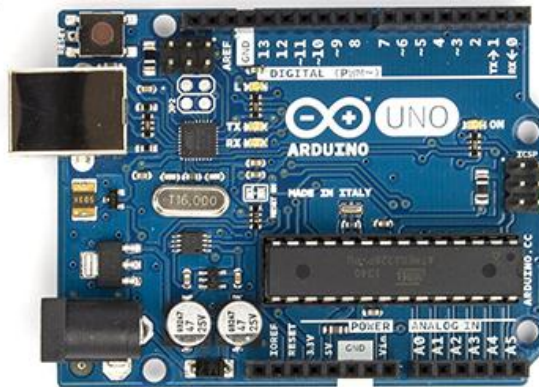


TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU



PHIÊN BẢN 1



BIÊN SOẠN: THẠC SĨ HUỲNH MINH PHÚ

TP.HCM, NGÀY 24 THÁNG 05 NĂM 2015

LỜI NÓI ĐẦU

Tự động hóa, robot hóa là xu hướng của nhân loại ngày nay. Từ ngàn xưa, con người luôn ước ao có những cỗ máy để làm việc thay thế mình. Trong thời đại hiện nay, với sự phát triển mạnh mẽ của khoa học kỹ thuật, con người đã dần tạo ra những máy móc thông minh, giúp mình từ các công việc thường ngày đến những công việc khó khăn nguy hiểm. Các máy tự động, các robot thông minh, tự hành do con người tạo nên không chỉ tồn tại ở trái đất mà còn có mặt tại mặt trăng, sao hỏa, trong không gian.

Đi cùng với nhịp điệu phát triển của khoa học kỹ thuật trên thế giới, người Việt Nam cũng đang rất nỗ lực trong nghiên cứu khoa học kỹ thuật, đặc biệt là trong lĩnh vực tự động hóa, robot hóa, phục vụ cho đời sống hàng ngày, cho công cuộc xây dựng và bảo vệ đất nước Việt Nam thân yêu. Sự nỗ lực đó đã giúp cho người Việt dần nắm bắt, làm chủ các công nghệ tiên tiến trên thế giới, từ xây cầu dây văng đến làm hầm Sài Gòn và hiện nay là thực hiện công trình tàu điện ngầm tại Việt Nam. Người Việt với sự cần cù sáng tạo và tình yêu quê hương đất nước mạnh mẽ, cùng vị trí địa lý chiến lược của quốc gia nhất định sẽ thực hiện thành công sự nghiệp công nghiệp hóa, hiện đại hóa của mình và nhất định trong tương lai gần sẽ bắt kịp các nước phát triển trong khu vực về khoa học và kỹ thuật.

Sự ra đời của mạch Arduino thúc đẩy sự yêu thích, tìm tòi nghiên cứu, ứng dụng tự động hóa, robot hóa vào đời sống và công nghiệp. Với những ưu điểm riêng của mình, Arduino đã nhanh chóng nổi tiếng toàn thế giới và được giới học sinh, sinh viên, giới nghiên cứu, những người yêu thích kỹ thuật, những người thích làm đồ tự chế sử dụng rộng rãi. Tại Việt Nam, số lượng người sử dụng mạch Arduino ngày càng tăng. Để đáp ứng nhu cầu tìm hiểu Arduino, tài liệu này đã được biên soạn. Tài liệu gồm có 6 phần, chủ yếu đáp ứng nhu cầu tự học của các bạn mới bắt đầu tìm hiểu về Arduino. Tài liệu được viết một cách ngắn gọn, dễ hiểu, đi vào những vấn đề cốt lõi trong việc sử dụng Arduino cho cuộc sống.

Dù đã cố gắng nhiều trong quá trình viết tài liệu, nhưng chắc chắn tài liệu không thể tránh khỏi sai sót trong quá trình biên soạn. Tác giả rất mong nhận được sự góp ý chân thành của các bạn để tác giả có thể hoàn thiện hơn nữa tài liệu này. Mọi ý kiến đóng góp xin gửi về địa chỉ email: hmpubt@gmail.com. Tác giả xin chân thành cảm ơn.

Tác giả

Thạc Sĩ Huỳnh Minh Phú

MỤC LỤC

Nội dung	Trang
Phần 1: Giới thiệu về Arduino.....	4
Phần 2: Phần cứng của Arduino Uno R3	5
Phần 3: Download và cài đặt chương trình IDE cho Arduino.....	9
Phần 4: Giao diện của phần mềm IDE.....	14
Phần 5: Cấu trúc của một chương trình trong phần mềm IDE.....	17
Phần 6: 4 bài tập trong bộ tự học Arduino căn bản.....	20
Bài 1. Lấy tín hiệu tương tự và tín hiệu số từ cảm biến.....	20
Bài 2. Sử dụng module Role.....	22
Bài 3. Sử dụng mạch cầu H.....	26
Bài 4. Sử dụng màn hình LCD.....	30

PHẦN 1: GIỚI THIỆU VỀ ARDUINO

1. Arduino là gì?

Arduino là một bo mạch vi điều khiển do một nhóm giáo sư và sinh viên Ý thiết kế và đưa ra đầu tiên vào năm 2005. Mạch Arduino được sử dụng để cảm nhận và điều khiển nhiều đối tượng khác nhau. Nó có thể thực hiện nhiều nhiệm vụ từ lấy tín hiệu từ cảm biến đến điều khiển đèn, động cơ, và nhiều đối tượng khác. Ngoài ra mạch còn có khả năng liên kết với nhiều module khác nhau như module đọc thẻ từ, ethernet shield, sim900A, để tăng khả năng ứng dụng của mạch.

Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM, Atmel 32-bit,.... Hiện phần cứng của Arduino có tất cả 6 phiên bản, Tuy nhiên phiên bản thường được sử dụng nhiều nhất là Arduino Uno và Arduino Mega. Arduino Uno được sử dụng rất rộng rãi trên thế giới, rất nhiều ví dụ trên youtube hoặc các trang hướng dẫn về Arduino sử dụng mạch này. Vì vậy đối với các bạn mới học Arduino, việc chọn Arduino Uno sẽ giúp các bạn có thể tự học dễ dàng.

Phần mềm để lập trình cho mạch Arduino là phần mềm IDE. Đây là phần mềm mã nguồn mở, và có thể được download từ trang web của Arduino: arduino.cc. Việc hướng dẫn download và sử dụng phần mềm này sẽ được đề cập đến trong những phần sau.

2. Tự học Arduino như thế nào cho hiệu quả?

Hiện trên thế giới có rất nhiều kênh youtube và rất nhiều trang web, các nhân chuyên hướng dẫn và chia sẻ miễn phí các dự án về Arduino. Vì vậy nếu các bạn có vốn tiếng Anh khá thì có thể tự học từ trên internet, từ các trang web nước ngoài.

Ở Việt Nam hiện cũng có nhiều trang web và kênh youtube hướng dẫn tự học Arduino. Bản thân tác giả cũng có 1 kênh youtube để chia sẻ code và các dự án về Arduino. Hiện tác giả đã có 97 clip hướng dẫn về Arduino với đầy đủ mã code kèm theo và chia sẻ miễn phí cho cộng đồng. Các bạn có thể tìm loạt clip trên youtube với từ khóa: “Lập trình vi điều khiển, Huỳnh Minh Phú”. Trong quá trình tự học, các bạn không nên copy cả mã code mà từ code mẫu, các bạn hãy tự tay mình đánh lại từng dòng lệnh, tự phân tích tại sao lại viết như vậy, rồi kiểm tra chương trình, rồi nạp chương trình cho bo. Như vậy các bạn có thể hiểu rõ hơn, cũng như sẽ thấy những sai sót khi mình đánh, từ đó hiểu các lỗi đó, cũng như hiểu hơn cách viết chương trình..

3. Tại sao lại chọn Arduino?

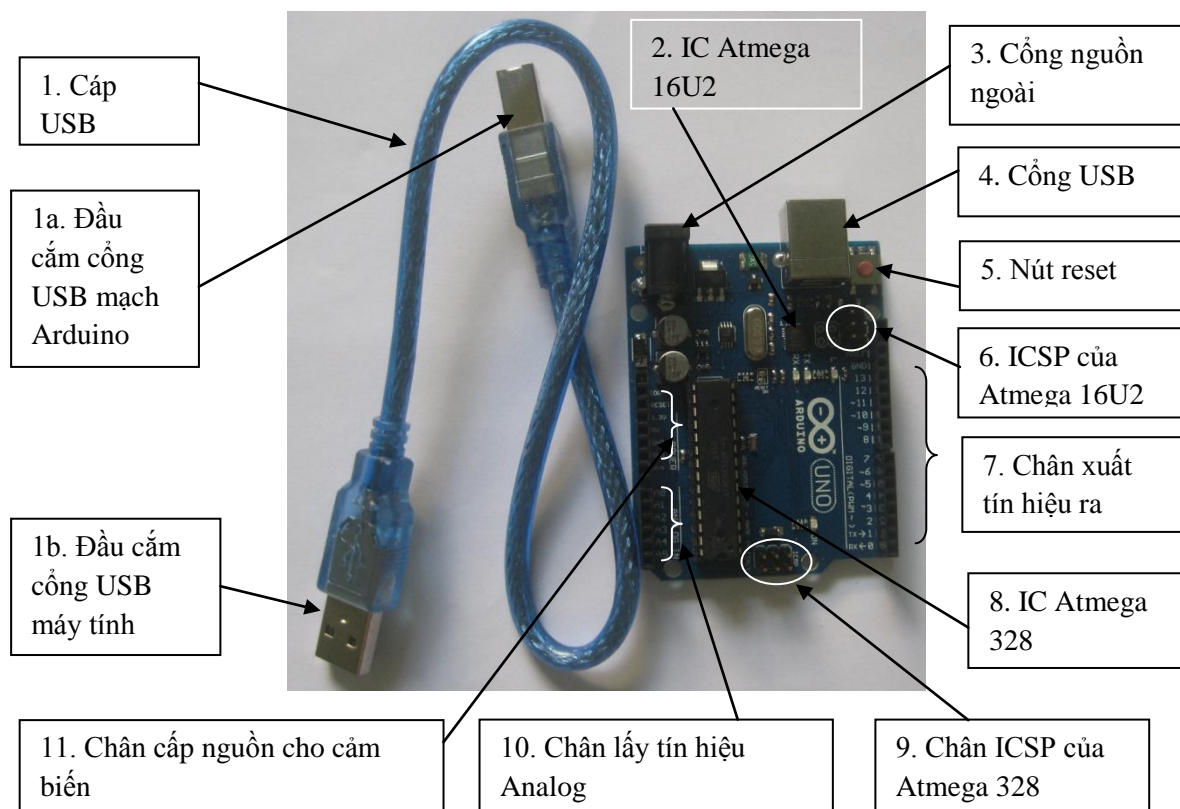
Hiện tại ở Việt Nam và trên thế giới cũng có nhiều bo mạch vi điều khiển khác nhau. Tuy nhiên Arduino có một số ưu điểm mà khiến nó trở nên nổi tiếng và hiện đang được sử dụng rộng rãi trên thế giới. Những ưu điểm đó là: rẻ, tương thích được với nhiều hệ điều hành, chương trình lập trình đơn giản, rõ ràng, dễ sử dụng, sử dụng mã nguồn mở và có thể kết hợp với nhiều module khác nhau.

