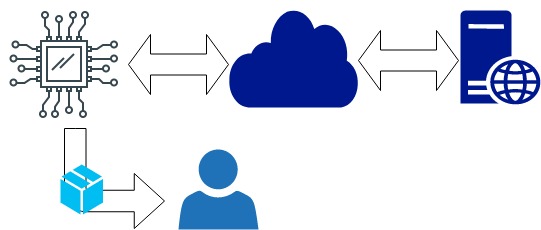
# Yêu cầu

Hệ thống yêu cầu một 1 thiết bị có khả năng trả hàng và giao tiếp với server thông qua internet.



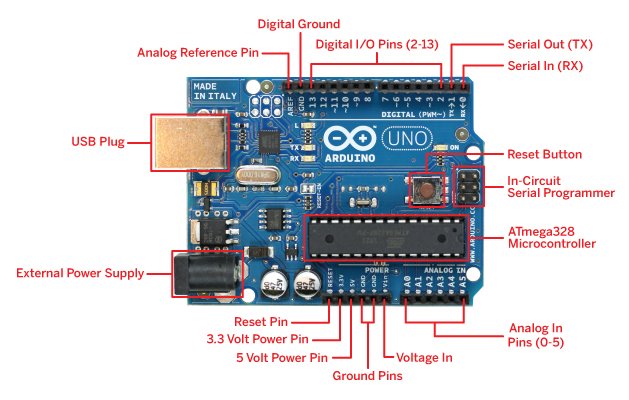
# Lựa chọn phần cứng

Phân tích yêu cầu trên ta thấy thiết bị trong tưởng tượng cần có:

* 1 bộ phận giao tiếp với server thông qua internet
* 1 bộ phận đảm nhiệm việc trả hàng hóa
* 1 bộ phận làm nhiệm vụ xữ lý ghép nối 2 bộ phần trên.

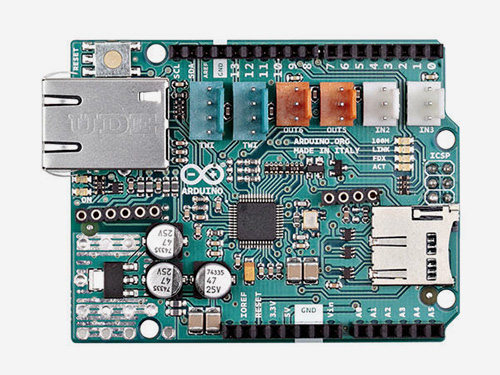
## Board mạch Arduino -Uno (bộ phận xữ lý)

Arduino là 1 board mạch xữ lý, nhận tác động từ mội trường và tác động ngược lại thông qua các pin của mình.



## Arduino ethernet shield bộ phận (kết nối internet)

Là một modun của arduino cho phép nó kết nối với internet một cách dễ dàng.

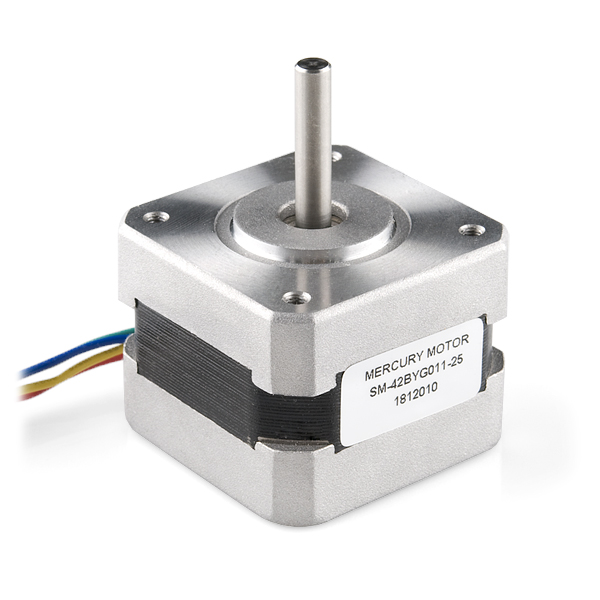


## Bộ phận trả hàng (chỉ làm demo, gồm nhiều thiết bị)

Đây là bộ phân phức tạp vì vữa phải thực hiện việc trả hàng vừa phải xác định được hàng đã được trả hay chưa. Để làm được việc này ta cần 3 bộ phận nhỏ gồm: 1 động cơ, 1 mạch điều khiển động cơ, 1 cảm biến và 1 phần nhỏ làm nhiệm vụ giao tiếp với arduino uno.

