của LCD 16x2:*

* VSS: tương đương với GND - cực âm
* VDD: tương đương với VCC - cực dương (5V)
* Constrast Voltage (Vo): điều khiển độ sáng màn hình
* Register Select (RS): điều khiển địa chỉ nào sẽ được ghi dữ liệu
* Read/Write (RW): Bạn sẽ đọc (read mode) hay ghi (write mode) dữ liệu? Nó sẽ phụ thuộc vào bạn gửi giá trị gì vào.
* Enable pin: Cho phép ghi vào LCD
* D0 - D7: 8 chân dư liệu, mỗi chân sẽ có giá trị HIGH hoặc LOW nếu bạn đang ở chế độ đọc (read mode) và nó sẽ nhận giá trị HIGH hoặc LOW nếu đang ở chế độ ghi (write mode)
* Backlight (Backlight Anode (+) và Backlight Cathode (-)): Tắt bật đèn màn hình LCD.

**15.Cảm biến hồng ngoại Ir Sensor**



#### Hình 15 Cảm biến hồng ngoại PIR

Tia hồng ngoại (IR) chính là các tia nhiệt phát ra trong khoảng các vật thể nóng. Trong những cơ thể sống, trong chúng ta luôn sở hữu thân nhiệt (thông thường là ở 37 độ C), và trong khoảng cơ thể chúng ta sẽ luôn phát ra những tia nhiệt, hay còn gọi là các tia hồng ngoại, người ta sẽ dùng 1 tế bào điện để chuyển đổi tia nhiệt ra dạng dấu hiệu điện và nhờ đó mà với thể khiến cho ra cảm biến phát hiện những vật thể nóng đang di chuyển.

*Nguyên tắc hoạt động của cảm biến hồng ngoại:*

* Cảm biến hồng ngoại sẽ hoạt động bằng cách sử dụng một cảm biến ánh sáng cụ thể để phát hiện bước sóng ánh sáng chọn trong phổ hồng ngoại (IR). Bằng cách sử dụng đèn LED tạo ra ánh sáng có cùng bước sóng với cảm biến đang tìm kiếm, bạn có thể xem cường độ của ánh sáng nhận được. Khi một vật ở gần cảm biến, ánh sáng từ đèn LED bật ra khỏi vật thể và đi vào cảm biến ánh sáng. Điều này dẫn đến một bước nhảy lớn về cường độ, mà chúng ta đã biết có thể được phát hiện bằng cách sử dụng một ngưỡng.
* Cảm biến Pir sensor là cảm biến thụ động phát hiện tia hồng ngoại phát ra từ môi trường xung quanh, được ứng dụng trong rất nhiều lĩnh vực trong đời sống, phổ biến là được ứng dụng trong hệ thống cửa tự động.

IR Sensor trong Tinkercad gồm 3 chân. Chân đỏ nối với nguồn, chân đen nối với cồng GND, chân xanh là chân OUTPUT.

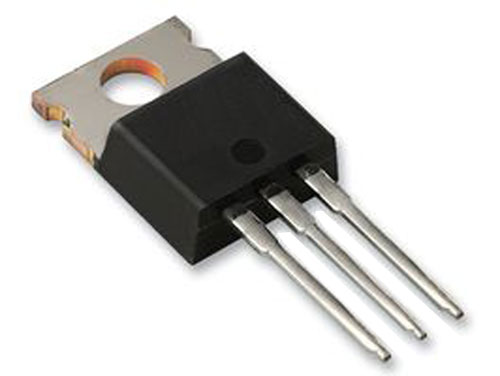
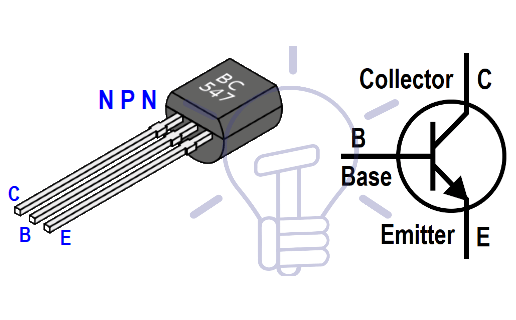
**16.Remote**



#### Hình 16 Remote trên Tinkercard

Trên Remote có 1 hoặc nhiều LEDs hồng ngoại được sử dụng để truyền tín hiệu hồng ngoại. Tín hiệu này sẽ được nhận bởi 1 bộ thu hồng ngoại đặc biệt và chuyển thành dạng xung điện, sau đó các xung điện này được chuyển đổi thành dữ liệu được sử dụng cho các thiết bị điện tử

**17.NPN Transistor**



#### Hình 17 NPN Transistor

Transistor NPN có hai điốt được kết nối trở lại. Các diode ở phía bên trái được gọi là một diode phát cơ sở và các điốt ở phía bên trái được gọi là diode collector-base. Những tên này được đưa ra theo tên của các thiết bị đầu cuối.