4. Mua các mạch Arduino ở đâu?

Hiện tác giả có 1 shop riêng bán mạch Arduino, shop Kỹ Thuật Phú Hưng, các bạn có thể tham khảo tại địa chỉ: www.ktphuhung.com.

PHẦN 2: PHẦN CỨNG CỦA ARDUINO UNO R3

Phần này nói về phần cứng của Arduino Uno R3, một bo mạch thông dụng hiện nay. Do đây là tài liệu hướng dẫn học nhanh nên tác giả chỉ đưa ra những kiến thức cơ bản cần thiết cho việc sử dụng board mạch này. Nếu người học cần những kiến thức chuyên sâu hơn thì có thể tham khảo tại website chính thức của Arduino: arduino.cc.



Hình 1. Mạch Arduino Uno

1. Cáp USB

Đây là dây cáp thường được bán kèm theo bo, dây cáp dùng để cắm vào máy tính để nạp chương trình cho bo và dây đồng thời cũng lấy nguồn từ nguồn usb của máy tính để cho bo hoạt động. Ngoài ra cáp USB còn được dùng để truyền dữ liệu từ bo Arduino lên máy tính. Dây cáp có 2 đầu, đầu 1a được dùng để cắm vào cổng USB trên bo Arduino, đầu 1b dùng để cắm vào cổng USB trên máy tính.

2. IC Atmega 16U2

IC này được lập trình như một bộ chuyển đổi USB –to-Serial dùng để giao tiếp với máy tính thông qua giao thức Serial (dùng cổng COM).

3. Cổng nguồn ngoài

Cổng nguồn ngoài nhằm sử dụng nguồn điện bên ngoài như pin, bình acquy hay các adapter cho bo Arduino hoạt động. Nguồn điện cấp vào cổng này là nguồn DC có hiệu điện thế từ 6V đến 20V, tuy nhiên hiệu điện thế tốt nhất mà nhà sản xuất khuyến dùng là từ 7 đến 12V

4. Cổng USB

Cổng USB trên bo Arduino dùng để kết nối với cáp USB

5. Nút reset

Nút reset được sử dụng để reset lại chương trình đang chạy. Đôi khi chương trình chạy gặp lỗi, người dùng có thể reset lại chương trình..

6. ICSP của ATmega 16U2

ICSP là chữ viết tắt của In-Circuit Serial Programming. Đây là các chân giao tiếp SPI của chip Atmega 16U2. Các chân này thường ít được sử dụng trong các dự án về Arduino.

7. Chân xuất tín hiệu ra

Có tất cả 14 chân xuất tín hiệu ra trong Arduino Uno, những chân có dấu ~ là những chân có thể băm xung (PWM), tức có thể điều khiển tốc độ động cơ hoặc độ sáng của đèn. Hình 2 thể hiện rất rõ những chân để băm xung này.

8. IC ATmega 328

IC Atmega 328 là linh hồn của bo mạch Arduino Uno, IC này được sử dụng trong việc thu thập dữ liệu từ cảm biến, xử lý dữ liệu, xuất tín hiệu ra,...

9. Chân ICSP của ATmega 328

Các chân ICSP của ATmega 328 được sử dụng cho các giao tiếp SPI (Serial Peripheral Interface), một số ứng dụng của Arduino có sử dụng chân này, ví dụ như sử dụng module RFID RC522 với Arduino hay Ethernet Shield với Arduino.

10. Chân lấy tín hiệu Analog

Các chân này lấy tín hiệu Analog (tín hiệu tương tự) từ cảm biến để IC Atmega 328 xử lý. Có tất cả 6 chân lấy tín hiệu Analog, từ A0 đến A5.

11. Chân cấp nguồn cho cảm biến

Các chân này dùng để cấp nguồn cho các thiết bị bên ngoài như role, cảm biến, RC servo,...trên khu vực này có sẵn các chân GND (chân nối đất, chân âm), chân 5V, chân 3.3V như được thể hiện ở hình 2. Nhờ những chân này mà người sử dụng không cần thiết bị biến đổi điện khi cấp nguồn cho cảm biến, role, rc servo,...Ngoài ra trên khu vực này còn có chân Vin và chân reset, chân IOREF. Tuy nhiên các chân này thường ít được sử dụng nên trong tài liệu này xin không đi sâu về nó.

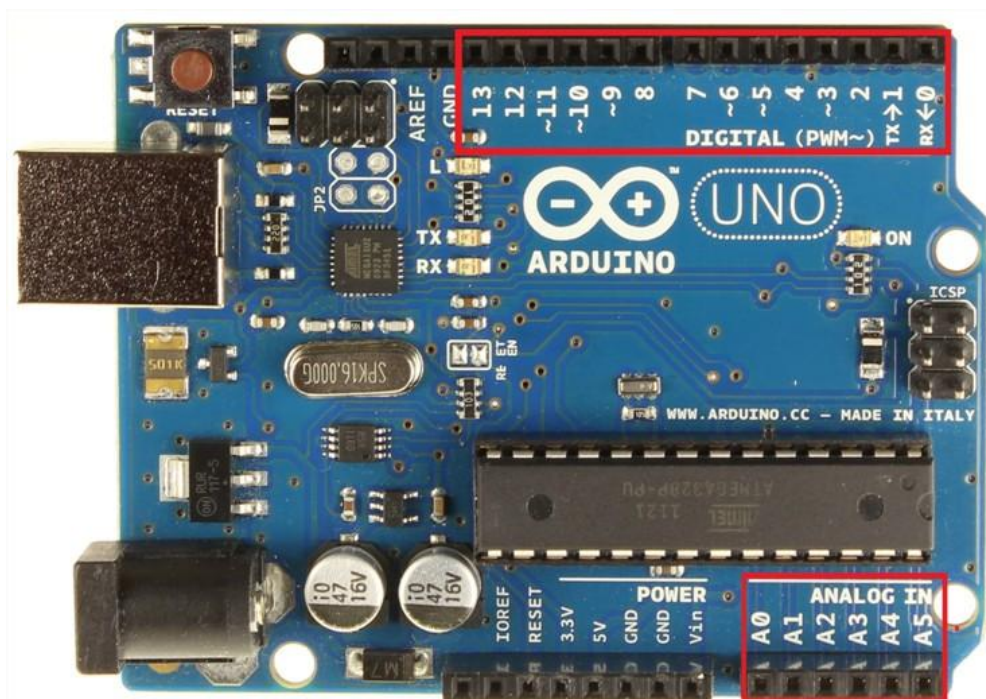
12. Các linh kiện khác trên board Arduino Uno

Ngoài các linh kiện đã liệt kê bên trên, Arduino Uno còn 1 số linh kiện đáng chú ý khác. Trên bo có tất cả 4 đèn led, bao gồm 1 led nguồn (led ON nhằm cho biết bo đã được cấp nguồn), 2 led Tx và Rx, 1 led L. Các led Tx và Rx sẽ nhấp nháy khi có dữ liệu truyền từ board lên máy tính hoặc ngược lại thông qua cổng USB. Led L được kết nối với chân số 13. Led này được gọi là led on board (tức led trên bo), led này giúp người dùng có thể thực hành các bài đơn giản mà không cần dùng thêm led ngoài.

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

Trong 14 chân ra của bo còn có 2 chân 0 và 1 có thể truyền nhận dữ liệu nối tiếp TTL. Có một số ứng dụng cần dùng đến tính năng này, ví dụ như ứng dụng điều khiển mạch Arduino Uno qua điện thoại sử dụng bluetooth HC05.

Thêm vào đó, chân 2 và chân 3 cũng được sử dụng cho lập trình ngắt (interrupt), đồng thời còn 1 vài chân khác có thể được sử dụng cho các chức năng khác, như được thể hiện ở hình 3. Bảng 1 thể hiện thêm các thông số cho bo Arduino Uno R3.



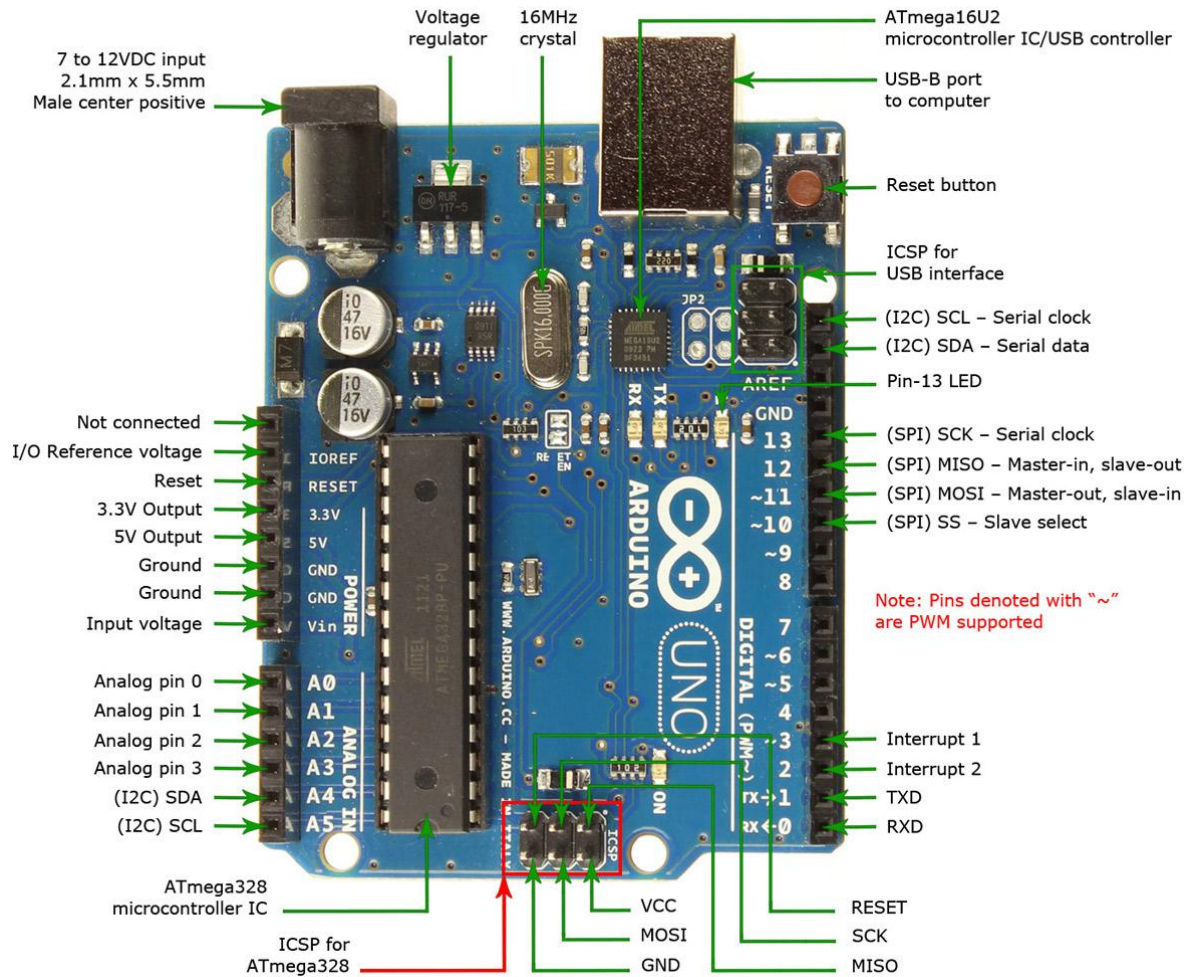
Hình 2. Các chân vào ra của Arduino Uno

Bảng 1. Một số thông số của Arduino Uno R3

Vi điều khiển	ATmega328 (họ 8bit)
Điện áp hoạt động	5V – DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	30mA
Điện áp vào khuyến dùng	7-12V – DC
Điện áp vào giới hạn	6-20V – DC
Số chân Digital I/O	14 (6 chân PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

