Arduino 101

AN INTRODUCTION TO ARDUINO

BY WOMEN IN ENGINEERING FT TINA AND AWESOME MENTORS

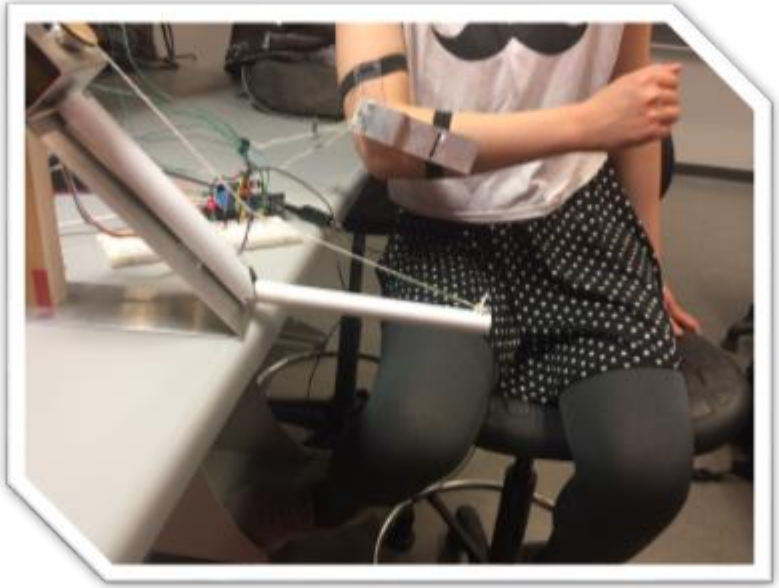
Overview

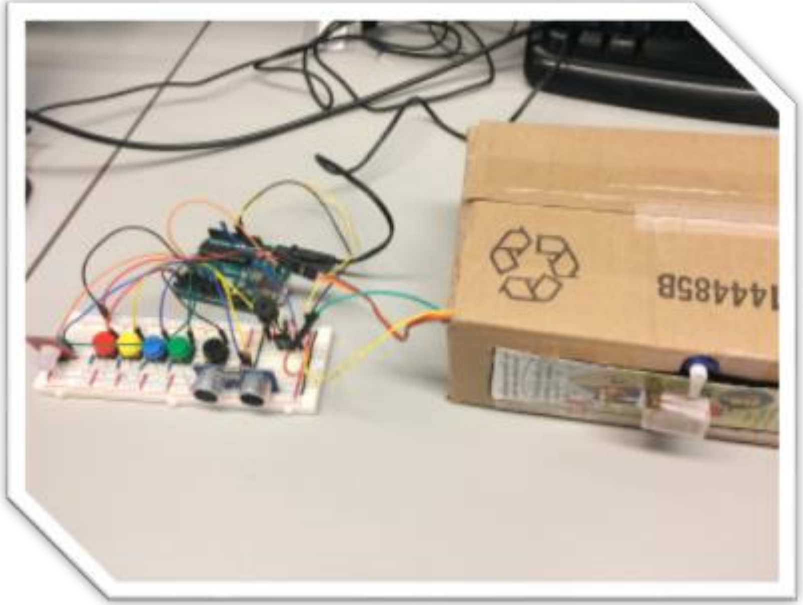
- Motivation
- Circuit Design and Arduino Architecture
- Projects
  - Blink the LED
  - Switch
  - Night Lamp
  - Servo Motor
  - Servo Motor and Potentiometer
- Additional Resources

Why Arduino?

- Easy to use
- Open source
- Inexpensive way to prototype

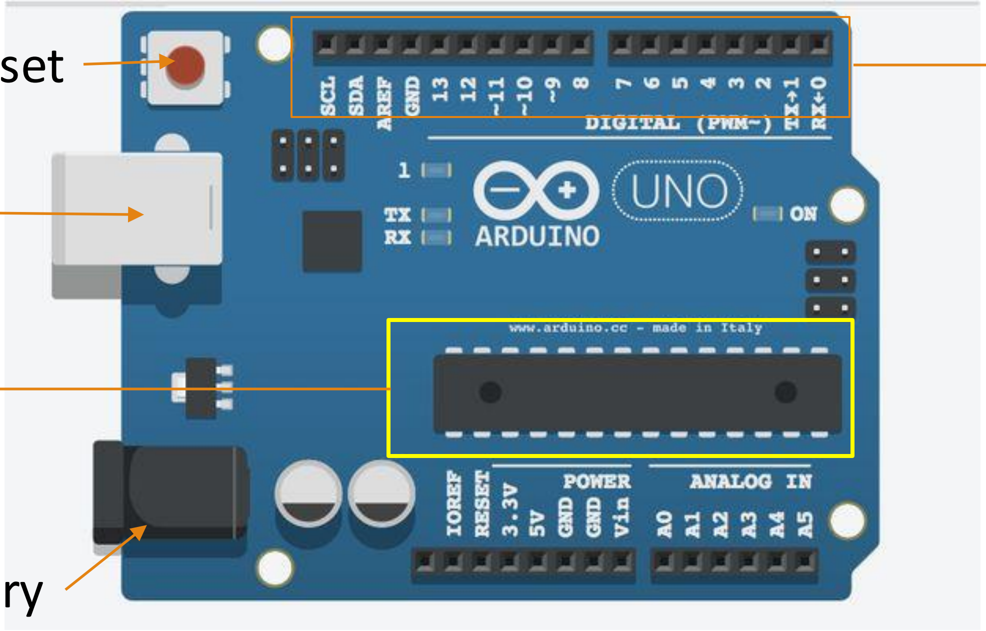
Motivation





Arduino

- Low-power **microcontroller**, which is a mini chip containing a processor, memory, and input/output (I/O) components