Transistor NPN có ba thiết bị đầu cuối, đó là bộ phát, bộ thu và cơ sở. Phần giữa của bóng bán dẫn NPN được pha tạp nhẹ,

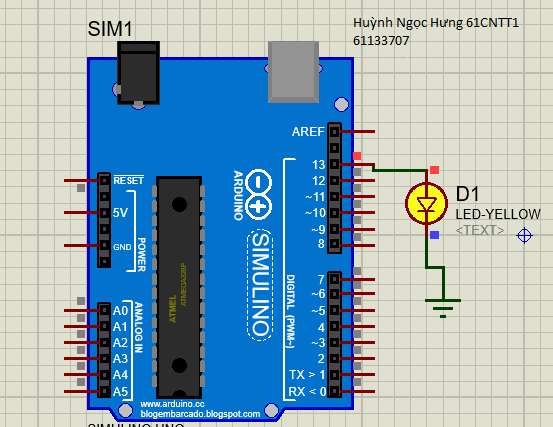
# **Chương 2: Lập trình ARDUINO**

## **Bài 1: Nháy Led**

**1.Mô tả**

Bài này thực hiện nháy đèn led sau khoảng thời gian là 1 giây. Led được nối Arduino cổng số 13

**2.Sơ đồ thiết kế**



#### Hình 1 : Nháy Led

**3.Đặc điểm linh kiện.**

* Led: màu vàng
* Điện trở: 100 Ω

**4.Mã lệnh chính**

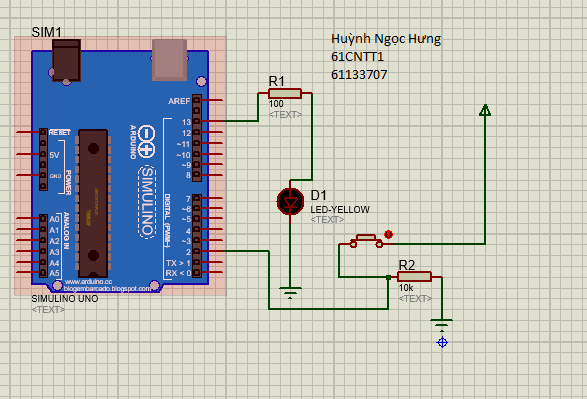
|  |
| --- |
| **int pin =13;**  **void setup()** {  pinMode(pin,OUTPUT);  }  **void loop(){**  digitalWrite(pin,HIGH);  delay(1000);  digitalWrite(LED\_BUILTIN,LOW);  delay(1000);  } |

## **Bài 2:Bật/Tắt đèn Led**

**1.Mô tả**

Hệ thông bật đèn thông qua nút bấm, đèn được kết nói Arduino ở cổng số 13, đầu ra nút bấm được kết nối vào pin 2.Đèn sáng khi được bật công tắc và ngược lại

**2.Sơ đồ thiết kế**

****

#### Hình 2: Bật/Tắt đèn Led

**3.Đặc điểm linh kiện**

* Led: màu vàng
* Điện trở: 2(10k)
* Nút nhấn(Push Button)

**4.Mã lệnh chính**

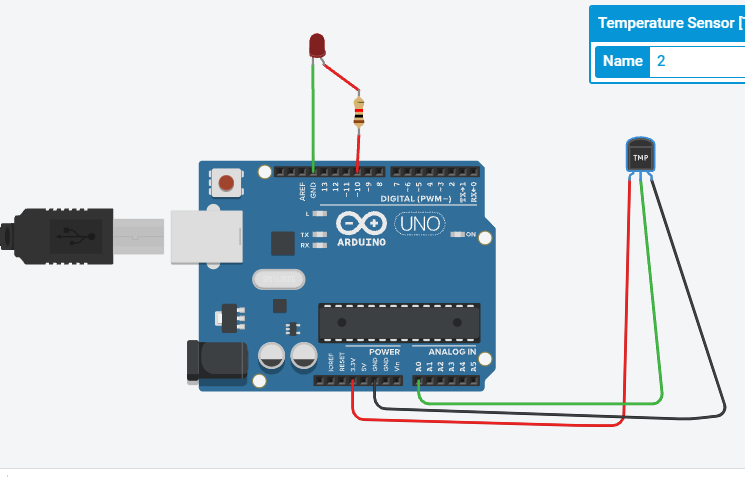
|  |
| --- |
| int x = 0;  **void setup()** {  pinMode(2,INPUT);  pinMode(13,OUTPUT);  }  **void loop()** {  x = digitalRead(2);  **if** (x==HIGH) {  //Bac led  digitalWrite(13,HIGH);  }  **else** {  //tat led  digitalWrite(13,LOW);  }  delay(1000);  } |

## **Bai 3: Cảm biến nhiệt độ**

1. **Mô tả**

Đọc nhiệt độ môi trường với cảm biến TMP36,khi nhiệt độ quá 370 thì sang Led, Led được nối vào cổng (pin 10),TMP(pin A0)

1. **Sơ đồ thiết kế**

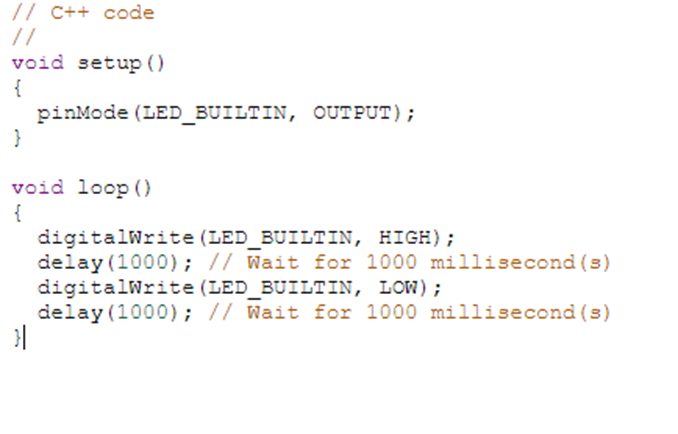


*Hình 3: Cảm biến nhiệt độ*

1. **Đặc điểm linh kiện**

Cảm biến TMP 36 này xuất tín hiệu là tín hiệu điện áp, tín hiệu này tỷ lệ thuận với nhiệt độ. Cứ 10mV tương đương với 1 độ C, nó có thể đo được nhiệt độ từ -40 ==>125 độ C

