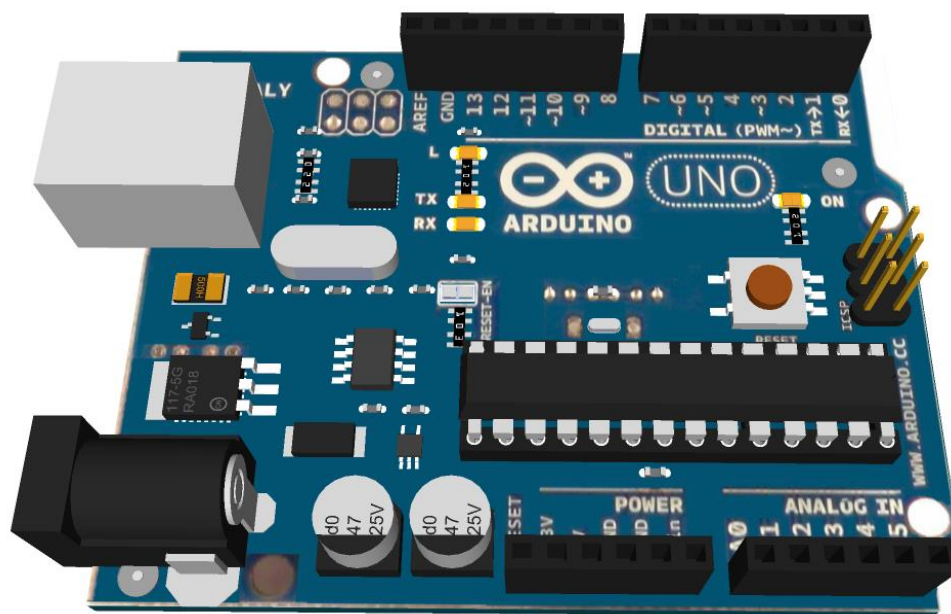


HỌC VIỆN HÀNG KHÔNG VIỆT NAM
KHOA ĐIỆN TỬ VIỄN THÔNG



HƯỚNG DẪN SỬ DỤNG CƠ BẢN ARDUINO



TP.HCM, Tháng 5, Năm 2014.

MỤC LỤC:

Lời nói đầu.

Chương 1: Tổng quan về Arduino Uno.....	1
1. Tổng quan.....	1
2. Sơ đồ chân của Arduino.....	2
Chương 2: Cài đặt chương trình Arduino IDE và Driver cho Arduino	4
1. Cài đặt chương trình Arduino IDE	4
2. Cài đặt Driver	5
3. Arduino IDE	7
Chương 3: Hướng dẫn cài đặt bản mô phỏng Arduino trên Proteus.....	11
Chương 4: Giao tiếp Arduino với một số linh kiện điện tử.....	13
1) Project 1: Led nhấp nháy.....	13
2) Project 2 : Đèn sáng khi nhấn phím.....	18
3) Project 3 : Led sáng dần từ led 1 đến led 10 và ngược lại.....	21
4) Project 4 : Led sáng dần từ led 1 đến led 10 và ngược lại thời gian delay thay đổi được.....	24
5) Project 5: Điều khiển tốc độ động cơ bằng PWM.....	27
6) Project 6 : Điều khiển động cơ bằng L293D.....	31
7) Project 7: Giao tiếp Arduino với LCD 16x2.....	34
8) Project 8: Giao tiếp với máy tính.....	47
9) Project 9. Đo nhiệt độ môi trường dùng LM35D hiển thị LCD và Serial Monitor.....	49
10) Project 10: Giao tiếp Arduino với Servo motor.....	54
Tài liệu tham khảo.....	56

Lời Nói Đầu.

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. Arduino thực sự đã gây sóng gió trên thị trường người dùng trên toàn thế giới trong vài năm gần đây, 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. Tuy nhiên tại Việt Nam Arduino vẫn còn chưa được biết đến nhiều, các tài liệu liên quan đến nó vẫn còn rất hạn chế. Được sự giới thiệu và chỉ dẫn của thầy **Nguyễn Thanh Dũng**, sau một thời gian tìm hiểu tác giả đã biên soạn tài liệu “**HƯỚNG DẪN SỬ DỤNG CƠ BẢN ARDUINO**”. Trong tài liệu này cung cấp cho bạn đọc một lượng kiến thức cơ bản nhất về Arduino cũng như các ứng dụng thực tế của nó.

Tài liệu gồm có các nội dung sau:

Chương 1: Tổng quan về Arduino Uno.

Chương 2: Cài đặt chương trình Arduino IDE và Driver cho Arduino.

Chương 3: Hướng dẫn cài đặt bản mô phỏng Arduino trên Proteus.

Chương 4: Giao tiếp Arduino với một số linh kiện điện tử.

Khi biên soạn, tác giả đã tham khảo một số tài liệu nước ngoài để tài liệu vừa đảm bảo về mặt nội dung vừa có thể tiếp cận được với bạn đọc.

Khi viết tác giả đã cố gắng để tài liệu được hoàn chỉnh nhất song chắc chắn không tránh khỏi sai sót, vì vậy rất mong nhận được sự góp ý của bạn đọc.

Mọi ý kiến đóng góp xin liên hệ: trungtin.vaa@gmail.com

Tác giả

SV: NGUYỄN TRUNG TÍN

Chương 1: Tổng quan về Arduino Uno.

