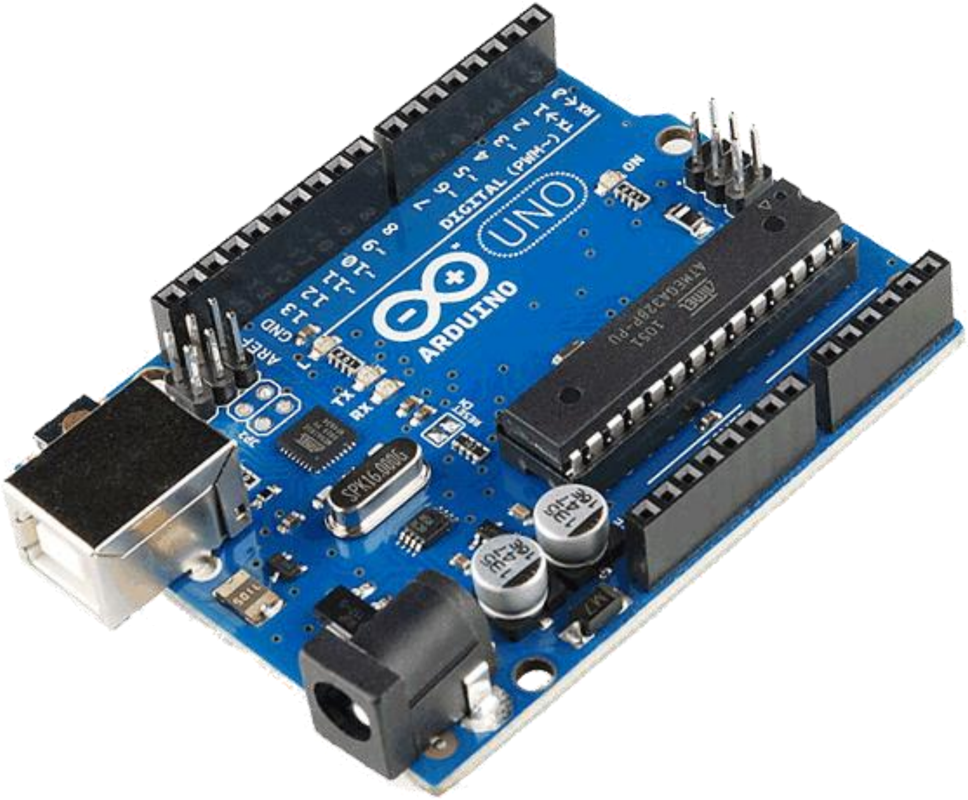
Basic Arduino Training



Internal research group training

Dr. Vinay Chamola, Assistant Professor, EEE Department, BITS-Pilani

Agenda

Module 1: Introduction to Arduino

- What is Arduino
- History of Arduino
- Arduino Boards
- Arduino Uno
- Setting Up Arduino IDE
- Your First Arduino Project

Module 2: Basic C Programming

- Conditional
- Loop
- Operators
- Function
- Comment
- Preprocessor

Agenda

Module 3: Arduino Programming

- –Components & Ohm's Law
- –pinMode
- –digitalWrite
- –analogWrite
- –Serial Communication
- –digitalRead
- –analogRead & Map
- –Challenge 1: Digital Read/Write
- –Challenge 2: Analog Read/Write
- –tone

Module 4: Arduino Shields

- Motor Shield
- Ethernet Shield
- WiFi Shield

Agenda

Module 5: Arduino Standard Libraries

- EEPROM
- Ethernet
- Firmata
- Liquid Crystal Display
- Servo
- SPI
- SoftwareSerial
- Stepper
- WIFI
- Wire
- Install Additional Libraries

Agenda

Module 6: Introduction to Processing

- Install Processing
- Basics of Processing
- Interface Processing with Arduino

Module 7: Arduino Sonar Project

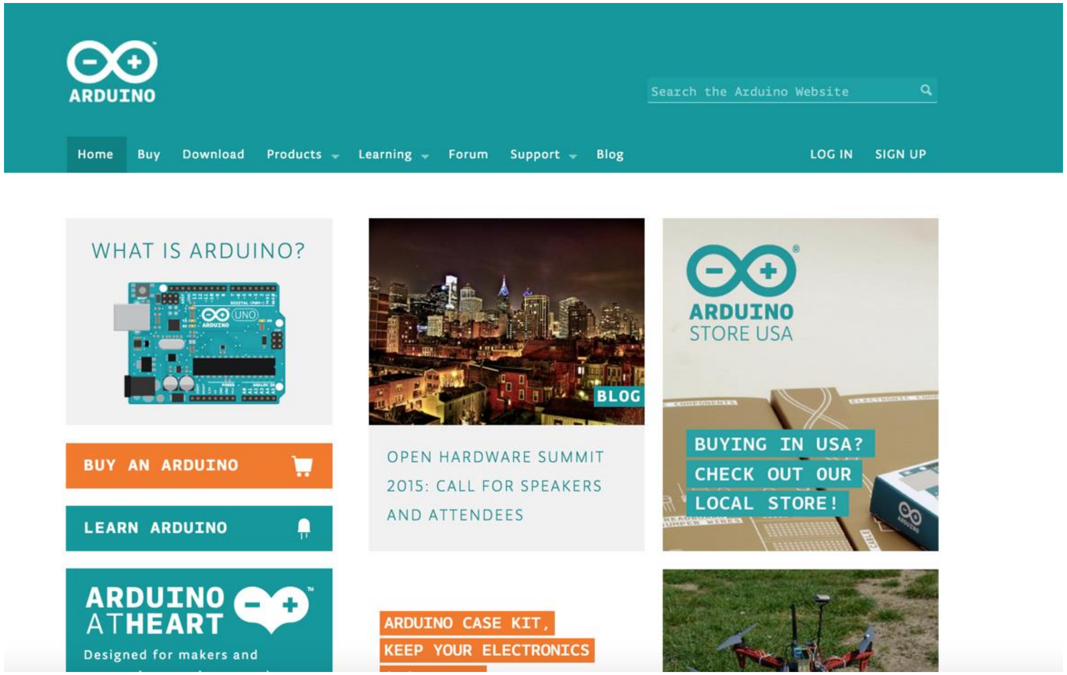
- Ultrasonic Ranging Sensor
- Servo Motor
- Building a Sonar System

Module 8: Arduino IoT

Module 1
Introduction to Arduino

What is Arduino?

www.arduino.cc



OPEN HARDWARE SUMMIT 2015: CALL FOR SPEAKERS AND ATTENDEES

BUYING IN USA? CHECK OUT OUR LOCAL STORE!

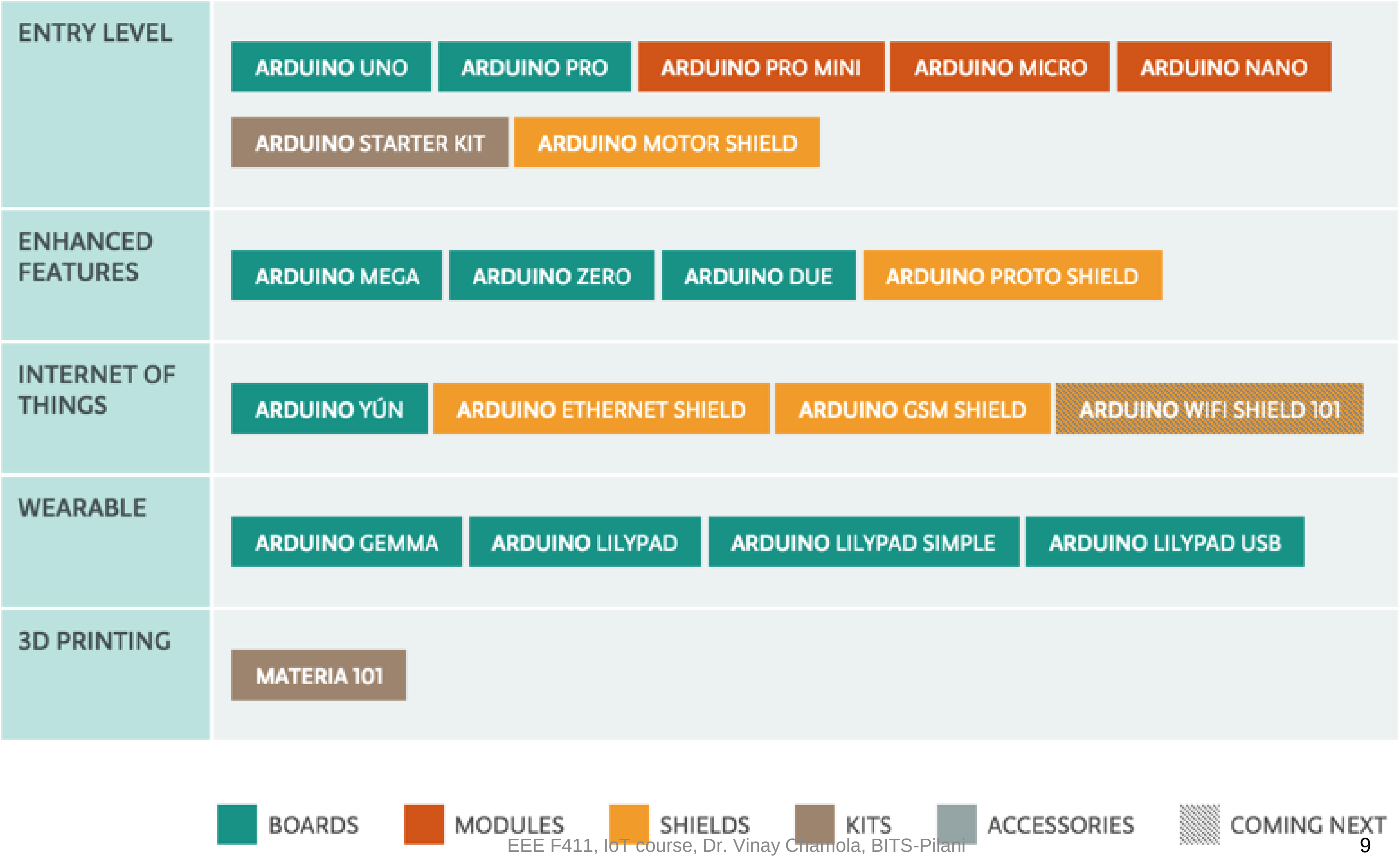
open source hardware

- Arduino is an open-source electronics platform
- The hardware and software are open source.
- Software code, based on C is transferable.
- Large community for Arduino.