Reset

USB Plug

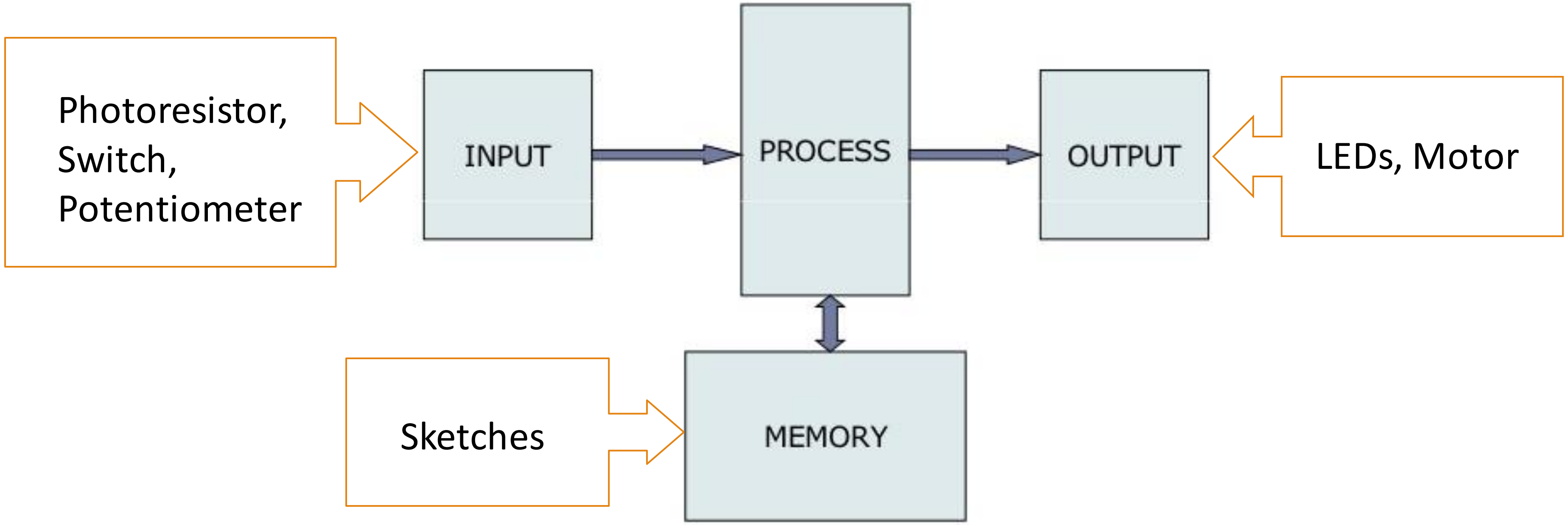
Processor /Memory

External Battery

Input/output pins

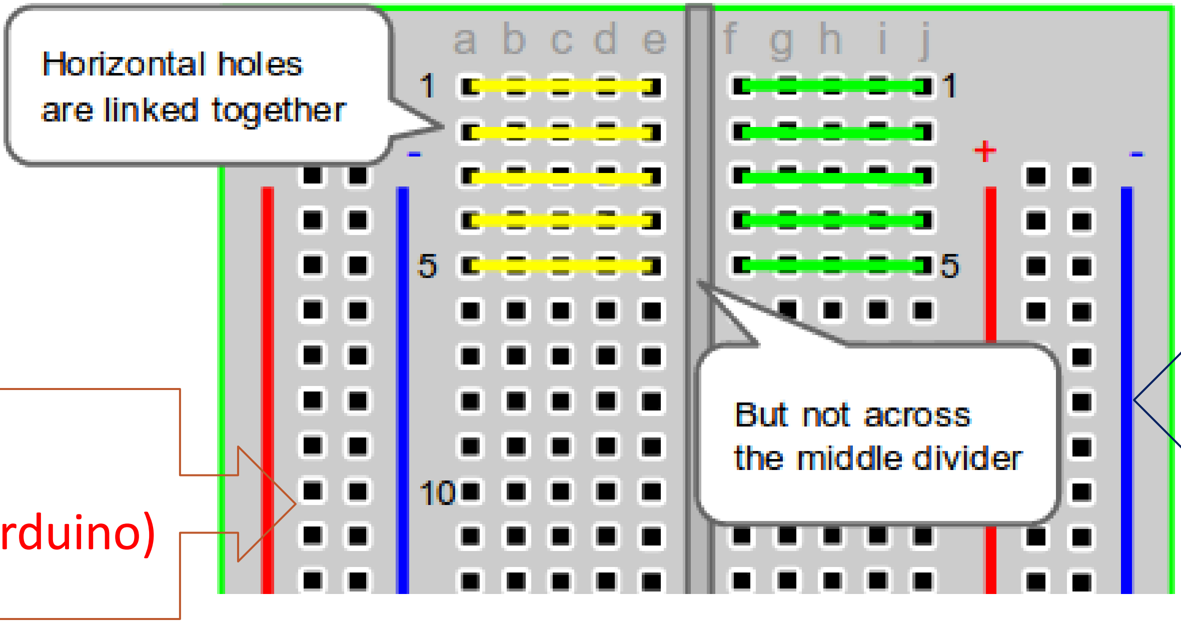
Embedded Systems

- Common place where microcontrollers are used



*in the orange boxes are the examples we will go through today

Breadboard


Horizontal holes are linked together
