

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập - Tự do - Hạnh phúc**

**ĐƠN YÊU CẦU CÔNG NHẬN SÁNG KIẾN**  
**Cấp: Ngành Năm học 2021-2022**

Kính gửi: - Hội đồng khoa học Trường THPT Nguyễn Du;  
- Hội đồng khoa học Sở GD&ĐT Phú Yên.

**1. Tên sáng kiến:** “Lập trình IoT – Bước khởi đầu cho sáng tạo khoa học kỹ thuật trong học sinh THPT”.

**2. Tác giả sáng kiến**

- Họ và tên: **LÊ QUYỀN** Giới tính: Nam
- Ngày tháng năm sinh: 20/08/1982;
- Mail: lquyen.thpt.nd@phuyen.edu.vn
- Điện thoại: 0974.881.001
- Trình độ chuyên môn: Thạc sĩ Khoa học máy tính
- Chức vụ : Giáo viên
- Cơ quan, đơn vị: Trường THPT Nguyễn Du.
- Địa chỉ: 04 Trần Hưng Đạo, TT Hai Riêng, H. Sông Hinh, T. Phú Yên.

**3. Đồng tác giả sáng kiến: (không)**

**4. Chủ đầu tư tạo ra sáng kiến: (không)**

**5. Các tài liệu đính kèm:**

- 5.1. Biên bản họp Hội đồng sáng kiến cấp cơ sở
- 5.2. Quyết định về việc công nhận sáng kiến cấp cơ sở

*Sông Hinh, ngày 01 tháng 03 năm 2022*

**Xác nhận của Hội đồng khoa học của  
đơn vị**

**TÁC GIẢ SÁNG KIẾN**

**LÊ QUYỀN**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**MÔ TẢ GIẢI PHÁP VÀ KẾT QUẢ THỰC HIỆN SÁNG KIẾN**

**1. Tên sáng kiến**

“Lập trình IoT – Bước khởi đầu cho sáng tạo khoa học kỹ thuật trong học sinh THPT”.

**2. Lĩnh vực áp dụng sáng kiến**

Chuyên đề trong Bồi dưỡng Học sinh THPT tham gia các kỳ thi KHKT.

Tài liệu tham khảo cho Học sinh bước đầu tiếp cận với lập trình IoT, hệ thống nhúng, ... nhằm tạo đam mê hứng thú trong triển khai các ý tưởng sáng tạo KHKT.

Tài liệu tham khảo cho Giáo viên tham gia các Dự án KHKT cấp tỉnh.

Tài liệu tham khảo cho Giáo viên chuẩn bị dạy Chuyên đề học tập Tin học 10 – Định hướng khoa học máy tính – Chương trình GDPT 2018.

**3. Mô tả các giải pháp cũ thường làm**

Hiện nay, trong các Dự án Khoa học kỹ thuật dành cho học sinh trung học, có một lĩnh vực được các em tham gia rất nhiều đó là ***Hệ thống nhúng***. Tại bước chuyển từ ý tưởng sang thực nghiệm, các em gặp rất nhiều khó khăn, vì không biết phải tham khảo nguồn tài liệu nào, các chương trình mẫu đa số chỉ trình bày lệnh chứ không có nhiều chú thích rõ ràng.

Ngôn ngữ lập trình hệ thống nhúng, IoT ... đa số sử dụng C++ trong khi học sinh các trường trong tỉnh chủ yếu được dạy bằng ngôn ngữ lập trình Pascal, nên việc tiếp cận đọc hiểu chương trình để tự học là khá khó khăn và mất thời gian của các em.

Các tài liệu về lập trình cho các thiết bị IoT đa số được xuất bản dưới dạng website, sách hoặc file pdf được mã hóa. Độ tin cậy của các tài liệu này chưa cao, gây rất nhiều khó khăn và chiếm rất nhiều thời gian của thầy và trò khi thực hiện viết chương trình những dự án lớn, nhiều dòng lệnh.

**4. Ngày sáng kiến được áp dụng lần đầu: Tháng 09/2021.**

**5. Nội dung:**

**5.1. Mô tả giải pháp mới hoặc cải tiến:**

Giải pháp mới trong đề tài này là tôi sưu tầm, tập hợp, tổng hợp và kiểm chứng các tri thức liên quan đến lập trình IoT thông qua Arduino IDE và hệ sinh thái các thiết bị của Arduino. Biên soạn chúng thành một Sổ tay Arduino, đồng thời tất cả các chương trình giao tiếp các thiết bị đều được kiểm chứng trực tiếp trên thiết bị tương ứng, giải thích rõ ràng và upload lên **github** ở chế độ public để tất cả Học sinh, Giáo viên và cộng đồng Arduino có thể bổ sung, chỉnh sửa để đạt được độ tối ưu cần thiết. Địa chỉ **github** của toàn bộ mã nguồn này tại:

<https://github.com/LQ-Computer/IoT-for-STEM>

Từ đó dễ dàng tạo ra một thư viện để các em học sinh và Giáo viên có thể tham khảo khi cần thiết nhằm hỗ trợ trong việc triển khai thực hiện các Dự án KHKT và giảng dạy bộ môn Tin học 10 ở Chuyên đề học tập Tin học 10 – Định hướng khoa học máy tính – Chương trình GDPT 2018.

Nội dung của giải pháp này được trình bày chi tiết trong phần Phụ lục III của sáng kiến này.

**• Kết quả của sáng kiến:**

Sáng kiến đã được áp dụng cho Học sinh thực hiện Dự án trong Cuộc thi Robocon do Trường ĐHSP Kỹ thuật Tp.HCM phối hợp cùng Tỉnh đoàn tổ chức.

Hoàn thành việc upload mã nguồn chương trình đã qua kiểm chứng lên github.

**• Sản phẩm được tạo ra từ giải pháp**

Chuyên đề “Lập trình IoT – Bước khởi đầu cho sáng tạo khoa học kỹ thuật trong học sinh THPT” (Phụ lục III).

Thư viện github: <https://github.com/LQ-Computer/IoT-for-STEM>

**5.2. Khả năng áp dụng của sáng kiến:**

Sáng kiến có thể được sử dụng làm Chuyên đề trong Bồi dưỡng Học sinh THPT tham gia các kỳ thi KHKT.

Sáng kiến có thể được sử dụng làm Tài liệu tham khảo cho Học sinh bước đầu tiếp cận với lập trình IoT, hệ thống nhúng, ... nhằm tạo đam mê hứng thú trong triển khai các ý tưởng sáng tạo KHKT.

Sáng kiến có thể được sử dụng làm Tài liệu tham khảo cho Giáo viên tham gia các Dự án KHKT cấp tỉnh.

Sáng kiến có thể được sử dụng làm Tài liệu tham khảo cho Giáo viên chuẩn bị dạy Chuyên đề học tập Tin học 10 – Định hướng khoa học máy tính – Chương trình GDPT 2018.

### **5.3. Đánh giá lợi ích kinh tế - xã hội của sáng kiến:**

Giáo viên Tin học và Học sinh có thể xem sản phẩm của sáng kiến như một tài liệu tham khảo để tự học, tự rèn luyện.

- ***Cam kết:*** Tôi xin cam đoan mọi thông tin nêu trên là trung thực, không sao chép và không vi phạm quyền sở hữu trí tuệ.

**XÁC NHẬN CỦA ĐƠN VỊ**

**TÁC GIẢ SÁNG KIẾN**

**Lê Quyền**

## LẬP TRÌNH IOT – BƯỚC KHỞI ĐẦU CHO SÁNG TẠO KHOA HỌC KỸ THUẬT TRONG HỌC SINH THPT

### MỞ ĐẦU

Trong khoảng vài năm trở lại đây, chắc hẳn mỗi bạn đọc đã không ít lần nghe thoáng qua về khái niệm “*Internet of Things*”. Và trong đó, những thiết bị gia dụng như lò nướng hay tủ lạnh có thể “nói chuyện” được với nhau. Nhưng chúng kết nối với nhau như thế nào? Thực thi ra sao? Làm sao để chúng nhận được sự điều khiển của con người?... Có rất nhiều câu hỏi cho những ai chưa biết đến IoT. Ở đề tài này, tôi sẽ mang đến những kiến thức căn bản về hệ sinh thái Internet of Things, hệ sinh thái Arduino, lập trình ứng dụng dựa trên các thiết bị Arduino.

#### I. Khái niệm “Internet of Things”

Thuật ngữ “*Internet of Things*” xuất hiện khá nhiều và thu hút không ít sự quan tâm chú ý của thế giới công nghệ. Tuy nhiên, không như nhiều xu hướng công nghệ trước đây, hiện nay vẫn chưa có một định nghĩa được chấp nhận rộng rãi nào cho khái niệm Internet of Things.

Theo Wikipedia thì khái niệm : *Internet of Things (IoT)* được dịch là **Mạng lưới vạn vật kết nối Internet** là một liên mạng, trong đó các thiết bị, phương tiện vận tải (được gọi là “thiết bị kết nối” và “thiết bị thông minh”), phòng ốc và các trang thiết bị khác được nhúng với các bộ phận điện tử, phần mềm, cảm biến, cơ cấu chấp hành cùng với khả năng kết nối mạng máy tính giúp cho các thiết bị này có thể thu thập và truyền tải dữ liệu. Năm 2013, tổ chức *Global Standards Initiative on Internet of Things (IoT-GSI)* định nghĩa IoT là “*hạ tầng cơ sở toàn cầu phục vụ cho xã hội thông tin, hỗ trợ các dịch vụ điện toán chuyên sâu thông qua các thiết bị được kết nối với nhau nhờ vào công nghệ thông tin và truyền thông*”.

Về cơ bản, Internet of Things cung cấp kết nối chuyên sâu cho các thiết bị, hệ thống và dịch vụ, đồng thời hỗ trợ đa dạng giao thức, và ứng dụng. Kết nối các thiết bị nhúng này được kỳ vọng sẽ mở ra kỷ nguyên tự động hóa trong hầu hết các ngành, mở rộng tới những lĩnh vực thành phố thông minh.

Trên cơ sở đó, hệ thống IoT cho phép các thiết bị được cảm nhận hoặc được điều khiển từ xa thông qua hạ tầng mạng hiện hữu, tạo cơ hội cho thế giới thực được tích hợp trực tiếp hơn vào hệ thống điện toán. Hệ quả mang lại là hiệu năng, độ tin cậy và lợi ích kinh tế được tăng cường bên cạnh việc giảm thiểu sự can dự của con người. IoT là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một

mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính.

## **II. Internet of Things đang thay đổi cách sống trên thế giới như thế nào?**

IoT ảnh hưởng trực tiếp tới tất cả chúng ta trong mọi mặt.

Một bối cảnh giả định điển hình sẽ diễn ra như sau: Khi chúng ta bước gần về đến cửa nhà, cơ chế điều khiển tự động tích hợp trong chìa khóa (hoặc điện thoại, smartwatch...) của chúng ta sẽ tự động mở cửa từ xa. Khóa cửa sẽ gửi tín hiệu không dây đến hệ thống mạng nội bộ trong nhà, trước hết là khiến đèn cửa và hàng lang được kích hoạt. Hệ thống điều hòa, vốn đã chuyển sang trạng thái chờ khi chúng ta rời đi, sẽ tiếp tục hoạt động trở lại. Theo một số cài đặt sẵn, thậm chí máy pha cà phê sẽ có thể tự động được kích hoạt để chuẩn bị sẵn 1 tách cà phê thơm phức ngay khi ta bước chân vào phòng khách. Mọi thiết bị trong một smart house sẽ giao tiếp và hoạt động một cách hài hòa, từ đó đưa chúng ta đến một định nghĩa đơn giản nhất cho IoT: “Một hệ sinh thái IoT thực sự là một thế giới trong đó mọi thiết bị đều có thể phối hợp được với nhau”.

Còn ở những địa điểm công cộng thì như thế nào? Lấy việc đỗ xe làm ví dụ. Nhiều thành phố trên thế giới hiện đã bắt đầu sử dụng hệ thống cảm biến của hãng Streetline trên các ô đỗ xe dọc các tuyến đường của mình. Các bác tài sẽ có thể sử dụng phần mềm trên điện thoại di động để tìm thông tin về các vị trí còn trống. Hãng này mới đây còn bổ sung thêm tính năng cảm ứng âm thanh và nhiệt độ bề mặt cho các sản phẩm của mình để giúp chính quyền thành phố các nước châu Âu lựa chọn thời điểm sử dụng muối trên bề mặt đường (giúp cải thiện tình trạng băng giá trơn trượt).

Bên cạnh đó Xe hơi tự lái và tự đỗ - Tesla đã làm được rất nhiều việc để giúp đỡ các bác tài. Và cũng có rất nhiều ứng dụng ngoài kia đang thu thập thông tin về những bãi đỗ xe lân cận và sử dụng chúng để nói cho người lái xe biết nơi có chỗ đỗ xe trống. Nó không phải là không tưởng để kết nối dữ liệu này trực tiếp đến chiếc xe có khả năng tự đỗ và tiềm năng là chiếc xe sẽ tìm ra một chỗ đỗ và tự đỗ xe bằng khả năng của nó mà không có sự can thiệp nào từ người lái xe. Và hoàn toàn bạn có thể lái một chiếc xe với bộ cảm ứng tích hợp cảnh báo tài xế khi bánh xe xẹp. Và khi lái xe trên đường, người ta đang bắt đầu tận dụng các cảm biến có sẵn trên đĩa số trên điện thoại thông minh để theo dõi độ gập ghềnh của đường. Tất cả những gì người ta cần là một phần mềm trên thiết bị di động để tính toán kết quả thu được từ các cảm biến này và kết luận về tình trạng đường.

Trong nhà các bạn sẽ là một tủ lạnh thông minh. Chúng ta sẽ mua các mặt hàng rau củ online và công ty cung cấp sẽ đưa đến tận nhà. Tuy nhiên thay vì ngồi từ PC lục lọi từng trang web, ta sẽ làm việc với giao diện của tủ lạnh thông minh do nhà sản xuất thiết kế sẵn – kết nối với các kênh phân phối đã được kiểm định và thậm chí các mặt hàng rau củ sẽ có ID rõ ràng. Chưa hết, chiếc tủ sẽ biết được bên trong nó còn những gì thông qua các khay có cảm biến trọng lượng, cũng như theo dõi được thời hạn sử dụng của từng

sản phẩm thông qua ID. Dĩ nhiên, trong tương lai, các nhà sản xuất thậm chí sẽ phải cung cấp cho người dùng các tùy chọn như nhập thêm các cửa hàng quen vào danh sách (miễn sao cửa hàng đó có kênh kết nối trực tuyến), tự động hóa việc đặt hàng mỗi cuối/đầu tuần hay thậm chí là cung cấp thông tin dinh dưỡng của từng chủng loại thực phẩm.

Còn về vấn đề theo dõi và chăm sóc sức khỏe thì sao nhỉ? Khi kết hợp với một số thiết bị theo dõi tình trạng y tế, chẳng hạn các vòng tay theo dõi sức khỏe khá phổ biến hiện nay hay cân điện tử, tủ lạnh sẽ hiện thông báo cảnh báo về tình trạng cân nặng, hay đường huyết mỗi khi một “bệnh nhân” mở tủ lấy ra que kem thứ...3 trong ngày. Khi mới xuất xưởng, các vòng tay hay cân điện tử này có thể vốn chưa được thiết kế để gửi thông tin cho tủ lạnh. Nhưng một phần mềm trên di động, hay trên home server trong nhà, hay trên một dịch vụ đám mây thậm chí là phần mềm được viết trên chính OS của tủ lạnh đó, có thể được thiết kế để tổng hợp thông tin từ tất cả các nguồn này và tiến hành yêu cầu tủ lạnh gửi cảnh báo cho người dùng. Và khi bạn đang được bác sĩ điều trị cho dùng thuốc thì có những Chai đựng thuốc thông minh. Những chai thuốc này theo dõi việc sử dụng thuốc của bạn, không chỉ để chắc chắn rằng bạn đã dùng đúng liều, mà còn cho bác sĩ của bạn biết khi bạn cần nhiều hơn. Và trong đó, những viên thuốc thông minh, cung cấp cho bác sĩ những thông tin tốt hơn về sức khỏe của bạn và hiệu quả của quá trình điều trị đang diễn ra.

Nếu chúng ta suy nghĩ theo các định hướng xưa cũ, trong thế giới mà đầu DVD phải nối với TV bằng cáp, và phải là cáp đúng chủng loại, cắm đúng vị trí.v.v. Việc kết nối giữa thiết bị y tế, cân điện tử và... tủ lạnh ở trên nghe có vẻ khó tin. Nhưng trong thế giới IoT, các lập trình viên sẽ có khả năng kết nối hầu như mọi loại thiết bị được sản xuất hướng tới hệ sinh thái này với nhau, từ đó tạo ra những chức năng hoàn toàn mới mà trước đó có thể chúng ta chưa từng nghĩ tới.