History of Arduino

Arduino was started in 2005 by Massimo Banzi and other co-founders from Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. It was named after the bar where the co-founders visited often. The name of the bar is named after King Arduin.

Arduino Products

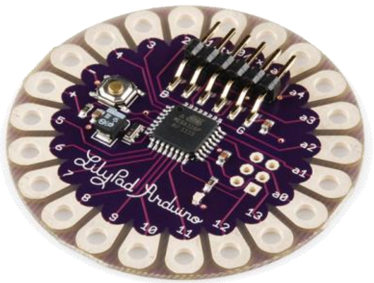


ENTRY LEVEL	🟩 ARDUINO UNO	🟩 ARDUINO PRO	🟥 ARDUINO PRO MINI	🟥 ARDUINO MICRO	🟥 ARDUINO NANO
	🟫 ARDUINO STARTER KIT	🟧 ARDUINO MOTOR SHIELD			
ENHANCED FEATURES	🟩 ARDUINO MEGA	🟩 ARDUINO ZERO	🟩 ARDUINO DUE	🟧 ARDUINO PROTO SHIELD	
INTERNET OF THINGS	🟩 ARDUINO YÚN	🟧 ARDUINO ETHERNET SHIELD	🟧 ARDUINO GSM SHIELD	🟧▨ ARDUINO WIFI SHIELD 101	
WEARABLE	🟩 ARDUINO GEMMA	🟩 ARDUINO LILYPAD	🟩 ARDUINO LILYPAD SIMPLE	🟩 ARDUINO LILYPAD USB	
3D PRINTING	🟫 MATERIA 101				

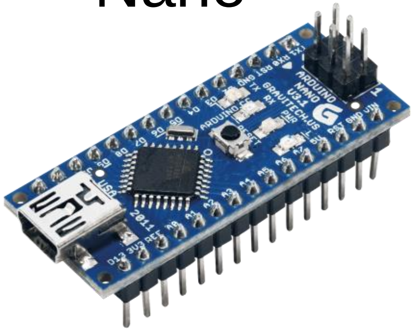
🟩 BOARDS 🟥 MODULES 🟧 SHIELDS 🟫 KITS ⬜ ACCESSORIES ▨ COMING NEXT

Arduino Boards

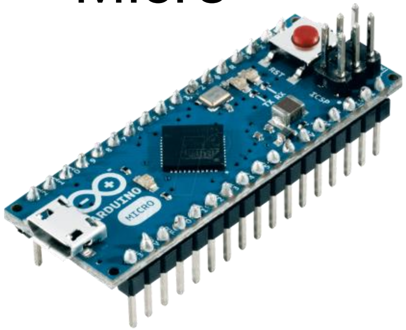
Lilypad



Nano

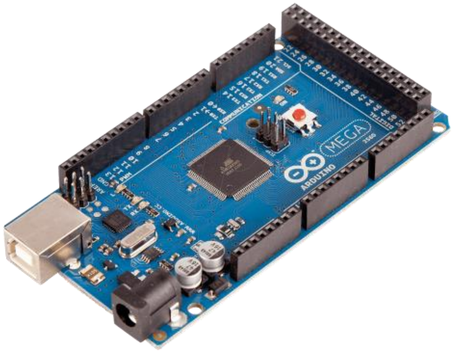


Micro



Mini Pro

Mega

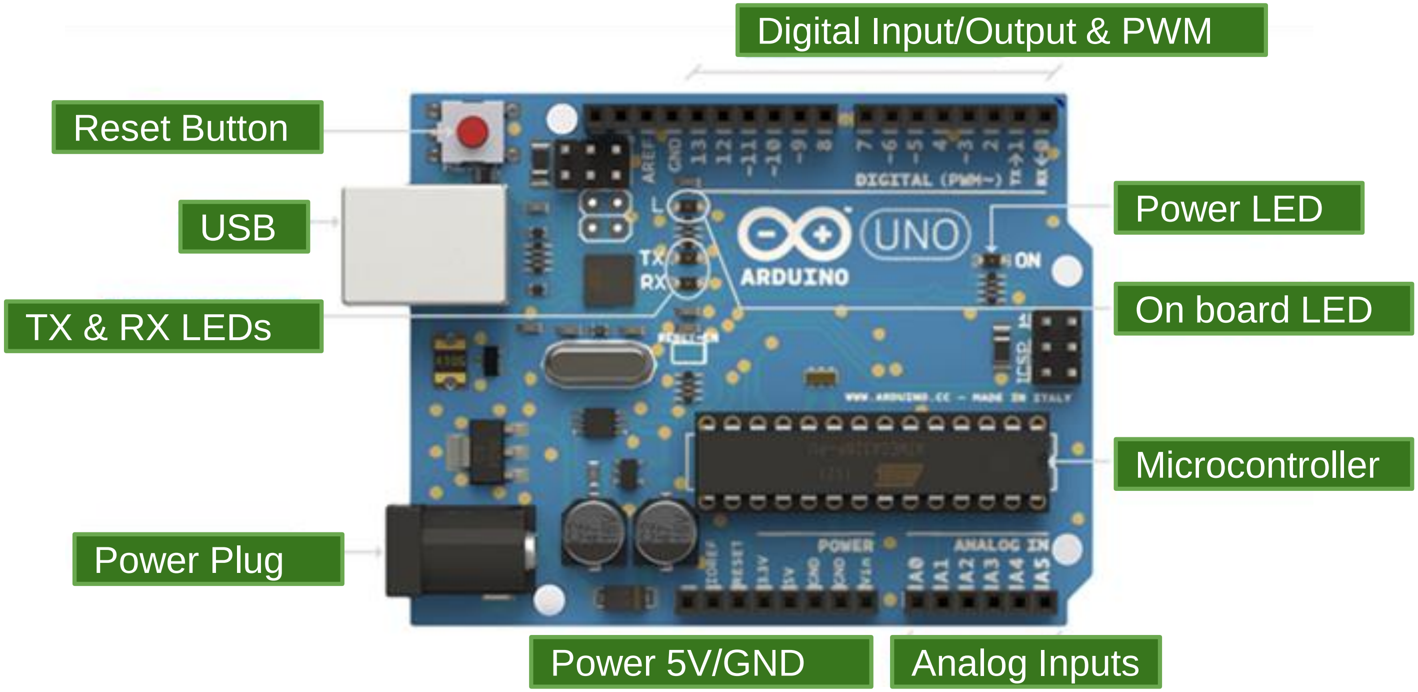


Leonardo

Due

Yun

Arduino Uno



Digital Input/Output & PWM

Reset Button

USB

TX & RX LEDs

Power Plug

Power LED

On board LED

Microcontroller

Power 5V/GND

Analog Inputs

Microcontroller: ATmega328

Digital I/O Pins: 14 (6 PWM output)

Flash Memory: 32 KB (ATmega328)

Operating Voltage: 5V

Analog Input Pins : 6

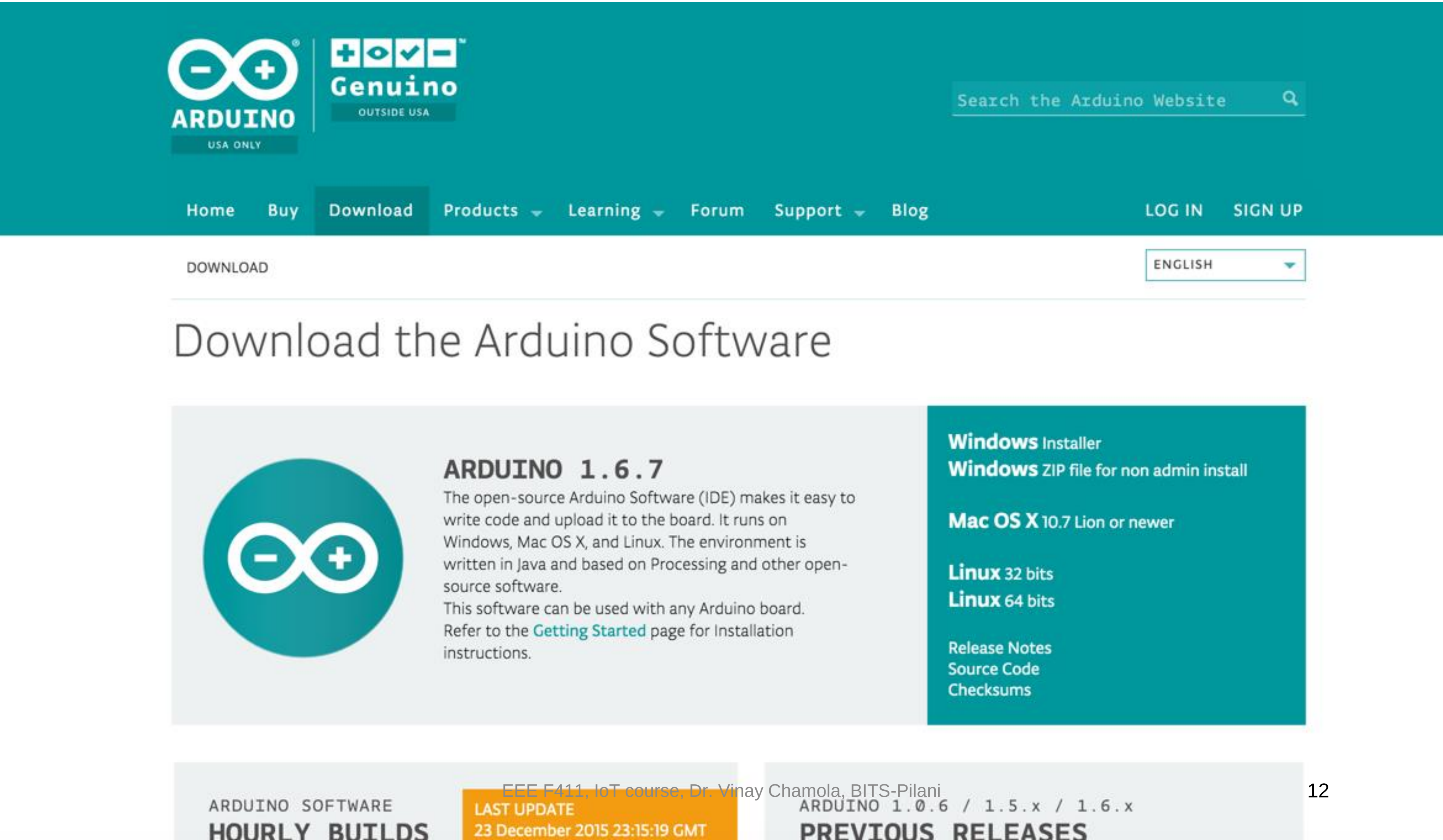
Clock Speed: 16 MHz

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

LED: 13. There is a built-in LED connected to digital pin 13.

