

C0Ding Workshop Wk02: Yuta Nakayama

Quick Review:Week01

- **Design with Parameter / Algorithm**
 - Constant / Variables
(int, float, char, string)
 - Loop
(for, while)
 - Conditionals
(if-else, switch-case)
 - Function
(size(100,100); rect(10,10,20,20);)

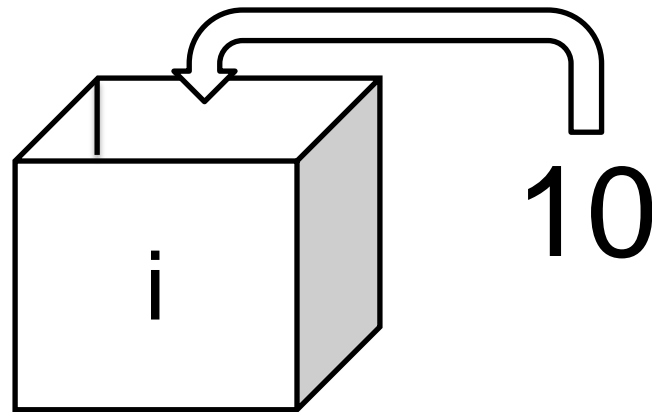
```
int i = 10;
```

Value Assignment

int **i** **=** **10;**

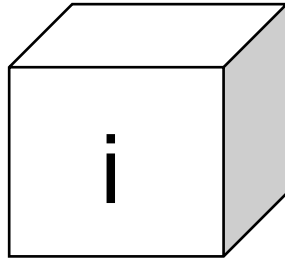
└────────┘ └──┘ └──┘ └──┘

type name assignment operator terminator



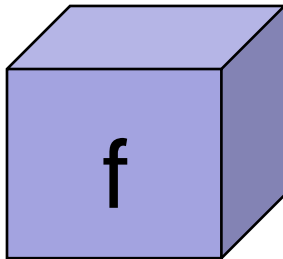
Value Types

int



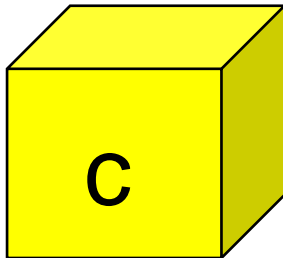
-1, 0, 1, 2 ...

float



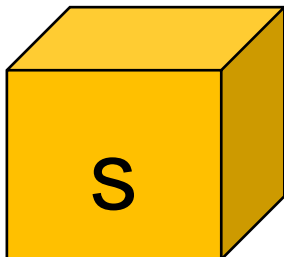
0.5, 1.2, 2.4, -1.1 ...

char



'a','b','c',...'1','2'...

string



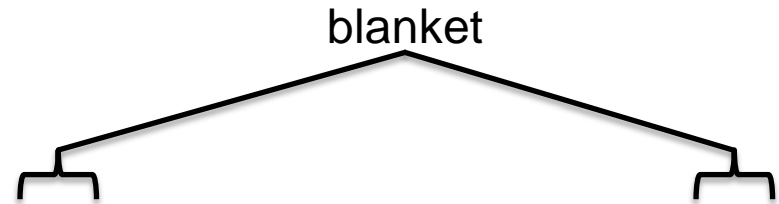
"this","is","teh","tarik!"...

```
int [] a = { 3, 4, 5 };
```

Array

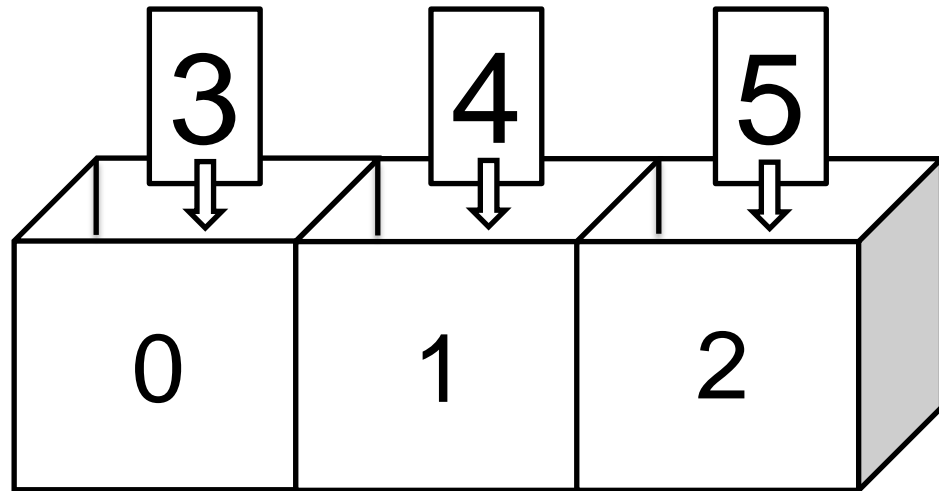
```
int [] a = { 3, 4, 5 };
```

array type name



Value 0 Value 1 Value 2

`a[] =`



if-else

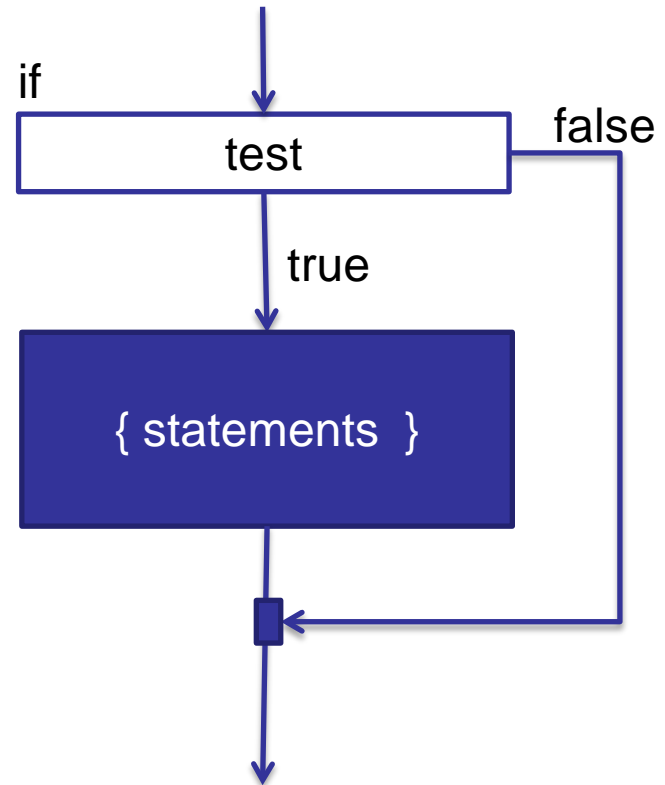
```
if( test ){  
    statements  
}
```


if-else

```
if( test ){
```

```
    statements
```

```
}
```



if-else

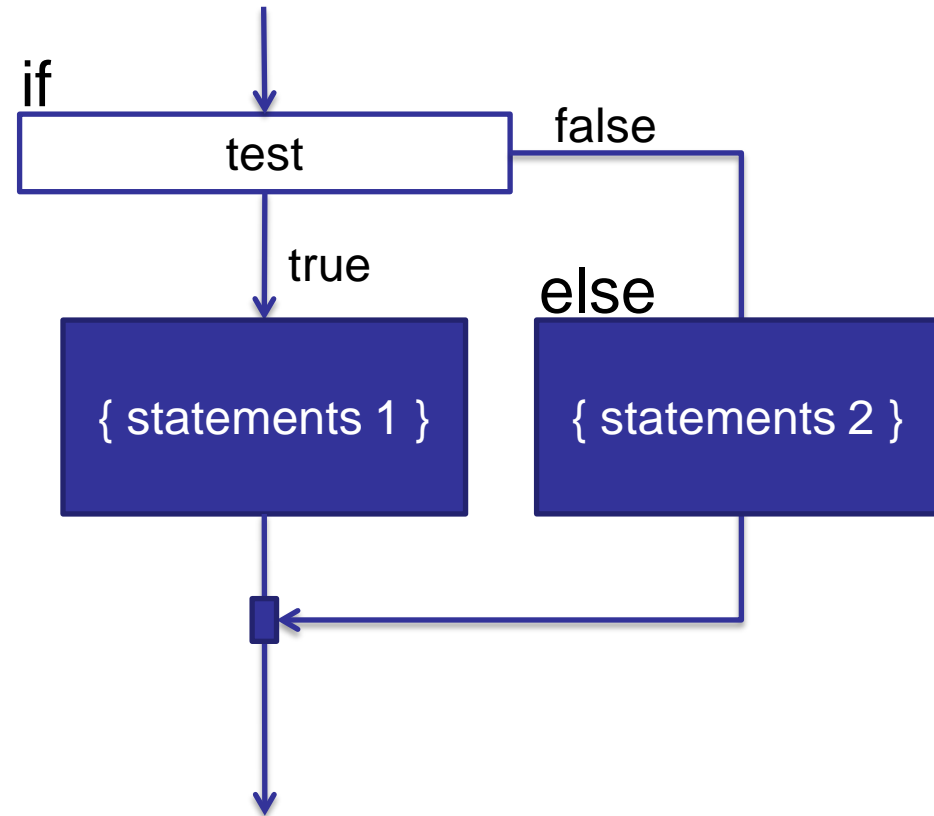
```
if( test ){
```

```
    statements 1
```

```
}else{
```

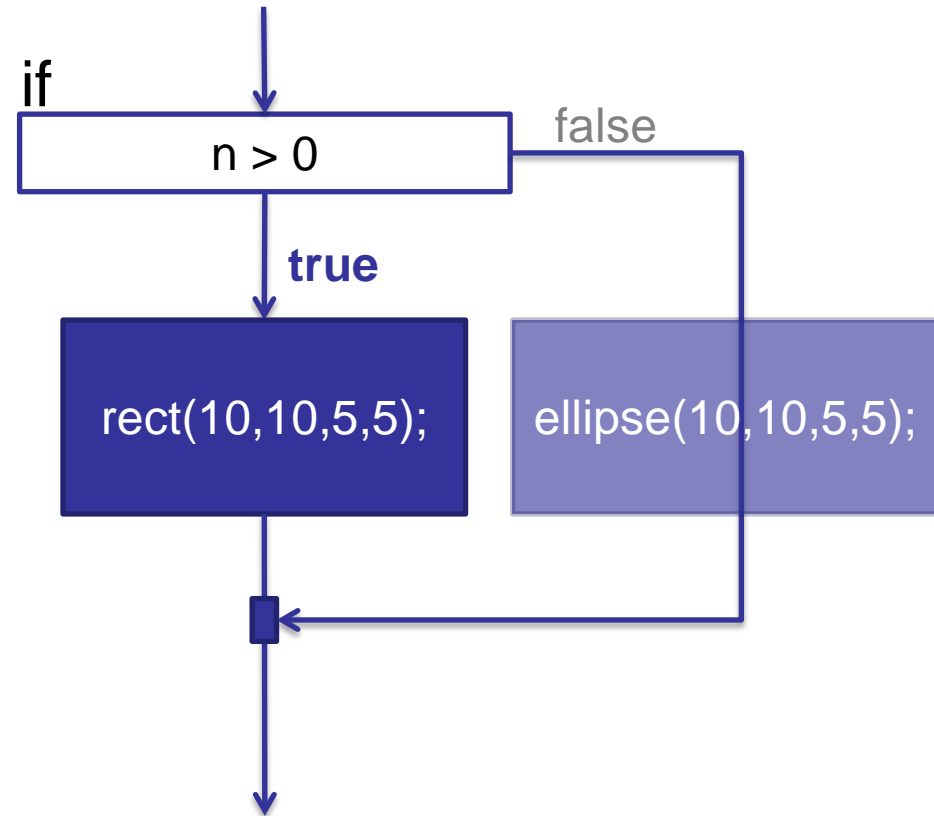
```
    statements 2
```

```
}
```



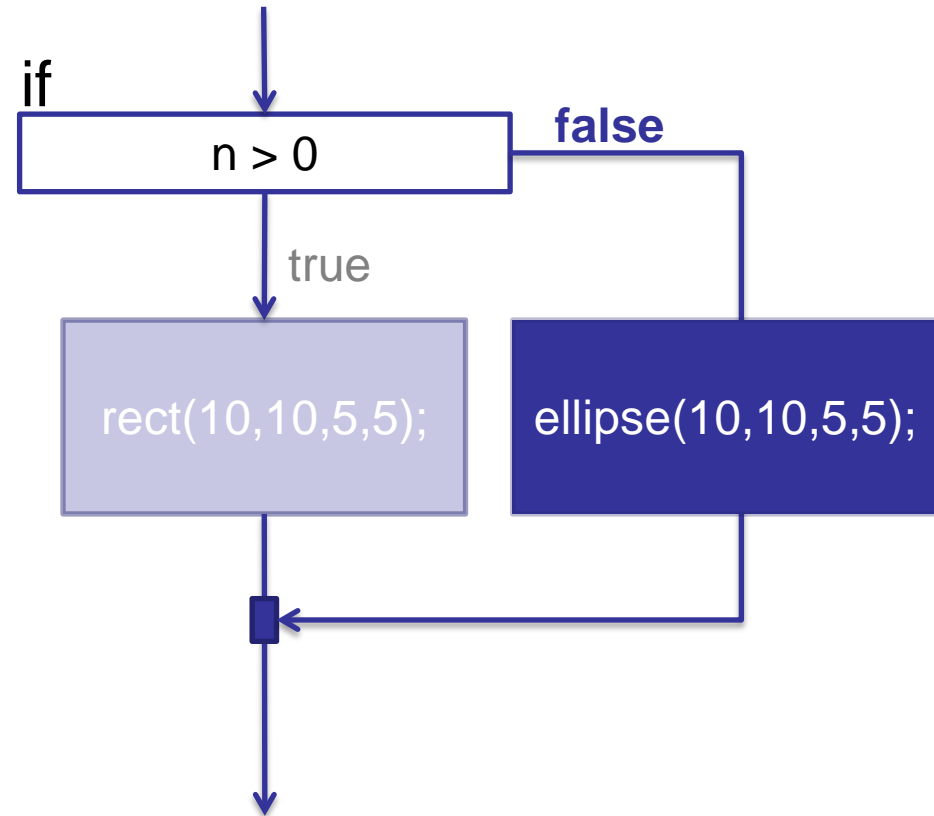
if-else

```
int n = 1;  
if( n > 0 ){  
    rect(10,10,5,5);  
}else{  
    ellipse(10,10,5,5);  
}
```



