

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN



BÁO CÁO ĐỒ ÁN 1

Đề tài: Tìm hiểu về bo mạch Arduino

Giảng viên hướng dẫn: TS. Nguyễn Hoàng Nam

Sinh viên thực hiện:

Họ tên	MSSV	Lớp	Khóa
Nguyễn Đình Tĩnh	20093643	ĐT VT09	54
Phạm Văn Tôn	20122583	ĐK-TĐH01	57
Lê Huy Tiệp	20122560	ĐK-TĐH01	57
Nguyễn Văn Tuấn	20122705	ĐK-TĐH02	57

Hà Nội, 11-2013

Mục lục

Mở đầu	3
Chương 1. Tổng quan về bo mạch Arduino	4
1.1. Giới thiệu chung về bo mạch Arduino	4
1.2. Một số ứng dụng nổi bật của bo mạch.	6
1.3. Khả năng của bo mạch Arduino.....	8
Chương 2. Các loại bo mạch Arduino thông dụng	14
2.1. Bo mạch Arduino Uno.....	14
2.2. Bo mạch Arduino Ethernet.....	20
2.3. Bo mạch Arduino Yun.....	26
2.4. Bo mạch Arduino Nano.....	34
2.5. Bo mạch Arduino Mega ADK.....	38
Chương 3. Cơ bản về lập trình Arduino	45
3.1. Ngôn ngữ lập trình trên Arduino.....	45
3.2. Điều kiện cần thiết để lập trình.	46
3.3. Cấu trúc của một chương trình lập trình.	47
3.4. Cài đặt Driver và phần mềm lập trình Arduino trên Windows.....	48
3.5. Arduino Fritzing.....	57
Kết luận	66
Danh mục tài liệu tham khảo	67
Phụ lục	68

Mở đầu

Kỹ thuật lập trình phần cứng ngày nay có sự phát triển mạnh mẽ, nó được ứng dụng vào hầu hết các lĩnh vực như sản xuất công nghiệp, tự động hóa và còn rất nhiều lĩnh vực khác nữa. So với kỹ thuật lập trình phần mềm thì kỹ thuật lập trình phần cứng có những ứng dụng riêng và được lập trình để điều khiển các hệ thống mang lại lợi ích đáng kể cho người sử dụng. Với tính ưu việt của kỹ thuật lập trình phần cứng thì nó hứa hẹn mang lại hiệu quả công việc rất cao. Chúng ta biết rằng lập trình phần cứng thực hiện lập trình trên các dòng vi xử lý khác nhau. Trong quá trình này, ta sử dụng ngôn ngữ lập trình phần cứng để thực hiện các thao tác liên kết điều khiển giữa vi xử lý với các thành phần trong bo mạch và các thiết bị ngoại vi. Vì vậy nó có sự ứng dụng rất rộng rãi trong tất cả các ngành. Nhưng trong khuôn khổ của đề tài này chúng em chỉ đi sâu vào một hướng cụ thể đó là “*tìm hiểu về bo mạch Arduino*”. Chúng em thực hiện đề tài nhằm mục đích có được những cái nhìn tổng quan về các dòng bo mạch Arduino thông dụng hiện nay, đồng thời tìm hiểu về cấu tạo và các chức năng cơ bản của từng bo mạch này. Để hoàn thành tốt đề tài này, các thành viên đã có sự tích lũy kiến thức, tìm hiểu về vi điều khiển và các loại linh kiện được sử dụng trong mạch. Đồng thời rèn luyện khả năng tư duy độc lập, liên kết ý tưởng và kỹ năng làm việc nhóm của các thành viên. Với những mục đích trên, đã tạo động lực cho chúng em hoàn thành đề tài này.

Em xin chân thành cảm ơn thầy Nguyễn Hoàng Nam – Giảng viên hướng dẫn môn “Đồ án 1” đã cung cấp những kiến thức hữu ích và định hướng nghiên cứu để giúp chúng em hoàn thành tốt đề tài này.

Đây là bài tập lớn lần đầu tiên mà nhóm chúng em chung sức làm việc cùng nhau, mọi người đã thật sự cố gắng để hoàn thành tốt nhiệm vụ của mình. Nhưng trong quá trình thực hiện không thể tránh khỏi những thiếu sót. Vì vậy, chúng em rất mong nhận được sự đóng góp ý kiến từ Thầy và các bạn, để có thể hoàn thiện hơn.

Em xin chân thành cảm ơn !!!

TỔNG QUAN VỀ BỘ MẠCH ARDUINO

1.1. Giới thiệu chung về bo mạch Arduino

Arduino đã và đang được sử dụng rất rộng rãi trên thế giới, và ngày càng chứng tỏ được sức mạnh của chúng thông qua vô số ứng dụng độc đáo của người dùng trong cộng đồng nguồn mở (open-source). Tuy nhiên tại Việt Nam Arduino vẫn còn chưa được biết đến nhiều. Nội dung của phân này nhằm giới thiệu một số thông tin về Arduino với hy vọng cung cấp cho người dùng DIY thêm một lựa chọn mới đầy tiềm năng để thực hiện các dự án của mình.



Hình 1. Hình ảnh mô tả kích thước nhỏ gọn của bo mạch Arduino

Từ khi xuất hiện trong cộng đồng mã nguồn mở và lập trình phần cứng, Arduino thực sự đã gây sóng gió trên thị trường người dùng DIY (là những người tự chế ra sản phẩm của mình) trên toàn thế giới trong vài năm gần đây, gần giống với những gì Apple đã làm được trên thị trường thiết bị di động. Số lượng người dùng cực

lớn và đa dạng với trình độ trải rộng từ bậc phổ thông lên đến đại học đã làm cho ngay cả những người tạo ra chúng phải ngạc nhiên về mức độ phổ biến.



Hình 2. Những thành viên khởi xướng Arduino

Vậy, Arduino là gì mà có thể khiến ngay cả những sinh viên và nhà nghiên cứu tại các trường đại học danh tiếng như MIT, Stanford, Carnegie Mellon phải sử dụng; hoặc ngay cả Google cũng muốn hỗ trợ khi cho ra đời bộ kit Arduino Mega ADK dùng để phát triển các ứng dụng Android tương tác với cảm biến và các thiết bị khác?

Thật vậy, Arduino là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm. Chỉ với khoảng \$30, người dùng đã có thể sở hữu một bo Arduino có 20 ngõ I/O có thể tương tác và điều khiển chùng ấy thiết bị.

Arduino ra đời tại thị trấn Ivrea thuộc nước Ý và được đặt theo tên một vị vua vào thế kỷ thứ 9 là King Arduin. Arduino chính thức được đưa ra giới thiệu vào năm 2005 như là một công cụ khiêm tốn dành cho các sinh viên của giáo sư Massimo Banzi, là một trong những người phát triển Arduino, tại trường Interaction Design Institute Ivrea (IDII). Mặc dù hầu như không được tiếp thị gì cả, tin tức về Arduino vẫn lan truyền với tốc độ chóng mặt nhờ những lời truyền miệng tốt đẹp của những

người dùng đầu tiên. Hiện nay Arduino nổi tiếng tới nỗi có người tìm đến thị trấn Ivrea chỉ để tham quan nơi đã sản sinh ra Arduino.

1.2. Một số ứng dụng nổi bật của bo mạch.

Arduino được chọn làm bộ não xử lý của rất nhiều thiết bị từ đơn giản đến phức tạp. Trong số đó có một vài ứng dụng thực sự chứng tỏ khả năng vượt trội của Arduino do chúng có khả năng thực hiện nhiều nhiệm vụ rất phức tạp. Sau đây là danh sách một số ứng dụng nổi bật của Arduino.

➤ *Máy in 3D.*

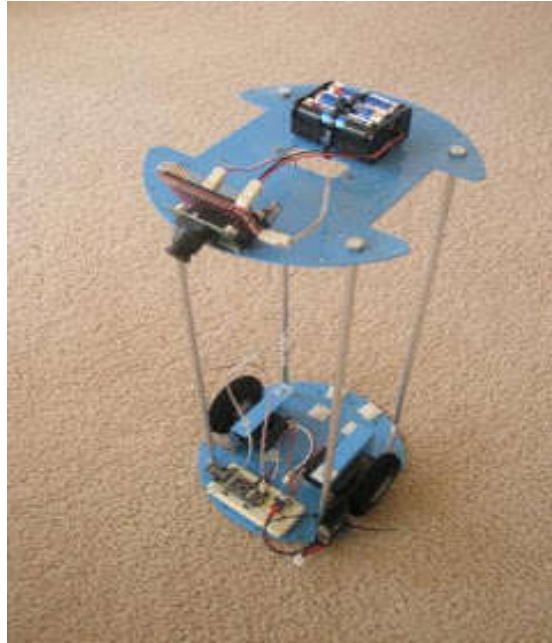
Một cuộc cách mạng khác cũng đang âm thầm định hình nhờ vào Arduino, đó là sự phát triển máy in 3D nguồn mở Reprap. Máy in 3D là công cụ giúp tạo ra các vật thể thực trực tiếp từ các file CAD 3D. Công nghệ này hứa hẹn nhiều ứng dụng rất thú vị trong đó có cách mạng hóa việc sản xuất cá nhân.



Hình 3. Máy in 3D Makerbot điều khiển bằng Arduino Mega2560

➤ *Robot.*

Do kích thước nhỏ gọn và khả năng xử lý mạnh mẽ, Arduino được chọn làm bộ xử lý trung tâm của rất nhiều loại robot, đặc biệt là robot di động.



Hình 4. Robot di động tránh vật cản dùng Arduino nano và camera CMUCam

➤ *Thiết bị bay không người lái UAV.*

UAV là một ứng dụng đặc biệt thích hợp với Arduino do chúng có khả năng xử lý nhiều loại cảm biến như Gyro, accelerometer, GPS... điều khiển động cơ servo và cả khả năng truyền tín hiệu từ xa.



Hình 5. Một thiết bị UAV

➤ *Game tương tác.*

Việc đọc cảm biến và tương tác với PC là một nhiệm vụ rất đơn giản đối với Arduino. Do đó rất nhiều ứng dụng game tương tác có sử dụng Arduino.

➤ *Điều khiển ánh sáng.*

Các tác vụ điều khiển đơn giản như đóng ngắt đèn LED hay phức tạp như điều khiển ánh sáng theo nhạc hoặc tương tác với ánh sáng laser đều có thể thực hiện với Arduino.



Hình 6. Trình diễn công nghệ Ambilight với Arduino

➤ *Kích hoạt chụp ảnh tốc độ cao*

Đây là một ứng dụng rất đơn giản nhưng đặc biệt hữu ích với những ai đam mê chụp ảnh. Ứng dụng này giúp tạo ra những bức ảnh độc đáo ghi lại những khoảnh khắc xảy ra cực nhanh mà nếu không có dụng cụ hỗ trợ chúng ta khó lòng ghi lại.

Trên đây chỉ là một vài ví dụ minh họa cho khả năng ứng dụng của Arduino. Khi tìm kiếm trên Google, bạn có thể tìm thấy vô số ứng dụng có sử dụng Arduino. Ngoài ra có thể tham khảo trên các trang web để tìm hiểu thêm nhiều ứng dụng rất độc đáo.

1.3. Khả năng của bo mạch Arduino.

Bo mạch Arduino thường sử dụng dòng vi xử lý 8-bit megaAVR của Atmel với hai chip phổ biến nhất là ATmega328 và ATmega2560. Các dòng vi xử lý này cho phép lập trình các ứng dụng điều khiển phức tạp do được trang bị cấu hình mạnh với các loại bộ nhớ ROM, RAM và Flash, các ngõ vào ra digital I/O trong đó có nhiều ngõ có khả năng xuất tín hiệu PWM, các ngõ đọc tín hiệu analog và các chuẩn giao tiếp đa dạng như UART, SPI, TWI (I2C).

➤ *Sức mạnh xử lý.*

Xung nhịp: 16MHz.

EEPROM: 1KB (ATmega328) và 4KB (ATmega2560).

SRAM: 2KB (Atmega328) và 8KB (Atmega2560).

Flash: 32KB (Atmega328) và 256KB (Atmega2560).

➤ *Đọc tín hiệu cảm biến ngõ vào:*

Digital: Các bo mạch Arduino đều có các cổng digital có thể cấu hình làm ngõ vào hoặc ngõ ra bằng phần mềm. Do đó người dùng có thể linh hoạt quyết định số lượng ngõ vào và ngõ ra.

Tổng số lượng cổng digital trên các mạch dùng Atmega328 là 14, và trên Atmega2560 là 54.

Analog: Các bo mạch Arduino đều có trang bị các ngõ vào analog với độ phân giải 10-bit (1024 phân mức, ví dụ với điện áp chuẩn là 5V thì độ phân giải khoảng 0.5mV). Số lượng cổng vào analog là 6 đối với Atmega328, và 16 đối với Atmega2560.

Với tính năng đọc analog, người dùng có thể đọc nhiều loại cảm biến như nhiệt độ, áp suất, độ ẩm, ánh sáng, gyro, accelerometer...

➤ *Xuất tín hiệu điều khiển ngõ ra:*

Digital output: Tương tự như các cổng vào digital, người dùng có thể cấu hình trên phần mềm để quyết định dùng ngõ digital nào là ngõ ra. Tổng số lượng cổng digital trên các mạch dùng Atmega328 là 14, và trên Atmega2560 là 54.

PWM output: Trong số các cổng digital, người dùng có thể chọn một số cổng dùng để xuất tín hiệu điều chế xung PWM. Độ phân giải của các tín hiệu PWM này là 8-bit.

Số lượng cổng PWM đối với các bo dùng Atmega328 là 6, và đối với các bo dùng Atmega2560 là 14.

PWM có nhiều ứng dụng trong viễn thông, xử lý âm thanh hoặc điều khiển động cơ mà phổ biến nhất là động cơ servos trong các máy bay mô hình.

➤ *Chuẩn Giao tiếp:*

Serial:

Đây là chuẩn giao tiếp nối tiếp được dùng rất phổ biến trên các bo mạch Arduino. Mỗi bo có trang bị một số cổng Serial cứng (việc giao tiếp do phần cứng trong chip thực hiện). Bên cạnh đó, tất cả các cổng digital còn lại đều có thể thực hiện giao tiếp nối tiếp bằng phần mềm (có thư viện chuẩn, người dùng không cần

phải viết code). Mức tín hiệu của các cổng này là TTL 5V. Lưu ý cổng nối tiếp RS-232 trên các thiết bị hoặc PC có mức tín hiệu là UART 12V. Để giao tiếp được giữa hai mức tín hiệu, cần phải có bộ chuyển mức, ví dụ như chip MAX232.

Số lượng cổng Serial cứng của Atmega328 là 1 và của Atmega2560 là 4. Với tính năng giao tiếp nối tiếp, các bo Arduino có thể giao tiếp được với rất nhiều thiết bị như PC, touchscreen, các game console...

USB: Các bo Arduino tiêu chuẩn đều có trang bị một cổng USB để thực hiện kết nối với máy tính dùng cho việc tải chương trình. Tuy nhiên các chip AVR không có cổng USB, do đó các bo Arduino phải trang bị thêm phần chuyển đổi từ USB thành tín hiệu UART. Do đó máy tính nhận diện cổng USB này là cổng COM chứ không phải là cổng USB tiêu chuẩn.

SPI: Đây là một chuẩn giao tiếp nối tiếp đồng bộ có bus gồm có 4 dây. Với tính năng này các bo Arduino có thể kết nối với các thiết bị như LCD, bộ điều khiển video game, bộ điều khiển cảm biến các loại, đọc thẻ nhớ SD và MMC...

I2C (TWI): Đây là một chuẩn giao tiếp đồng bộ khác nhưng bus chỉ có hai dây. Với tính năng này, các bo Arduino có thể giao tiếp với một số loại cảm biến như thermostat của CPU, tốc độ quạt, một số màn hình OLED/LCD, đọc real-time clock, chỉnh âm lượng cho một số loại loa...

➤ *Môi trường lập trình bo mạch Arduino.*

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.



Hình 7. Giao diện IDE của Arduino

Môi trường lập trình Arduino IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh OSX và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR C vào chương trình nếu muốn.

➤ Các loại bo mạch Arduino.

Về mặt chức năng, các bo mạch Arduino được chia thành hai loại: loại bo mạch chính có chip Atmega và loại mở rộng thêm chức năng cho bo mạch chính (thường được gọi là shield).

Các bo mạch chính về cơ bản là giống nhau về chức năng, tuy nhiên về mặt cấu hình như số lượng I/O, dung lượng bộ nhớ, hay kích thước có sự khác nhau. Một số bo có trang bị thêm các tính năng kết nối như Ethernet và Bluetooth.

Các bo mở rộng chủ yếu mở rộng thêm một số tính năng cho bo mạch chính ví dụ như tính năng kết nối Ethernet, Wireless, điều khiển động cơ v.v...

Dưới đây là thông số kỹ thuật của một số loại bo mạch Arduino thông dụng và có các tính năng xử lý mạnh mẽ.

Bảng 1. So sánh đặc điểm của một số loại bo mạch Arduino.

Loại bo Mạch Thông số kỹ thuật	Arduino Uno	Arduino Ethernet	Arduino Yun	Arduino Nano	Arduino Mega ADK
Chíp xử lý	ATmega328	ATmega328	ATmega32u4	ATmega168	ATmega2560
Điện áp hoạt động	5V	5V	5V	5V	5V
Điện áp đầu vào	7-12V	7-12V	5V	7-12V	7-12V
Điện áp giới hạn đầu vào	6-20V	6-20V	5-12V	6-20V	6-20V
Chân I/O digital	14 (6 đầu ra PWM)	14 (4 đầu ra PWM)	20 (7 đầu ra PWM)	14 (6 đầu ra PWM)	54 (15 đầu ra PWM)
Số chân analog đầu ra	6	6	12	8	16
Dòng DC trên chân I/O	40 mA	40 mA	40 mA	40mA	40 mA
Dòng DC trên chân 3.3V.	50 mA	50 mA	50 mA	Không có	50 mA
Flash Memory	32 KB (0.5 KB bootloader)	32 KB (0.5 KB bootloader)	32 KB (4 KB bootloader)	16 KB (2KB bootloader)	256 KB (8KB bootloader).
SRAM	2 KB	2 KB	2.5 KB	1 KB	8 KB
EEPROM	1 KB	1 KB	1 KB	512Bytes	4KB
Tốc độ xung	16 MHz	16 MHz	16 MHz	16 MHz	16 MHz
Giá trị trường	605.000đ	935.000đ	52€	34.99\$	1.540.000đ
Địa chỉ mua	Sotatec.com	Sotatec.com.	Store.arduin	gravitech.us	Sotatec.com.

	.vn	vn	o.cc		vn
Giá rẻ nhất có thể mua	310.000đ	280 000đ	Sản phẩm chưa phân phối tại Việt Nam	240.000đ	420.000đ
Địa chỉ mua	Machtudong.vn	kme.com.vn		dieukhientudonghoa.com	Thegioiic.com

Từ bảng so sánh trên, nhận thấy một điều là các sản phẩm Arduino trên thị trường là rất đa dạng. Do đây là những loại bo mạch mã nguồn mở nên giá cả trên thị trường cũng rất biến động. Sinh viên thực hiện lập trình trên Arduino nên tìm nguồn sản phẩm chất lượng uy tín và mua với giá rẻ nhất.

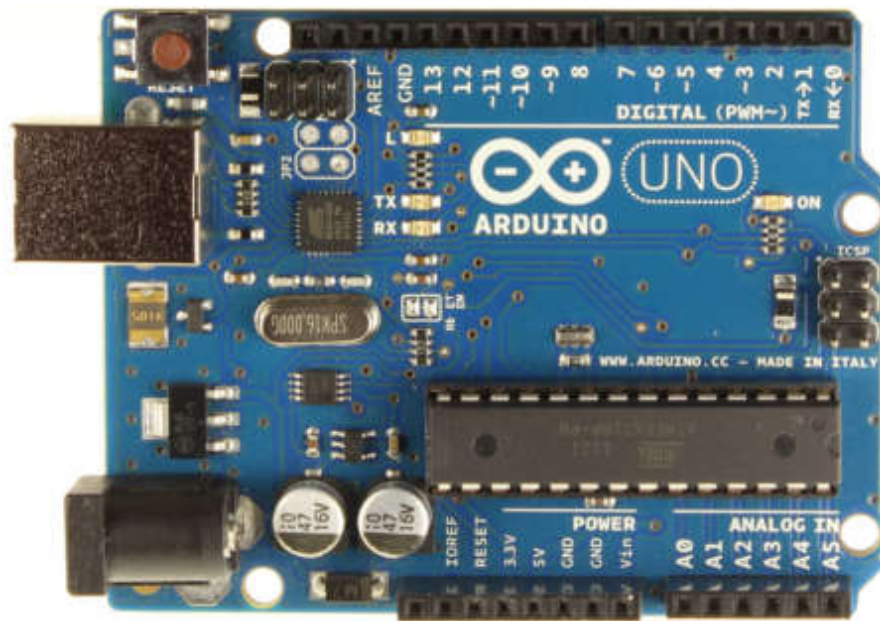
Kết luận: Trong chương này ta đã trình bày những cái nhìn tổng quan nhất về bo mạch Arduino. Trong chương 2, ta sẽ lựa chọn trình bày chi tiết về các loại bo mạch Arduino được sử dụng thông dụng nhất hiện nay với những tính năng xử lý chuyên biệt.