a b c d e
f g h i j
1
5
10
+
-
-
1
5
But not across the middle divider

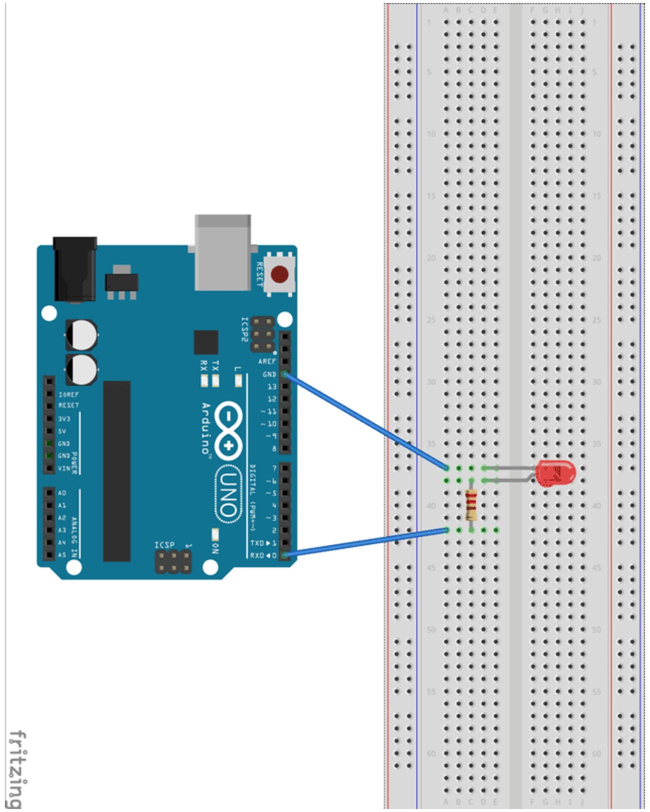
Power

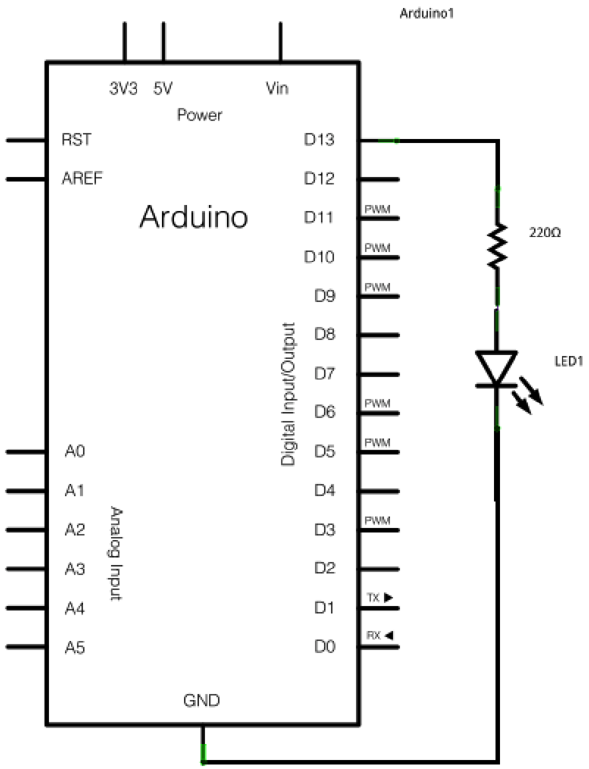
(5V on Arduino)

Ground

(GND on Arduino)

Project 1: Blink the LED (Hardware)



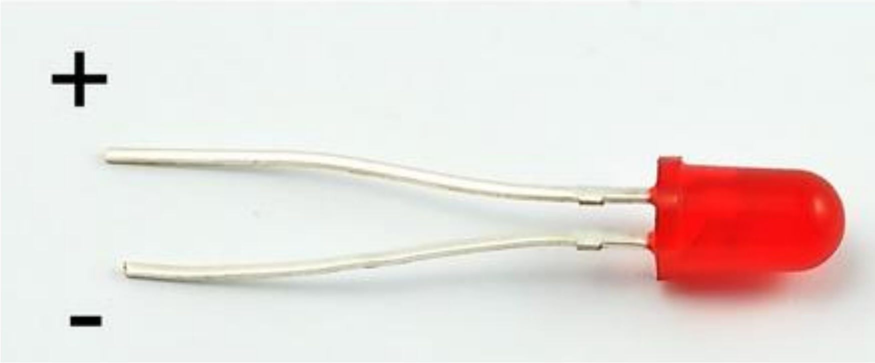


You will need:

- 2 jumper wires
- resistors
- 1 LED

Project 1: Blink the LED (Hardware)

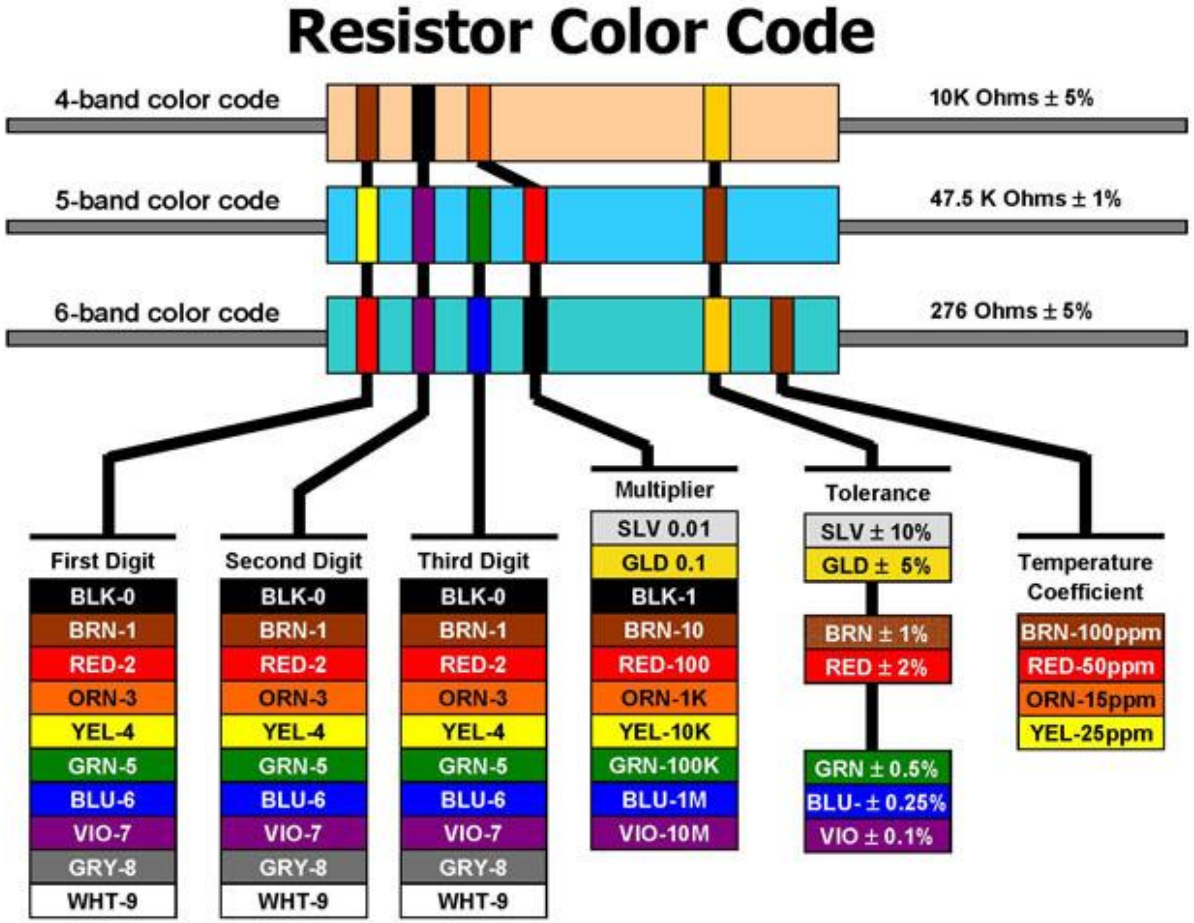
- LEDs
  - Long leg = +ve terminal
  - Short leg = -ve terminal
  - **MUST** be used with a resistor to limit the amount of current flowing through the LED, otherwise you might burn it out!