Rõ ràng, Internet of Things có thể thay đổi hoàn toàn cách sống của con người trong tương lai. Khi mọi thứ đã được “Internet hóa”, người dùng hoàn toàn có thể điều khiển chúng từ bất cứ đâu, chỉ cần một chiếc điện thoại có kết nối Internet. Sở hữu những thành tựu trong lĩnh vực này nghĩa là bạn đang nắm giữ trong tay chìa khóa thành công của mọi thời đại. Internet of Things chính là xu hướng của tương lai.

# CHƯƠNG I: ARDUINO LÀ GÌ?

## I. Khái niệm, lịch sử hình thành và phát triển.

Theo định nghĩa từ [www.arduino.cc](http://www.arduino.cc) , **Arduino là nền tảng điện tử mã nguồn mở, dựa trên phần cứng và phần mềm, linh hoạt và dễ sử dụng**, các board Arduino có khả năng đọc dữ liệu từ môi trường (ánh sáng, nhiệt độ, độ ẩm,...), trạng thái nút nhấn, tin nhắn từ Twitter,... và điều khiển trở lại với các thiết bị như động cơ, đèn LED, gửi thông tin đến 1 nơi khác,... Chúng ta có thể điều khiển các vi điều khiển trên board Arduino bằng cách sử dụng **ngôn ngữ lập trình C++**, được điều khiển biên dịch bởi **Arduino IDE** và các trình biên dịch đi kèm ra mã máy nhị phân. Lúc này Vi điều khiển có thể dễ dàng thực thi chương trình.

Hiện tại, Arduino là một công ty hoạt động trong lĩnh vực phần cứng và phần mềm máy tính mã nguồn mở. Dự án Arduino được sinh ra tại học viện Interaction Design ở Ivrea, Italy vào năm 2003. Mục đích ban đầu của board Arduino là giúp cho các sinh viên ở học viện - những người không có nền tảng kiến thức về điện tử có thể tạo ra các sản phẩm 1 cách nhanh chóng với chi phí thấp và dễ sử dụng. Đó là 1 dự án mã nguồn mở, Arduino phát triển thông qua việc cho phép người dùng trên toàn thế giới có thể xây dựng, phát triển và đóng góp vào dự án.

### Tại sao là Arduino ?

Hiện nay, Arduino được sử dụng trong rất nhiều dự án và trong nhiều lĩnh vực khác nhau. Chính vì sự đơn giản, dễ sử dụng và đặc biệt là mã nguồn và phần cứng mở nên nó nhận được sự hỗ trợ rất lớn từ các lập trình viên trên toàn thế giới. Phần mềm rất dễ cho những người mới bắt đầu nhưng cũng không thiếu sự linh hoạt cho những lập trình viên lâu năm. Cộng đồng Arduino rất lớn nên khi sử dụng với Arduino bạn dễ dàng tìm được

Những người xây dựng nền tảng Arduino đã tạo ra những lệnh vô cùng đơn giản, giúp cho người dùng dễ tiếp cận.

Các ví dụ mẫu đi kèm với thư viện và tất cả đều open-source nên khi tìm đến các ví dụ mẫu, chúng ta sẽ hiểu cách thức thư viện hoạt động. đồng thời có thể xem source code của họ viết để có thể hiệu chỉnh, tối ưu thư viện theo cách của mình.

Việc upload code thông qua cổng USB, giúp đơn giản quá trình nạp code.

## II. Những board mạch Arduino thông dụng trên thị trường

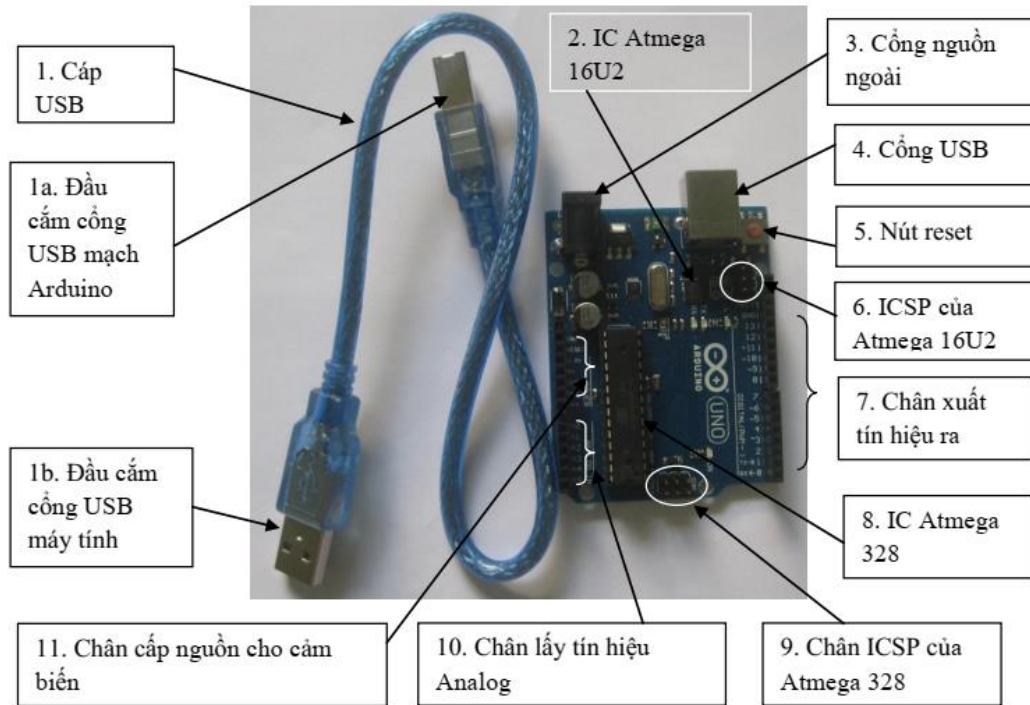
Hiện nay trên thị trường có hàng trăm board mạch Arduino khác nhau, Chúng đa phần là các biến thể PCB (các board mạch điện) của những board mạch chính đến từ nhà sản xuất Arduino. Những board mạch này hoặc có thêm 1 số tính năng cải tiến nào đó hoặc đơn giản là được thiết kế lại nhằm giảm giá thành sản phẩm để có thể tới tay người



dùng nhiều hơn. Chúng ta hãy cùng điểm qua 1 số board mạch Arduino chính như bên dưới:

### 1. Arduino Uno R3

Đây là board mạch được đánh giá là tốt nhất cho những người mới bắt đầu về điện tử và lập trình. Nó được sử dụng nhiều nhất trong các board mạch thuộc họ Arduino. Hình ảnh và các chức năng của board Arduino Uno R3 :



**Hình 1.1 Hình ảnh và các chức năng của board Arduino Uno R3 (Nguồn [www.arduino.cc](http://www.arduino.cc))**

### Giới thiệu về vi điều khiển ATmega328:

Vi điều khiển (tiếng Anh là microcontroller hoặc microcontroller unit) là trái tim của các board mạch lập trình được. Nó có khả năng thực thi code khi chúng ta yêu cầu. Bên trong vi điều khiển bao gồm rất nhiều các mạch điện với các khối chức năng như CPU (Central Processing Unit), RAM (Random Access Memory), ROM (Read Only Memory), Input/output ports, các bus giao tiếp (I2C, SPI),...

Vi điều khiển giúp chúng ta có thể giao tiếp với sensor, điều khiển thiết bị.

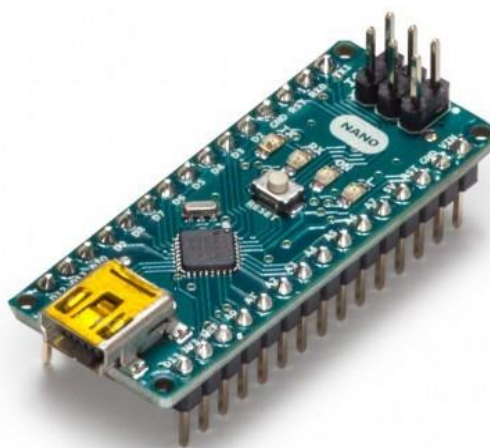
Board Arduino Uno R3 sử dụng vi điều khiển ATmega328 của hãng Atmel (một công ty thiết kế và chế tạo vật liệu bán dẫn ở Mỹ). ATmega328 là vi điều khiển thuộc dòng vi điều khiển 8 bits (data bus là 8 bit)

Dưới đây là bảng thông số kỹ thuật của board Arduino Uno R3

**Bảng 1.1. Bảng thông số kỹ thuật của ATmega32 (theo [www.arduino.cc](http://www.arduino.cc))**

Parameter	Information
<b>Microcontroller</b>	ATmega328P
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limit)</b>	6-20V
<b>Digital I/O Pins</b>	14 (of which 6 provide PWM output)
<b>PWM Digital I/O Pins</b>	6
<b>Analog Input Pins</b>	6
<b>DC Current per I/O Pin</b>	20 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P) of which 0.5 KB used by bootloader
<b>SRAM</b>	2 KB (ATmega328P)
<b>EEPROM</b>	1 KB (ATmega328P)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Length x Width</b>	68.6 mm x 53.4 mm
<b>Weight</b>	25 g

## 2. Arduino Nano



**Hình 1.2. Hình ảnh board Arduino Nano (Nguồn [www.arduino.cc](http://www.arduino.cc))**

Arduino Nano là một board mạch sử dụng chip ATmega328 (loại Arduino Nano 3.x) hoặc dùng ATmega168 (Arduino Nano 2.x), tuy nhiên có kích thước nhỏ gọn hơn để có thể tích hợp vào các hệ thống, đi kèm với đó là 1 vài điểm khác khi so sánh với board Arduino Uno R3 :

Sử dụng cổng Mini-B USB thay vì cổng USB chuẩn.

- Bổ sung thêm 2 chân Analog.
- Không có jack nguồn DC.

### 3. Arduino Leonardo



*Hình 1.3. Hình ảnh board Arduino Leonardo (Nguồn [www.arduino.cc](http://www.arduino.cc))*

Arduino Leonardo sử dụng vi điều khiển ATmega32u4, một số điểm khác biệt chính so với board Arduino Uno được liệt kê bên dưới:

- Bên trong chip ATmega32u4 được tích hợp 1 chip usb to serial thay vì phải dùng 2 MCU trên board.
- Có thể giả lập board Leonardo như chuột, bàn phím, joystick thay vì phải dùng 1 thiết bị serial khác.
- Giá thành rẻ hơn
- 20 digital I/O (7 chân PWM).
- 12 chân Analog (các chân PWM có thể được dùng như Analog)

### 4. Arduino mega2560



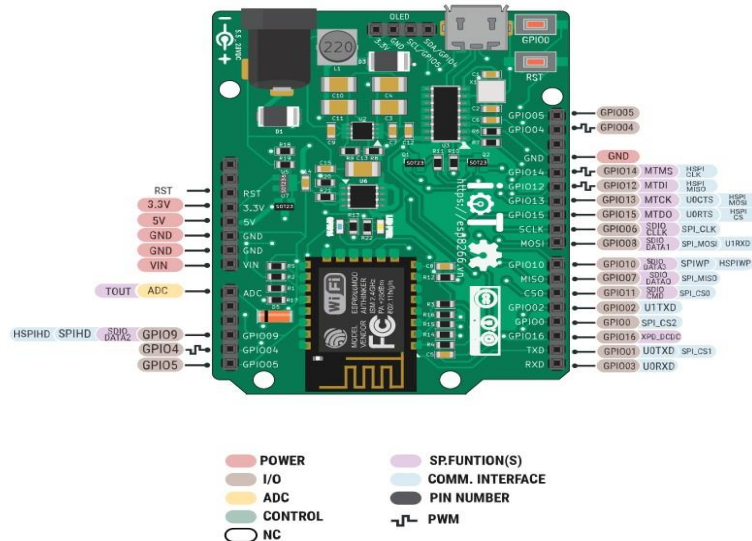
*Hình 1.4. Hình ảnh board Arduino Mega 2560 (Nguồn [www.arduino.cc](http://www.arduino.cc))*

Arduino mega2560 sử dụng chip ATmega2560 với 54 chân digital I/O (15 chân có thể dùng với PWM), 16 chân Analog, 4 UARTs,... board mạch này là phiên bản nâng cao của Arduino Uno, được dùng trong các dự án phức tạp như máy in 3D, robot,...

## 5. ESP8266 WiFi Uno

Board mạch ESP8266 WiFi Uno là một dự án mã nguồn mở giúp hỗ trợ môi trường phát triển Arduino cho ESP8266. Giúp bạn có thể viết 1 Sketch sử dụng các thư viện và hàm tương tự của Arduino, có thể chạy trực tiếp trên ESP8266 mà không cần bất kỳ vi điều khiển nào khác.

ESP8266 Arduino core đi kèm với thư viện kết nối WiFi hỗ trợ TCP, UDP và các ứng dụng HTTP, mDNS, SSDP, DNS Servers. Ngoài ra còn có thể thực hiện cập nhật OTA, sử dụng Filesystem dùng bộ nhớ Flash hay thẻ SD, điều khiển servos, ngoại vi SPI, I2C.



**Hình 1.5. Mạch phát triển ESP8266 - IoT WiFi Uno sản xuất bởi IoT Maker VietNam**

## 6. ESP8266 NodeMCU

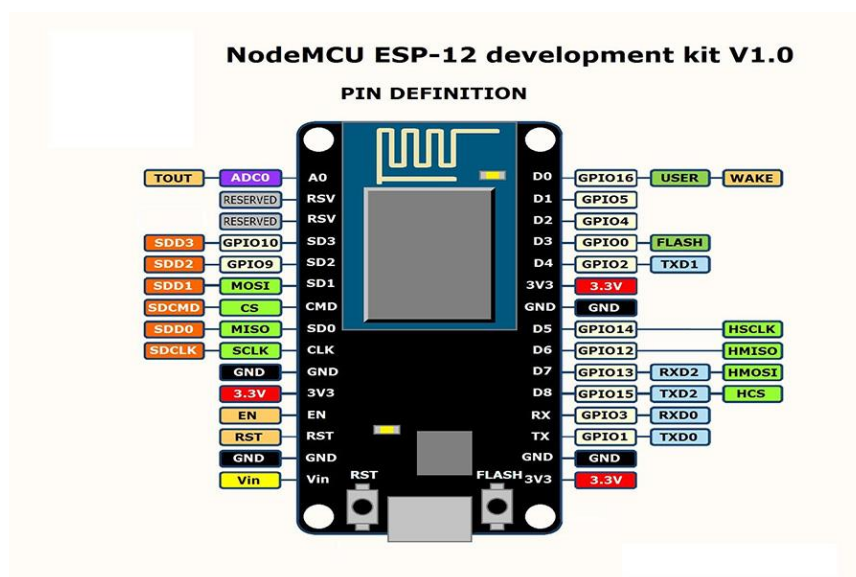
ESP8266 NodeMCU (Node MicroController Unit) là một môi trường phát triển phần mềm và phần cứng mã nguồn mở được xây dựng trên một hệ thống trên chip (SoC) được gọi là ESP8266. ESP8266, được thiết kế và sản xuất bởi Espressif Systems, chứa các thành phần quan trọng của một máy tính: CPU, RAM, mạng (WiFi), thậm chí cả hệ điều hành và SDK hiện đại.

Board mạch thu phát wifi ESP8266 NodeMCU với kích thước nhỏ gọn, giá rẻ được dùng nhiều cho các thiết bị IoT, các ứng dụng cần kết nối mạng wifi.



*Hình 1.6. ESP8266 NodeMCU*

Thông số kỹ thuật



*Hình 1.7. Mô hình ESP8266 NodeMCU*

- WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
- Điện áp hoạt động: 3.3V
- Điện áp vào: 5V thông qua cổng USB
- Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
- Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
- Bộ nhớ Flash: 4MB
- Giao tiếp: Cable Micro USB (tương đương cáp sạc điện thoại)
- Hỗ trợ bảo mật: WPA/WPA2
- Tích hợp giao thức TCP/IP
- Lập trình trên các ngôn ngữ: C/C++, Micropython,...