1. **Mã lệnh chính**

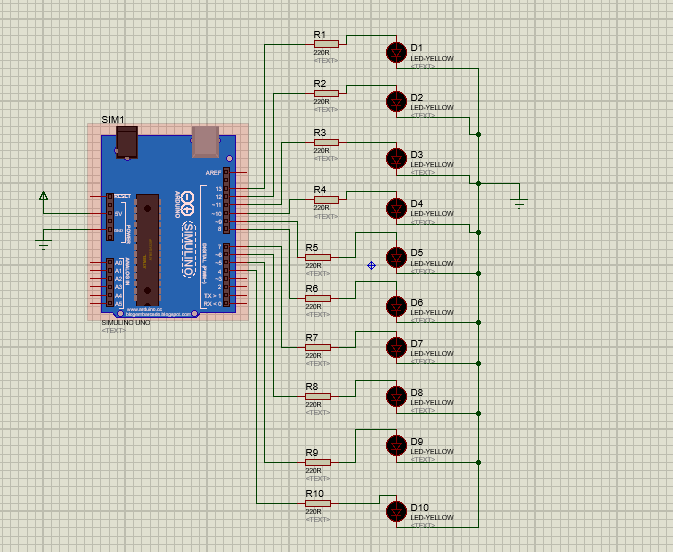
****

## **Bai 4:Đèn Led sáng từ 1-10**

**1.Mô tả**

-Đèn led được nối với Arduino,đèn sáng từ 1 đến 10 và ngược lại

**2.Sơ đồ thiết kế**

****

#### Hình 4 : Đèn Led sáng từ 1-10

**3.Linh kiện**

+10 LED màu vàng

+ 10 Điện trở

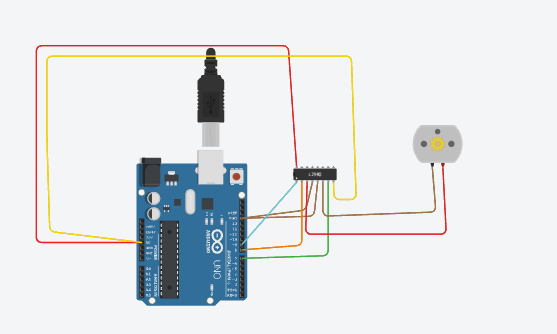
**4.Mã lệnh chính**

|  |
| --- |
| byte ledPin[] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13};  int direction = 1;  int currentLED = 0;  **void setup()** {  for (int x=0; x<10; x++) {  pinMode(ledPin[x], OUTPUT);} }  **void loop()** {  for (int x=0; x<10; x++) {  digitalWrite(ledPin[x], LOW);  }  digitalWrite(ledPin[currentLED], HIGH);  currentLED += direction;  if (currentLED == 9) {direction = -1;}  if (currentLED == 0) {direction = 1;}  **delay(500);**  } |

**Bai 5 :Điều khiển động cơ**

**1.Mô tả**

**2.Sơ đồ thiết kế**



*Hình 5: Điều khiển động cơ*

**3.Linh kiện**

+L293D

+DC motor

**4.Mã lệnh chính**

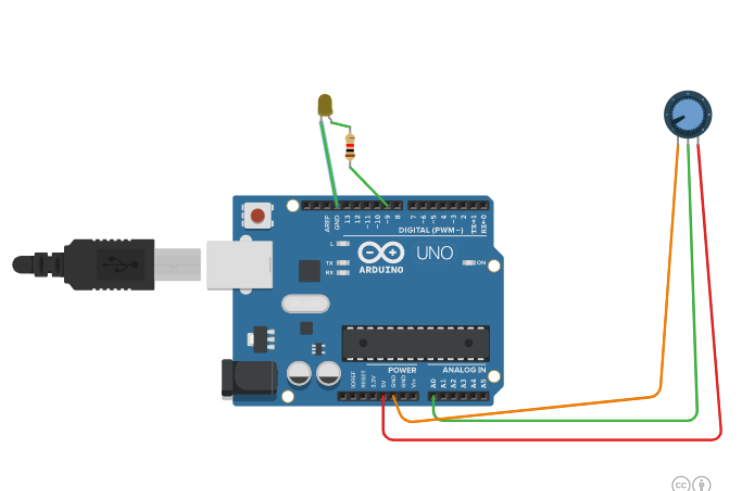
|  |
| --- |
| int inches = 0;  int cm = 0;  long readUltrasonicDistance(int triggerPin, int echoPin)  {  pinMode(triggerPin, OUTPUT); // Clear the trigger  digitalWrite(triggerPin, LOW);  delayMicroseconds(2);  // Sets the trigger pin to HIGH state for 10 microseconds  digitalWrite(triggerPin, HIGH);  delayMicroseconds(10);  digitalWrite(triggerPin, LOW);  pinMode(echoPin, INPUT);  digitalWrite(LED\_BUILTIN, HIGH);  delay(1000); // Wait for 1000 millisecond(s)  digitalWrite(LED\_BUILTIN, LOW);  delay(1000); // Wait for 1000 millisecond(s)  // Reads the echo pin, and returns the sound wave travel time in microseconds  return pulseIn(echoPin, HIGH);  }  void setup()  {  Serial.begin(9600);  pinMode(LED\_BUILTIN, OUTPUT);  }  void loop()  {  // measure the ping time in cm  cm = 0.01723 \* readUltrasonicDistance(7, 7);  // convert to inches by dividing by 2.54  inches = (cm / 2.54);  Serial.print(inches);  Serial.print("in, ");  Serial.print(cm);  Serial.println("cm");  delay(100); // Wait for 100 millisecond(s)  **}** |