Download Arduino IDE

http://www.arduino.cc/download



Download the Arduino Software

ARDUINO 1.6.7

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits

Release Notes
Source Code
Checksums

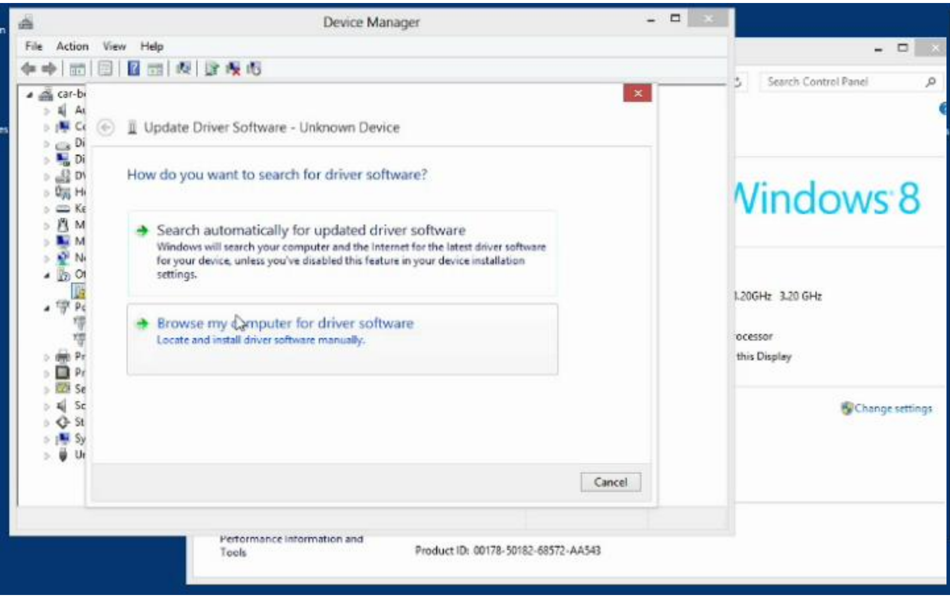
ARDUINO SOFTWARE
HOURLY BUILDS

LAST UPDATE
23 December 2015 23:15:19 GMT

ARDUINO 1.0.6 / 1.5.x / 1.6.x
PREVIOUS RELEASES

Installing Driver on Window

Goto Device Manager



Update Driver Software - Unknown Device

How do you want to search for driver software?

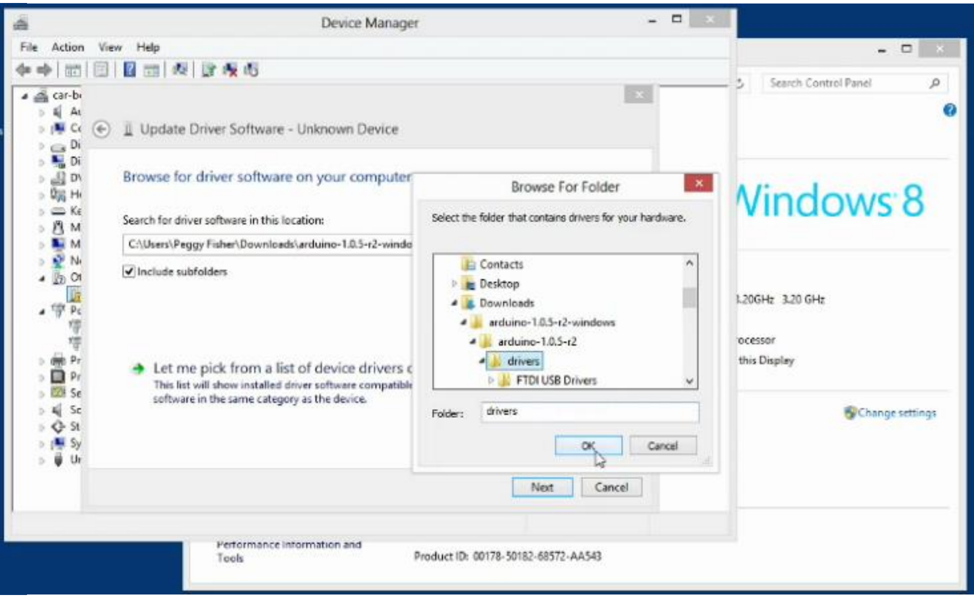
Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.

Performance Information and Tools

Product ID: 00178-50182-68572-AA543

Locate Driver Location



Update Driver Software - Unknown Device

Browse for driver software on your computer

Search for driver software in this location:
C:\Users\Peggy Fisher\Downloads\arduino-1.0.5-r2-windo

☑ Include subfolders

Let me pick from a list of device drivers c
This list will show installed driver software compatibl
software in the same category as the device.

Browse For Folder

Select the folder that contains drivers for your hardware.

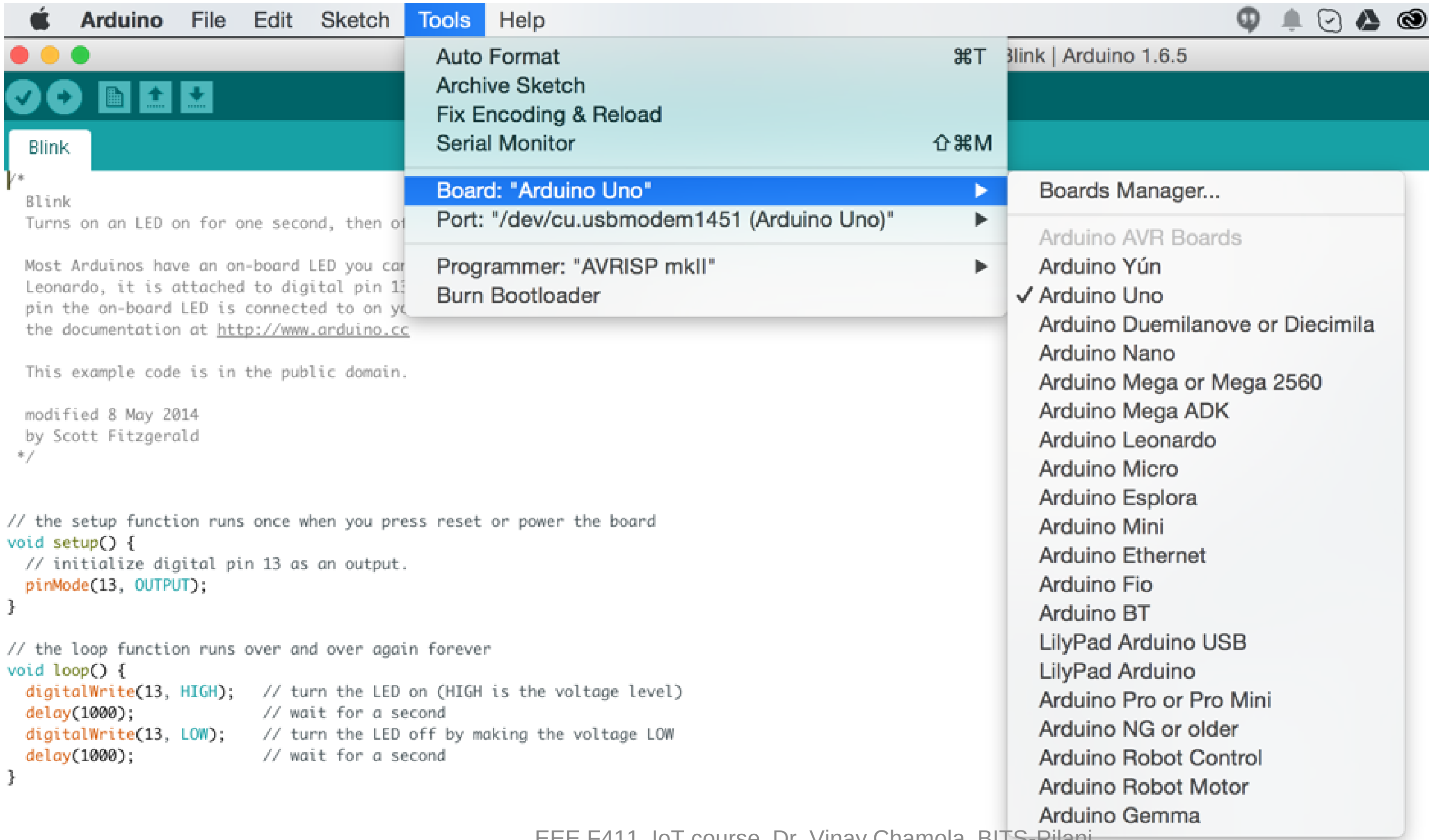
Folder: drivers

Performance Information and Tools

Product ID: 00178-50182-68572-AA543

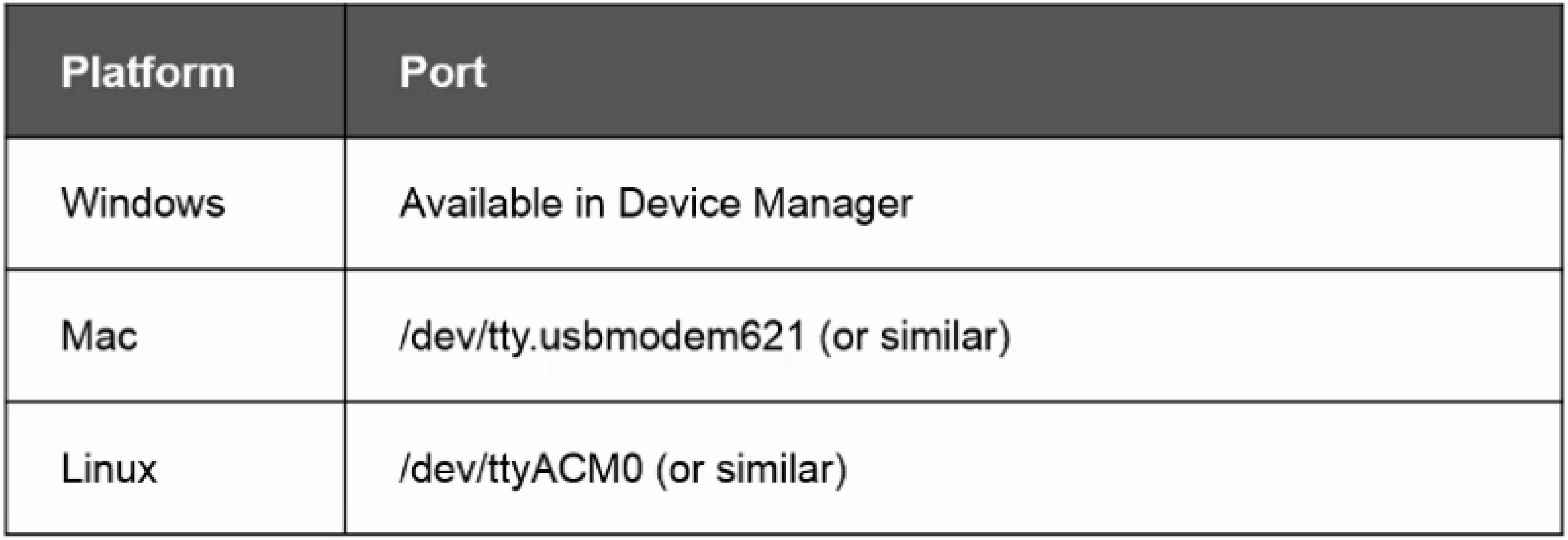
Setup Arduino IDE

Select the Board : Tools->Board->Arduino Uno
Select the Port: Tools->Port -> COM3 (Window)