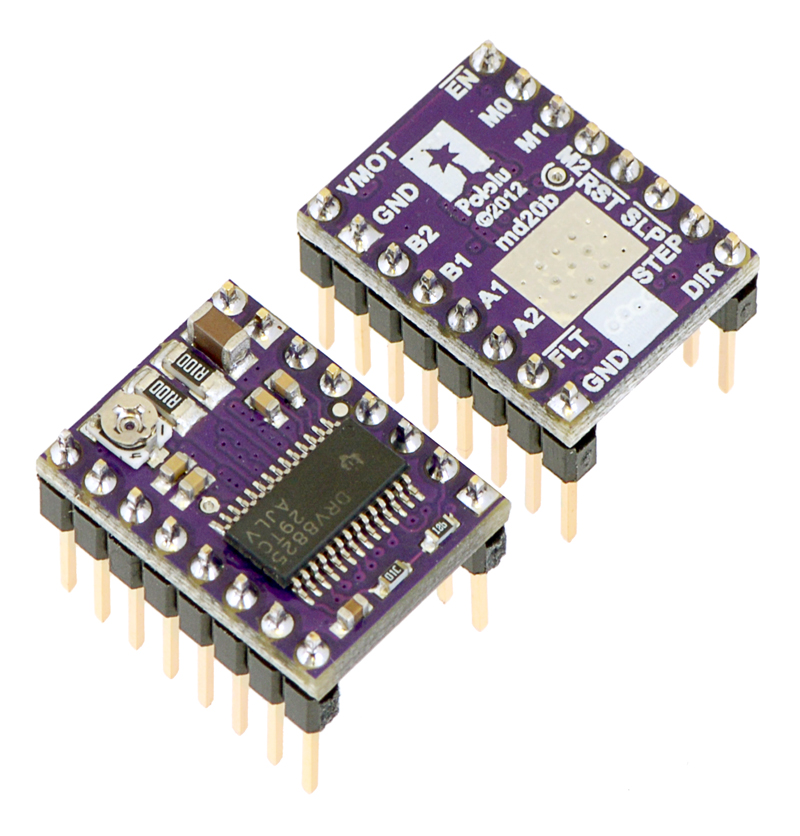
### Stepper motor (động cơ bước)

Động cơ bước là loại động cơ không xoay liên tục mà thực hiển những bước xoay rời rạc. Điều này có lợi khi cần độ chính xác.



### Breakout board DRV8825 (bộ phận điều khiển động cơ bước)

Đây là mạch điều khiển động cơ bước có thêm các chức năng bảo vệ qua dòng, quá nhiệt.



