

Giới thiệu Arduino Uno

Tuần 4: Lược thuyết Arduino

Vi điều khiển/Microcontroller là gì?

- Một vi điều khiển xem như máy tính nhỏ trên 1 chip bao gồm: vi xử lý, bộ nhớ, vào/ra, v.v.
- Lệnh được “**nhúng/embedded**” bên trong vi điều khiển
- Vi điều khiển thường có kích thước nhỏ, giá rẻ

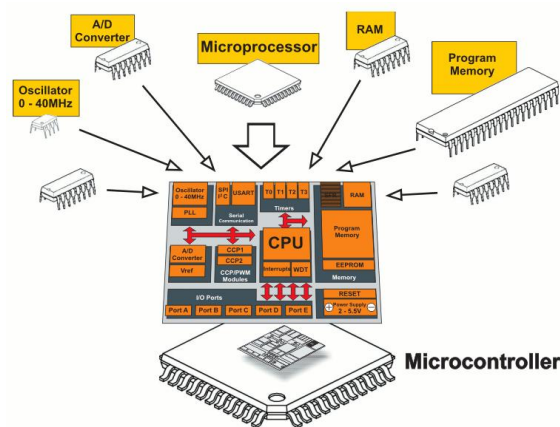
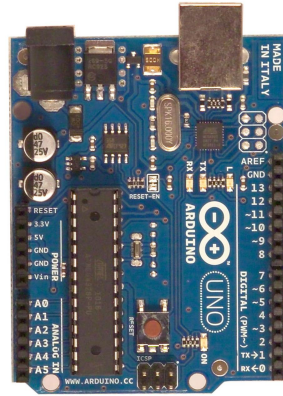


Fig. 9-1 Microcontroller versus Microprocessor

Arduino là gì?

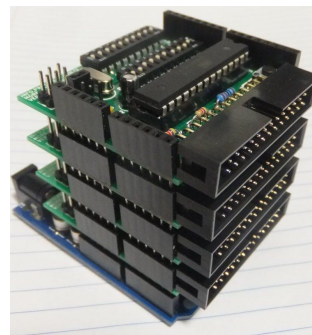
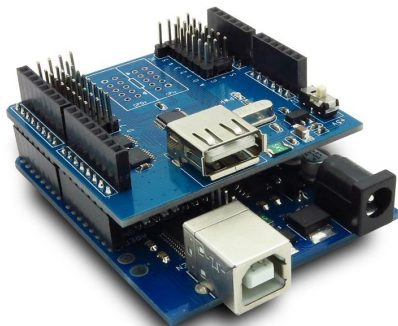
- Arduino là một bo mạch vi điều khiển được dùng để lập trình và có thể giao tiếp với các thiết bị phần cứng khác như máy tính, động cơ, cảm biến...
- Ưu điểm của các bo Arduino:
 - ❖ Môi trường phát triển ứng dụng có thể tiếp xúc nhanh, dễ lập trình,
 - ❖ Cộng đồng phát triển hay hỗ trợ rộng lớn,
 - ❖ Giá thành thấp và phổ biến trên thị trường.
 - ❖ Gắn nhiều bo theo dạng shield hay stack



3

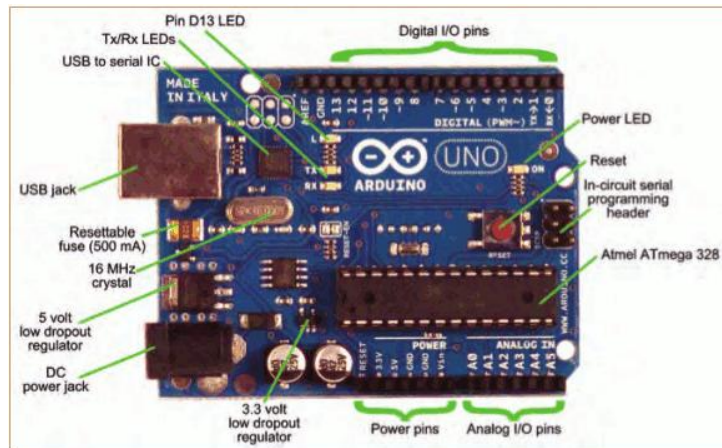
Arduino là gì?

- Dạng bo Arduino Shield



4

Arduino là gì?