**Bai 6 :Điều khiển Led qua chiết áp**

**1.Mô tả**

Hệ thống nhúng được thiết kế cho phép điều khiển của led(pin9) thông qua 1 biến trở gắn ở chân A0

**2.Sơ đồ thiết kế**



*Hình 6 : Điều khiển Led qua chiết áp*

**3.Linh kiện**

+Chiết áp

+Đen led

+ Điện trở

**4.Mã lệnh chính**

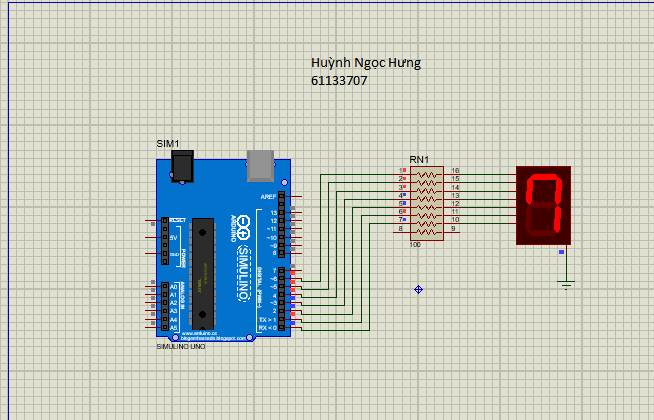
|  |
| --- |
| // C++ code  //  int x = 0;  **void setup()**  {  pinMode(A0, OUTPUT);  pinMode(9, OUTPUT);  }  **void loop()**  {  x = analogRead(A0);  int brightness = map(x,0,1023,0,255);  analogWrite(9,brightness);  } |

**Bai 7 :Led 7 đoạn**

**1.Mô tả**

Led 7 đoạn là 7 đèn led được sắp xếp thành hình chữ nhật như hình bên dưới

Mỗi led là một đoạn. Khi mỗi đoạn chiếu sáng thì một phần của chữ số (hệ thập phân hoặc thập lục phân) sẽ được hiển thị. Đôi khi có thêm led thứ 8 để biểu thị dấu thập phân khi có nhiều led 7 đoạn nối với nhau để hiển thị các số lớn hơn 1

**2.Sơ đồ thiết kế **

*Hình 7: Led 7 đoạn*

**3.Linh kiện**

+Điện trở

+ 7 segment dislay

**4.Mã lệnh chính**

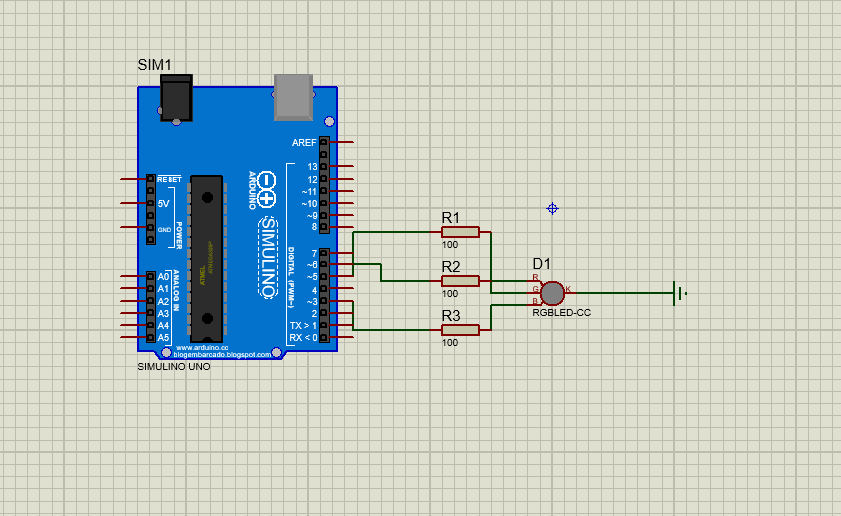
|  |
| --- |
| **int a=6, b=5, c=4, d=3, e=2,f=1,g=0;**  **void setup(){**  **pinMode(a,OUTPUT); pinMode(b,OUTPUT); pinMode(c,OUTPUT);**  **pinMode(d,OUTPUT); pinMode(e,OUTPUT); pinMode(f,OUTPUT); pinMode(g,OUTPUT);}**  **void MOT(){**  **digitalWrite(a,LOW); digitalWrite(b,HIGH); digitalWrite(c,HIGH);**  **digitalWrite(d,LOW); digitalWrite(e,LOW); digitalWrite(f,LOW); digitalWrite(g,LOW);}**  **void HAI(){**  **digitalWrite(a,HIGH); digitalWrite(b,HIGH); digitalWrite(c,LOW);**  **digitalWrite(d,HIGH); digitalWrite(e,HIGH); digitalWrite(f,LOW); digitalWrite(g,HIGH);}**  **void loop(){**  **MOT();**  **delay(1000);**  **HAI();**  **delay(1000);}** |

**Bai 8 :Led RGB**

**1.Mô tả**

**Đèn LED RGB** là hệ thống chiếu sáng có sự phối hợp của 3 nguồn ánh sáng cơ bản là **đỏ**(Red), **xanh lá** (Green)và **xanh dương**(Blue).