CÁC LOẠI BO MẠCH ARDUINO THÔNG DỤNG

2.1. Bo mạch Arduino Uno.

2.1.1. Tổng quan.

Arduino Uno là bo mạch vi xử lý hoạt động dựa trên ATmega328. Bo mạch này có 14 chân input/output digital (trong đó có 6 chân được dùng cho điều chế xung đầu ra PWM), 6 đầu vào analog, tần số giao động thạch anh là 16MHz, kết nối USB, jack cắm nguồn, chân tiêu đề ICSP, một nút reset. Bo mạch này chứa tất cả các tính năng cần thiết để hỗ trợ kết nối với các vi điều khiển khác. Nguồn sử dụng cho bo mạch có thể qua USB, sử dụng pin hoặc nguồn thông qua bộ chuyển đổi AC-DC.



Hình 8. Bo mạch Arduino Uno

Arduino Uno khác với tất cả các bo mạch khác ở chỗ nó không sử dụng chip điều khiển nối tiếp FTDI USB. Thay vào đó các tính năng của ATmega16U2 được lập trình để chuyển đổi USB nối tiếp.

Ở phiên bản thứ hai: bo mạch Uno có điện trở nối đường 8U2 HWB với đất, do đó ta dễ dàng hơn trong việc thiết lập chế độ DFU.

Ở phiên bản sửa đổi thứ 3 của bo mạch có các tính năng mới dưới đây:

- Sơ đồ chân 1.0: Thêm các chân SDA và SCL gần với chân AREF và 2 chân mới được đặt gần chân RESET, IOREF cho phép Shield nhận nguồn cấp từ bo mạch. Trong tương lai, Shield sẽ tương thích với các bo mạch có sử dụng AVR, có điện áp hoạt động là 5V và Arduino Due hoạt động ở 3.3V. Một số chân trong bo mạch không được kết nối gì để dành cho mục đích trong tương lai.
- Mạch Reset mạnh mẽ hơn.
- ATmega 16U2 thay thế cho 8U2.

"Uno" là từ trong tiếng Ý và được đặt tên để đánh dấu việc phát hành phiên bản Arduino 1.0. Uno và phiên bản 1.0 sẽ là phiên bản tham khảo của Arduino, và luôn có sự cải tiến. Uno là một trong những bo mạch mới nhất trong một loạt các bo mạch USB Arduino, và là mô hình tham chiếu nền tảng của Arduino, để so sánh với phiên bản trước đó, xem các thông số của bo mạch Arduino.

2.1.2. Tóm tắt các thông số.

Bảng 2. Tóm tắt các thông số chính của bo mạch Arduino Uno

Chip xử lý	ATmega328
Điện áp hoạt động	5V
Điện áp vào (khuyến nghị)	7-12V
Điện áp vào (giới hạn)	6-20V
Số chân I/O Digital	14 (6 chân đầu ra PWM)
Số chân đầu vào Analog	6
Dòng DC trên chân I/O	40 mA
Dòng DC trên chân 3.3V	50 mA
Flash Memory	32 KB (0.5 KB sử dụng cho bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Tốc độ xung	16 MHz

2.1.3. Thiết kế cấu trúc mạch tham khảo (Schematic).

Xem phần phụ lục 1: *Sơ đồ mạch nguyên lý Arduino Uno* hoặc xem qua đường link: http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

Lưu ý: Thiết kế tham khảo Arduino có thể sử dụng ATmega8, 168, hoặc 328, các mô hình hiện tại sử dụng ATmega328, Nhưng ATmega8 được thể hiện trong sơ đồ để tham khảo. Cấu hình chân là giống hệt nhau trên tất cả ba bộ vi xử lý.

2.1.4. Nguồn cấp.

Arduino Uno có thể được cấp nguồn thông qua kết nối USB hoặc với một nguồn cung cấp điện bên ngoài. Nguồn điện được chọn một cách tự động. Nguồn cấp bên ngoài (không phải là USB) có thể lấy từ bộ chuyển đổi AC-DC hoặc nguồn pin. Các bộ chuyển đổi có thể được kết nối bằng cách cắm chân cắm đường kính 2.1mm vào lỗ cắm điện trên bo mạch. Nếu nguồn lấy từ pin có thể được lắp vào 2 đầu GND và Vin chân tiêu đề của kết nối POWER.

Bo mạch có thể hoạt động với các nguồn cấp ngoài từ 6 đến 20 volt. Nếu nguồn ít nhất thường là 7V, tuy nhiên, các chân 5V có thể được cấp nguồn bé hơn 5V nhưng khi đó mạch có thể hoạt động không ổn định. Nếu sử dụng hơn 12V, bộ ổn áp bị nóng và hỏng mạch, khuyến nghị nên sử dụng ở khoảng 7 đến 12 volts.

Nguồn cấp của các chân như sau:

- Vin. Điện áp đầu vào của bo mạch Arduino khi nó sử dụng nguồn cấp ngoài (khác với các nguồn cấp 5V từ kết nối USB và các nguồn điện theo quy định). Ta có thể cấp nguồn qua chân Vin hoặc cấp nguồn thông qua các jack cắm kết nối với chân này.
- 5V. Chân đầu ra được quy định là 5V. Bo mạch có thể có thể được cấp nguồn điện từ các jack (7-12V), kết nối USB (5V), hoặc chân Vin của bo mạch (7-12V). Cung cấp điện áp thông qua chân 5V hoặc 3.3V bỏ qua các khuyến cáo có thể gây hỏng mạch. Không nên sử dụng nó.
- 3.3V. Nguồn cấp 3.3V được quy định trên bo mạch. Dòng cấp tối đa là 50mA.
- GND Chân nối đất.
- IOREF. Chân này cấp điện áp tham chiếu cho vi điều khiển hoạt động. Bộ hỗ trợ cấu hình chuẩn đọc điện áp trên chân OIREF và lựa chọn nguồn cấp thích hợp hoặc kích hoạt dịch điện áp trên đầu ra để làm việc với các nguồn 5V hoặc 3.3V.

2.1.5. Đầu vào đầu ra.

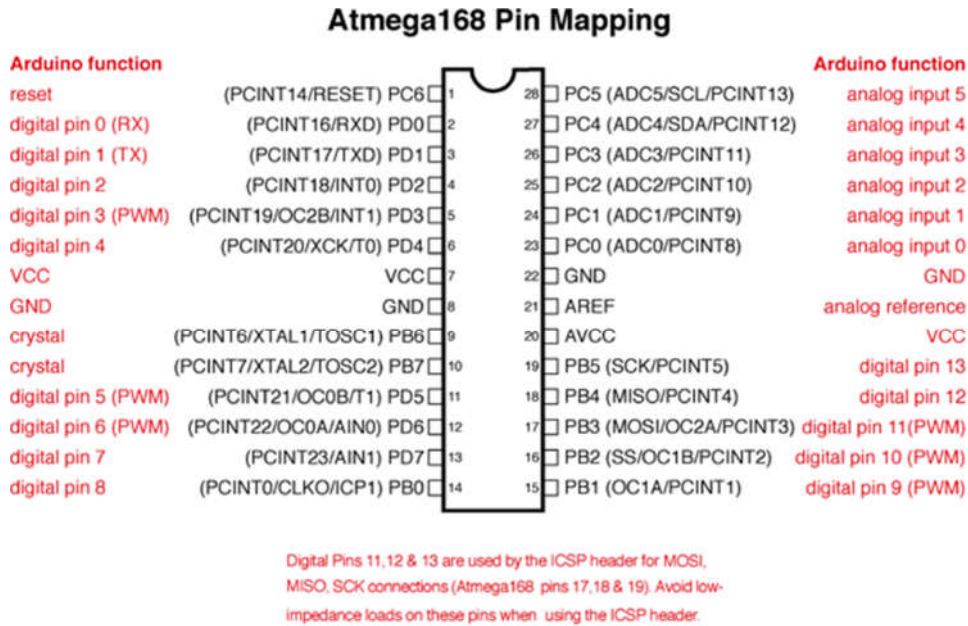
Trên bo mạch có 14 chân digital có thể được sử dụng cho mạch đích vào hoặc ra, sử dụng các hàm pinMode(), digitalWrite(), và digitalRead(). Chúng hoạt động ở mức điện áp 5V. mỗi chân có thể cung cấp hoặc nhận dòng cực đại là 40mA và được nối với điện trở mặc định từ 20-50 KΩ. Ngoài ra còn có một số chân có chức năng đặc biệt:

- Chân nối tiếp: 0 (RX) và 1 (TX). Sử dụng để nhận và truyền dữ liệu TTL. Chân này được nối với chân tương ứng của ATmega82U.
- Chân ngắt ngoài: 2 và 3. Chân này có thể được thiết lập để thực hiện ngắt khi điện áp quá thấp hoặc thay đổi đột biến giá trị điện áp. Thường sử dụng hàm ngắt attachInterrupt().
- PWM: 3, 5, 6, 9 và 10. Xung PWM có độ rộng là 8 bits. Khi điều chế xung đầu ra sử dụng hàm analogWrite().
- SPI: 10(SS), 11 (MOSI), 12 (MISO), 13 (SCK). Các chân này hỗ trợ truyền dẫn SPI sử dụng thư viện SPI.
- LED: 13. Có một đèn LED được nối với chân 13. Khi điện áp ở mức cao LED sáng và ở mức thấp thì LED tắt.

Uno có 6 đầu ra analog, có nhãn là A0 đến A5, mỗi chân được cung cấp 10 bits dữ liệu (tương ứng với 1024 giá trị khác nhau). Trên bo mạch Ethernet có 6 chân đầu vào tương tự, có nhãn từ A0 đến A5, mỗi chân sử dụng 10 bits (tức là có 1024 giá trị khác nhau). Nguồn cấp cho các chân này là từ 0-5V, mặc dù nó có thể thay đổi phạm vi hoạt động của chúng bằng cách sử dụng chân AREF và hàm analogReference(). Ngoài ra, một số chân có chức năng chuyên biệt:

- TWI: A4 (SDA) và A5 (SCL). Hỗ trợ truyền thông TWI sử dụng thư viện Wire. Còn có hai chân khác trên bo mạch là:
- AREF: Điện áp tham chiếu cho đầu vào tương tự. Sử dụng với hàm analogReference().
- Reset. Được dùng để thiết lập lại vi điều khiển. Thường sử dụng nút reset để hỗ trợ các khối trên bo mạch.

Ta có thể xem kết nối giữa các chân Arduino và ATmega328 như sau:



Hình 9. Sơ đồ nối chân Arduino và ATmega328

Đối với sơ đồ chân của ATmega8, 168 và 328 thì tương tự.

2.1.6. Các thông số khác.

➤ *Bộ nhớ.*

ATmega328 có 32KB (với 0.5KB dành cho bootloader). Nó có 2KB bộ nhớ SRAM và 1KB bộ nhớ EEPROM (có thể đọc và ghi thông qua thư viện EEPROM library).

➤ *Truyền dẫn.*

Arduino Uno có một số phương thức để giao tiếp với máy tính và các Arduino khác, hoặc các dòng vi xử lý khác. ATmega328 cung cấp giao tiếp nối tiếp UART TTL (5V), trong đó có sẵn trên các chân digital 0 (RX) và 1 (TX). ATmega16U2 trên các kênh giao tiếp nối tiếp với bo mạch này thông qua cổng USB và xuất hiện cổng COM ảo kết nối với phần mềm máy tính. 16U2 sử dụng các trình điều khiển tiêu chuẩn phần cứng USB COM. Tuy nhiên khi thực hiện trên Window, một tập tin được yêu cầu. Phần mềm Arduino bao gồm một màn hình cho ta nhìn thấy được dữ liệu được chuyển đến bo mạch Arduino. LED RX và TX sẽ nhấp nháy khi dữ liệu được chuyển thông qua kết nối cổng USB của chip và cổng USB của máy tính. (không thực hiện giao tiếp nối tiếp trên chân 0 và 1).

Thư viện SoftwareSerial cho phép giao tiếp nối tiếp trên bất kỳ chân digital nào của Uno. ATmega328 cũng hỗ trợ giao tiếp I2C (TWI) và SPI. Phần mềm

Arduino bao gồm một thư viện wire library để đơn giản hóa việc sử dụng bus I2C. Với truyền dẫn SPI, sử dụng thư viện SPI.

➤ *Lập trình.*

Arduino Uno có thể được lập trình với các phần mềm Arduino. Chọn " Arduino Uno từ thanh công cụ Tools → Board menu (theo vị trí điều khiển trên bo mạch). Để biết chi tiết, xem tài liệu tham khảo và hướng dẫn.

ATmega328 trên Arduino Uno đi kèm với một bộ nạp khởi động preburned cho phép tải code mới lên mà không cần sử dụng lập trình phần cứng bên ngoài. Nó giao tiếp bằng cách sử dụng giao thức bus đầu STK500 (tham chiếu, tập tin tiêu đề viết bằng ngôn ngữ C).

Ta cũng có thể bỏ qua bộ nạp khởi động và chương trình vi điều khiển thông qua ICSP (In-Circuit Serial Programming).

Ở các dòng ATmega16U2 (hoặc 8U2) mã nguồn phần mềm có sẵn. Các ATmega16U2/8U2 được nạp với một bộ nạp khởi động DFU, mà có thể được kích hoạt bằng cách:

- Trên thế hệ bo mạch thứ 1: kết nối jumper được hàn ở mặt sau của bo mạch và sau đó cài đặt lại 8U2.
- Trên thế hệ bo mạch thứ 2 hoặc cuối cùng: có một điện trở nối HWB 8U2/16U2 với đất, làm cho nó dễ dàng hơn trong việc đặt chế độ DFU.

Sau đó có thể sử dụng phần mềm FLIP Atmel (Windows) hoặc lập trình DFU (Mac OS X và Linux) để tải một phần mềm mới. Hoặc có thể sử dụng header ISP lập trình ngoài (ghi đè lên các bộ nạp khởi động DFU).

➤ *Tự động reset (phần mềm).*

Thay vì reset vật lý trước khi tải dữ liệu, Arduino Uno được thiết kế để cho phép thay thế bằng một phần mềm chạy trên máy tính khi được kết nối. Một trong những dòng điều khiển phần cứng (DTR) của ATmega8U2/16U2 được kết nối với đường dây được thiết lập của ATmega328 thông qua một tụ điện 100 nF. Khi dòng này được xác định (thấp nhất), dòng reset rơi đủ lớn để reset lại chip. Phần mềm Arduino sử dụng khả năng này để cho phép tải code lên bằng cách nhấn nút upload trong môi trường Arduino. Điều này có nghĩa là bộ nạp khởi động có thể có một thời gian chờ ngắn hơn, giảm thiểu DTR có thể phối hợp khi bắt đầu tải lên.

Thiết lập này có ý nghĩa khác. Khi Uno được kết nối với một trong hai máy tính chạy Mac OS X hoặc Linux, nó reset mỗi khi kết nối được thực hiện từ phần mềm (thông qua cổng USB). Trong nửa giây hoặc lâu hơn, bộ nạp khởi động chạy trên Uno. Trong khi nó được lập trình để bỏ qua dữ liệu bị thay đổi, nó sẽ ngăn chặn các byte đầu tiên của dữ liệu gửi đến bo mạch khi kết nối được thiết lập. Nếu

một bản sketch chạy trên bo mạch nhận được cấu hình một lần hoặc các dữ liệu khác khi lần đầu tiên bắt đầu, chắc chắn rằng phần mềm mà bo mạch giao tiếp chờ đợi một giây sau khi mở kết nối và trước khi gửi dữ liệu này.

➤ *Bảo vệ quá dòng USB.*

Arduino Uno có một bảng thiết lập được sử dụng để bảo vệ cổng USB của máy tính khi ngắn mạch và quá dòng. Mặc dù hầu hết các máy tính cung cấp chế độ bảo vệ nội bộ, nhưng còn có thêm các cầu chì, các cầu chì này có thêm một lớp bảo vệ. Nếu dòng cao hơn 500 mA được áp dụng cho các cổng USB, cầu chì sẽ tự động phá vỡ các kết nối khi ngắn mạch và quá dòng xảy ra.

➤ *Đặc tính vật lý.*

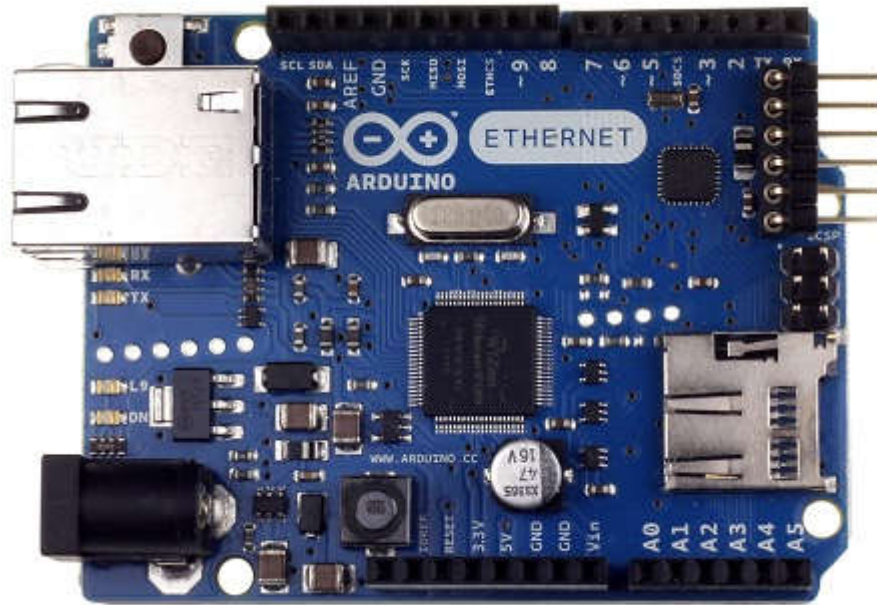
Chiều dài tối đa và chiều rộng tương ứng của Uno PCB là 2,7 và 2,1 inch, với các kết nối USB và jack nguồn thì sẽ vượt ra ngoài kích thước cũ. Bốn lỗ vít trên bo mạch dùng để gắn vào một bề mặt trong một số trường hợp. Lưu ý rằng khoảng cách giữa chân số 7 và 8 là 160 mil (0,16 ").

2.2. Bo mạch Arduino Ethernet.

2.2.1. Tổng quan

Arduino Ethernet là bo mạch vi xử lý xây dựng dựa trên chip ATmega328. Bo mạch có 14 chân số input/output, 6 chân input tương tự, dao động thạch anh 16MHz, kết nối mạng thông qua cổng RJ45, jack cắm nguồn, đầu ICSP, và nút Reset.

Trên bo mạch này, chân 10, 11, 12 và 13 được dành riêng để giao tiếp với module Ethernet và không nên sử dụng các chân này cho mục đích khác. Do vậy mà số chân dành sẵn bây giờ chỉ còn là 9 chân, với 4 chân sử dụng cho điều chế xung đầu ra PWM. Có thể thiết lập nguồn cấp trên module Ethernet.



Hình 10. Bo mạch Arduino Ethernet

Module Ethernet khác với các bo mạch khác ở chỗ nó không có các cổng USB kết nối trực tiếp với chip nhưng lại có giao tiếp Wiznet Ethernet. Điều này có thể tìm thấy tương tự như trên các bo mạch Ethernet Shield.

Trên bo mạch có đầu đọc thẻ microSD, nó có thể được dùng để lưu trữ files cho dịch vụ mạng, và truy nhập thông qua thư viện SD. Chân 10 được dành cho giao tiếp Wiznet, chân 4 (SS) để kết nối với thẻ nhớ SD.

Bo mạch có 6 chân header lập trình liên tiếp tương thích với chuẩn giao tiếp USB và phù hợp với các loại cáp FTDI USB hoặc với Sparkun và Adafruit FTDI dựa trên kết nối các kết nối USB cơ bản. Nó có các tính năng tự động reset, cho phép thực hiện tải dữ liệu mà không cần phải ấn nút reset trên bo mạch. Khi cổng USB được kết nối với nguồn cấp, Arduino Ethernet sẽ được cấp nguồn USB.