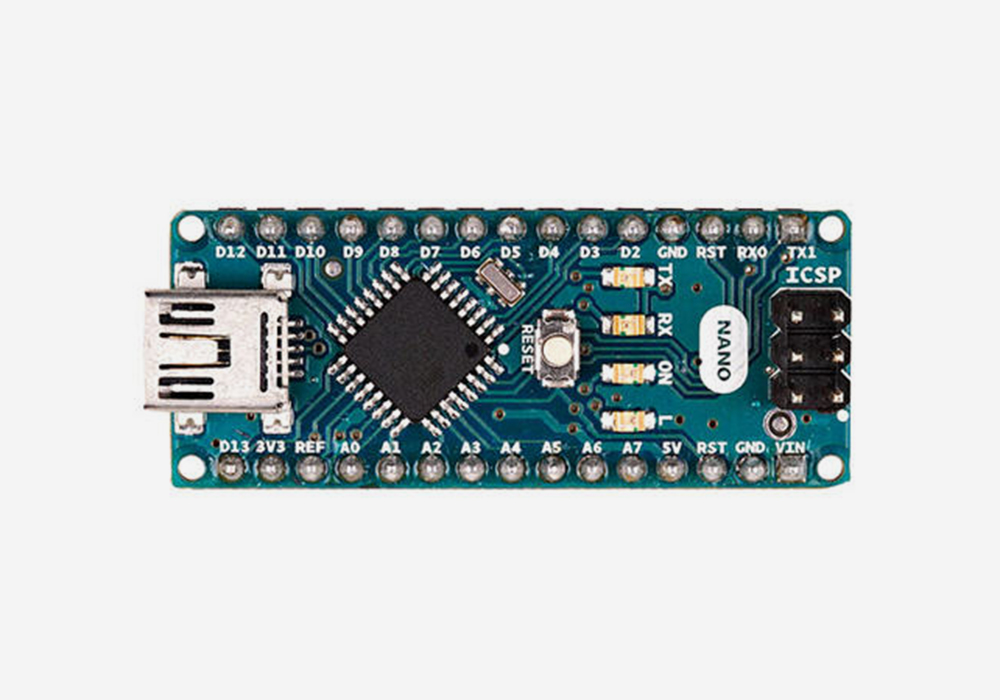
## Quang trở

Sử dụng loại quang trở kĩ thuật số độ chính xác cao và dễ sử dụng.

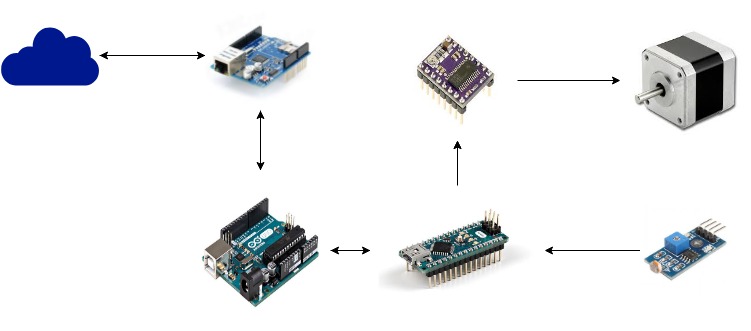


### Arduino Nano (bộ phận trung gian giữa Arduino Uno và mạch điều khiển động cơ bước)

Arduino Nano cũng giống như arduino uno nhưng được thiết kế nhỏ hơn, rẻ tiền dùng để đảm nhận các công việc đơn giản.



## Sơ đồ kết nối phần cứng



# Ghép nối phần cứng.

## Arduino uno với Ethernet shield

Như đã nói ở phần trên Ethernet Shield là một modun của Arduino nên chỉ việc cắm trực tiếp vào Arduino Uno, các chân của Ethernet Shield sẽ khớp với các chân của Arduino Nano.



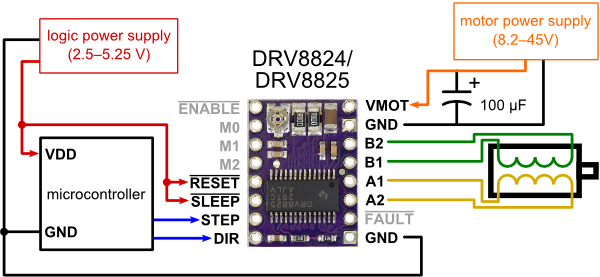
Việc cuối cùng là kết nối với cáp mạng Rj45.