## CHƯƠNG II: GIỚI THIỆU ARDUINO IDE

### I. Khái niệm.

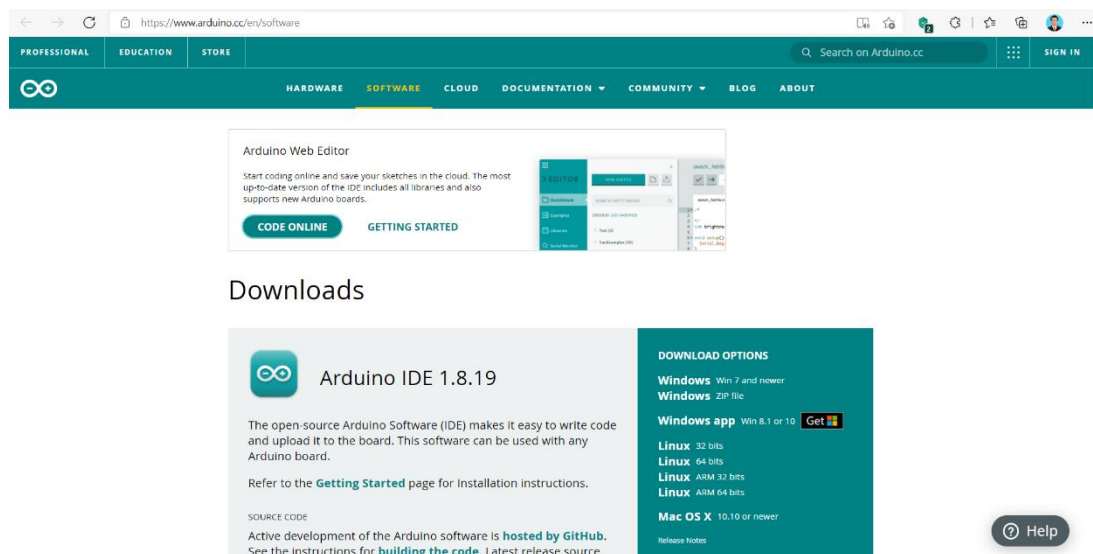
Hiểu một cách đơn giản, Arduino IDE là 1 phần mềm giúp chúng ta nạp code đã viết vào board mạch và thực thi ứng dụng. Arduino IDE là chữ viết tắt của Arduino Integrated Development Environment, một công cụ lập trình với các board mạch Arduino. Nó bao gồm các phần chính là Editor (trình soạn thảo văn bản, dùng để viết code), Debugger (công cụ giúp tìm kiếm và sửa lỗi phát sinh khi build chương trình), Compiler hoặc interpreter (công cụ giúp biên dịch code thành ngôn ngữ mà vi điều khiển có thể hiểu và thực thi code theo yêu cầu của người dùng).

### II. Cài đặt

Chúng ta đã đề cập đến tính năng và lợi ích mang lại ở mục trước, phần này sẽ hướng dẫn các bạn cách cài đặt Arduino IDE. Có 2 cách sử dụng, bao gồm sử dụng online (nếu có kết nối internet ổn định) và cài đặt offline trên máy. Khuyến cáo nên sử dụng cài đặt offline.

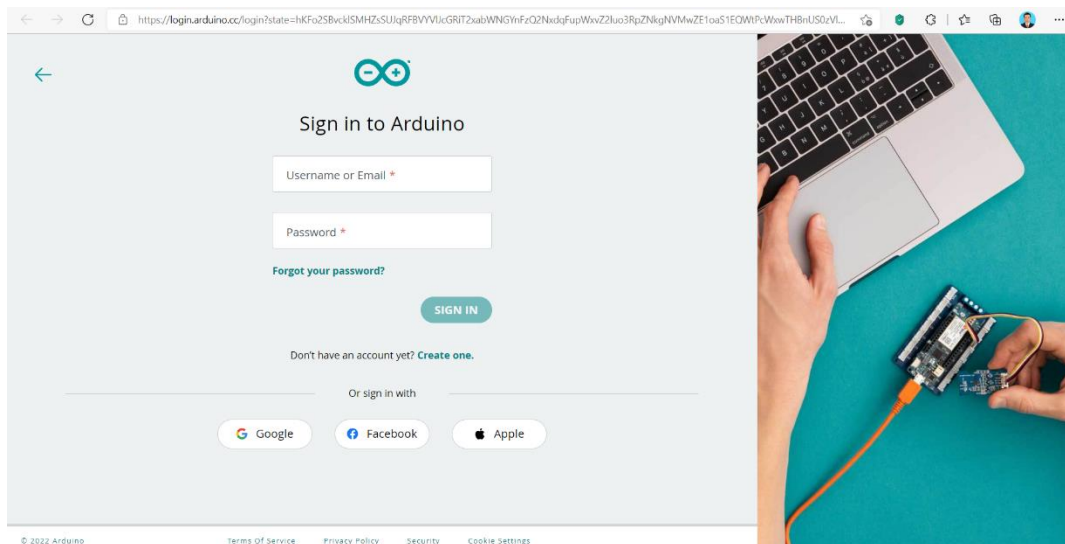
#### 1. Dùng Online IDE

**Bước 1:** Truy cập vào đường dẫn <https://www.arduino.cc/en/Main/Software>. Chọn **CODE ONLINE**.



**Hình 2.1. Giao diện Online IDE**

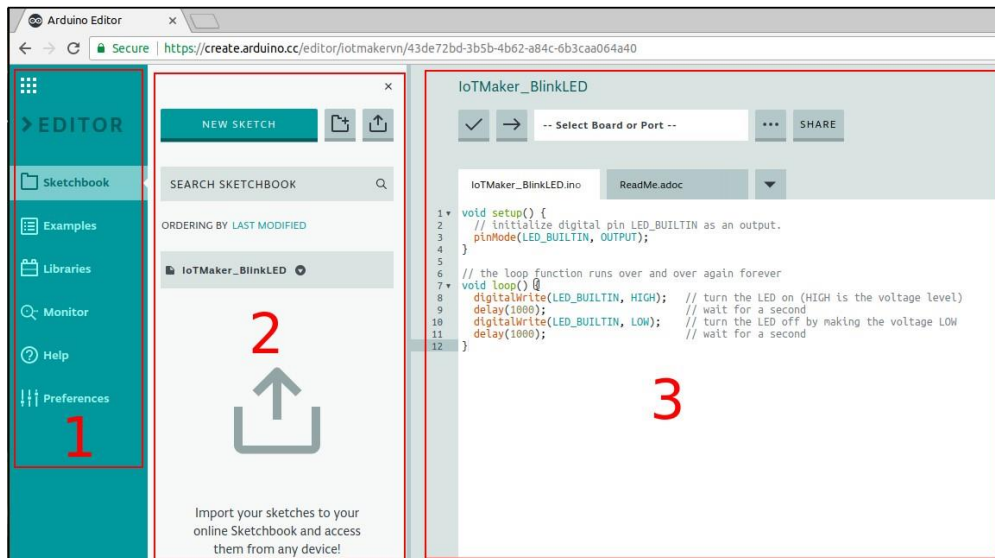
**Bước 2:** Tạo tài khoản bằng cách nhấn vào **Create one** nếu lần đầu sử dụng, đăng nhập bằng cách nhập Username và Password và nhấn vào **SIGN IN** khi đã có tài khoản. Giao diện như hình bên dưới:



**Hình 2.2. Giao diện đăng kí, đăng nhập.**

### **Bước 3: Lập trình trên Web Editor.**

Truy cập vào Arduino Web Editor như ở bước 2, thực hiện đăng nhập. Giao diện chia làm 3 phần như hình bên dưới:



**Hình 2.3. Giao diện Arduino Web Editor**

### **Mô tả các tính năng ở mục số 1:**

- **Your Sketchbook:** Giúp chúng ta có thể thấy các sketch (các chương trình trong Arduino được gọi là Sketch)

- **Examples:** Đây là những sketch tham khảo nhằm giúp người dùng xem các chức năng cơ bản của các thư viện Arduino, các thư viện này mặc định ở chế độ read-only (chỉ đọc và không cho phép chỉnh sửa).
- **Libraries:** Cho phép chúng ta "include" những thư viện vào trong sketch để thực hiện các chức năng theo nhu cầu sử dụng.
- **Serial monitor:** Cho phép truyền nhận dữ liệu của board thông qua USB cable.
- **Help:** Cung cấp các hướng dẫn để bắt đầu lập trình Arduino Web Editor.
- **Preferences:** Những cài đặt về các thuộc tính của trình soạn thảo code đang sử dụng, như cỡ chữ, màu nền,...

### **Mô tả các tính năng ở mục số 2:**

Trong mục này hiển thị các folder, sketch, nó cũng bao gồm các tùy chọn như tạo 1 folder mới, tạo sketch mới, import 1 sketch khác từ máy tính lên Arduino Web Editor.

### **Mô tả các tính năng ở mục số 3:**

Phần này bao gồm trình soạn thảo code và các option để có thể nạp code vào board, bao gồm:

- Nút **Verify:** Giúp biên dịch các file của chương trình, sẽ có thông báo lỗi nếu phát sinh lỗi trong code.
- Nút **Upload:** Upload code vào board Arduino, quá trình này bao gồm cả biên dịch các file trong sketch.
- Hộp thoại **Select board and port:** Cần chọn board và port để nạp code.
- Nút **...:** Có các tùy chọn liên quan đến sketch như lưu, xóa, đổi tên, ...
- Nút **Share:** Cho phép chúng ta chia sẻ sketch tới người khác.

Việc tạo 1 dự án trên Arduino Web Editor là khá đơn giản, chỉ cần tạo 1 sketch mới, viết code trên vùng soạn thảo, chọn đúng board, port và nạp chương trình. Chi tiết các bước thực hiện có thể xem tại [getting-started-with-arduino-web-editor](#).

## **2. Dùng Offline**

### **Cài đặt**

**Bước 1:** Download Arduino IDE.

Truy cập đến trang chủ [www.arduino.cc](http://www.arduino.cc), tùy hệ điều hành đang làm việc mà chọn gói cài đặt thích hợp.

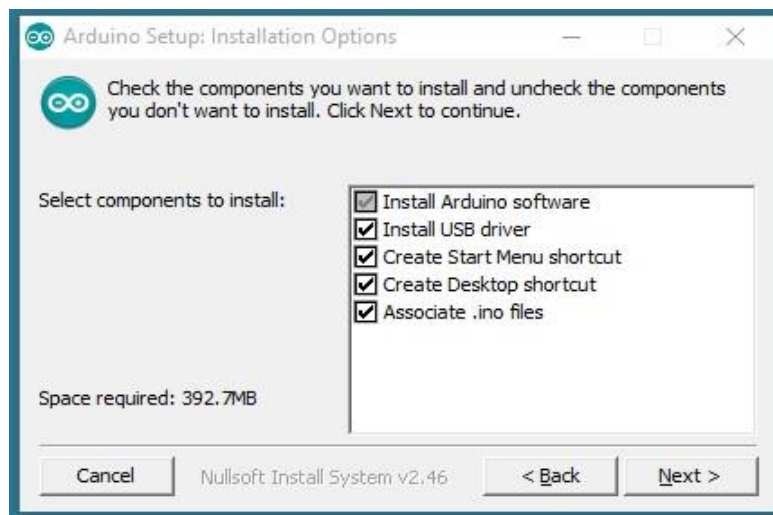
**Bước 2:** Cài đặt Arduino IDE vào máy tính.



## 2.1. Với hệ điều hành Windows:

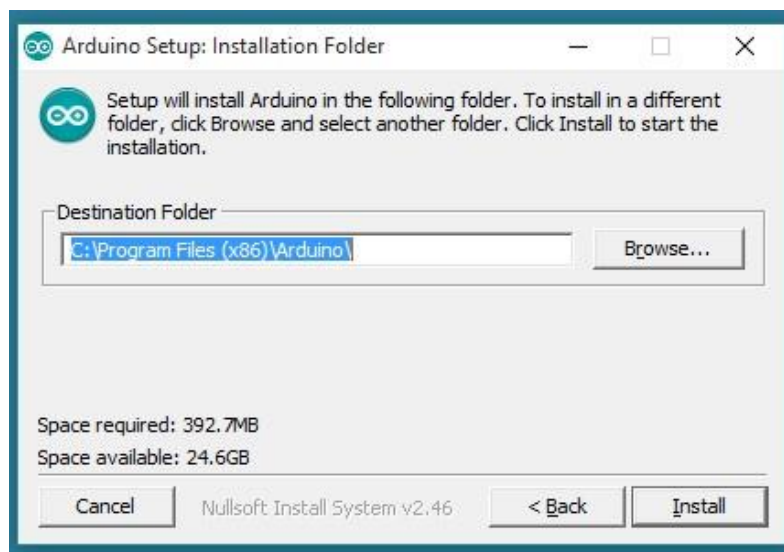
Chúng ta có thể download phiên bản Windows Installer (.exe) hoặc Windows Zip package. Installer giúp cài đặt trực tiếp mọi thứ ta cần bao gồm cả driver (khuyến dùng), với Zip package thì cần giải nén tập tin và cài đặt bằng tay. Tuy nhiên Zip package hữu ích khi muốn cài đặt với **phiên bản Portable**.

Sau khi download hoàn thành, tiến hành cài đặt chương trình. Cho phép quá trình cài đặt driver lên máy tính nếu có thông báo từ hệ điều hành. Check vào các hộp thoại để cài đặt các thành phần đi kèm như hình bên dưới:

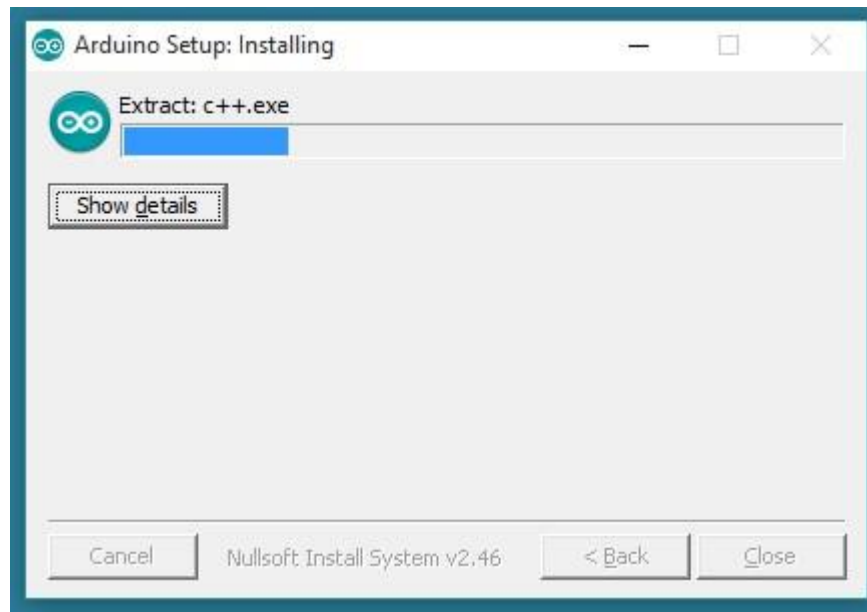


**Hình 2.4. Cài đặt Arduino IDE (Nguồn [www.arduino.cc](http://www.arduino.cc))**

Tiếp theo, chọn thư mục cài đặt (nên để theo mặc định) và chờ quá trình cài đặt hoàn tất.



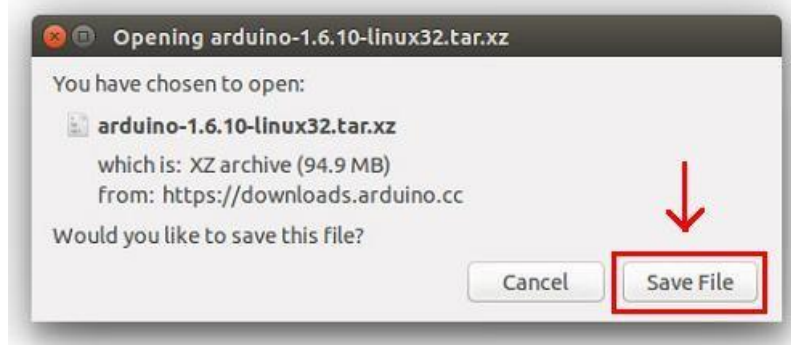
**Hình 2.5. Chọn thư mục để cài đặt Arduino IDE (Nguồn [www.arduino.cc](http://www.arduino.cc))**



**Hình 2.6. Quá trình giải nén và cài đặt Arduino lên máy tính (Nguồn [www.arduino.cc](http://www.arduino.cc))**

## **2.2. Với hệ điều hành Linux:**

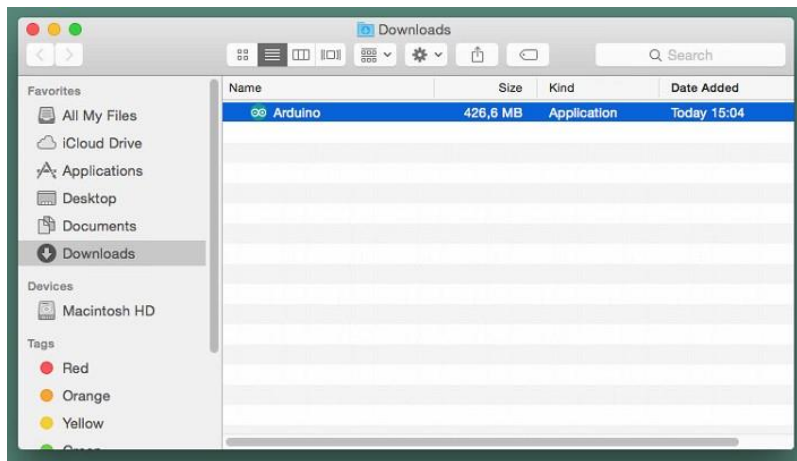
Lựa chọn phiên bản của hệ điều hành để chọn gói cài đặt thích hợp (Linux 32 bits, Linux 64 bits hoặc Linux ARM). Chọn **Save File** để download phần mềm về máy như hình dưới:



**Hình 2.7. Download phần mềm Arduino (Nguồn [www.arduino.cc](http://www.arduino.cc))**

## **2.3. Với hệ điều hành MacOSX**

Download phần mềm về máy và giải nén nó. Nếu sử dụng trình duyệt Safari thì sau khi download nó sẽ tự động giải nén.