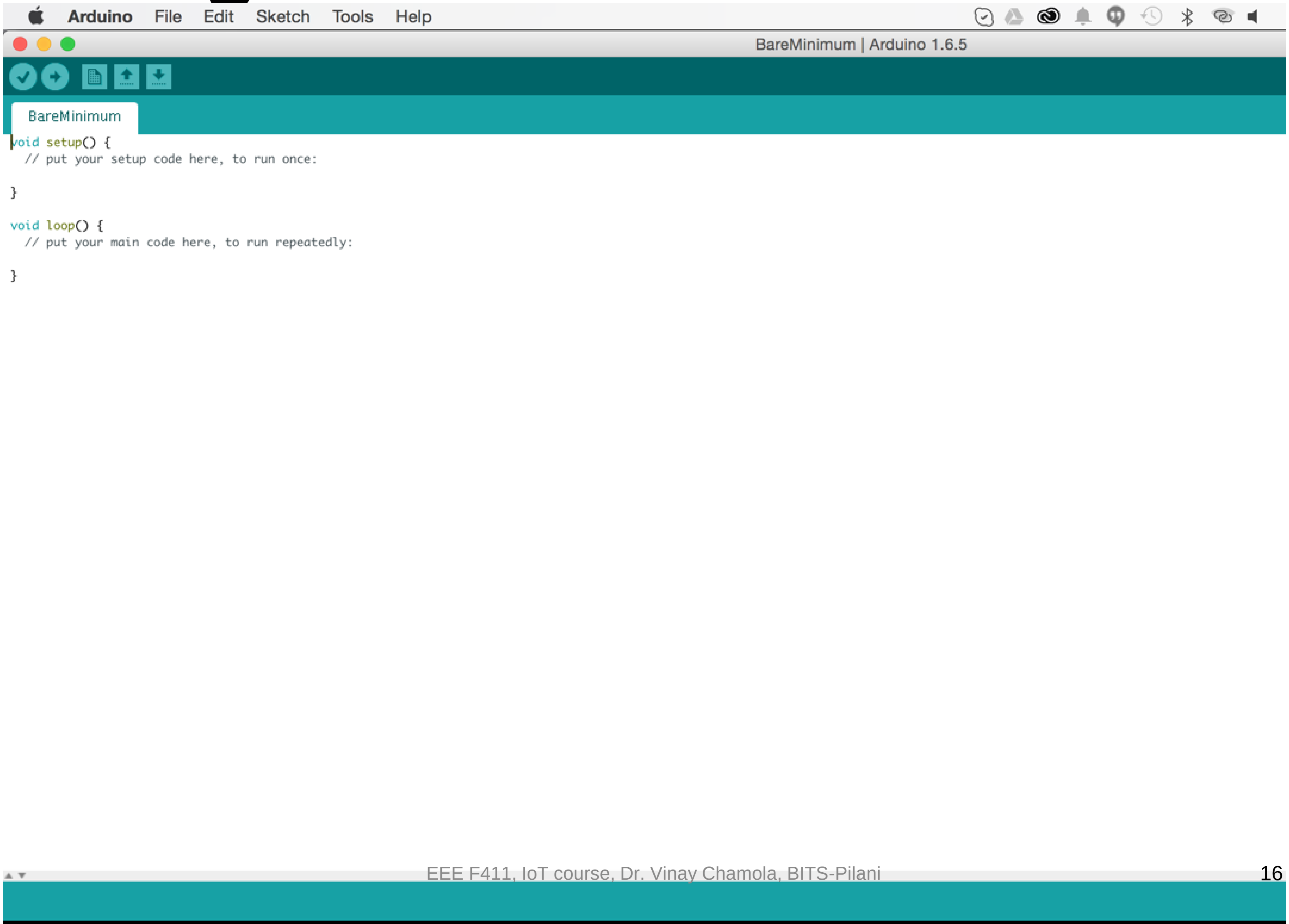
```
/*
  Blink
  Turns on an LED on for one second, then of

  Most Arduinos have an on-board LED you car
  Leonardo, it is attached to digital pin 13
  pin the on-board LED is connected to on yc
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

Arduino Communication Ports



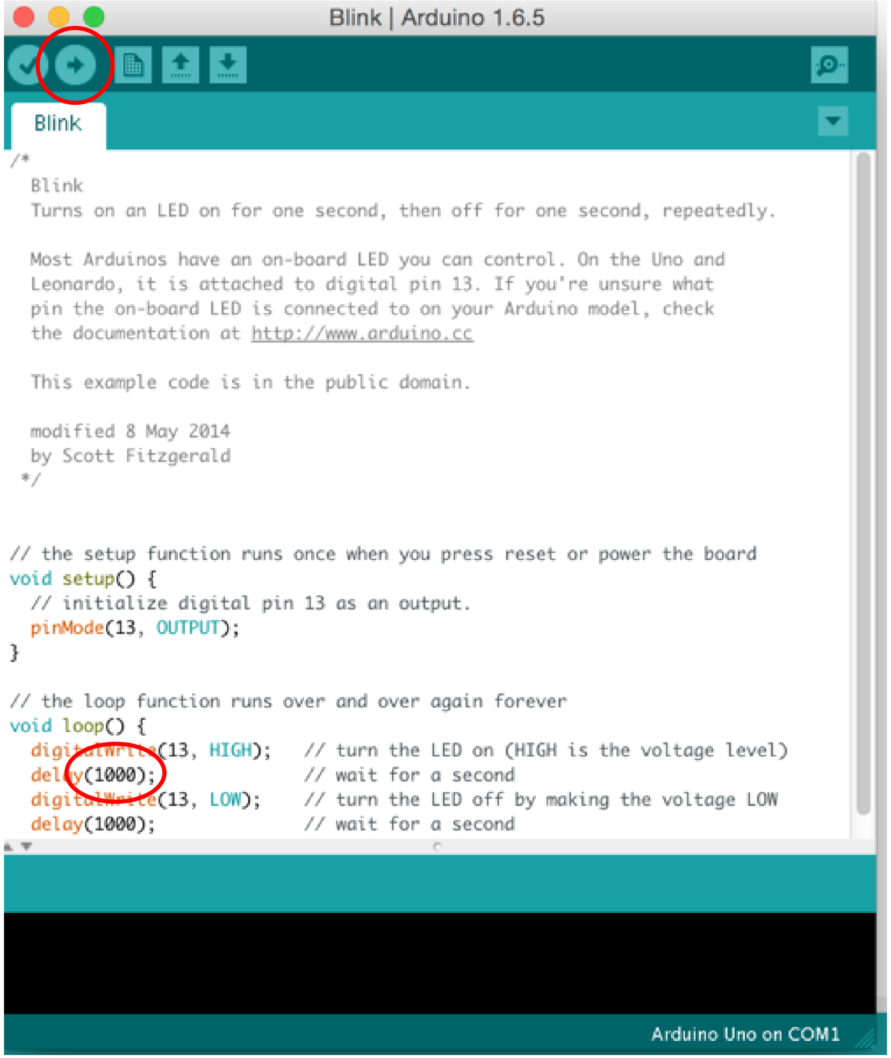
Platform	Port
Windows	Available in Device Manager
Mac	/dev/tty.usbmodem621 (or similar)
Linux	/dev/ttyACM0 (or similar)

Navigate Arduino IDE



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Blink Project



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
```

- Open File->Example-> 0.1Basic->Blink
- Click the upload button
- Observe the LED blink
- Change the delay value to 100
- Click the upload button
- Observe the LED blink
- Change the delay value to 2000
- Click the upload button
- Observe the LED blink