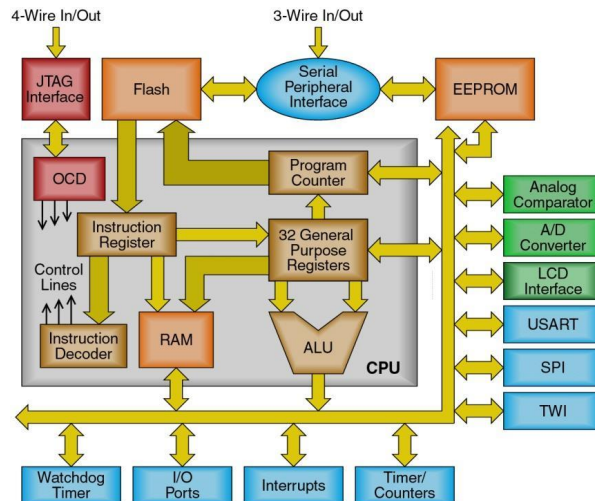
5

Sơ đồ chân Atmel AVR ATmega 328

Arduino function		ATmega328P pin mapping		Arduino function	
reset	PC6	1	28	PC5	analog input 5
digital pin 0 RX	PD0	2	27	PC4	analog input 4
digital pin 1 TX	PD1	3	26	PC3	analog input 3
digital pin 2	PD2	4	25	PC2	analog input 2
digital pin 3 PWM	PD3	5	24	PC1	analog input 1
digital pin 4	PD4	6	23	PC0	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	PB6	9	20	AVCC	AVCC
crystal	PB7	10	19	PB5	digital pin 13
digital pin 5 PWM	PD5	11	18	PB4	digital pin 12
digital pin 6 PWM	PD6	12	17	PB3	digital pin 11
digital pin 7	PD7	13	16	PB2	digital pin 10
digital pin 8	PB0	14	15	PB1	digital pin 9

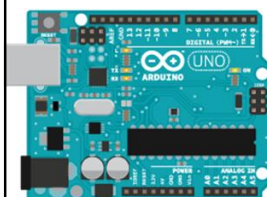
6

Kiến trúc AVR Atmega 328

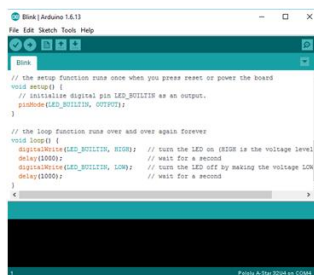


7

Lập trình Arduino



Bo mạch phần cứng



Công cụ phần mềm lập trình



Cộng đồng mã nguồn mở

8

Cấu hình của Arduino Uno

Chip vi xử lý	ATmega328
Điện áp cung cấp	5-12V
Số chân I/O số	14 (có 6 chân điều chế độ rộng xung PWM)
Số chân Analog (Input)	6
Dòng điện DC cung cấp I/O	40 mA
Dòng điện DC cung cấp cho chân 3.3V	50 mA
Kích thước của bộ nhớ Flash	32KB (ATmega328) Với 0.5KB sử dụng bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Xung nhịp (clock)	16 MHz
Chân đèn LED	Chân 13

9

Cấu hình của Arduino Uno

➤ Nguồn cung cấp từ 5 đến 12VDC

Có thể cung cấp cho bo Arduino Uno theo các 3 cách sau:

- Cung cấp theo cổng USB (5V)
- Cung cấp theo jack cắm 2.1mm (cực dương ở giữa)
- Cung cấp theo 2 chân Vin và GND.

Chú ý:

- Không cấp nguồn vào chân 3.3VDC
- Các chân GND đồng bộ mức 0VDC

10

Cấu hình của Arduino Uno

- Chân I/O số (Digital I/O)
 - Có 14 chân I/O số từ 0 đến 13
 - Có 6 chân có khả năng điều khiển PWM: 3, 5, 6, 9, 10, 11
 - Có 2 khả năng giao tiếp nối tiếp (Serial): 0, 1
 - Có khả năng giao tiếp SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)
- Chân I/O tương tự (Analog I/O)
 - ✓ Có 6 chân từ A0 đến A5
 - ✓ Có 2 khả năng giao tiếp I2C: A4 (SDA) và A5 (SCL)
- Nút Reset hoặc chân Reset: dùng để khởi tạo lại Arduino khi được nối xuống GND (0VDC)

11

Thuật ngữ

- **Sketch**: Chương trình viết chạy trên bo mạch Arduino
- **Pin**: ngõ vào hoặc ngõ ra (ví dụ: ngõ vào nối với nút nhấn, ngõ ra nối với đèn LED)
- **Digital**: giá trị HIGH hoặc LOW ứng với mức 1 hoặc 0, BẬT hoặc TẮT
- **Analog**: các khoảng giá trị, từ 0 – 255 (ví dụ: độ sáng đèn LED, tốc độ motor)

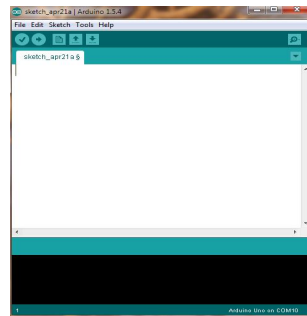
12

Cài đặt Arduino IDE

- **Bước 1:** Tải chương trình cài đặt tại website:
<http://arduino.cc/en/Main/Software>.