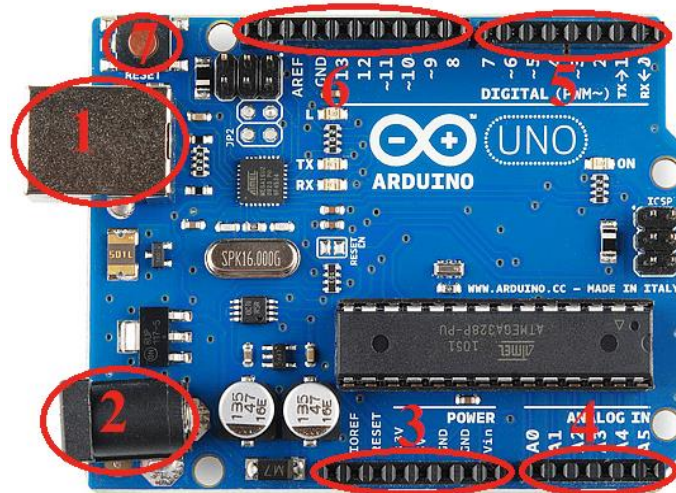
1. Tổng quan.

Arduino thật ra 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.

Arduino Uno là sử dụng chip Atmega328. Nó có 14 chân digital I/O, 6 chân đầu vào (input) analog, thạch anh dao động 16Mhz. Một số thông số kỹ thuật như sau :

Chip	ATmega328
Điện áp cấp nguồn	5V
Điện áp đầu vào (input) (kiến nghị)	7-12V
Điện áp đầu vào(giới hạn)	6-20V
Số chân Digital I/O	14 (có 6 chân điều chế độ rộng xung PWM)
Số chân Analog (Input)	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32KB (ATmega328) với 0.5KB sử dụng bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Xung nhịp	16 MHz

2. Sơ đồ chân của Arduino.



Hình 1: Arduino Uno.

a) USB (1).

Arduino sử dụng cáp USB để giao tiếp với máy tính. Thông qua cáp USB chúng ta có thể Upload chương trình cho Arduino hoạt động, ngoài ra USB còn là nguồn cho Arduino.

b) Nguồn (2 và 3).

Khi không sử dụng USB làm nguồn thì chúng ta có thể sử dụng nguồn ngoài thông qua jack cắm 2.1mm (cực dương ở giữa) hoặc có thể sử dụng 2 chân V_{in} và GND để cấp nguồn cho Arduino.

Bo mạch hoạt động với nguồn ngoài ở điện áp từ 5 – 20 volt. Chúng ta có thể cấp một áp lớn hơn tuy nhiên chân 5V sẽ có mức điện áp lớn hơn 5 volt. Và nếu sử dụng nguồn lớn hơn 12 volt thì sẽ có hiện tượng nóng và làm hỏng bo mạch. Khuyết cáo các bạn nên dùng nguồn ổn định là 5 đến dưới 12 volt.

Chân 5V và chân 3.3V (Output voltage) : các chân này dùng để lấy nguồn ra từ nguồn mà chúng ta đã cung cấp cho Arduino. Lưu ý : không được cấp nguồn vào các chân này vì sẽ làm hỏng Arduino.

GND: chân mass.

c) Chip Atmega328.

Chip Atmega328 Có 32K bộ nhớ flash trong đó 0.5k sử dụng cho bootloader. Ngoài ra còn có 2K SRAM, 1K EEPROM.

d) Input và Output (4, 5 và 6).

Arduino Uno có 14 chân digital với chức năng input và output sử dụng các hàm *pinMode()*, *digitalWrite()* và *digitalRead()* để điều khiển các chân này tôi sẽ đề cập chúng ở các phần sau.

Cũng trên 14 chân digital này chúng ta còn một số chân chức năng đó là:

Serial : chân 0 (Rx), chân 1 (Tx). Hai chân này dùng để truyền (Tx) và nhận (Rx) dữ liệu nối tiếp TTL. Chúng ta có thể sử dụng nó để giao tiếp với cổng COM của một số thiết bị hoặc các linh kiện có chuẩn giao tiếp nối tiếp.

PWM (pulse width modulation): các chân 3, 5, 6, 9, 10, 11 trên bo mạch có dấu “~” là các chân PWM chúng ta có thể sử dụng nó để điều khiển tốc độ động cơ, độ sáng của đèn...

SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK), các chân này hỗ trợ giao tiếp theo chuẩn SPI.

I2C: Arduino hỗ trợ giao tiếp theo chuẩn I2C. Các chân A4 (SDA) và A5 (SCL) cho phép chúng ta giao tiếp giữa Arduino với các linh kiện có chuẩn giao tiếp là I2C.