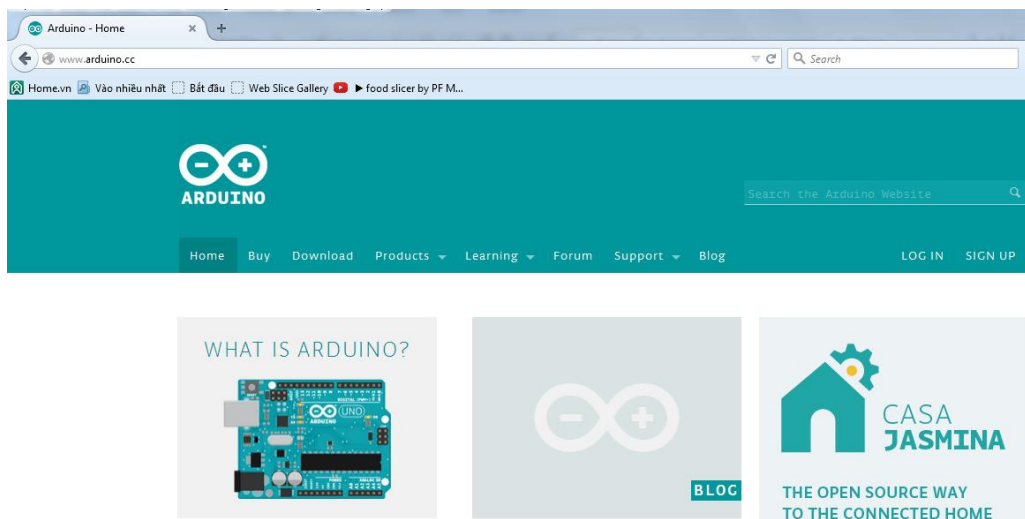
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)



Hình 3. Tham khảo thêm một số chức năng của các chân trên Arduino.

PHẦN 3: DOWNLOAD VÀ CÀI ĐẶT CHƯƠNG TRÌNH IDE CHO ARDUINO

1. Vào trang chủ của Arduino: arduino.cc



Hình 4. Trang chủ của Arduino

2. Nhấn vào mục download



Hình 5. Màn hình của phần download.

3. Chọn hệ điều hành, ví dụ hệ điều hành là Windows:

Có 2 tùy chọn cho hệ điều hành Windows là Windows installer và Windows zip file for non admin install. Tùy chọn thứ nhất dùng cho người là admin của máy, tùy chọn thứ 2 cho người không phải admin của máy. Thông thường đa số chọn tùy chọn thứ nhất, tức Windows installer.

4. Download chương trình về để cài đặt

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

Arduino có đưa ra 2 lựa chọn, bao gồm chỉ download về (just download) và download về cùng quyên góp (contribute and download), như được thể hiện ở hình 6. Arduino mong muốn người dùng nếu có điều kiện hãy quyên góp ủng hộ hãng, mức quyên góp thấp nhất là 3USD.

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)

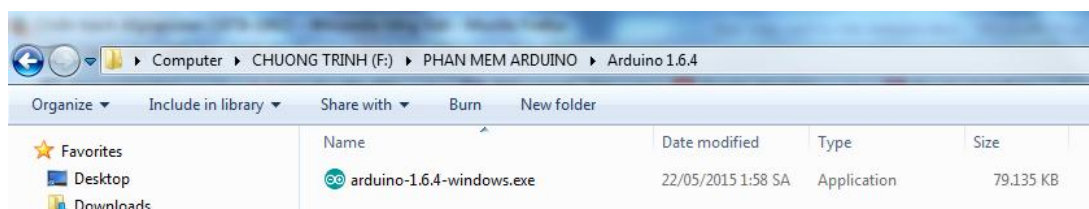


Hình 6. Màn hình trang download phần mềm.

5. Cài đặt

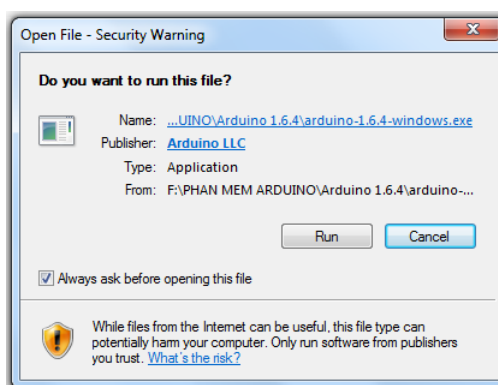
Phiên bản được viết trong bài này là phiên bản IDE 1.6.4. Đây là phiên bản mới nhất của phần mềm IDE. Sau này có thể sẽ có những phiên bản kế tiếp, tuy nhiên về cơ bản thì cách cài đặt giống nhau.

Sau khi download về thì người dùng sẽ được 1 file .exe như hình 7 bên dưới. Để cài đặt, người dùng nhấp đúp vào file đó, phần mềm sẽ bắt đầu cài đặt.



Hình 7. File chạy phần mềm sau khi download về.

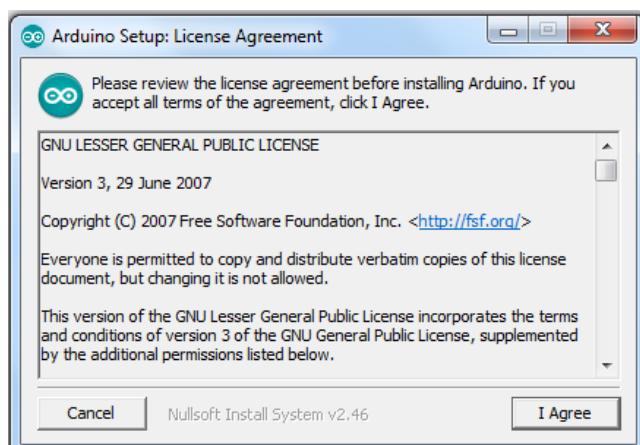
Một thông báo có thể sẽ được hiện lên với hàng chữ: “do you want to run this file?”, tức là bạn có muốn chạy file này không? Người dùng click và run (chạy) để cài chương trình.



Hình 8. Bảng thông báo bắt đầu cài đặt phần mềm

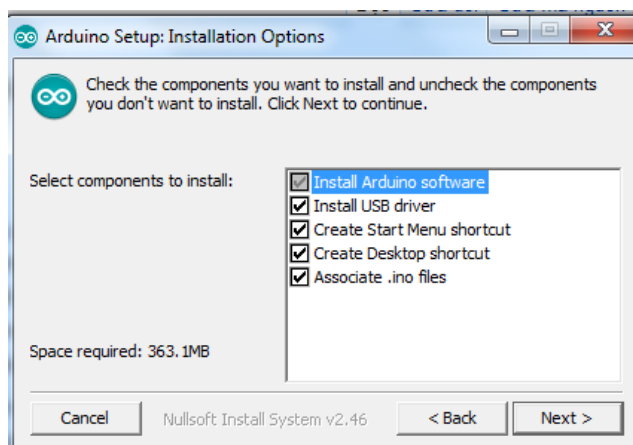
TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

Một thông báo về các điều khoản khi sử dụng phần mềm sẽ hiện ra. Người dùng nhấn vào I Agree (tôi đồng ý), để tiếp tục cài đặt chương trình.



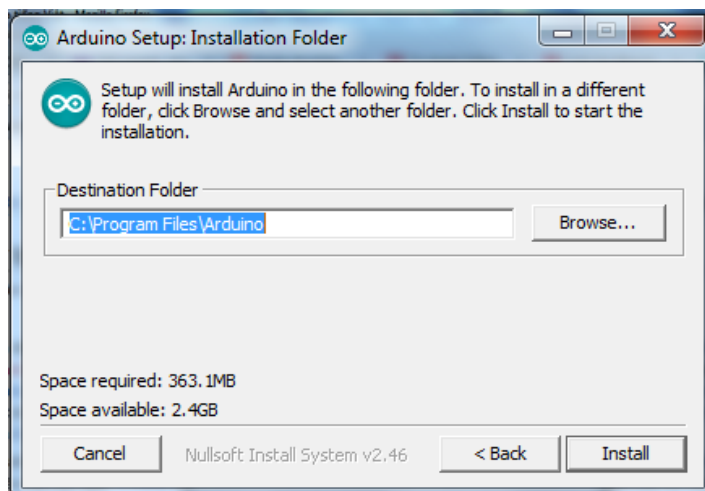
Hình 9. Thông báo về điều khoản phần mềm

Một màn hình để người dùng lựa chọn việc cài đặt các phần liên quan được hiện ra, như ở hình 10. Các bạn có thể để nguyên như vậy và bấm Next để tiếp tục quá trình cài đặt.



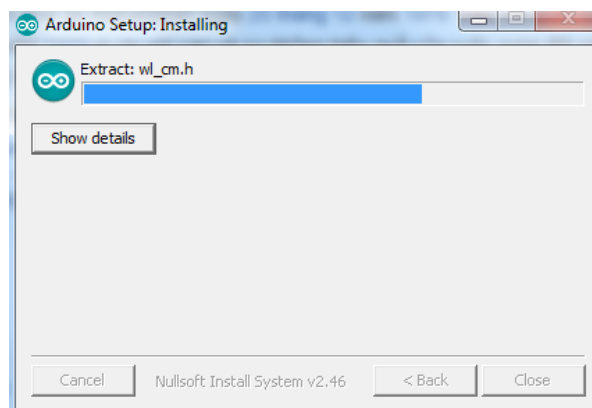
Hình 10. Các lựa chọn khi cài đặt

Phần mềm sẽ yêu cầu bạn chọn vị trí để cài đặt. Các bạn có thể để nơi cài đặt mặc định như phần mềm đưa ra, không thay đổi gì rồi bấm Install (Cài đặt).



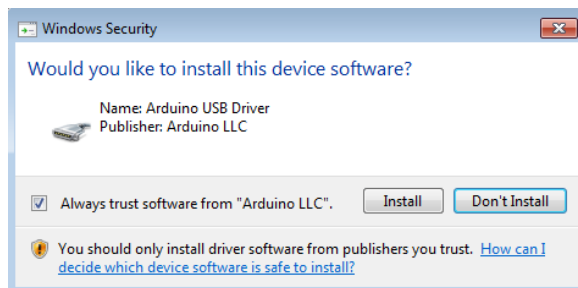
Hình 11. Chọn đường dẫn cho phần mềm

Màn hình thể hiện việc cài đặt sẽ xuất hiện, người dùng có thể nhấp vào Show detail (Xem chi tiết) để xem chi tiết quá trình cài đặt.