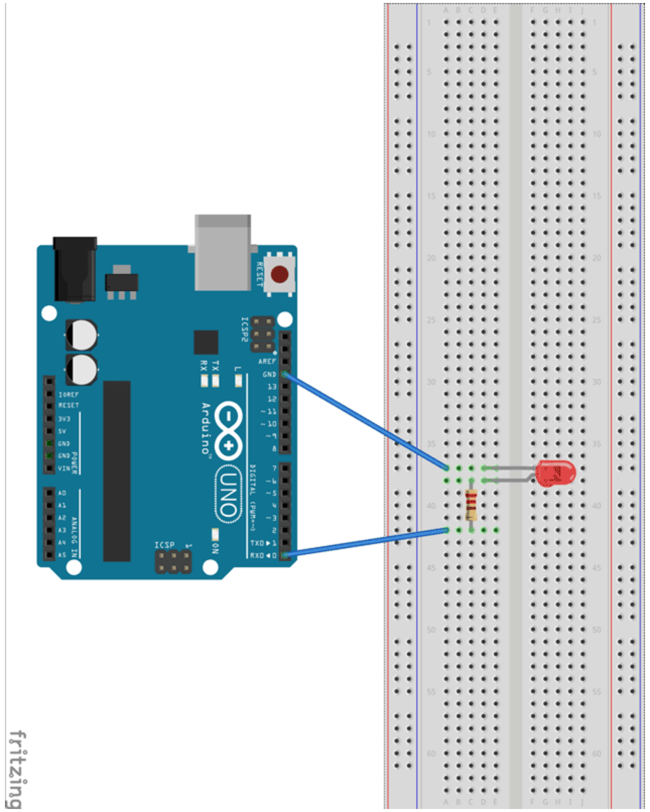
Project 1: Blink the LED (Hardware)

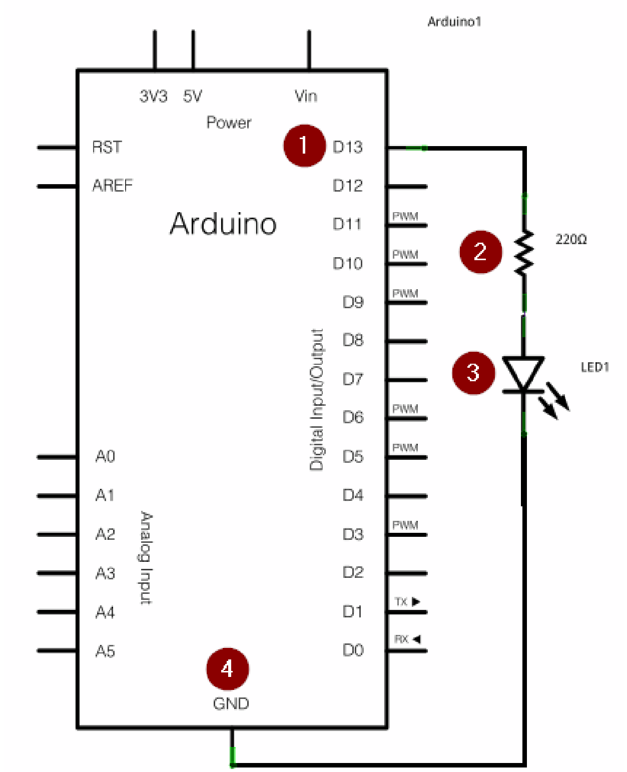
Resistor (Ω)

- Can be connected either way
- What does the colours mean?→
- Note: In this workshop we are using 4-band resistors



Project 1: Blink the LED (Hardware)





1. Choose a number between 4-13; connect a jumper wire from that slot

2. Test the circuit with these resistors
   - 150 Ω
   - 1.5 kΩ

3. Connect the +ve terminal of LED to the same row

4. Ground the circuit

Arduino IDE

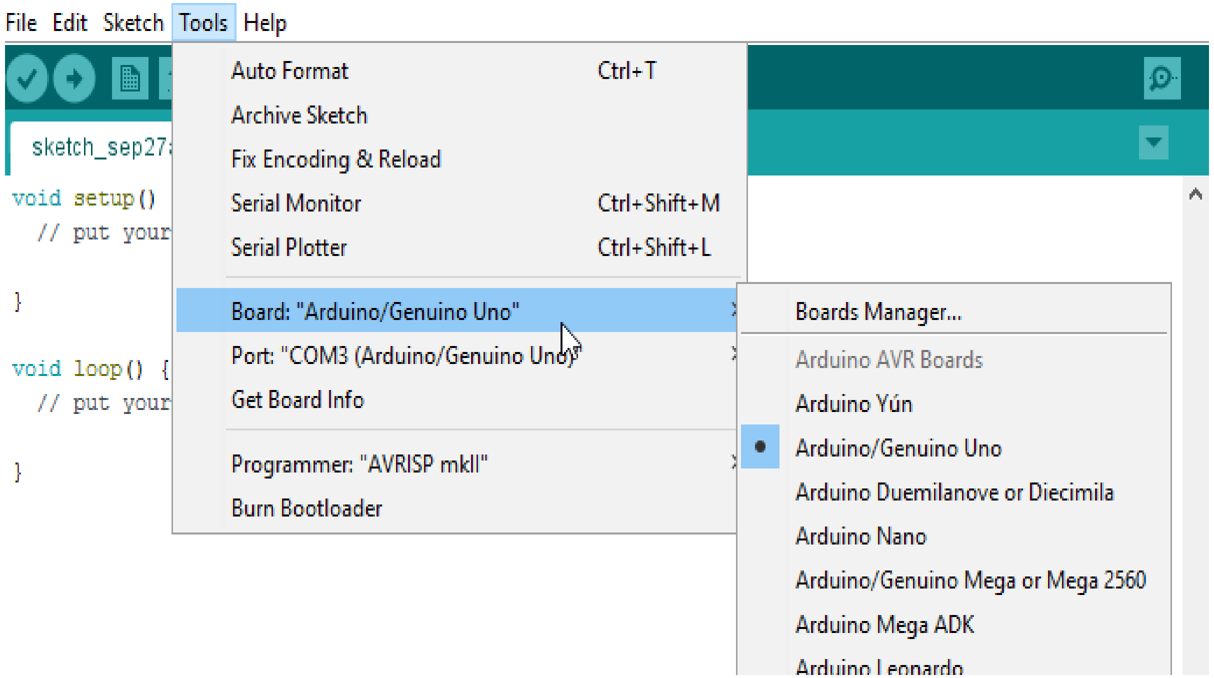
Before we start programming, open Arduino