**2.Sơ đồ thiết kế**

****

*Hình 8 : Led RGB*

**3.Linh kiện**

+3 điện trở

+1 LED RGB

**4.Mã lệnh chính**

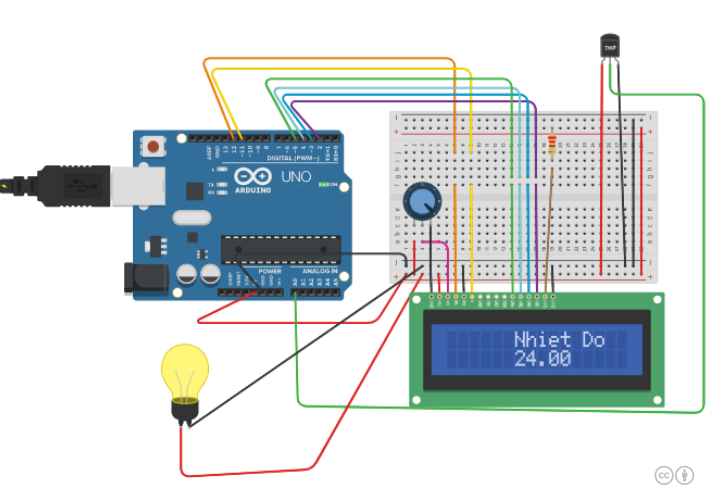
|  |
| --- |
| int a=13, b=12, c=11, d=10;  void setup() {  pinMode(a, OUTPUT);  pinMode(b, OUTPUT);  pinMode(c, OUTPUT);  pinMode(d, OUTPUT);  }  void Khong(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Mot(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void Hai(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,HIGH); digitalWrite(a,LOW);  }  void Ba(){  digitalWrite(d,LOW); digitalWrite(c,LOW);  digitalWrite(b,HIGH); digitalWrite(a,HIGH);  }  void Bon(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Nam(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void Sau(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,HIGH); digitalWrite(a,LOW);  }  void Bay(){  digitalWrite(d,LOW); digitalWrite(c,HIGH);  digitalWrite(b,HIGH); digitalWrite(a,HIGH);  }  void Tam(){  digitalWrite(d,HIGH); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,LOW);  }  void Chin(){  digitalWrite(d,HIGH); digitalWrite(c,LOW);  digitalWrite(b,LOW); digitalWrite(a,HIGH);  }  void loop() {  Khong();  delay(1000); // đợi 1 giây  Mot();  delay(1000);// đợi 1 giây  Hai();  delay(1000);// đợi 1 giây  Ba();  delay(1000);// đợi 1 giây  Bon();  delay(1000);// đợi 1 giây  Nam();  delay(1000);// đợi 1 giây  Sau();  delay(1000);// đợi 1 giây  Bay();  delay(1000);// đợi 1 giây  Tam();  delay(1000);// đợi 1 giây  Chin();  delay(1000);// đợi 1 giây  } |

## **Bai 9: Liên tục đọc nhiệt độ môi trường và hiện ra LCD mỗi dây,lập trình ngắt nếu có ngắt thì bật đèn và ngược lại**

**1.Mô ta**

-Sử dụng TMP36 đọc nhiệt độ môi trời và hiện ra trên LCD

**2.Sơ đồ**



#### Hình 9: Đọc nhiệt độ môi trường và hiện ra LCD

**3.Linh kiên**

+Đèn

+TMP36

**4.Mã lệnh chính**

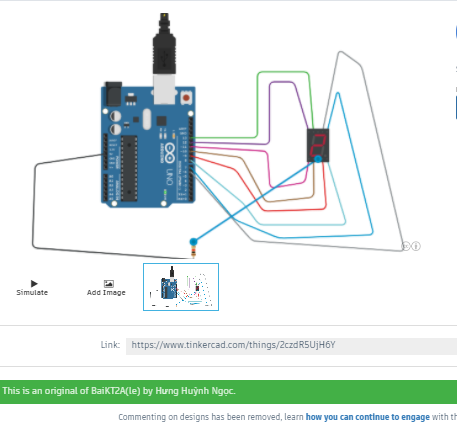
|  |
| --- |
| // include the library code:  **#include <LiquidCrystal.h>**  // initialize the library with the numbers of the interface pins  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  **void setup() {**  Serial.begin(96000);  lcd.begin(16, 2);  lcd.setCursor(6, 0);  // Print a message to the LCD.  lcd.print("Nhiet Do");  }  **void loop() {**  int GiaTri = analogRead(A0);  float NhietDo=map(GiaTri,20,358,-40,125);  //Serial.print(NhietDo);  lcd.setCursor(6, 1);    lcd.print(NhietDo);    } |

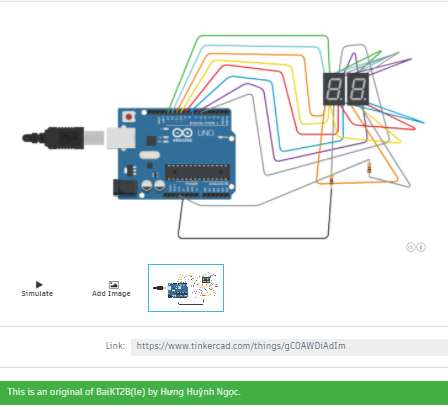
## **Bài 10 : LED 7 đoạn hiện ra các số đếm 0-9 ,00-99**