Hình 12. Quá trình cài đặt đang được thực hiện.

Trong quá trình cài đặt 1 số phiên bản sẽ hỏi có cài driver USB cho phần mềm IDE không, như hình 13 bên dưới, người dùng nên tích chọn vào ô vuông “Always trust software from “Arduino LLC””, sau đó bấm Install (Cài đặt) để cài đặt driver USB. Cần phải cài driver này thì chương trình mới nhận cổng USB của mạch Arduino

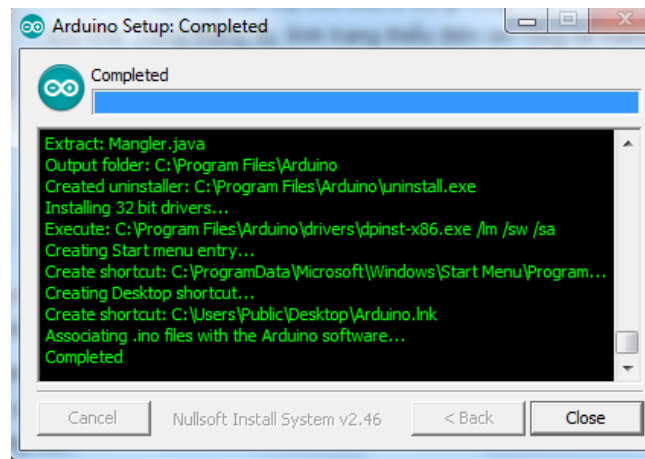


Hình 13. Cài đặt driver USB cho IDE

Sau khi việc cài đặt hoàn thành, trên màn hình sẽ xuất hiện dòng chữ Completed (Đã hoàn thành), lúc này người dùng nhấn nút Close (đóng) để hoàn tất quá trình cài đặt và bây giờ có thể

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

sử dụng phần mềm. Biểu tượng của phần mềm sẽ xuất hiện trên màn hình Desktop và bây giờ người dùng chỉ việc nhấp đúp vào nó để sử dụng như các phần mềm khác.

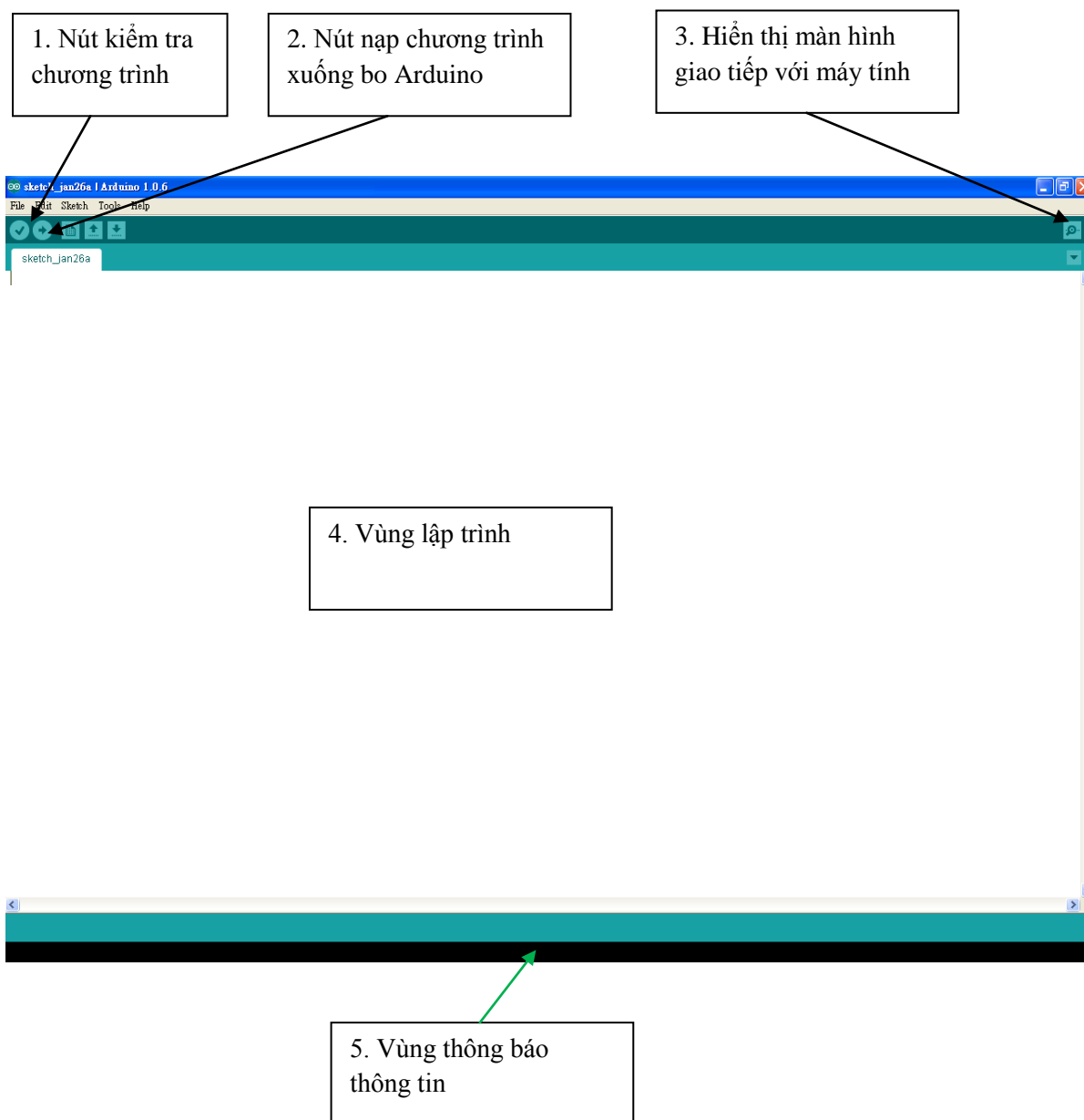


Hình 14. Phần mềm đã được cài đặt xong

Phần kế tiếp sẽ nói về giao diện của chương trình để người dùng có thể hiểu rõ về cách sử dụng phần mềm này.

PHẦN 4: GIAO DIỆN CỦA PHẦN MỀM IDE

Phần này nói về giao diện của phần mềm IDE, hình bên dưới thể hiện những phần cơ bản của giao diện. Người dùng có thể tìm hiểu sâu hơn về giao diện này, tuy nhiên đây là phần cơ bản nhất và thường dùng nhất để có thể học nhanh Arduino, khi có thời gian các bạn có thể tìm hiểu thêm các bài viết về nó trên arduino.cc hoặc google,... Các chức năng cơ bản của các biểu tượng trên phần mềm được trình bày chi tiết ở các phần bên dưới.



Hình 15. Giao diện của phần mềm IDE

Giao diện của phần mềm IDE có nhiều phần, tuy nhiên chúng ta chú ý đến những phần quan trọng như được nêu ra trong hình trên. Chức năng của từng phần như sau:

1. Nút kiểm tra chương trình

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

Dùng để kiểm tra xem chương trình được viết có lỗi không. Nếu chương trình bị lỗi thì phần mềm sẽ hiển thị thông tin lỗi ở vùng số 5.

2. Nút nạp chương trình xuống bo Arduino

Dùng để nạp chương trình được viết xuống mạch Arduino. Trong quá trình nạp, chương trình sẽ được kiểm tra lỗi trước sau đó mới thực hiện nạp xuống mạch Arduino.

3. Hiện thị màn hình giao tiếp với máy tính

Khi nhấp vào biểu tượng cái kính lúp thì phần giao tiếp với máy tính sẽ được mở ra. Phần này sẽ hiển thị các thông số mà người dùng muốn đưa lên màn hình. Muốn đưa lên màn hình phải có lệnh `Serial.print()` mới có thể đưa thông số cần hiển thị lên màn hình

4. Vùng lập trình

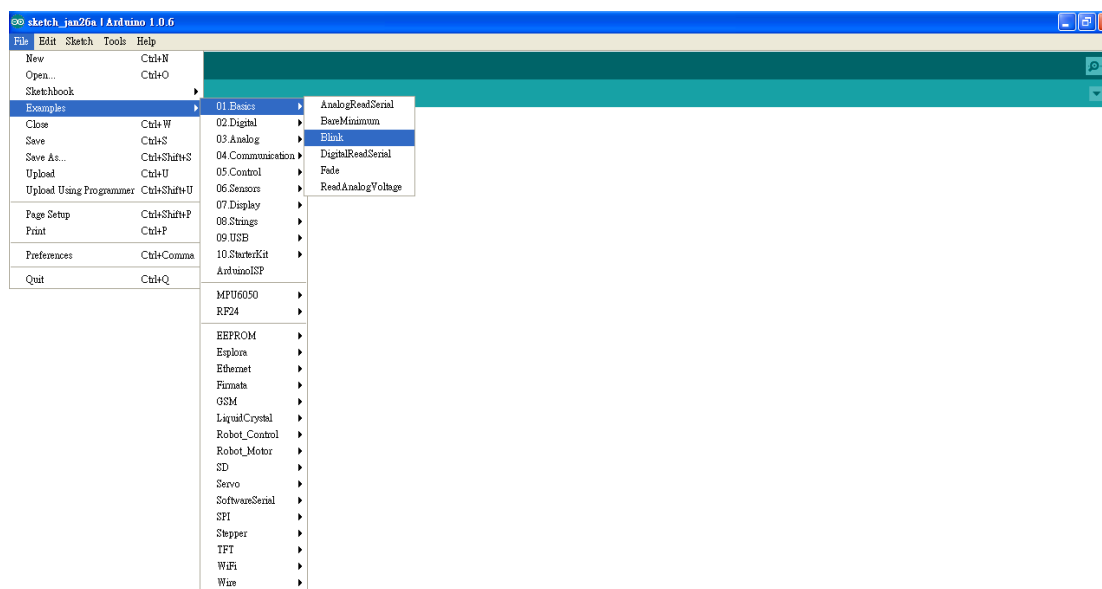
Vùng này để người lập trình thực hiện việc lập trình cho chương trình của mình.

5. Vùng thông báo thông tin:

Có chức năng thông báo các thông tin lỗi của chương trình hoặc các vấn đề liên quan đến chương trình được lập.