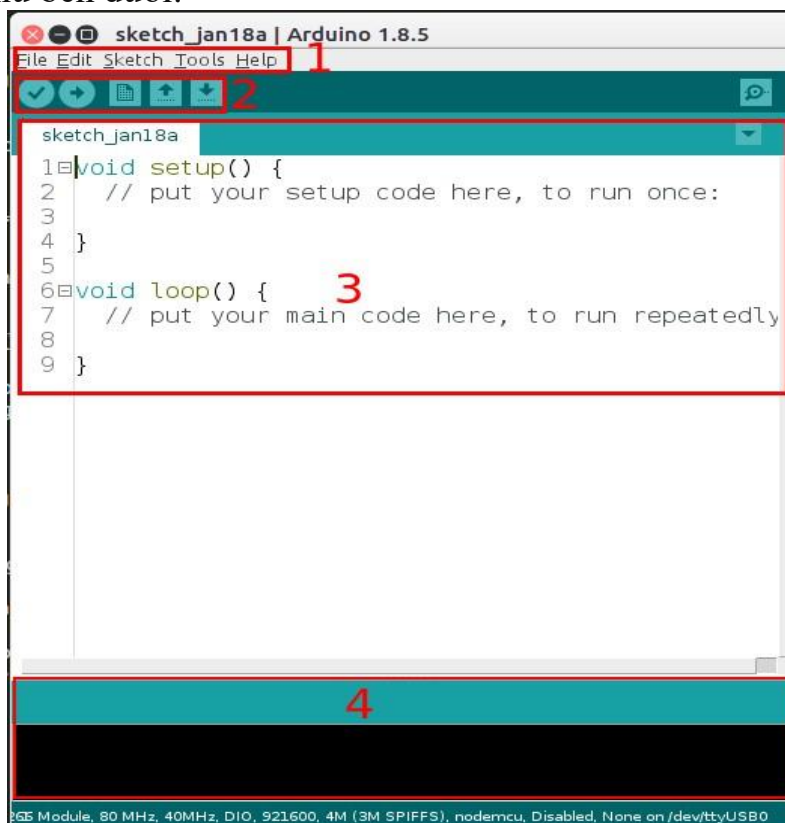


**Hình 2.8. Hình ảnh phần mềm Arduino sau khi download (Nguồn [www.arduino.cc](http://www.arduino.cc))**

### **III. Sử dụng Arduino IDE**

#### **1. Khởi động**

Click vào icon Arduino để khởi động Arduino IDE, sau khi khởi động, phần mềm sẽ có giao diện như bên dưới:



**Hình 2.9. Hình ảnh giao diện Arduino IDE**

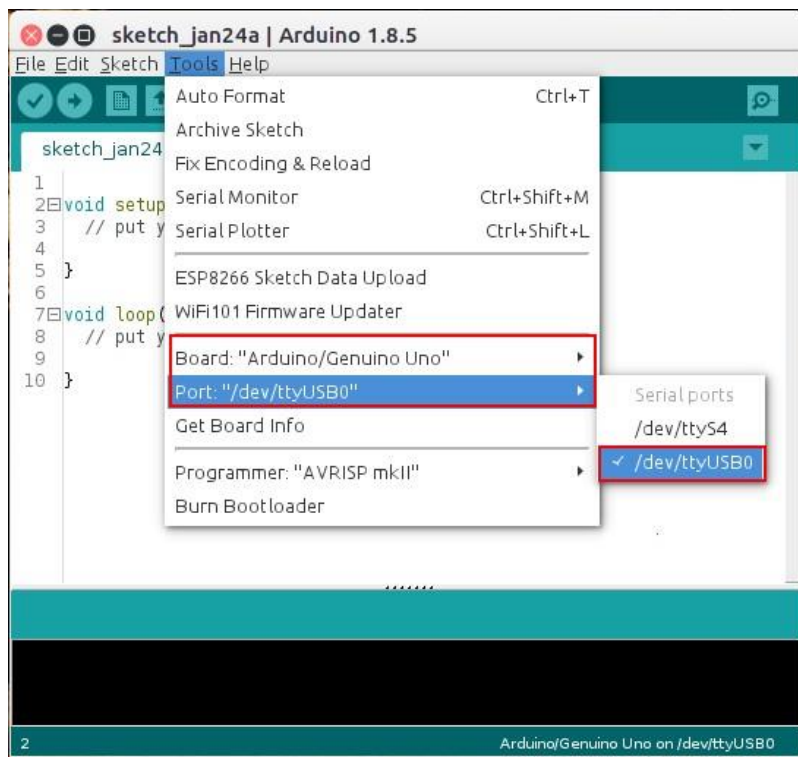
- **Mục 1:** Là thanh menu bar, bao gồm các tùy chọn thiết lập cho phần mềm Arduino và cho sketch đang thực hiện.
- **Mục 2:** Là thanh symbol bar, gồm các nút nhấn **Verify** để biên dịch sketch, **Upload** để nạp sketch vào board, **New** để tạo sketch mới, **Open** để mở sketch, **Save** lưu sketch và **Serial Monitor** để mở serial port.
- **Mục 3:** Vùng để soạn thảo code cho sketch.
- **Mục 4:** Vùng hiển thị thông tin khi biên dịch, hiển thị quá trình nạp sketch và các thông báo lỗi khi biên dịch sketch (nếu có).

Sử dụng cable micro USB kết nối với máy tính và board như hình bên dưới:



**Hình 2.10. Hình ảnh kết nối board với máy tính.**

Trên thanh menu bar chọn **Tools**, ở mục **Boards** chọn **Arduino/Genuino Uno**, mục **Port** chọn cổng micro USB đã kết nối vào máy tính, tùy thuộc vào hệ điều hành mà các port này có tên gọi khác nhau, trên Linux thường là **/dev/ttyUSB0**, **/dev/ttyUSB0,...**, với Windows thường là **COM1**, **COM2,...**, với Mac OS thường là **/dev/tty.wchusbserial1420,...**



**Hình 2.11. Hình ảnh cấu hình cho board trên Arduino**

Sau đó, chúng ta đã có thể viết source code, biên dịch và nạp code lên board.

## 2. Cấu trúc chương trình

Ngay khi bạn mở Arduino IDE, bạn sẽ được chào đón bởi các lệnh **setup()** và **loop()**.



**Hình 2.12. Màn hình code ban đầu của Arduino IDE**

### 1.1. Phần khai báo:

- Khai báo thư viện: **#include** <Tên thư viện>
- Định nghĩa (gán) chân kết nối: **#define** <Định danh> <Giá trị>
- Khai báo biến: <Kiểu dữ liệu> <Tên biến>;

Ví dụ: `int pos = 0;`

- Khai báo hằng: **const** <Kiểu dữ liệu> <Tên hằng> = <Giá trị>;

Ví dụ: `const long interval = 1000;`

- ✓ Phần khai báo có thể có hoặc không, tùy mục đích sử dụng của chương trình.
- ✓ Khai báo biến, hằng có thể khai báo bất kỳ vị trí nào trong chương trình, miễn là phải khai báo trước khi sử dụng nó.

### 1.2. Setup()

Mỗi khi sketch bắt đầu, lệnh setup sẽ giúp khởi tạo các biến và bắt đầu sử dụng các thư viện.

```
void setup() {  
  // Những lệnh trong hàm setup() sẽ được chạy khi Arduino khởi động  
  // chỉ thực hiện một lần duy nhất.  
}
```

### 1.3. Loop():

Một vòng lặp theo sau setup và thực sự là trái tim của chương trình. Dùng để viết các lệnh trong chương trình để mạch Arduino thực hiện các nhiệm vụ mà chúng ta mong muốn

```
void loop() {  
  // đây là nơi bạn đặt code cần chạy mọi lúc.  
  // sẽ được chạy và lặp đi lặp lại cho đến khi reset hoặc shutdown Arduino  
  // vd: kiểm tra xem đã đến lúc nhấp nháy đèn LED chưa...  
}
```

## 3. Một số lệnh điều khiển cơ bản

Sau khi làm quen với các sketch, bạn cần biết những lệnh điều khiển. Những lệnh quan trọng nhất là:

- **break:** Dùng khi muốn thoát khỏi một lệnh vô điều kiện.

```
if (sens > threshold) {  
  // bail out on sensor detect x = 0;
```



```
break;  
}
```

- **if - else:** Các lệnh logic khởi tạo một hành động mỗi khi một điều kiện được thỏa mãn.

```
if(điều kiện){Các câu lệnh}  
else{Các câu lệnh};
```

- **return:** Trả về một giá trị nhất định.
- **while:** Một vòng lặp diễn ra liên tục trong một điều kiện nhất định.  

```
while <Điều kiện> {Các câu lệnh};
```
- **goto:** Lệnh này cho phép bạn đi đến một dòng nhất định trong code.

## 4. Hàm và thủ tục

### 4.1. Các hàm Arduino tích hợp sẵn

- **digitalRead():** Đọc giá trị từ một pin kỹ thuật số nhất định. Ngoài ra còn có **digitalWrite()**.
- **pinMode(PIN\_NUMBER, INPUT/OUTPUT):** Thiết lập pin tại vị trí PIN\_NUMBER thành INPUT hoặc OUTPUT.
- **pinMode(PIN\_NUMBER, INPUT\_PULLUP):** Thiết lập pin tại vị trí PIN\_NUMBER trở thành đầu vào bằng cách sử dụng điện trở kéo lên (pull-up resistor) tích hợp của bo mạch Arduino.
- **analogRead(PIN\_NUMBER):** Đọc số chân analog PIN\_NUMBER và trả về một số nguyên từ 0 đến 1023.
- **analogWrite(PIN\_NUMBER, VALUE):** Giả lập analog output VALUE bằng cách sử dụng PWM trên PIN\_NUMBER (Lưu ý: Chỉ khả dụng trên các chân 3, 5, 6, 9, 10 và 11).
- **analogReference(DEFAULT):** Sử dụng điện áp tham chiếu mặc định (5V hoặc 3.3V tùy thuộc vào điện áp bo mạch).
- **analogReference(INTERNAL):** Sử dụng điện áp tham chiếu bên trong (1.1v cho ATmega168/328p, 2.56 cho ATmega 32U4/8).
- **analogReference(EXTERNAL):** Sử dụng điện áp được áp dụng cho chân AREF làm điện áp tham chiếu (Lưu ý: chỉ 0-5V).

### 4.2. Các hàm liên quan đến thời gian

- **Delay():** Tạo độ trễ.
- **millis():** Trả về thời gian tính bằng mili giây kể từ khi Arduino sketch bắt đầu chạy dưới dạng một số nguyên dài không dấu.
- **micros():** Trả về thời gian tính bằng micro giây kể từ khi Arduino sketch bắt đầu chạy dưới dạng một số nguyên dài không dấu.

- **delayMicroseconds (số nguyên):** Trì hoãn thực thi chương trình trong (số nguyên) micro giây.

#### 4.3. Các hàm toán học

- **min(i, j):** Trả về giá trị thấp nhất trong hai giá trị i và j
- **max(i, j):** Trả về giá trị cao nhất trong hai giá trị i và j
- **abs(i):** Trả về giá trị tuyệt đối của i
- **sin(angle):** Trả về sin của một góc tính bằng radian
- **cos(angle):** Trả về cosin của một góc tính bằng radian
- **tan(angle):** Trả về tang của một góc tính bằng radian
- **sqrt(i):** Trả về căn bậc hai của i
- **pow(cơ số, số mũ):** Tính lũy thừa bậc (số mũ) của (cơ số).
- **constrain(i, minval, maxval):** Ràng buộc giá trị i giữa minval (giá trị nhỏ nhất) và maxval (giá trị lớn nhất)
- **map (val, fromL, fromH, toL, toH):** Chuyển đổi val từ phạm vi này sang phạm vi khác
- **random(i):** Trả về một số nguyên dài ngẫu nhiên nhỏ hơn i
- **random(i, j):** Trả về một số nguyên dài ngẫu nhiên giữa i và j
- **randomSeed(k):** Sử dụng giá trị k để seed hàm random()

#### 4.4. Giao tiếp nối tiếp (serial communication)

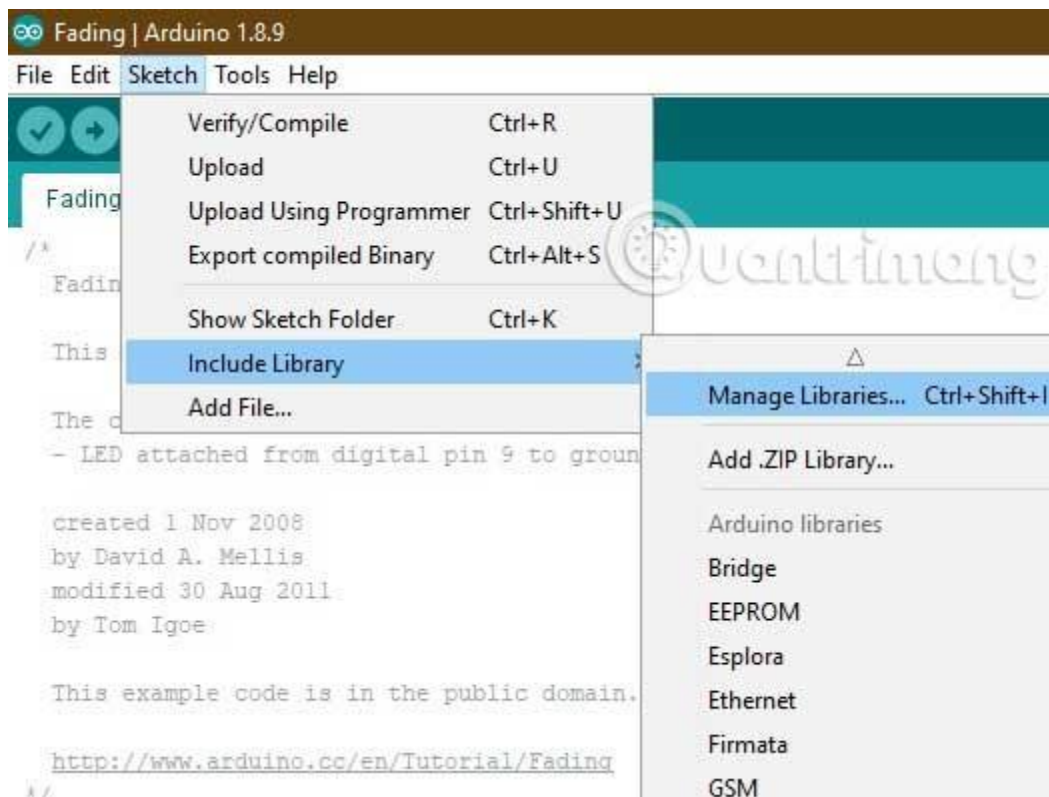
- **Serial.begin(speed):** Bắt đầu giao tiếp nối tiếp ở tốc độ xác định
- **Serial.end():** Kết thúc Kết nối giao tiếp nối tiếp
- **Serial.print(DATA):** Xuất DATA vào cổng nối tiếp. DATA có thể là ký tự, chuỗi, số nguyên và số thực dấu phẩy động (floating point number)
- **Serial.available():** Trả về số ký tự có sẵn để đọc trong bộ đệm nối tiếp
- **Serial.read():** Đọc ký tự đầu tiên trong bộ đệm nối tiếp (trả về -1 nếu không có dữ liệu)
- **Serial.write(DATA):** Ghi DATA vào bộ đệm nối tiếp. DATA có thể là một ký tự, số nguyên hoặc array (mảng)
- **Serial.flush():** Xóa bộ đệm nối tiếp sau khi giao tiếp chiều đi hoàn tất