Trong bo mạch sửa đổi lần 3 giới thiệu về tiêu chuẩn sơ đồ chân 1.0, trong đó bao gồm:

- Thêm các chân SDA và SCL là các chân gần chân AREF và hai chân mới được đặt gần chân Reset, điều này tạo điều kiện để bảo vệ việc sử dụng thành phần I2C và TWI để tương thích với tất cả các bo mạch Arduino.
- IOREF cho phép hỗ trợ để thích ứng với nguồn cung cấp trên bo mạch. Thành phần hỗ trợ này sử dụng chân IOREF sẽ tương thích với bo mạch sử dụng AVR,

hoạt động ở 5V và với Arduino Due hoạt động ở 3.3V. Chân IOREF là một chân không được kết nối và được sử dụng trong tương lai.

2.2.2. Tóm tắt các thông số.

Bảng 3. Tóm tắt các thông số của bo mạch Arduino Ethernet

Chip xử lý	ATmega328
Điện áp hoạt động	5V
Điện áp vào (khuyến nghị)	7-12V
Điện áp vào (giới hạn)	6-20V
Điện áp vào PoE (giới hạn)	36-57V
Chân số Input/Output	14 (4 chân điều chế xung đầu ra PWM)
Các chân Arduino dành riêng:	Chân 10 đến 13 sử dụng cho SPI
	Chân 4 sử dụng kết nối SD card
	2 chân W5100 – chân ngắt (khi nối cầu)
Chân đầu vào tương tự	6
Dòng DC trên chân Input/Output	40 mA
Dòng DC với chân 3.3V	50 mA
Flash Memory	32 KB (ATmega328) trong đó 0.5 KB dùng cho bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Tốc độ xung	16 MHz
W5100 TCP/IP Điều khiển nhúng Ethernet Nguồn cấp Ethernet từ Jack chuẩn MicroSD card, với chuyển đổi điện áp hoạt động	

2.2.3. Thiết kế cấu trúc mạch tham khảo (Schematic)

Xem phụ lục 2: *Sơ đồ nguyên lý bo mạch Arduino Ethernet* hoặc xem theo đường link: <http://arduino.cc/en/uploads/Main/arduino-ethernet-R3-schematic.pdf>

2.2.4. Nguồn cấp

Bo mạch có thể được cấp nguồn thông qua nguồn cấp bên ngoài, nguồn tùy chọn trên Modul Ethernet (PoE), hoặc sử dụng thông qua kết nối cáp FTDI/USB.

Các nguồn bên ngoài có thể được sử dụng thông qua bộ chuyển đổi AC-DC hoặc nguồn pin. Adapter có thể kết nối thông qua jack cắm đường kính 2.1mm trên bo mạch. Khi sử dụng nguồn pin thì cần phải có các cực Vin và Gnd.

Bo mạch có thể hoạt động với các nguồn cấp ngoài từ 6 đến 20 volt. Nếu nguồn ít nhất thường là 7V, tuy nhiên, các chân 5V có thể được cấp nguồn bé hơn 5V nhưng khi đó mạch có thể hoạt động không ổn định. Nếu sử dụng hơn 12V, bộ ổn áp bị nóng và phá hủy mạch, khuyến nghị nên sử dụng ở khoảng 7 đến 12 volts.

Nguồn cấp của các chân như sau:

- Vin. Điện áp đầu vào của bo mạch Arduino khi nó sử dụng nguồn cấp ngoài (khác với các nguồn cấp 5V từ kết nối USB và các nguồn điện theo quy định). Ta có thể cấp nguồn qua chân Vin hoặc cấp nguồn thông qua các jack cắm kết nối với chân này.
- 5V. Chân đầu ra được quy định là 5V. Bo mạch có thể có thể được cấp nguồn điện từ các jack (7-12V), kết nối USB (5V), hoặc chân Vin của bo mạch (7-12V). Cung cấp điện áp thông qua chân 5V hoặc 3.3V bỏ qua các khuyến cáo có thể gây hỏng mạch. Không nên sử dụng nó.
- 3.3V. Nguồn cấp 3.3V được quy định trên bo mạch. Dòng cấp tối đa là 50mA.
- GND Chân nối đất.
- IOREF. Chân này cấp điện áp tham chiếu cho vi điều khiển hoạt động. Bộ hỗ trợ cấu hình chuẩn đọc điện áp trên chân OIREF và lựa chọn nguồn cấp thích hợp hoặc kích hoạt dịch điện áp trên đầu ra để làm việc với các nguồn 5V hoặc 3.3V.

Các tùy chọn module PoE được thiết kế để tách nguồn từ cáp xoắn thường Category 5 Ethernet.

Nguồn cấp trên module Ethernet là phần cứng riêng rẽ không được thực hiện bởi Arduino, nó là một phụ kiện thứ 3. Để biết thêm thông tin chi tiết nên tham khảo datasheet.

2.2.5. Đầu vào và đầu ra

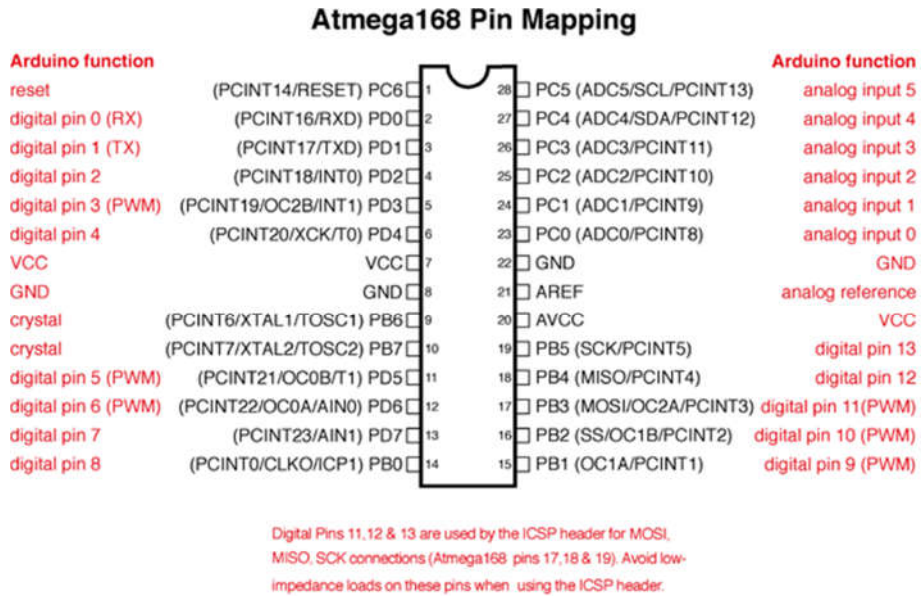
Trên bo mạch có 14 chân digital có thể được sử dụng cho mạch đích vào hoặc ra, sử dụng các hàm pinMode(), digitalWrite(), và digitalRead(). Chúng hoạt động ở mức điện áp 5V. mỗi chân có thể cung cấp hoặc nhận dòng cực đại là 40mA và được nối với điện trở mặc định từ 20-50 K Ω . Ngoài ra còn có một số chân có chức năng đặc biệt.

- Chân nối tiếp: 0 (RX) và 1 (TX). Sử dụng để nhận và truyền dữ liệu TTL.
- Chân ngắt ngoài: 2 và 3. Chân này có thể được thiết lập để thực hiện ngắt khi điện áp quá thấp hoặc thay đổi đột biến giá trị điện áp. Thường sử dụng hàm ngắt `attachInterrupt()`.
- PWM: 3, 5, 6, 9 và 10. Xung PWM có độ rộng là 8 bits. Khi điều chế xung đầu ra sử dụng hàm `analogWrite()`.
- SPI: 10(SS), 11 (MOSI), 12 (MISO), 13 (SCK). Các chân này hỗ trợ truyền dẫn SPI sử dụng thư viện SPI.
- LED: 9. Có một đèn LED được nối với chân 9. Khi điện áp ở mức cao LED sáng và ở mức thấp thì LED tắt. Trong đa số các bo mạch Arduino khác, LED thường nằm ở chân 13. Nhưng trên bo mạch Ethernet lại là chân số 9 bởi vì chân 13 đã được sử dụng cho kết nối SPI.

Trên bo mạch Ethernet có 6 chân đầu vào tương tự, có nhãn từ A₀ đến A₅, mỗi chân sử dụng 10 bits (tức là có 1024 giá trị khác nhau). Nguồn cấp cho các chân này là từ 0-5V, mặc dù nó có thể thay đổi phạm vi hoạt động của chúng bằng cách sử dụng chân AREF và hàm `analogReference()`. Ngoài ra, một số chân có chức năng chuyên biệt:

- TWI: A4 (SDA) và A5 (SCL). Hỗ trợ truyền thông TWI sử dụng thư viện `Wire`. Còn có hai chân khác trên bo mạch là:
- AREF: Điện áp tham chiếu cho đầu vào tương tự. Sử dụng với hàm `analogReference()`.
- Reset. Được dùng để thiết lập lại vi điều khiển. Thường sử dụng nút reset để hỗ trợ các khối trên bo mạch.

Dưới đây là sơ đồ chân của Arduino và ATmega328.



Hình 11. Sơ đồ nối chân của Arduino và ATmega328.

2.2.6. Các thông số khác.

➤ Bộ nhớ

ATmega328 có 32 Kb (với 0.5 Kb sử dụng cho bootloader). Nó cũng có bộ nhớ SRAM 2Kb và EEPROM.

➤ Truyền dẫn

Arduino Ethernet được có các tính năng được sử dụng để giao tiếp với máy tính và Arduino khác và các vi điều khiển khác. Thư viện Software cho phép truyền dẫn trực tiếp trên bất kỳ chân digital nào của Uno.

ATmega328 cũng hỗ trợ truyền dẫn TWI và SPI. Phần mềm Arduino bao gồm thư viện Wire thường sử dụng cho bus TWI. Với truyền dẫn SPI, ta sử dụng thư viện SPI.

Bo mạch cũng có thể kết nối với dây mạng thông qua Ethernet. Khi kết nối với mạng, bo mạch cần được cung cấp địa chỉ IP và địa chỉ MAC. Hỗ trợ thư viện Ethernet. Trên bo mạch có khe đọc thẻ microSD thông qua thư viện SD. Kết nối SS tại chân 4.

➤ Lập trình.

Có thể lập trình trên bo mạch Ethernet Arduino trong hai cách: thông qua 6 chân header lập trình nối tiếp, hoặc lập trình ISP bên ngoài.

Ta có thể lập trình bo mạch Ethernet với chương trình ngoài như AVRISP mkII hoặc USBTinyISP. Thiết lập môi trường để ghi bản thảo lập trình theo hướng dẫn. Điều này sẽ xóa các bộ bootloader nối tiếp. Tuy nhiên để khắc phục điều này người ta làm việc với các bo mạch hỗ trợ Ethernet Shield. Chắc chắn điều này sẽ làm thay đổi các thiết lập mạng.

➤ *Đặc điểm vật lý*

Chiều dài và chiều rộng tối đa của PCB Ethernet tương ứng là 2.7 và 2.1 inch, với các kết nối RJ45 và jack nguồn sẽ vượt ra ngoài kích thước cũ. Bốn lỗ vít cho phép gắn bo mạch vào các bề mặt trong một số trường hợp cần thiết.

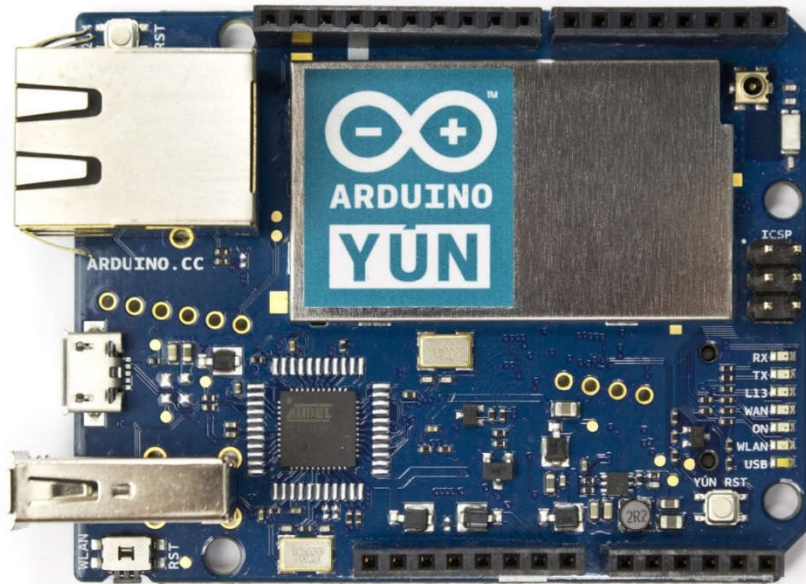
➤ *Cài đặt*

Nếu muốn tải giữ liệu về bo mạch Arduino Ethernet nên tham khảo các hướng dẫn: nâng cấp bootloader Arduino Ethernet theo phiên bản mới nhất.

2.3. Bo mạch Arduino Yun.

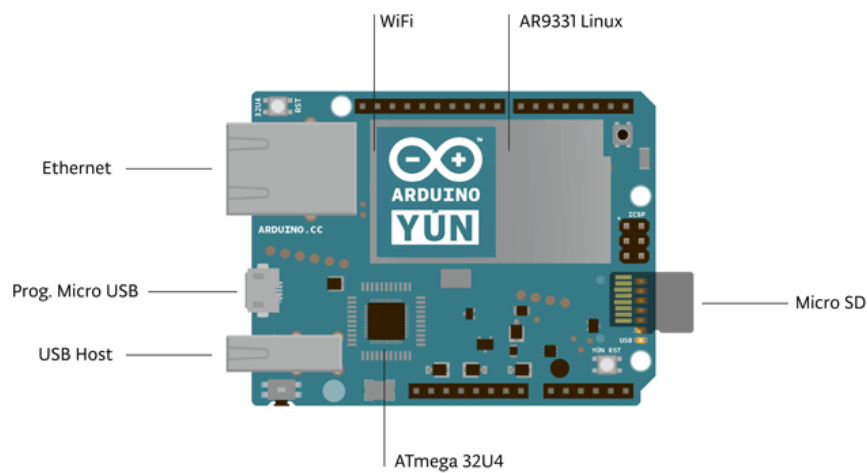
2.3.1. Tổng quan.

Arduino Yun là một bo mạch vi điều khiển hoạt động dựa trên ATmega32u4 và Atheros AR9331. Bộ vi xử lý Atheros hỗ trợ hệ điều hành Linux dựa trên OpenWRT có tên Linino. Bo mạch được xây dựng có hỗ trợ Ethernet, Wi-Fi, cổng USB, khe cắm thẻ nhớ microSD, 20 đầu vào / đầu ra digital (trong đó có 7 chân có thể được sử dụng để điều chế xung đầu ra PWM và 12 đầu vào analog), dao động thạch anh 16 MHz, kết nối micro USB, một header ICSP, và 3 nút reset.



Hình 12. Bo mạch Arduino Yun.

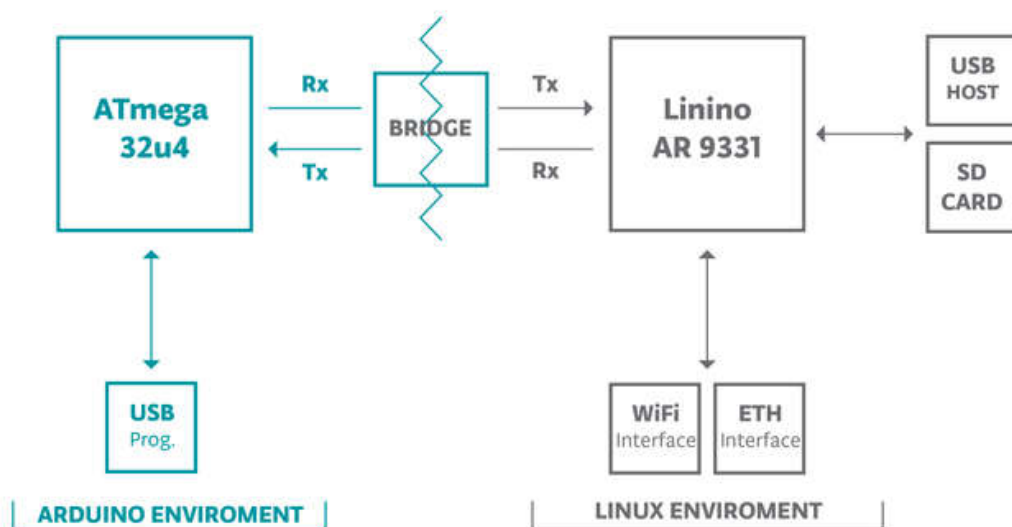
Lưu ý: Ở một số nước, cấm bán các thiết bị kích hoạt WiFi nếu không có sự chấp thuận của chính phủ. Trong khi chờ đợi chấp nhận, một số nhà phân phối địa phương vô đã hiệu hóa chức năng Wi-Fi. Vì vậy việc sử dụng Yun trong trường hợp này là cần thiết. Nếu muốn vô hiệu hóa Wi-Fi, chạy sketch. Để biết thêm thông tin, tham khảo bài đăng này diễn đàn về Arduino.



Hình 13. Một số thành phần chính trên bo mạch.

Yun khác biệt với những bo mạch Arduino khác ở chỗ nó có thể giao tiếp với hệ điều hành Linux (được phân phối cho bo mạch), do vậy máy tính có kết nối mạng mạnh mẽ và đơn giản hơn với Arduino. Ngoài lệnh Linux như cURL, có thể viết thêm các chương trình riêng và các tập lệnh cho các tương tác mạnh mẽ.

Yun cũng tương tự như Leonardo trong đó ATmega32u4 được xây dựng sử dụng giao tiếp USB, loại bỏ một số thứ không cần thiết so với bộ vi xử lý thứ cấp. Điều này cho phép Yun tương tác với một máy tính khi kết nối giống như là chuột và bàn phím, cổng (CDC) nối tiếp / cổng COM ảo.



Hình 14. Sơ đồ kết nối dữ liệu của các thành phần bo mạch.

Thư viện Bridge tạo điều kiện truyền thông tin giữa hai bộ vi xử lý, cho phép Arduino chạy các shell scripts, giao tiếp với giao diện mạng, và nhận thông tin từ bộ vi xử lý AR9331. Các cổng USB, giao tiếp mạng và thẻ nhớ SD không kết nối với 32U4, nhưng kết nối với AR9331, và thư viện Bridge cũng cho phép Arduino giao tiếp với những thiết bị ngoại vi.

2.3.2. Tóm tắt các thông số.

Do Arduino Yun có hai vi xử lý, vì vậy phải có hai bộ thông số được trình bày bằng hai bảng riêng biệt. Trong bảng dưới đây ta sẽ trình bày về vi xử lý đóng vai trò quan trọng nhất là ATmega32u4.

Bảng 4. Tóm tắt thông số Arduino Yun

Chíp xử lý AVR Arduino	
Chíp xử lý	ATmega32u4
Điện áp hoạt động	5V
Điện áp vào	5V
Số chân I/O digital	20
Số kênh PWM	7
Số kênh đầu vào analog	12
Dòng DC trên chân I/O	40 mA
Dòng DC cho chân 3.3V	50 mA
Flash Memory	32 KB (4 KB dùng cho bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Tốc độ xung	16 MHz
Vi xử lý Linux	
Vi xử lý	Atheros AR9331
Architecture	MIPS @400MHz
Điện áp hoạt động	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host/Device
Card Reader	Micro-SD only
RAM	64 MB DDR2
Flash Memory	16 MB
PoE hỗ trợ tương thích với card 802.3af	

2.3.3. Sơ đồ mạch nguyên lý tham khảo (Schematic).

Có thể vào đường link này để xem thêm chi tiết:

<http://arduino.cc/en/uploads/Main/arduino-Yun-schematic.pdf>

2.3.4. Nguồn cấp.

Khuyến nghị nên sử dụng nguồn cấp cho bo mạch thông qua kết nối micro-USB với điện áp 5VDC. Nếu cung cấp năng lượng cho bo mạch qua chân Vin thì

cũng phải cấp nguồn 5VDC như quy định. Không có điều chỉnh điện áp trên bo mạch lên cao hơn, như vậy sẽ làm hỏng bo mạch.

Yun cũng tương thích với nguồn PoE nhưng để sử dụng tính năng này cần phải gắn một module PoE trên bo mạch hoặc mua bo mạch mà PoE được lắp ráp sẵn.