**1.Mô ta**

LED 7 đoạn có cấu tạo bao gồm 7 led đơn có dạng thanh xếp theo hình dạng số 8 và có thêm một led đơn hình tròn nhỏ thể hiện dấu chấm tròn ở góc dưới. 8 led đơn trên LED 7 đoạn có Anode(cực +) hoặc Cathode(cực -) được nối chung với nhau vào một điểm. Nếu LED 7 đoạn có Anode(cực +) chung, điểm chung này được nối với VCC, led chỉ sáng khi tín hiệu điều khiển ở các chân điều khiển trạng thái sáng tắt của các led đơn ở mức 0

**2.Sơ đồ thiết kế**





#### Hình 10: LED 7 đoạn hiện ra các số đếm 0-9 ,00-99

**3.Linh kiện**

+Điện trở

+**7 segment dislay**

**4.Mã lệnh chính**

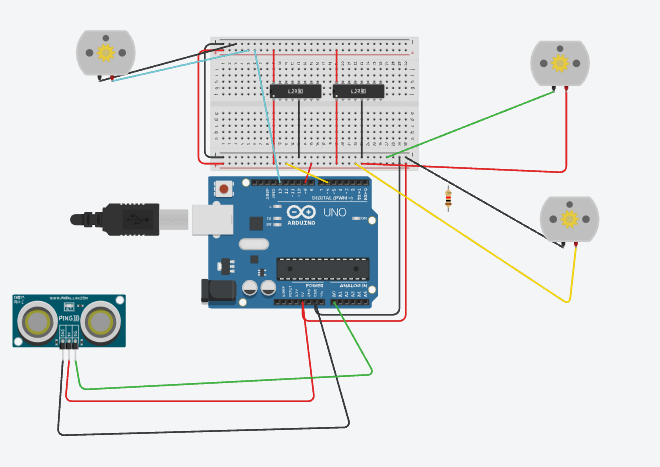
|  |
| --- |
| int a=6 , b=7 , c=9, d=10, e=11, f=13, g=12;  **void setup()**  {  pinMode(a, OUTPUT);  pinMode(b, OUTPUT);  pinMode(c, OUTPUT);  pinMode(d, OUTPUT);  pinMode(e, OUTPUT);  pinMode(g, OUTPUT);  pinMode(f, OUTPUT);  }  **void KHONG(){**  digitalWrite (a,LOW);  digitalWrite (b,LOW);  digitalWrite (c,LOW);  digitalWrite (d,LOW);  digitalWrite (g,HIGH);  digitalWrite (e,LOW);  digitalWrite (f,LOW);  }  **void MOT(){**  digitalWrite (a,HIGH);  digitalWrite (b,LOW);  digitalWrite (c,LOW);  digitalWrite (d,HIGH);  digitalWrite (g,HIGH);  digitalWrite (e,HIGH);  digitalWrite (f,HIGH);  }  **void HAI(){**  digitalWrite (a,LOW);  digitalWrite (b,LOW);  digitalWrite (c,HIGH);  digitalWrite (d,LOW);  digitalWrite (g,LOW);  digitalWrite (e,LOW);  digitalWrite (f,HIGH);  }  **void BA(){**  digitalWrite (a,LOW);  digitalWrite (b,LOW);  digitalWrite (c,LOW);  digitalWrite (d,LOW);  digitalWrite (g,LOW);  digitalWrite (e,HIGH);  digitalWrite (f,HIGH);  }  **void BON(){**  digitalWrite (a,HIGH);  digitalWrite (b,LOW);  digitalWrite (c,LOW);  digitalWrite (d,HIGH);  digitalWrite (g,LOW);  digitalWrite (e,HIGH);  digitalWrite (f,LOW);  }  **void NAM(){**  digitalWrite (a,LOW);  digitalWrite (b,HIGH);  digitalWrite (c,LOW);  digitalWrite (d,LOW);  digitalWrite (g,LOW);  digitalWrite (e,HIGH);  digitalWrite (f,LOW);  }  **void SAU(){**  digitalWrite (a,LOW);  digitalWrite (b,HIGH);  digitalWrite (c,LOW);  digitalWrite (d,LOW);  digitalWrite (g,LOW);  digitalWrite (e,LOW);  digitalWrite (f,LOW);  }  **void BAY(){**  digitalWrite (a,LOW);  digitalWrite (b,LOW);  digitalWrite (c,LOW);  digitalWrite (d,HIGH);  digitalWrite (g,HIGH);  digitalWrite (e,HIGH);  digitalWrite (f,HIGH);  }  **void TAM(){**  digitalWrite (a,LOW);  digitalWrite (b,LOW);  digitalWrite (c,LOW);  digitalWrite (d,LOW);  digitalWrite (g,LOW);  digitalWrite (e,LOW);  digitalWrite (f,LOW);  }  **void CHIN(){**  digitalWrite (a,LOW);  digitalWrite (b,LOW);  digitalWrite (c,LOW);  digitalWrite (d,LOW);  digitalWrite (g,LOW);  digitalWrite (e,HIGH);  digitalWrite (f,LOW);  }  **void loop(){**  **KHONG();**  delay(1000);  **MOT();**  delay(1000);  **HAI();**  delay(1000);  **BA();**  delay(1000);  **BON();**  delay(1000);  **NAM();**  delay(1000);  **SAU();**  delay(1000);  **BAY();**  delay(1000);  **TAM();**  delay(1000);  **CHIN();**  delay(1000);  **}** |