### IV. Cài đặt thư viện

#### 1. Cài đặt thư viện Arduino bằng Arduino Library Manager

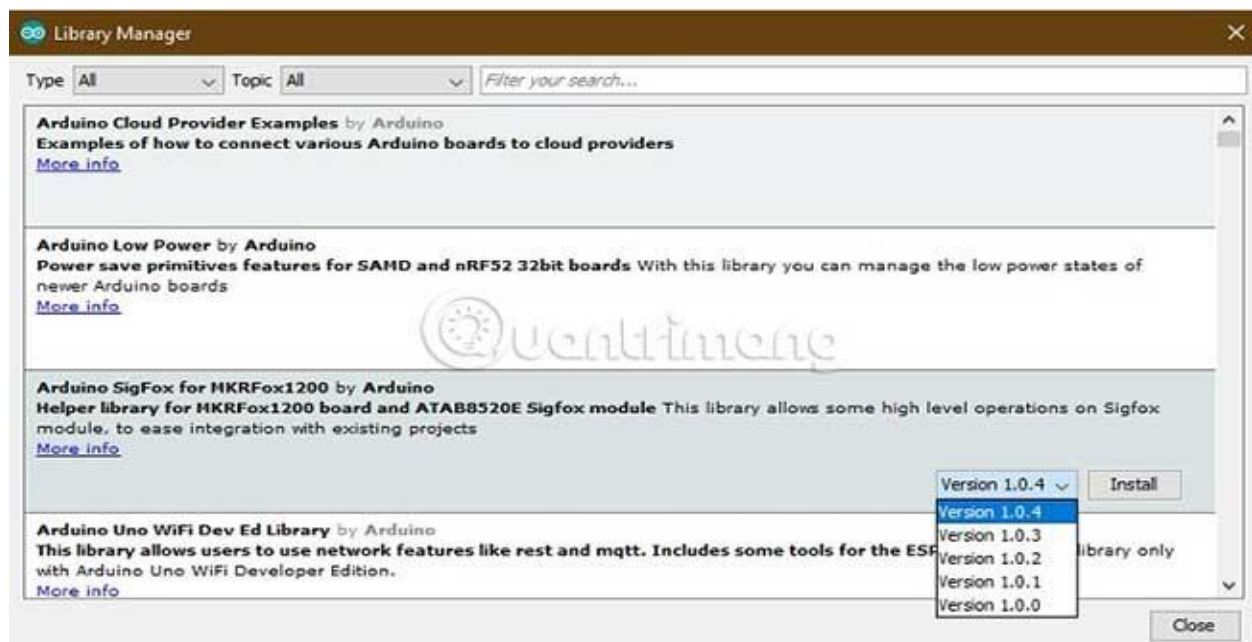
Để thêm các thư viện Arduino từ trình quản lý thư viện riêng của IDE, hãy truy cập vào **Sketch > Include Library > Manage Libraries**.





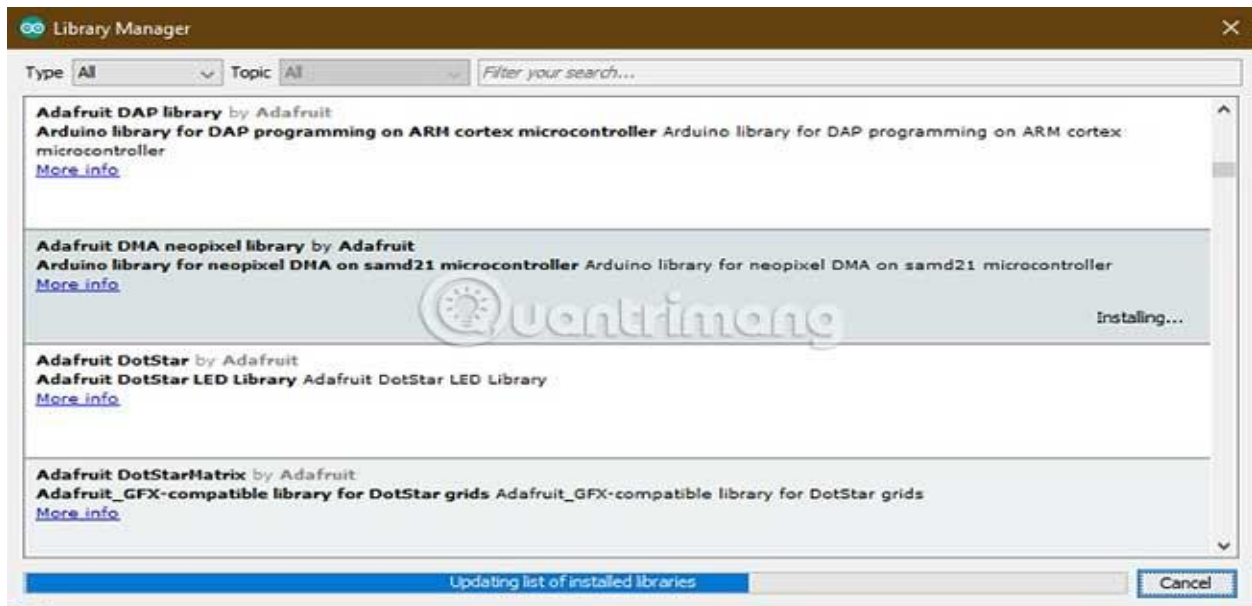
**Hình 2.13. Cài đặt thư viện Arduino bằng Arduino Library Manager**

Ta có thể thấy một loạt các thư viện drop-down trong **Library Manager**. Để cài đặt một trong số chúng, hãy cài đặt nó cho phiên bản mới nhất hiện có.



**Hình 2.14. Arduino Library Manager**

Việc cài đặt chỉ mất khoảng vài giây. Dội **Library Manager** cập nhật danh sách các thư viện.



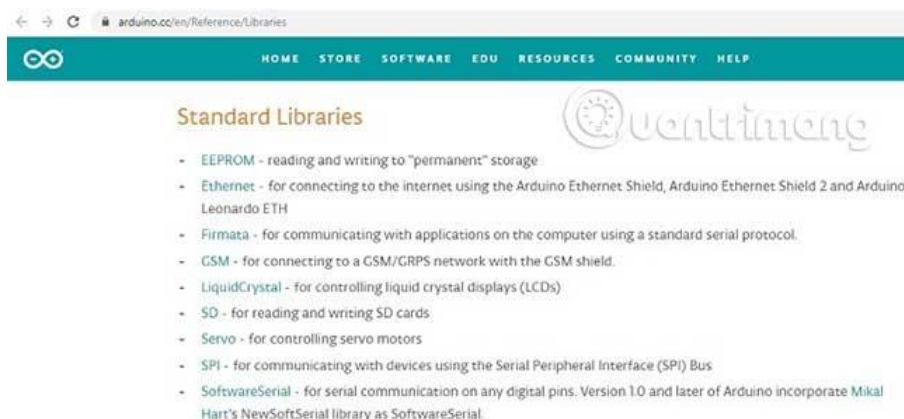
**Hình 2.15. Arduino Library Manager cập nhật thư viện**

Khai báo thư viện khi cần sử dụng nó:

```
#Include <Servo.h>
```

## 2. Cài đặt thư viện Arduino từ các trang web khác

Trang web chính thức của Arduino duy trì một danh sách tham khảo các thư viện có sẵn để tải xuống và cài đặt. Đây là những tùy chọn hiển thị trong quy trình được mô tả ở trên.

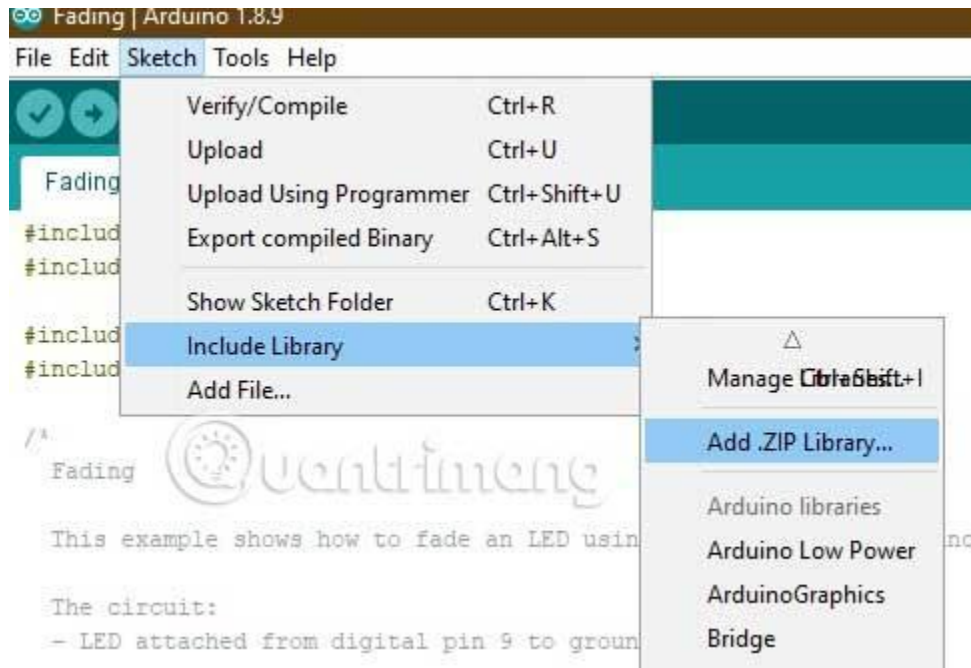


**Hình 2.16. Thư viện từ <https://www.arduino.cc/reference/en/libraries/>**

Nếu muốn thiết lập thư viện của bên thứ ba trong Arduino IDE, bạn phải tìm kiếm nó trực tuyến. GitHub là một nơi tốt để tìm thấy chúng.

Bạn sẽ được cung cấp một tùy chọn để tải xuống file zip. Lưu file trên máy tính nhưng đừng giải nén nó.

Trong bước tiếp theo, bạn có thể thêm file zip từ **Sketch > Include Library > Add .ZIP Library**.



**Hình 2.16. Thêm Thư viện từ file zip**

# CHƯƠNG III: THỰC CHIẾN LẬP TRÌNH ĐIỀU KHIỂN CÁC MODULE NGOẠI VI

## I. Module cảm biến

### 1. Cảm biến ánh sáng dùng quang trở

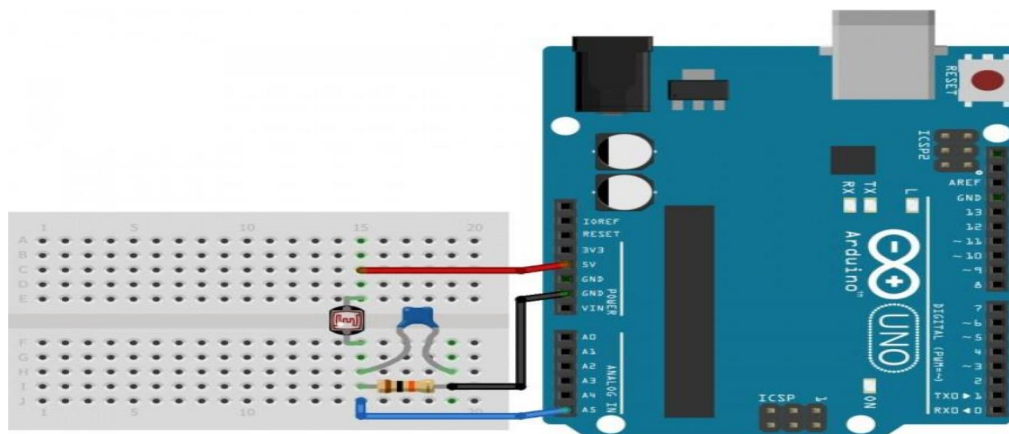
Quang trở là một loại "vật liệu" điện tử rất hay gặp và được sử dụng trong những mạch cảm biến ánh sáng. Có thể hiểu một cách dễ dàng rằng, quang trở là một loại ĐIỆN TRỞ có điện trở thay đổi theo cường độ ánh sáng. Nếu đặt ở môi trường có ít ánh sáng, có bóng râm hoặc tối thì điện trở của quang trở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



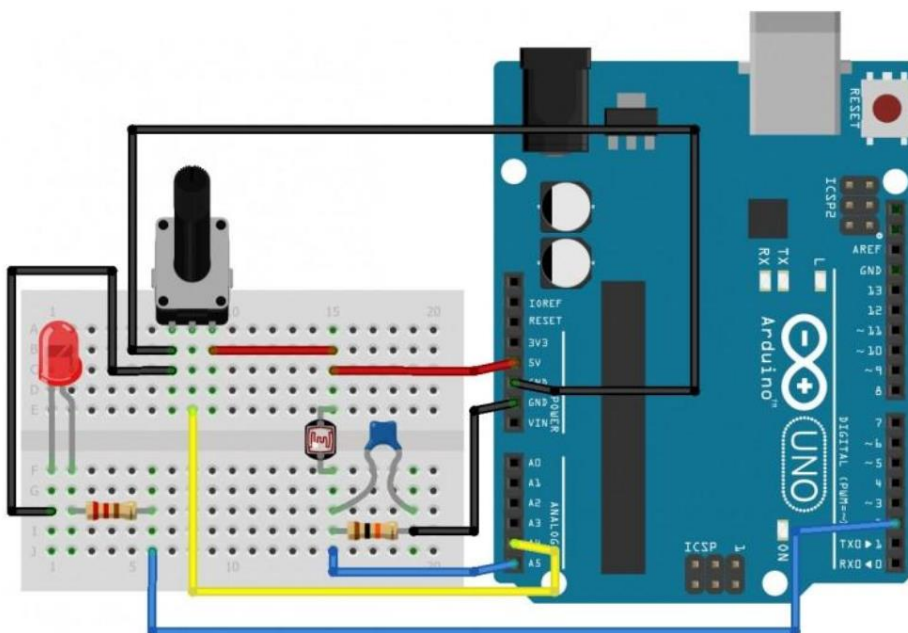
*Hình 3.1. Sơ đồ nối chân quang trở 1*

Bây giờ hãy 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).

**Code1:**

<https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Quangtro1.ino>

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 (trong quá trình lắp mạch không cần gắn tụ cũng được).



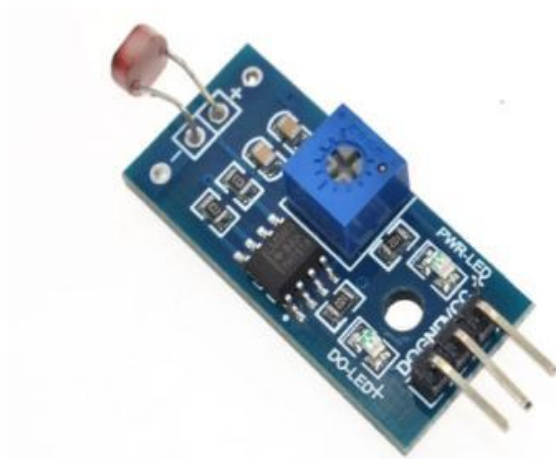
*Hình 3.2. Sơ đồ nối chân quang trở 2*

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

**Code 2:**

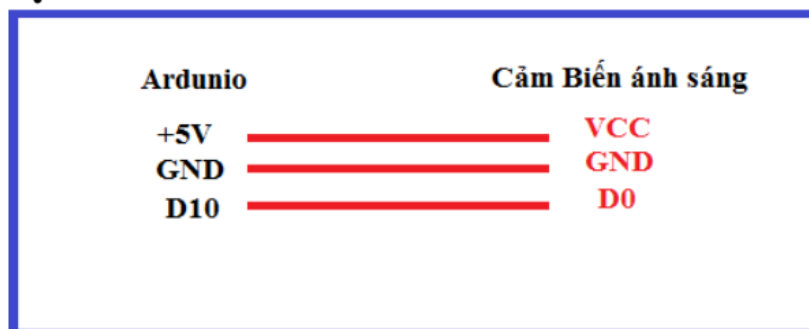
<https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Quangtro2.ino>

Trên thị trường hiện nay đa số dùng kit quang trở đã được tích hợp sẵn:



**Hình 3.2. Module quang trở**

**Sơ đồ nối dây:**



**Hình 3.3. Sơ đồ nối dây Module quang trở**

**Code 3:**

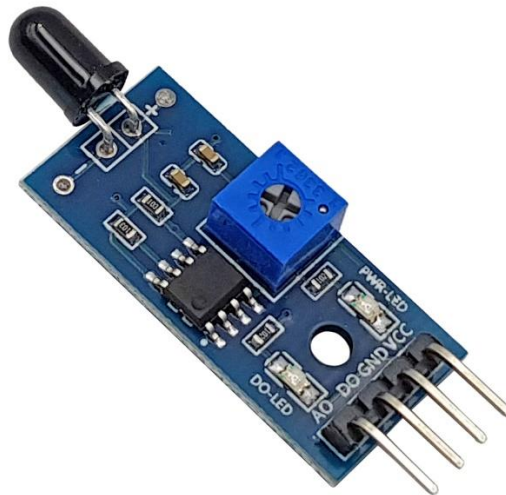
[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Quangtro\\_module.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Quangtro_module.ino)

## **2. Cảm biến phát hiện lửa (Flame Sensor - báo cháy)**

**TÍNH NĂNG:**

- Khả năng phát hiện lửa hoặc nguồn sáng có bước sóng tương tự.
- Sử dụng cảm biến hồng ngoại YG1006 với tốc độ đáp ứng nhanh và độ nhạy cao.
- Tích hợp IC LM393 để chuyển đổi ADC, tạo 2 ngõ ra cả số và tương tự, rất linh động trong việc sử dụng.
- Biến trở để tùy chỉnh độ nhạy cảm biến.
- Có thể ứng dụng trong các hệ thống báo cháy, robot chữa cháy,...