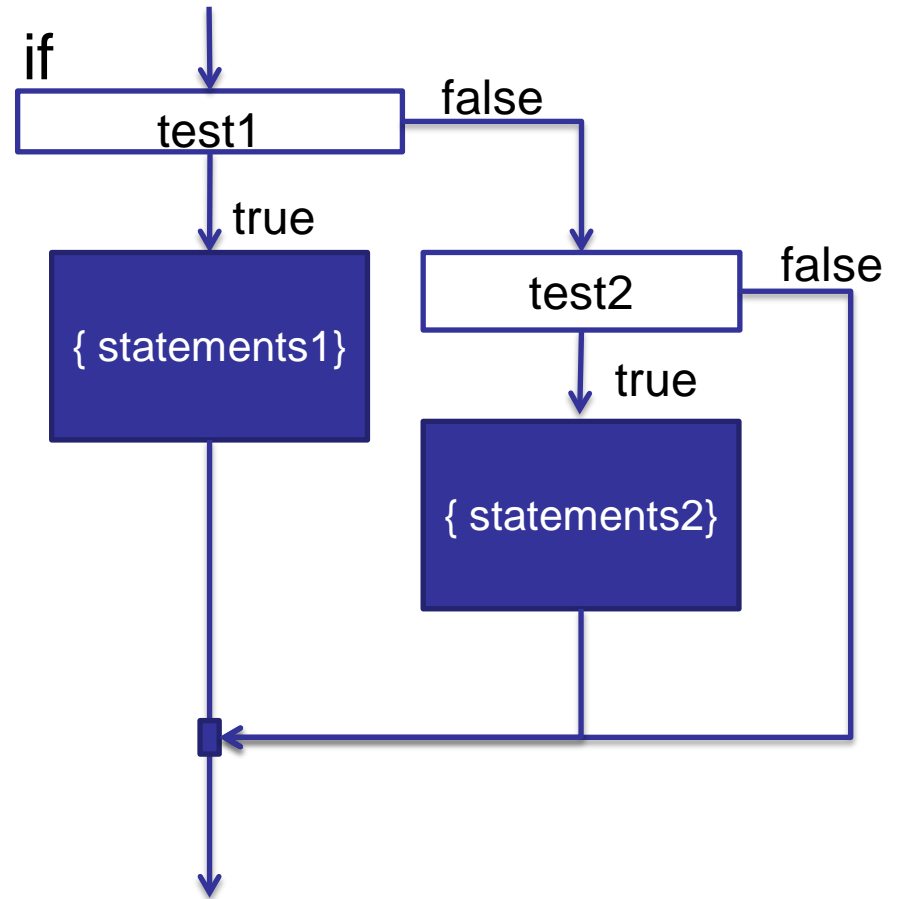
if-else

```
int n = 1;  
if( n < 0 ){  
    rect(10,10,5,5);  
}else{  
    ellipse(10,10,5,5);  
}
```



if-else if

```
if( test1 ){  
    statements 1  
}else if( test2 ){  
    statements 2  
}
```

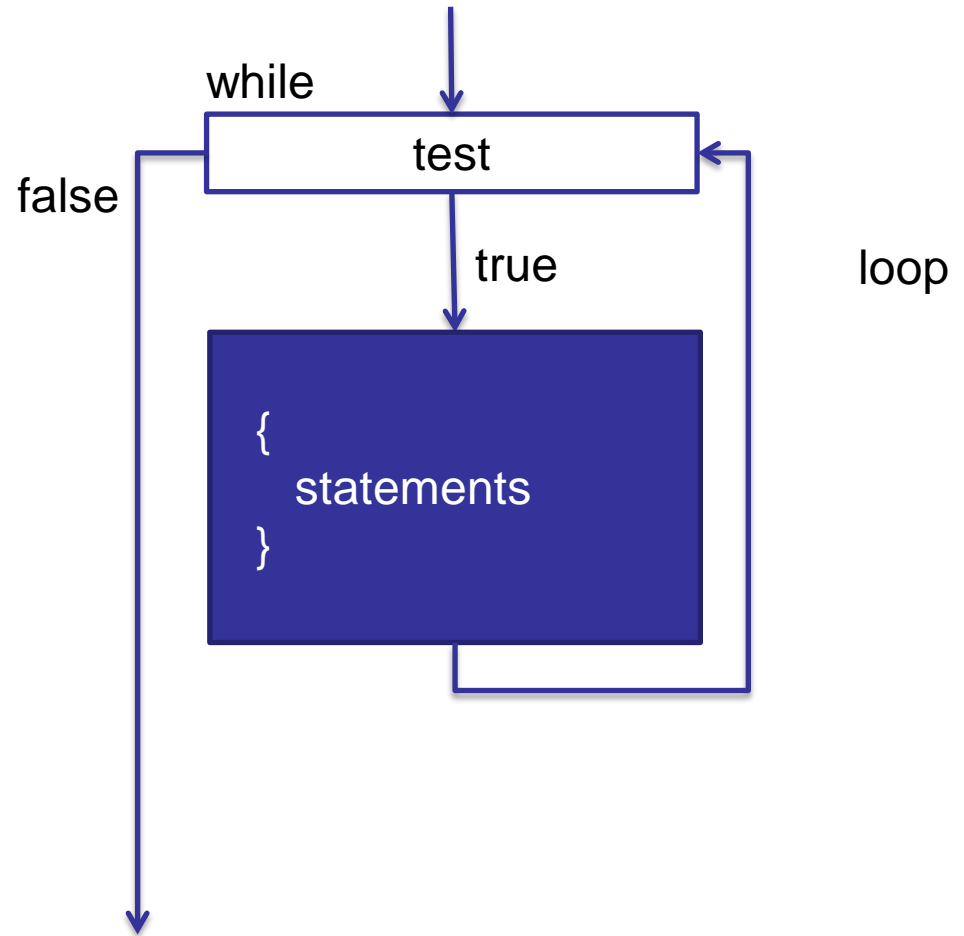


while-loop

```
while( test ){
```

statements

```
}
```

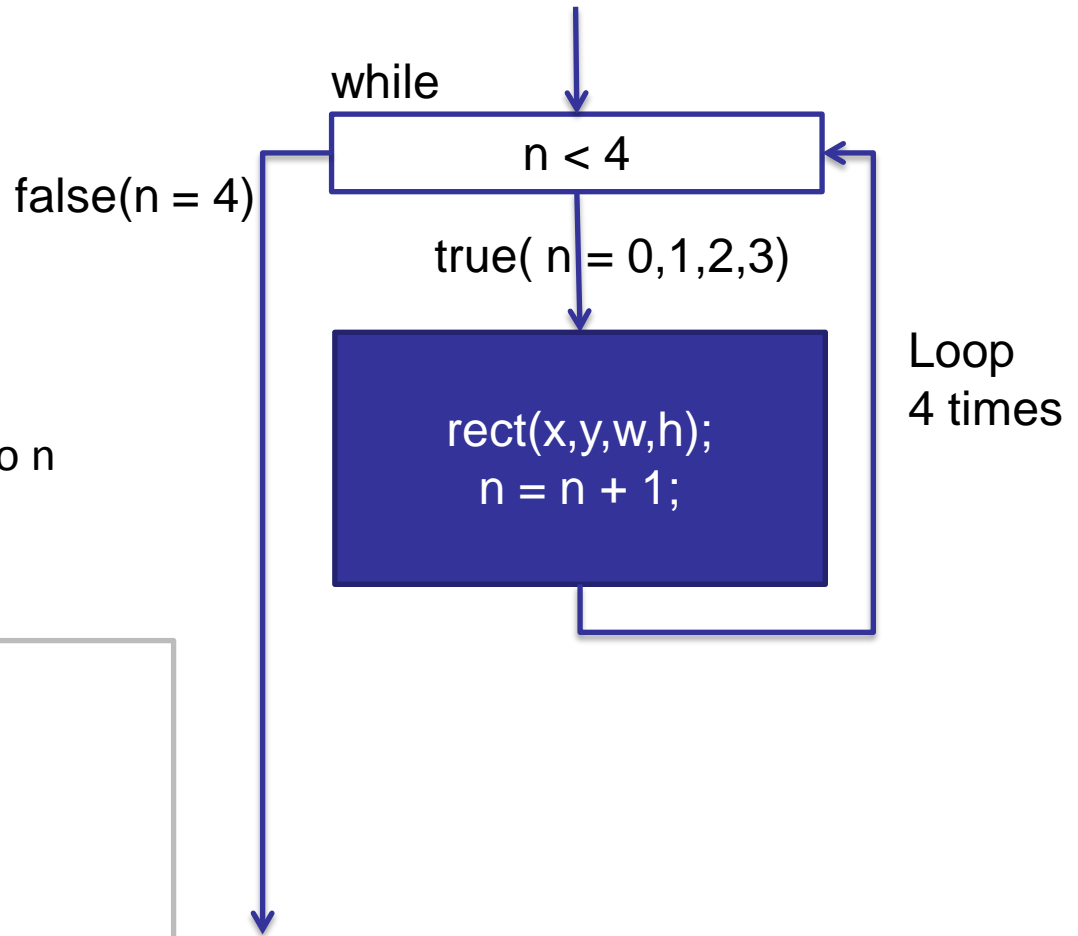


while-loop

```
int n = 0;  
while( n < 4 ){  
    println(n); //print n  
    n = n + 1;  //add 1 to n  
}
```

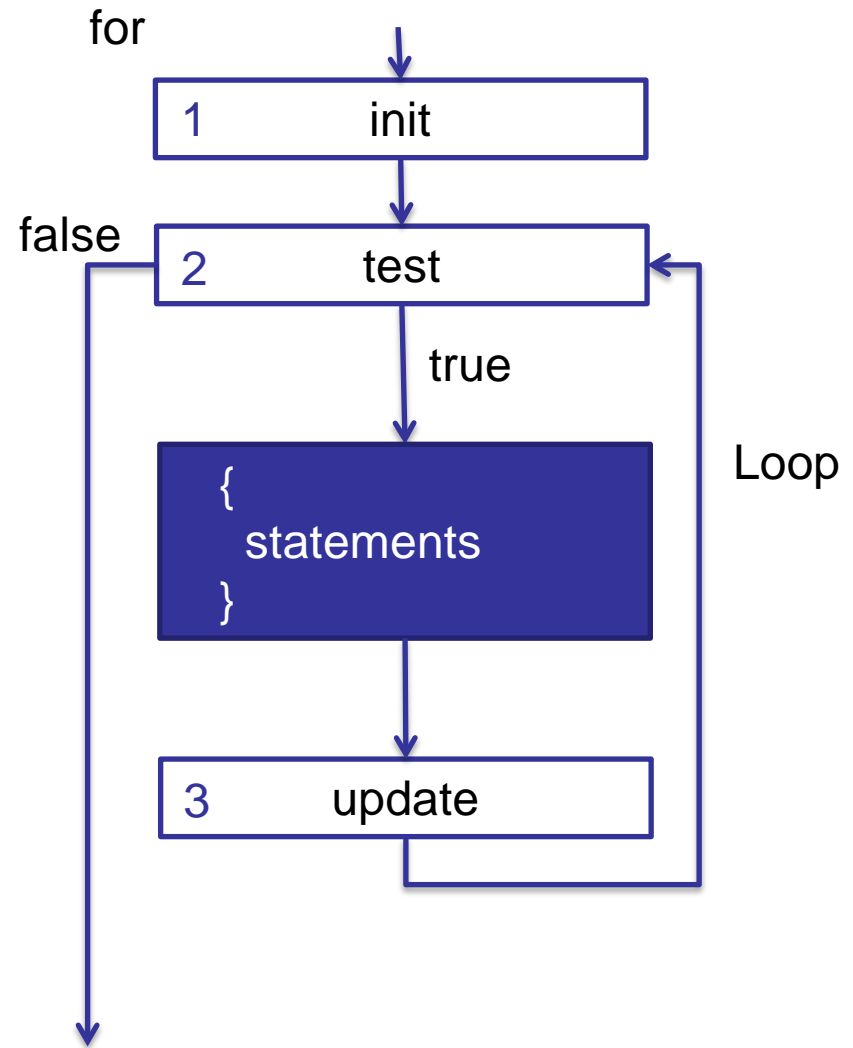
result:

0
1
2
3



for-loop

```
for(1init;2test;3update){  
    statements  
}
```

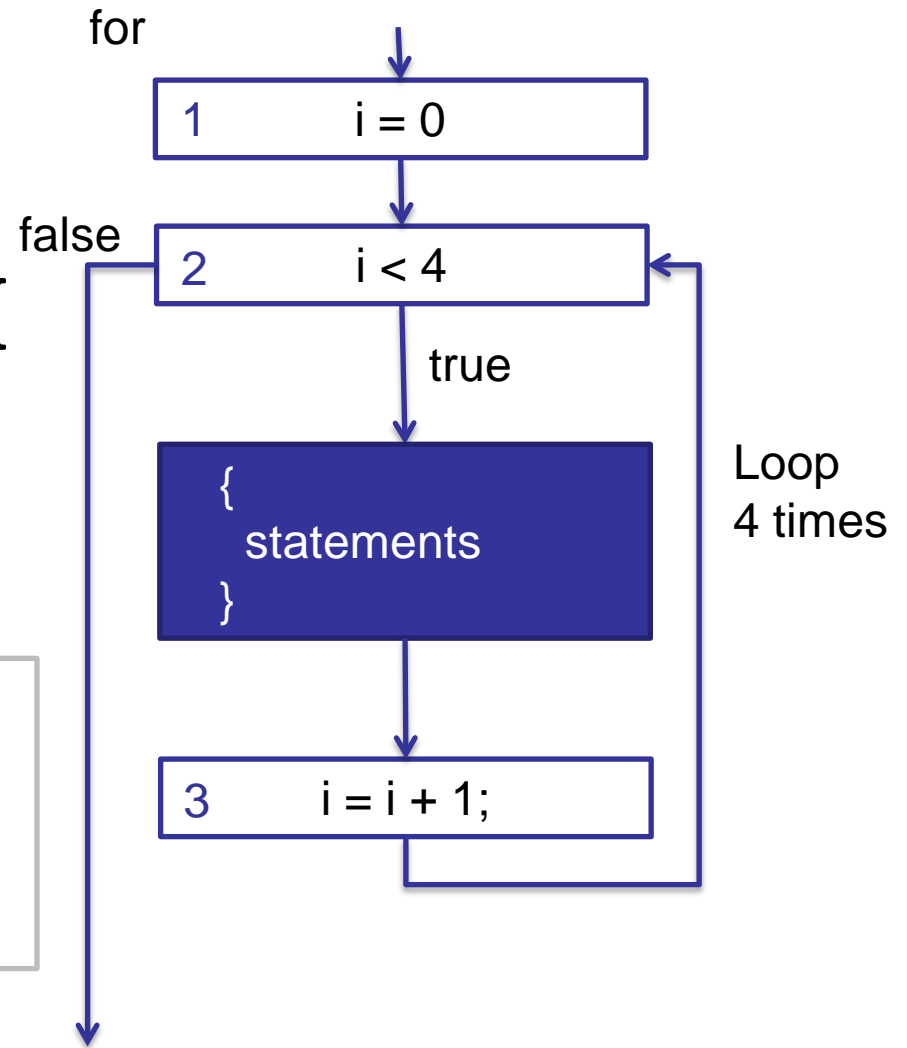


for-loop

```
int i;  
for(1i = 0; 2i < 4; 3i = i + 1 ){  
    println(i);  
}
```

result:

```
0  
1  
2  
3
```



void setup() / void draw()?

```
/**
 * Brightness
 * by Rusty Robison.
 *
 * Brightness is the relative lightness or darkness of a color.
 * Move the cursor vertically over each bar to alter its brightness.
 *
 * Updated 28 February 2010.
 */

int barWidth = 5;
int lastBar = -1;

void setup() {
  size(200, 200);
  colorMode(HSB, 360, 100, height);
  noStroke();
  background(0);
}

void draw() {
  int whichBar = mouseX / barWidth;
  if (whichBar != lastBar) {
    int barX = whichBar * barWidth;
    fill(barX, 100, mouseY);
    rect(barX, 0, barWidth, height);
    lastBar = whichBar;
  }
}
```

void setup() / void draw()?

```
/**
 * Brightness
 * by Rusty Robison.
 *
 * Brightness is the relative lightness or darkness of a color.
 * Move the cursor vertically over each bar to alter its brightness.
 *
 * Updated 28 February 2010.
 */
```

```
int barWidth = 5;
int lastBar = -1;
```