## Động cơ bước, mạch điều khiển động cơ, quang trở, Arduino Nano.

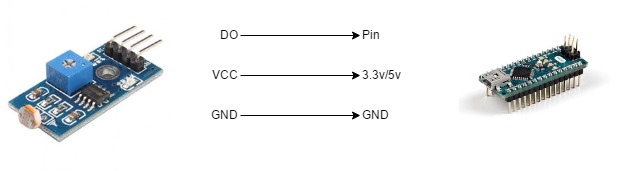
### Động cơ bước, mạch điều khiển động cơ, Arduino Nano.

Ta mắc dây để kết nối động cơ bước, mạch điều khiển động cơ với arduino nano như sơ đồ bên dưới.



### Quang trở, Arduino Nano

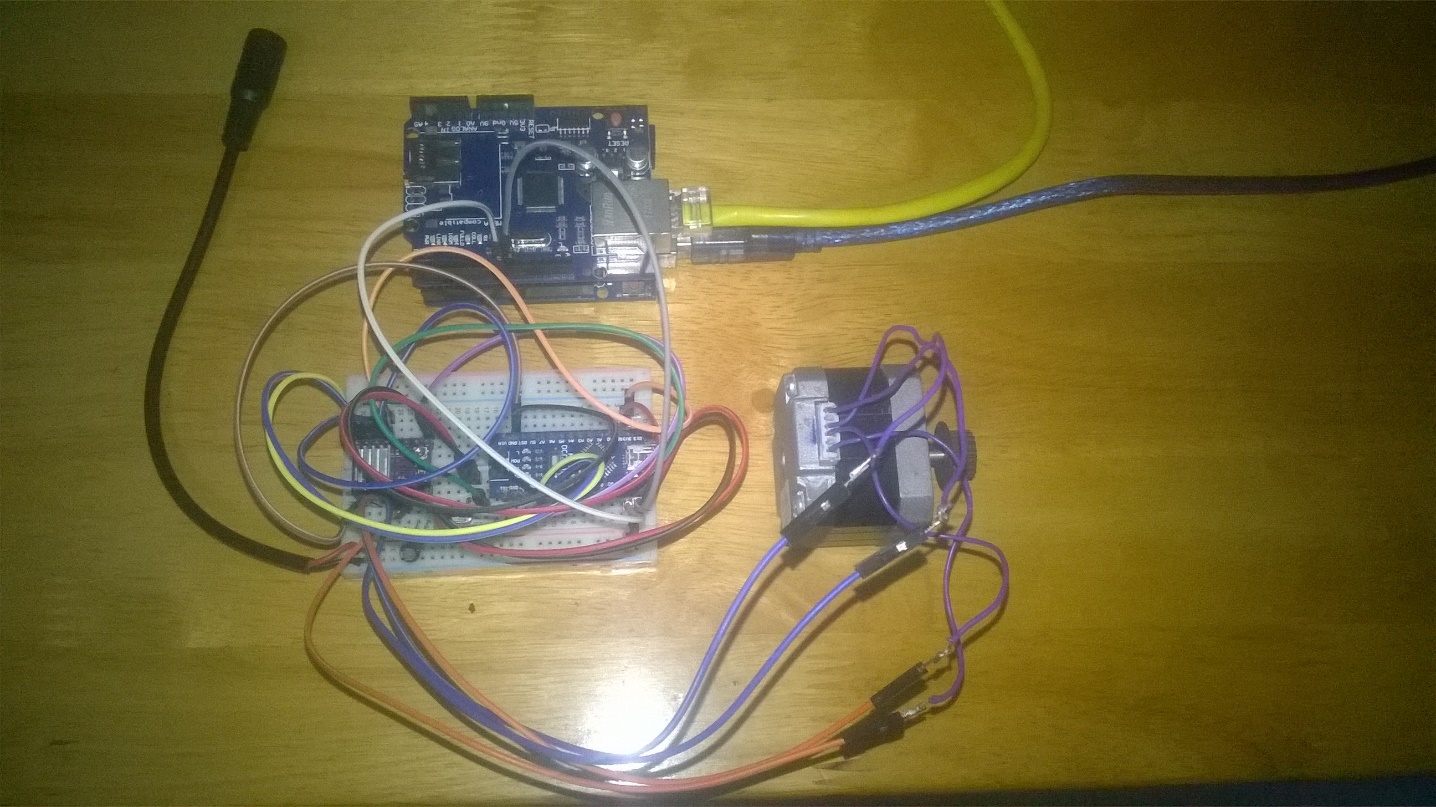
Nối các chân của quang trở với các chân của Arduino Nano theo hình bên dưới.