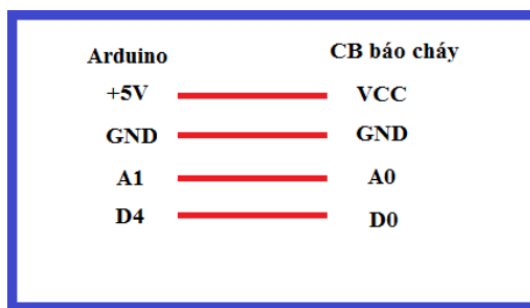
**Hình 3.4. Cảm biến phát hiện lửa (flame sensor)**

### THÔNG SỐ CẢM BIẾN PHÁT HIỆN LỬA

- Nguồn cấp: 3.3V – 5VDC
- Dòng tiêu thụ: 15mA
- Tín hiệu ra: Digital 3.3 – 5VDC tùy nguồn cấp hoặc Analog.
- Khoảng cách: 80cm
- Góc quét: 60 độ
- Kích thước: 3.2 x 1.4 cm

### SƠ ĐỒ CHÂN

- VCC → 3.3V ~ 5.3V
- GND → power supply ground
- AOUT (AO) → analog output
- DOUT (DO) → digital output



**Hình 3.5. Sơ đồ nối dây cảm biến phát hiện lửa (flame sensor)**

Code:

<https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cambienbaochay.ino>

### 3. Cảm biến dò line



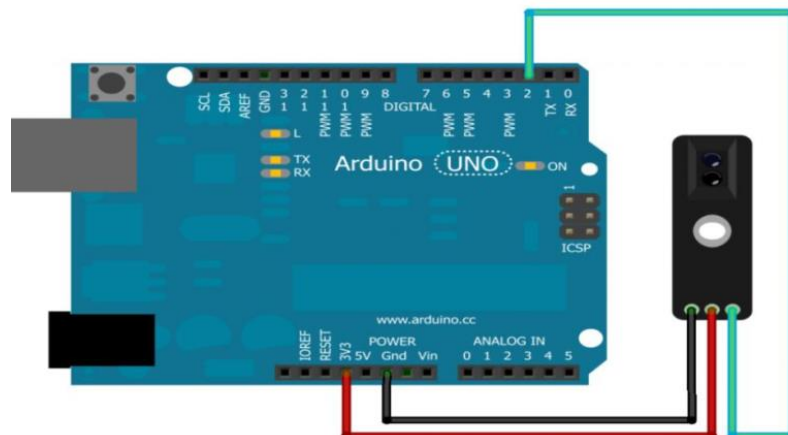
*Hình 3.6. Cảm biến dò line đơn KY-033*

#### THÔNG SỐ MODULE CẢM BIẾN DÒ LINE ĐƠN

- Nguồn cung cấp: 5VDC.
- Dòng điện tiêu thụ: <10mA.
- Dải nhiệt độ hoạt động: 0oC ~ 50oC.
- Ngõ giao tiếp: 3 dây VCC, GND, S

#### SƠ ĐỒ CHÂN

Arduino		CB dò line
+5V	—	VCC
GND	—	GND
D2	—	S



*Hình 3.7. Sơ đồ nối dây cảm biến dò line đơn*



### Code:

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam\\_bien\\_do\\_line.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam_bien_do_line.ino)

## 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, chúng ta 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 đề tài này, tôi chỉ đề cập đến phần làm thế nào để đọc cảm biến mưa thôi.



*Hình 3.8. Cảm biến mưa*

### *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.

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).

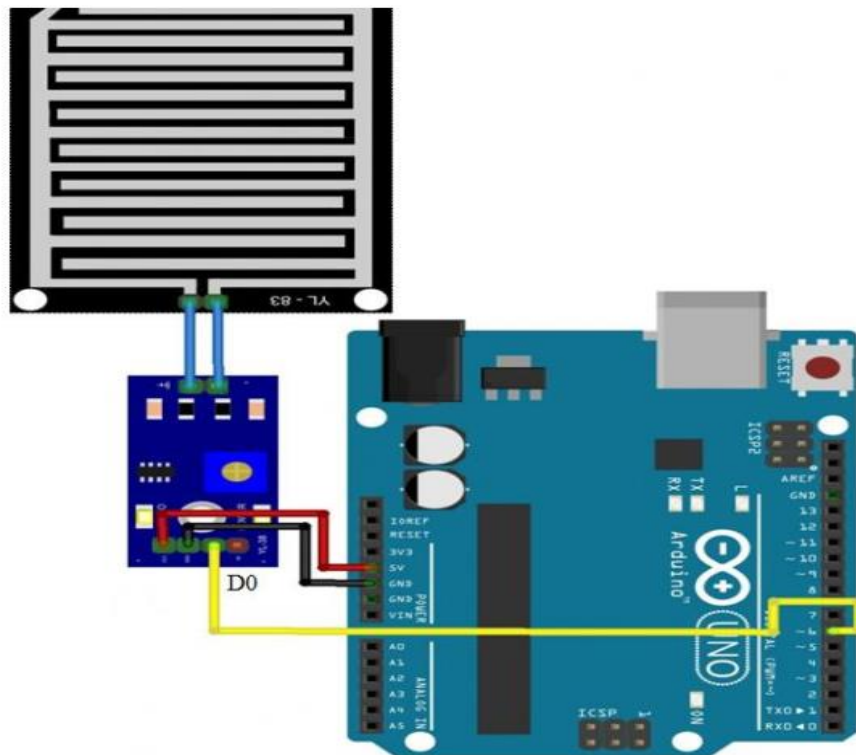
### THÔNG SỐ KỸ THUẬT

- Điện áp: 5V
- Led báo nguồn (Màu xanh)
- Led cảnh báo mưa (Màu đỏ)

- Hoạt động dựa trên nguyên lý: Nước rơi vào board sẽ tạo ra môi trường dẫn điện.
- Có 2 dạng tín hiệu: Analog( AO) và Digital (DO)
- Dạng tín hiệu : TTL, đầu ra 100mA ( Có thể sử dụng trực tiếp Relay, Còi công suất nhỏ...)
- Điều chỉnh độ nhạy bằng biến trở.
- Sử dụng LM358 để chuyển AO → DO
- Kích thước: 5.4\*4.0mm
- Dày 1.6mm

### SƠ ĐỒ CHÂN

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



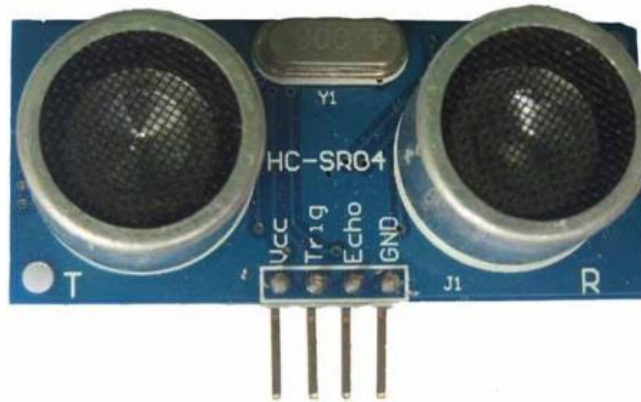
*Hình 3.9. Sơ đồ nối dây cảm biến mưa*

**Code:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam\\_bien\\_mua.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam_bien_mua.ino)

## 5. Cảm biến siêu âm

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.



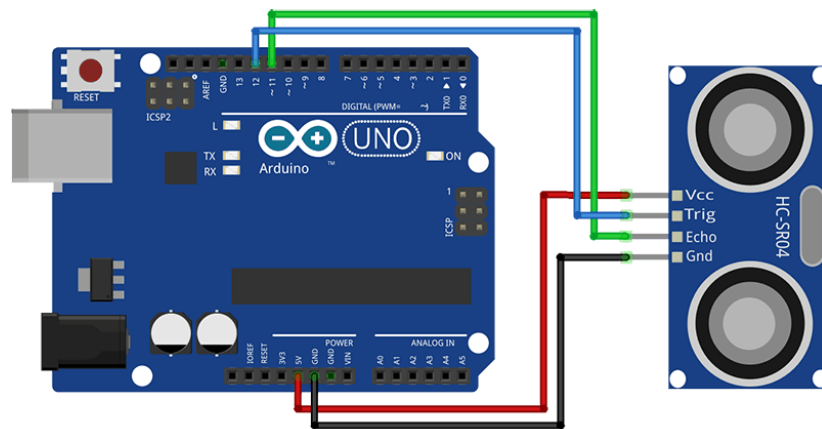
*Hình 3.10. Cảm biến siêu âm HC-SR04*

### THÔNG SỐ KỸ THUẬT CẢM BIẾN SIÊU ÂM HC-SR04

- Điện áp: 5V DC
- Dòng hoạt động: < 2mA
- Mức cao: 5V
- Mức thấp: 0V
- Góc tối đa: 15 độ
- Khoảng cách: 2cm – 450cm (4.5m)
- Độ chính xác: 3mm

### SƠ ĐỒ CHÂN

Arduino		CB siêu âm HC-SR04
+5V	=====	VCC
GND	=====	GND
11	=====	Echo
12	=====	Trig



*Hình 3.11. Sơ đồ nối dây cảm biến siêu âm*

**Code:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam\\_bien\\_sieu\\_am.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam_bien_sieu_am.ino)

### **GIẢI THÍCH**

- ***duration = pulseIn(echo,HIGH);*** hàm ***pulseIn()*** được dùng để đo độ rộng xung HIGH ở chân echo.

- ***distance = int(duration/2/29.412);*** Thời gian sóng truyền từ cảm biến đến vật sẽ bằng  $\text{duration}/2$ , sau đó ta chia tiếp cho 29,412 để tính khoảng cách.

### **6. 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.



*Hình 3.12. Cảm biến âm thanh KY-037*

### **THÔNG SỐ KỸ THUẬT CỦA CẢM BIẾN ÂM THANH KY-037**

- Điện áp: 3V-15V
- Chân dương (+): kết nối 5V của **arduino**
- Chân âm (-): kết nối GND của arduino

- Chân D0: Đầu ra tín hiệu số
- Chân A0: Đầu ra tương tự
- Độ nhạy có thể điều chỉnh bằng chiết áp trên mạch

## SƠ ĐỒ CHÂN

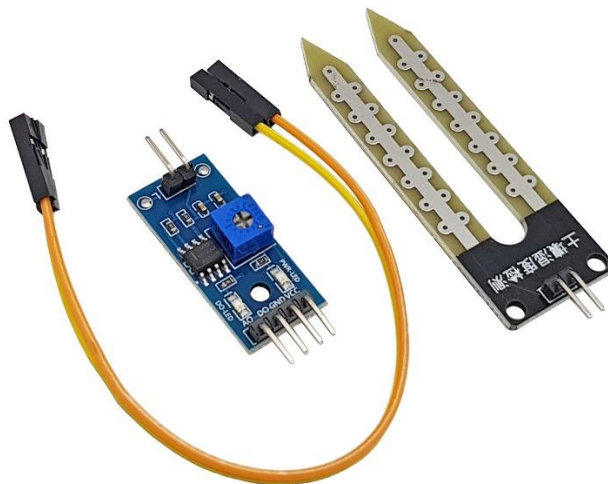
Arduino		CB âm thanh
+5V	—	VCC
GND	—	GND
A1	—	A0
D4	—	D0
D12 NỐI VỚI LED		

**Code:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam\\_bien\\_am\\_thanh.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam_bien_am_thanh.ino)

## 7. 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



*Hình 3.13. Cảm biến độ ẩm đất*

## 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

## SƠ ĐỒ CHÂN

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)



**D12 NỐI VỚI CHÂN KÍCH    ROLE**

*Hình 3.14. Sơ đồ nối dây cảm biến độ ẩm đất*

**Code:**

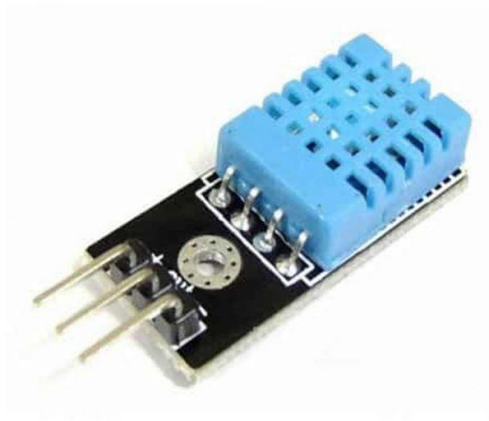
[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam\\_bien\\_do\\_am\\_dat.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam_bien_do_am_dat.ino)

## 8. Cảm biến đo độ ẩm đất và nhiệt độ 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.

Cảm biến độ ẩm nhiệt độ DHT11 ra chân được tích hợp sẵn điện trở 5,1k giúp người dùng dễ dàng kết nối và sử dụng hơn so với cảm biến DHT11 chưa ra chân, module lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp 1 dây). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không cần phải qua bất kỳ tính toán nào. Module được thiết kế hoạt động ở mức điện áp 5VDC.

Xin lưu ý chỉ sử dụng cảm biến độ ẩm nhiệt độ DHT11 trong môi trường độ ẩm thuần là hơi nước, các môi trường đặc biệt ủ kín như ủ tỏi đen, ủ yếm khí... sẽ sinh ra nấm và vi khuẩn bám lên bề mặt cảm biến làm hư hỏng cảm biến.



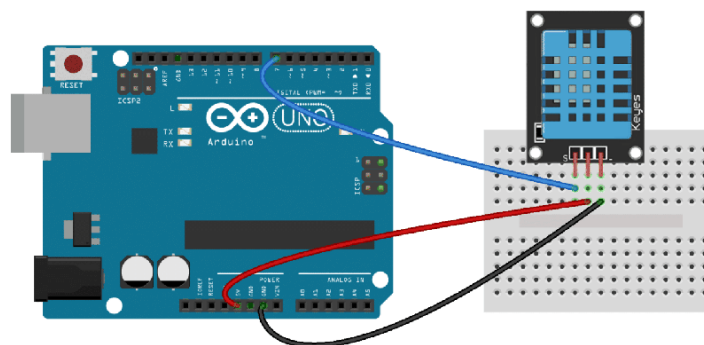
*Hình 3.15. Cảm biến nhiệt độ và độ ẩm DHT11*

### THÔNG SỐ KỸ THUẬT

- Điện áp hoạt động: 5VDC
- Chuẩn giao tiếp: TTL, 1 wire.
- Khoảng đo độ ẩm: 20%-80%RH sai số  $\pm 5\%$  RH
- Khoảng đo nhiệt độ: 0-50°C sai số  $\pm 2^{\circ}\text{C}$
- Tần số lấy mẫu tối đa 1Hz (1 giây / lần)
- Kích thước: 28mm x 12mm x 10mm

### SƠ ĐỒ CHÂN

DHT11	Arduino UNO R3
GND	5V
Vcc	GND
Signal	D2



*Hình 3.16. Sơ đồ nối dây cảm biến DHT11*

Để sử dụng DHT11 cần phải cài bổ sung thư viện DHT11 cho Arduino IDE:

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

**Code:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam\\_bien\\_do\\_am\\_nhiet\\_do\\_DHT11.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam_bien_do_am_nhiet_do_DHT11.ino)

Một lưu ý 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!

## **9. Cảm biến khí ga (MQ-2)**

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.

Cảm biến khí gas MQ-2 sử dụng phần tử SnO<sub>2</sub> có độ dẫn điện thấp hơn trong không khí sạch, khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của SnO<sub>2</sub> sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện.

Cảm biến khí gas MQ-2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan (CH<sub>4</sub>) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau.