6. Sử dụng một số menu thông dụng trên phần mềm IDE

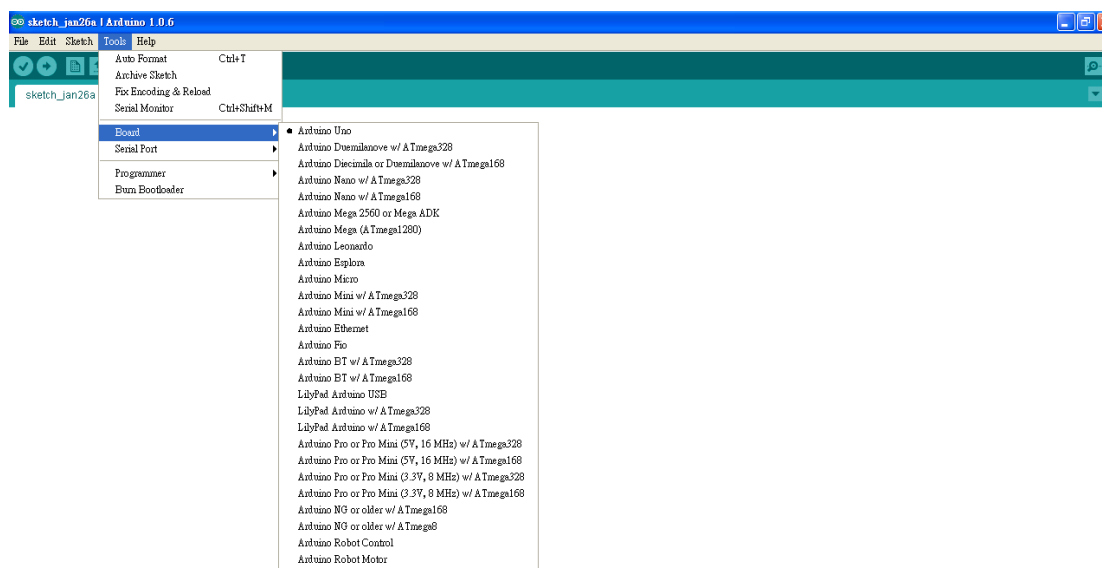
Có vài menu trong phần mềm IDE, tuy nhiên thông dụng nhất vẫn là menu File, ngoài những tính năng như mở một file mới hay lưu một file, phần menu này có một mục đáng chú ý là Example. Phần Example (ví dụ) đưa ra các ví dụ sẵn để người lập trình có thể tham khảo, giảm bớt thời gian lập trình. Hình bên dưới thể hiện việc chọn một ví dụ cho led chớp tắt (blink) để nạp cho mạch Arduino. Ví dụ về led chớp tắt này thường được dùng để kiểm tra bo khi mới mua về.



Hình 16. Menu file trên phần mềm IDE

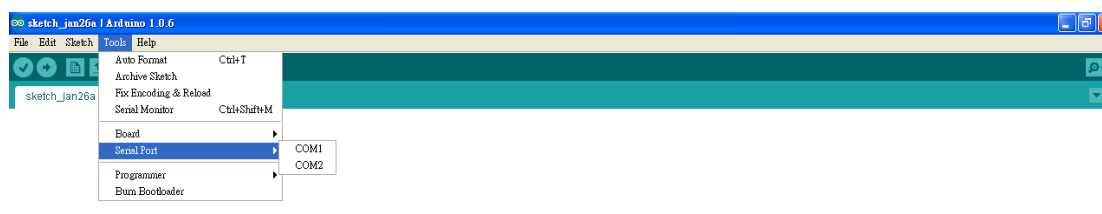
TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

Một menu thường được sử dụng khác là menu Tools. Khi mới kết nối bo Arduino với máy tính ta click vào Tools->board để chọn loại board sử dụng. Phần mềm chọn sẵn kiểu bo là bo Arduino Uno, nếu người dùng dùng kiểu bo khác thì chọn kiểu bo đang dùng.



Hình 17. Cách chọn bo trên màn hình IDE

Bên cạnh việc chọn bo thì một phần quan trọng nữa là chọn cổng COM. Hình bên dưới minh họa cho việc chọn cổng COM. Khi lần đầu gắn mạch Arduino vào máy tính, người sử dụng cần nhấn chọn cổng COM bằng cách vào Tools -> Serial Port (một số phiên bản dùng từ Port) sau đó nhấn chọn cổng COM, ví dụ như COM1. Những lần sau khi đưa chính board Arduino đó vào máy tính thì không cần chọn cổng COM, nếu đưa bo Arduino khác vào máy thì cần phải chọn lại cổng COM, quy trình thực hiện cũng tương tự.



Hình 18. Cách chọn cổng COM trên màn hình IDE

Sau khi thực hiện các bước trên xong, người dùng sẽ bắt tay vào việc lập trình. Phần kế tiếp sẽ trình bày cấu trúc của một chương trình trong phần mềm IDE.

PHẦN 5: CẤU TRÚC CỦA MỘT CHƯƠNG TRÌNH TRONG PHẦN MỀM IDE

Phần này sẽ đưa ra cấu trúc của một chương trình trong IDE, đồng thời giải thích một số lệnh thường được sử dụng để thuận tiện cho người dùng.

Xét 1 ví dụ đơn giản, ví dụ làm cho led nhấp nháy:

```
#define led = 13 // khai báo chân led là chân 13

void setup() {
  pinMode(led, OUTPUT); //Thiết lập chân led (chân 13) là chân ra (OUTPUT)
}

void loop() {
  digitalWrite(led, HIGH); // Ra lệnh mở led
  delay(1000);           // Mở trong 1s (1000ms), có thể thay đổi giá trị này
  digitalWrite(led, LOW); // ra lệnh tắt led
  delay(1000);           // Tắt trong 1s (1000ms), có thể thay đổi giá trị này
}
```

Sau đây chúng ta sẽ cùng tìm hiểu cấu trúc của một chương trình trong Arduino qua ví dụ này.

1. Phần 1: Khai báo biến

Đây là phần khai báo kiểu biến, tên các biến, định nghĩa các chân trên board một số kiểu khai báo biến thông dụng:

```
* #define
```

Nghĩa của từ define là định nghĩa, hàm #define có tác dụng định nghĩa, hay còn gọi là gán, tức là gán một chân, một ngõ ra nào đó với 1 cái tên.

Ví dụ #define led 13

Chú ý: sau #define thì không có dấu “,” (dấu phẩy)

*Khai báo các kiểu biến khác như: int (kiểu số nguyên), float,...

Các bạn có thể tham khảo thêm các kiểu biến cũng như công dụng tại arduino.cc

2. Phần 2: Thiết lập (void setup())

Phần này dùng để thiết lập cho chương trình, cần nhớ rõ cấu trúc của nó

```
void setup()
{
.....
}
```

Cấu trúc của nó có dấu ngoặc nhọn ở đầu và ở cuối, nếu thiếu phần này khi kiểm tra chương trình thì chương trình sẽ báo lỗi.

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

Phần này dùng để thiết lập các tốc độ truyền dữ liệu, kiểu chân là chân ra hay chân vào. Trong đó:

Serial.begin(9600);	Dùng để truyền dữ liệu từ board Arduino lên máy tính.
pinMode(biến, kiểu vào hoặc ra); Ví dụ: pinMode(ChanDO, INPUT);	Dùng để xác định kiểu chân là vào hay ra

3. Phần 3: Vòng lặp

Dùng để viết các lệnh trong chương trình để mạch Arduino thực hiện các nhiệm vụ mà chúng ta mong muốn, thường bắt đầu bằng:

```
void loop()
```

```
{
```

```
.....
```

```
}
```

Một số câu lệnh, cấu trúc thường gặp:

Bảng 2. Một số câu lệnh thường gặp

Ký hiệu, câu lệnh	Ý nghĩa
//	Dấu // dùng để giải thích, khi nội dung giải thích nằm trên 1 dòng, khi kiểm tra chương trình thì phần kiểm tra sẽ bỏ qua phần này, không kiểm tra,
/* */	Ký hiệu này cũng dùng để giải thích, nhưng giải thích dành cho 1 đoạn, tức có thể xuống dòng được
#define biến chân	Define nghĩa là định nghĩa, xác định. Câu lệnh này nhằm gán tên 1 biến vào 1 chân nào đó. Ví dụ #define led 13
digitalWrite(chân, trạng thái);	Dùng để tắt, mở 1 chân ra. Cú pháp của nó là digitalWrite(chân, trạng thái chân);. Ở đây trạng thái chân có thể là HIGH hoặc LOW. Ví dụ: digital(led,HIGH); , hoặc digital(led,LOW); . Chú dấu chấm phẩy đằng sau câu lệnh.
analogWrite(chân, giá trị);	Có ý nghĩa dùng để băm xung (PWM), thường dùng để điều khiển tốc độ động cơ, độ sáng led,..
digitalRead(chân);	Read nghĩa là đọc, lệnh này dùng để đọc giá trị digital tại chân muốn đọc
analogRead(chân);	Read nghĩa là đọc, lệnh này dùng để đọc giá trị analog tại chân muốn đọc
delay(Thời gian);	Delay nghĩa là chờ, trì hoãn, duy trì. Lệnh này dùng để duy trì trạng thái đang thực hiện chờ một thời gian. Thời gian ở đây được tính bằng mili giây, 1 giây bằng 1 ngàn mili giây.
if() { Các câu lệnh } else () { Các câu lệnh }	if nghĩa là nếu, sau if là dấu (), bên trong dấu ngoặc là một biểu thức so sánh. Ví dụ trong bài về cảm biến độ ẩm đất (phần 5) thì: if (giatriAnalog>500) //nếu giá trị đọc được của biến giatriAnalog lớn hơn 500 { digitalWrite(Led,HIGH); //Ra lệnh cho led sáng delay(1000); //chờ 1s } else nghĩa là ngược lại

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

<code>Serial.print()</code>	In ra màn hình máy tính, lệnh này in không xuống dòng
<code>Serial.println()</code>	In ra màn hình máy tính, in xong xuống dòng, giá trị tiếp theo sẽ được in ở dòng kế tiếp

PHẦN 6: 4 BÀI TẬP TRONG BỘ TỰ HỌC ARDUINO CĂN BẢN

BÀI 1. LẤY TÍN HIỆU TƯƠNG TỰ VÀ TÍN HIỆU SỐ TỪ CẢM BIẾN

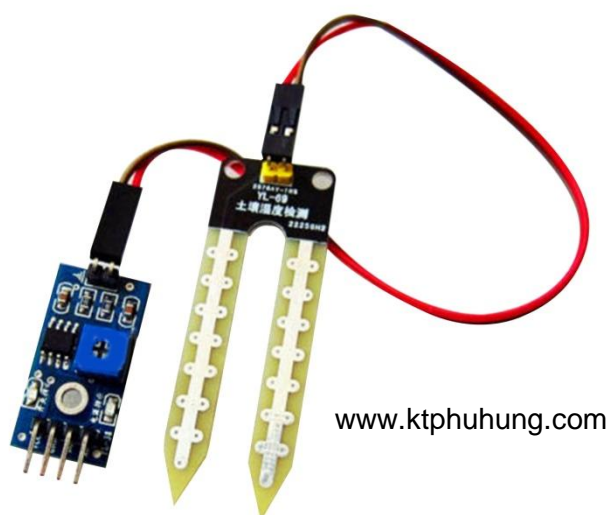
Mục đích của bài này là giúp các bạn hiểu cách lấy tín hiệu tương tự (Analog) và tín hiệu số (digital) từ cảm biến. Cảm biến được sử dụng trong ví dụ này là cảm biến đo độ ẩm đất. Các bạn có thể tham khảo thêm video hướng dẫn trên youtube của tác giả. Video có tiêu đề là: Lập trình vi điều khiển-phần 82-Cảm biến độ ẩm đất.