Nguồn cấp của các chân như sau:

- VIN. Điện áp đầu vào bo mạch Arduino. Không giống như các bo mạch Arduino khác, nếu muốn cấp nguồn cho bo mạch thông qua chân này, phải cung cấp nguồn 5V như quy định.
- 5V. Là nguồn cấp được sử dụng chạy vi điều khiển và các thành phần khác trên bo mạch. Điện áp này có thể lấy từ chân Vin hoặc USB.
- 3V3. Nguồn cung cấp 3.3V của một số chân trên bo mạch. Dòng điện tối đa qua các chân này là 50 mA.
- GND. Chân nối đất.
- IOREF. Điện áp mà tại đó các chân I/O hoạt động (tức là nguồn VCC cho bo mạch). Với Yun là 5V.

2.3.5. Bộ nhớ.

Bộ nhớ ATmega32u4 có 32 KB (với 4 KB được sử dụng cho bootloader). Nó cũng có 2,5 KB SRAM và 1 KB EEPROM.

Bộ nhớ trên AR9331 không được nhúng vào bên trong bộ vi xử lý. Bộ nhớ RAM và bộ nhớ lưu trữ được kết nối với bên ngoài. Yun có 64 MB bộ nhớ RAM DDR2 và 16 MB bộ nhớ flash. Bộ nhớ flash được cài đặt khi sản xuất kết hợp với hệ điều hành Linux dựa trên OpenWRT có tên là Linino. Ta có thể thay đổi nội dung của hình ảnh mà nhà máy sản xuất, chẳng hạn như khi cài đặt một chương trình hoặc khi thay đổi một tập tin cấu hình. Có thể trở lại cấu hình ban đầu bằng cách nhấn nút "WLAN RST" trong 30 giây.

2.3.6. Đầu vào đầu ra.

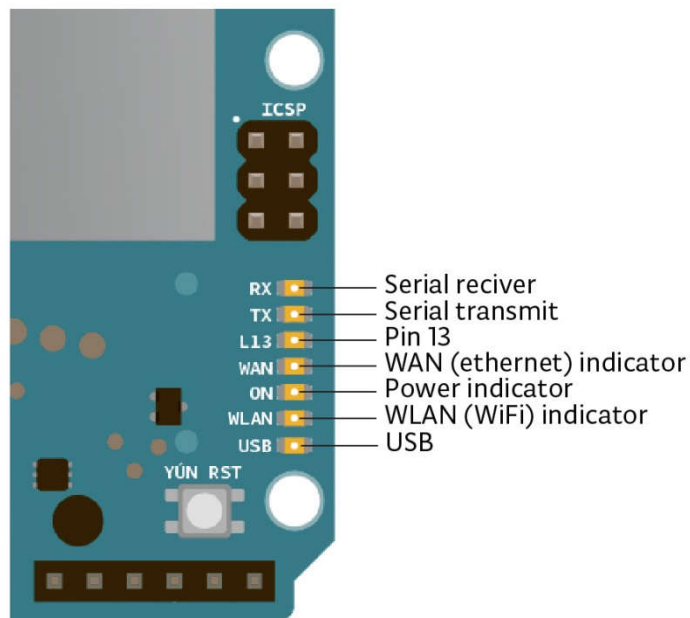
Bo mạch có thể không truy nhập I/O của Atheros AR33. Tất cả các đầu vào I/O dòng được gắn với 32U4. Trong số 20 chân I/O trên Yun có thể được sử dụng khai báo đầu vào hoặc đầu ra, sử dụng các hàm pinMode(), digitalWrite(), và digitalRead (), hoạt động ở 5 volts. Mỗi chân có thể cung cấp hoặc nhận được tối đa

dòng là 40 mA và có một điện trở 20-50 k Ω . Ngoài ra, một số chân có chức năng riêng:

- Serial: 0 (RX) và 1 (TX). Được sử dụng để nhận (RX) và truyền tải (TX) dữ liệu nối tiếp TTL bằng cách sử dụng khả năng ghép nối ATmega32U4 với phần cứng. Lưu ý rằng trên Yun, ghép nối tiếp thông qua USB (CDC); với TTL nối tiếp trên các chân 0 và 1, sử dụng class Serial1. Các phần cứng của ATmega32U4 và AR9331 trên Yun được kết nối với nhau và được sử dụng để giao tiếp giữa hai bộ xử lý. Như là phổ biến trong các hệ thống Linux, trên các cổng nối tiếp của AR9331 được ghép nối với các giao tiếp điều khiển để truy cập vào hệ thống.
- TWI: 2 (SDA) và 3 (SCL). Hỗ trợ truyền TWI bằng cách sử dụng thư viện Wire library.
- Chân ngắt ngoài: 3 (interrupt 0), 2 (interrupt 1), 0 (interrupt 2), 1 (interrupt 3) và 7 (interrupt 4). Các chân có thể được cấu hình để kích hoạt ngắt khi ngắn mạch hoặc thay đổi bất thường giá trị điện áp. Xem hàm attachInterrupt() để biết chi tiết. Không nên sử dụng chân 0 và 1 như chân ngắt bởi vì chúng là đường nối tiếp phần cứng được sử dụng để giao tiếp với bộ vi xử lý Linux. Chân 7 được kết nối với bộ vi xử lý AR9331 và nó có thể được sử dụng cho việc “bắt tay tín hiệu” trong tương lai.
- PWM: 3, 5, 6, 9, 10, 11, và 13. Cung cấp đầu ra PWM 8-bit sử dụng hàm analogWrite ().
- SPI: header ICSP. Các chân hỗ trợ SPI giao tiếp bằng cách sử dụng thư viện SPI library. Lưu ý rằng các chân SPI không kết nối với bất kỳ chân I/O digital nào khi chúng đang ở trên Uno, Chúng chỉ có sẵn trên các kết nối ICSP. Điều này có nghĩa rằng nếu có một shield có sử dụng SPI, nhưng không có một kết nối ICSP chân 6 kết nối đến chân 6 header ICSP của Yun, Shield sẽ không làm việc.

Chân SPI cũng được kết nối với AR9331 chân Gpio, nơi mà nó được thực hiện trong phần mềm giao diện SPI. Điều này có nghĩa rằng ATmega32u4 và AR9331 cũng có thể giao tiếp bằng cách sử dụng giao thức SPI.

- LED: 13. Có tích hợp đèn LED kết nối với chân digital 13. Khi điện áp trên chân ở mức cao, đèn LED được bật, và ngược lại.
- Có một số đèn LED trạng thái khác trên Yun, cho thấy khả năng kết nối WLAN, kết nối WAN và USB.



Hình 15. Vị trí một số chân chuyên biệt.

- Đầu vào tương tự: A0 - A5, A6 - A11 (trên digital 4, 6, 8, 9, 10, và 12). Yun có 12 đầu vào tương tự, nhãn A0 đến A11, tất cả trong số đó cũng có thể được sử dụng như chân I/O digital. Chân A0-A5 xuất hiện ở những vị trí tương tự như trên Uno; đầu vào A6-A11 trên chân I/O digital tương ứng là 4, 6, 8, 9, 10, và 12. Mỗi đầu vào tương tự cung cấp độ phân giải 10 bit (1024 mức giá trị khác nhau). Theo mặc định, đầu vào tương tự là 5 volts.
- Aref. Điện áp tham chiếu cho các đầu vào tương tự. Được sử dụng với hàm `analogReference()`.

Có 3 nút reset với các chức năng khác nhau trên bo mạch:

- Yun RST. Mang dòng thấp để reset bộ vi xử lý AR9331. Reset AR9331 để reboot lại hệ thống Linux. Tất cả các dữ liệu được lưu trữ trong RAM sẽ mất và tất cả các chương trình đang chạy sẽ chấm dứt.
- 32U4 RST. Mang dòng thấp để reset vi điều khiển ATmega32U4. Được sử dụng để Reset khi gắn thêm khối Shield trên bo mạch.
- WLAN RST. Nút này có hai chức năng. Chức năng sơ cấp để khôi phục lại cấu hình WiFi của máy. Cấu hình máy bao gồm việc đặt WiFi cho Yun trong chế độ điểm truy cập (AP) và gán cho nó địa chỉ IP mặc định là 192.168.240.1 , trong

tình huống này, có thể kết nối với máy tính của bạn với một mạng WiFi xuất hiện với tên SSID. Để khôi phục lại cấu hình WiFi của bạn, bạn phải bấm và giữ nút WLAN RST trong 5 giây. Khi bạn bấm nút WLAN LED màu xanh sẽ bắt đầu nhấp nháy và sẽ tiếp tục vẫn nhấp nháy khi bạn bỏ tay ra nút sau 5 giây chỉ ra rằng các thủ tục khôi phục WiFi đã được ghi nhận. Chức năng thứ hai của nút WLAN RST là để khôi phục lại chế độ dữ liệu mặc định linux, để khôi phục lại môi trường Linux, bạn phải nhấn nút trong 30 giây. Lưu ý rằng khôi phục lại sẽ làm mất tất cả các tập tin được lưu và các phần mềm cài đặt trên bộ nhớ flash-on-board kết nối với AR9331.

2.3.7. Truyền dẫn.

Yun có một số phương thức để giao tiếp với máy tính, Arduino khác, hoặc vi điều khiển khác. ATmega32U4 cung cấp giao tiếp nối tiếp chuyên dụng UART TTL (5V). 32U4 cũng cho phép ghép nối tiếp (CDC) thông qua cổng USB và xuất hiện như cổng com ảo để giao tiếp phần mềm trên máy tính. Chip này cũng hoạt động như một thiết bị USB 2.0 tốc độ cao, sử dụng chuẩn COM USB Driver. Phần mềm Arduino bao gồm một màn hình nối tiếp cho phép dữ liệu văn bản đơn giản được gửi đến và đi từ bo mạch Arduino. LED RX và TX trên bo mạch sẽ nhấp nháy khi dữ liệu được truyền qua kết nối USB với máy tính.

Chân digital 0 và 1 được sử dụng cho giao tiếp nối tiếp giữa 32U4 và AR9331. Thông tin giữa các bộ vi xử lý được xử lý bởi các thư viện Bridge. Thư viện SoftwareSerial cho phép giao tiếp nối tiếp trên bất kỳ chân digital nào của Yun trừ chân 0 và 1.

ATmega32U4 cũng hỗ trợ giao tiếp I2C (TWI) và SPI. Phần mềm Arduino bao gồm một thư viện Wire để đơn giản hóa việc sử dụng bus I2C. Với truyền dẫn SPI, sử dụng thư viện SPI library. Yun thực hiện các công việc như bàn phím và chuột, và có thể được lập trình để điều khiển các thiết bị đầu vào bằng cách sử dụng bàn phím và chuột.

Trên bo mạch Ethernet và giao diện WiFi được giao tiếp trực tiếp với bộ vi xử lý AR9331. Để gửi và nhận dữ liệu thông qua chúng, sử dụng thư viện Bridge. Để cấu hình các giao diện, có thể truy cập vào bảng điều khiển mạng như được mô tả trong phần đầu. Yun cũng có khả năng lưu trữ như USB thông qua Linino. Có thể kết nối thiết bị ngoại vi giống như thiết bị flash USB để thêm tính năng lưu trữ, bàn phím, hoặc webcam.

2.3.8. Lập trình

Yun có thể được lập trình với phần mềm Arduino. Lựa chọn Arduino Yun từ Tools → Board của thanh menu. Để biết thêm chi tiết nên tham khảo các tài liệu hướng dẫn.

ATmega32U4 trên Arduino Yun đi kèm với một bộ bootloader preburned cho phép tải code mới cho bo mạch mà không cần lập trình phần cứng bên ngoài. Truyền dẫn sử dụng giao thức AVR109. Ta cũng có thể bỏ qua bộ bootloader và chương trình vi điều khiển thông qua ICSP header (In-Circuit Serial Programming).

2.3.9. Một số tính chất khác.

➤ Tự động Reset và bootloader.

Thay vì yêu cầu ấn nút reset vật lý trước khi tải dữ liệu, Yun được thiết kế để reset bằng phần mềm trên máy tính thông qua kết nối. Các thiết lập lại được kích hoạt khi (CDC) serial của Yun/cổng COM được mở ra ở mức 1200baud và khi đó sẽ đóng lại. Khi điều này xảy ra, bộ xử lý sẽ thiết lập lại, phá vỡ kết nối USB với máy tính (có nghĩa là các cổng nối tiếp/ cổng COM ảo sẽ biến mất). Sau khi reset bộ vi xử lý, bộ nạp bootloader hoạt động trong khoảng 8 giây. Bộ bootloader cũng có thể được bắt đầu bằng cách nhấn nút reset trên Yun. Khi không reset được bằng phần mềm thì có thể reset vật lý trên bo mạch.

➤ Bảo vệ quá dòng USB.

Arduino Uno có một bảng thiết lập được sử dụng để bảo vệ cổng USB của máy tính khi ngắn mạch và quá dòng. Mặc dù hầu hết các máy tính cung cấp chế độ bảo vệ nội bộ, nhưng còn có thêm các cầu chì, các cầu chì này có thêm một lớp bảo vệ. Nếu dòng cao hơn 500 mA được áp dụng cho các cổng USB, cầu chì sẽ tự động phá vỡ các kết nối khi ngắn mạch và quá dòng xảy ra.

➤ Đặc tính vật lý.

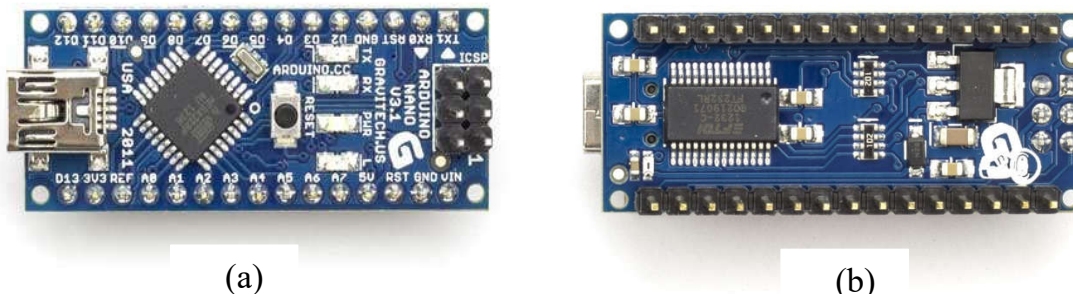
Chiều dài và chiều rộng tối đa của PCB Ethernet tương ứng là 2.7 và 2.1 inch, với các kết nối RJ45 và jack nguồn sẽ vượt ra ngoài kích thước cũ. Bốn lỗ vít cho phép gắn bo mạch vào các bề mặt trong một số trường hợp cần thiết. Khối lượng của bo mạch này là 0.0406Kg.

2.4. Bo mạch Arduino Nano.

2.4.1. Tổng quan.

Arduino Nano là bo mạch kích thước rất nhỏ, đầy đủ và thân thiện dựa trên ATmega328 (Arduino Nano 3.x) hoặc Atmega168 (Arduino Nano 2.x). Nó chỉ thiếu

một jack cắm điện DC, và làm việc với một cáp USB Mini-B thay. Nano được thiết kế và được sản xuất bởi Gravitech.



Hình 16. Arduino nano: (a) mặt trước, (b) mặt sau

➤ Sơ đồ mạch nguyên lý Schematic:

Đối với Arduino Nano 3.0 sử dụng chip ATmega328: Xem ở phần phụ lục 3: *Sơ đồ nguyên lý bo mạch Arduino Nano*. Hoặc download tại đường link: <http://arduino.cc/en/uploads/Main/ArduinoNano30Schematic.pdf>

2.4.2. Tóm tắt các thông số.

Bảng 5. Tóm tắt các thông số Arduino Nano

Chip xử lý	Atmel ATmega168 or ATmega328
Điện áp hoạt động (mức logic cao)	5 V
Điện áp vào (khuyến nghị)	7-12 V
Điện áp vào (giới hạn)	6-20 V
Số chân I/O digital	14 (6 chân đầu ra PWM)
Số chân đầu vào analog	8
Dòng Dc trên chân I/O	40 mA
Flash Memory	16 KB (ATmega168) hoặc 32 KB (ATmega328) với 2 KB sử dụng cho bootloader
SRAM	1 KB (ATmega168), 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168), 1 KB (ATmega328)
Tốc độ xung	16 MHz
Kích thước	0.73" x 1.70"

2.4.3. Nguồn cấp.

Arduino Nano có thể được cấp nguồn thông qua các kết nối mini-B USB, điện áp 6-20V không được kiểm soát nguồn điện bên ngoài (pin 30), hoặc 5V theo quy định cung cấp điện bên ngoài (pin 27). Nguồn điện sẽ tự động lựa chọn nguồn điện áp cao nhất. Các chip FTDI FT232RL trên Nano chỉ được cấp nguồn nếu bo mạch được hỗ trợ thông qua cổng USB. Kết quả là, khi cấp nguồn bên ngoài (không USB), đầu ra 3.3V (được cung cấp bởi các chip FTDI) là không có sẵn và đèn LED TX và RX sẽ nhấp nháy nếu chân số 0 hoặc 1 là ở mức cao.

2.4.4. Bộ nhớ.

Atmega168 có bộ nhớ flash 16 KB để lưu trữ code (trong đó có 2 KB được sử dụng cho bootloader); ATmega328 có 32 KB, (cũng với 2 KB được sử dụng cho bootloader). Atmega168 có 1 KB SRAM và 512 byte EEPROM (có thể được đọc và ghi bằng thư viện EEPROM); ATmega328 có 2 KB SRAM và 1 KB EEPROM.

2.4.5. Đầu vào và đầu ra.

Có 14 chân digital trên Nano có thể được sử dụng thiết lập đầu vào hoặc đầu ra, sử dụng các hàm pinMode(), digitalWrite(), và digitalRead(), chúng hoạt động ở điện áp 5 volts. Mỗi chân có dòng tối đa là 40 mA và nối với điện trở 20-50 kΩ. Ngoài ra, có một số chân chức năng chuyên biệt:

- Serial: 0 (RX) và 1 (TX). Được sử dụng để nhận (R) và truyền tải (TX) dữ liệu nối tiếp TTL.
- Chân ngắt ngoài: 2 và 3. Các chân có thể được cấu hình để kích hoạt ngắt khi ngắn mạch hoặc quá dòng. Xem hàm attachInterrupt() để biết chi tiết.
- PWM: 3, 5, 6, 9, 10 và 11. Cung cấp đầu ra PWM 8-bit sử dụng hàm analogWrite ().
- SPI: 10 (SS), 11 (Mosi), 12 (miso), 13(SCK). Các chân hỗ trợ truyền dẫn SPI.
- LED: 13. Có tích hợp đèn LED kết nối với chân digital 13. Mức điện áp cao LED sáng và ngược lại.

Nano có 8 đầu vào analog, độ phân giải bits (1024 giá trị khác nhau). Hoạt động ở điện áp 5V, có thể thay đổi điện áp hoạt động bằng việc sử dụng hàm analogReference (). Chân analog 6 và 7 không được sử dụng như chân digital. Ngoài ra, một số chân có chức năng chuyên biệt:

- I2C: 4 (SDA) và 5 (SCL). Hỗ trợ I2C (TWI) truyền thông bằng cách sử dụng thư viện Wire library.

Có một vài chân khác trên bo mạch:

- Aref. Điện áp tham chiếu cho các đầu vào tương tự. Được sử dụng với hàm `analogReference()`.
- Reset. mang dòng thấp để thiết lập lại vi điều khiển. Có thể nhấn nút reset khi gắn thêm các khối Shield trên bo mạch.

Xem thêm các ảnh xạ giữa Arduino chân và các port của Atmega168.

2.4.6. Truyền dẫn.

Arduino Nano có một số phương thức để giao tiếp với máy tính, Arduino khác, hoặc vi điều khiển khác. Atmega168 và ATmega328 cung cấp giao tiếp nối tiếp UART TTL (5V), có sẵn trên chân digital 0 (RX) và 1 (TX). FTDI FT232RL trên các kênh của bo mạch này nối tiếp qua cổng USB và chương trình điều khiển FTDI (kèm với phần mềm Arduino) cung cấp một cổng com ảo cho phần mềm trên máy tính. Phần mềm Arduino bao gồm một màn hình nối tiếp cho phép dữ liệu văn bản đơn giản được gửi đến và đi từ bo mạch Arduino. LED RX và TX trên bo mạch sẽ nhấp nháy khi dữ liệu đang được truyền qua chip FTDI và kết nối USB với máy tính (nhưng không cho giao tiếp nối tiếp trên các chân 0 và 1). Thư viện `SoftwareSerial` cho phép giao tiếp nối tiếp trên bất kỳ chân digital của Nano.