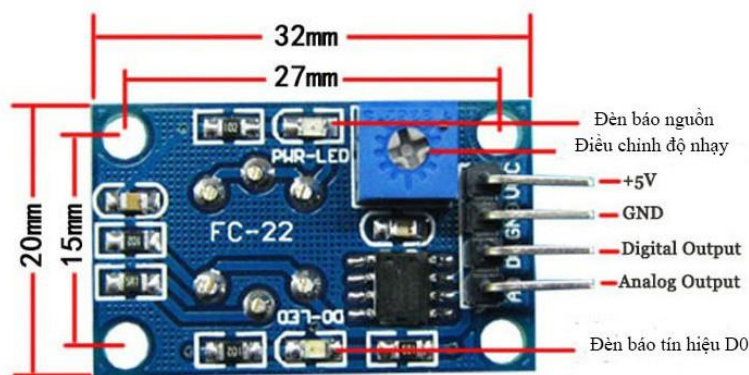
Cảm biến xuất ra cả hai dạng tín hiệu là Analog và Digital, tín hiệu Digital có thể điều chỉnh mức báo bằng biến trở.

### **THÔNG SỐ KỸ THUẬT**

- Nguồn hoạt động: 5V
- Loại dữ liệu: Analog
- Phạm vi phát hiện rộng
- Tốc độ phản hồi nhanh và độ nhạy cao
- Mạch đơn giản
- Ổn định khi sử dụng trong thời gian dài

### **SƠ ĐỒ MẠCH CẢM BIẾN KHÍ GAS MQ2:**





## SƠ ĐỒ CHÂN

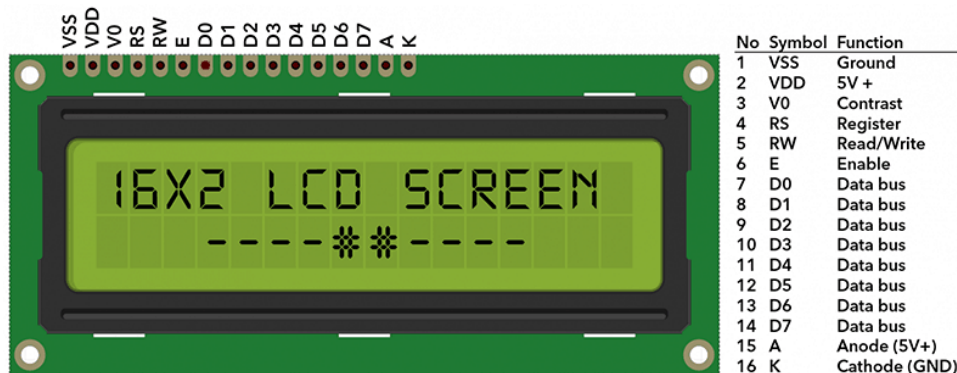
Arduino	MQ2
A0	Aout
VCC	5V
GND	GND

Code:

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam\\_bien\\_khi\\_ga.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Cam_bien_khi_ga.ino)

## II. 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.



Hình 3.17. Màn hình LCD 1602 Xanh Lá

## THÔNG SỐ KỸ THUẬT

- Điện áp hoạt động là 5V.
- Kích thước: 80 x 36 x 12.5mm
- Chữ trắng, nền xanh dương

- Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
- Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu
- Có bộ ký tự được xây dựng hỗ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet để biết thêm chi tiết.



**Hình 3.18. Mạch chuyển đổi giao tiếp I2C cho LCD**

### THÔNG SỐ MẠCH CHUYỂN ĐỔI GIAO TIẾP I2C

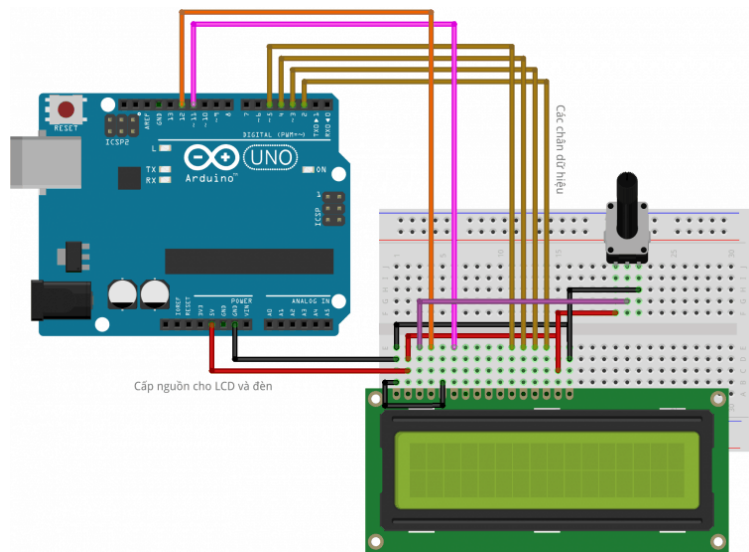
- Kích thước: 41.5mm(L)X19mm(W)X15.3MM(H)
- Trọng lượng: 5g
- Điện áp hoạt động: 2.5v-6v
- Jump chốt: Cung cấp đèn cho LCD hoặc ngắt
- Biến trở xoay độ tương phản cho LCD

**Lưu ý:** Các phiên bản cũ địa chỉ của bus i2c là 0X27, loại mới là 0x3F

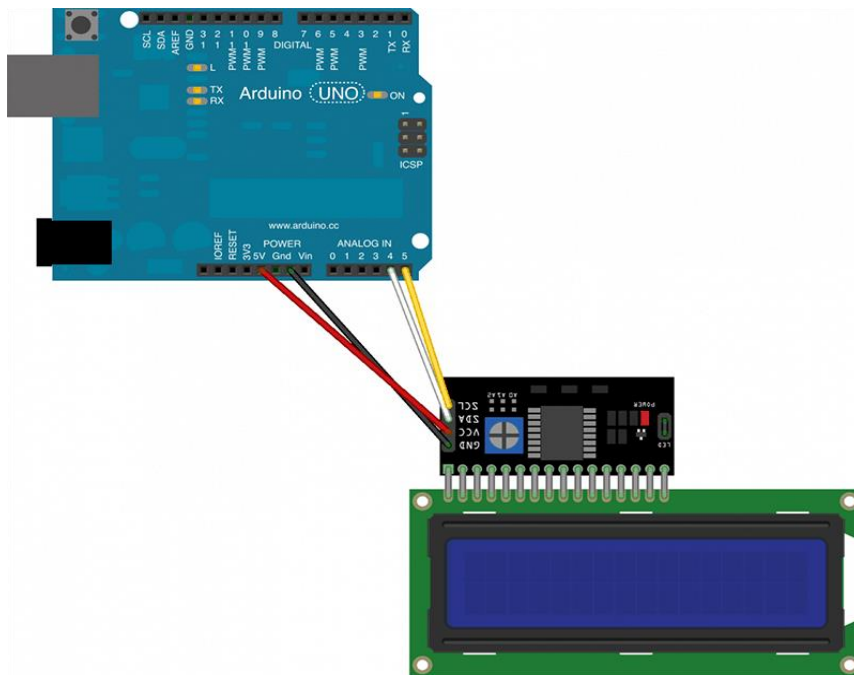
**Download và cài đặt thư viện hỗ trợ sử dụng màn hình LCD qua giao tiếp I2C:** <http://tdhshop.com.vn/tong-hop-cac-thu-vien-cho-arduino>

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!

### SƠ ĐỒ CHÂN



*Hình 3.19. Sơ đồ kết nối trực tiếp LCD*



*Hình 3.20. Sơ đồ kết nối LCD qua giao tiếp I2C*

**Kết nối:**

I2C	UNO R3	MEGA
VIN	5V	5V
GND	GND	GND
SCL	A5	SCL
SDA	A4	SDA

### **Code 1: CHƯƠNG TRÌNH HIỂN THỊ LCD TRỰC TIẾP**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/HienLCD\\_Ket\\_noi\\_truc\\_tiep.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/HienLCD_Ket_noi_truc_tiep.ino)

### **Code 2: CHƯƠNG TRÌNH HIỂN THỊ LCD QUA GIAO TIẾP I2C**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/HienLCD\\_I2C.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/HienLCD_I2C.ino)

### **Code 3: ĐỌC DỮ LIỆU TỪ DHT11 VÀ HIỂN THỊ TRÊN LCD**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Doc\\_Nhiệt\\_do\\_và\\_do\\_âm\\_lên\\_LCD.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Doc_Nhiệt_do_và_do_âm_lên_LCD.ino)

## **III. Điều khiển động cơ**

### **1. Động cơ servo**

Trong hướng dẫn này, chúng ta sẽ điều khiển một động cơ Servo bằng Arduino Uno. Động cơ Servo được sử dụng khi cần có sự di chuyển chính xác trục hoặc vị trí. Loại động cơ này không phù hợp cho các ứng dụng tốc độ cao. Nó chỉ phù hợp với các ứng dụng có tốc độ thấp, mô-men xoắn trung bình và đòi hỏi vị trí chính xác. Những động cơ này thường được sử dụng trong các cánh tay robot, điều khiển bay và hệ thống điều khiển.



**Hình 3.21. Động cơ Servo SG90**

Động cơ Servo là động cơ tốc độ thấp, mô-men xoắn cao với nhiều hình dạng và kích cỡ khác nhau. Không giống như động cơ DC và động cơ bước, động cơ Servo thường không quay tròn đủ 360 độ. Thay vào đó nó được giới hạn trong một phạm vi 180, 270 hoặc 90 độ.

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:

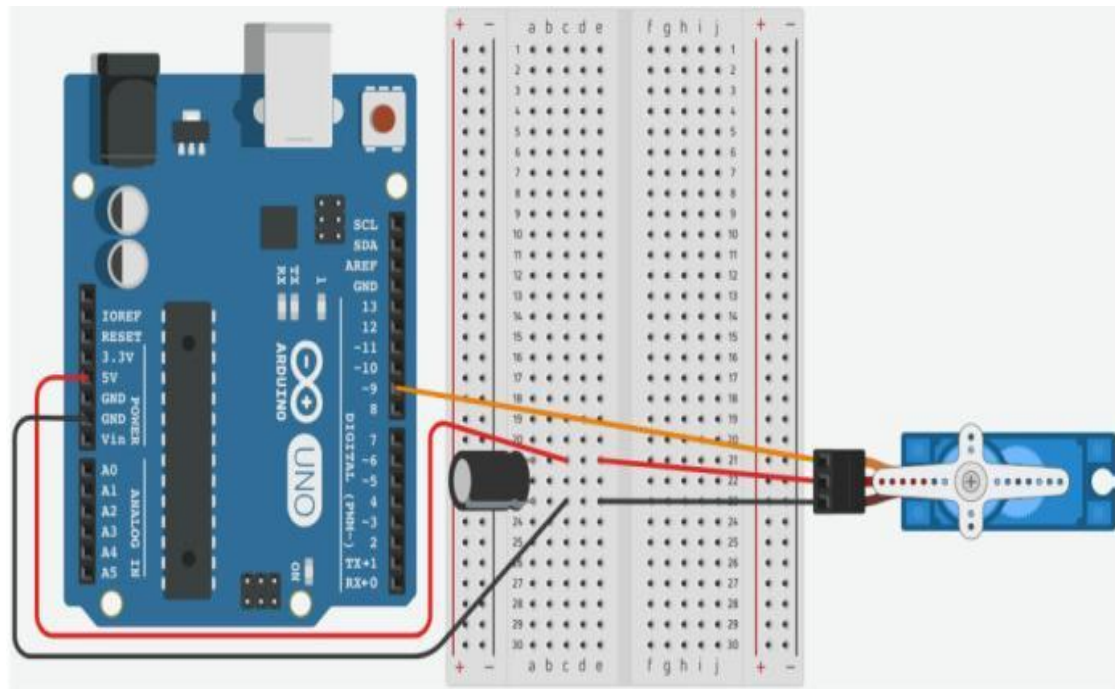
- **Đỏ:** nhận điện nguồn, tùy vào loại động cơ mà giá trị này có thể khác nhau
- **Đen/Nâu:** nối với cực âm của mạch
- **Vàng/Cam:** nhận tín hiệu từ mạch điều khiển

## THÔNG SỐ KỸ THUẬT

- Loại: SG90 – 360
- Kích thước: 23mmX12.5mmX29.5mm
- Trọng lượng: 9 gram
- Tốc độ không tải: 0,12 giây / 60 độ (4,8V)
- Mô-men xoắn chặn: 1,2-1,4 kg / cm (4,8V)
- Nhiệt độ hoạt động: -30 độ C ~ + 60 độ C
- Cài đặt vùng chết: 7 micro giây
- Điện áp làm việc: 4,8V-6V

## THÍ NGHIỆM 1

### SƠ ĐỒ CHÂN



*Hình 3.22. Sơ đồ chân điều khiển động cơ Servo SG90 theo góc*

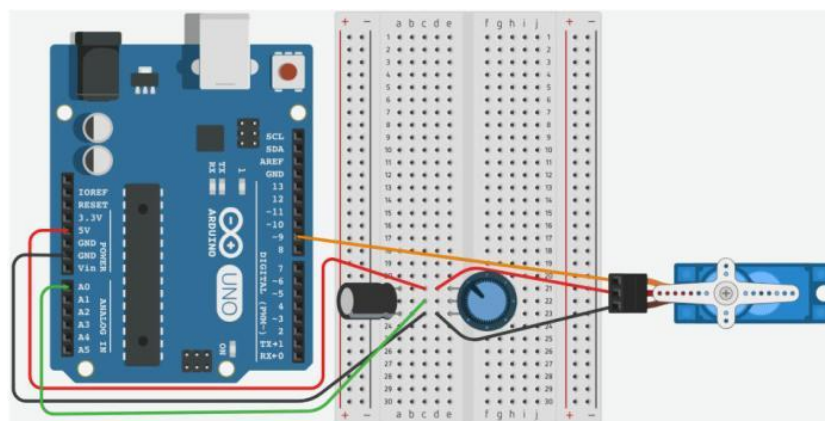
**Code: CHƯƠNG TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ SERVO THEO GÓC**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong\\_co\\_Servo\\_theo\\_goc.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong_co_Servo_theo_goc.ino)

## THÍ NGHIỆM 2

### SƠ ĐỒ CHÂN





*Hình 3.23. Sơ đồ chân điều khiển động cơ Servo SG90 theo giá trị biến trở*

**Code: CHƯƠNG TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ SERVO THEO GIÁ TRỊ BIẾN TRỞ**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong\\_co\\_Servo\\_theo\\_bien\\_tro.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong_co_Servo_theo_bien_tro.ino)

## 2. Động cơ bước (stepper motor)

Động cơ bước 28BYJ-48 + mạch đệm ULN2003 với ưu điểm nhỏ gọn, có sẵn mạch đệm cho động cơ, thích hợp dùng cho những ai mới làm quen với động cơ bước, từ đây sẽ hiểu rõ hơn về cấu tạo cũng như phương thức điều khiển của động cơ.



*Hình 3.24. Động cơ bước 28BYJ-48 + mạch đệm ULN2003*

**THÔNG SỐ KỸ THUẬT:**

**Mạch đệm ULN2003:**

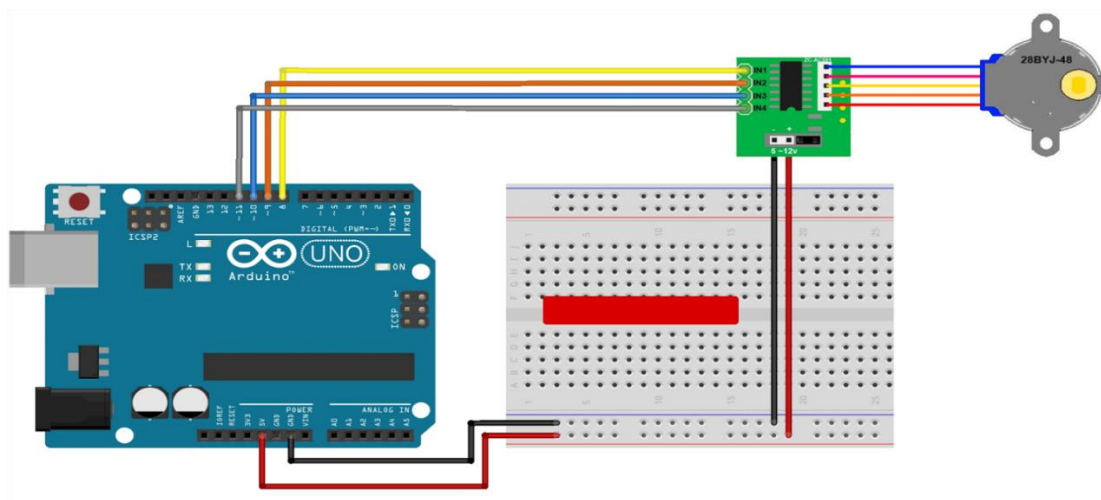
- Điện áp cung cấp: 5 ~ 12VDC.
- Tín hiệu ngõ vào: 4 chân in1, in2, in3, in4.
- Tín hiệu ngõ ra: Jack cắm động cơ bước 28BYJ-48.
- 4 led hiển thị trạng thái hoạt động của động cơ.