## **Bài 11: Động cơ Điều khiển 3 bánh sử dụng vật cản HC-SR4**

**1.Mô ta**

-Đông cơ điều khiển để tránh vật cản

**2.Sơ đồ thiết kế**



#### Hình 11: Động cơ Điều khiển 3 bánh sử dụng vật cản HC-SR4

**3.Linh kiện**

+3 DC Motor

+ HC-SR-04

**4.Mã lệnh chính**

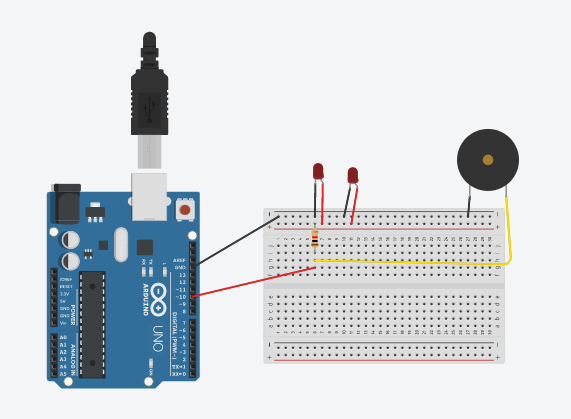
|  |
| --- |
| // C++ code  //  int inches = 0;  int cm = 0;  long readUltrasonicDistance(int triggerPin, int echoPin)  {  pinMode(triggerPin, OUTPUT); // Clear the trigger  digitalWrite(triggerPin, LOW);  delayMicroseconds(2);  // Sets the trigger pin to HIGH state for 10 microseconds  digitalWrite(triggerPin, HIGH);  delayMicroseconds(10);  digitalWrite(triggerPin, LOW);  pinMode(echoPin, INPUT);  return pulseIn(echoPin, HIGH);  }  **void setup()**  {  Serial.begin(9600);  }  **void loop()**  {  // measure the ping time in cm  cm = 0.01723 \* readUltrasonicDistance(7, 7);  // convert to inches by dividing by 2.54  inches = (cm / 2.54);  Serial.print(inches);  Serial.print("in, ");  Serial.print(cm);  Serial.println("cm");  delay(100); // Wait for 100 millisecond(s)  } |

## **Bài 12 : Player**

**1.Mô tả**

-Sử dụng piezo phát ra nhạc

**2.Sơ đồ thiết kế**



#### Hình 12: Player

**3.Linh kiện**

2 đèn led

1 piezo

**4.Mã lệnh chính**

|  |
| --- |
| **// C++ code**  **//**  **int t = 0;**  **void setup()**  **{**  **pinMode(10, OUTPUT);**  **}**  **void loop()**  **{**  **tone(10, 262, 300); // play tone 48 (C4 = 262 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 262, 300); // play tone 48 (C4 = 262 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 392, 300); // play tone 55 (G4 = 392 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 392, 300); // play tone 55 (G4 = 392 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 440, 300); // play tone 57 (A4 = 440 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 440, 300); // play tone 57 (A4 = 440 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 392, 600); // play tone 55 (G4 = 392 Hz)**  **delay(1000); // Wait for 1000 millisecond(s)**  **tone(10, 349, 300); // play tone 53 (F4 = 349 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 349, 300); // play tone 53 (F4 = 349 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 330, 300); // play tone 52 (E4 = 330 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 330, 300); // play tone 52 (E4 = 330 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 294, 300); // play tone 50 (D4 = 294 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 294, 300); // play tone 50 (D4 = 294 Hz)**  **delay(500); // Wait for 500 millisecond(s)**  **tone(10, 262, 900); // play tone 48 (C4 = 262 Hz)**  **delay(5000); // Wait for 5000 millisecond(s)**  **}** |

## **Bai 13 Cảm biến chuyển động**

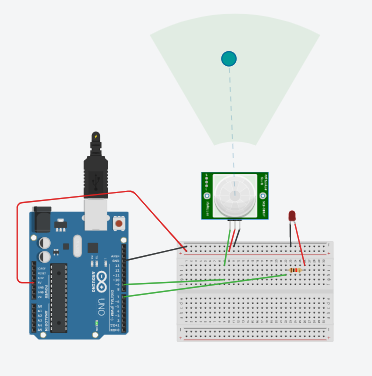
**1.Mo ta**

-Sử dụng sensor cảm biến chuyển động

**2.Linh kien**

1 Led

1 Pir sensor

**3.Sơ đồ thiết kế**

#### Hình 13: Cảm biến chuyển động

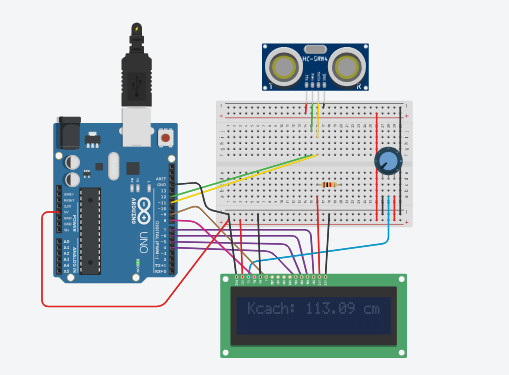
**4.Mã lệnh chính**

|  |
| --- |
| **// C++ code**  **int sensor =9;**  **int state;**  **int led = 7;**  **void setup()**  **{**  **pinMode(sensor, INPUT);**  **pinMode(led, OUTPUT);**  **}**  **void loop()**  **{**  **state = digitalRead(sensor);**  **if(state ==HIGH) digitalWrite(led, HIGH);**  **else digitalWrite(led, LOW);**  **}** |

