[LỜI MỞ ĐẦU 2](#_Toc119932246)

[CHƯƠNG I. GIỚI THIỆU VỀ BÀI TOÁN 3](#_Toc119932247)

[CHƯƠNG II: thuật toán phát hiện tọa độ các điểm khớp 3](#_Toc119932248)

[CHƯƠNG III: xây dựng phần cứng, xây dựng phần mềm và thử nghiệm 6](#_Toc119932249)

[1. Cơ cấu phần cứng: 6](#_Toc119932250)

[2. Linh kiện điện tử: 6](#_Toc119932251)

[2.1. Arduino 6](#_Toc119932252)

[2.2. Module cảm biến vật cản hồng ngoại – LM393 10](#_Toc119932253)

[2.3. Driver điều khiển động cơ bước TB6600 4.0A 9~42VDC 12](#_Toc119932254)

[2.4. Module Relay 4 kênh 15](#_Toc119932255)

[2.5. Mạch giảm áp DC LM2596 3A 15](#_Toc119932256)

[3. Thiết bị điện 15](#_Toc119932257)

[4. Các phần mềm 15](#_Toc119932258)

[4.1. Arduino IDE 15](#_Toc119932259)

[4.2. Pycharm Community 16](#_Toc119932260)

[4.3. Altium 16](#_Toc119932261)

# LỜI MỞ ĐẦU

# CHƯƠNG I. GIỚI THIỆU VỀ BÀI TOÁN

Nhận diện hành động đứng lên ngồi xuống Squat. Khi dếm đủ 20 lần, máy tự động rót nước

# CHƯƠNG II: thuật toán phát hiện tọa độ các điểm khớp

AI (Trí tuệ nhân tạo) đang dần "xâm chiếm" hầu như tất cả các lĩnh vực trong cuộc sống. Với tính chất tự động hóa cao, có thể thực hiện các công việc khó và có độ chính xác ngày càng được cải thiện, AI đang dần trở thành một công cụ khó có thể thay thế được. Tuy nhiên, các mô hình AI hiện tại được đánh giá là "nặng, yêu cầu phần cứng cao" khiến cho việc áp dụng vào trong các dự án thực tế trở nên khá khó khăn, nhất là với các thiết bị mobile hoặc edge devices - xu thế hiện tại. Vì vậy, Google đã đưa ra một giải pháp, chính xác hơn là một bộ công cụ, cung cấp các công cụ cho các bài toán AI/ML đã được tối ưu để chạy trên nhiều nền tảng khác nhau, với tên gọi là MediaPipe. Trong bài này, mình sẽ giới thiệu về bộ công cụ trên và áp dụng nó vào một ứng dụng cụ thể: Hand Drawing bằng webcam.

Về tổng quan, MediaPipe là tập hợp của một loạt các giải pháp Machine Learning đa nền tảng, có thể can thiệp được và cực kỳ lightweight. Một số ưu điểm có thể kể tới của giải pháp này bao gồm:

* Cung cấp một giải pháp inference nhanh chóng: Google khẳng định rằng bộ công cụ này có thể chạy ổn định trên hầu hết các cấu hình phần cứng thông dụng.
* Dễ dàng cài đặt và triển khai: Việc cài đặt cực kỳ dễ dàng và tiện lợi, có thể triển khai trên nhiều nền tảng khác nhau như Mobile (Android/iOS), Desktop/Cloud, Web và IoT devices.
* Mã nguồn mở và miễn phí: Toàn bộ source code được công khai trên MediaPipe, người dùng hoàn toàn có thể sử dụng và tùy chỉnh trực tiếp để phù hợp với bài toán của mình.

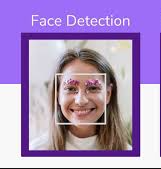


Các giải pháp trong MediaPipe:

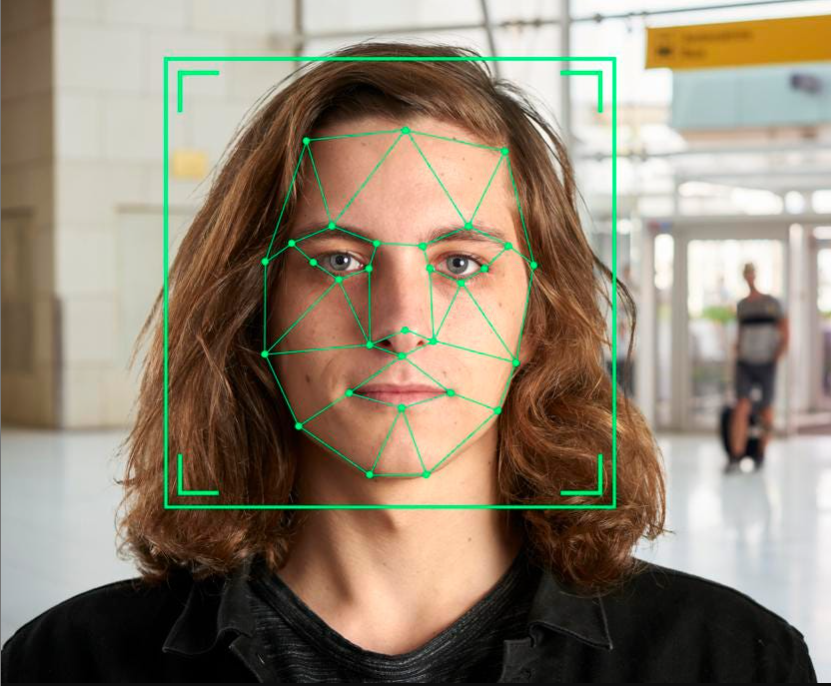
Hầu hết các bài toán nổi bật trong lĩnh vực Computer Vision - Thị giác máy tính, đều được Google cài đặt trong MediaPipe. Ta sẽ đi qua một lượt các giải pháp được cung cấp để hiểu rõ hơn về độ đa dạng của MediaPipe.

1. Phát hiện khuôn mặt/Face Detection

Đây là một bài toán quen thuộc với tất cả mọi người. Với đầu vào là một ảnh hoặc một video, nhiệm vụ của chúng ta là tìm ra vị trí và đóng hộp (bounding box) những khuôn mặt con người xuất hiện trên đấy, cũng như đánh dấu các điểm quan trọng (MediaPipe sử dụng 5-landmarks) trên khuôn mặt đó. MediaPipe Face Detection sử dụng mạng BlazeFace làm nền tảng nhưng thay đổi backbones. Ngoài ra, thuật toán NMS (non-maximum suppression) cũng được thay thế bởi một chiến thuật khác, giúp thời gian xử lý giảm đáng kể.



1. Tìm lưới khuôn mặt: dùng cho các ứng dụng biến đổi khuôn mặt như Tiktok



1. Iris: tìm khoảng cách từ đồng tử của mắt đến camera mà không cần Depth webcam
2. Detect cử chỉ bàn tay
3. Tìm hình dáng của cơ thể
4. Thay đổi màu tóc: có thể hair salon sẽ cần
5. Object detection & Box tracking: tìm vật thể, quá quen thuộc rồi
6. Theo dõi chuyển động của vật thể
7. Objectron: tìm hình lập phương chứa vật thể
8. KNIFT: Tìm vật thể bằng các đặc trưng đã biết

# CHƯƠNG III: xây dựng phần cứng, xây dựng phần mềm và thử nghiệm

Máy được chia làm 5 bộ phận:

* Cơ cấu phần cứng
* Module/Driver/Vi điều khiển
* Thiết bị điện\

## Cơ cấu phần cứng:

## Linh kiện điện tử:

### Arduino

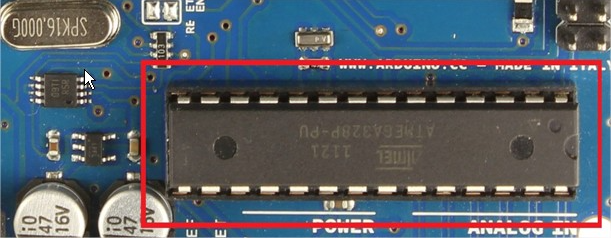


Hình 3.1. Arduino UNO

1. Một vài thông số của Arduino UNO R3

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp vào khuyên dùng | 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) |

1. Vi điều khiển



Hình 3..2. Vi điều khiển ATMEGA328P-PU

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,… hay những ứng dụng khác mà bạn đã được xem ở đây.

Thiết kế tiêu chuẩn của Arduino UNO sử dụng vi điều khiển ATmega328 với giá khoảng 90.000đ. Tuy nhiên nếu yêu cầu phần cứng của bạn không cao hoặc túi tiền không cho phép, bạn có thể sử dụng các loại vi điều khiển khác có chức năng tương đương nhưng rẻ hơn như ATmega8 (bộ nhớ flash 8KB) với giá khoảng 45.000đ hoặc ATmega168 (bộ nhớ flash 16KB) với giá khoảng 65.000đ.

1. 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 bạn không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, bạn sẽ làm hỏng Arduino UNO.