### Động cơ 28BYJ-48:

- Điện áp cung cấp: 5VDC.
- Số phase: 4.
- Bước: 5.6250/64

Động cơ bước 28BYJ-48 là động cơ bước 4 pha (thực ra là 2 pha được chia ra làm 2 ở mỗi pha ngay tại vị trí giữa) (gồm 5 dây), 4 trong 5 dây này được kết nối với 2 cuộn dây trong động cơ và 1 dây là dây nguồn chung cho cả 2 cuộn dây. Mỗi bước của động cơ quét 1 góc 5.625 độ, vậy để quay 1 vòng động cơ phải thực hiện 64 bước.

### SƠ ĐỒ CHÂN:



ULN2003 Driver Board	Arduino
IN1	Pin 8
IN2	Pin 9
IN3	Pin 10
IN4	Pin 11
—	GND power supply
+	5 V power supply

**Hình 3.25. Sơ đồ chân điều khiển động cơ bước 28BYJ-48 + mạch đệm ULN2003**

Arduino IDE có một thư viện hỗ trợ điều khiển động cơ bước tên là Stepper Motor, nó cực kì dễ sử dụng, chỉ việc kết nối như trên rồi nạp code, thế là xong! Nhưng mọi việc không dễ dàng như thế, chúng ta cần phải quan tâm một số điều sau:



Động cơ **28BYJ-48** chỉ có sẵn bộ hộp số và nó cho ta đến \*64 (tỉ lệ bánh răng) => nó có đến  $64 * 64 = 4096$  bước. Quá đã phải không nào!

### **Công thức tính số bước**

- Số bước thật = Số bước lý thuyết \* tỉ lệ bánh răng
- Trong trường hợp này: Số bước thật =  $64 * 64 = 4096$ .

### **Code 1: CHƯƠNG TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong\\_co\\_buoc1.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong_co_buoc1.ino)

### **Code 2: CHƯƠNG TRÌNH ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC BẰNG JOYSTICK**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong\\_co\\_buoc\\_theo\\_joystick.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong_co_buoc_theo_joystick.ino)

## **3. Động cơ DC**

### **Giới thiệu về mạch cầu H L298N**

Trên thị trường hiện có nhiều loại mạch cầu H, tuy nhiên loại thông dụng và thích hợp với board Arduino là mạch cầu H L298N. Loại mạch cầu H này có thể điều khiển được 2 động cơ.

### **THÔNG SỐ KỸ THUẬT**

- IC chính: L298 – Dual Full Bridge Driver
- Điện áp đầu vào: 5~30VDC
- 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
- Mức điện áp logic: Low -0.3V~1.5V, High: 2.3V~Vss
- Kích thước: 43x43x27mm

Module này gồm có IC L298N, 1 khối điều khiển động cơ A, 1 khối điều khiển động cơ B, các chân cấp nguồn, 2 jump.

- **Khối điều khiển động cơ A:** Khối này có 2 chân nhằm để kết nối với động cơ A
- **Khối điều khiển động cơ B:** Khối này cũng có 2 chân nhằm để kết nối với động cơ B
- **Các chân cấp nguồn:** Chân cấp nguồn gồm 2 chân, chân GND nối với nguồn âm còn chân +12V nối với nguồn dương của acquy hoặc adapter 12V. Chân +5V dùng khi cần sử dụng nguồn 5V từ mạch cầu H ra để dùng. Thường thì chân này ít sử dụng.



**Code 2: CHƯƠNG TRÌNH VÍ DU VỀ SỬ DỤNG MẠCH CẦU H - MẠCH CẦU L298N, 2 ĐỘNG CƠ**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong\\_co\\_DC\\_qua\\_mach\\_cau\\_L298N\\_2DC.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Dong_co_DC_qua_mach_cau_L298N_2DC.ino)



- Vào google Play gõ tìm kiếm APP: "**Arduino Bluetooth RC Car**" để điều khiển xe qua Bluetooth.
- Vào trang <http://ai2.appinventor.mit.edu/> để tự tạo app điều khiển.

### **Code tham khảo:**

#### **Code 1:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE\\_ROBOT\\_Bluetooth-TranhVatCan.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE_ROBOT_Bluetooth-TranhVatCan.ino)

#### **Code 2:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE\\_ROBOT\\_HongNgoai-Bluetooth.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE_ROBOT_HongNgoai-Bluetooth.ino)

#### **Code 3:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE\\_ROBOT\\_HongNgoai-Do-Line.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE_ROBOT_HongNgoai-Do-Line.ino)

## **2. Điều khiển qua tay cầm PS2**

Remote điều khiển PS2 Không Dây ( PS2 Robot Controller ) là bộ Remote điều khiển robot, xe thông minh, điều khiển từ xa các thiết bị phong phú. Giao tiếp với vi điều khiển ngoại vi khác, dễ sử dụng.

Tay cầm điều khiển PS2 sử dụng điện áp 3.3VDC cho cấp nguồn và giao tiếp GPIO, nếu các bạn giao tiếp với các vi điều khiển 5VDC thì cần thêm một bộ chuyển đổi Logic để tránh trường hợp Receiver của tay bị cháy.

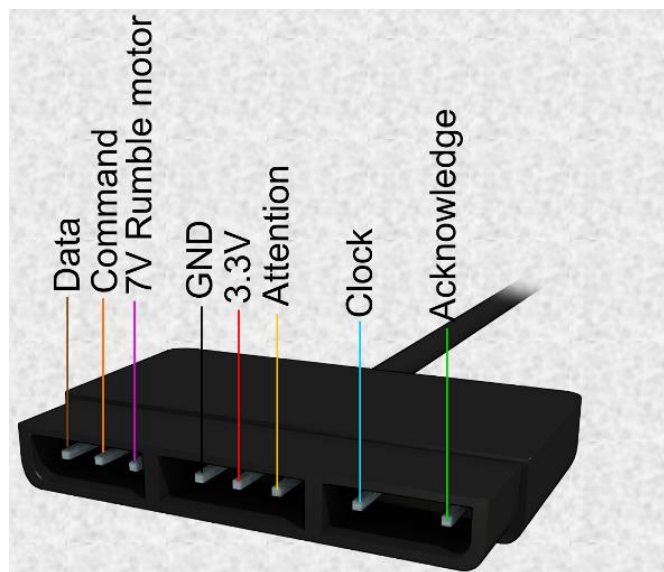
Đối với điều khiển robot, chỉ cần dùng các dây sau: Clock, Data, Command, VCC & GND, Attention. Clock, Data, Command, Attention nối với các chân I/O bất kỳ. Chân Data nên được kéo nguồn bằng điện trở từ 1k-10k. Clock: xung, đồng bộ hóa quá trình truyền dữ liệu. Data: dữ liệu từ gamepad về vi điều khiển; Command: dữ liệu từ vi điều khiển đến gamepad. Attention: Chip select VCC: 3-5V; GND:0V

- Điện áp bộ thu: 3.3VDC
- Tay cầm PS2, vật liệu ABS chất lượng cao.
- Công nghệ nhận không dây 2.4GHz, khoảng cách có thể lên đến 10m.
- Chế độ hoạt động kỹ thuật số / analog, có thể được chuyển bất cứ lúc nào, có đèn báo
- 4-trục 12-nút thiết kế để đáp ứng nhu cầu của các nút trò chơi (chế độ kỹ thuật số: 2-trục 10 nút; chế độ tương tự: 4-trục 12 nút).
- Nút có chức năng chụp liên tục và thao tác trò chơi thuận tiện hơn.
- Hai joystick analog có độ chính xác cao để điều khiển 360 độ

- Các phím trái và phải có mỗi động cơ rung tích hợp có thể thể hiện hiệu ứng rung động tinh tế.
- Kết nối nhanh, phản ứng nhanh, không chậm trễ.
- Hỗ trợ USB1.1 / 2.0 (yêu cầu bổ sung USB để PS2 chuyển đổi)
- Hỗ trợ WIN 98 / ME / 2000 / XP / Vista
- Nguồn: 2 pin AAA

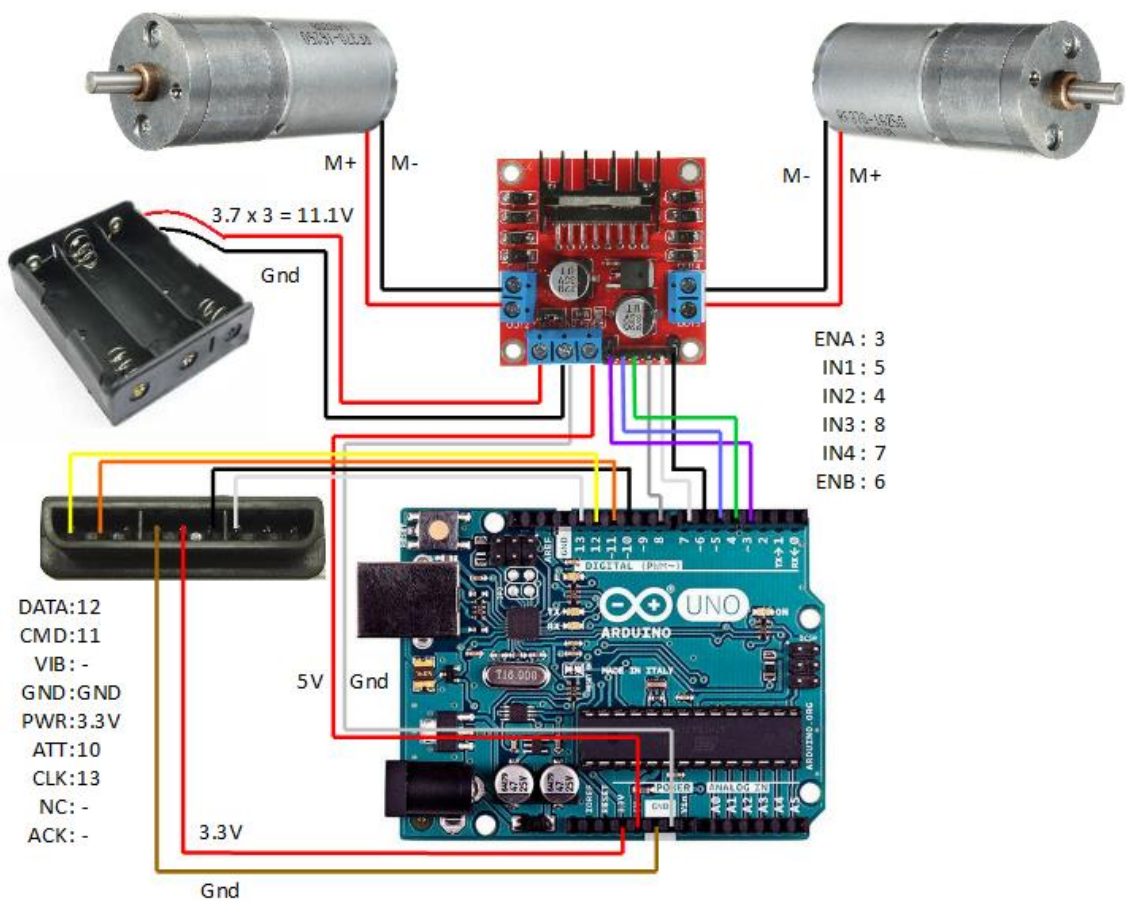


**Hình 4.2. Sơ đồ và mã các nút bấm trên PS2**



**Hình 4.3. Sơ đồ chân trên Receiver**





**Hình 4.4. Sơ đồ chân kết nối Receiver với Arduino**

**Code tham khảo:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE\\_ROBOT\\_Su\\_dung\\_tay\\_game\\_PS2.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/XE_ROBOT_Su_dung_tay_game_PS2.ino)

## **II. Dự án 2: Vườn thông minh**

### **1. Nguyên lý hoạt động**

- Dự án sử dụng board ESP8266 NodeMCU để lấy thông tin lên server thông qua Wifi.
- Cảm biến nhiệt độ, độ ẩm (DHT) thu thập nhiệt độ thông số môi trường sau đó cập nhật lên màn hình LCD và đồng thời cập nhật lên app.
- Cảm biến độ ẩm đất thu thập giá trị của độ ẩm đất hiện tại sau đó cập nhật lên màn hình LCD và đồng thời cập nhật trên app.
- Các thiết bị điều khiển: đèn, quạt, bơm
- Điều khiển chế độ tự động(auto) hoặc bằng tay (manual) thông qua nút nhấn lựa chọn ở trên app(đồng thời hiển thị giá trị tương ứng trên màn hình LCD)



- Chế độ bằng tay(manual): điều khiển các thiết bị thông qua nút nhấn trên app(nếu nhấn on sẽ kích hoạt relay và thiết bị sẽ được hoạt động, ngược lại nếu nhấn off thì thiết bị sẽ ngưng hoạt động)
- Chế độ tự động(auto): cài đặt giá trị độ ẩm đất chuẩn thông qua app và hiển thị trên màn hình LCD, nếu độ ẩm đất thực tế nhỏ hơn độ ẩm đất cài đặt thì bơm sẽ kích hoạt và ngược lại

## **2. Sử dụng app**

- Tải app blynk (version cũ)
- Tạo tài khoản và đăng nhập.
- Thực hiện phát wifi cho mạch (từ trên điện thoại)

### **Code tham khảo:**

[https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Vuon\\_thong\\_minh.ino](https://github.com/LQ-Computer/IoT-for-STEM/blob/main/Vuon_thong_minh.ino)

## TÀI LIỆU THAM KHẢO

- [1]. [Tài liệu "Thiết kế và lập trình Internet of Things: Từ A-Z" - Chương 1 \(vnezlink.blogspot.com\)](http://vnezlink.blogspot.com)
- [2]. “*Tự học nhanh Arduino cho người mới bắt đầu*”, Ths. Huỳnh Minh Phú.
- [3]. “*Arduino cho người mới bắt đầu*”, IoT Maker Viet Nam.
- [4]. “*Lập trình Arduino cho người mới bắt đầu - V2*”
- [5]. <https://nshopvn.com/>
- [6]. <http://arduino.vn/>

## MỤC LỤC

MỞ ĐẦU.....	5
I.    Khái niệm “Internet of Things” .....	5
II.   Internet of Things đang thay đổi cách sống trên thế giới như thế nào? .....	6
CHƯƠNG I: ARDUINO LÀ GÌ? .....	8
I.    Khái niệm, lịch sử hình thành và phát triển.....	8
II.   Những board mạch Arduino thông dụng trên thị trường .....	8
1.  Arduino Uno R3 .....	9
2.  Arduino Nano .....	10
3.  Arduino Leonardo .....	11
5.  ESP8266 WiFi Uno.....	12
6.  ESP8266 NodeMCU .....	12
CHƯƠNG II: GIỚI THIỆU ARDUINO IDE.....	14
I.    Khái niệm.....	14
II.   Cài đặt.....	14
1.  Dùng Online IDE .....	14
2.  Dùng Offline .....	16
III.  Sử dụng Arduino IDE .....	19
1.  Khởi động.....	19
2.  Cấu trúc chương trình .....	21
3.  Một số lệnh điều khiển cơ bản .....	22
4.  Hàm và thủ tục .....	23
IV.  Cài đặt thư viện .....	24
1.  Cài đặt thư viện Arduino bằng Arduino Library Manager .....	24
2.  Cài đặt thư viện Arduino từ các trang web khác.....	26
CHƯƠNG III: THỰC CHIẾN LẬP TRÌNH ĐIỀU KHIỂN CÁC MODULE NGOẠI VI.....	28
I.    Module cảm biến .....	28
1.  Cảm biến ánh sáng dùng quang trở.....	28
2.  Cảm biến phát hiện lửa (Flame Sensor - báo cháy) .....	30

3. Cảm biến dò line .....	32
4. Cảm biến mưa .....	33
5. Cảm biến siêu âm .....	35
6. Cảm biến âm thanh.....	36
7. Cảm biến độ ẩm đất.....	37
8. Cảm biến đo độ ẩm đất và nhiệt độ DHT11 .....	38
9. Cảm biến khí ga (MQ-2) .....	40
II. Màn hình LCD.....	41
III. Điều khiển động cơ .....	44
1. Động cơ servo .....	44
2. Động cơ bước (stepper motor) .....	46
3. Động cơ DC .....	48
CHƯƠNG IV: MỘT VÀ DỰ ÁN THAM KHẢO .....	51
I. Dự án 1: Xe robot .....	51
1. Điều khiển qua app điện thoại.....	51
2. Điều khiển qua tay cầm PS2 .....	52
II. Dự án 2: Vườn thông minh.....	54
1. Nguyên lý hoạt động .....	54
2. Sử dụng app.....	55
TÀI LIỆU THAM KHẢO.....	56