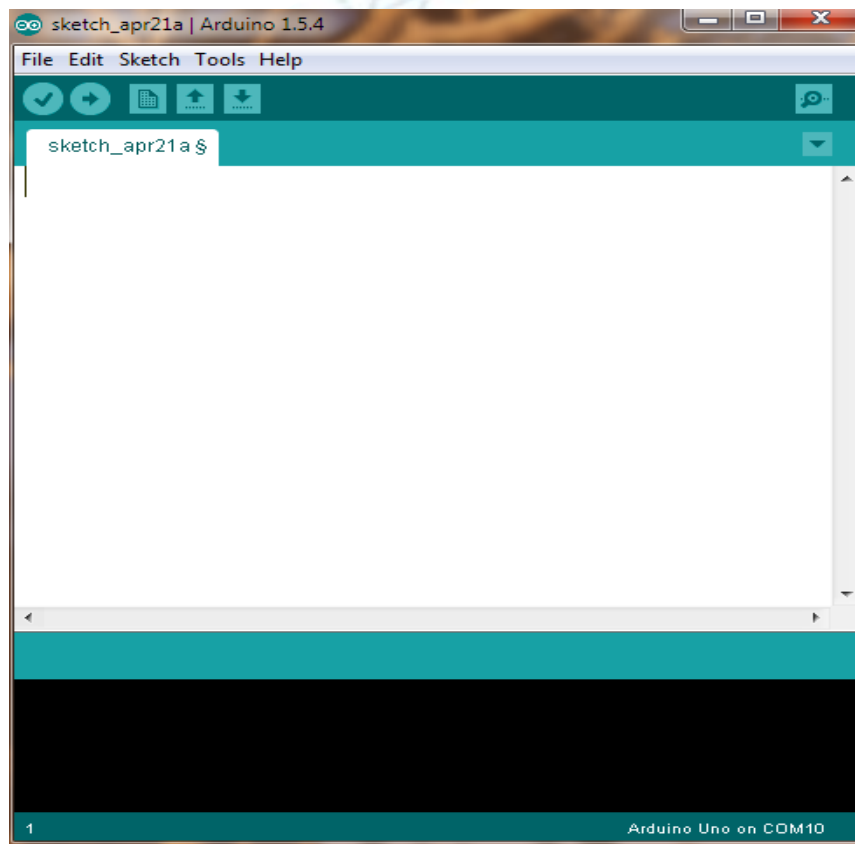
e) Reset (7): dùng để reset Arduino.

Chương 2: Cài đặt chương trình Arduino IDE và Driver cho Arduino

1. Cài đặt chương trình Arduino IDE

Các bạn truy cập vào trang web <http://arduino.cc/en/Main/Software> và tải về chương trình Arduino IDE phù hợp với hệ điều hành của máy mình bao gồm Windows, Mac OS hay Linux. Đối với Windows có bản cài đặt (.exe) và bản Zip, đối với Zip thì chỉ cần giải nén và chạy chương trình không cần cài đặt.

Sau khi cài đặt xong thì giao diện chương trình như sau:



Hình 2: Arduino IDE

2. Cài đặt Driver

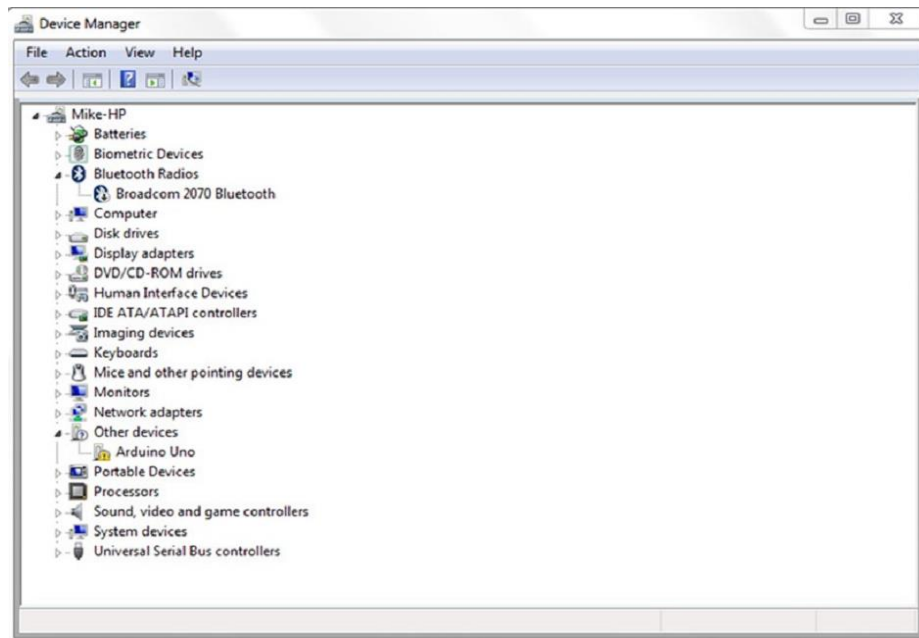
Sử dụng cáp USB kết nối Arduino với máy tính, lúc này bạn sẽ thấy đèn led power của bo sáng. Máy tính sẽ nhận dạng thiết bị và bạn sẽ nhận được thông báo:

“Device driver software was not successfully installed”



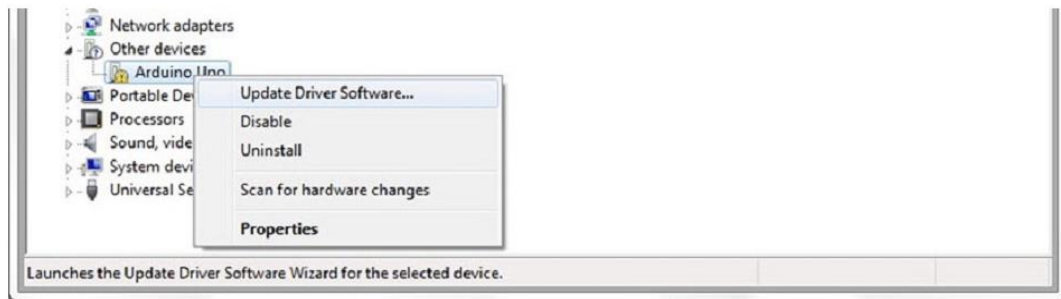
Hình 3: Driver Software Installation.

Bây giờ bạn click vào *Start Menu* chọn *Control Panel* kể đến chúng ta chọn *System and Security*, click *System* và sau đó chọn *Device Manager*.