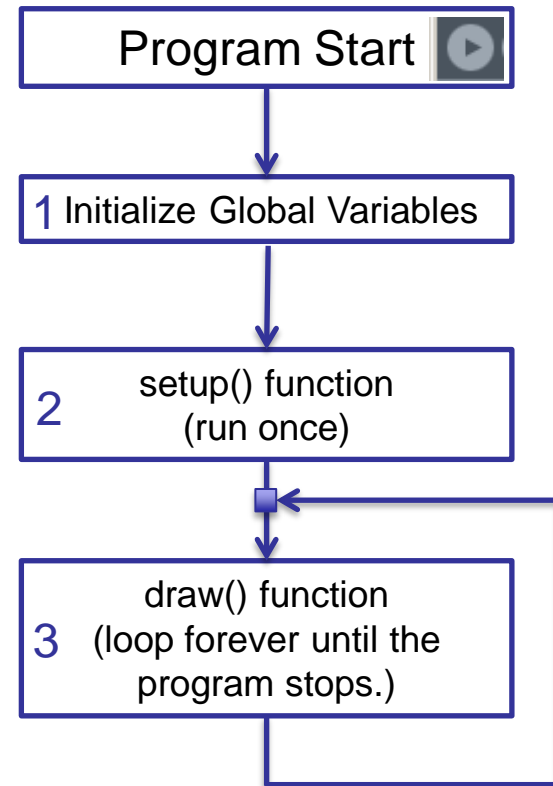
1

```
void setup() {
  size(200, 200);
  colorMode(HSB, 360, 100, height);
  noStroke();
  background(0);
}
```

2

```
void draw() {
  int whichBar = mouseX / barWidth;
  if (whichBar != lastBar) {
    int barX = whichBar * barWidth;
    fill(barX, 100, mouseY);
    rect(barX, 0, barWidth, height);
    lastBar = whichBar;
  }
}
```

3



void?

```
void setup(){  
    size(400,400);  
    int i = 10;  
}
```

void: return nothing.

return value function name

```
void setup( ){  
    size(400,400);  
    int i = 10;  
}
```

function block

Creating a function.

return value function name parameter1 parameter2

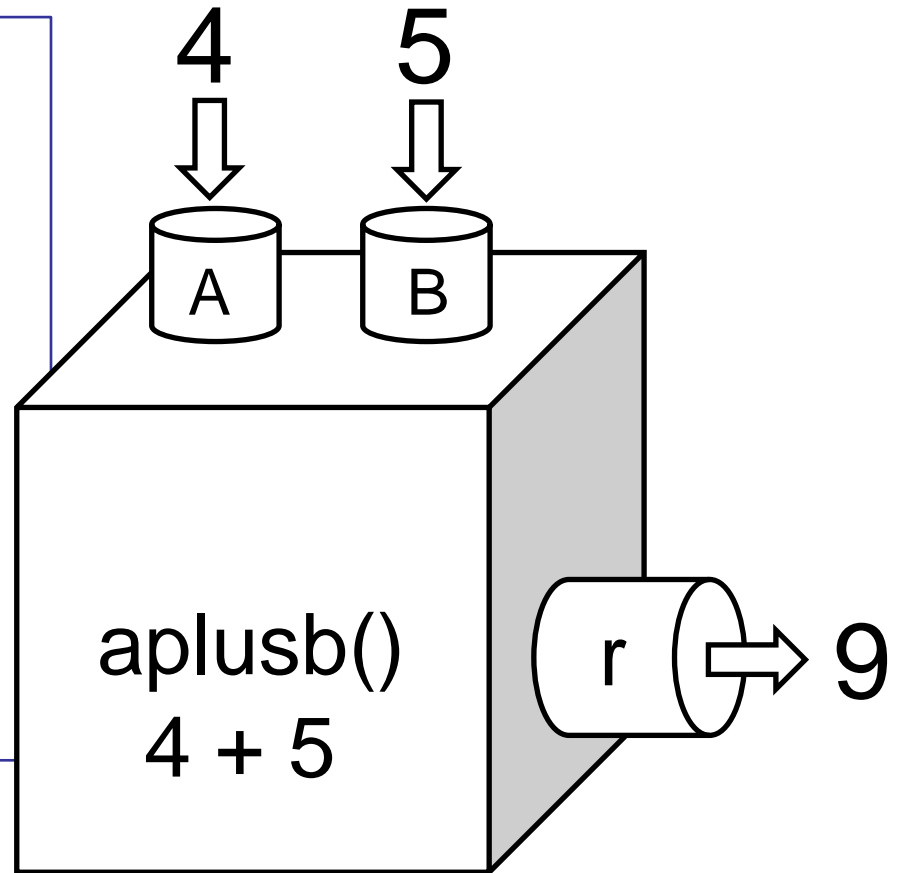
```
int      aplusb(int A, int B){  
    int r; } Local Variable  
    r = A + B;  
    return r; } Return value statement.  
}
```

function block

Function()

```
int aplusb(int A, int B){  
    int r;  
    r = A + B;  
    return r;  
}
```

```
int i = aplusb(4,5);  
// i == 9
```



Ex: Create a function for your font drawing

```
int x = 100;
int y = 100;

void drawMyF(int x, int y){

}

void setup(){
  size(400,400);
  background(255);
  stroke(0);
  fill(255);
}

void draw(){

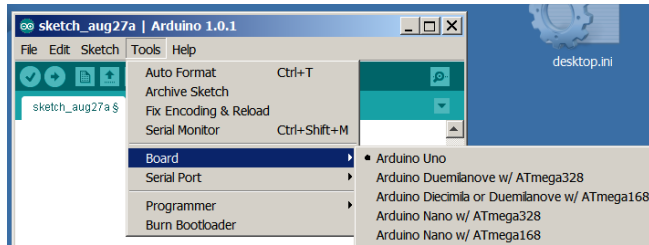
  drawMyF(x,y);
}
```


Week02: Things to learn today.

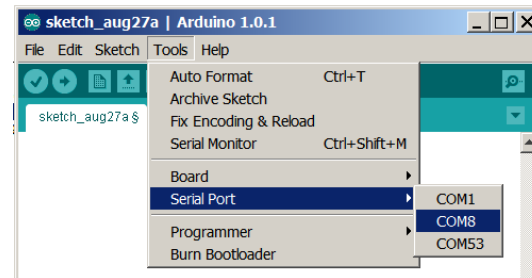
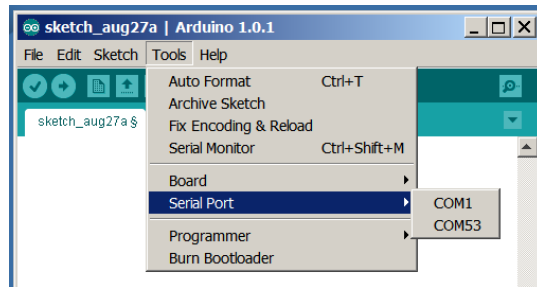
- Basic Electronics
 - (Tools/ LED/ Potentiometer)
- Arduino Programming Basic
 - `setup();/ loop();/`
 - `digitalWrite()` /`digitalRead()`
 - `analogWrite()` / `analogRead()`
- Sensor/Actuator modules
 - (Light/Touch/Pressure/Servo/DC motor)

Setting up Arduino(windows)

1. Install Arduino Software & Driver (<http://arduino.cc>)
2. Select a type of Arduino board “Arduino Uno”



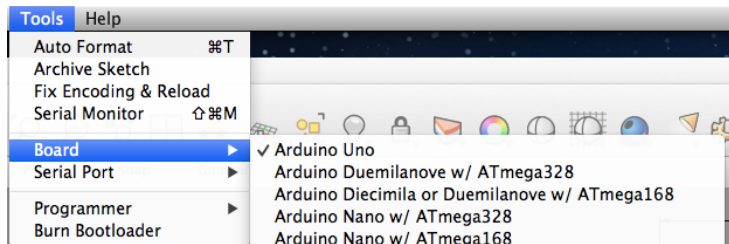
3. Unplug and re-insert the USB cable and select the Port appeared.



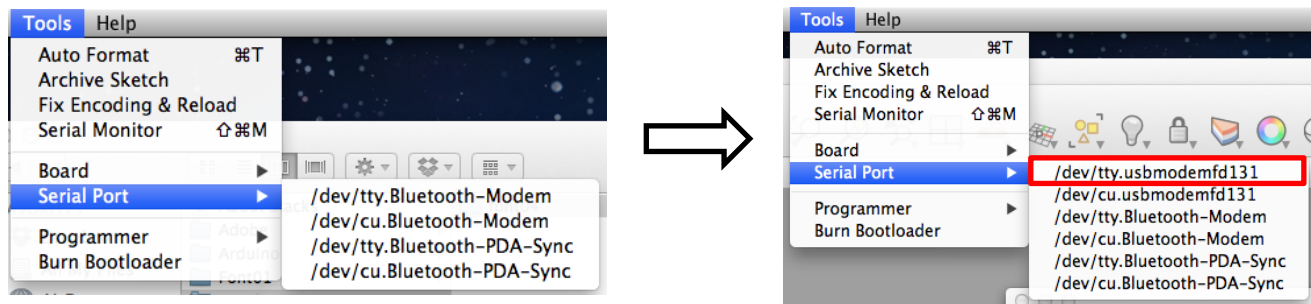
4. Open Sketchbook -> Examples -> Digital -> Blink

Setting up Arduino(mac)

1. Install Arduino Software & Driver (<http://arduino.cc>)
2. Select a type of Arduino board “Arduino UNO”

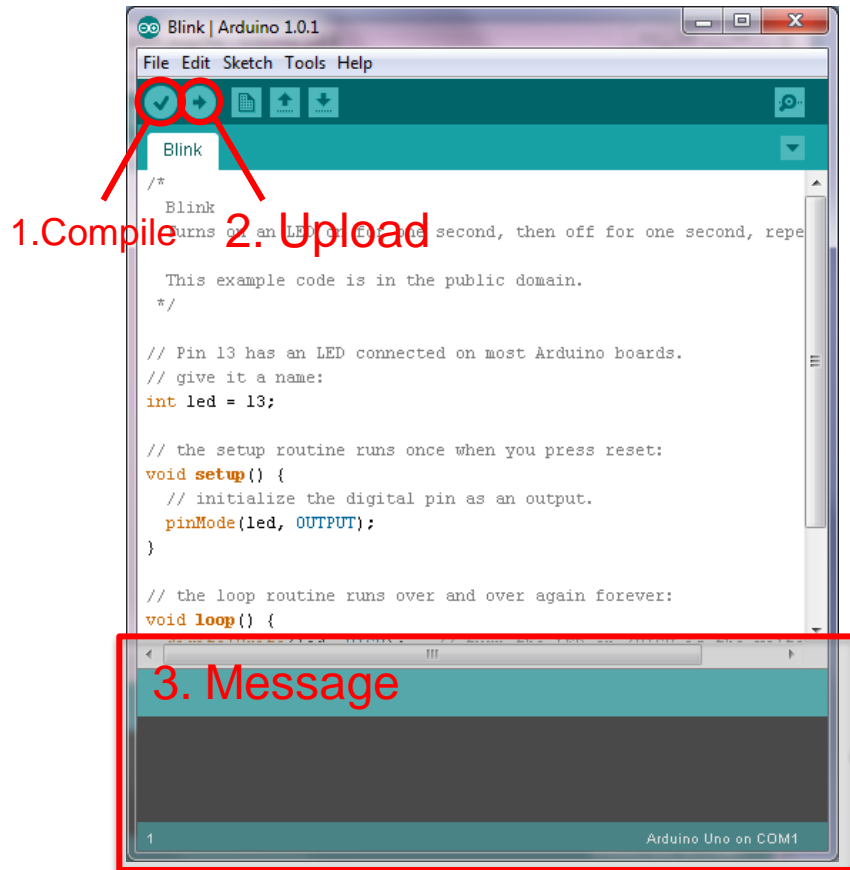


3. Unplug and re-insert the USB cable and select the Port appeared (/dev/tty.**)

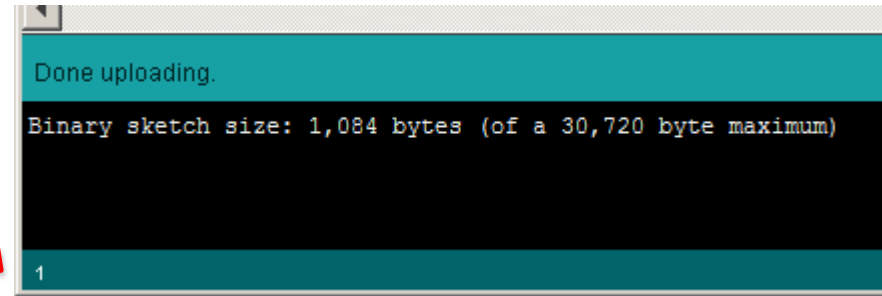


4. Open Sketchbook -> Examples -> Digital -> Blink

Running Arduino Code

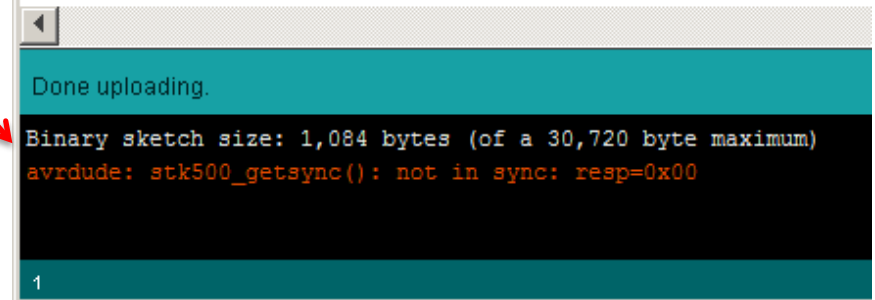


Success:

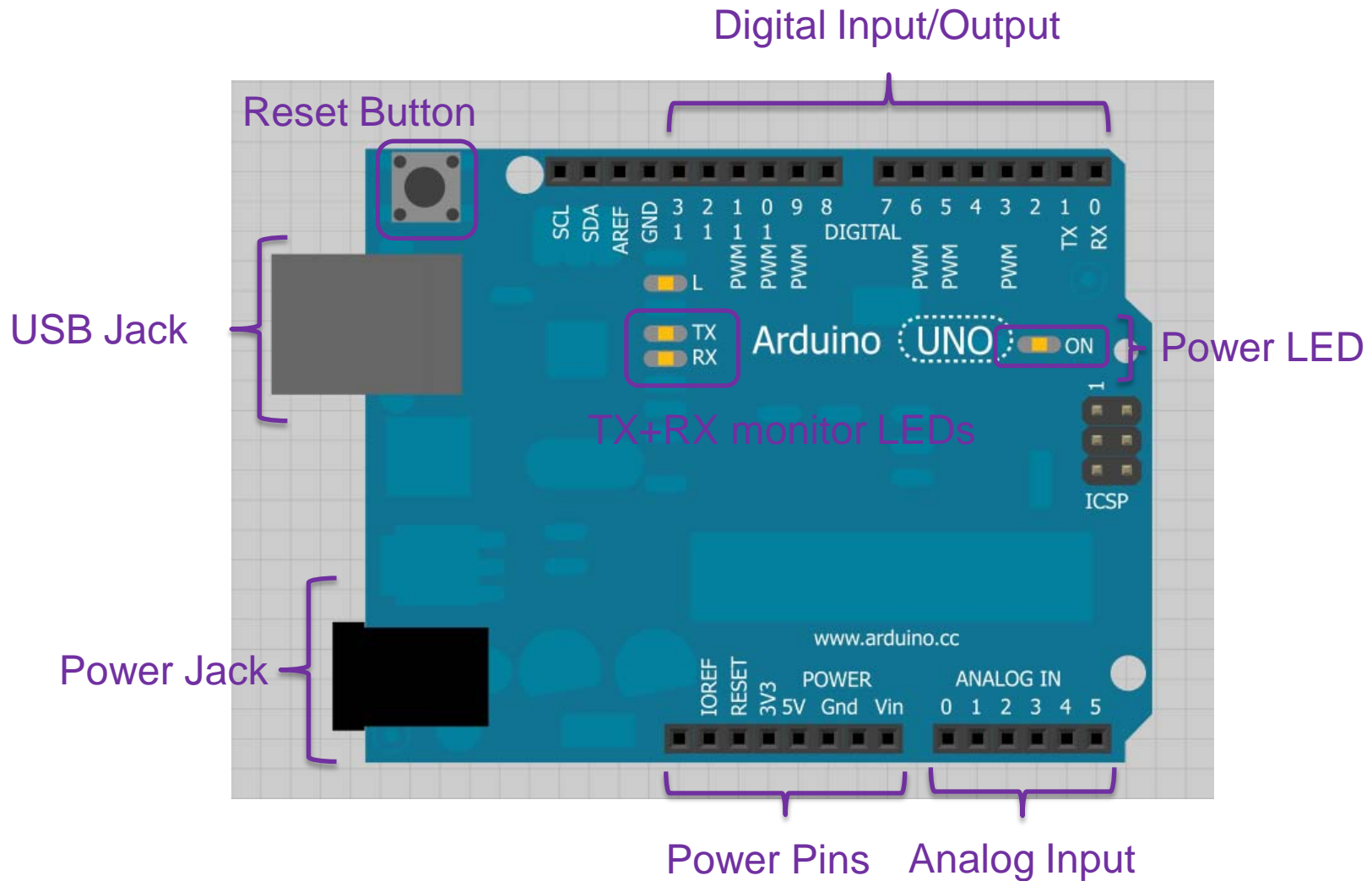


Fail:

avrdude: stk500_getsync(): not in sync: resp=0x00



Arduino Board



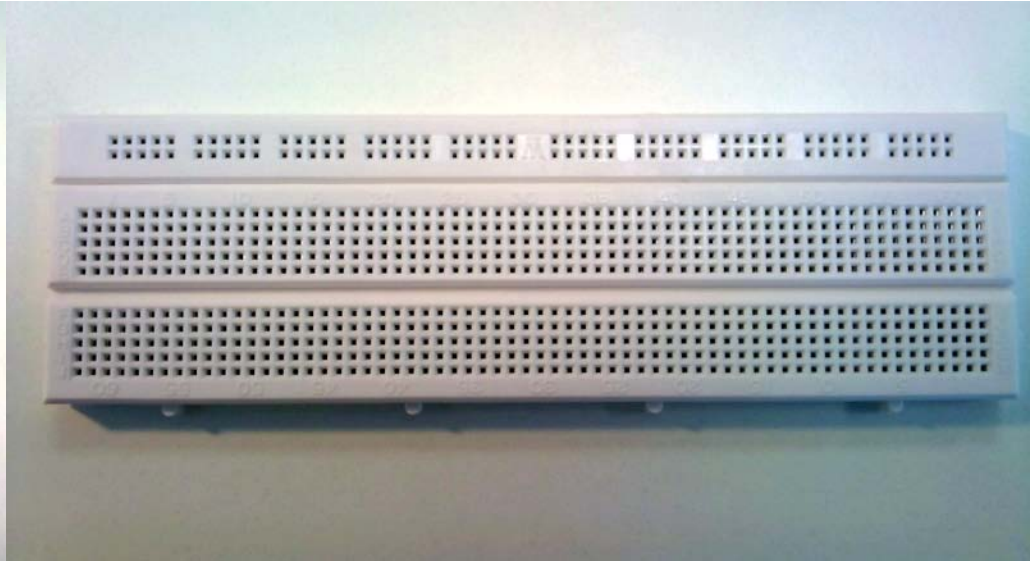
Basic Tools



Multi-meter



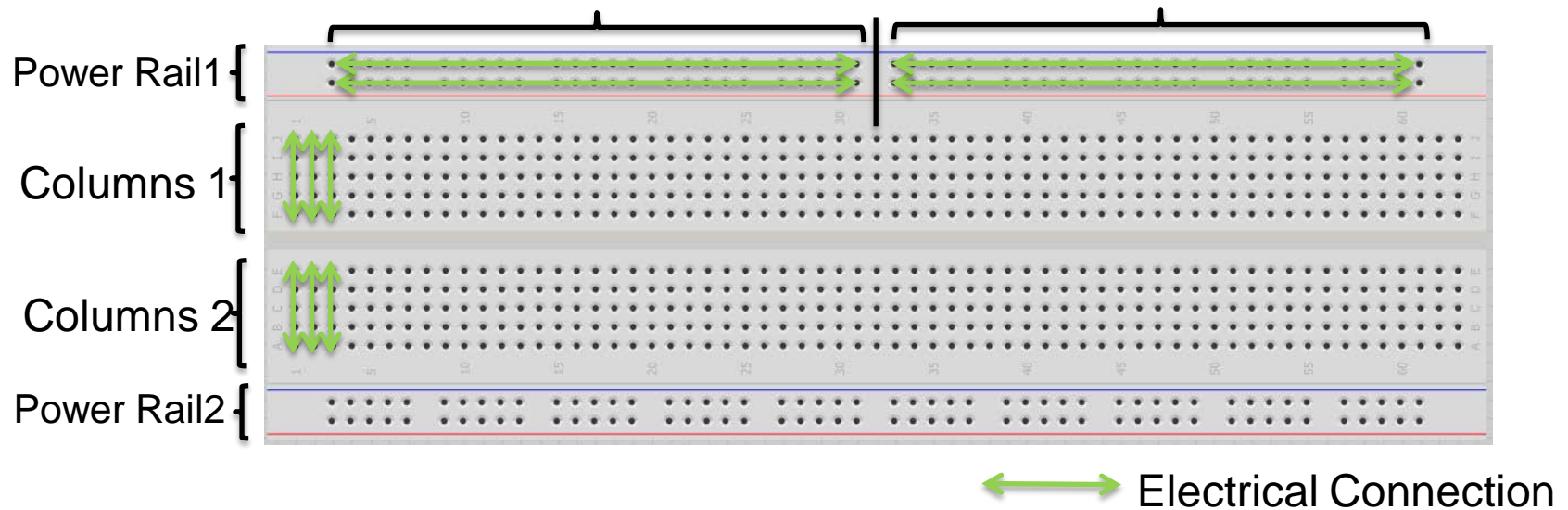
Cable Stripper



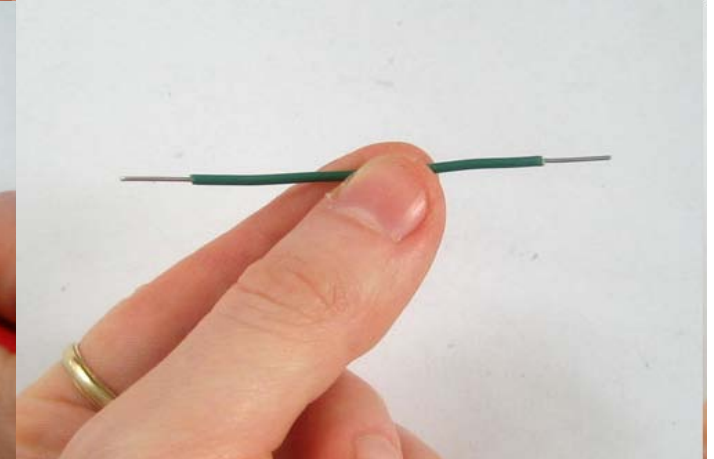
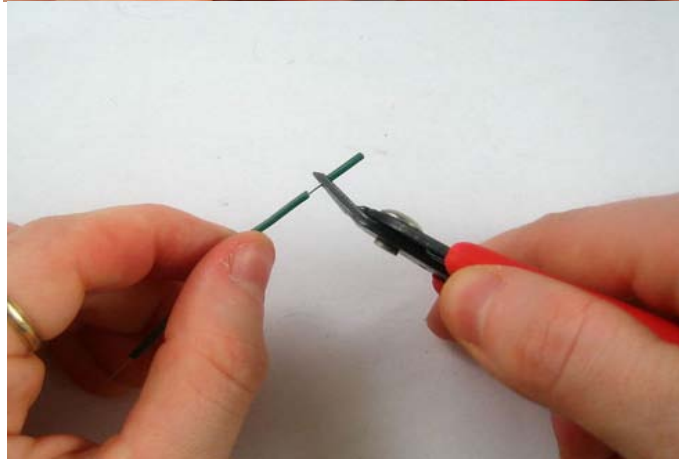
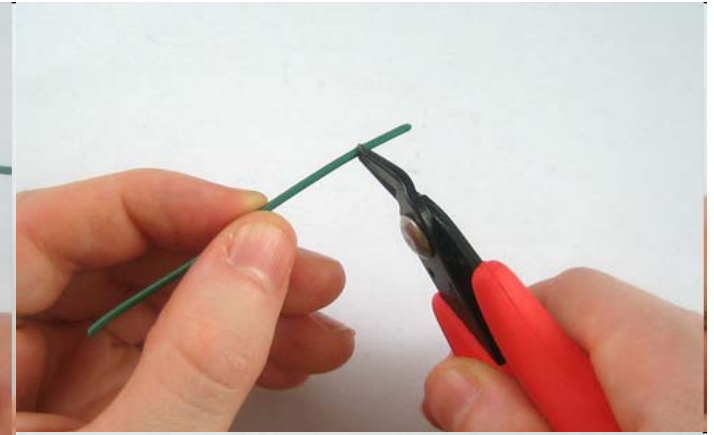
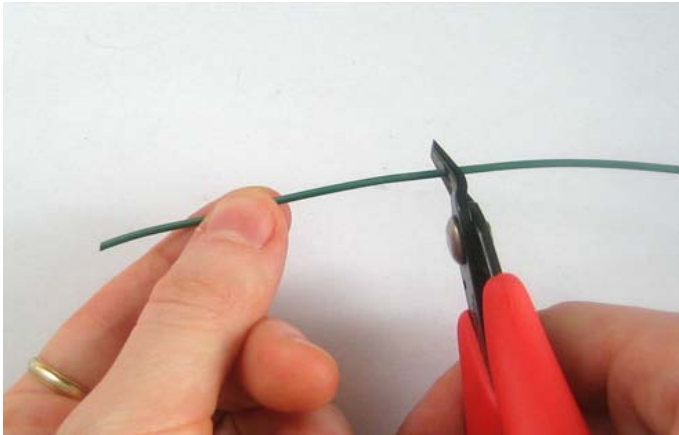
Breadboard

Solderless Breadboard

- A **breadboard** is a reusable construction base for prototyping of electronics.
- No soldering -> insert wire / components
- Reusable/ Easy to change connections