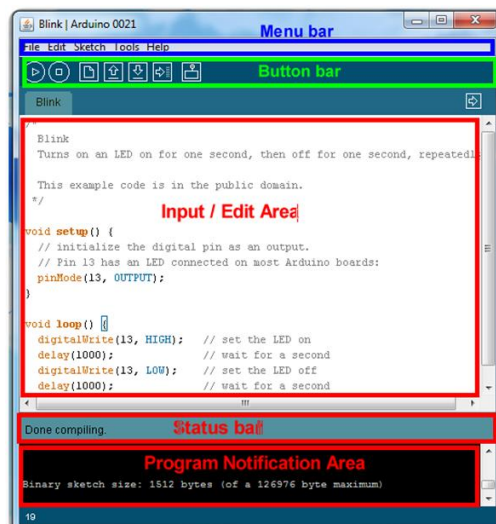
Đối với Windows có phiên bản cài đặt .exe và phiên bản nén .zip.

- **Bước 2:** Bấm (click) vào tập tin vừa tải (.exe) để cài đặt hoặc giải nén tập tin .zip vào thư mục bất kỳ và chạy chương trình không cần cài đặt.



13

Trình soạn thảo Arduino IDE

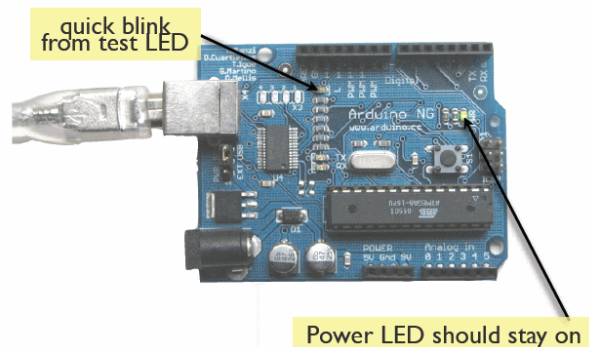


Tham khảo: <http://arduino.cc/en/Guide/Environment>

14

Cài đặt driver cho máy tính

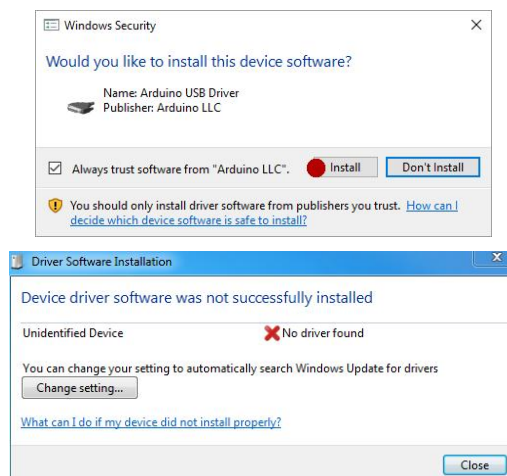
- **Bước 1:** Kết nối Arduino với máy tính bằng cáp USB → đèn led báo nguồn sáng



15

Cài đặt driver cho máy tính

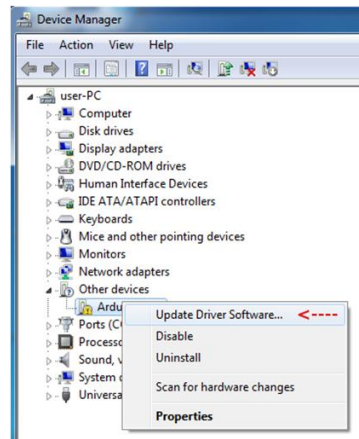
- **Bước 1:** Máy tính nhận dạng và thông báo nhận cài đặt driver.



16

Cài đặt driver cho máy tính

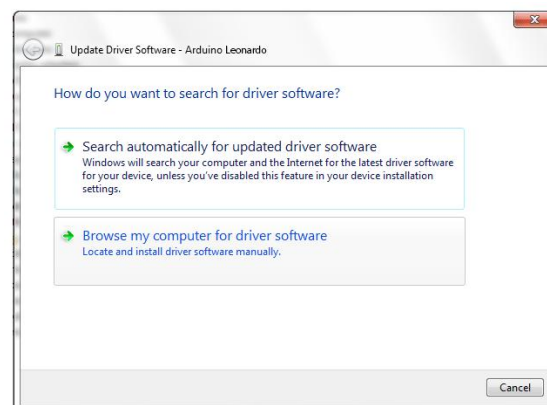
- **Bước 2:** Vào Start Menu → Control Panel → System and Security → System → Device Manager.



17

Cài đặt driver cho máy tính

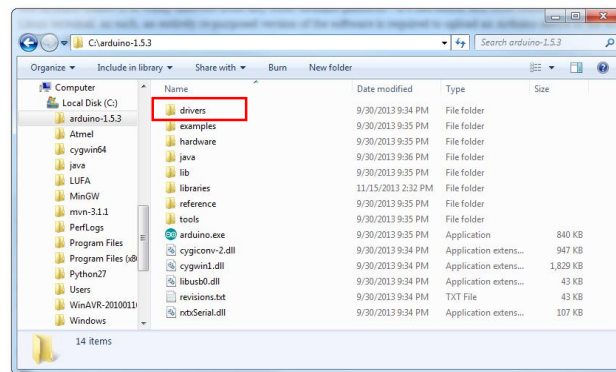
- **Bước 3:** Click chuột phải lên Arduino Uno → Update Driver Software