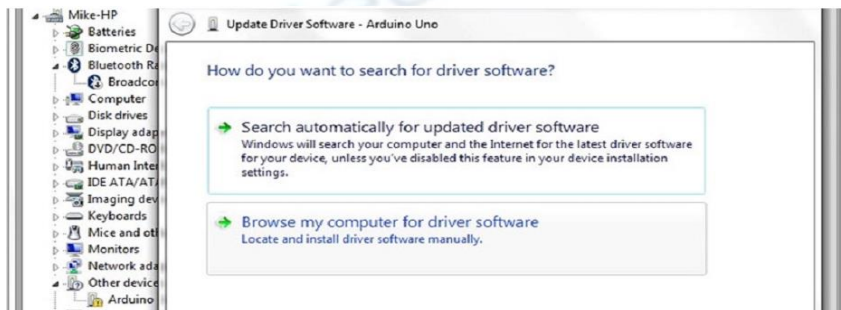
Hình 4: Device Manager.

Chúng ta sẽ thấy cảnh báo màu vàng thiếu driver trên Arduino. Click chuột phải trên Arduino Uno icon sau đó chọn “Update Driver Software”



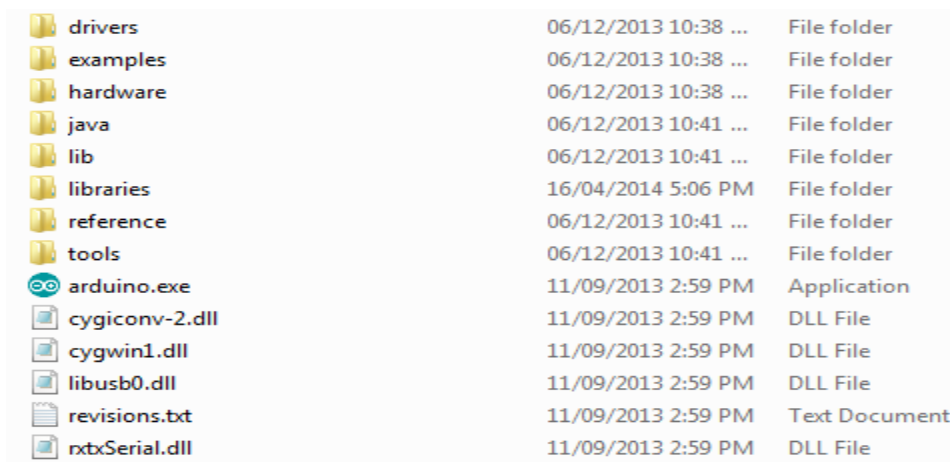
Hình 5: Right click và chọn ”Update Driver Software”

Chọn “Browse my computer for driver software”.



Hình 6: Click chọn “Browse my computer for driver software”

Chọn đường dẫn tới folder “driver” nơi mà phần mềm Arduino được lưu trữ.

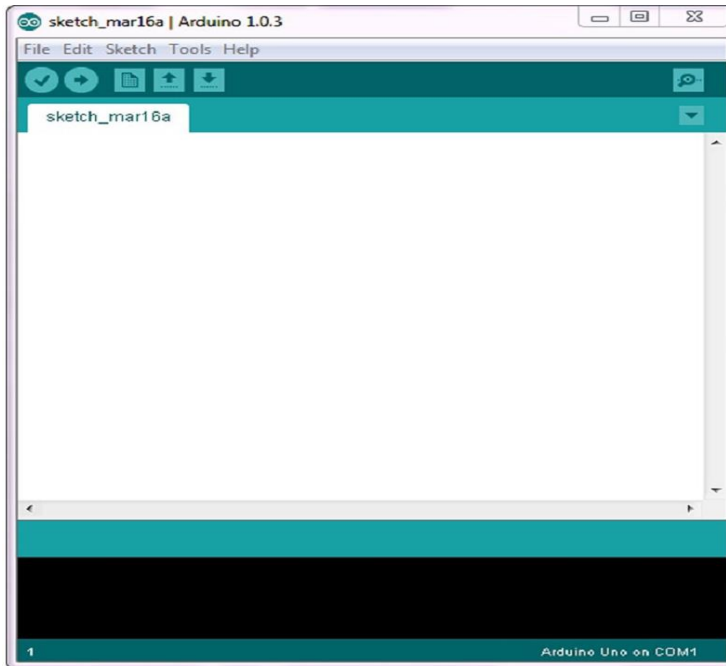


Hình 6: Driver

Click “Next” Windown tự động cài đặt driver, qua trình cài đặt driver hoàn tất.

3. Arduino IDE

Arduino IDE là nơi để soạn thảo code, kiểm tra lỗi và upload code cho arduino







Hình 7: Arduino IDE.

a) Arduino Toolbar: có một số button và chức năng của chúng như sau :



Hình 8: Arduino Toolbar.