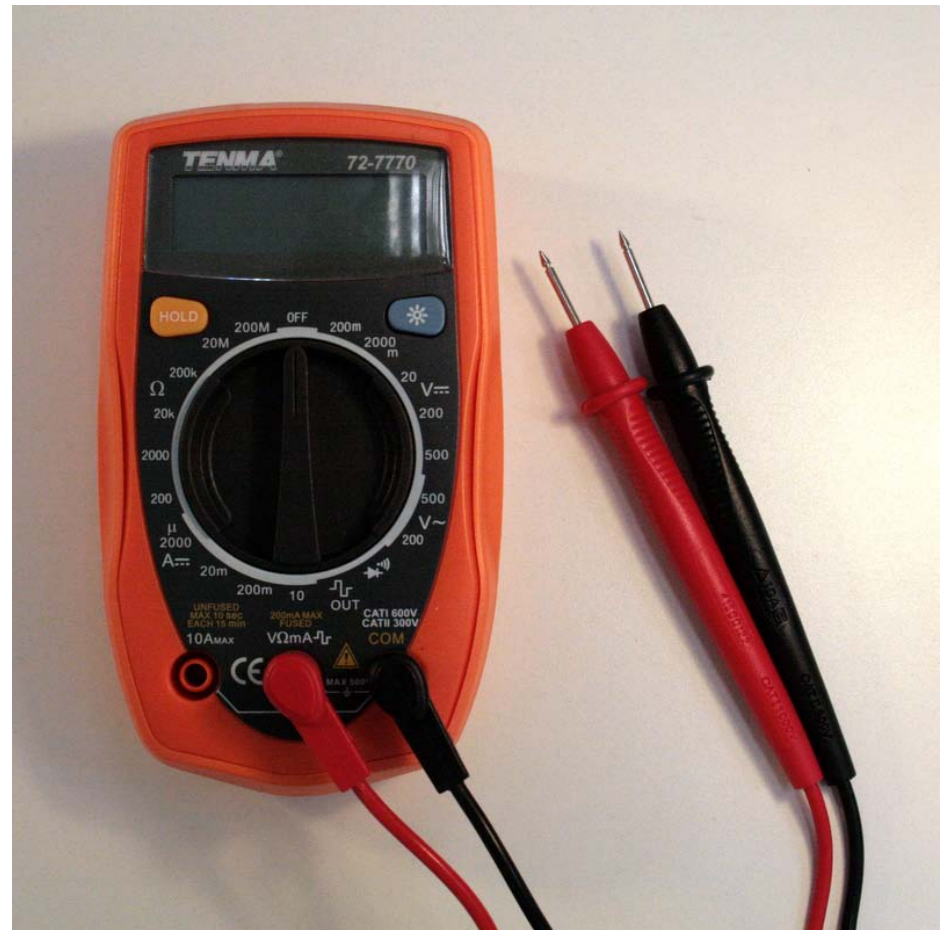


Wire Stripper

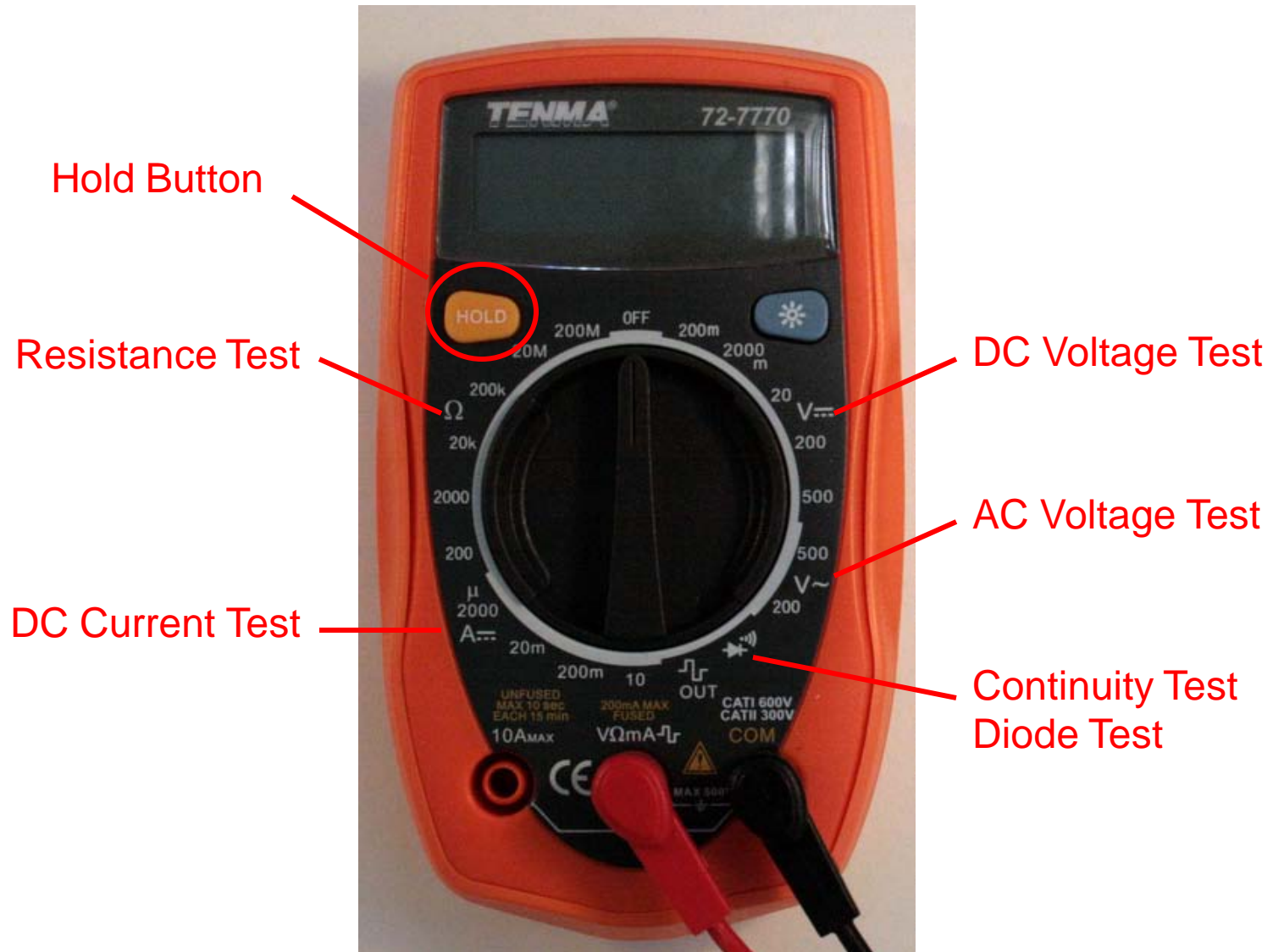


Multimeter

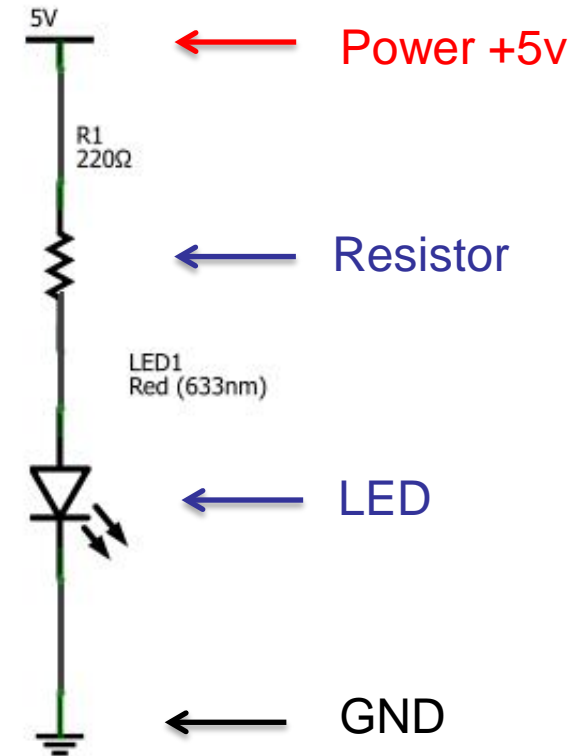
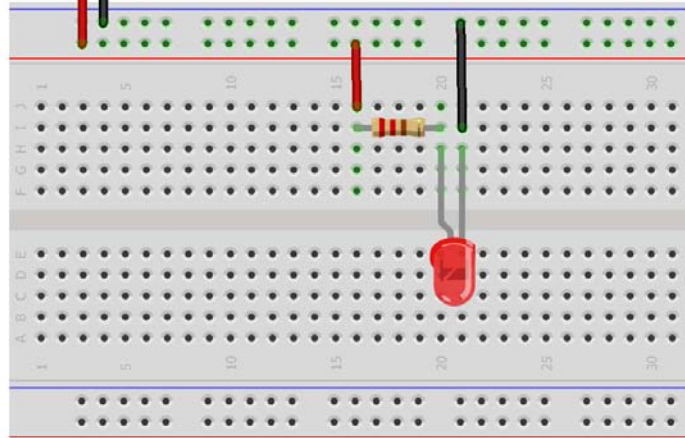
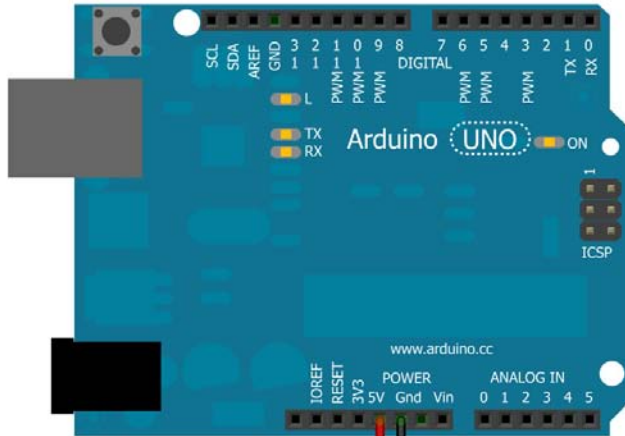
- Resistance (Ω)
- DC Voltage (V)
- AC Voltage (V)
- DC Current (A)
- Continuity Test
- Diode Test



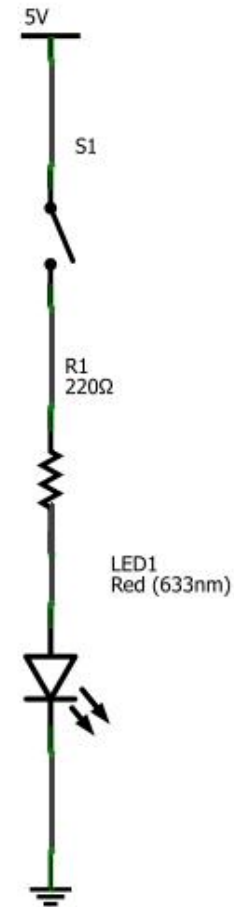
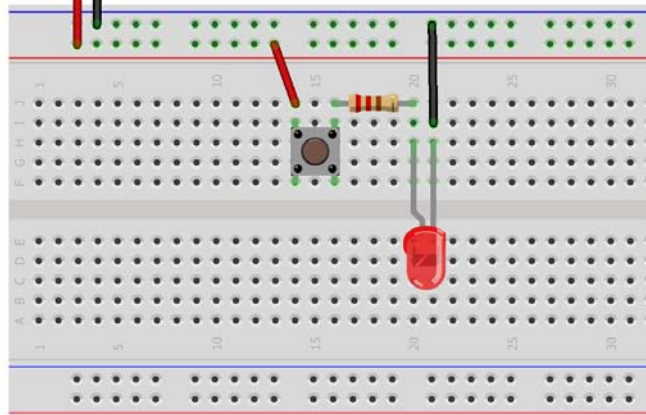
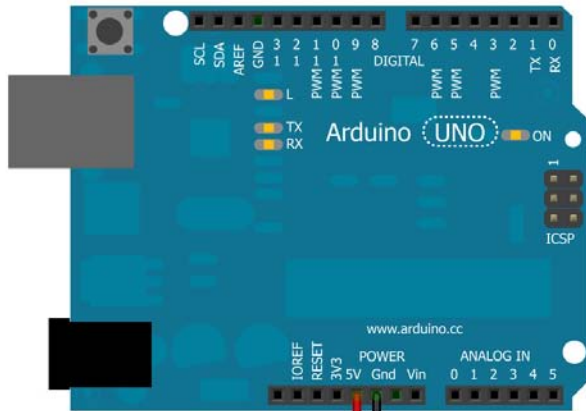
Multimeter



LED1:: My 1st LED Circuit

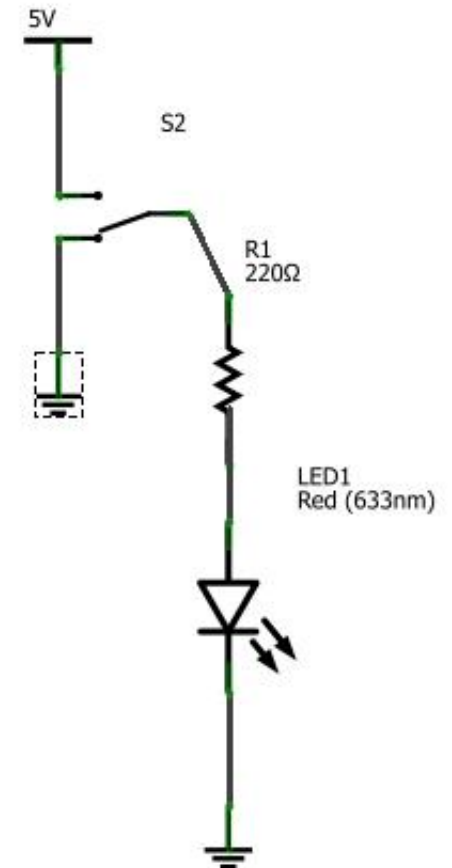
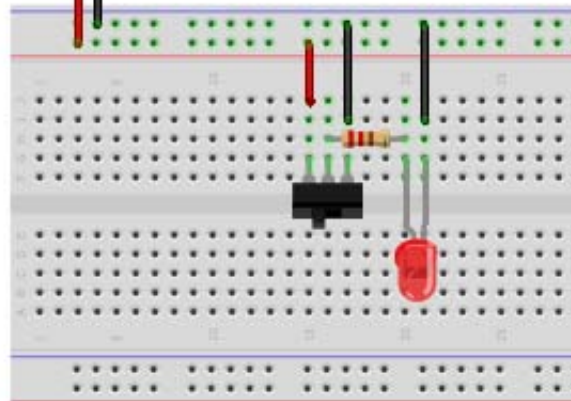
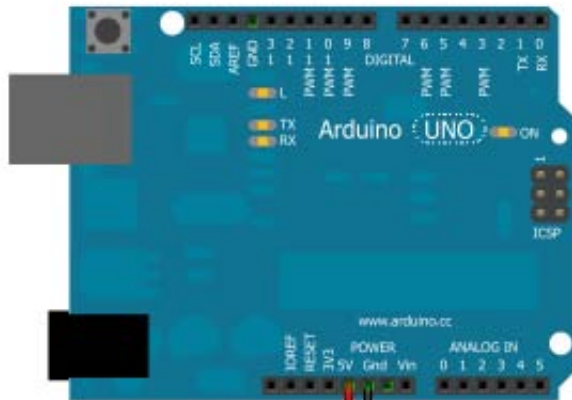
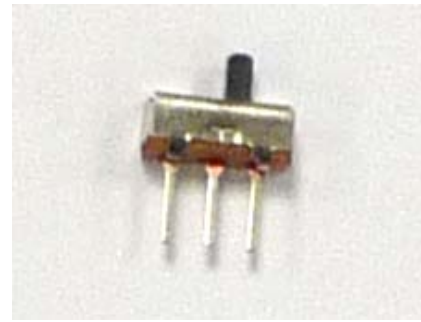


LED2: LED + push button

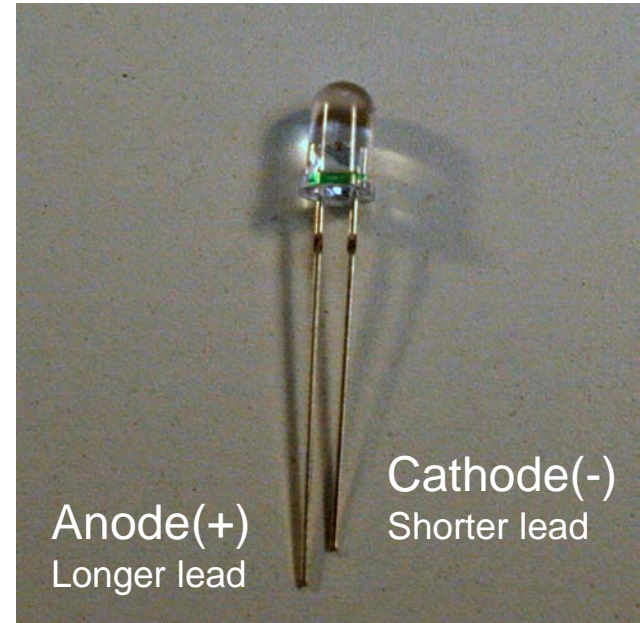
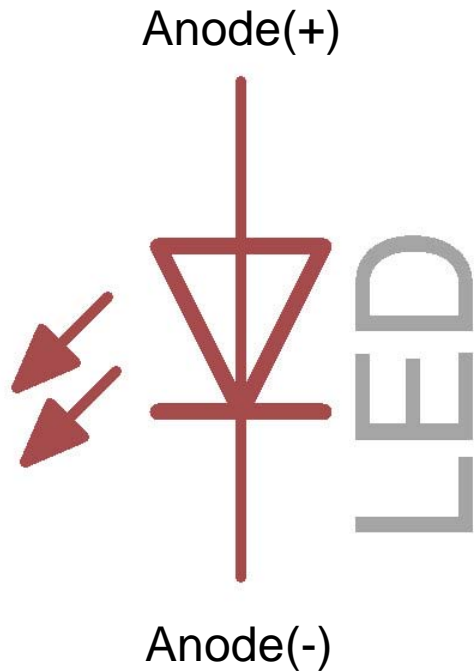