18

Cài đặt driver cho máy tính

- **Bước 4:** Chọn “Browse my computer for driver software” và chọn đường dẫn tới thư mục driver của bo Arduino Uno được lưu trữ. Click Next để hoàn thành quá trình cài đặt driver



19

Trình soạn thảo Arduino IDE

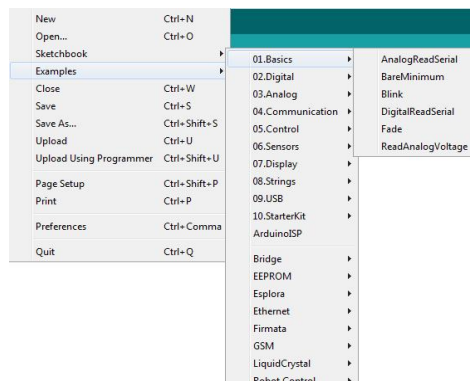


- ✓ New, Open, Save : Tạo mới, mở và Save sketch
- ✓ Upload: nạp code đang soạn thảo vào Arduino
- ✓ Verify : kiểm tra code có lỗi hay không
- ✓ Serial Monitor : Đây là màn hình hiển thị dữ liệu từ Arduino gửi lên máy tính

20

Trình soạn thảo Arduino IDE

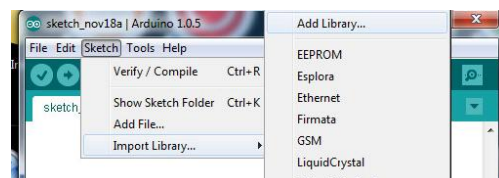
- Menu: **File Edit Sketch Tools Help**
- File: Tạo file mới New, Mở file Open, Ví dụ mẫu Example, ...



21

Trình soạn thảo Arduino IDE

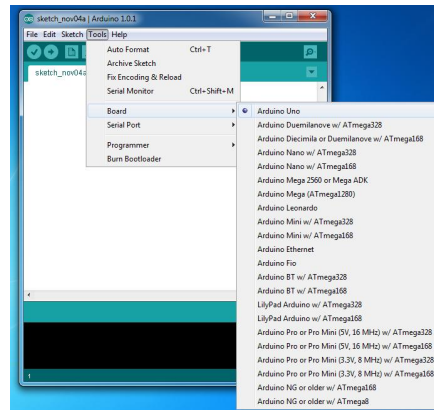
- Sketch menu
 - Verify/ Compile: chức năng kiểm tra lỗi code.
 - Show Sketch Folder: hiển thị nơi code lưu trữ code.
 - Add File: thêm vào một Tập code mới.
 - Import Library: thêm thư viện cho IDE



22

Trình soạn thảo Adruino IDE

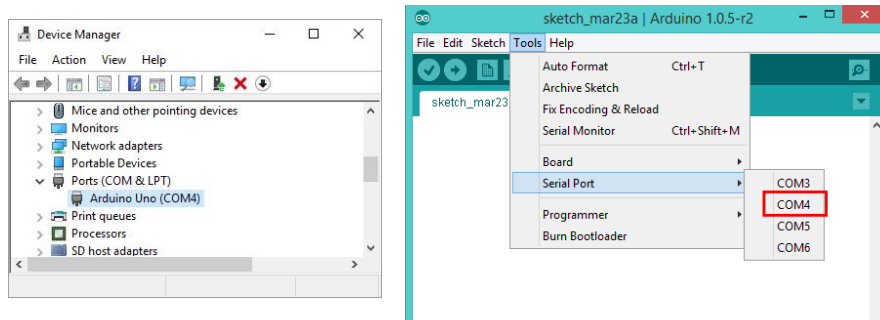
- Board : Chọn bo Arduino phù hợp



23

Trình soạn thảo Adruino IDE

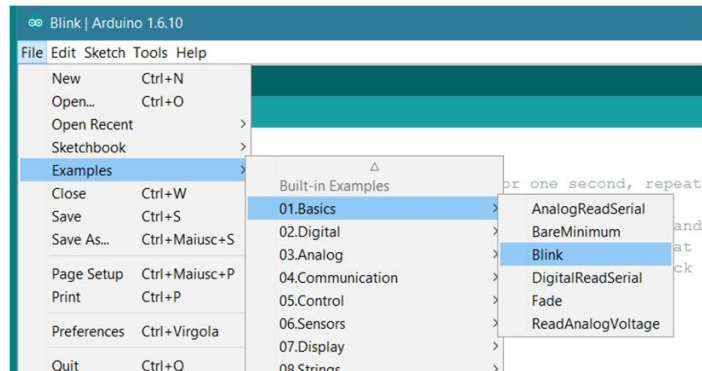
- Serial Port: Chọn cổng giao tiếp phù hợp. Thông tin cổng tham khảo trong Device Manager.