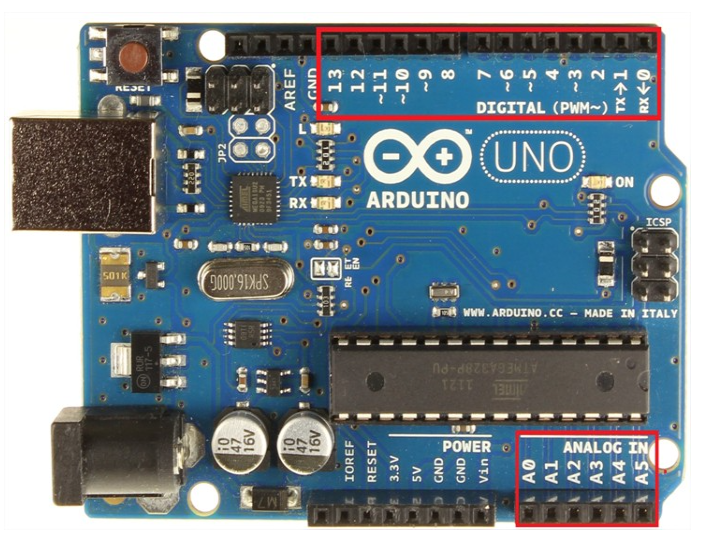
1. Các chân năng lượng

* **GND (Ground)**: cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn 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, bạn 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 bạn 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.

1. 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 đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.
* **2KB cho SRAM** (**S**tatic **R**andom **A**ccess **M**emory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn 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 cho 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.

1. Các cổng vào/ra



Hình 3.3. Các cổng vào

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).

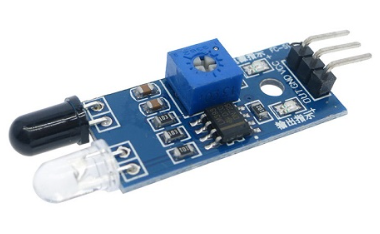
Một số chân digital có các chức năng đặc biệt như sau:

* **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.

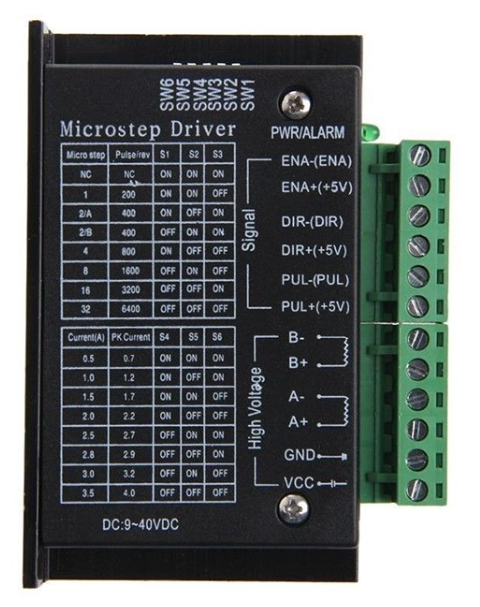
### Module cảm biến vật cản hồng ngoại – LM393



Hình 3.4. Module cảm biến vật cản hồng ngoại – LM393

* Cảm biến vật cản hồng ngoại có khả năng thích nghi với môi trường, có một cặp truyền và nhận tia hồng ngoại. Tia hồng ngoại phát ra một tần số nhất định, khi phát hiện hướng truyền có vật cản (mặt phản xạ), phản xạ vào đèn thu hồng ngoại, sau khi so sánh, đèn màu xanh sẽ sáng lên, đồng thời đầu cho tín hiệu số đầu ra (một tín hiệu bậc thấp).
* Khoảng cách làm việc hiệu quả 2 ~ 5cm, điện áp làm việc là 3.3 V đến 5V. Độ nhạy sáng của cảm biến vật cản hồng ngoại được điều chỉnh bằng chiết áp, cảm biến dễ lắp ráp, dễ sử dụng,….
* Có thể được sử dụng rộng rãi trong robot tránh chướng ngại vật, xe tránh chướng ngại vật và dò đường….
* THÔNG SỐ KỸ THUẬT
* Bộ so sánh sử dụng LM393, làm việc ổn định
* Điện áp làm việc: 3.3V – 5V DC.
* Khi bật nguồn, đèn báo nguồn màu đỏ sáng.
* Lỗ vít 3 mm, dễ dàng cố định, lắp đặt.
* Kích thước: 3.2cm \* 1.4cm
* Các mô-đun đã được so sánh điện áp ngưỡng thông qua chiết áp, nếu sử dụng ở chế độ thông thường, xin vui lòng không tự ý điều chỉnh chiết áp
* Cổng giao tiếp:
* VCC: điện áp chuyển đổi từ 3.3V đến 5V (có thể được kết nối trực tiếp đến vi điều khiển 5V và 3.3V)
* GND: GND ngoài
* OUT: đầu ra kỹ thuật số (0 và 1)