LED2-1: Switch

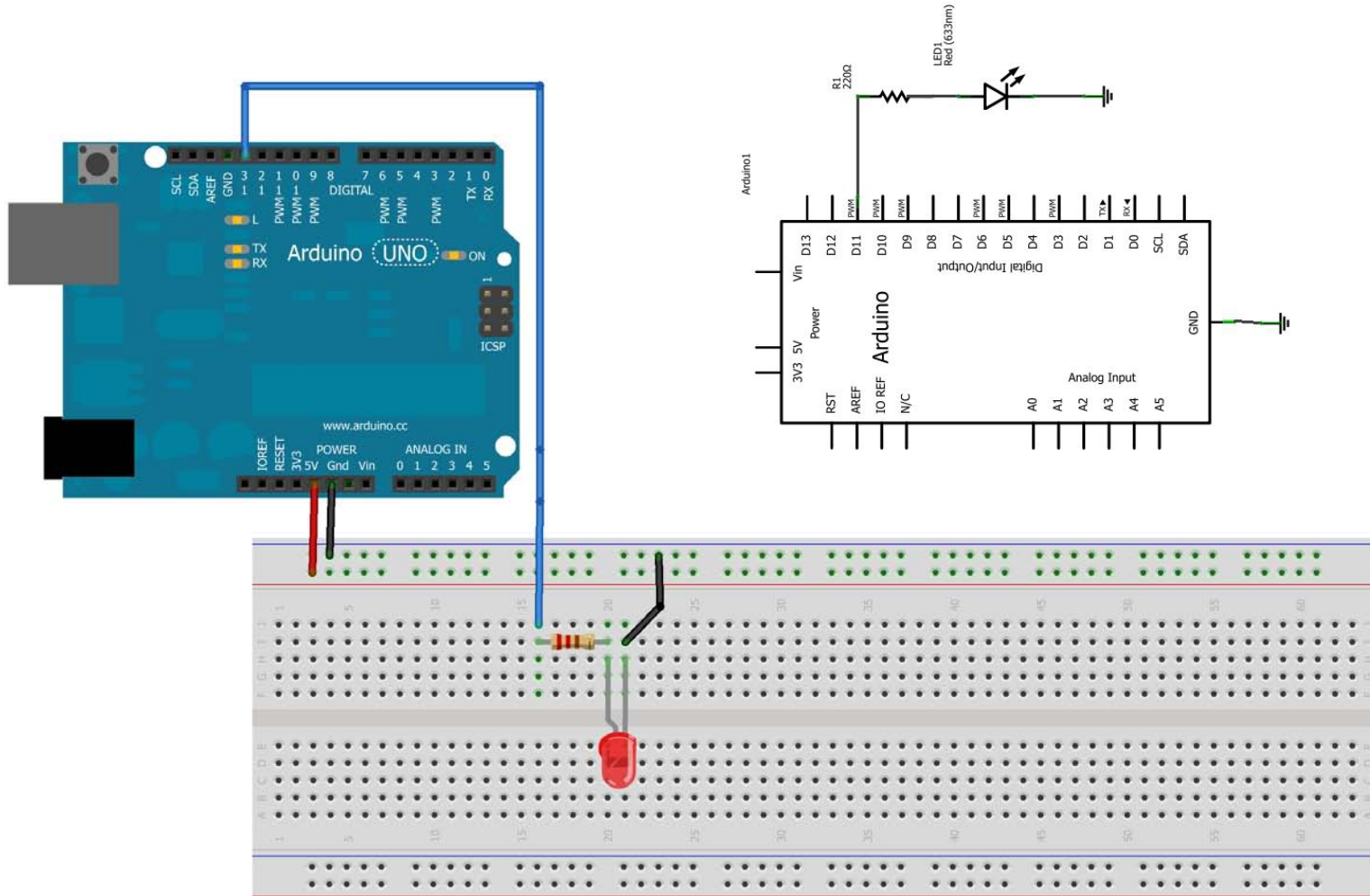
- Replace the button with a switch



LED: Light Emitting Diode

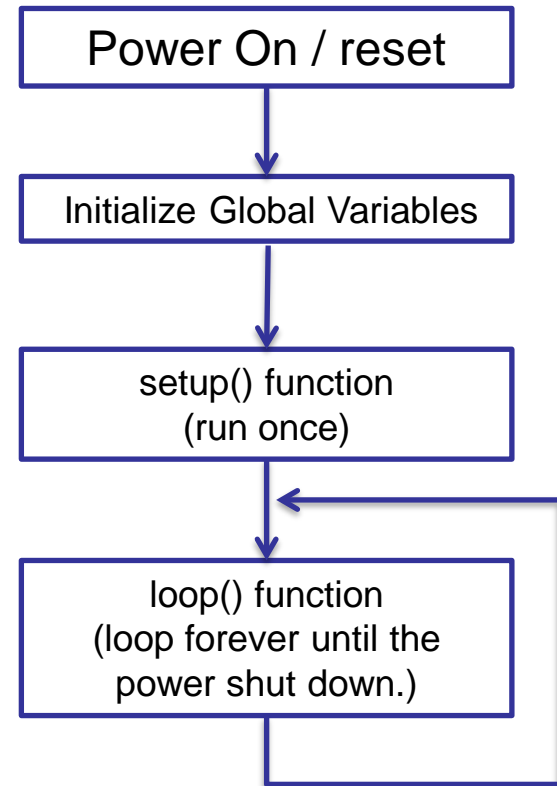


LED3: LED + Arduino Blink



Code Structure

```
/* Blink */  
int ledPin = 11; // LED connected to digital pin 11  
  
// The setup() method runs once, when the sketch starts  
void setup() {  
  // initialize the digital pin as an output:  
  pinMode(ledPin, OUTPUT);  
  
}  
  
// the loop() method runs over and over again,  
// as long as the Arduino board has a power supply.  
  
void loop()  
{  
  digitalWrite(ledPin, HIGH); // set the LED on  
  delay(500); // wait for half second(500ms)  
  digitalWrite(ledPin, LOW); // set the LED off  
  delay(500); // wait for half second(500ms)  
}
```



pinMode(pin);

Description

Configures the specified pin to behave either as an input or an output. See the description of [digital pins](#) for details.

Syntax

```
pinMode(pin, mode)
```

Parameters

pin: the number of the pin whose mode you wish to set

mode: either [INPUT](#) or [OUTPUT](#)

Returns

None

```
pinMode(pin, mode)
```

digitalWrite(pin,value)

digitalWrite(pin,HIGH);



5V

internal Pin State

Pin Output

digitalWrite(pin,LOW);



0V

```
digitalWrite(pin,value);
```

```
digitalWrite(pin,HIGH);
```



5V

internal Pin State

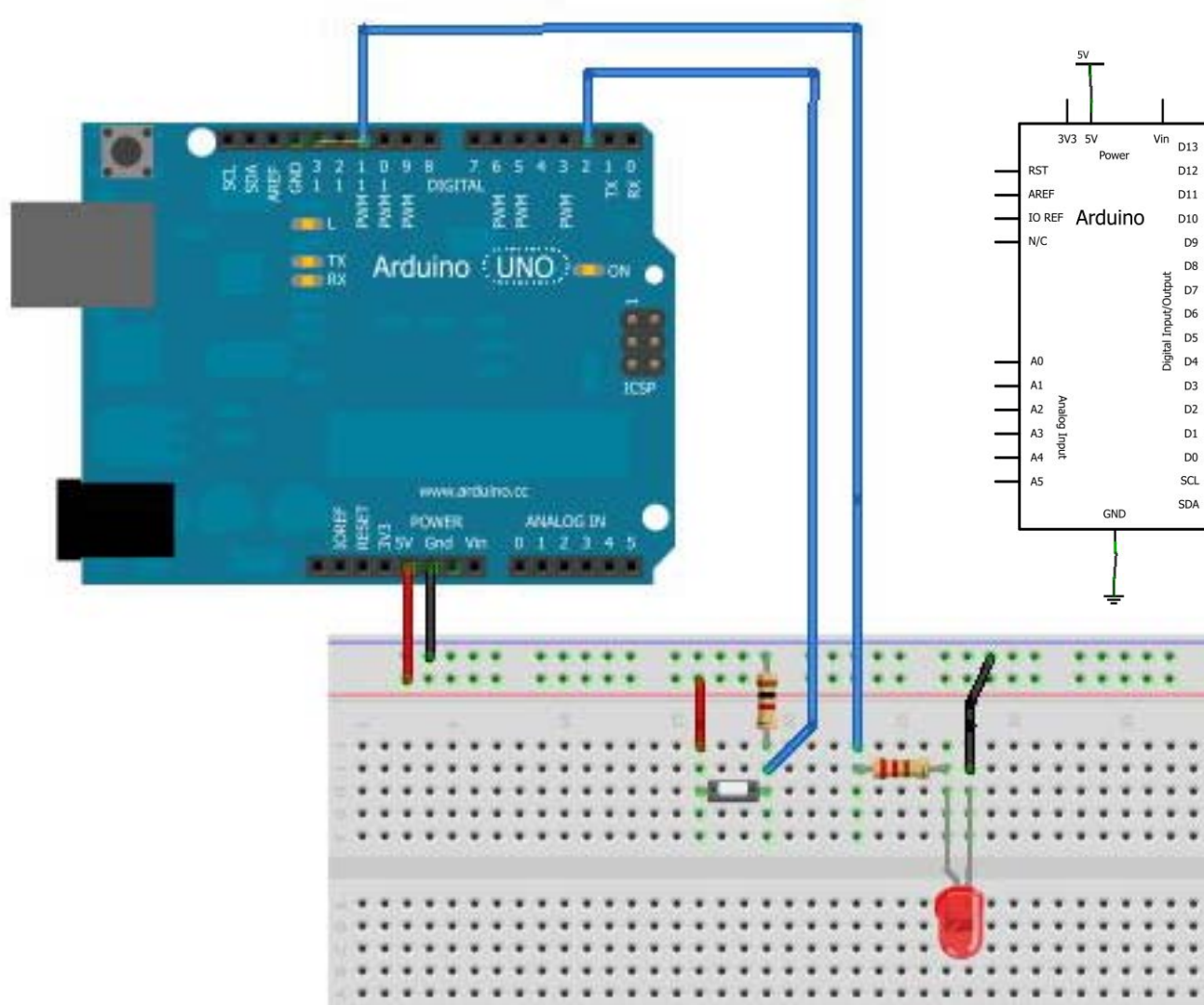
Pin Output

```
digitalWrite(pin,LOW);
```



0V

LED4: LED + Button input



Blink + Button input

```
// Pin 11 has an LED connected on most Arduino boards.
```

```
// give it a name:
```

```
int led = 11;
```

```
// digital pin 2 has a pushbutton attached to it. Give it a name:
```

```
int pushButton = 2;
```

```
int buttonState = LOW;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {
```

```
  // initialize the digital pin as an output.
```

```
  pinMode(led, OUTPUT);
```

```
  pimode(pushButton, INPUT);
```

```
}
```

```
// the loop routine runs over and over again forever:
```

```
void loop() {
```

```
  // read the input on digital pin 2:
```

```
  buttonState = digitalRead(pushButton);
```

```
  digitalWrite(led, buttonState);
```

```
  delay(100);      // wait for a while
```

```
}
```

digitalRead()

Description

Reads the value from a specified digital pin, either HIGH or LOW.

Syntax

```
digitalRead(pin)
```

Parameters

pin: the number of the digital pin you want to read
(*int*)

Returns

HIGH or LOW

Fading LED + Button

```
/*
  Blink + Button + Fade
  Press Button to Fade Up the LED, Release Button to Fade
  Out
*/

int ledPin = 9; // LED connected to digital pin 13
int buttonPin = 2; // Button connected to digital pin 2
int buttonRead = LOW; // Button state
int fadeValue = 0; // LED fading brightness

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  // initialize Serial communication port at 9600 bps:
  Serial.begin(9600);
  //Serial.println("type 1:LED On 2: LED Off");
}

void loop()
{
  buttonRead = digitalRead(buttonPin);

  if(buttonRead == LOW){ // check if button pressed
    if(fadeValue < 256) fadeValue++; //increment fadeValue

  }else if(buttonRead == HIGH){ //if button is not pressed
    if(fadeValue > 0) fadeValue--;
  }
  Serial.println(fadeValue);
  analogWrite(ledPin, fadeValue);
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```


analogWrite()

Description

Writes an analog value ([PWM wave](#)) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to **analogWrite()**, the pin will generate a steady square wave of the specified duty cycle until the next call to **analogWrite()** (or a call to **digitalRead()** or **digitalWrite()** on the same pin). The frequency of the PWM signal is approximately 490 Hz.

On most Arduino boards (those with the ATmega168 or ATmega328), this function works on pins 3, 5, 6, 9, 10, and 11. On the Arduino Mega, it works on pins 2 through 13. Older Arduino boards with an ATmega8 only support analogWrite() on pins 9, 10, and 11. You do not need to call pinMode() to set the pin as an output before calling analogWrite().

The *analogWrite* function has nothing whatsoever to do with the analog pins or the *analogRead* function.

Syntax