## Arduino Uno, Arduino Nano

Để kết nối hai Arduino với nhau ta chỉ cần nối hai pin digital bất kì trên hai Arduino với nhau.

## Hình ảnh thực tế



# Lập trình

## Đọc giá trị quang trở

#define PHOTO\_RESISTOR 7 // chân DO của quang trở

void setupPhotoresistor() {

pinMode(PHOTO\_RESISTOR, INPUT);

}

digitalRead(PHOTO\_RESISTOR);

Vì ta sử dụng quang trở kỹ thuật số nên giá trị trả về là 0 hoặc 1.

## Điều khiển động cơ bước

Để làm động cơ bước xoay 1 bước ta chỉ cần kéo chân **STEP** của mạch DRV8825 lên rồi xuống. Mỗi bước là .

// DRV8825

#define ENABLE 4

#define STEP 3

#define DIR 2

void enableMotor() {

digitalWrite(ENABLE, LOW);

}

void disableMotor() {

digitalWrite(ENABLE, HIGH);

}

void setupMotor() {

pinMode(STEP, OUTPUT);

pinMode(DIR, OUTPUT);

pinMode(ENABLE, OUTPUT);

disableMotor();

}

void doQuarterRevolution() {

int x;

for (x = 0; x < 50; x++)

{

digitalWrite(STEP, HIGH);

delayMicroseconds(1000);

digitalWrite(STEP, LOW);

delayMicroseconds(1000);

}

}

## Kết nối Internet

Arduino đã hỗ trợ sẵn thư viện cho Ethernet Shield. Để sử dụng ta cần chuẩn bị trước 1 địa chỉ MAC, IP ta có thể lấy động bằng DHCP.

void setupEthernetShield() {

// start the Ethernet connection:

if (Ethernet.begin(mac) == 0) {

Serial.println("Failed to configure Ethernet using DHCP");

// no point in carrying on, so do nothing forevermore:

for (;;)

;

}

// print your local IP address:

Serial.println(Ethernet.localIP());

}

## Sử dụng MQTT

Để sự dụng gia thức MQTT trên Arduino ta sử dụng thư viện PubSubClient <https://github.com/knolleary/pubsubclient>.

EthernetClient ethClient;

PubSubClient client(ethClient);

client.setServer("test.mosquitto.org", 1883);

client.setCallback(callback);

void reconnect() {

// Loop until we're reconnected

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

// Attempt to connect

if (client.connect("arduinoClient")) {

Serial.println("connected");

// Once connected, publish an announcement...

client.publish("a", "hello server");

// ... and resubscribe

client.subscribe("s");

}

else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

// Wait 5 seconds before retrying

delay(5000);

}

}

}

void callback(char\* topic, byte\* payload, unsigned int length) {

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

String inString((char\*)payload);

Serial.println(inString);

commandTransporter.println(inString.toInt());

}

## Giao tiếp giữa 2 Arduino

Như đã nói ở trên để giao tiếp giữa hai Arduino với nhau ta dùng thư viện có sẵn SoftwareSerial. Kết nối 2 pin của Arduino này với 2 pin của Arduino kia. 1 pin làm nhiệm vụ nhận, 1 pin làm nhiệm vụ gửi dữ liệu đi.

Ta cần cài đặt sao cho pin nhận của Arduino này là pin gửi của Arduino kia và ngược lài.

SoftwareSerial commandTransporter(RX, TX);

commandTransporter.begin(BAUD);

// gửi dữ liệu

commandTransporter.println(“data”);

// đọc dữ liệu

commandTransporter.read();