Module 2
Basic C Programming

Conditional

if Statement

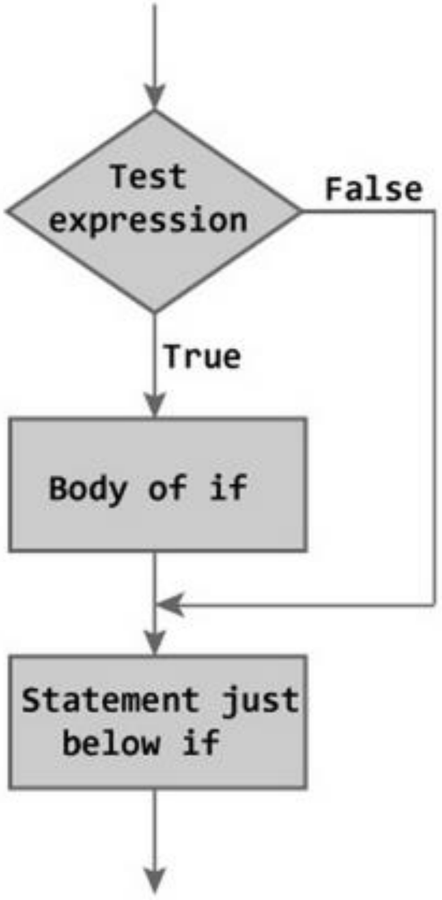


Figure: Flowchart of if Statement

if Syntax

```
if (condition)
{
  // do something here
}
```

if Statement Example

```
if (x > 120) {
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}
```

if-else Statement

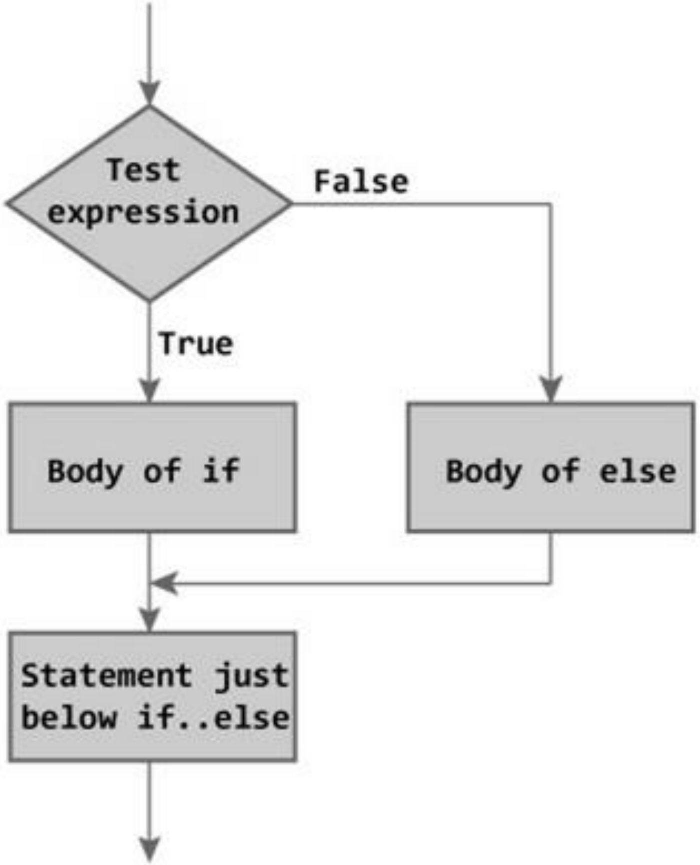


Figure: Flowchart of if...else Statement

if-else Syntax

```
if (condition)
{
  // action A
}
else
{
  // action B
}
```

If-else Statements

```
if (x > 120) {
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}  else {
  digitalWrite(LEDpin1, LOW);
  digitalWrite(LEDpin2, LOW);
}
```

if-else-if Syntax

```
if (condition) {
      do Something;}
else if (condition2) {
      do Something Else;}
else {
              do Another Thing;}
```

if-else-if Example

```
if (x > 120) {
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}  else if  (x > 200) {
  digitalWrite(LEDpin1, LOW);
  digitalWrite(LEDpin2, HIGH);
}  else {
     digitalWrite(LEDpin1, LOW);
  digitalWrite(LEDpin2, LOW);
}
```

Switch Syntax

```
switch (var) {
  case label:
    // statements
    break;
  case label:
    // statements
    break;
  default:
    // statements
  break;
}
```

Switch Example

```
char grade = 'B';
  switch(grade)
  {
  case 'A' :
    printf("Excellent!\n" );
    break;
  case 'B' :
  case 'C' :
    printf("Well done\n" );
    break;
   …
```

Goto Syntax

```
label:

goto label;
```

Goto Example

```
for(byte r = 0; r < 255; r++) {
  if (analogRead(0) > 250) { goto bailout;}
  }

bailout:
```

Loop

While Loop Statement

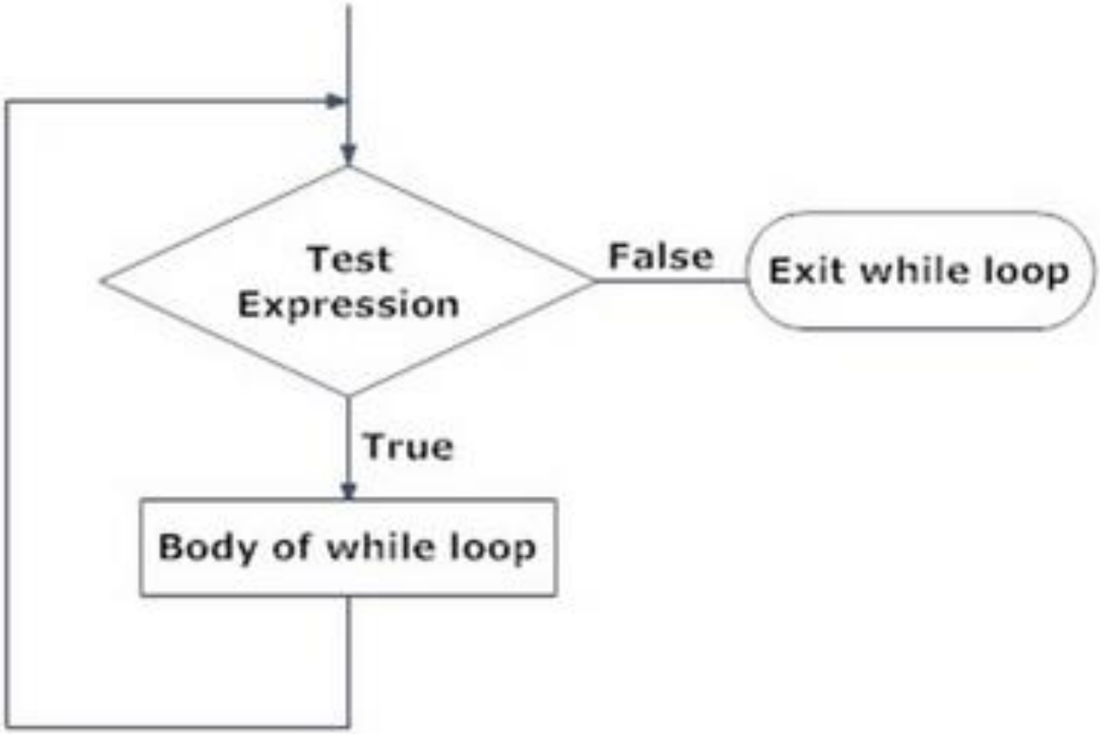


Figure: Flowchart of while loop

While Loop Syntax

```
while(expression) {
  // statement(s)
}
```

While Loops Examples

```
int i = 10;
while ( i > 0 ) {
      printf("Hello %d\n", i );
      i = i - 1;
}
```

Do While Loop Statement

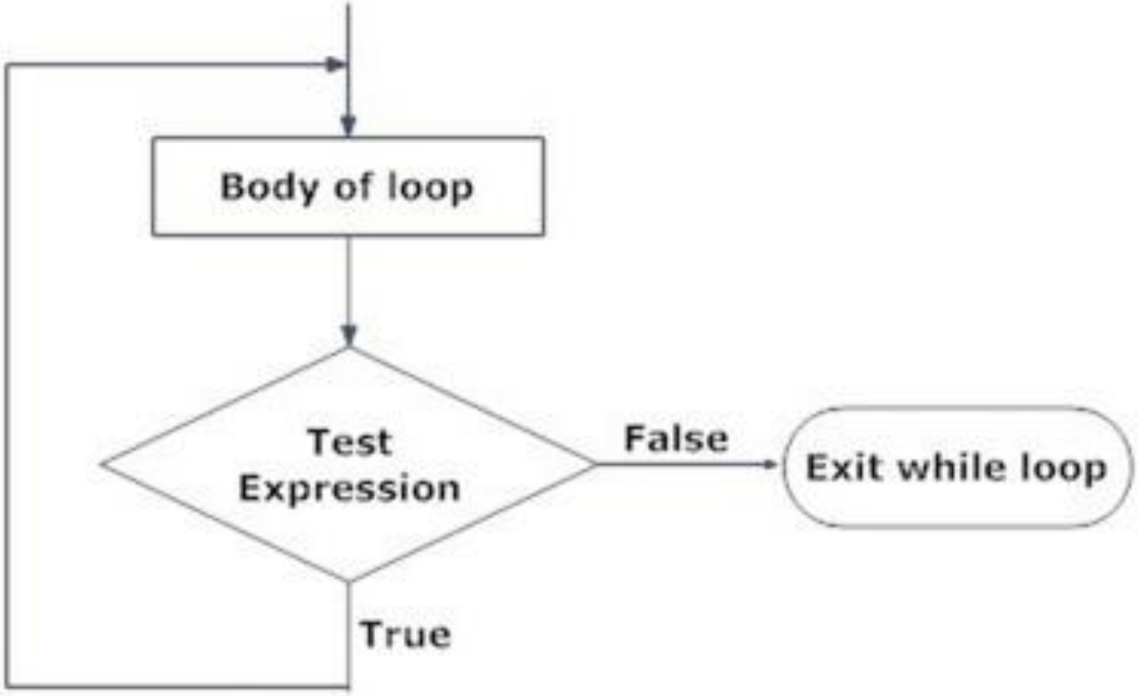


Figure: Flowchart of do...while loop

Do While Loops Syntax

```
do
{
   // statement block
} while (test condition);
```

Do While Loops Examples

```
do{
  delay(50);
  x = readSensors();
} while (x < 100);
```

For Loop Statement

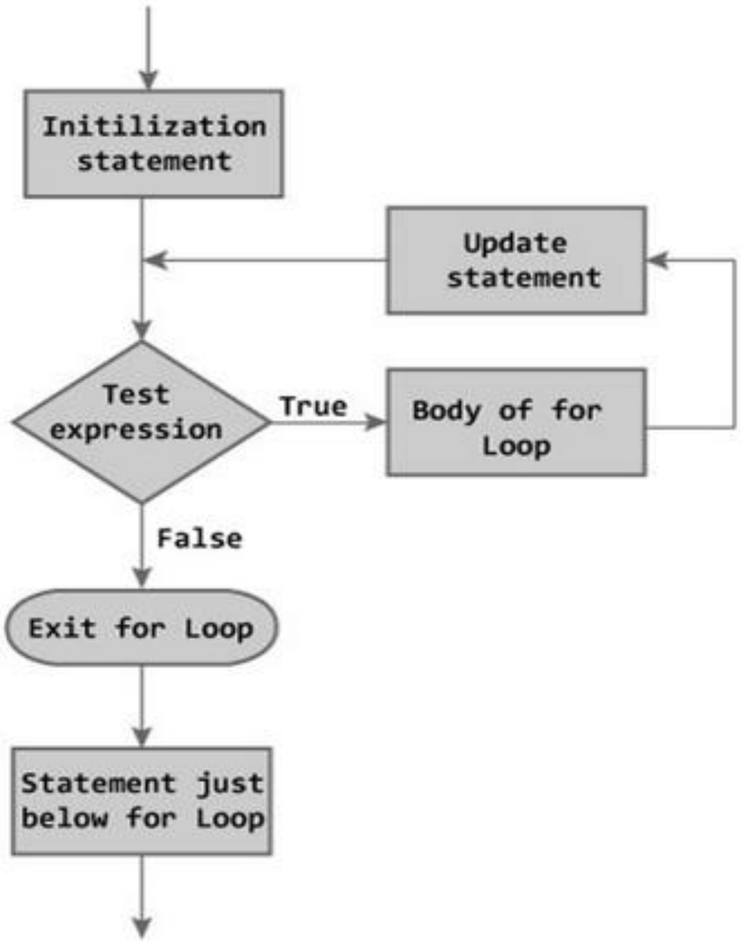


Figure: Flowchart of for Loop

For Loops Syntax

```
for (initialization; condition; increment) {
//statement(s);
}
```