## **Bài 14 : LCD-CAM BIEN AM THANH**

**1.Mô tả2.Linh kiện**

**3.Sơ đồ thiết kế**



#### Hình 14: LCD-CAM BIEN AM THANH

**4.Mã lệnh chính**

|  |
| --- |
| **#include <LiquidCrystal.h>**  **int RS = 8, E = 9;**  **int D4 = 4, D5 = 5, D6 = 6, D7 = 7;**  **int trig = 12, echo = 11;**  **float distance;**  **float t;**  **bool warn;**  **// Khoi tao LCD**  **LiquidCrystal lcd(RS,E,D4,D5,D6,D7);**  **float getDistance(){**  **digitalWrite(trig,HIGH);**  **delayMicroseconds(5);**  **digitalWrite(trig,LOW);**  **return pulseIn(echo, HIGH)/(58.3f);**  **}**  **void setup() {**  **pinMode(trig, OUTPUT);**  **pinMode(echo, INPUT);**  **lcd.begin(16,2);**  **lcd.setCursor(0,0); // cot = 0, hang = 0**  **lcd.print("Kcach:");**  **lcd.setCursor(14,0); // cot = 0, hang = 1**  **lcd.print("cm");**  **}**  **void loop(){**  **distance = getDistance();**  **lcd.setCursor(7,0);**  **lcd.print(distance);**  **if(distance < 30) {**  **lcd.setCursor(4,1);**  **lcd.print("Canh bao");**  **}**  **else {**  **lcd.setCursor(4,1);**  **lcd.print(" ");**  **}**  **}** |

## **Bai 15: Remote điều khiển đèn**

**1.Mô tả**

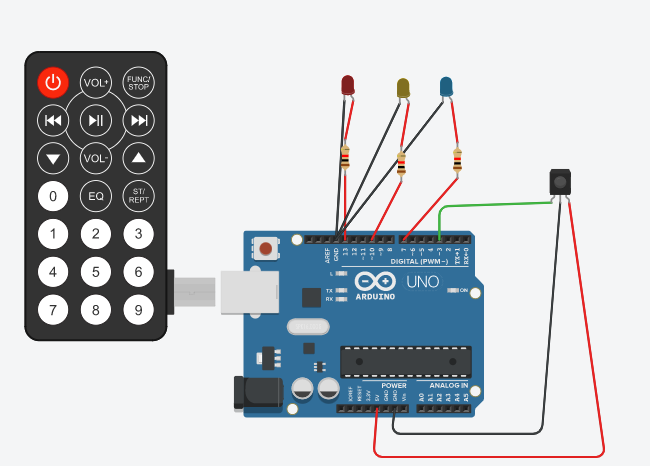
-Sử dụng remote để điều khiển 3 bóng đèn led

**2.Linh kiện**

-3 led

- 1 Remote

**3.Sơ đồ thiết kế**



#### Hình 15: Remote điều khiển 3 bóng đèn

**4.Mã lệnh chính**

|  |
| --- |
| **#include <IRremote.h>**  **int Re\_pin = 3;**  **IRrecv Bothu(Re\_pin);**  **decode\_results results;**  **void setup()**  **{**  **pinMode(13, OUTPUT);**  **pinMode(10, OUTPUT);**  **pinMode(7, OUTPUT);**  **Serial.begin(9600);**  **Bothu.enableIRIn();**  **}**  **void loop()**  **{**  **if(Bothu.decode(&results)) {**  **Serial.println(results.value); // HIỂN THỊ GIÁ TRỊ BIN CỦA CÁC PHÍM RA SERIAL MONITOR**  **Bothu.resume();**    **if(results.value == 16582903) { //phím 1**  **digitalWrite(13, HIGH);**  **digitalWrite(10, LOW);**  **digitalWrite(7, LOW);**  **}**  **if(results.value == 16615543) { //phím 2**  **digitalWrite(10, HIGH);**  **digitalWrite(13, LOW);**  **digitalWrite(7, LOW);**  **}**  **if(results.value == 16599223) { //phím 3**  **digitalWrite(7, HIGH);**  **digitalWrite(10, LOW);**  **digitalWrite(13, LOW);**  **}**  **if(results.value == 16580863) { //phím tắt**  **digitalWrite(13, LOW);**  **digitalWrite(10, LOW);**  **digitalWrite(7, LOW);**  **}**  **}**    **delay(100);**  **}** |

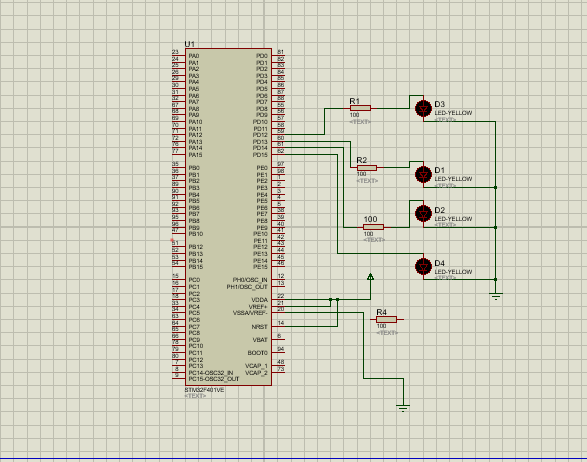
**Chương 3 : STM 32**

**Bài 1 : Nháy 4 LED**

**1.Mô tả**

-Nháy 4 đèn led sử dụng STM32 cổng 12-15

**2.Sơ đồ thiết kế**



#### Hình 16: Nháy 4 led với STM32

**3.Linh kiện**

STM32F401VE, 4 Led

**4.Mã lệnh chính**

|  |
| --- |
| #include "main.h"  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();    while (1)  {  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_Delay(1000);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_Delay(1000);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_Delay(1000);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);    HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);  HAL\_Delay(1000);  HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_RESET);  }    } |

## **Bài 2 : STM32 led trái tim**