Atmega168 và ATmega328 cũng hỗ trợ truyền dẫn I2C (TWI) và SPI. Phần mềm Arduino bao gồm một thư viện `Wire` để đơn giản hóa việc sử dụng bus I2C. Để sử dụng giao tiếp SPI, phải xem datasheet của các chân Atmega168 hoặc ATmega328.

2.4.7. Lập trình.

Arduino Nano có thể được lập trình với phần mềm Arduino. Chọn "Arduino Diecimila, Duemilanove, hoặc Nano w/Atmega168" hoặc "Arduino Duemilanove hoặc Nano w/ATmega328". Từ chương trình chọn Tools → trên thanh menu (theo vi điều khiển trên bo mạch).

Atmega168 hoặc ATmega328 trên Arduino Nano đi kèm với một bộ bootloader Preburned cho phép tải code mới mà không cần sử dụng công cụ lập trình phần cứng bên ngoài. Truyền dẫn sử dụng giao thức STK500. Ta cũng có thể bỏ qua bộ bootloader và chương trình vi điều khiển thông qua ICSP header (In-Circuit Serial Programming)..

2.4.8. Chức năng tự động reset.

Thay vì reset vật lý trước khi tải code lên, Arduino Nano được thiết kế cho phép reset trên phần mềm thông qua kết nối với máy tính. Một trong những dòng điều khiển phần cứng (DTR) của FT232RL được kết nối với các đường dây của Atmega168 hoặc ATmega328 thông qua một tụ điện 100 nF. Khi dòng thấp thì thực hiện reset. Phần mềm Arduino sử dụng khả năng này để cho phép tải code lên bằng cách nhấn nút tải lên trong môi trường Arduino. Điều này có nghĩa là bootloader có thể thời gian chờ ngắn hơn.

Thiết lập này ứng dụng khác. Khi Nano được kết nối với máy tính chạy MacOS X hoặc Linux , reset mỗi lần kết nối được thực hiện giữa bo mạch và phần mềm (thông qua cổng USB). Trong nửa giây hoặc lâu hơn, bộ bootloader chạy trên Nano.

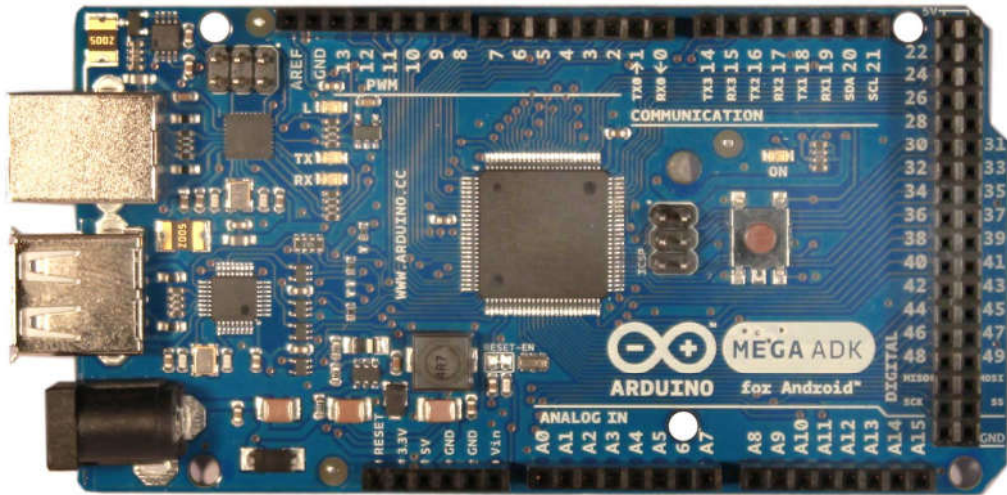
2.5. Bo mạch Arduino Mega ADK.

2.5.1. Tổng quan.

Arduino ADK là bo mạch vi xử lý hoạt động dựa trên chip ATmega2560. Bo mạch có giao tiếp USB để kết nối với điện thoại chạy hệ điều hành Android dựa IC MAX3421e. Bo mạch có 54 chân input/output digital (trong đó 15 chân được sử dụng cho điều chế xung đầu ra PWM), 16 đầu vào analog, 4 cổng phần cứng nối tiếp UARTs (hardware serial ports), dao động thạch anh 16MHz, kết nối USB, jack cắm nguồn, đầu ICSP, nút reset. ADK dựa trên chip Mega 2560.

Tương tự như bo mạch Mega 2560 và Uno, nó có tính năng ATmega8U2 lập trình như một bộ chuyển đổi USB nối tiếp.

Ở phiên bản sử dụng thứ 2, bo mạch MegaADK có điện trở nối đường 8U2 HWB xuống đất, giúp ta dễ dàng hơn trong việc đặt chế độ DFU.



Hình 17. Arduino Mega ADK

Phiên bản sửa đổi thứ 3 của bo mạch có các đặc tính mới dưới đây:

- Sơ đồ chân 1.0. Thêm chân SDA và SCL ở gần chân AREF và hai chân mới ở gần chân RESET, chân IOREF cho phép các mạch hỗ trợ nhận nguồn cấp từ bo mạch Arduino. Trong tương lai, bo mạch hỗ trợ sẽ tương thích với cả bo mạch chính và đều sử dụng AVR, bo mạch hoạt động ở điện áp 5V. Một số chân tạm thời không được kết nối và sẽ được dành cho các mục đích trong tương lai.
 - Chức năng RESET mạch mạnh hơn.
- Sơ đồ mạch nguyên lý Schematic.
Xem phần phụ lục 4: Sơ đồ nguyên lý bo mạch Arduino Mega ADK. Hoặc download tại đường link:
http://arduino.cc/en/uploads/Main/ADK_MEGA_2560-Rev2-sch.pdf

2.5.2. Tóm tắt các thông số.

Bảng 6. Tóm tắt các thông số Arduino Mega ADK

Chip xử lý	ATmega2560
Điện áp hoạt động	5V
Điện áp đầu vào (khuyến nghị)	7-12V
Điện áp đầu vào (giới hạn)	6-20V
Số chân I/O digital	54 (trong đó 15 chân dùng cho điều chế xung đầu ra PWM)
Số chân đầu vào analog	16
Dòng DC trên chân I/O	40 mA
Dòng DC trên chân 3.3V	50 mA
Flash Memory	256 KB, 8 KB sử dụng cho bootloader
SRAM	8 KB
EEPROM	4 KB
Tốc độ xung	16 MHz

2.5.3. Nguồn cấp.

Arduino ADK có thể được cấp nguồn thông qua kết nối USB hoặc được cấp nguồn từ bên ngoài. Nguồn điện được chọn tự động. Nguồn cấp bên ngoài (không phải USB) có thể được lấy qua bộ chuyển đổi AC-DC hoặc nguồn Pin. Các bộ chuyển đổi có thể được kết nối bằng cách cắm jack 2.1mm vào jack cắm điện của bo mạch. Khi sử dụng nguồn pin thì cần phải có các cực Vin và Gnd.

Lưu ý: Vì ADK là bo mạch chứa host-USB, điện thoại sẽ lấy ra năng lượng cần thiết từ bo mạch để sạc Pin. Khi ADK được cấp nguồn qua cổng USB, tổng dòng 500mA có sẵn cho điện thoại và bo mạch. Các nguồn cấp bên ngoài có thể cấp nguồn lên đến 1500mA: 750mA dành sẵn cho điện thoại và bo mạch ADK, 750mA còn lại được phân bổ cho các thiết bị truyền động và cảm biến gắn với bo mạch. Nguồn cấp 1.5A đang được sử dụng thông dụng nhất hiện nay.

Bo mạch có thể hoạt động với các nguồn cấp ngoài từ 5.5 đến 16 volt. Nếu nguồn ít nhất thường là 7V, tuy nhiên, các chân 5V có thể được cấp nguồn bé hơn 5V nhưng khi đó mạch có thể hoạt động không ổn định. Nếu sử dụng hơn 12V, bộ ổn áp bị nóng và phá hủy mạch, khuyến nghị nên sử dụng ở khoảng 7 đến 12 volts.

Nguồn cấp của các chân như sau:

- Vin. Điện áp đầu vào của bo mạch Arduino khi nó sử dụng nguồn cấp ngoài (khác với các nguồn cấp 5V từ kết nối USB và các nguồn điện theo quy định). Ta có thể cấp nguồn qua chân Vin hoặc cấp nguồn thông qua các jack cắm kết nối với chân này.
- 5V. Chân đầu ra được quy định là 5V. Bo mạch có thể có thể được cấp nguồn điện từ các jack (7-12V), kết nối USB (5V), hoặc chân Vin của bo mạch (7-12V). Cung cấp điện áp thông qua chân 5V hoặc 3.3V bỏ qua các khuyến cáo có thể gây hỏng mạch. Không nên sử dụng nó.
- 3.3V. Nguồn cấp 3.3V được quy định trên bo mạch. Dòng cấp tối đa là 50mA.
- GND Chân nối đất.
- IOREF. Chân này cấp điện áp tham chiếu cho vi điều khiển hoạt động. Bộ hỗ trợ cấu hình chuẩn đọc điện áp trên chân OIREF và lựa chọn nguồn cấp thích hợp hoặc kích hoạt dịch điện áp trên đầu ra để làm việc với các nguồn 5V hoặc 3.3V.

2.5.4. Đầu vào và đầu ra.

Mỗi chân trong số 50 chân digital trên ADK có thể được sử dụng như đầu vào hoặc đầu ra, sử dụng các hàm pinMode(), digitalWrite (), và digitalRead() để thiết lập chức năng. Chúng hoạt động ở điện áp 5 volts. Mỗi chân có thể cung cấp hoặc nhận tối đa dòng 40 mA và nối với điện trở có trở kháng 20-50 kΩ. Ngoài ra, một số chân có chức năng riêng biệt:

- Chân nối tiếp (Serial): 0 (RX) và 1 (TX); Serial 1: 19 (RX) và 18 (TX); Serial 2: 17 (RX) và 16 (TX); Serial 3: 15 (RX) và 14 (TX). Sử dụng để nhận và truyền dữ liệu TTL. Chân 0 và 1 cũng được kết nối với các chân tương ứng của ATmega8U2.
- Chân ngắt ngoài: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). Chân này có thể được thiết lập để thực hiện ngắt khi điện áp quá thấp hoặc thay đổi đột biến giá trị điện áp. Thường sử dụng hàm ngắt attachInterrupt().
- PWM: 2 đến 13 và 44 đến 46. Xung PWM có độ rộng là 8 bits. Khi điều chế xung đầu ra sử dụng hàm analogWrite().

- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Các chân này hỗ trợ truyền dẫn SPI sử dụng thư viện SPI.

- USB Host: MAX3421E. MAX3421E liên kết với Arduino và với bus SPI. Vì vậy, nó sử dụng các chân như sau:

Digital: 7 (RST), 50 (miso), 51 (Mosi), 52 (SCK). Lưu ý: Không sử dụng chân 7 làm đầu vào hoặc đầu ra vì nó được sử dụng trong liên kết với MAX3421E.

Không chia nhỏ trên các chân tiêu đề: PJ3 (GP_MAX), PJ6 (INT_MAX), PH7 (SS).

- LED: 13. Có một đèn LED được nối với chân 13. Khi điện áp ở mức cao LED sáng và ở mức thấp thì LED tắt.

- TWI: 20 (SDA) và 21 (SCL). Hỗ trợ truyền thông TWI sử dụng thư viện Wire.

Trên bo mạch Ethernet có 16 chân đầu vào tương tự, mỗi chân sử dụng 10 bits (tức là có 1024 giá trị khác nhau). Nguồn cấp cho các chân này là từ 0-5V, mặc dù nó có thể thay đổi phạm vi hoạt động của chúng bằng cách sử dụng chân AREF và hàm `analogReference()`. Ngoài ra, một số chân có chức năng chuyên biệt:

Còn có hai chân khác trên bo mạch là:

- AREF: Điện áp tham chiếu cho đầu vào tương tự. Sử dụng với hàm `analogReference()`.

- Reset. Được dùng để thiết lập lại vi điều khiển. Thường sử dụng nút reset để hỗ trợ các khối trên bo mạch.

2.5.5. Các thông số khác

➤ Bộ nhớ

ADK có 256 KB bộ nhớ flash để lưu trữ code (trong đó có 8 KB được sử dụng cho bộ nạp khởi động), 8 KB SRAM và 4 KB EEPROM (có thể được đọc và ghi bằng thư viện EEPROM).

➤ Truyền dẫn

Arduino ADK có một số thiết bị để giao tiếp với máy tính, Arduino khác hoặc vi điều khiển khác. ATmega2560 cung cấp bốn phần cứng UARTs cho TTL (5V) giao tiếp nối tiếp. ATmega8U2 trên các kênh bo mạch này nối tiếp qua cổng USB và cung cấp một cổng com ảo cho phần mềm trên máy tính (Windows sẽ cần

một tập tin.inf), nhưng OSX và Linux sẽ nhận biết bo mạch như một cổng COM tự động. Phần mềm Arduino bao gồm một màn hình nối tiếp cho phép dữ liệu văn bản đơn giản được gửi đến và đi từ bo mạch Arduino. LED RX và TX trên bo mạch sẽ nhấp nháy khi dữ liệu được truyền qua chip ATmega8U2/16U2 và kết nối USB với máy tính (không được kết nối qua các chân 0 và 1).

Thư viện SoftwareSerial cho phép giao tiếp nối tiếp trên bất kỳ chân digital nào của ADK.

ATmega2560 cũng hỗ trợ truyền thông TWI và SPI. Phần mềm Arduino bao gồm một thư viện Wire để đơn giản hóa việc sử dụng bus TWI. Với truyền dẫn SPI, sử dụng thư viện SPI library.

Giao diện USB-host được đưa ra bởi IC MAX3421E cho phép Arduino ADK kết nối và tương tác với bất kỳ loại thiết bị có cổng USB nào. Ví dụ, cho phép tương tác với nhiều loại điện thoại, điều khiển máy ảnh Canon, giao tiếp với bàn phím, chuột và các trò chơi điều khiển như Wiimote và PS3.

➤ *Lập trình*

Arduino ADK có thể được lập trình với phần mềm Arduino. Để biết chi tiết, xem tài liệu tham khảo và hướng dẫn. ATmega2560 trên Arduino ADK đi kèm với một bộ nạp khởi động preburned (giống như trên Mega 2560) cho phép tải lên code mới cho bo mạch mà không sử dụng lập trình phần cứng bên ngoài. Bo mạch giao tiếp bằng cách sử dụng giao thức ban đầu STK500v2.

Ta cũng có thể bỏ qua bộ nạp khởi động và chương trình vi điều khiển thông qua chân tiêu đề ICSP (In-Circuit Serial Programming). Mã nguồn ATmega8U2 firmware có sẵn trong kho Arduino. ATmega8U2 được nạp với một bộ nạp khởi động DFU và có thể được kích hoạt bằng cách:

- Trên bo mạch REV1: kết nối với các jumper hàn ở mặt sau của bo mạch và sau đó reset lại 8U2.
- Trên bo mạch Rev2 hoặc các bo mạch sau này: có một điện trở nối 8U2/16U2 HWB với đất, giúp ta dễ dàng hơn trong việc đặt chế độ DFU. Sau đó ta có thể sử dụng phần mềm của Atmel FLIP (Windows) hoặc lập trình DFU (Mac OS X và Linux) để tải một phần mềm mới. Hoặc ta có thể sử dụng chân tiêu đề ISP được lập trình bên ngoài (ghi đè lên các bộ nạp khởi động DFU).

➤ *Reset tự động*

Thay vì reset vật lý trước khi tải code lên, Arduino mega ADK được thiết kế cho phép reset trên phần mềm thông qua kết nối với máy tính. Một trong những

dòng điều khiển phần cứng (DTR) của ATmega8U2 được kết nối với các đường dây của ATmega2560 thông qua một tụ điện 100nF. Khi dòng thấp thì thực hiện reset. Phần mềm Arduino sử dụng khả năng này để cho phép tải code lên bằng cách nhấn nút tải lên trong môi trường Arduino. Điều này có nghĩa là bootloader có thể thời gian chờ ngắn hơn.

Thiết lập này có ý nghĩa khác. Khi ADK được kết nối với một trong hai máy tính chạy Mac OS X hoặc Linux, reset mỗi lần kết nối được thực hiện giữa bo mạch và phần mềm (thông qua cổng USB). Khoảng nửa giây sau hoặc lâu hơn, bộ bootloader chạy trên ADK. Trong khi nó được lập trình để bỏ qua dữ liệu bị thay đổi (tức là bất cứ điều gì ngoài việc tải code mới), nó sẽ ngăn chặn những byte đầu tiên của dữ liệu gửi đến bo mạch sau khi được kết nối. Nếu phác thảo chạy trên bo mạch nhận được một cấu hình thời gian hoặc các dữ liệu khác khi lần đầu tiên khởi động. Phải đảm bảo rằng, phần mềm mà ta kết nối phải chờ một vài giây sau khi kết nối và trước khi gửi dữ liệu.

➤ Bảo vệ quá dòng USB.

Arduino Uno có một bảng thiết lập được sử dụng để bảo vệ cổng USB của máy tính khi ngắn mạch và quá dòng. Mặc dù hầu hết các máy tính cung cấp chế độ bảo vệ nội bộ, nhưng còn có thêm các cầu chì, các cầu chì này có thêm một lớp bảo vệ. Nếu dòng cao hơn 500 mA được áp dụng cho các cổng USB, cầu chì sẽ tự động phá vỡ các kết nối khi ngắn mạch và quá dòng xảy ra.

➤ Đặc tính vật lý và Shield tương thích.

Chiều dài tối đa và chiều rộng tương của ADK PCB là 4 và 2,1 inch, với các kết nối USB và jack cắm điện vượt ra ngoài kích thước cũ. Ba lỗ vít cho phép gắn bo mạch vào một bề mặt trong một số trường hợp. Lưu ý rằng khoảng cách giữa chân số 7 và 8 là 160 mil (0,16 ").

ADK được thiết kế để tương thích với hầu hết shield được thiết kế cho Uno, Diecimila hoặc Duemilanove. Chân digital 0-13 (lân cận AREF và chân GND), đầu vào analog 0-5, chân header power và chân tiêu đề ICSP nằm ở các vị trí tương ứng. Hơn nữa phần chính UART (serial port) nằm trên các chân giống nhau (0 và 1), cũng như chân ngắt ngoài 0 và 1 (chân 2 và 3 tương ứng). SPI có sẵn thông qua các chân tiêu đề ICSP trên cả ADK và Duemilanove/Diecimila.

CƠ BẢN VỀ LẬP TRÌNH ARDUINO

3.1. Ngôn ngữ lập trình trên Arduino.

Có thể bạn đã quen lập trình trên PC, với những ngôn ngữ như C, C++, C#, Java, Python, Ruby. Nhưng bạn có biết là phần mềm trên PC chỉ chiếm khoảng 10% sản lượng phần mềm trên thị trường. 90% còn lại là code điều khiển tivi, máy giặt, điều hòa, tủ lạnh... tóm lại là tất cả các thiết bị điện tử xung quanh bạn. Đây cũng là một mảng theo tôi là khá thú vị. Lập trình theo hướng này được gọi là embedded computing, hay physical computing, tức là lập trình để con người tương tác với các thiết bị thực.

Để người thiết kế có thể nhanh chóng đưa ra được mẫu thể hiện ý tưởng của mình, rất cần phải có những platform để dễ dàng prototyping. Và một trong những platform đang được sử dụng rất nhiều trong prototyping là Arduino.

Arduino là một bo mạch xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ,... Điểm hấp dẫn ở Arduino với anh em lập trình là ngôn ngữ cực kì dễ học (giống C/C++), các ngoại vi trên bo mạch đều đã được chuẩn hóa, nên không cần biết nhiều về điện tử, chúng ta cũng có thể lập trình được những ứng dụng thú vị. Thêm nữa, vì Arduino là một platform đã được chuẩn hóa, nên đã có rất nhiều các bo mạch mở rộng (gọi là shield) để cắm chồng lên bo mạch Arduino, có thể hình dung nôm na là “library” của các ngôn ngữ lập trình. Ví dụ, muốn kết nối Internet thì có Ethernet shield, muốn điều khiển động cơ thì có Motor shield, muốn kết nối nhận tin nhắn thì có GSM shield,... Rất đơn giản, và ta chỉ phải tập trung vào việc “lắp ghép” các thành phần này và sáng tạo ra các ứng dụng cần thiết.