1. Dụng cụ thực hành:

Dụng cụ thực hành bao gồm 1 cảm biến độ ẩm đất, 1 board Arduino Uno, 1 đèn led

2. Cảm biến đo độ ẩm đất:

Loại cảm biến này được sử dụng để đo độ ẩm đất, được dùng trong việc tưới tiêu cho nông nghiệp, cây cảnh,... Cảm biến có 4 chân, gồm 2 chân cấp nguồn VCC và GND, 2 chân AO và DO. Chân VCC của cảm biến đo độ ẩm đất được nối với chân 5V trên board Arduino Uno. Chân AO là viết tắt của Analog Output và chân DO là viết tắt của Digital Output



Hình 19. Cảm biến đo độ ẩm đất.

2. Nối dây:

Chân VCC nối nguồn 5V, chân GND nối chân GND trên board Arduino, chân AO nối chân, chân DO nối chân số 4, led nối chân 12.

3. Lập trình

Để lấy được tín hiệu digital, trong phần set up () người lập trình cần khai báo pinMode cho chân là INPUT, chân Analog không cần khai báo, phần mềm sẽ tự hiểu.

4. Phân tích mã code

Sau đây là mã code của chương trình:

```
//Chương trình đo độ ẩm đất với loại cảm biến đo độ ẩm đất được bán thông dụng ở Việt Nam
//Người viết: HUỖNH MINH PHÚ
//Cách nối chân:
//AO nối với A1, DO nối với chân 4, GND nối nguồn âm, VCC nối nguồn dương.
#define Analog 1// dùng chân A1 trên board Arduino để đọc tín hiệu Analog
```

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

```
#define Digital 4 //Dùng chân digital số 4 để đọc tín hiệu digital
#define Led 12//chân 12 làm chân điều khiển led
int giatriAnalog, giatriDigital; //Kiểu biến của giatriAnalog và giatriDigital là kiểu int (số nguyên)
void setup()
{
  Serial.begin(9600); // thiết lập tốc độ truyền dữ liệu
  pinMode(Digital,INPUT); //xác định kiểu chân 4 là chân vào (INPUT)
  pinMode(Led,OUTPUT); //Xác định kiểu chân led (chân 12) là chân ra (OUTPUT)
}
void loop()
{
  giatriAnalog=analogRead(Analog); //Lấy giá trị chân Analog (chân A1) bằng hàm analogRead()
  giatriDigital=digitalRead(Digital); //Lấy giá trị chân digital bằng hàm digitalRead()
  Serial.print("Gia tri Analog: "); //In ra màn hình máy tính hàng chữ: Gia tri Analog:
  Serial.println(giatriAnalog); //In ra màn hình giá trị của biến giatriAnalog sau đó xuống dòng
  Serial.print("Gia tri digital: "); //In ra màn hình máy tính dòng chữ: Gia tri digital:
  Serial.println(giatriDigital); // In ra màn hình giá trị của biến giatriDigital sau đó xuống dòng
  delay(1000); //chờ trong 1s
  if (giatriAnalog>500) //nếu giá trị đọc được của biến giatriAnalog lớn hơn 500
  {
    digitalWrite(Led,HIGH); //Ra lệnh cho led sáng
    delay(1000);//chờ 1s
  }
  else if(giatriAnalog<=500) // nếu giá trị đọc được của biến giatriAnalog nhỏ hơn hoặc bằng 500
  {
    digitalWrite(Led,LOW);//Ra lệnh cho led tắt
    delay(1000);//Chờ 1 giây
  }
}
```

BÀI 2. SỬ DỤNG MODULE ROLE

Khi cần sử dụng Arduino để đóng ngắt một động cơ vài trăm Wat đến vài KW, một bóng đèn 220V thì cần phải sử dụng thêm role với board Arduino. Role có tác dụng như một công tắc mà thiết bị điều khiển công tắc đó chính là Arduino.

I. Cấu tạo của Role

Mạch Arduino bản chất là một mạch điều khiển, dòng qua Arduino rất nhỏ, thông thường chỉ đủ chạy led hoặc RC Servo loại nhỏ (loại 9g). Nếu muốn dùng mạch Arduino để điều khiển đóng ngắt thiết bị có dòng lớn hơn 0.5A thì phải dùng role, nếu không dùng role mà cắm trực tiếp động cơ vào mạch Arduino thì sẽ làm hư mạch. Role (hay relay) là một thiết bị trung gian dùng để đóng ngắt các thiết bị có công suất lớn như các động cơ DC hay các động cơ AC, bao gồm máy bơm nước, quạt máy, đèn 220V, các thiết bị gia dụng khác,...

Loại Role được thường được sử dụng cùng Arduino là Role 5V, tức dùng nguồn 5V để điều khiển. Hình 20 thể hiện một module role 5V được bán thông dụng trên thị trường.

II. Phân tích các chức năng của Module Role

1. Chân thường mở:

Về bản chất, Role giống như một công tắc điện mà thành phần ra lệnh nó chính là mạch Arduino. Chân thường mở nghĩa là ở trạng thái bình thường, khi role không nhận lệnh từ Arduino thì nó mở ra, khi nào có lệnh từ Arduino thì nó đóng lại với chân chung. Khi chân thường mở đóng với chân chung (giống như công tắc được bật) thì dòng điện sẽ đi từ chân chung qua chân thường đóng.

2. Chân chung (chân COM)

Chân chung là chân nối với nguồn cần cấp cho động cơ, thường chân chung này được nối với nguồn dương của pin, bình acquy hoặc dây nóng của điện 220V.

3. Chân thường đóng

Trái ngược với chân thường mở là chân thường đóng. Chân thường đóng là chân mà bình thường, lúc role chưa nhận lệnh gì từ mạch Arduino, thì nó đóng với chân chung (chân com). Khi chân thường mở đóng thì chân thường đóng này sẽ hở ra, không nối với chân chung nữa.

Chú ý: chân thường đóng và chân thường mở có trạng thái trái ngược nhau, cái này đóng thì cái kia sẽ mở.

4. Khả năng chịu tải

Khả năng chịu tải thể hiện tải tốt đa mà role có thể chịu được. Như đối với module này, thông số trên role ghi là 10A, 250VAC, tức là tải tối đa đối với điện xoay chiều 220V (250V) là 10A. Lấy 2 thông số này nhân với nhau ta được công suất tải, tức công suất tối đa là 2500W hay 2,5KW, tương đương 1 máy bơm nước 2,5KW hay 3 ngựa. Tuy nhiên đây là công suất tối đa, khi sử dụng thì công suất của tải bằng phân nửa hoặc hai phần ba giá trị ghi trên role là tốt nhất. Cái thông số khác được hiểu tương tự.

5. Chế độ LOW

Loại Role này có 2 chế độ, LOW và HIGH. Khu vực đó có 3 chân, 1 chân giữa và 2 chân LOW và HIGH ở 2 bên, có 1 cái jump màu đen, có thể rút ra được, dùng để kết nối 2 chân với chân giữa để chọn chế độ.

Chế độ LOW là chế độ mà khi mạch Arduino xuất ra tín hiệu LOW (hay giá trị bằng 0) thì sẽ kích role hoạt động, đóng tiếp điểm thường mở.

6. Chế độ HIGH

Chế độ HIGH là chế độ khi Arduino xuất ra tín hiệu HIGH (hay giá trị 1) thì sẽ kích role hoạt động, đóng tiếp điểm thường mở. Thông thường khi mua về thì chế độ mặc định của Role là chế độ HIGH

7. Chân nguồn +

Là chân cấp nguồn +, 5V cho Role, chân này được nối với chân 5V của board Arduino.

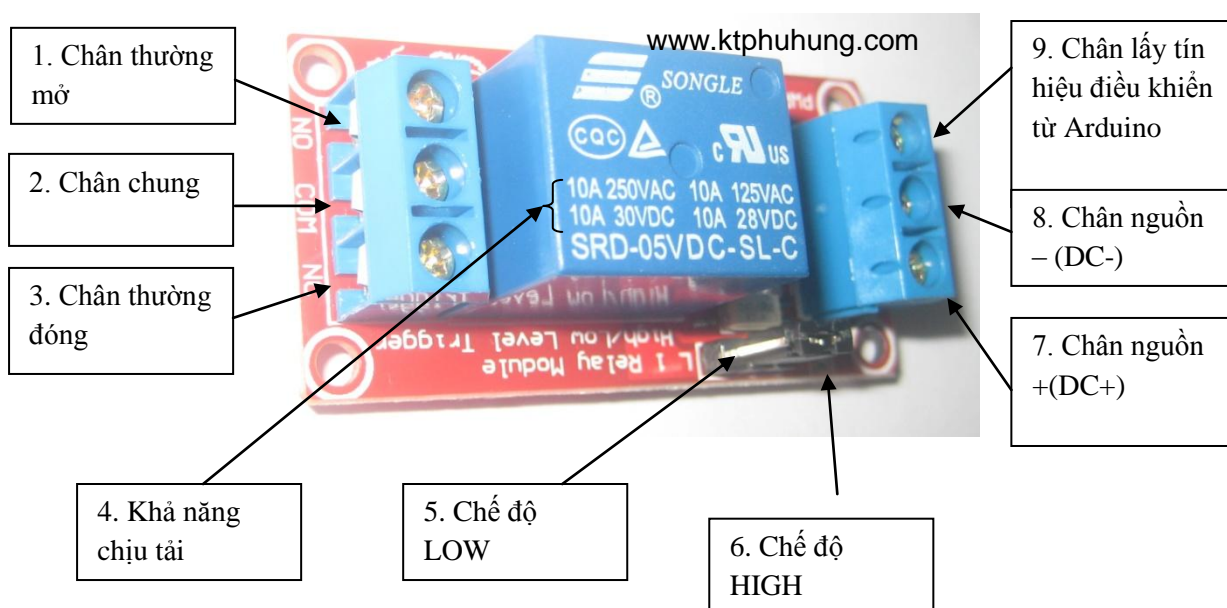
8. Chân nguồn âm:

Là chân nối với nguồn âm, chân này được nối với chân GND của board Arduino

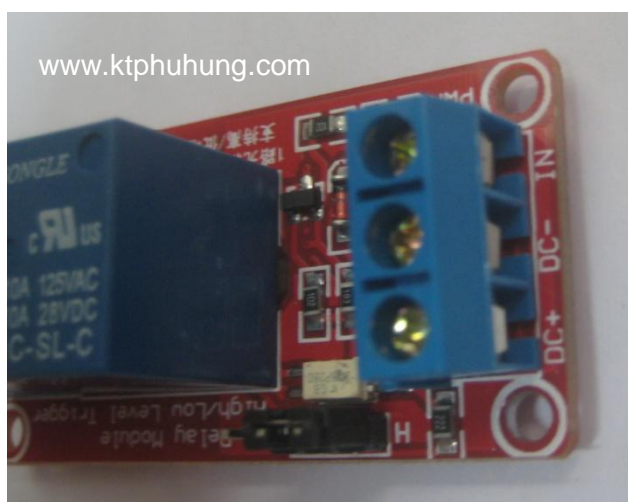
9. Chân lấy tín hiệu điều khiển từ Arduino

Chân này sẽ nối với các chân ra của Arduino nhằm để điều khiển ra role như ý người lập trình mong muốn.