For Loops Example

```
for (int i=0; i <= 255; i++){
    analogWrite(PWMpin, i);
    delay(10);
  }
```

Break

```
for (x = 0; x < 255; x ++) {
   analogWrite(PWMpin, x);
   sens = analogRead(sensorPin);
   if (sens > threshold) {
      x = 0;
      break;
   }
   delay(50);
}
```

Continue

```
for (x = 0; x < 255; x ++) {
    if (x > 40 && x < 120){
        continue;
    }

    analogWrite(PWMpin, x);
    delay(50);
}
```

Operators

Arithmetic Operators

=	(assignment operator)
+	(addition)
-	(subtraction)
*	(multiplication)
/	(division)
%	(modulo)

Comparison Operators

==	(equal to)
!=	(not equal to)
<	(less than)
>	(greater than)
<=	(less than or equal to)
>=	(greater than or equal to)

Boolean Operators

&&	(and)
||	(or)
!	(not)

Compound Operators

++ (increment)

-- (decrement)

+= (compound addition)

-= (compound subtraction)

*= (compound multiplication)

/= (compound division)

%= (compound modulo)

&= (compound bitwise and)

|= (compound bitwise or)

Function

Function Example

```
int checkSensor()

{
   if (analogRead(0) > 400) {
      return 1;
   else{
      return 0;
   }
}
```

Comment

Comment

Single line comment

```
// ….
```

Multiple lines comment

```
/*
…..
*/
```

Preprocessor

#define

```
#define Name value
#define ledPin 3
```

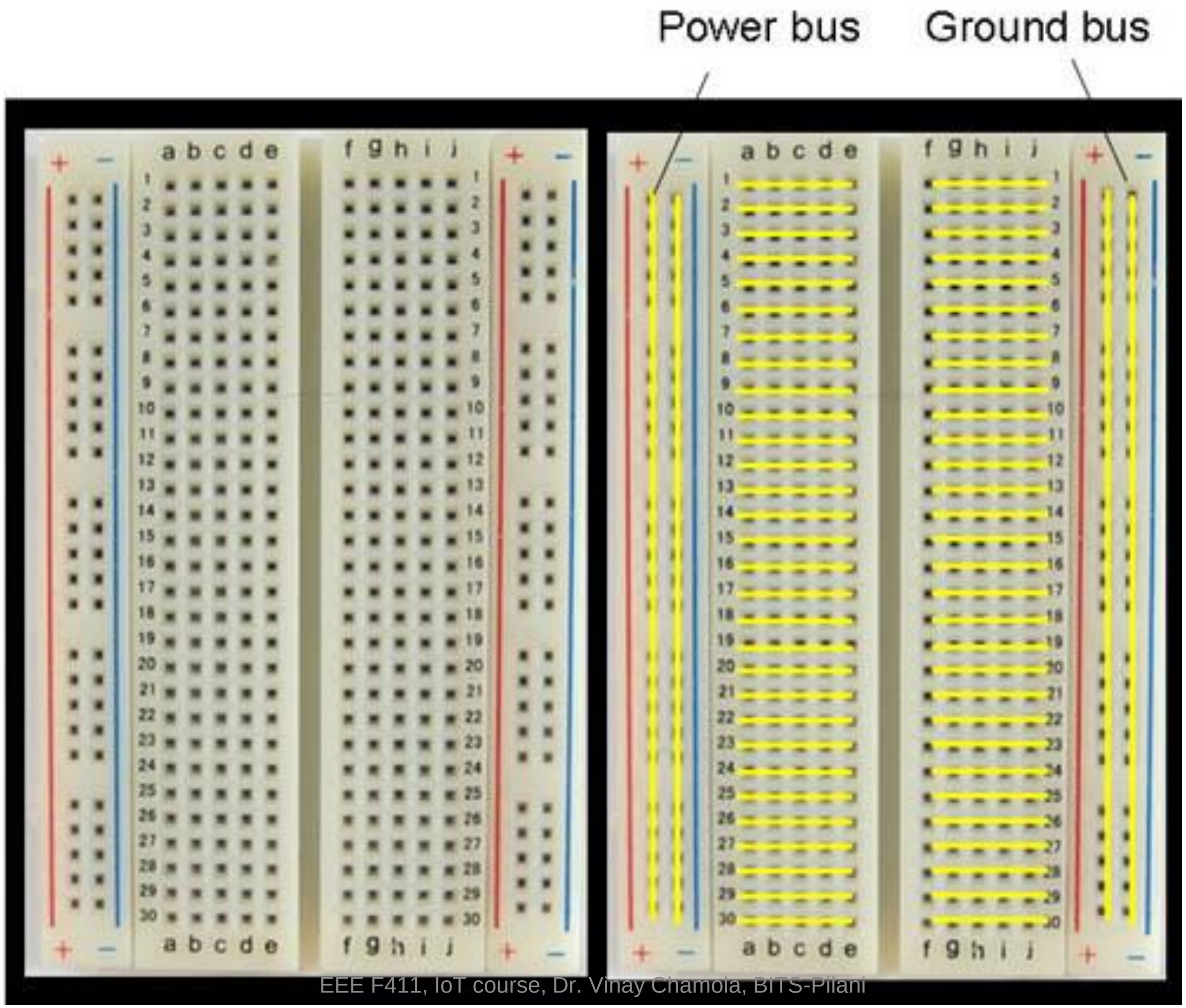
#include

```
#include  <Wire.h>
```

Module 3
Arduino Programming

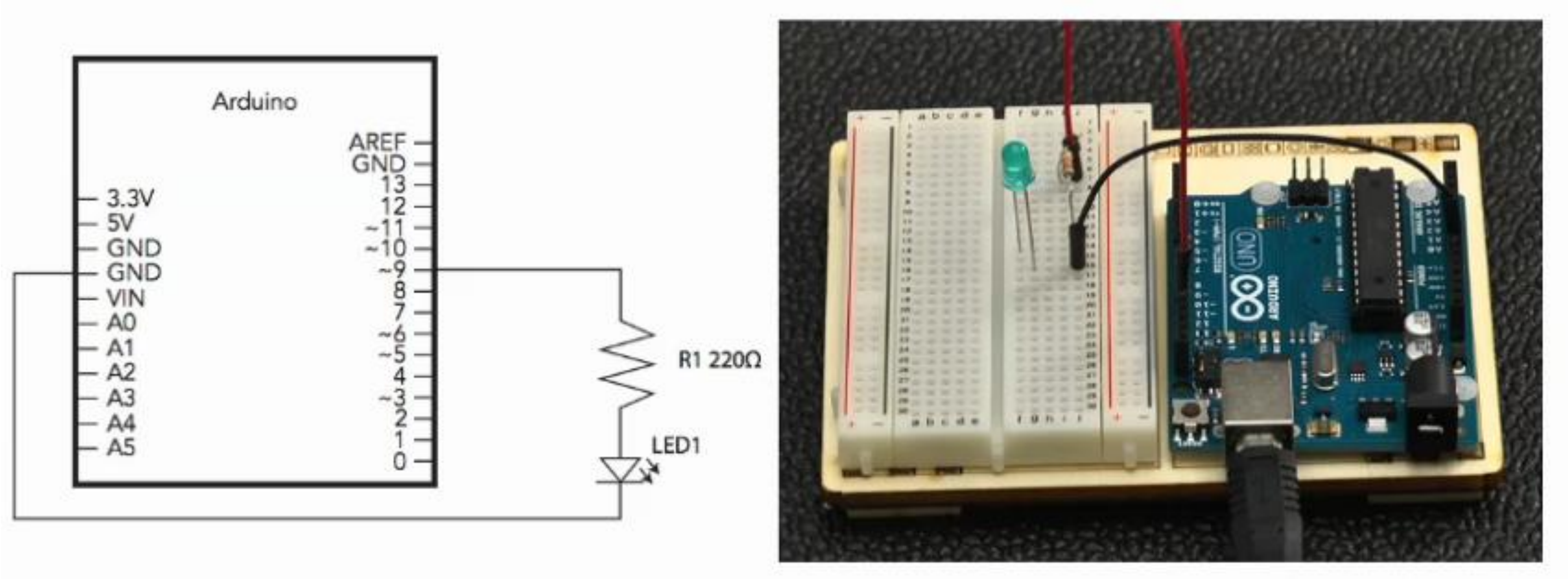
Electrical Components

Solderless Breadboard



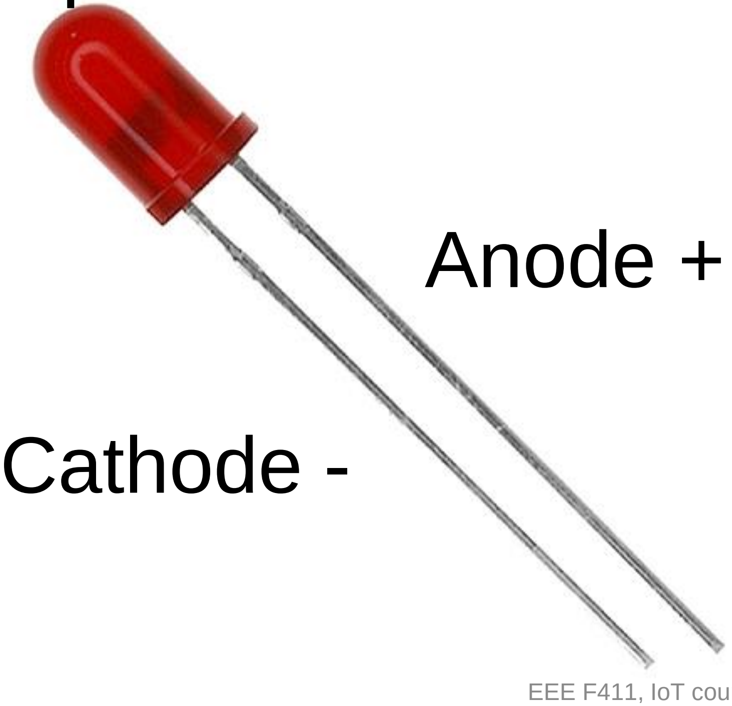
Solderless Breadboard

Prototyping without soldering



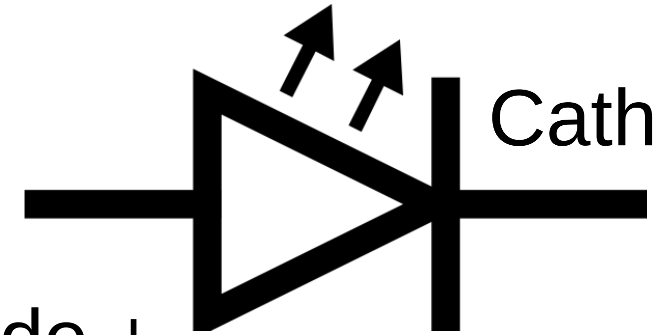
LED

- Light-emitting Diode
- Pasess current one way
- Emits Lights
- spec of current threshold : 20mA