Có thể kể ở đây một số ứng dụng hay ho của Arduino:

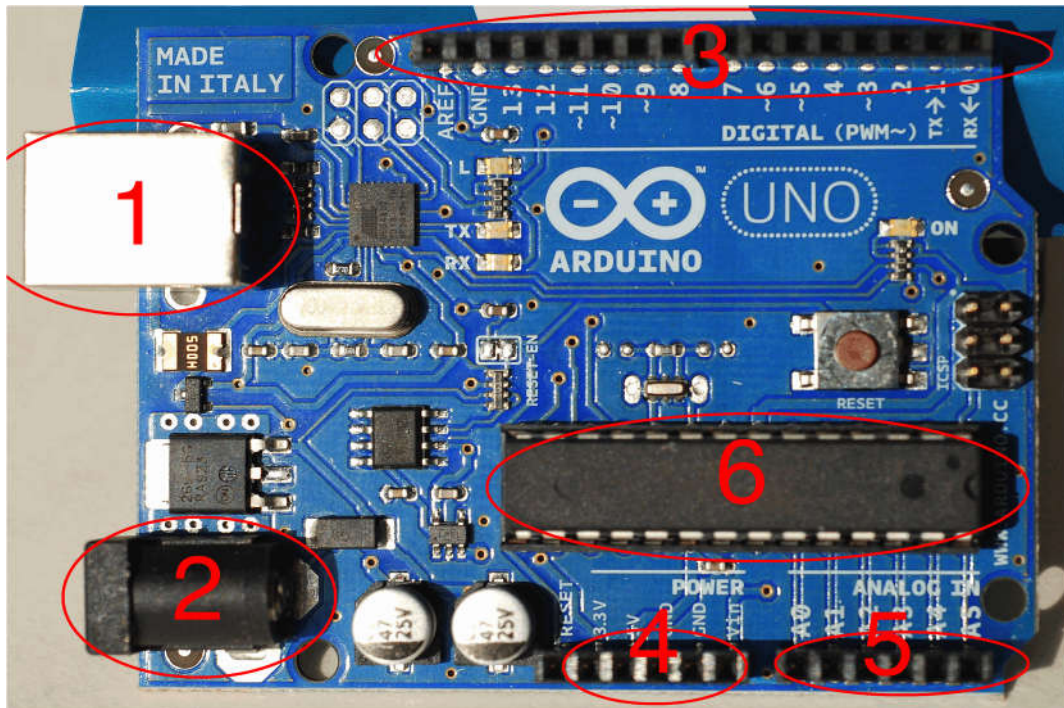
- Robot: Arduino được dùng để làm bộ xử lý trung tâm của rất nhiều loại robot. Đó là nhờ vào khả năng đọc các thiết bị cảm biến, điều khiển động cơ,... của Arduino.
- Game tương tác: chúng ta có thể dùng Arduino để tương tác với Joystick, màn hình,... để chơi các trò như Tetrix, phá gạch, Mario...

- Máy bay không người lái.
- Mô phỏng Ipod.
- Và nhiều nhiều ứng dụng khác nữa...

3.2. Điều kiện cần thiết để lập trình.

Như vậy, tuy là một bo mạch nhỏ như thế, Arduino có thể dùng vào rất nhiều ứng dụng thú vị khác nhau. Vậy để phát triển ứng dụng dựa trên Arduino, ta chỉ cần IDE phát triển (download ở <http://arduino.cc>), một dây kết nối USB loại A-B, và một bo mạch Arduino là có thể bắt đầu được rồi.

Ngôn ngữ lập trình của Arduino dựa trên ngôn ngữ lập trình Wiring cho phần cứng. Nếu quá quen thuộc với ngôn ngữ C/C++, như vậy việc viết code Wiring là rất dễ dàng. Cộng thêm các tài liệu trên website, có khá nhiều các library viết sẵn để điều khiển ngoại vi: LCD, sensor, motor... nên việc cần làm chỉ là kết hợp chúng với nhau để tạo ứng dụng cho riêng của mình.



Hình 18. Bo mạch Arduino Uno

Hình trên là cận cảnh bo mạch Arduino Uno. Đối với chúng ta lập trình cho Arduino thì trước tiên quan tâm những thành phần được đánh số ở trên:

1. Cổng USB (loại B): đây là cổng giao tiếp để ta upload code từ PC lên vi điều khiển. Đồng thời nó cũng là giao tiếp serial để truyền dữ liệu giữa vi điều khiển với máy tính.
2. Jack nguồn: để chạy Arduino thì có thể lấy nguồn từ cổng USB ở trên, nhưng không phải lúc nào cũng có thể cắm với máy tính được. Lúc đó, ta cần một nguồn 9V đến 12V.
3. Hàng Header: đánh số từ 0 đến 12 là hàng digital pin, nhận vào hoặc xuất ra các tín hiệu số. Ngoài ra có một pin đất (GND) và pin điện áp tham chiếu (AREF).
4. Hàng header thứ hai: chủ yếu liên quan đến điện áp đất, nguồn.
5. Hàng header thứ ba: các chân để nhận vào hoặc xuất ra các tín hiệu analog. Ví dụ như đọc thông tin của các thiết bị cảm biến.
6. Vi điều khiển AVR: đây là bộ xử lý trung tâm của toàn bo mạch. Với mỗi mẫu Arduino khác nhau thì con chip này khác nhau. Ở con Arduino Uno này thì sử dụng ATmega328.

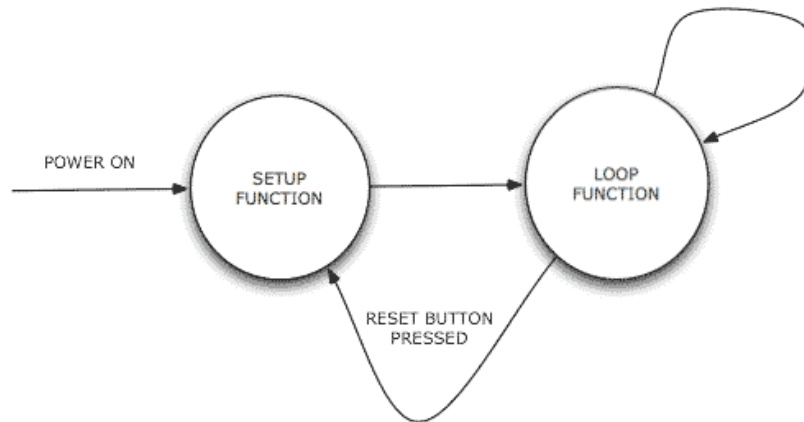
3.3. Cấu trúc của một chương trình lập trình.

Mỗi chương trình Arduino được chia làm 2 chức năng:

Đầu tiên là cài đặt chức năng. Việc này giúp khởi tạo và dùng để Arduino biết sẽ kết nối đến thiết bị nào, cũng như khởi tạo biến mà ta cần trong chương trình.

Thứ hai là vòng lặp. Đây là nền tảng của mỗi chương trình Arduino. Khi Arduino hoạt động, sau khi chức năng cài đặt hoàn tất, vòng lặp sẽ chạy tất cả các code, sau đó thực hiện lại đến khi nào mất nguồn hoặc bị reset. Quãng thời gian để hoàn thành dựa trên code được nạp vào. Ta có thể viết code để chạy vài giờ nếu không muốn lặp quyết định.

Hàm setup sẽ chứa tất cả các khai báo ban đầu của chương trình (vì nó luôn được chạy đầu tiên và chỉ chạy 1 lần). Hàm loop sẽ là nơi thực hiện các tác vụ của người thiết lập, nơi các lệnh được lặp đi lặp lại vô hạn. Ta cần phải vạch định rõ chương trình hoạt động như thế nào và hoạt động ra sao để chọn vào đúng hàm.



Hình 19. Cấu trúc chương trình Arduino.

- Tập các lệnh thông dụng trong lập trình.

Chương trình Arduino có thể được chia thành ba phần chính: cấu trúc, giá trị (các biến và các hằng số), và các hàm. Để biết về các tập lệnh được sử dụng. Vào đường link sau: <http://arduino.cc/en/Reference/HomePage>

3.4. Cài đặt Driver và phần mềm lập trình Arduino trên Windows.

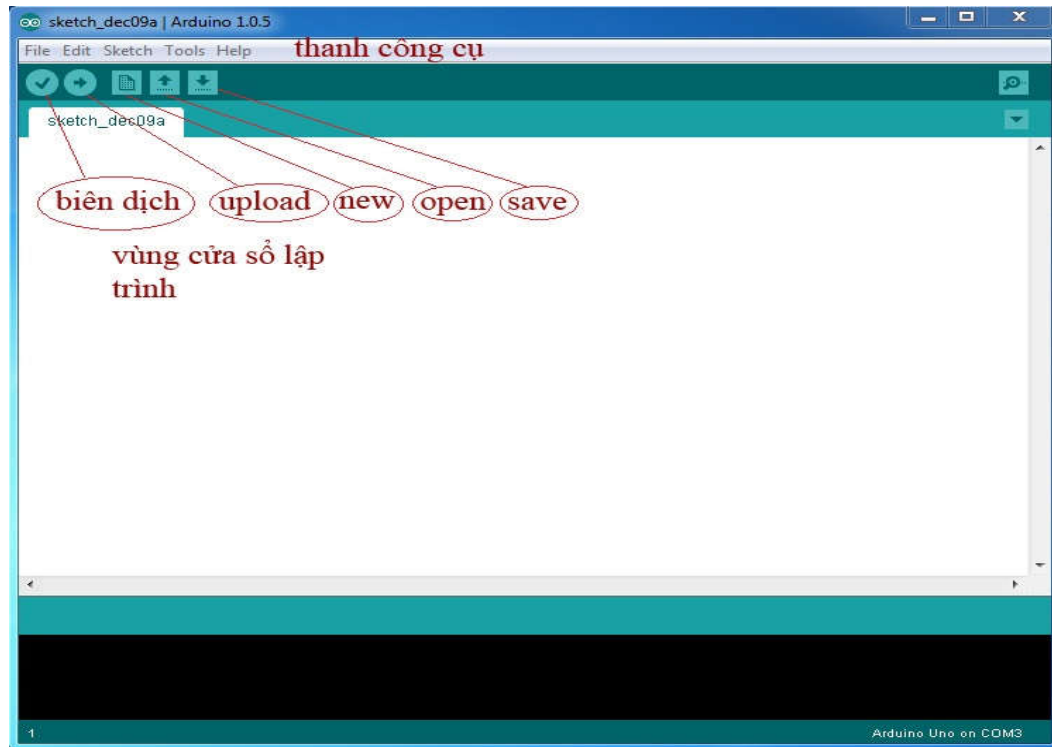
Trong phần này đảm bảo mọi thứ được thiết lập và chuẩn bị sẵn sàng cho việc lập trình sau quá trình kết nối phần cứng. Ta sẽ kiểm tra xem Arduino có làm việc và có tương thích với máy tính đang sử dụng hay không. Những đối tượng cần có trong phần này là: bo mạch Arduino, cáp nối USB chuẩn có chiều dài vừa đủ, Adapter 9V với jack cắm 2.1mm, một số chân dăm cần thiết.

Tùy thuộc vào bo mạch Arduino sử dụng mà ta sẽ lựa chọn các môi trường lập trình khác nhau: Windows, Mac OS hoặc Linux. Vì hiện nay môi trường lập trình trên windows thân thiện với người dùng nên trong phần này ta sẽ trình bày về các thiết lập trên môi trường Windows. Ta sẽ thực hiện theo các bước sau:

Bước 1: Tải phần mềm Arduino.

- Vào trang web của Arduino www.arduino.cc để tải phần mềm, trong hướng dẫn này ta sử dụng phiên bản phần mềm là Arduino 1.0.5 hoặc dùng link download: <http://arduino.googlecode.com/files/arduino-1.0.5-windows.zip>

- Từ file .zip tên “arduino-1.0.1-windows”, tiến hành giải nén và đặt trong thư mục bất kỳ. Trong thư mục này click icon “arduino” để chạy chương trình (không cần cài đặt).
- Sau cài đặt giao diện chương trình có dạng như sau:



Hình 20. Giao diện chương trình Arduino IDE

Bước 2: Cài Driver cho dây cáp FTDI.

- Để giao tiếp bo mạch với Arduino với máy tính cần có dây cáp FTDI như hình 20, là dây cáp USB nối vào bo mạch FTDI Adapter. Chức năng của dây cáp FTDI này là chuyển cổng USB thành cổng COM.



Hình 21. Dây cáp FTDI

- Dây cáp FTDI là một dụng cụ USB nên đầu tiên cắm vào cổng USB, máy tính sẽ yêu cầu cài Driver tên “Driver_FT232RL.zip” từ mục “Tài Liệu” trong trang web sản phẩm. Thực hiện các bước cài drivers theo hướng dẫn của máy tính. Nếu cài driver thành công máy tính sẽ xuất hiện một cổng COM mới, để kiểm tra cần mở chương trình Arduino, chọn **Tools** → **Serial Port** thì sẽ xuất hiện cổng COM mới. Lưu ý, cũng cần click chọn cổng COM này thì chương trình Arduino mới có thể nạp code xuống bo mạch Arduino được.
- Dưới đây là các bước cài driver cho FTDI trên hệ điều hành Windows.

Khi cắm bo mạch Arduino, ta có thể nghe thấy tiếng kết nối từ máy tính, xem góc dưới bên phải màn hình thấy thông báo **Found New Hardware FT232R USB UART**.



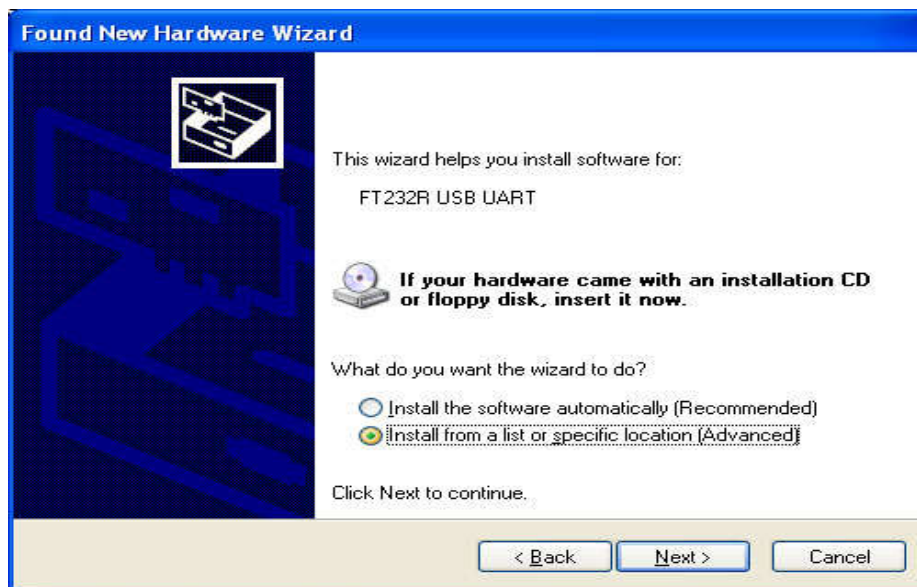
Hình 22. Thông báo nhận phần cứng từ windows

Sau đó vài giây, các định hướng cài đặt phần cứng bắt đầu. Lựa chọn “**No not this time**” và click **Next**.



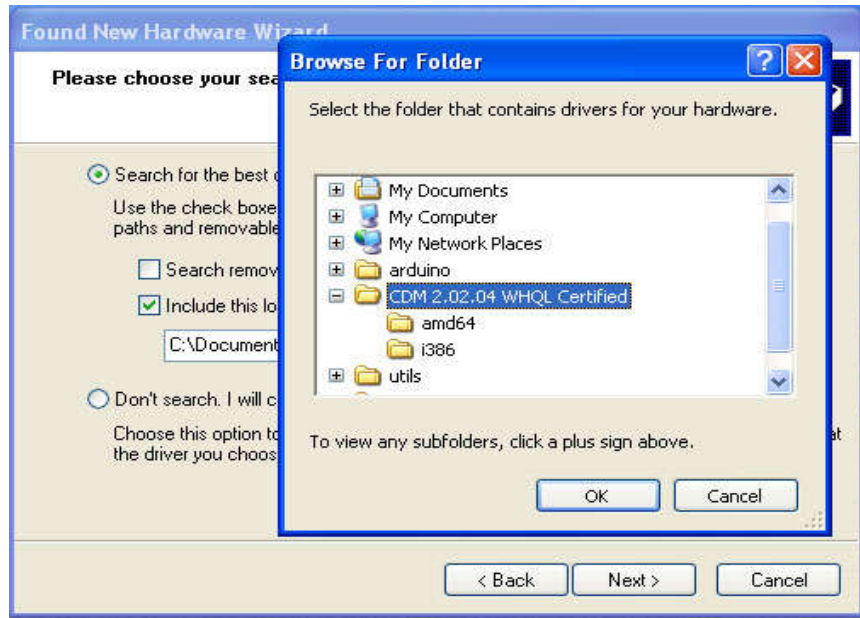
Hình 23. Xác nhận cài đặt 1

Giao diện phần tiếp theo, lựa chọn **Install from a list or specific location**



Hình 24. Xác nhận cài đặt 2

Tại cửa sổ tiếp theo ta lựa chọn **Include this location** và browse đến folder chứa driver vừa download. Lựa chọn folder và click **OK**



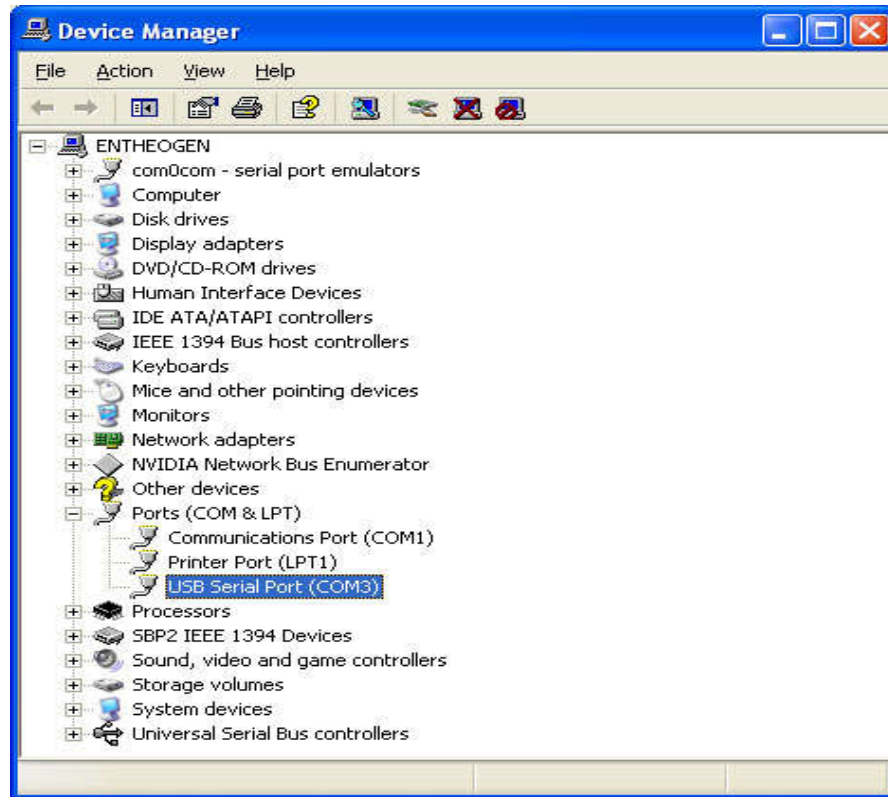
Hình 25. Duyệt Folder

Sau khi thực hiện bước trên. Click Finish. Khi quá trình cài đặt hoàn tất, ta cần khởi động lại máy tính.



Hình 26. Thông báo xác nhận

Khi máy tính khởi động xong Arduino được cắm vào máy tính. Đi đến **mục Device Manager** (từ thanh **Start Menu**, lựa chọn **Settings** -> **Control Panel**. Click vào **System** và lựa chọn tab **Hardware**. Sau đó click vào nút **Device Manager**)



Hình 27. Kiểm tra cổng được kết nối

Từ hình trên ta thấy tên cổng COM Port được kết nối trong thiết lập này là COM3.