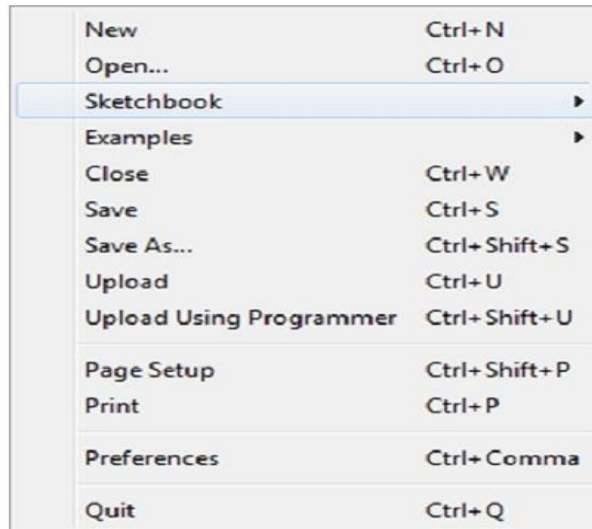
- Verify : kiểm tra code có lỗi hay không 
- Upload: nạp code đang soạn thảo vào Arduino 
- New, Open, Save : Tạo mới, mở và Save sketch 
- Serial Monitor : Đây là màn hình hiển thị dữ liệu từ Arduino gửi lên máy tính 

b) Arduino IDE Menu:



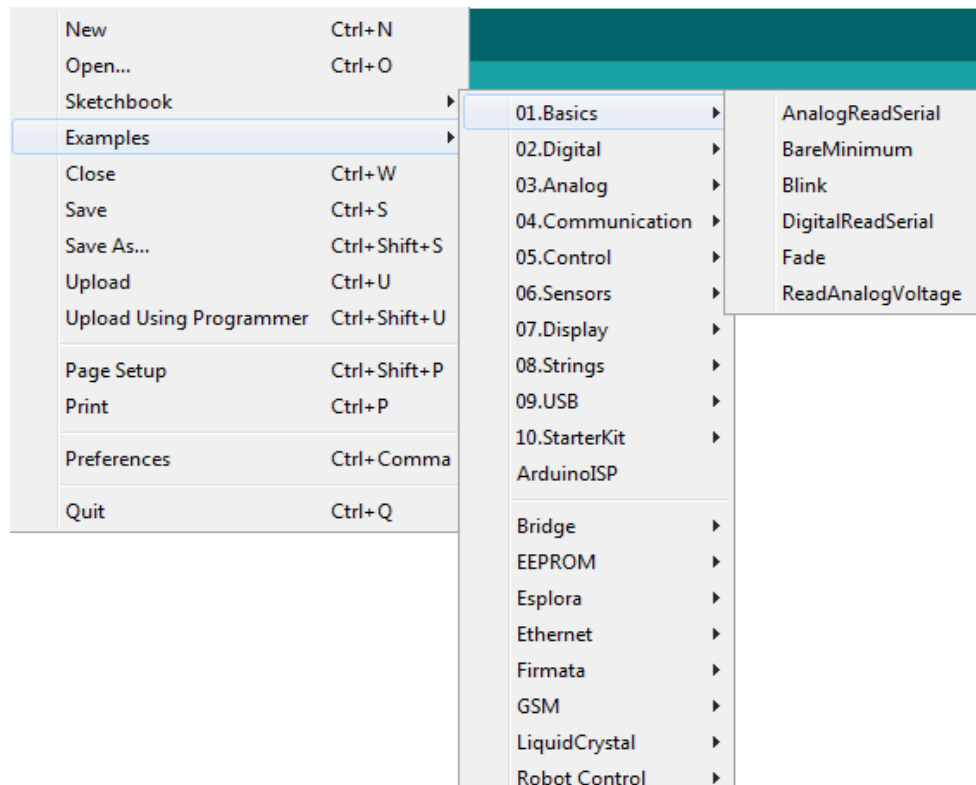
Hình 9: IDE Menu

➤ File menu:



Hình 10: File menu.

Trong file menu chúng ta quan tâm tới mục Examples đây là nơi chứa code mẫu ví dụ như: cách sử dụng các chân digital, analog, sensor ...



Hình 11: Click Examples.

➤ Edit menu:

Undo	Ctrl+Z
Redo	Ctrl+Y
<hr/>	
Cut	Ctrl+X
Copy	Ctrl+C
Copy for Forum	Ctrl+Shift+C
Copy as HTML	Ctrl+Alt+C
Paste	Ctrl+V
Select All	Ctrl+A
<hr/>	
Comment/Uncomment	Ctrl+Slash
Increase Indent	Ctrl+Close Bracket
Decrease Indent	Ctrl+Open Bracket
<hr/>	
Find...	Ctrl+F
Find Next	Ctrl+G
Find Previous	Ctrl+Shift+G
Use Selection For Find	Ctrl+E

Hình 11: Edit menu

➤ Sketch menu

Verify / Compile	Ctrl+R
<hr/>	
Show Sketch Folder	Ctrl+K
Add File...	
Import Library...	▶

Hình 12: Sketch menu

Trong Sketch menu :

- ✓ Verify/ Compile : chức năng kiểm tra lỗi code.
- ✓ Show Sketch Folder : hiển thị nơi code được lưu.
- ✓ Add File : thêm vào một Tập code mới.
- ✓ Import Library : thêm thư viện cho IDE

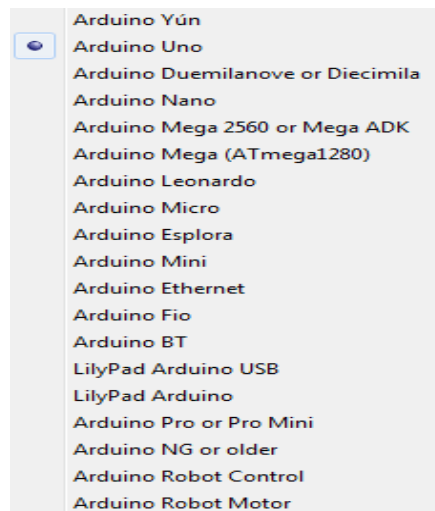
➤ Tool menu:



Hình 13: Tool menu.