Anode +

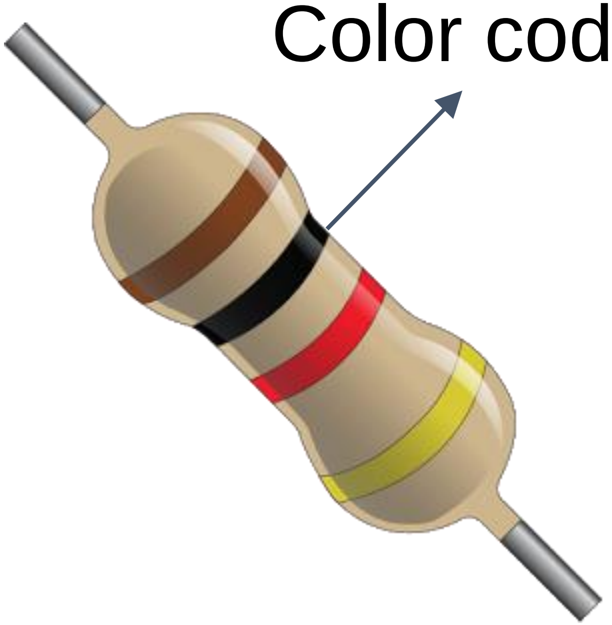
Cathode -



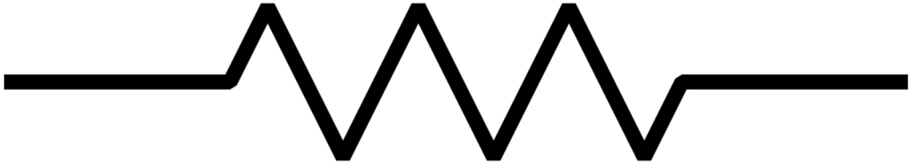
Cathode -

Anode +

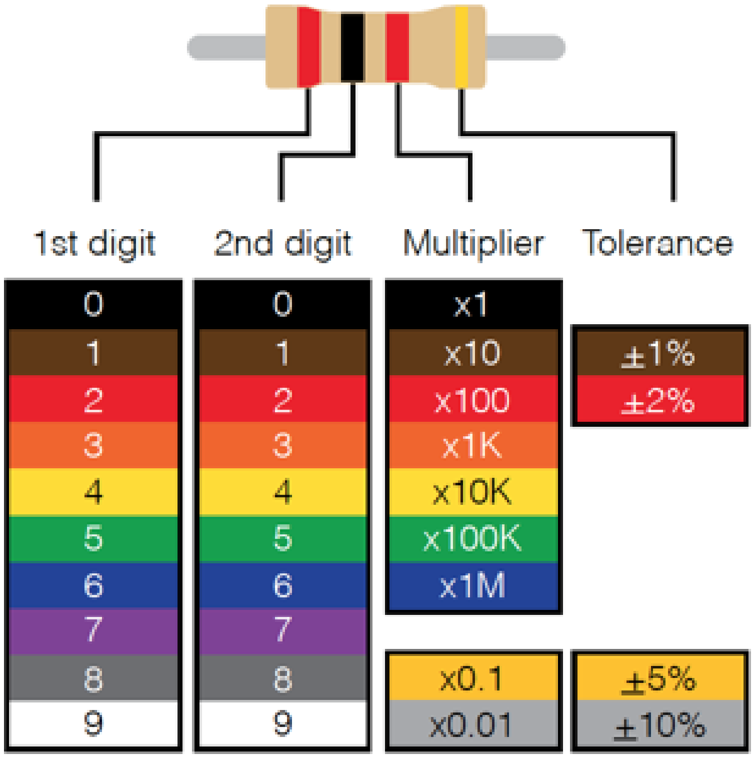
Resistor



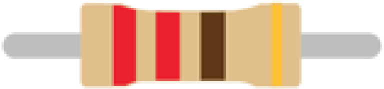
Color code represent the resistance value



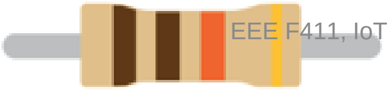
Resistor Color Code



1st digit	2nd digit	Multiplier	Tolerance
0	0	x1	
1	1	x10	±1%
2	2	x100	±2%
3	3	x1K	
4	4	x10K	
5	5	x100K	
6	6	x1M	
7	7		
8	8	x0.1	±5%
9	9	x0.01	±10%