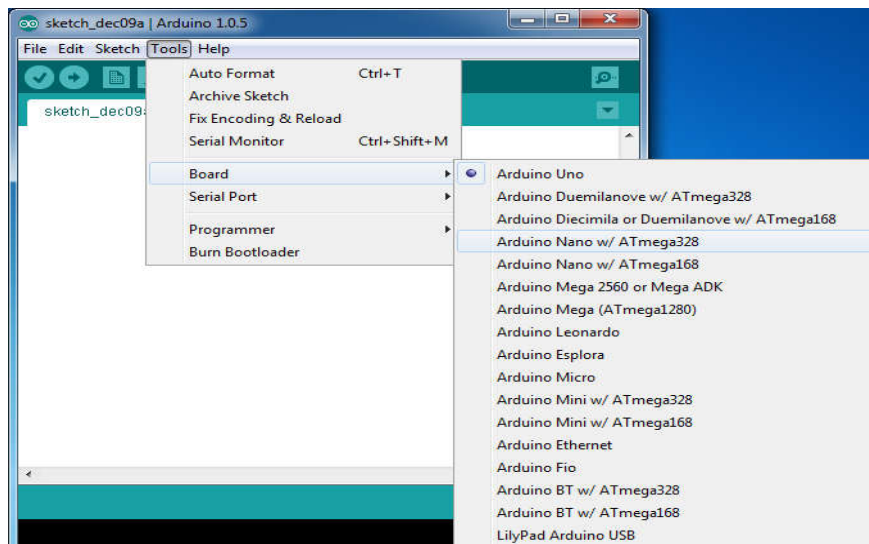
Sau khi thiết lập cổng COM như trên, cắm đầu còn lại của dây cáp FTDI vào trạm nối FTDI Adapter của bo mạch Arduino như hình 27. Lưu ý quan sát dấu “chấm tròn màu trắng” để cắm đúng chiều.



Hình 28. Nối cáp FTDI vào bo mạch Arduino

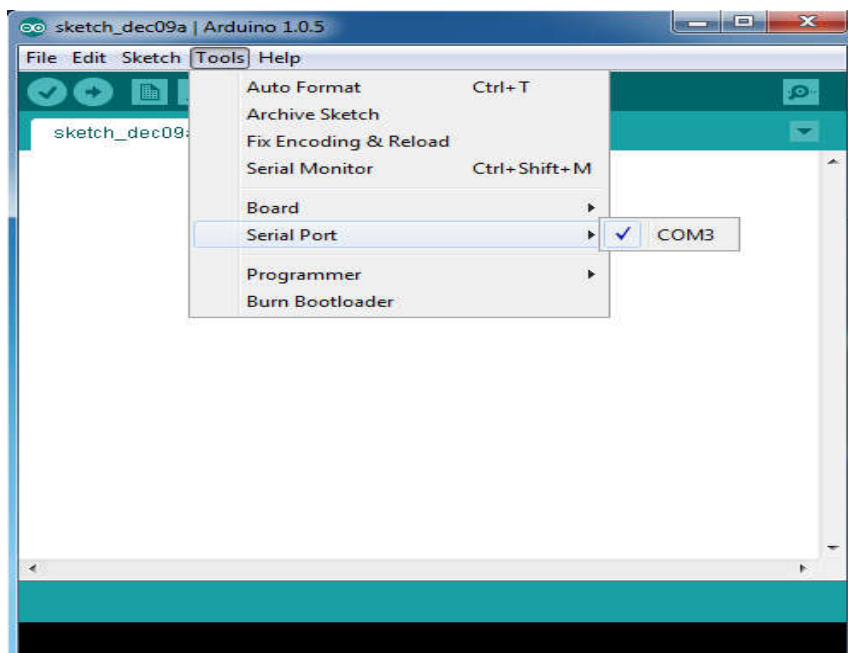
Bước 3: Thực hiện các thao tác trên phần mềm IDE Arduino 1.0.5

- Chọn bo mạch lập trình: Vào **Tools** → **Board** và click chọn dòng Arduino mà mình muốn lập trình.



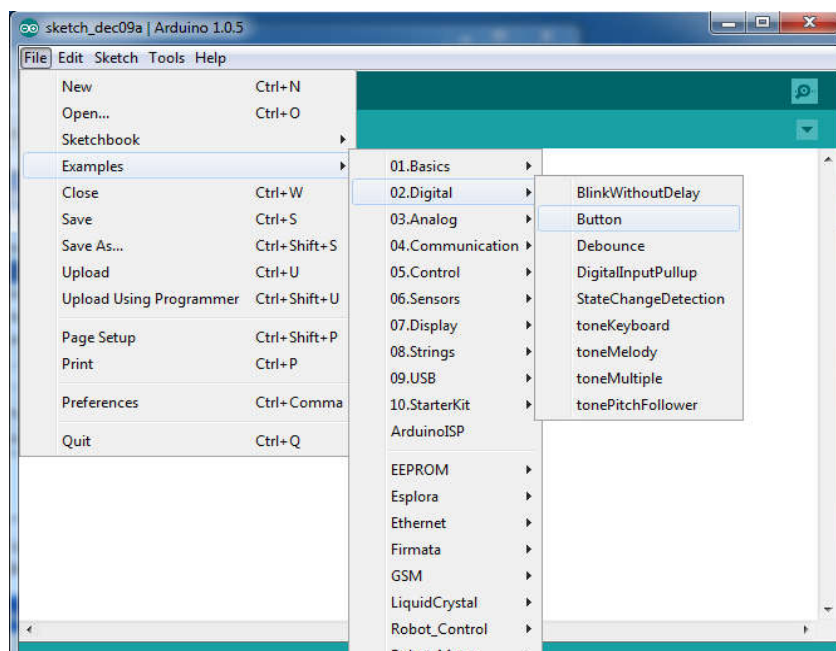
Hình 29. Lựa chọn bo mạch cần lập trình

- Lựa chọn cổng: để xem cổng COM Port mà bo mạch đang kết nối với máy tính ta vào **Tools** → **Serial Port**. Trong trường hợp mà ta thực hiện là COM3.



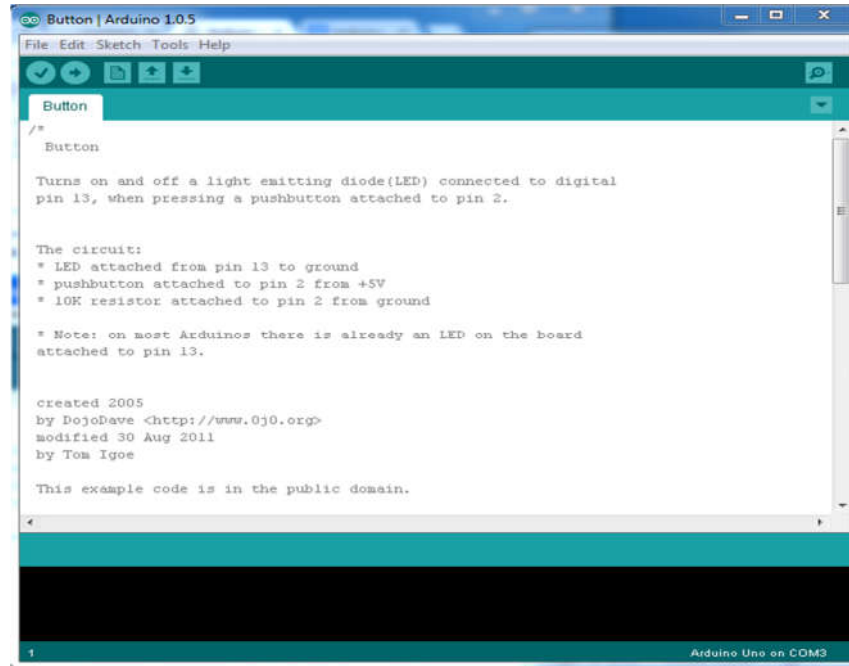
Hình 30. Kiểm tra cổng kết nối với máy tính

- Mở một bản thảo chạy thử trên chương trình Arduino: vào **File** → **Example...** Các lựa chọn trong hình dưới đây mô tả ví dụ về “Button”.




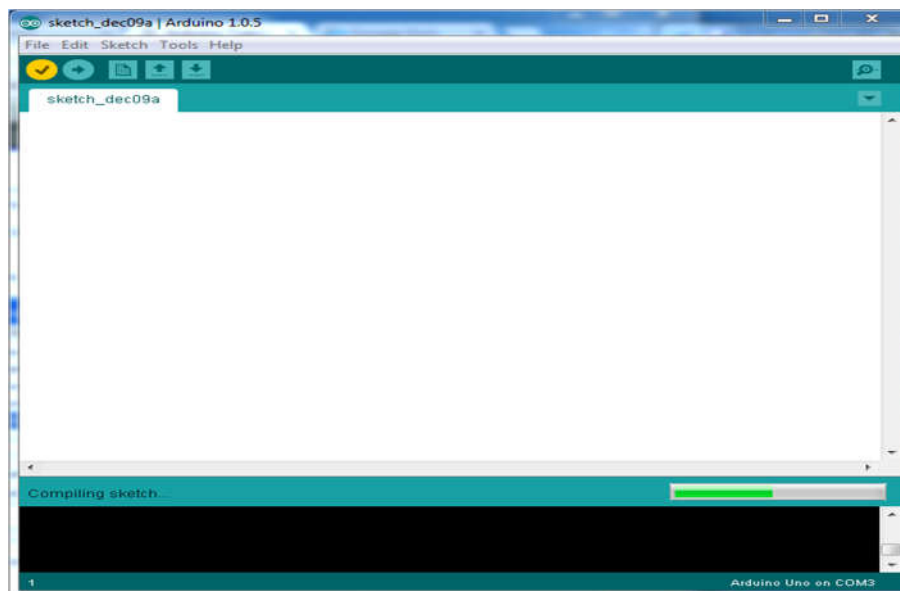
Hình 31. Mở bản thảo chạy thử trên Arduino

Sau khi mở ví dụ lên sẽ xuất hiện một cửa sổ mới có dạng như sau:




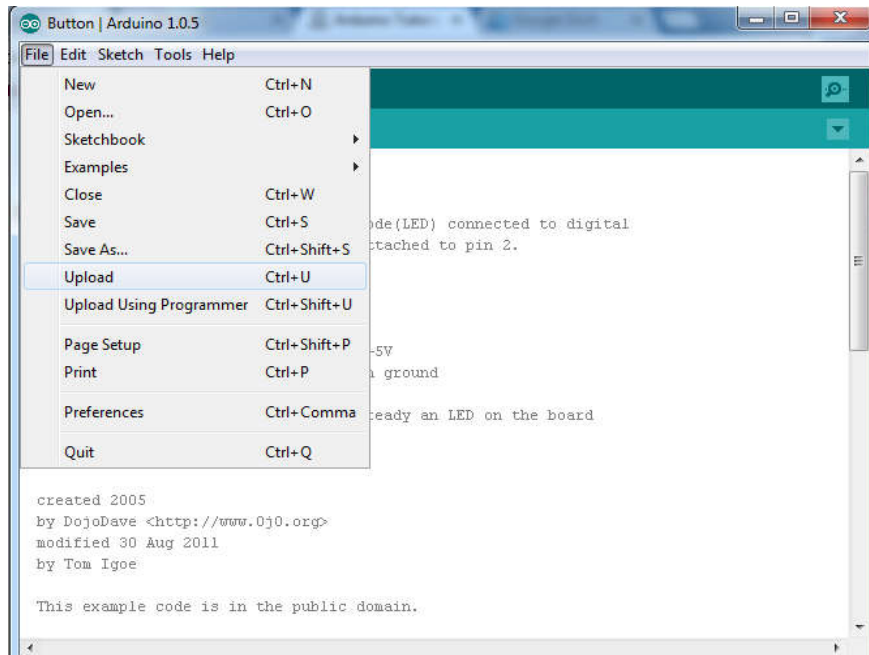
Hình 32. Cửa sổ sketch vừa được mở

- *Biên dịch chương trình*: Bước đầu tiên để có được bản phác thảo đã sẵn sàng để chuyển sang Arduino hay không là phải biên dịch nó. Có nghĩa là ta kiểm tra lỗi chương trình và thực hiện chỉnh sửa để có sự tương thích với phần cứng Arduino. Vào **Sketch** → **Verify/Compile** hoặc ấn **Ctrl+R** hoặc nhấn biểu tượng .



Hình 33. Biên dịch chương trình

- *Upload code lên bo mạch:* Sau khi biên dịch chương trình thành công. Ta tiến hành ấn nút Reset trên bo mạch, để thực hiện upload code lên bo mạch. Lựa chọn **File** → **Upload** hoặc ấn tổ hợp phím Ctrl+U hoặc ấn biểu tượng .



Hình 34. Upload code lên bo mạch

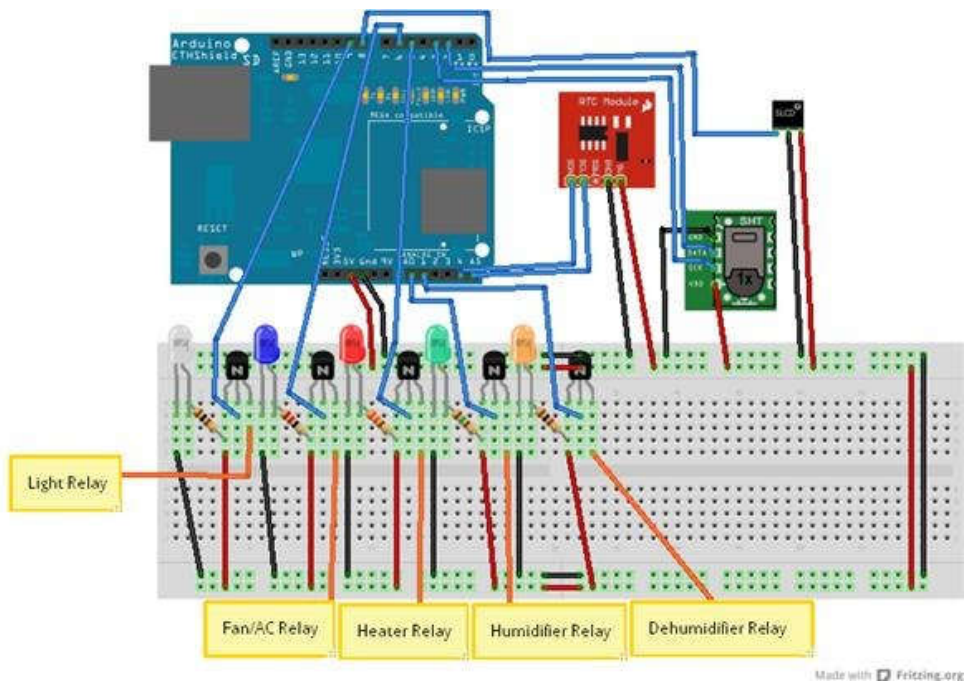
Khi quá trình upload dữ liệu được thực hiện LED nhỏ bắt đầu nhấp nháy.

3.5. Arduino Fritzing

Để thực hiện mô phỏng, tạo schematic, layout các mạch thông thường nói chung và bo mạch Arduino với các thiết bị ngoại vi nói riêng ta sử dụng phần mềm mô phỏng thiết kế Fritzing. Fritzing là phần mềm thiết kế PCB mã nguồn mở tương thích với môi trường Windows, Mac và Linux. Nó có giao diện đơn giản và dễ sử dụng. Để làm việc với phần mềm này, ta thực hiện các bước sau:

Bước 1: Tải và cài đặt Fritzing.

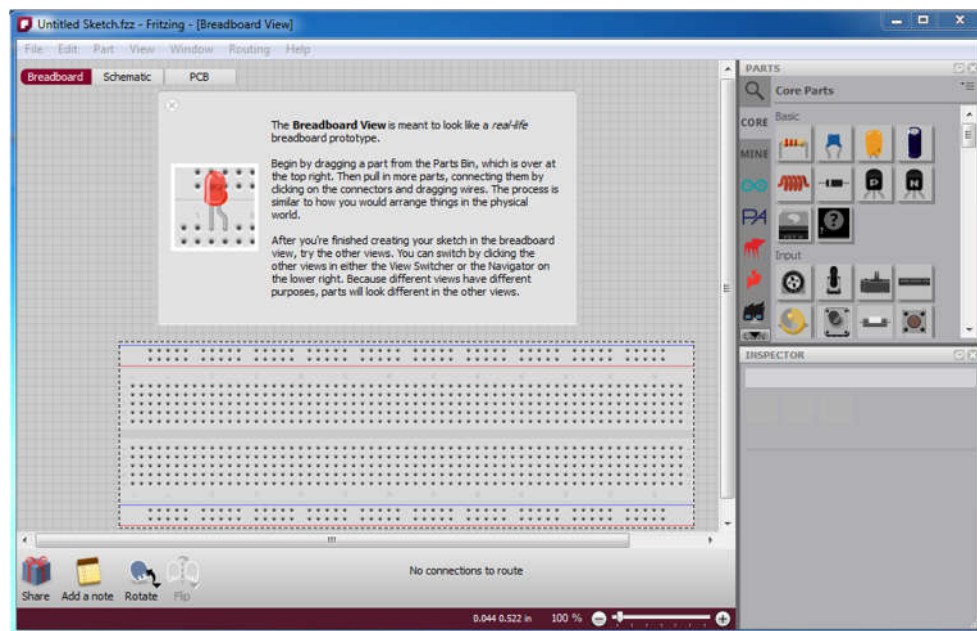
Vào [Fritzing download page](#) và lựa chọn hệ thống mà bạn đang sử dụng. Làm theo hướng dẫn trên trang này để cài đặt vào máy tính. Khi cài đặt Fritzing, ta sẽ có các thành phần thư viện, dây, điện trở, nút...Ngoài ra các thành phần như bo mạch Arduino và cảm biến cũng có thể dễ dàng thêm vào thiết kế đang thực hiện.



Hình 35. Ghép Arduino với breadboard và các linh kiện khác bằng Fritzing

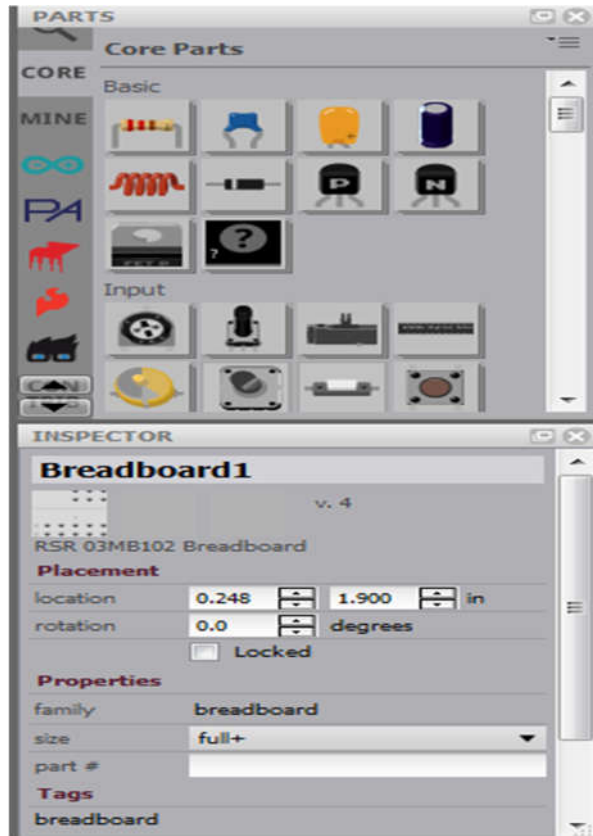
Bước 2: Bắt đầu thực hiện project đơn giản.

Khi bắt đầu mở Fritzing project, ta sẽ bắt đầu thực hiện với breadboard như hình dưới đây:



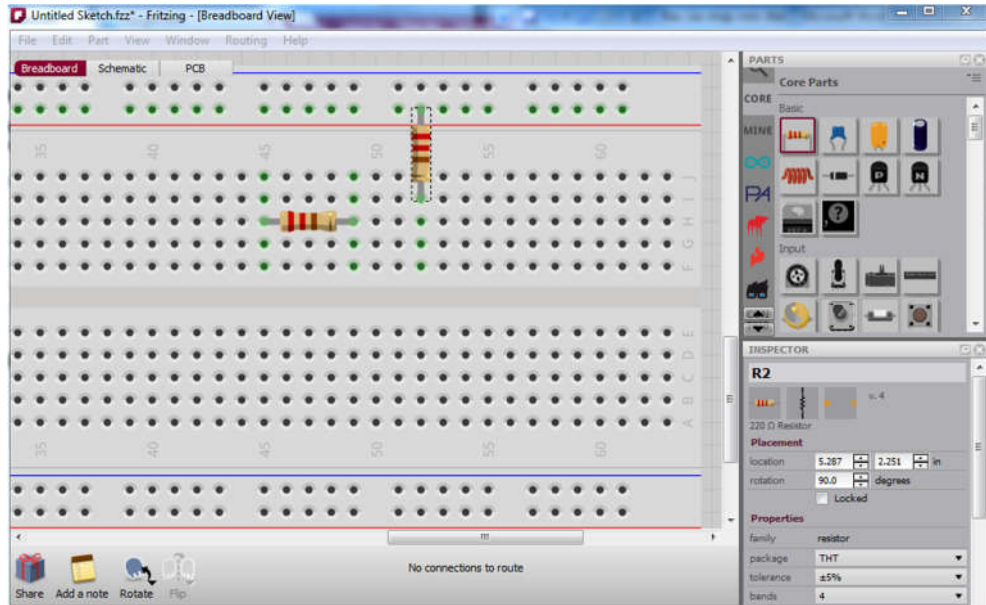
Hình 36. Giao diện phần mềm Fritzing

Phía bên phải của màn hình là thanh menu với tất cả các thành phần và cài đặt. Nếu các thành phần này là tùy chỉnh, nửa dưới của thanh công cụ sẽ hiển thị các tùy chọn tùy chỉnh có sẵn cho sẵn cho một thành phần cụ thể.