Trong Tool menu ta quan tâm các mục Board và Serial Port

Mục Board : các bạn cần phải lựa chọn bo mạch cho phù hợp với loại bo mà bạn sử dụng nếu là Arduino Uno thì phải chọn như hình:



Hình 14: Chọn Board

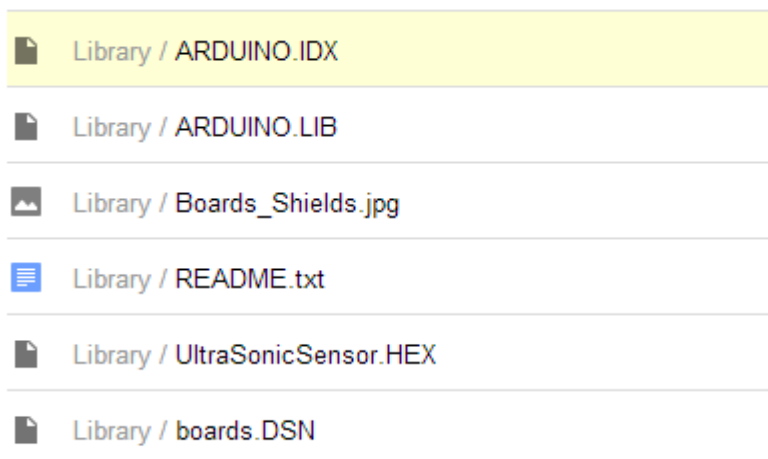
Nếu các bạn sử dụng loại bo khác thì phải chọn đúng loại bo mà mình đang có nếu sai thì code Upload vào chip sẽ bị lỗi.

Serial Port: đây là nơi lựa chọn cổng Com của Arduino. Khi chúng ta cài đặt driver thì máy tính sẽ hiện thông báo tên cổng Com của Arduino là bao nhiêu, ta chỉ việc vào Serial Port chọn đúng cổng Com để nạp code, nếu chọn sai thì không thể nạp code cho Arduino được.

Chương 3: Hướng dẫn cài đặt bản mô phỏng Arduino trên Proteus.

Để mô phỏng được Arduino trên proteus thì chúng ta cần phải download thư viện arduino cho proteus. Để có được thư viện này các bạn cần truy cập vào trang web:

<http://blogembarcado.blogspot.com/search/label/Proteus>



Hình 15: Thư viện mô phỏng Arduino.

Sau khi download về các bạn chép 2 file ARDUINO.IDX và ARDUINO.LIB vào thư mục:

Proteus 7:

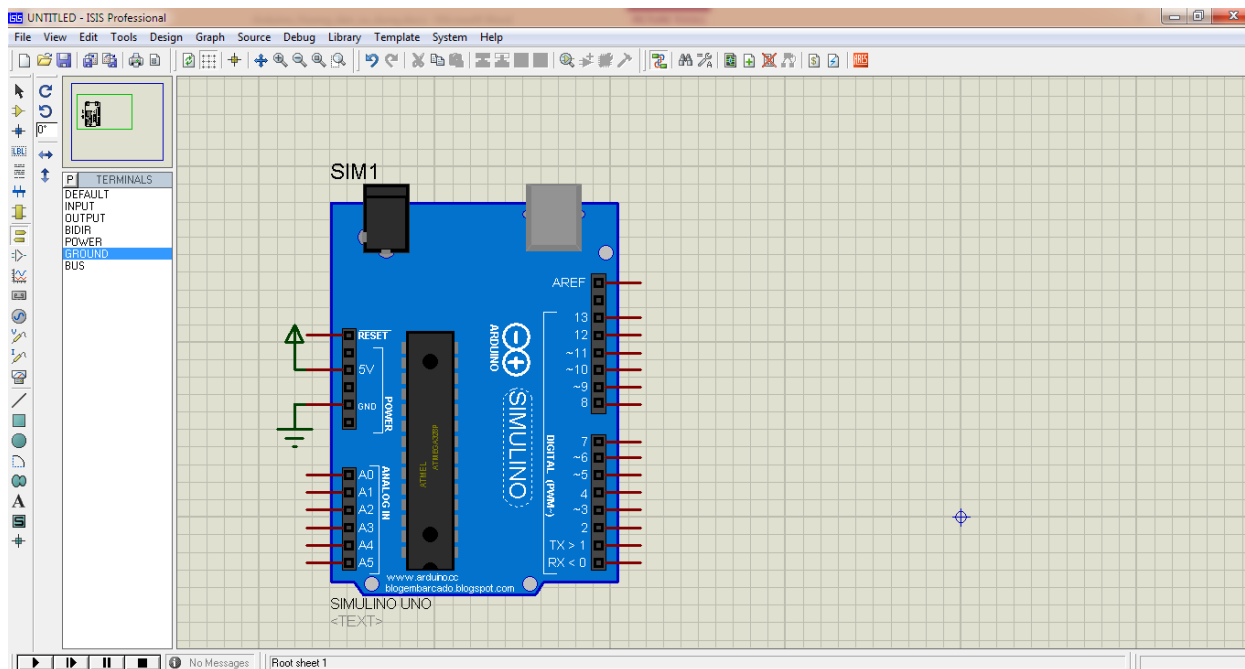
C:\Program Files (hoặc x86) \Labcenter Electronics\Proteus 7 Professional\LIBRARY

Proteus 8:

C:\Program Files (hoặc x86) \Labcenter Electronics\Proteus 8 professional\Data\LIBRARY

Trong thư viện này hỗ trợ 5 loại board Arduino khác nhau trong đó gồm có Arduino Uno, MEGA, NANO, LILYPAD và UNO SMD và một cảm biến siêu âm Untrasonic.