### Driver điều khiển động cơ bước TB6600 4.0A 9~42VDC



Hình 3.5. Driver điều khiển động cơ bước TB6600

**THÔNG SỐ KỸ THUẬT**

* Nguồn đầu vào là 9V – 42V.
* Dòng cấp tối đa là 4A.
* Ngõ vào có cách ly quang, tốc độ cao.
* Có tích hợp đo quá dòng quá áp.
* Cân nặng: 200G.
* Kích thước: 96 \* 71 \* 37mm.

**Cài đặt và ghép nối:**

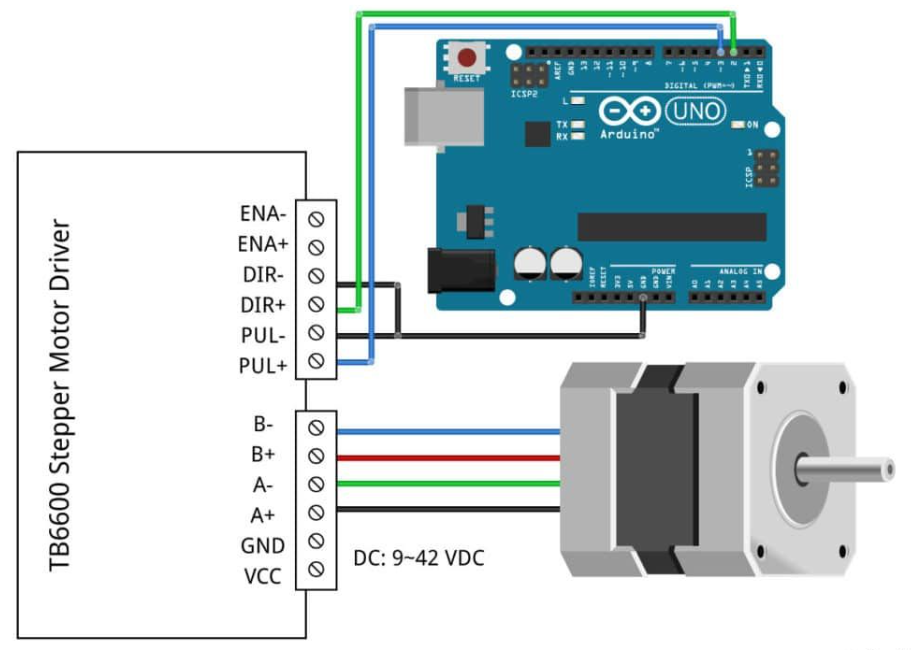
* DC+: Nối với nguồn điện từ 9 – 40VDC
* DC- : Điện áp (-) âm của nguồn
* A+ và A -: Nối vào cặp cuộn dây của động cơ bước
* B+ và B- : Nối với cặp cuộn dây còn lại của động cơ
* PUL+: Tín hiệu cấp xung điều khiển tốc độ (+5V) từ BOB cho M6600
* PUL-: Tín hiệu cấp xung điều khiển tốc độ (-) từ BOB cho M6600
* DIR+: Tín hiệu cấp xung đảo chiều (+5V) từ BOB cho M6600
* DIR-: Tín hiệu cấp xung đảo chiều (-) từ BOB cho M6600
* ENA+ và ENA -: khi cấp tín hiệu cho cặp này động cơ sẽ không có lực momen giữ và quay nữa
* Có thể đấu tín hiệu dương (+) chung hoặc tín hiệu âm (-) chung
* Cài đặt cường độ dòng điện

|  |  |  |  |
| --- | --- | --- | --- |
| **I(A)** | **SW4** | **SW5** | **SW6** |
| 4.0 | 1 | 1 | 1 |
| 3.5 | 0 | 1 | 1 |
| 3.0 | 1 | 0 | 1 |
| 2.5 | 0 | 0 | 1 |
| 2.0 | 1 | 1 | 0 |
| 1.5 | 0 | 1 | 0 |
| 1.0 | 1 | 0 | 0 |
| 0.5 | 0 | 0 | 0 |

* Cài đặt vi bước cho driver

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Micro** | **Pulse/rev** | **SW1** | **SW2** | **SW3** |
| OFF | 0 | 0 | 0 | 0 |
| 1 | 200 | 0 | 0 | 1 |
| 1/2A | 400 | 0 | 1 | 0 |
| 1/2B | 400 | 0 | 1 | 1 |
| 1/4 | 800 | 1 | 0 | 0 |
| 1/8 | 1600 | 1 | 0 | 1 |
| 1/16 | 3200 | 1 | 1 | 0 |
| OFF | 0 | 1 | 1 | 1 |