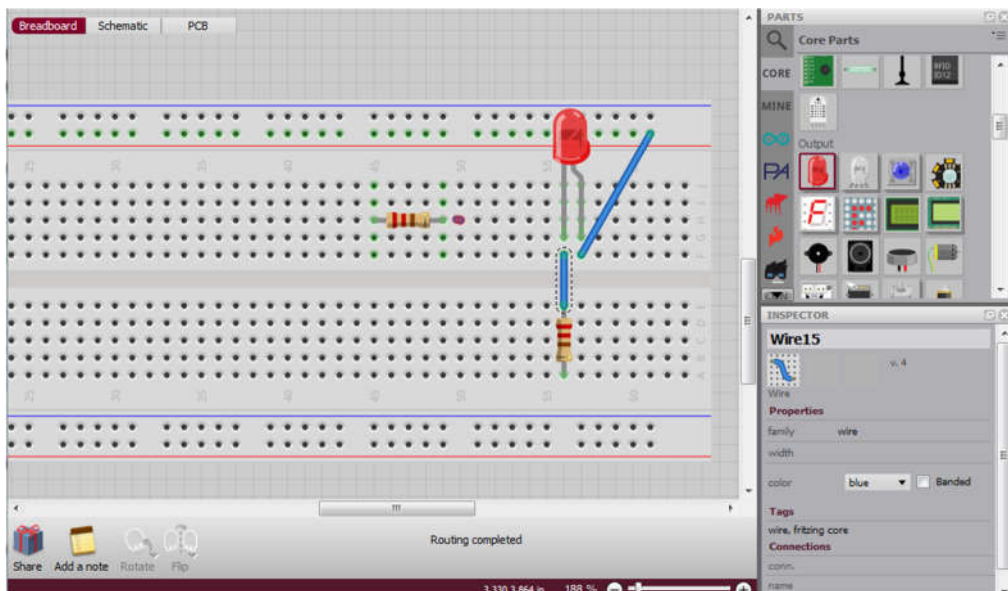
Hình 37. Cửa sổ lựa chọn và cài đặt component

Để đặt các thành phần này trên Breadboard. Ta thực hiện chọn và kéo các thành phần này vào khu vực làm việc như hình dưới. Hộp văn bản giới thiệu trên Breadboard sẽ biến mất khi ta đặt thành phần đầu tiên. Ví dụ: kéo thành phần điện trở xuống breadboard để các chân được kết nối với cột trên bảng mạch. Khi các chân được kết nối với bảng mạch, toàn bộ cột hoặc hàng sẽ có màu xanh lá cây, khi vực màu xanh lá chỉ ra kết nối điện giữa các lỗ breadboard.



Hình 38. Đặt linh kiện lên Breadboard

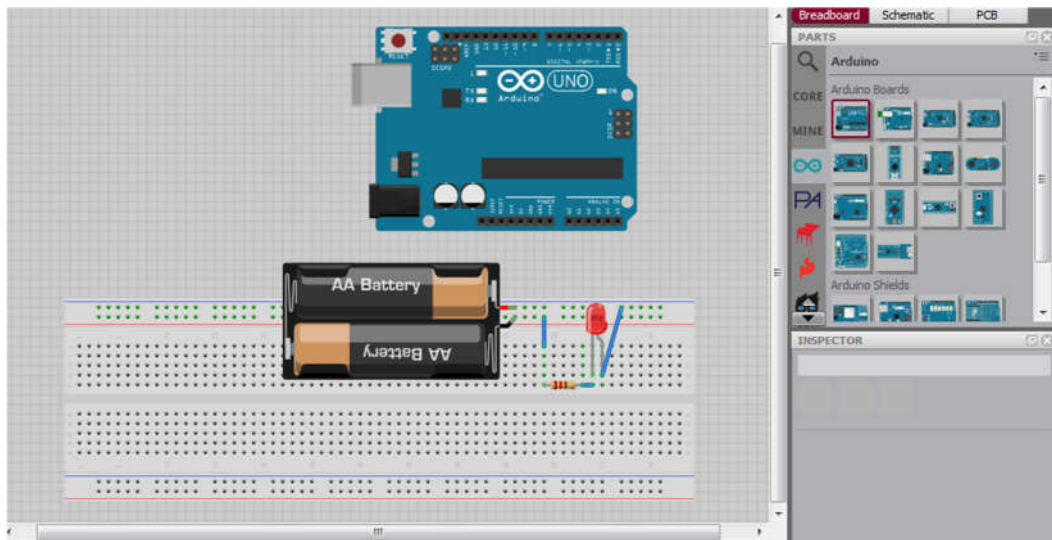
Để nối dây giữa các thành phần, ta dùng chuột chỉ vào vị trí cần nối dây đầu tiên và thực hiện kéo đến vị trí cần nối dây thứ 2.



Hình 39. Nối dây linh kiện

Để hoàn thành việc thiết kế, chúng ta cần thêm nguồn cấp. Click và kéo các thành phần nguồn ra khỏi thanh công cụ bên phải vào khu vực làm việc. Vị

trí các dây dẫn điện như hình dưới. Ta có thể điều chỉnh các vị trí đặt nguồn cách nối dây sao cho dễ quan sát nhất.

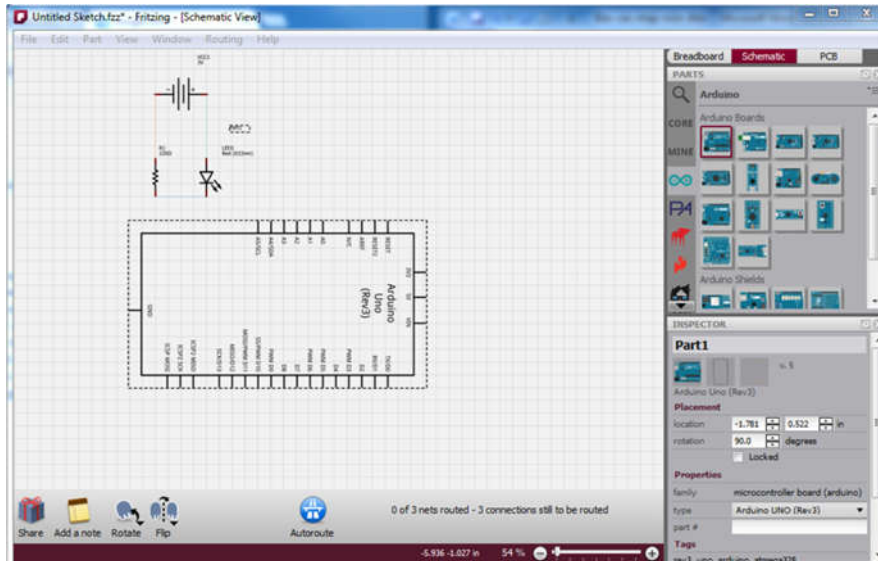


Hình 40. Cấp nguồn

Khi hoàn thành việc thiết kế toàn bộ một mạch chỉ với cách kéo và thả một vài thành phần! Các tính năng breadboard không tìm thấy trong bất kỳ phần mềm thiết kế PCB khác. Nó thực sự làm cho toàn bộ quá trình thực hiện dễ dàng hơn để có thể nhìn thấy hình ảnh trực quan giống như trên thiết kế mạch thật.

Bước 4: Schematic

Để thực hiện mạch sơ đồ nguyên lý của mạch vừa thực hiện, phần mềm Fritzing sẽ hỗ trợ tạo một sơ đồ kết nối mạch chính xác. Chọn nút Schematic phía trên bên phải cửa sổ làm việc. Khi đó sẽ xuất hiện sơ đồ vừa thực hiện. Fritzing đảm bảo tất cả các kết nối là chính xác, ta có thể tiếp tục kéo thả, xoay linh kiện và điều chỉnh nối dây trên sơ đồ này để có được cách sắp xếp hợp lý.

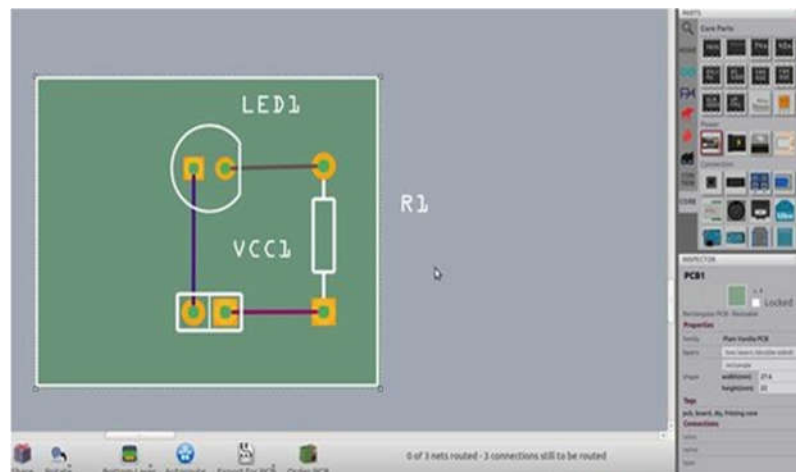


Hình 41. Cửa sổ schematic


Bước 5: Thực hiện mạch Layout

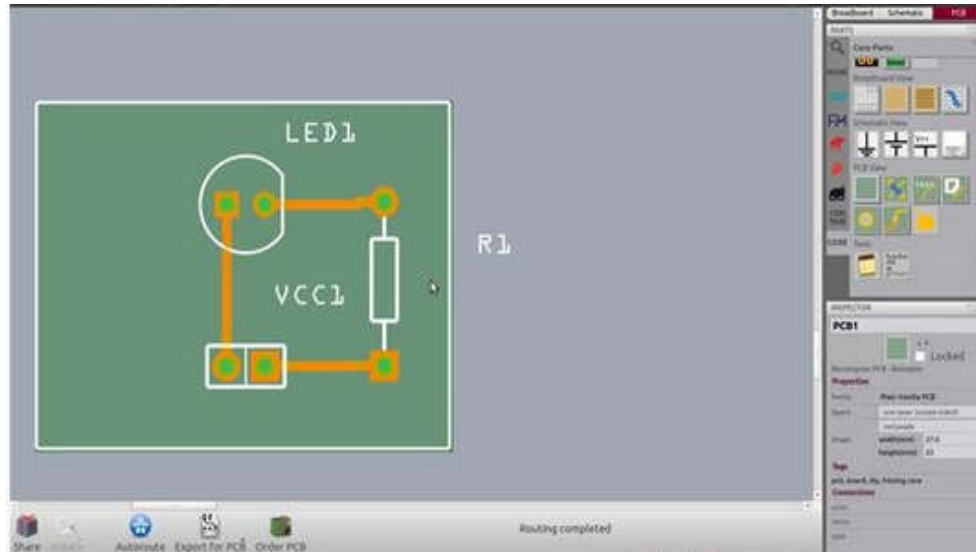
Để thực hiện layout cho mạch đơn giản gồm: nguồn pin, LED, trở như trên. Ta ấn sang tab PCB trên góc bên phải màn hình.

Thu nhỏ thanh trượt phía dưới màn hình để xem được tất cả các linh kiện. Thay đổi vị trí, kéo, thả và xoay linh kiện để có cách sắp xếp hợp lý. Vì mục đích của việc thiết kế bảng mạch là làm cho mạch tốn ít không gian nhất có thể, ta sẽ thu nhỏ bảng màu xanh lá cây xuống đến khi có đủ không gian sắp xếp các thành phần. Để thu nhỏ bo mạch chỉ cần nhấp vào góc bất kỳ và kéo về phía tâm của bo mạch.



Hình 42. Cửa sổ layout

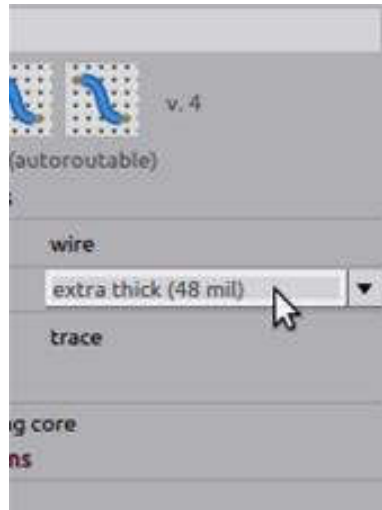
Sau khi lựa chọn được kích thước băng mạch màu xanh phù hợp ở phía góc dưới bên phải màn hình. Ta kích vào nút Autoroute  ở phía trung tâm cửa sổ làm việc. Các vết đồng sẽ thay thế các đường mảnh lúc trước.



Hình 43. Đồ đồng dây nối

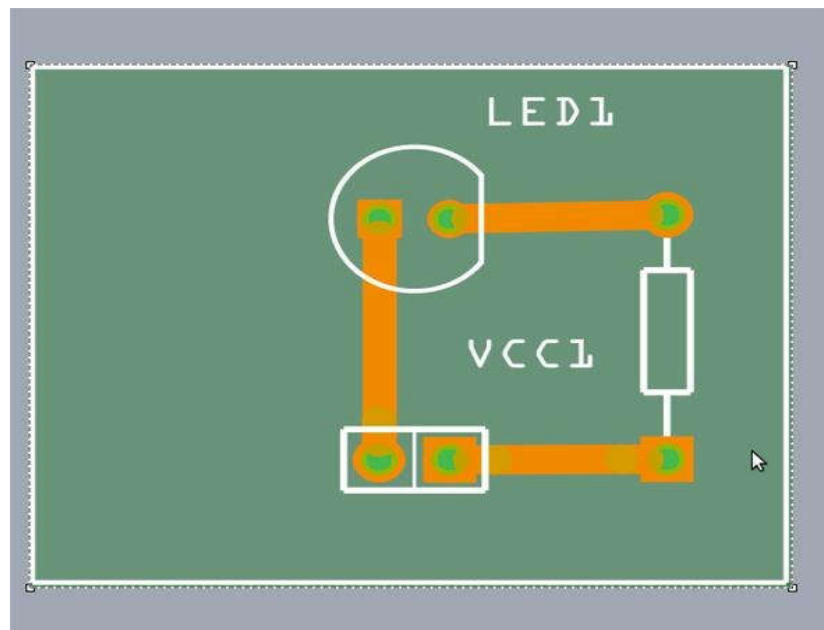
Bước 6: Chỉ sửa và làm sạch bo mạch

Sau khi thực hiện nối dây đồng tự động, trên mạch sẽ xuất hiện một số đường nối không chuẩn xác. Các autoroute tạo ra một số điểm uốn cong không cần thiết trên đường dây. Đưa chuột có vòng tròn màu cam nhỏ đến vị trí cần chỉnh trên dây trên dây. Khi bendpoint chuyển màu xanh, nhấp chuột phải và chọn "remove bendpoint". Chọn dây và đưa chuột để nửa dưới của thanh công cụ phía bên phải của màn hình. Có tùy chọn để điều chỉnh độ rộng của vết đồng trên bo mạch. Đối với các thiết kế phức tạp, vết đồng phải nhỏ để tiết kiệm chi phí và không gian. Chế tạo PCB luôn tạo ra các vết đồng nhỏ à chính xác.



Hình 44. Lựa chọn kích thước dây

Lưu ý: các vết đồng cũng phải đủ lớn để tránh các hiện tượng xấu khi mạch hoạt động.

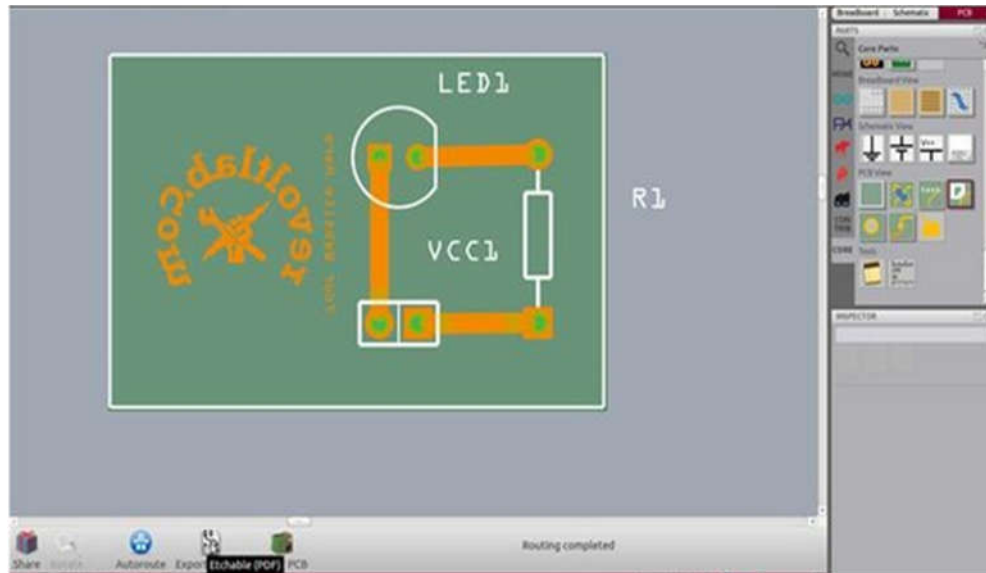


Hình 45. Kết quả sau điều chỉnh

Bước 7: tùy chỉnh bo mạch

Trong phần này, các văn bản và hình ảnh có thể dễ dàng cho vào project Fritzing. Trên thanh công cụ di chuyển cuộn chuột xuống tiêu đề cài đặt “PCB

view”. Các tùy chọn bên phải với chữ F là cái mà ta cần lựa chọn. Click và kéo vào kho vực làm việc. Lớp in trên bo mạch này mặc định là màu trắng chứ không phải là lớp màu đồng. Thông thường các lớp tiêu đề bo mạch được thiết kế ngược để lúc in rabo mạch sẽ có dạng đúng.



Hình 46. Tạo tiêu đề cho mạch

Phía trên là các bước cơ bản để thực hiện một project Fritzing đơn giản. Đối với các trường hợp sử dụng bo mạch Arduino thì thiết kế của ta sẽ phức tạp hơn rất nhiều. Do đó yêu cầu đặt ra là phải sắp xếp các linh kiện trên bo mạch sao cho tiết kiệm được mức tối đa chi phí và không gian nhưng phải đảm bảo là mạch hoạt động tốt .

Kết luận

Trong quá trình thực hiện đề tài này, các thành viên đã thật sự cố gắng hết sức mình bên cạnh sự chỉ dẫn tận tình thầy giáo. Đồ án 1 được thực hiện trong vòng 15 tuần, sau khi hoàn thành đề tài này chúng em thực sự hiểu được những vấn đề căn bản đối với việc thiết kế và lập trình một số loại bo mạch Arduino. Ngoài ra cung cấp những kiến thức về các phần mềm ứng dụng trong thiết kế mạch và những kỹ năng lập trình cơ bản của ngôn ngữ lập trình phân cứng. Nhưng một điều quan trọng không thể không nói đến đó là tạo cho chúng em những kỹ năng mềm của bản thân, khả năng tư duy độc lập và khả năng làm việc nhóm.

Một lần nữa em xin chân thành cảm ơn thầy Nguyễn Hoàng Nam đã giúp đỡ chúng em hoàn thành tốt bài tập lớn này !!!

Danh mục tài liệu tham khảo

1. Arduino Programming- Ebook.
2. Introduction to Arduino - Ebook
3. <http://arduino.cc/>
4. <http://machtudong.vn/arduino>
5. <http://mad-science.wonderhowto.com>
6. <http://fritzing.org>
7. Các nguồn tài liệu khác...

Phụ lục

1. Phụ lục 1: Sơ đồ nguyên lý bo mạch Arduino Uno.
2. Phụ lục 2: Sơ đồ nguyên lý bo mạch Arduino Ethernet.
3. Phụ lục 3: Sơ đồ nguyên lý bo mạch Arduino Nano.
4. Phụ lục 4: Sơ đồ nguyên lý bo mạch Arduino Mega ADK.