24

Ví dụ mẫu























- File → Examples → Basics → Digital → Blink



25

Lập trình cho bo mạch

- Ngôn ngữ lập trình C/C++
- Lập trình phần mềm, lập trình nhúng, lập trình vi điều khiển, lập trình di động, lập trình vi mạch, viễn thông, v.v.

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	100.0
3. Java	  	99.4
4. C++	  	96.9
5. C#	  	88.6
6. R		88.1
7. JavaScript	 	85.3
8. Go		75.7
9. Swift	 	74.3
10. Ruby	 	72.0

26

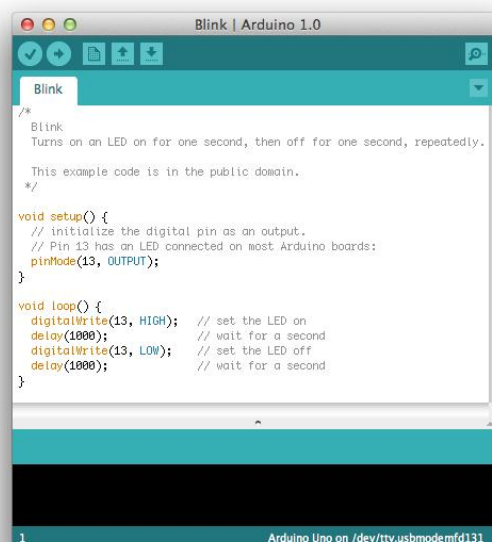
Lập trình cho bo mạch

- Ngôn ngữ lập trình C/C++
- Các câu lệnh phải kết thúc bởi dấu chấm phẩy ;
- Các “chú thích/comments” ở sau ký tự //comments hoặc ở trong /*comments */
- Sử dụng các hàm (functions)
- Các hàm không có biến trả về void()
- Các hàm thường có cấu trúc
 - Void setup()

```
{
// ...
}
```

27

Chương trình ví dụ

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The code editor shows the following text:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);             // wait for a second
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on /dev/tty.usbmodemfd131".

28

Các lệnh Digital I/O

- `pinMode(pin, mode)`
 - *mode* = INPUT hoặc OUTPUT
- `digitalRead(pin)`
 - Đọc giá trị từ *pin* = HIGH hoặc LOW (mức cao hoặc mức thấp)
- `digitalWrite(pin, value)`
 - Ghi giá trị *value* = HIGH hoặc LOW ra *pin*

29

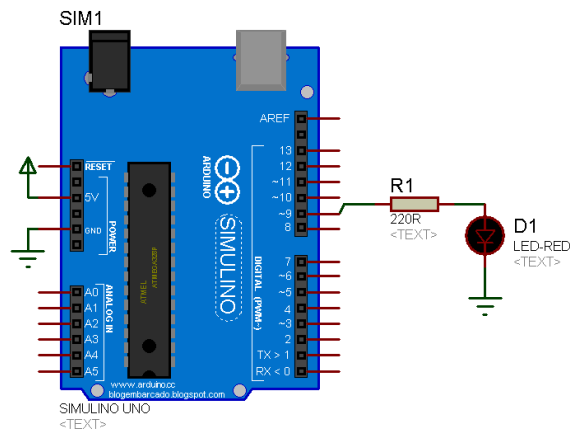
Các lệnh định thời (timing)

- `delay(xxx)`
 - Dừng trong *xxx* ms
 - VD: `delay(1000);` //dừng trong 1000ms
- `delayMicroseconds(yyy)`
 - Dừng trong *yyy* microseconds (us)
 - VD: `delayMicroseconds(500);` //dừng trong 500us
- Lệnh khác:
arduino.cc/en/Reference/HomePage

30

Ví dụ: điều khiển đèn LED

Sơ đồ mạch:




31

Ví dụ: điều khiển đèn LED

- Bước 1: Mở IDE và gõ đoạn code vào chương trình soạn thảo

```
• Code chương trình:  
int ledPin = 9; //Khai báo chân giao  
tiếp LED  
void setup() { //Cấu hình cho Arduino  
  pinMode(ledPin, OUTPUT);  
}  
void loop() { //Vòng lặp vô hạn khi  
chạy chương trình  
  digitalWrite(ledPin, HIGH);  
  delay(1000);  
  digitalWrite(ledPin, LOW);  
  delay(1000);  
}
```

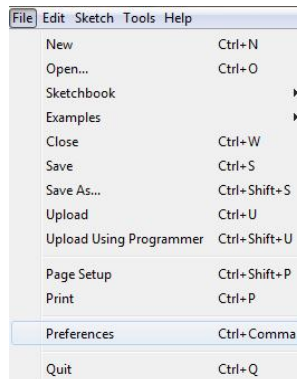
- Bước 2: click  để biên dịch chương trình và kiểm tra lỗi
- Bước 3: Tạo tập tin Hex để mô phỏng trên Proteus hoặc để lưu trữ khi chưa nạp code vào Arduino

32

Ví dụ: điều khiển đèn LED

- Bước 3:

Vào File chọn Preferences.