Select the type of microcontroller:

- Tools >> Board
- Select “Arduino Uno”

Select the Serial Port in which the Microcontroller is connected to:

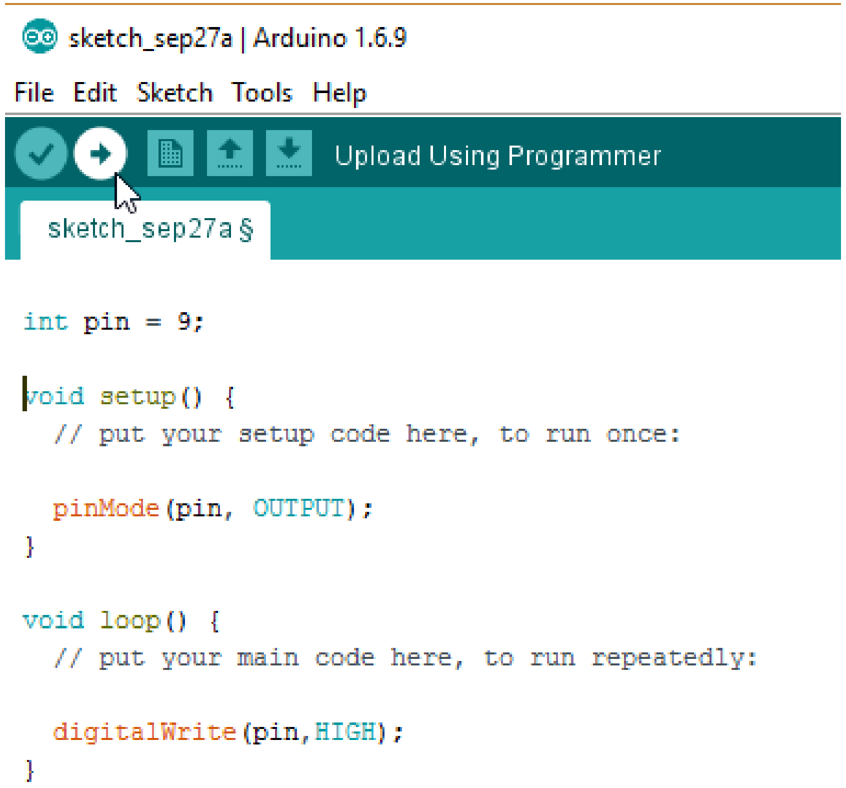
- Tools >> Serial Port
- Select the serial port which Arduino is connected to



```
void setup()
  // put your

}

void loop() {
  // put your

}
```

File Edit Sketch Tools Help
sketch_sep27
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
Board: "Arduino/Genuino Uno"
Port: "COM3 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
● Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo

Example – Turn on LED



```
int pin = 9;

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

  pinMode(pin, OUTPUT);
}

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

  digitalWrite(pin,HIGH);
}
```

Three parts to this example:

1. Global variables: declaration and initialization
   - int for integer
   - boolean (true/false)
   - string
2. setup()
   - Called when the sketch starts; only executed once
   - Attach I/O pins
   - (Sometimes) initialize timer, etc
3. loop()
   - Repeats infinitely as long as the board is powered and memory is available

Note: Single line comment = //; block comment = /* */

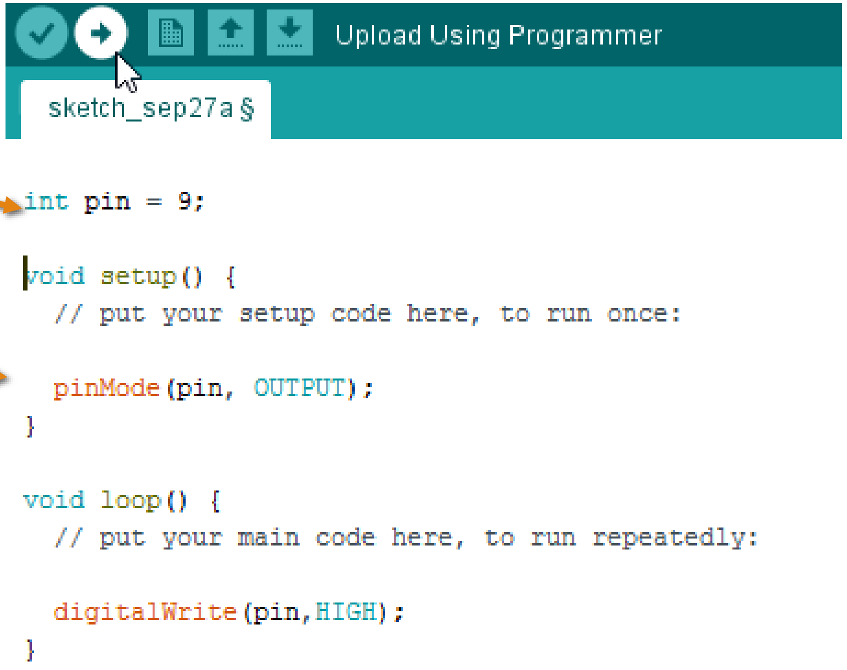
Project 1: Blink the LED (Software)