```
analogWrite(pin, value)
```

Parameters

pin: the pin to write to.

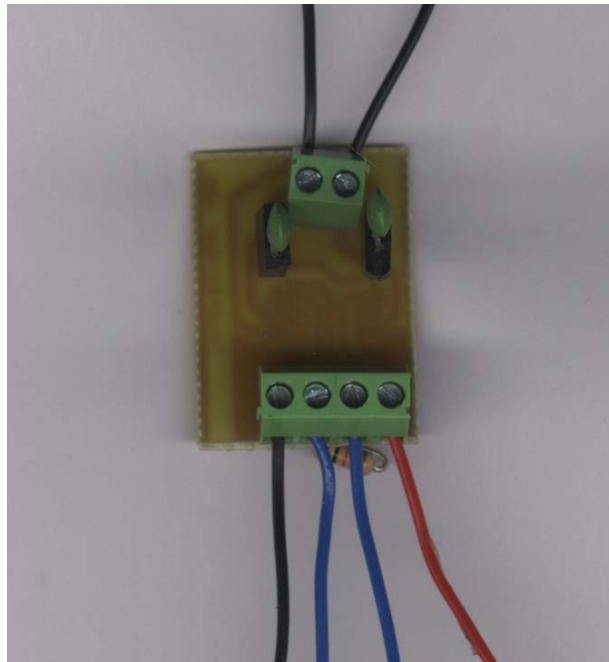
value: the duty cycle: between 0 (always off) and 255 (always on).

Returns

nothing

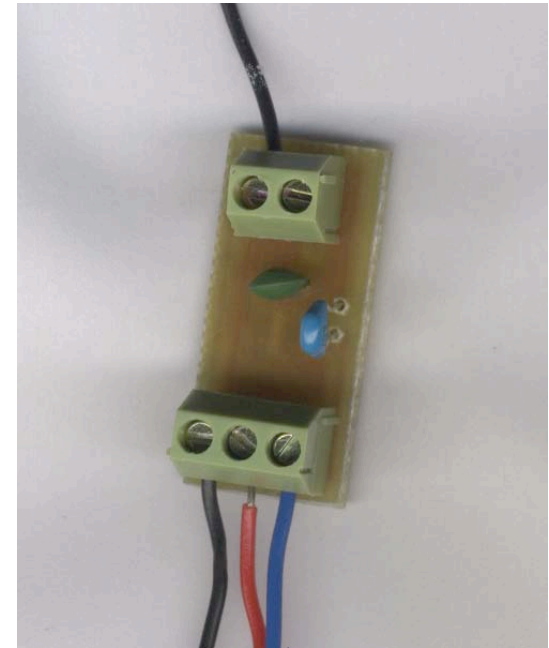
LED4: LED + Touch Sensor

2ch Touch Sensor Module



GND DPin2 5V

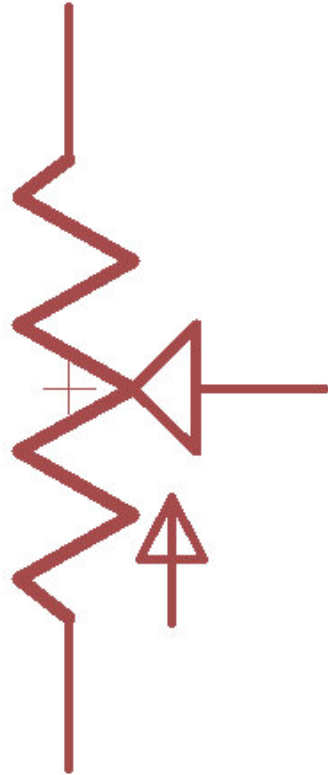
1ch Touch Sensor Module



GND 5V DPin2

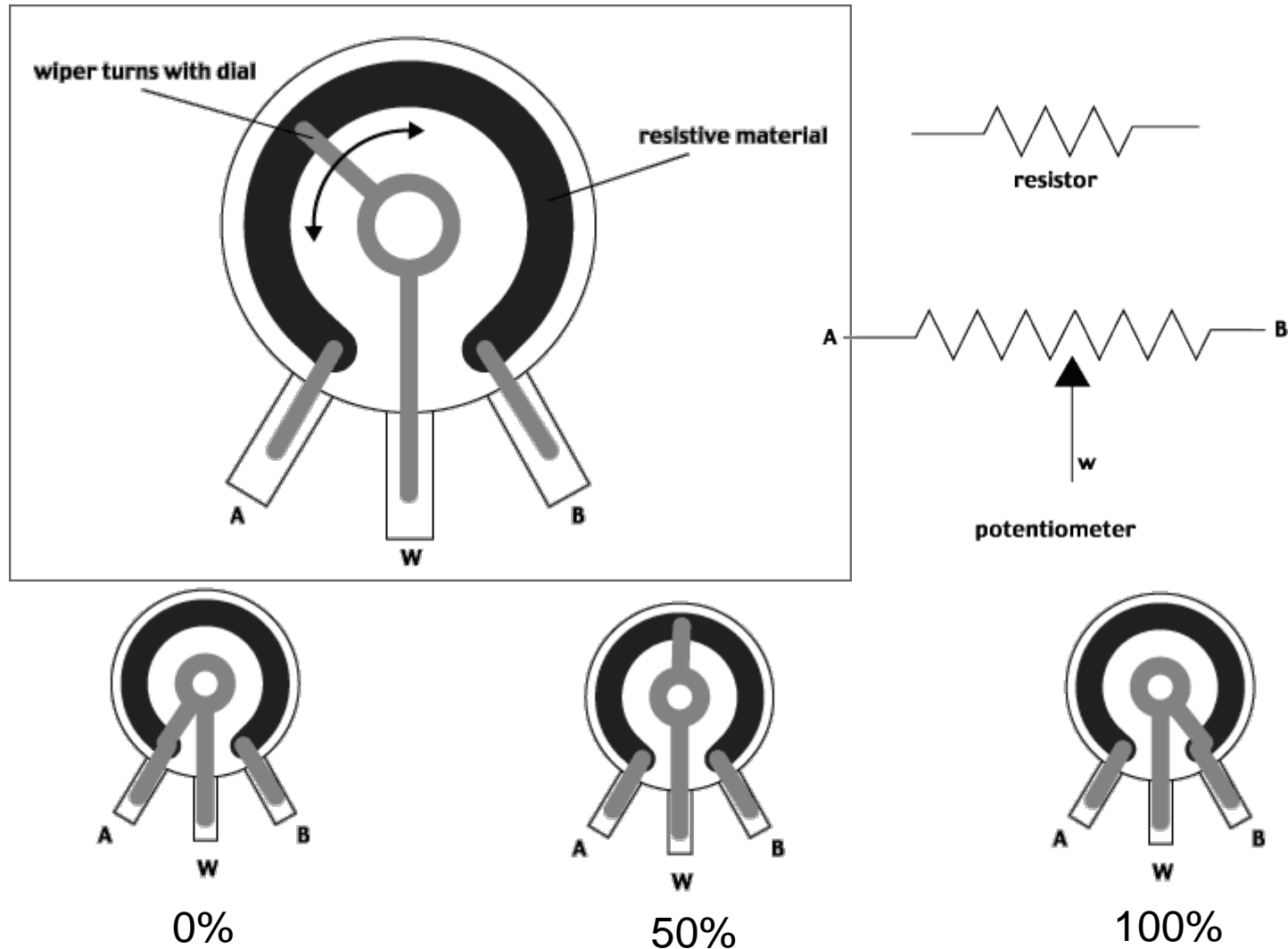
Potentiometer(Variable Resistor)

R2

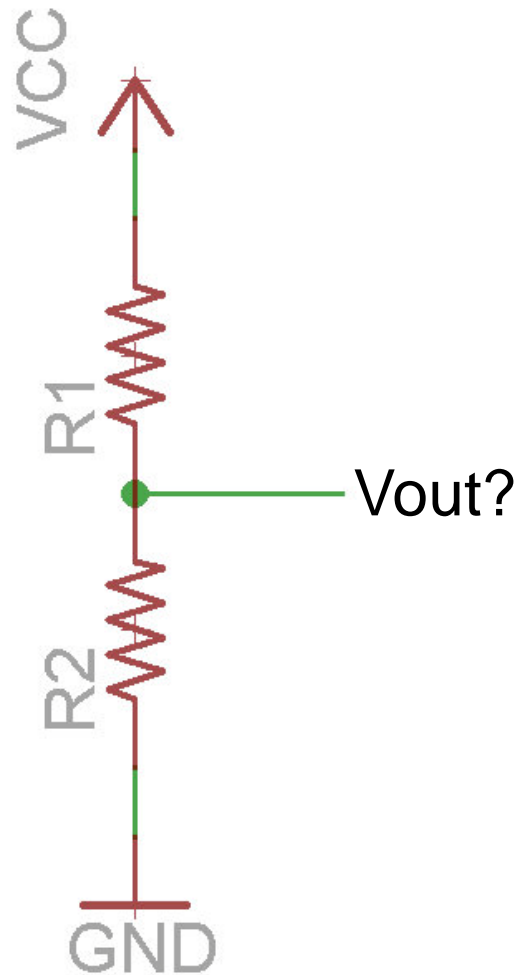


Inside Potentiometer

<http://fddrsn.net/pcomp/examples/potentiometers.html>



Voltage Divider



$V_{out} =$

$$R2 / (R1 + R2) \times V_{CC}$$

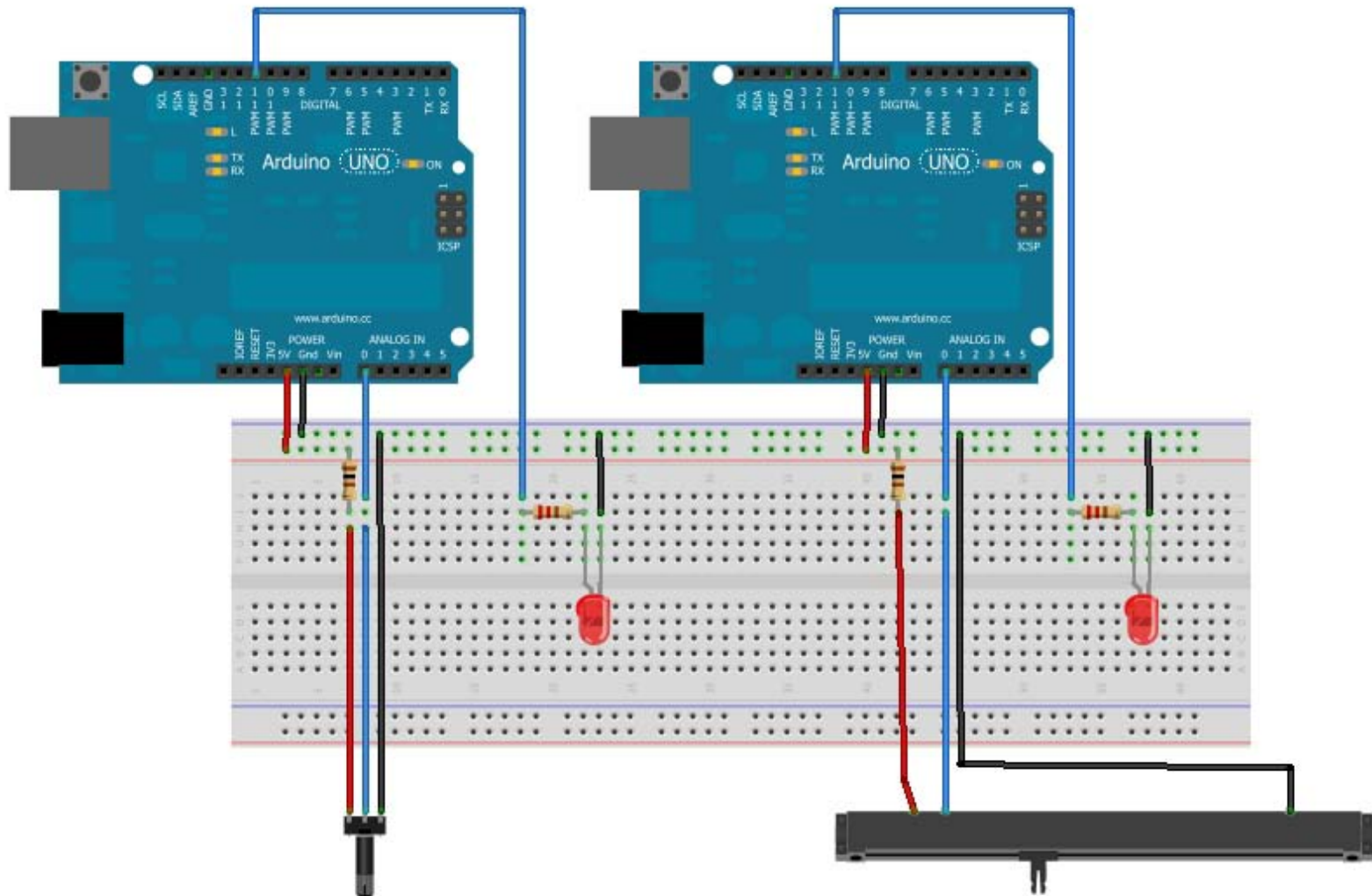
If $R1 = R2?$

$V_{out} =$

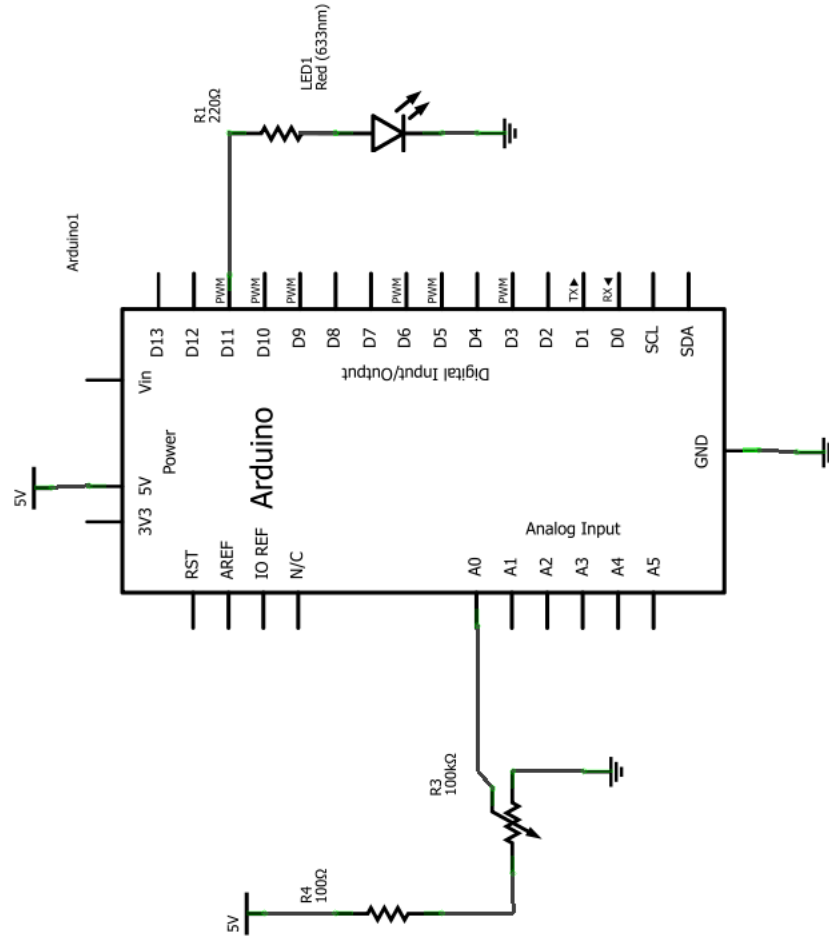
$$R / (R + R) \times V_{CC}$$

$$= 1/2 V_{CC}$$

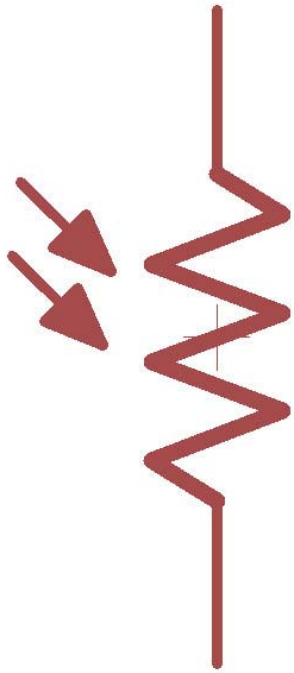
LED4 LED + VR Circuit



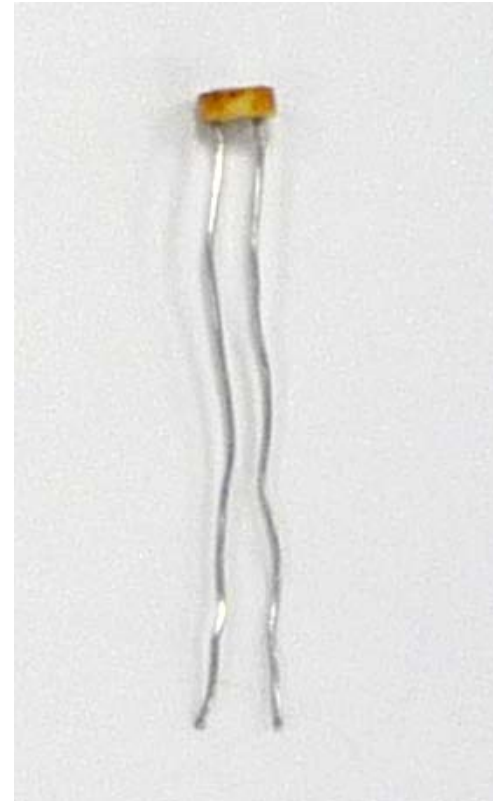
LED4 LED + VR schematic



LDR(light sensor)



LDR1



analogRead(pin)

Description

Reads the value from the specified analog pin. The Arduino board contains a 6 channel, **10-bit analog to digital converter**.

This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit.

It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.

Syntax