Hình 21 thể hiện rất rõ nét hơn về các chân 7,8,9 trong module Role.



Hình 20. Module role 1 kênh cho Arduino.



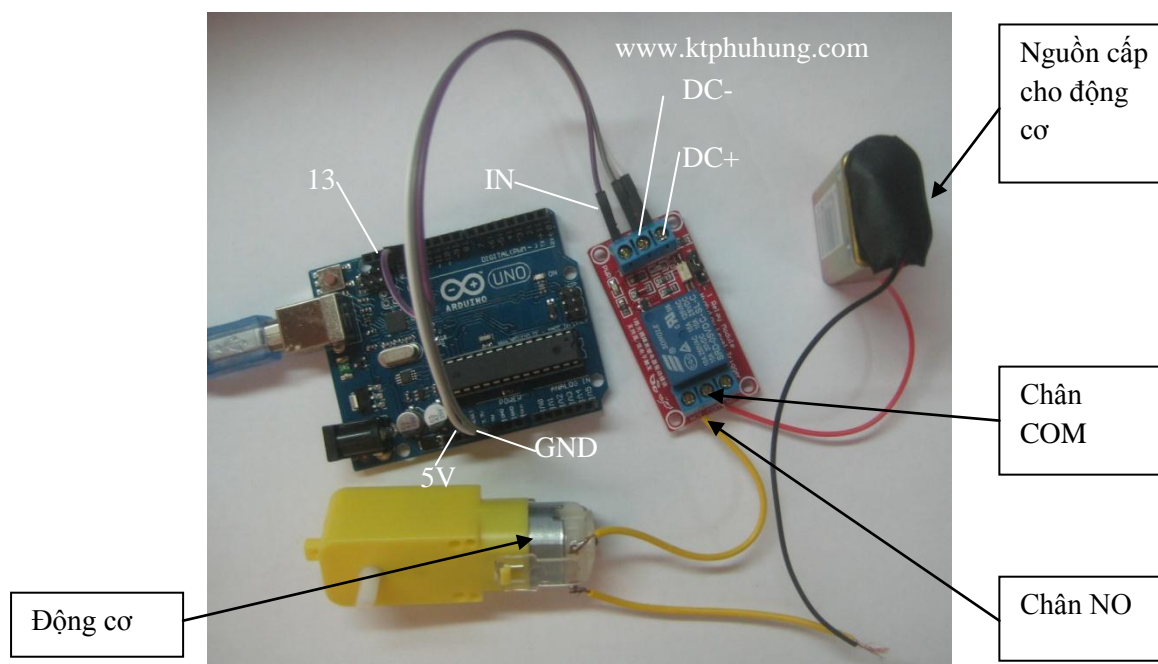
Hình 21. Các chân cấp nguồn và chân lấy tín hiệu từ Role

III. CODE VÍ DỤ

Sau đây sẽ là một ví dụ với Role, code sử dụng là code trong bài đèn led nhấp nháy. Trong ví dụ này chân IN của role sẽ được nối với chân 13, chân DC+ được nối với chân 5V trên Arduino, chân DC- được nối với chân GND trên board Arduino.

Mạch này có thể ứng dụng dùng để tưới lan, trồng nấm, chỉ cần thay đổi thời gian delay thì có thể thay đổi thời gian máy bơm nước chạy.

1. Cách nối dây



Hình 22. Cách nối dây của ví dụ.

2. Mã code

Phần này sử dụng mã code của ví dụ cho đèn led nhấp nháy, role sẽ đóng ngắt luân phiên trong 1 giây. Sau đây là mã code :

```
int led = 13;// khai báo chân led là chân 13
void setup() {
  pinMode(led, OUTPUT); //Thiết lập chân led (chân 13 là chân ra, OUTPUT)
}
void loop() {
  digitalWrite(led, HIGH); // Ra lệnh mở led
  delay(1000);           // Mở trong 1s (1000ms), có thể thay đổi giá trị này
  digitalWrite(led, LOW); // ra lệnh tắt led
  delay(1000);           // Tắt trong 1s (1000ms), có thể thay đổi giá trị này
}
```

3. Kết quả:

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

Động cơ sẽ quay và tắt tuần tự trong 1s, 1s quay, 1 giây tắt. Nguồn điện pin 9V có thể được thay thế bằng bình acquy hoặc điện 220V (khi sử dụng động cơ 220V AC). Trước khi sử dụng role để điều khiển các thiết bị điện 220V, các bạn nên thực hành thử với điện của acquy, pin,... để có thể nắm vững phần này và tránh các sự cố đáng tiếc có thể xảy ra. Hình 23 thể hiện module role hoạt động khi cấp nguồn cho hệ thống, mỗi lần led 13 sáng là đèn led tác động của role cũng sáng theo.



Hình 23. Hệ thống đang hoạt động.

BÀI 3. SỬ DỤNG MẠCH CẦU H

Mạch cầu H là module rất thông dụng khi cần điều khiển tốc độ và chiều quay của động cơ, thường được sử dụng trong các xe tự hành tránh vật cản, hay các robot. Bài viết này hướng dẫn các bạn cách sử dụng mạch cầu H L298N để điều khiển động tốc độ và chiều của động cơ. Mục đích của bài này là giúp các bạn có thể tự kết nối mạch cầu H với động cơ, làm một ví dụ nhỏ về đảo chiều và điều chỉnh tốc độ của động cơ.

1. 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 chịu được dòng tối đa là 2A, điện áp điều khiển từ 5V đến 35V, có thể điều khiển được 2 động cơ. Hình bên dưới là module mạch cầu H L298.

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.

2. Khối điều khiển động cơ A

Khối này có 2 chân nhằm để kết nối với động cơ A

3. Khối điều khiển động cơ B

Khối này cũng có 2 chân nhằm để kết nối với động cơ B

4. 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 lấy nguồn 5V từ mạch cầu H ra để dùng. Thường thì chân này ít sử dụng.

5. Các jump

Có 2 jump tại ENA và ENB khi 2 jump này ở 2 vị trí như trên thì sẽ điều khiển động cơ chạy 100% công suất, ENA được dùng cho động cơ A và ENB được dùng cho động cơ B. Ta có thể rút 2 jump này ra và kết nối các ENA, ENB với các chân băm xung của Arduino. Việc này sẽ giúp người sử dụng điều khiển được tốt độ của động cơ.

6. Các chân IN1, IN2, IN3, IN4

Các chân này dùng để kết nối với chân ra của Arduino nhằm điều khiển chiều của động cơ. IN1, IN2 dùng cho động cơ A và IN3, IN4 được dùng cho động cơ B.

Việc điều khiển chiều của động cơ được thực hiện như sau:

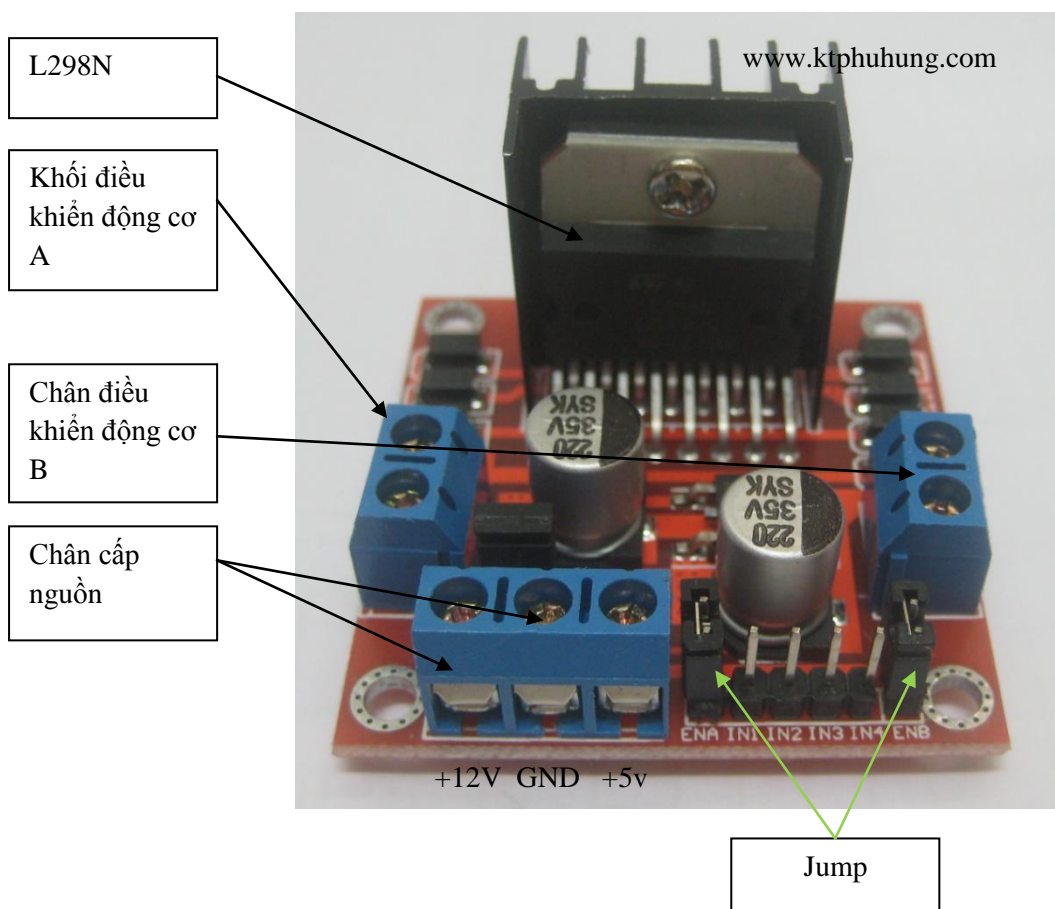
Bảng 3. Bảng điều khiển động cơ A

Chân	Trạng thái	Động cơ A
IN1	LOW	Chiều thuận
IN2	HIGH	
IN1	HIGH	Chiều nghịch
IN2	LOW	
IN1	LOW	Thắng động cơ
IN2	LOW	

Bảng 4. Bảng điều khiển động B

Chân	Trạng thái	Động cơ B
IN3	LOW	Chiều thuận
IN4	HIGH	
IN3	HIGH	Chiều nghịch

IN4	LOW	
IN3	LOW	Thắng động cơ
IN4	LOW	



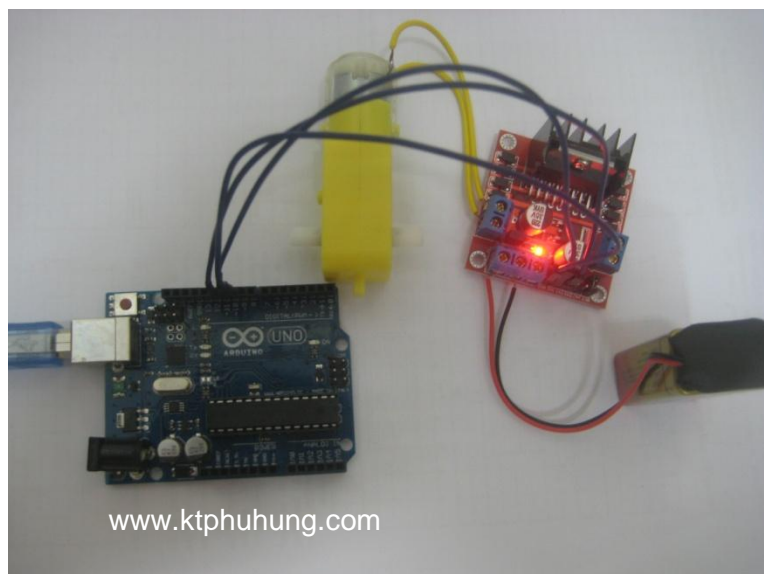
Hình 24. Module mạch cầu H

7. Kết nối mạch cầu H với Arduino

Phần này đưa ra ví dụ dùng module mạch cầu H L298N, như được thể hiện ở hình 25, để điều khiển động cơ có hộp giảm tốc, nguồn cấp có thể là pin 9V, hoặc acquy 12V hoặc Adapter 12V. Về cách nối dây thì mạch này được nối như sau:

- ENA nối với chân 10 trên Arduino
- IN1 nối với chân 11
- IN2 nối với chân 12.