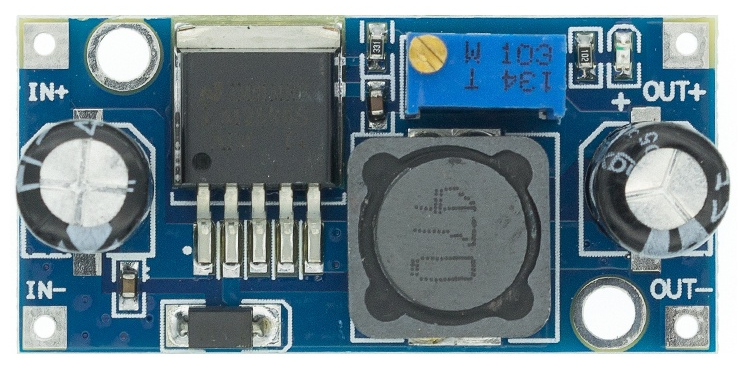
* Sơ đồ nối dây mạch điều khiển động cơ bước TB6600



Hình 3.6. Sơ đồ nối dây mạch điều khiển động cơ bước TB6600

### Module Relay 4 kênh

### Mạch giảm áp DC LM2596 3A



Hình Mạch giảm áp DC LM2596 3A

* [**Mạch giảm áp DC LM2596 3A**](https://nshopvn.com/product/mach-giam-ap-dc-lm2596-3a/) nhỏ gọn có khả năng giảm áp từ 30V xuống 1.5V mà vẫn đạt hiệu suất cao (92%) . Thích hợp cho các ứng dụng chia nguồn, hạ áp, cấp cho các thiết bị như camera, motor, robot,…

Thông số kỹ thuật

* Điện áp đầu vào: Từ 3V đến 30V.
* Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5V đến 30V.
* Dòng đáp ứng tối đa là 3A.
* Hiệu suất: 92%
* Công suất: 15W
* Kích thước: 45 (dài) \* 20 (rộng) \* 14 (cao) mm

## Thiết bị điện

## Các phần mềm

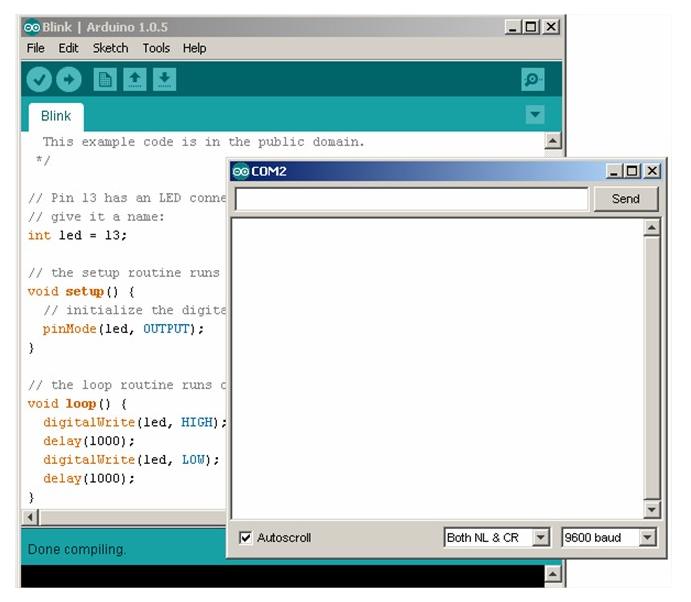
### Arduino IDE

Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “ngôn ngữ Arduino”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu. Nếu học tốt chương trình Tin học 11 thì việc lập trình Arduino sẽ rất dễ thở đối với bạn.

Để lập trình cũng như gửi lệnh và nhận tín hiệu từ mạch Arduino, nhóm phát triển dự án này đã cũng cấp đến cho người dùng một môi trường lập trình Arduino được gọi là Arduino IDE (**I**ntergrated **D**evelopment **E**nvironment)  như hình dưới đây.

### Pycharm Community

## Altium



1. Phương án thực hiện

Sử dụng máy tính nhúng Raspberry Pi để lập trình giao diện người dùng trên màn hình cảm ứng, đọc/gửi dữ liệu cho Arduino để điều khiển thiết bị.

Sử dụng vi điều khiển Arduino Uno để điều khiển thiết bị.

Chương 1: Giới thiệu về bài toán;