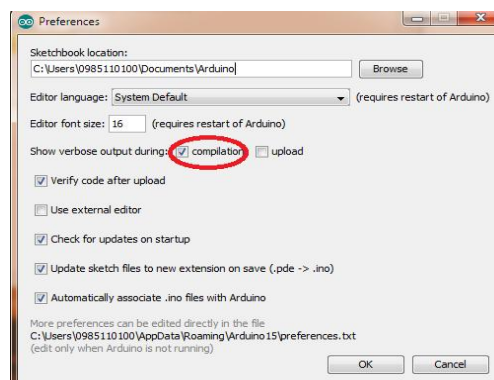


33

Ví dụ: điều khiển đèn LED


Bước 3:

Check vào compilation và OK.



34

Ví dụ: điều khiển đèn LED

Bước 4: Click 

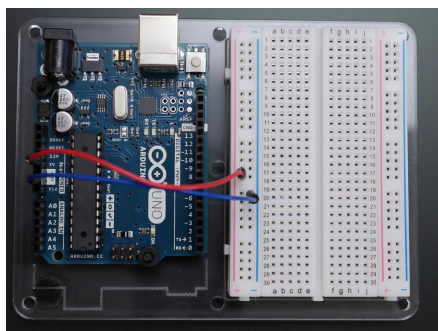
Check vào compilation và OK.

IDE sẽ tự động biên dịch một tập tin HEX được lưu ở đường dẫn
C:\Users\user_name\AppData\Local\Temp\filename.tmp\Blink.cpp.
hex

```
C:\Users\088611~1\AppData\Local\Temp\build918449727674380763.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
```

35

Dùng bo mạch Arduino với Bread board



36

Ứng dụng của Arduino

- <https://www.youtube.com/watch?v=eJg3yuAAawA>
- <https://www.youtube.com/watch?v=PfvqunpN7M4>
- <https://www.youtube.com/watch?v=9Xj3VuyoxOs>

37

Tham khảo

Tutorial, bo mạch, lập trình, ứng dụng, projects, v.v.

<https://www.arduino.cc>
<http://arduino.vn>

<https://www.tutorialspoint.com/arduino/index.htm>

38

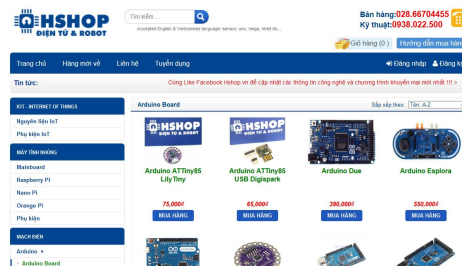
Tham khảo ứng dụng Arduino

- <https://create.arduino.cc/projecthub>
- <https://electronicsforu.com/arduino-projects-ideas>

39

Mua bo mạch Arduino ở đâu ?

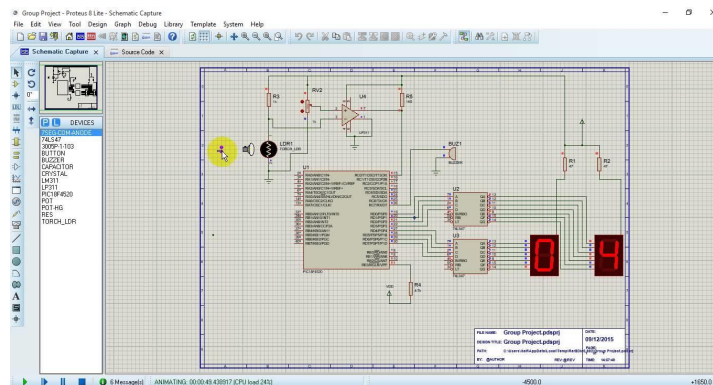
- <https://hshop.vn>
- <https://icdayroi.com/>



40

Proteus

- Là phần mềm thiết kế mạch (schematics, layout, mô phỏng, giả lập hoạt động của mạch điện)
- Trang web: <https://www.labcenter.com/downloads/>



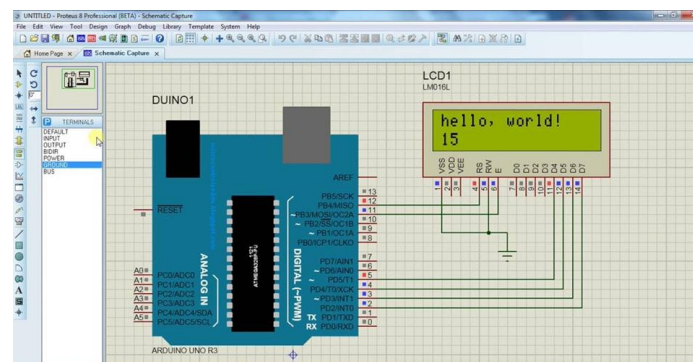
41

Hướng dẫn sử dụng Proteus

- Sau khi cài đặt Proteus, thực hiện các bước sau:

Bước 1: Tải Isis Proteus tại:

<http://blogembarcado.blogspot.com/p/downloads.html>



42

Hướng dẫn sử dụng Proteus

- Sau khi cài đặt Proteus, thực hiện các bước sau:

Bước 2: Sao chép BLOGEMBARCADO.LIB thư viện trên vào thư mục:

C:\Program Files (hoặc x86) \Labcenter Electronics
 \Proteus X Professional\LIBRARY Proteus X
C:\Program Files (hoặc x86)\Labcenter Electronics
 \ProteusX professional\Data\LIBRARY

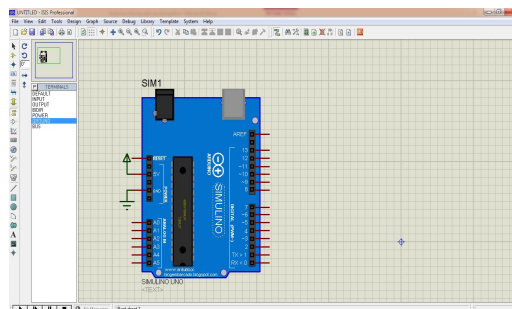
Bước 3: Khởi động Proteus lên vào thư viện linh kiện bằng cách bấm phím P và gõ từ khoá là ARDUINO, xuất hiện danh sách các bo hiện và chọn Arduino Uno.

43

Hướng dẫn sử dụng Proteus

- Sau khi cài đặt Proteus, thực hiện các bước sau:

Bước 3: Khởi động Proteus lên vào thư viện linh kiện bằng cách bấm phím P và gõ từ khoá là ARDUINO, xuất hiện danh sách các bo hiện và chọn Arduino Uno.

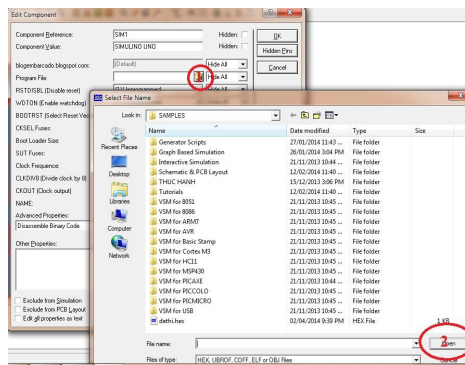


44

Hướng dẫn sử dụng Proteus

- Sau khi cài đặt Proteus, thực hiện các bước sau:
- **Bước 4:** Cấp nguồn vào 2 chân 5V và GND
- **Bước 5:** Mở hộp thoại Edit Component và chọn file HEX theo đường dẫn đã lưu ở bài trước.

Chọn PLAY để mô phỏng chương trình



45

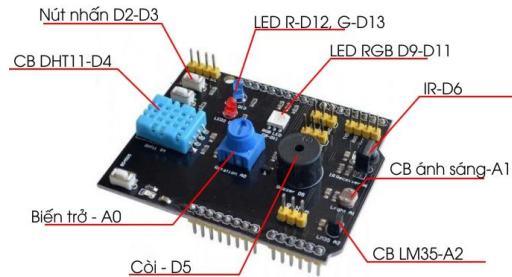
Hướng dẫn demo trên Youtube

- <https://www.youtube.com/watch?v=jy--XGV6ofk>
- <https://www.youtube.com/watch?v=WYkTHvtrnLk>
- Link down Arduino Lib bổ sung
- <https://github.com/cyberponk/Arduino-Library-for-Proteus/tree/master/Copy%20to%20Proteus%20folder/LIBRARY>

46

Mạch 9-trong-1

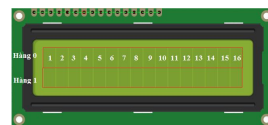
- **Cảm biến độ ẩm nhiệt độ DHT11:** Giao tiếp 1-Wire
- **Cảm biến nhiệt độ LM35:** giao tiếp Analog
- **Còi Buzzer:** tập âm thanh, có thể kích on off bằng tín hiệu Digital hoặc cấp xung PWM để tạo ra tần số âm thanh khác nhau
- **Cảm biến hồng ngoại IR1838:** sử dụng để nhận tín hiệu hồng ngoại 38khz từ các loại Remote, tay cầm điều khiển
- **Cảm biến ánh sáng quang trở Photoresistor**
- **Biến trở Variable Resistor:** có thể dùng hàm AnalogRead() để đọc giá trị điện áp trên biến trở tương ứng.
- **Button:** 2 button nối vào chân D2 và D3 của Arduino, 1 button nối vào chân Reset.
- **Led:** 2 led đơn nối vào chân D12 và D13 của Arduino.
- **Led RGB:** nối vào 2 chân D9, D10, D11 của Arduino, cấp tín hiệu hoặc xung để biến đổi màu sắc.