1. **Before setup()**
   - Declare an integer variable to store the pin connected to LED.
   - Syntax: int variableName = yourNumber;

2. **setup():**
   - initialize the pin as an output pin
   - Syntax: pinMode(variableName, OUTPUT);

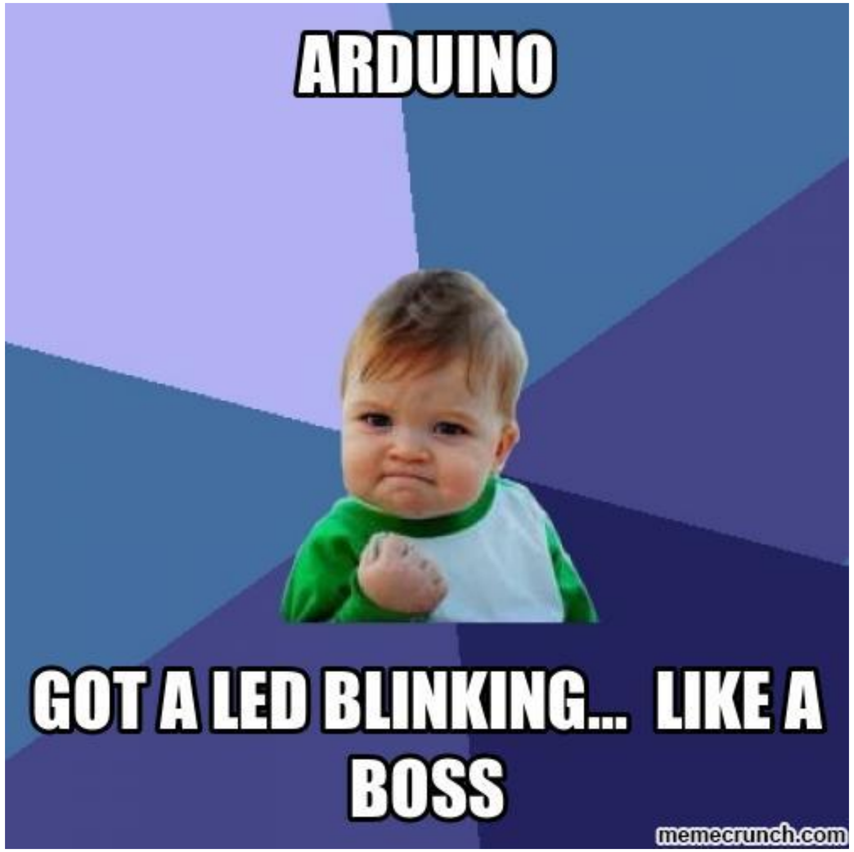
3. **loop():**
   - delay(time); //where time is in milliseconds
   - digitalWrite(variableName,STATE); //STATE = HIGH (5V) or LOW (0V)
   - HIGH = LED on, LOW = LED off

4. Once you are done, press the **arrow button to upload** your code on Arduino!



```
int pin = 9;

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

  pinMode(pin, OUTPUT);
}

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

  digitalWrite(pin,HIGH);
}
```

Break



Project 2: Digital Inputs

Hardware

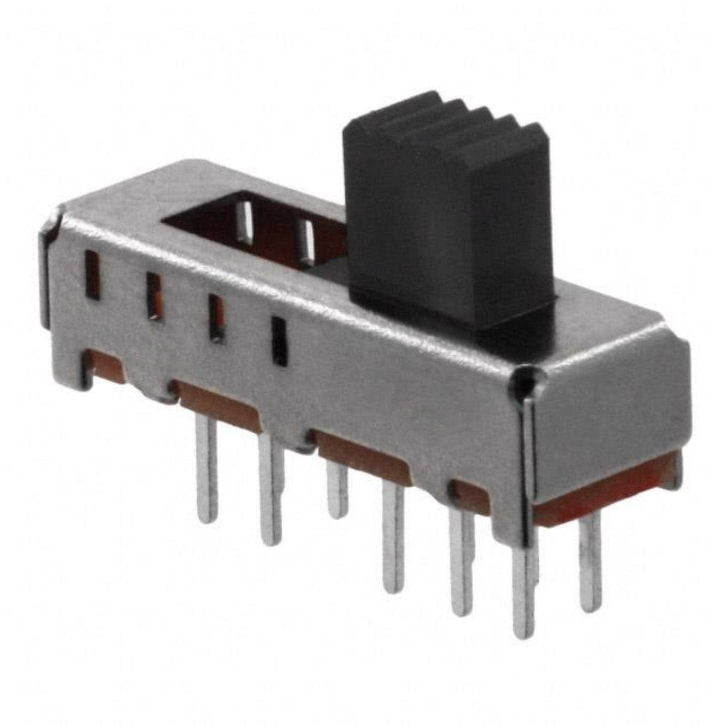
- Components: Red LED, four wires, 150 Ω, and a dip switch