Người dùng có thể thay đổi chân nhưng chú ý là chân ENA nên nối với chân có dấu ~ (chân có thể bấm xung để có thể điều chỉnh tốc độ của động cơ). Về nguồn điện, nếu người dùng sử dụng pin 9V thì nên dùng pin 9V mới vì khi cấp nguồn qua mạch cầu H thì công suất sẽ bị tiêu hao dẫn đến có thể không đủ dòng cấp cho động cơ chạy nếu dùng pin 9V cũ. Người dùng cũng có thể sử dụng bình acquy 12V hoặc adapter 12V cho ví dụ này. Phần kế tiếp sẽ đưa ra một ví dụ về đảo chiều, điều chỉnh tốc độ động cơ dùng mạch cầu H.



Hình 25. Một ví dụ về kết nối module mạch cầu H với Arduino

8. Mã code:

Sau đây là một ví dụ về dùng mạch cầu H để đảo chiều và tốc độ động cơ. Để đảo chiều động cơ DC (động cơ điện một chiều) ta cần đảo cực cấp nguồn cho động cơ. Để thay đổi tốc độ, ta thay đổi xung cấp cho ENA, Arduino sử dụng chip AVR 328, đây là chip dòng 8 bit, nên khả năng băm xung của nó là từ 0 đến 255 (256 xung), tức là có thể băm xung từ 0 đến 255 (256 xung). Đây là ví dụ cho điều khiển 1 động cơ, module mạch cầu H có thể điều khiển tối đa 2 động cơ, động cơ còn lại được điều khiển tương tự.

```
//MỘT VÍ DU VỀ SỬ DỤNG MẠCH CẦU H
//NGƯỜI VIẾT: HUỖNH MINH PHÚ.
#define ENA 10 //Xác định ENA là chân 10
#define IN1 11 // Xác định IN1 là chân 11
#define IN2 12 // Xác định IN2 chân 12
void setup()
{
  pinMode(ENA, OUTPUT); //Xác định kiểu chân ENA là chân xuất tín hiệu ra (OUTPUT)
  pinMode(IN1, OUTPUT); // Xác định kiểu chân IN1 là chân xuất tín hiệu ra (OUTPUT)
  pinMode(IN2, OUTPUT); // Xác định kiểu chân IN2 là chân xuất tín hiệu ra (OUTPUT)
}
void loop()
{
  //Ban đầu cho động cơ thẳng (phanh) lại
  digitalWrite(IN1, LOW); //Cho chân IN1 ở trạng thái LOW
  digitalWrite(IN2, LOW); //Cho chân IN2 ở trạng thái LOW
  analogWrite(ENA, 255); //Tốc độ là 100% (255)
  delay(1000); //Chờ 1 giây
  //Cho động cơ chạy tới
  digitalWrite(IN1, HIGH); //Cho chân IN1 ở trạng thái HIGH
  digitalWrite(IN2, LOW); //Cho chân IN2 ở trạng thái LOW
  analogWrite(ENA, 120); //Tốc độ là 120/255
```

TỰ HỌC NHANH ARDUINO CHO NGƯỜI MỚI BẮT ĐẦU

```
delay(3000); //cho chạy trong 3 giây (chờ trong 3 giây)
//Dừng lại để đảo chiều, nếu không dừng lại mà đảo chiều vẫn được, nhưng về lâu dài không tốt
cho động cơ
digitalWrite(IN1, LOW); //Cho chân IN1 ở trạng thái LOW
digitalWrite(IN2, LOW); //Cho chân IN2 ở trạng thái LOW
analogWrite(ENA, 255); // Tốc độ là 100% (255)
delay(1000); //Chờ trong 1 s
//Cho động cơ chạy lui, trạng thái của IN1 và IN2 trong phần chạy lui sẽ ngược với trạng thái
chạy tới.
digitalWrite(IN1, LOW);//Cho chân IN1 ở trạng thái LOW
digitalWrite(IN2, HIGH); //Cho chân IN2 ở trạng thái HIGH
analogWrite(ENA, 255); //Chạy 100% tốc độ (255 xung)
delay(3000); //Chạy trong 3s (chờ 3s)
}
```

BÀI 4. SỬ DỤNG MÀN HÌNH LCD

Màn hình LCD là một trong những thiết bị thường được sử dụng trong để hiển thị các thông số trong quá trình sử dụng Arduino. Bài viết này hướng dẫn cách sử dụng màn hình lcd 16x2 thông dụng trên thị trường.

1. Dụng cụ thực hành:

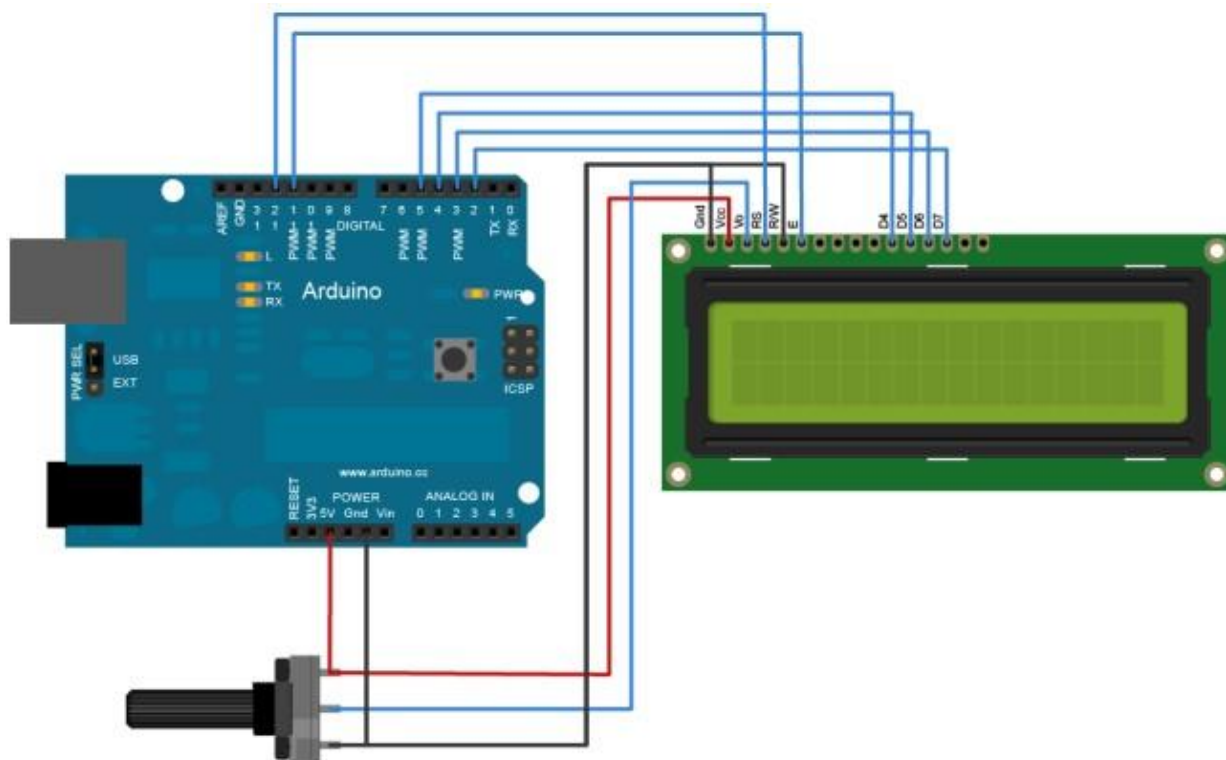
1 mạch Arduino Uno, 1 màn hình lcd 16x2, 1 rào đục đơn, 1 biến trở 10K, 10 dây đục đục, 10 dây đục cái.

2. Màn hình lcd 16x2:

Màn hình lcd 16x2 là loại màn hình có 2 dòng, mỗi dòng có thể viết được 16 ký. Loại màn hình này cần được dùng kèm 1 biến trở 10K để chỉnh độ tương phản của chữ so với nền. Nếu không có biến trở này thì có thể không thấy được chữ hiển thị.

3. Cách nối dây

- Chân VSS (GND) nối với chân GND trên board Arduino
- Chân VDD(tương ứng chân VCC trong hình 26) nối với chân 5V trên board Arduino
- Chân V0 (chân chỉnh độ tương phản) nối với chân giữa của biến trở 10K (xem hình 26)
- Chân RS nối với chân số 12
- Chân RW nối với 1 chân của biến trở 10K, chân này của biến trở cũng được nối với chân GND
- Chân E (viết tắt của Enable) nối với chân số 11
- Chân D4 nối với chân số 5
- Chân D5 nối với chân số 4
- Chân D6 nối với chân số 3
- Chân D7 nối với chân số 2
- Ngoài ra chân còn lại của biến trở cũng được nối với nguồn 5V như được minh họa ở hình 26



Hình 26. Cách nối dây cho màn hình LCD 16x2 với Arduino Uno

4. Mã code của chương trình

Sau đây là một mã code mà các bạn có thể test màn hình LCD:

```
#include <LiquidCrystal.h>//Sử dụng thư viện màn hình LCD, thư viện này có sẵn trong
Arduino
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);//Sử dụng thư viện với các chân đã được kết nối như phần
trên
void setup() {
  // Thiết lập số lượng hàng và cột của LCD:
  lcd.begin(16, 2);//có 2 hàng, 16 cột (16 ký tự)
  // Print a message to the LCD.
  lcd.print("Toi yeu Arduino");//In lên màn hình LCD dòng chữ “Tôi yêu Arduino
//Dòng chữ này sẽ xuất hiện ở dòng đầu tiên, trong phần mềm tính là dòng số 0
}
//Lập trình in số giây từ khi bắt đầu chương trình lên LCD
void loop() {
  // Thiết lập vị trí in là ở cột số 0, dòng 1
  //Chú ý: dòng 1 thực chất là dòng 2 vì phần mềm tính từ 0
  lcd.setCursor(0, 1);
  // In số giây từ khi bắt đầu hoặc từ khi reset
  lcd.print(millis()/1000);
}
```