```
analogRead(pin)
```

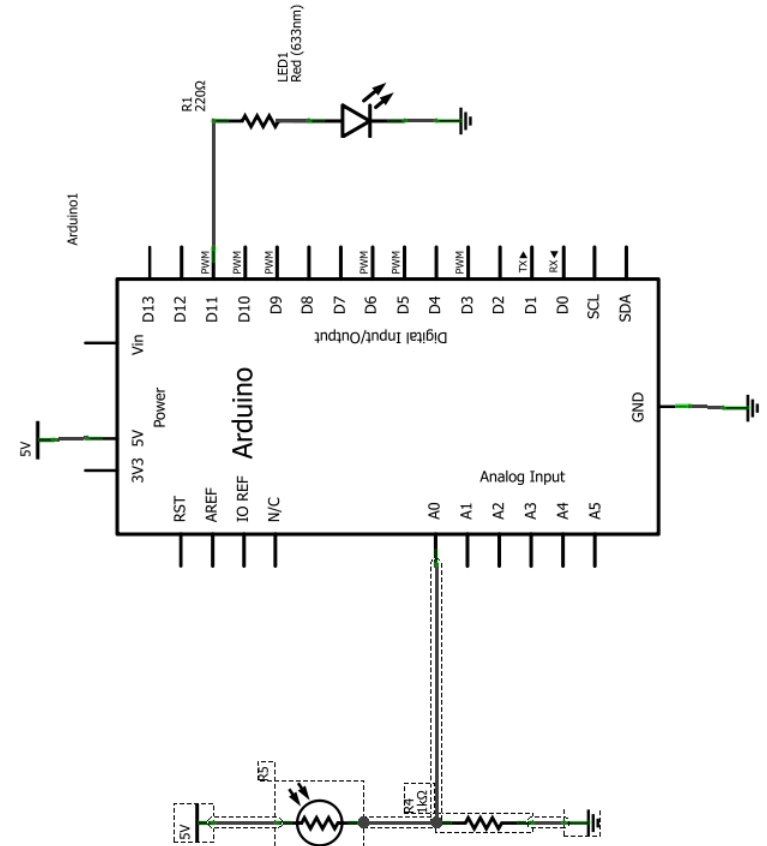
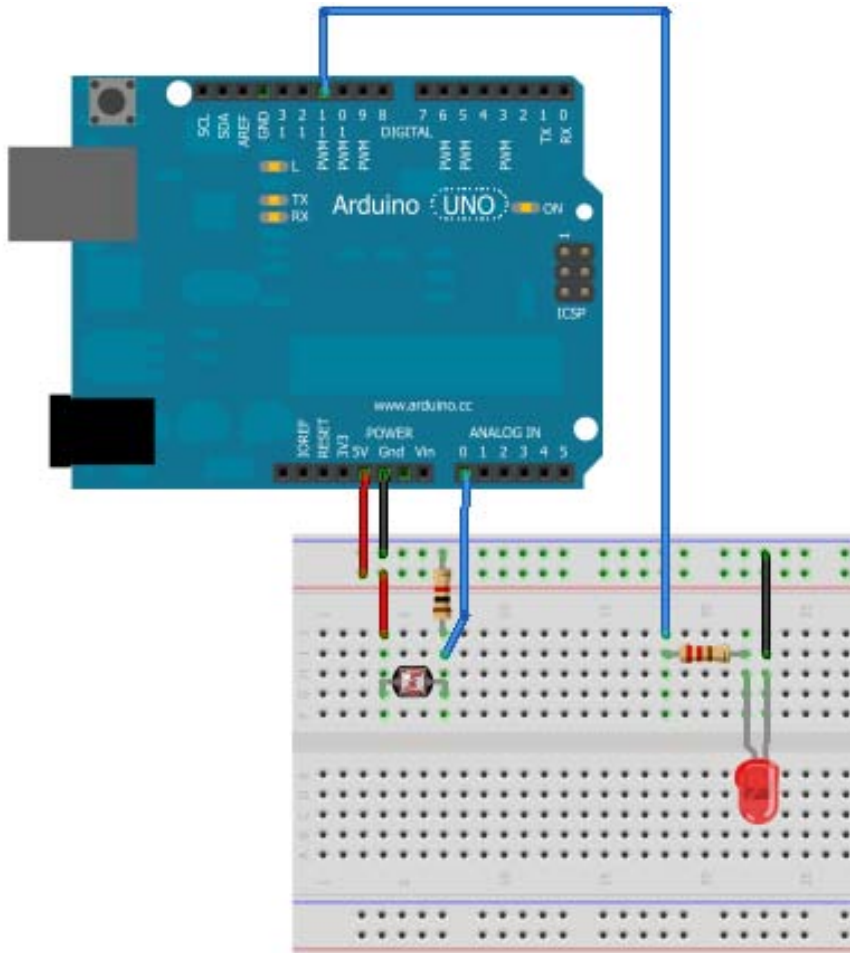
Parameters

pin: the number of the analog input pin to read from (0 to 5 on most boards, 0 to 7 on the Mini and Nano, 0 to 15 on the Mega)

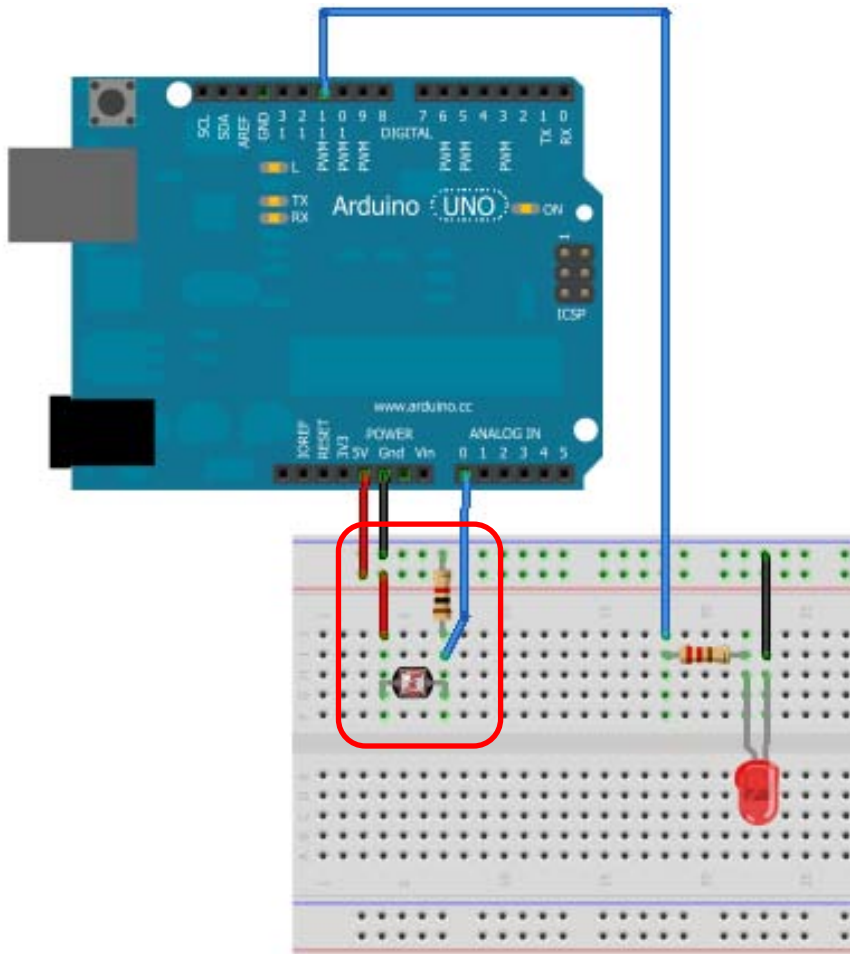
Returns

int (0 to 1023)

LightSensor(LDR) Input



LightSensor(LDR) Input



5V A0 GND

Assignment:

TOYSRUS /Sungei Thief market/

Mustafa Ctr hunting

- **Purchase a commercial electronic product (Toy/kitchen ware/etc...) which work with batteries or an AC adaptor (not 245V power-line).**



Assignment Task:

- Open up the product and observe inner-components (electrical/mechanical)
- Document the process of deconstruction and create a breakdown list of components. (and recover back to the original condition, if you can.)



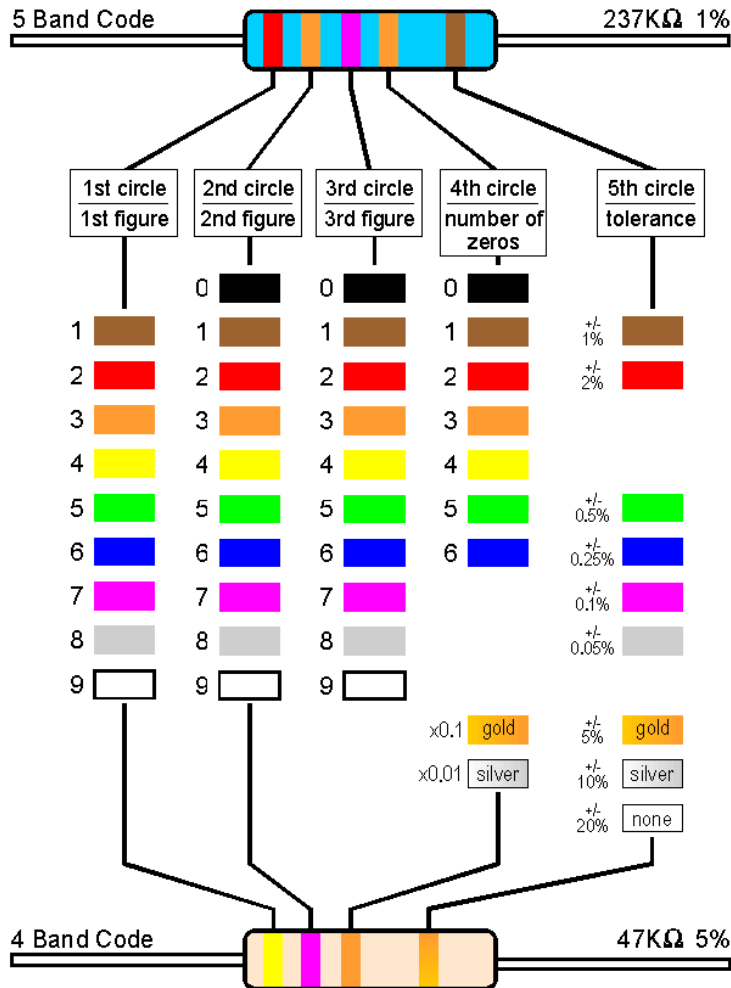
Resistors



Resistor Symbol



Resistor ColorCode



1 : Brown

2 : Red

3 : Orange

4 : Yellow

5 : Green

6 : Blue

7 : Purple

8 : Gray

9 : White

0 : Black

Tolerance

+/- 5% : Gold

+/- 10% : Silver

Reading Color Code

Brown Black Red Gold

$$1 \quad 0 \quad \times 10^2 \quad \pm 5\% = 1000 = 1\text{k Ohm}$$

Yellow Purple Orange Gold

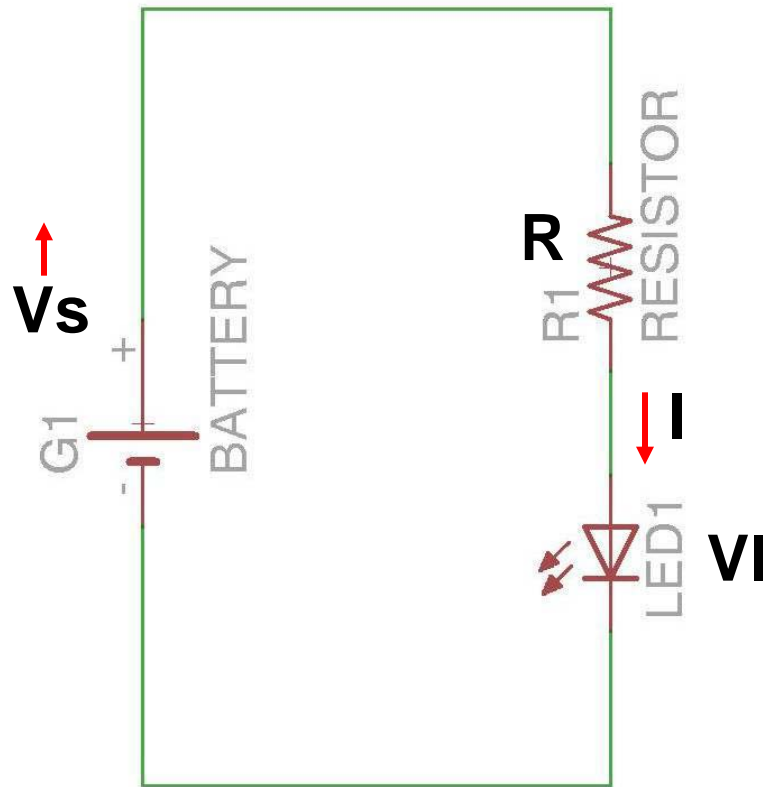
$$4 \quad 7 \quad \times 10^3 \quad \pm 5\% = 47000 = 47\text{k}$$

Brown Black Orange Gold

$$? \quad ? \quad \times 10^? \quad ? = ???$$

How to lit a LED?

= How to choose resistor value?



- $R = (V_s - V_I) / I$

What If...

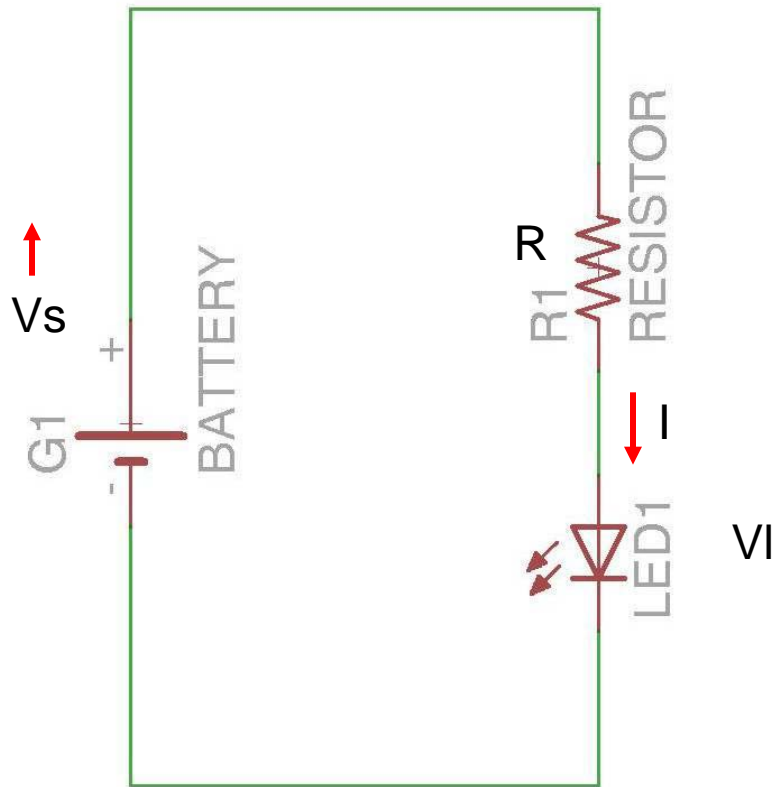
$$V_s = 9\text{v}$$

$$V_I = 1.5$$

$$I = 20\text{mA}(0.02\text{A})$$

$$R = (9 - 1.5) / 0.02$$
$$= 375 \text{ ohm}$$

How to lit a LED?



- V_s = Battery Voltage
- V_f = LED forward voltage
- R = Resistance
- I = LED Current

LED Brightness = I = LED current. Usually 10mA ~ 30mA

Ohm's Law

- $V = IR$:
 - Voltage = Current x Resistance
- $I = V / R$:
 - Current = Voltage / Resistance
- $R = V / I$:
 - Resistance = Voltage / Current