STT	Tên linh kiện	Số chân
1	Cảm biến DHT11	D4
2	Cảm biến LM35	A2
3	Còi buzzer	D5
4	Cảm biến hồng ngoại IR1838	D6
5	Cảm biến ánh sáng quang trở	A1
6	Biến trở Variable Resistor	A0
7	Nút nhấn SW1, SW2	D2, D3
8	LED Đỏ, Xanh	D12, D13
9	LED RGB	D9, D10, D11

Mạch LCD

- LCD1602 Keypad Shield hoạt động ở điện áp 5V.
- Bao gồm LCD 16x2 và 6 nút nhấn.
- Sử dụng chân 4,5,6,7,8,9,10 và chân A0 của Arduino để giao tiếp với LCD.
- Không cần sử dụng chân 10 sau khi LCD đã được kết nối.
- Đèn nền xanh dương với chữ màu trắng.
- Sử dụng thư viện LCD 4bit.
- Điều chỉnh độ sáng đèn nền qua biến trở

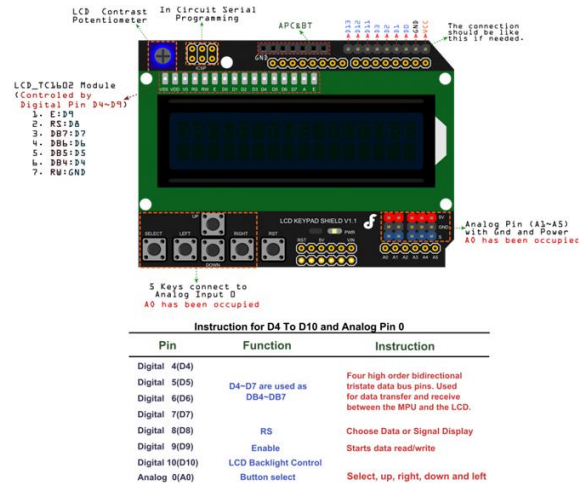


LCD 16x2 gồm có 2 hàng, mỗi hàng 16 cột



Bộ mạch LCD 1602 Keypad LCD Shield

Mạch LCD



49

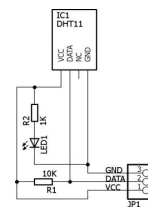
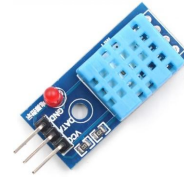
Mạch LCD

- Thư viện LiquidCrystal: `#include <LiquidCrystal.h>`
- `LiquidCrystal(rs, enable, d4, d5, d6, d7)`: Mạch LCD có thể được điều khiển bởi 4 hoặc 8 đường data. Bỏ qua chân d0, d1, d2, d3 và không kết nối. Chân RW có thể nối xuống GND thay vì nối vào Arduino.
 - Ví dụ: `LiquidCrystal lcd(8, 9, 4, 5, 6, 7);`
- `lcd.begin(cols, rows)`: Khởi tạo giao tiếp LCD, xác định số hàng và cột hiển thị. Hàm `begin()` cần được gọi trước các lệnh khác trong thư viện LCD này.
 - Ví dụ: `lcd.begin(16, 2);` //thiết lập 16 cột, 2 hàng cho LCD 6x2
- `lcd.setCursor(col, row)`: Đặt vị trí con trỏ mà ký tự/chuỗi ký tự sẽ xuất hiện trên LCD.
 - Ví dụ: `lcd.setCursor(0,0);` //con trỏ sẽ đặt ở hàng 0, cột 0
- `lcd.print(data)`: Ghi 1 chuỗi ký tự ra LCD.
 - Ví dụ: `lcd.print("hello, world!");`
- `lcd.write(data)`: Gửi 1 ký tự ra LCD

50

Mạch cảm biến

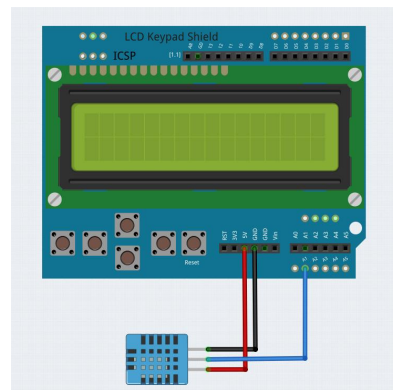
- Cảm biến DHT11 là cảm biến nhiệt độ và độ ẩm
- Thông số của cảm biến như sau:
 - Điện áp hoạt động : 3V - 5V (DC)
 - Dải độ ẩm hoạt động : 20% - 90% RH, sai số $\pm 5\%RH$
 - Dải nhiệt độ hoạt động : $0^{\circ}C \sim 50^{\circ}C$, sai số $\pm 2^{\circ}C$
 - Thời gian đáp ứng: 1s
 - Khoảng cách truyền tối đa: 20m
- Chân linh kiện: VCC, DATA, GND



51

Mạch cảm biến

- Mạch kết nối cảm biến DHT11 với bo mạch LCD1602 và bo mạch Arduino
- Mạch cảm biến gồm có 3 chân
 - Dữ liệu cảm biến (DATA): S(xanh) – nối vào chân A1
 - Chân nguồn: VCC (đỏ) -- VCC
 - Chân nguồn; GND (đen) -- GND
- Ghi chú: Ngõ vào analog A0 được nối với các button nên không dùng chân A0.



52

Hỏi & Trả lời



53