Sau khi chép xong chúng ta khởi động Proteus lên vào thư viện linh kiện bằng cách bấm phím P và gõ từ khoá là ARDUINO chúng sẽ hiện ra danh sách các board hiện có ở đây tôi chọn Arduino Uno.



Hình 16: Mô phỏng Arduino bằng Proteus.

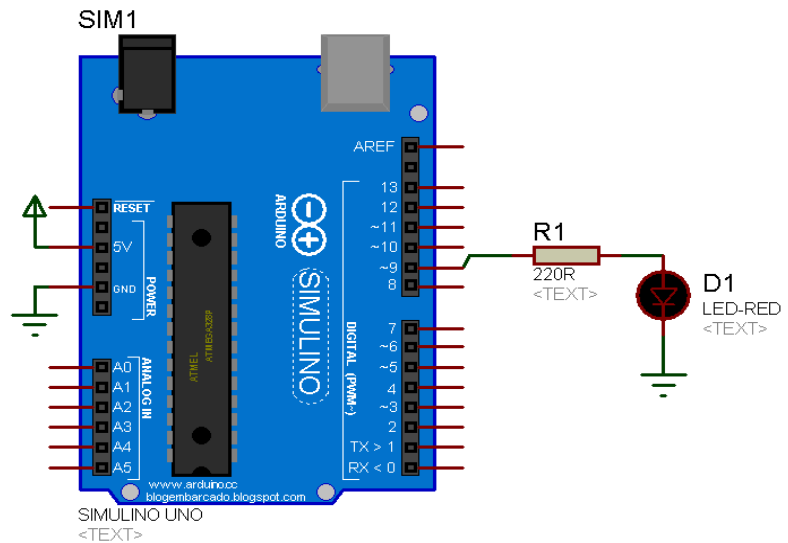
Lưu ý chúng ta cần phải cấp nguồn vào 2 chân 5V và Gnd trên mạch như hình trên.

Chương 4: Giao tiếp Arduino với một số linh kiện điện tử.

1) Project 1: Led nhấp nháy.

Sau đây tôi sẽ tạo một project nhấp nháy led thời gian delay là 1 giây sử dụng proteus để mô phỏng.

✓ Sơ đồ mạch:




Hình 17: Led nhấp nháy.

✓ Code chương trình.

```
int ledPin = 9;

void setup() {
    pinMode(ledPin, OUTPUT);
}

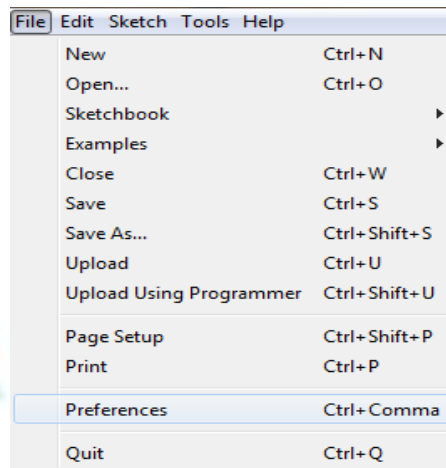
void loop() {
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}
```

Sau khi gõ code vào chương trình soạn thảo bạn cần click và  để kiểm tra lỗi.

Tạo File Hex.

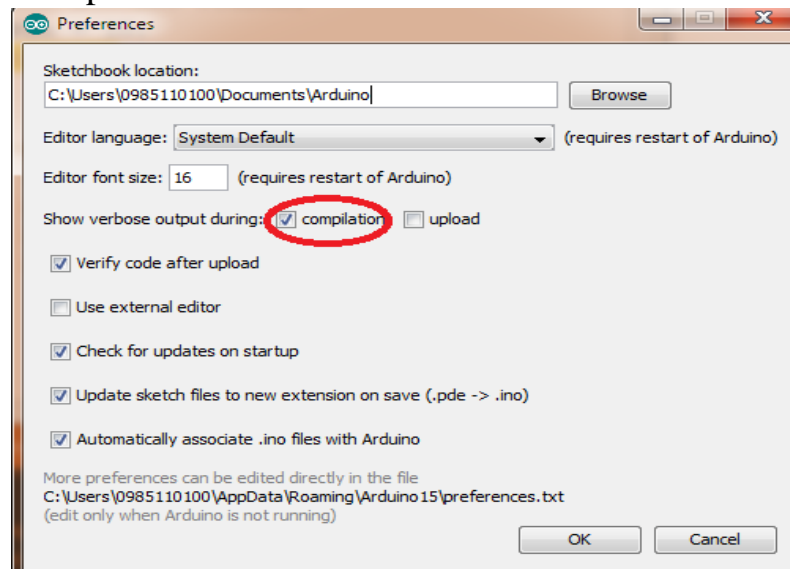
Chúng ta cần phải có file Hex để cung cấp cho proteus và khi bấm play chương trình mới hoạt động được. Cách tạo file Hex trên Arduino IDE như sau:

Click vào File chọn Preferences.




Hình 18: Click Preferences.

Các bạn check vào compilation và OK.



Hình 19: Check compilation.