2, 2, x 10 = 220Ω

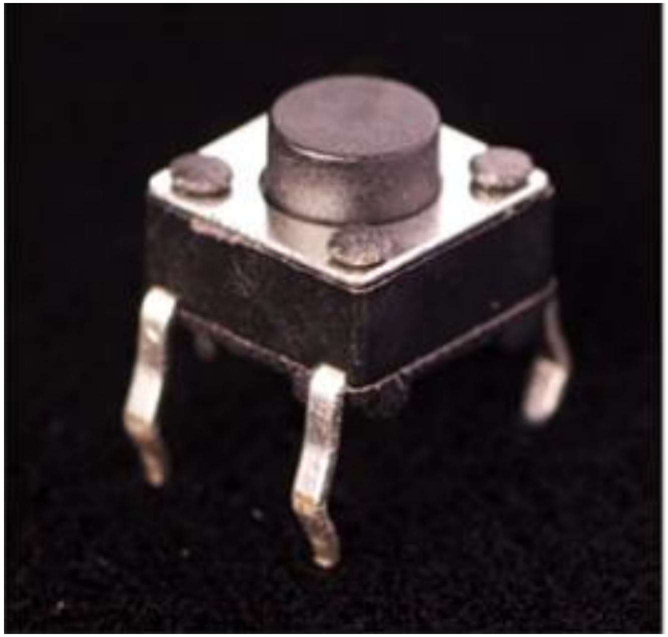


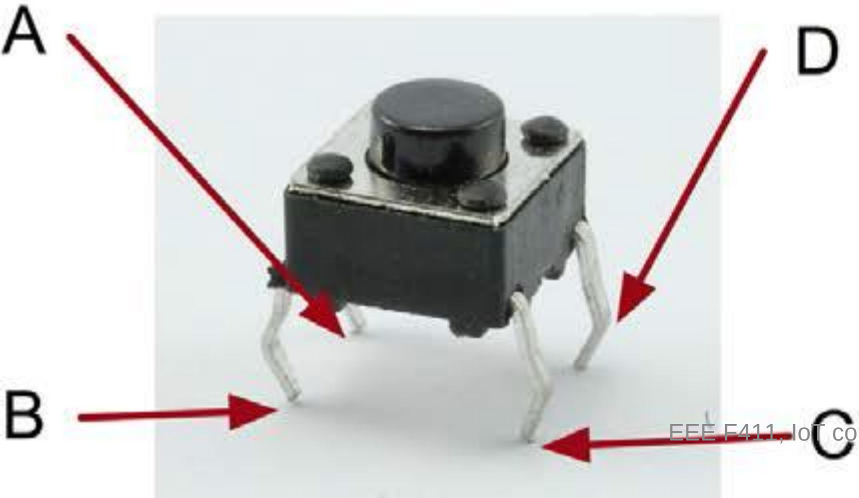
1, 1, x 1,000 = 11KΩ

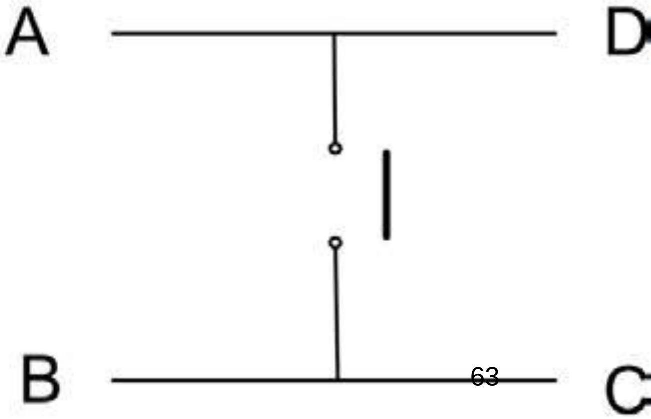
Push Button

Press to Turn On
Release to Turn Off

Generate Digital Input

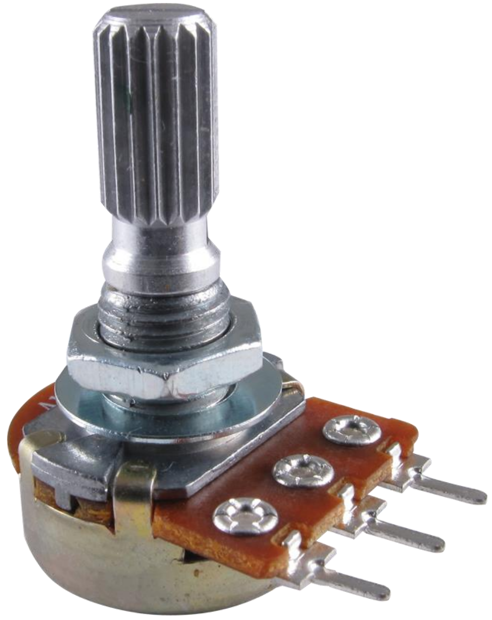


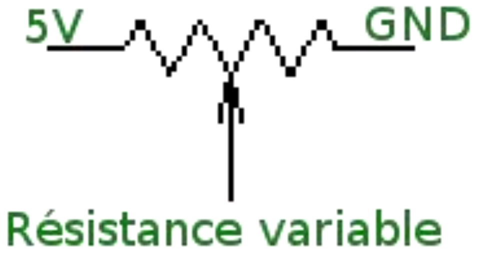


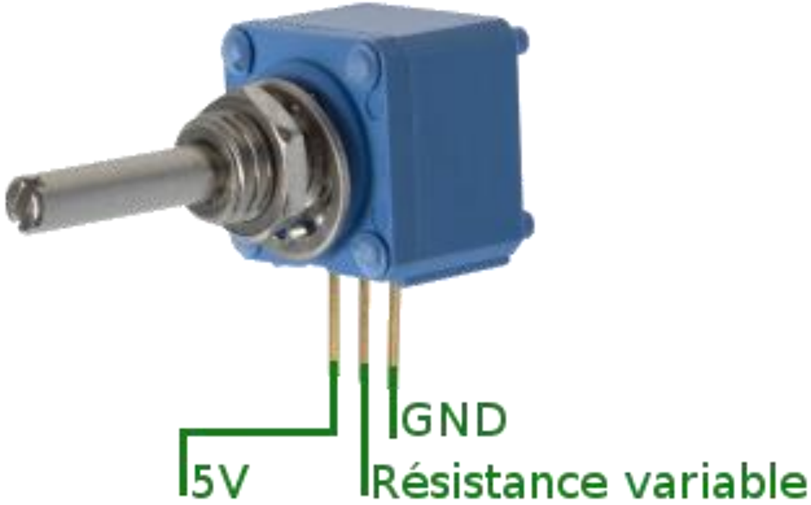


Potentiometer

Variable Resistor

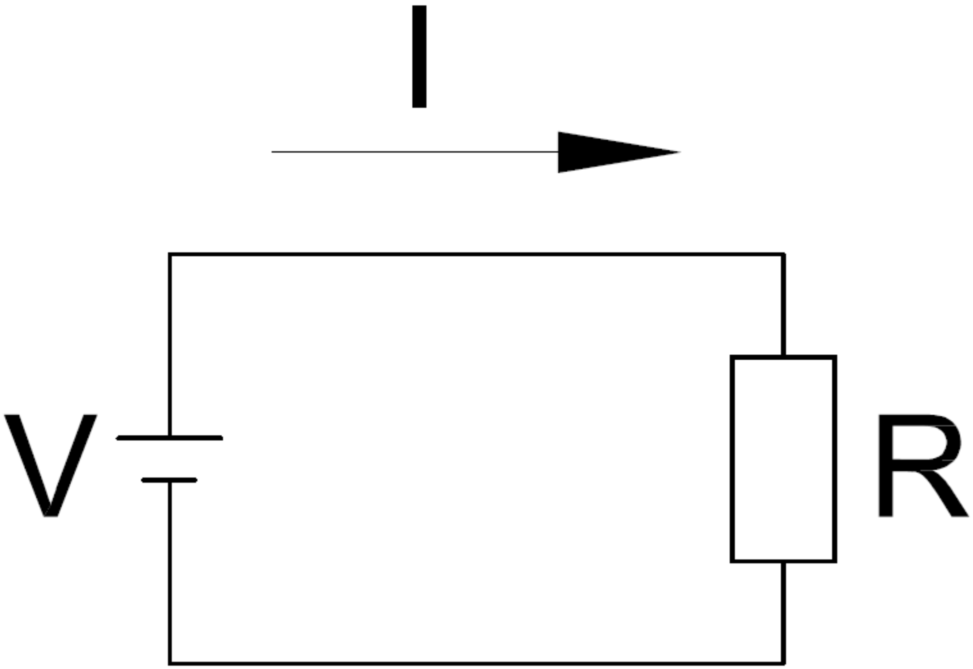






Ohm's Law

Basic Concept: Ohm's Law

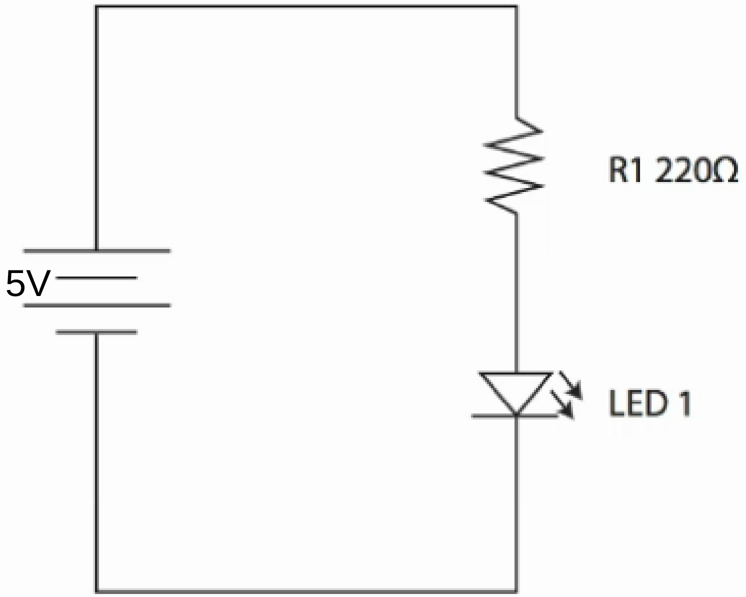


Ohm's Law

$$I = \frac{V}{R}$$

Electric current = Voltage / Resistance

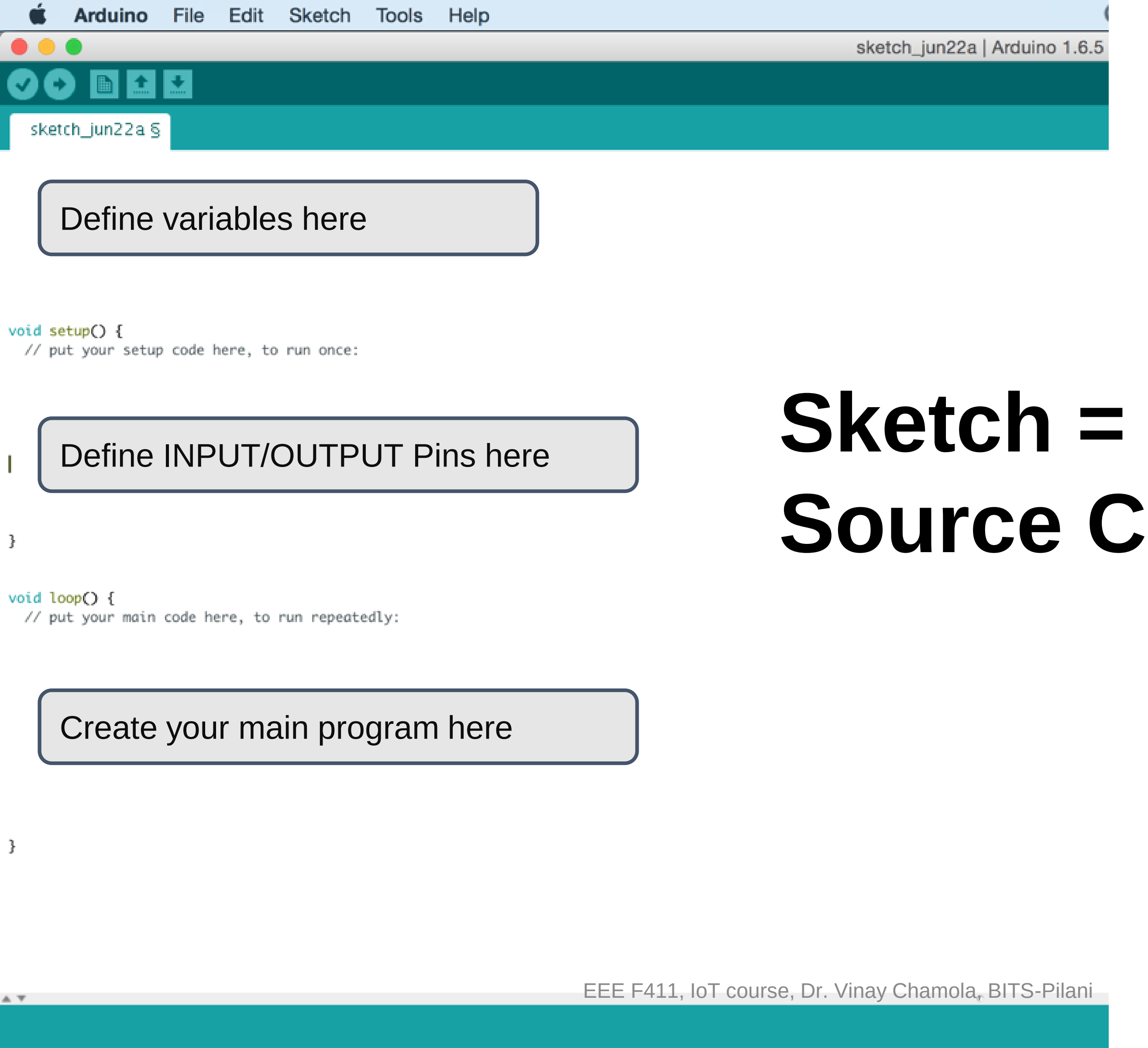
Applying Ohm's Law to LED



Voltage across LED is 2V
Voltage across R1 resistor is 5V
Maximum current for LED is 0.02A

R= 5V/0.02A= 250Ω

Arduino Program Structure



Define variables here

```
void setup() {
  // put your setup code here, to run once:
```

Define INPUT/OUTPUT Pins here

```
}

void loop() {
  // put your main code here, to run repeatedly:
```

Create your main program here

```
}
```

Sketch = Program = Source Code

Pin Setup

PinMode Syntax

```
pinMode(pin, INPUT/OUTPUT)
```

Digital Output

digitalWrite

Syntax:
digitalWrite(pin, HIGH/LOW)

Eg
digitalWrite(9,HIGH)

delay

Pauses the program for the amount of time (in miliseconds) specified as parameter

Syntax
delay(ms)