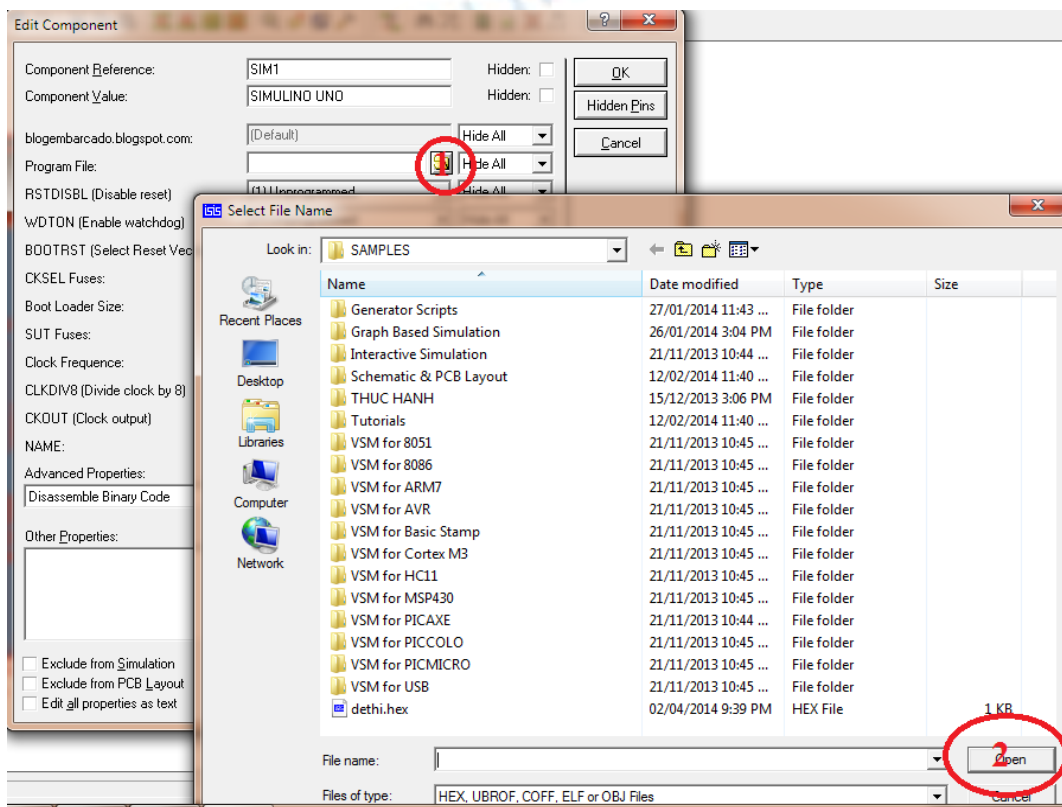
Sau đó tiếp tục bấm 

Chương trình sẽ tự động build một file hex được lưu ở đường dẫn như hình dưới

```
C:\Users\098511~1\AppData\Local\Temp\build9118448727674380763.tmp/Blink.cpp.hex  
  
Sketch uses 1,116 bytes (3%) of program storage space. Maximum is 32,256 bytes.  
Global variables use 11 bytes (0%) of dynamic memory, leaving 2,037 bytes for local variables. Maximum is 2,048 bytes.  
11  
Arduino Uno on COM10
```

Hình 20: Đường dẫn chứa file hex.

Các bạn chép file hex ra một thư mục nào đó sau đó mở proteus lên và double click vào Arduino Uno.



Hình 21: Add file Hex cho Proteus.

Bấm vào vị trí số 1 và chọn nơi lưu file hex ở trên chọn tiếp Open, OK và Play.

Các bạn sẽ thấy led nhấp nháy tắt và sáng thời gian delay là 1s.

- ✓ Giải thích chương trình.

```
int ledPin = 9;
```

Khai báo một giá trị biến integer là *ledPin = 9*.

```
void setup() {  
  
pinMode(ledPin, OUTPUT); }
```

Trong Arduino sketch cần phải có hàm *setup()* và *loop()* nếu không có thì chương trình báo lỗi. Hàm *Setup()* chỉ chạy một lần kể từ khi bắt đầu chương trình. Hàm này có chức năng thiết lập chế độ vào, ra cho các chân digital hay tốc độ baud cho giao tiếp Serial...

Cấu trúc của hàm *pinMode()* là như sau:

```
pinMode(pin, Mode);
```

pin : là vị trí chân digital.

Mode: là chế độ vào (INPUT), ra (OUTPUT).

Lệnh tiếp theo.

```
pinMode(ledPin, OUTPUT);
```

Lệnh này thiết lập chân số 9 trên board là chân ngõ ra (*OUTPUT*). Nếu không khai báo “ *int ledPin = 9;* ” thì bạn có thể viết cách sau nhưng ý nghĩa không thay đổi:

```
pinMode(9, OUTPUT);
```

Bắt buộc khai báo một hàm *loop()* trong Arduino IDE. Hàm này là vòng lặp vô hạn

```
void loop() {  
  
digitalWrite(ledPin, HIGH);  
  
delay(1000);  
  
digitalWrite(ledPin, LOW);
```

```
    delay(1000);  
}
```

Tiếp theo ta sẽ phân tích hàm *digitalWrite(ledPin, HIGH)*; lệnh này có ý nghĩa là xuất ra chân digital có tên là *ledPin* (chân 9) mức cao (*HIGH*), mức cao tương ứng là 5 volt.

```
    delay(1000);
```

Lệnh này tạo một khoảng trễ với thời gian là 1 giây. Trong hàm *delay()* của IDE thì 1000 tương ứng với 1 giây.

```
    digitalWrite(ledPin, LOW);
```

Cũng giống như *digitalWrite(ledPin, HIGH)*; lệnh này xuất ra chân *ledPin* mức thấp (*LOW*) tức là 0 volt.

Và tiếp tục là một hàm *delay()*.

Như vậy chúng ta có thể thấy chương trình sẽ thực hiện tắt sáng led liên tục không ngừng trừ khi ta ngắt nguồn.