Setup

1. Connect one end of the switch to power, and the other end to ground
2. Red LED and the 150 Ω resistor are in series (recall long and short legs)
3. Long leg of red LED in series with resistor is connected to pin 5, and the short leg of the LED is connected to ground

Project 2: Digital Inputs

- Switches
  - Completes the circuit when flip to ON

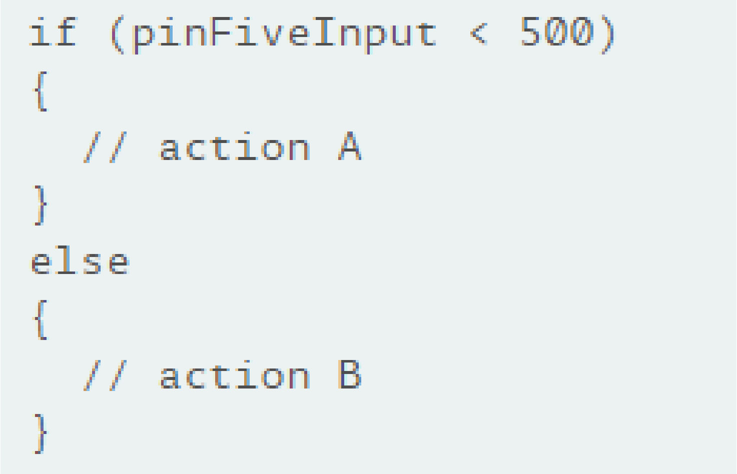


Project 2: Digital Inputs

- Before we go into the software, switch the connector from “5V” to pin 7
- **Setup():**
  - Initialize the LED pin
  - Initialize pinMode of the switch to INPUT

Project 2: Digital Inputs if and else

- Actively checking if the condition specified in () is met
- If there are 2+ conditions that are related, you can use
  - If (condition A) { }
  - else if (condition B) {} … // can have many else if
  - else {}
- Notice that else does NOT have a condition statement



```
if (pinFiveInput < 500)
{
  // action A
}
else
{
  // action B
}
```

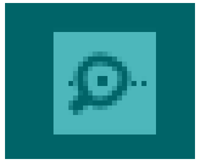
Project 2: Digital Inputs

- **loop():**
  - digitalRead(pin#);
    - Read input pin# and see if it is HIGH or LOW
  - digitalWrite(pin# , STATE)
    - write STATE, which is either HIGH or LOW, to output pin#

  - Step 2:
    - A) if button is LOW, turn on LED
    - B) otherwise, if the button is HIGH turn off LED
    - Repeat A to B

Project 2: Digital Inputs ft Serial Monitor

- Serial communication to let you know when a button is pressed
  - In setup:
    - **Serial.begin(9600);**
  - In loop:
    - **Serial.println(“Pressed Button”);**
      - //when the button is pressed write this to the serial monitor
  - After uploading your code unto the Arduino
  - Click on the Serial Monitor button (Top right side of the IDE)

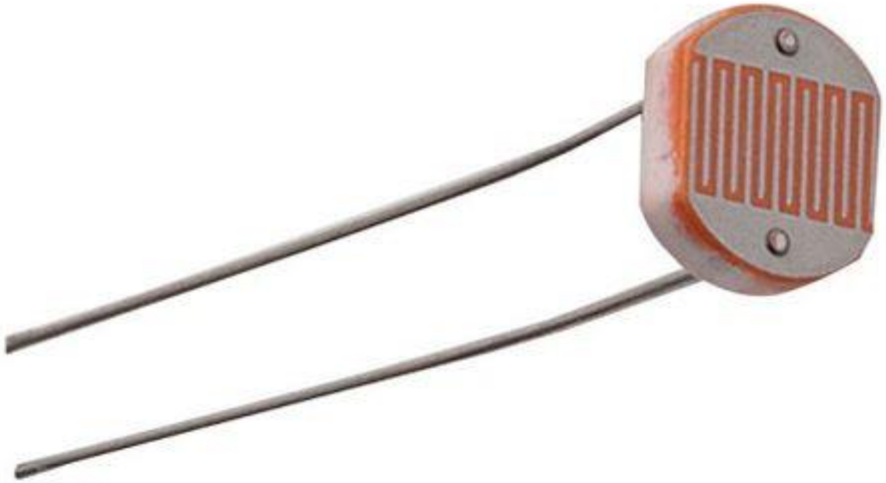


Project 2: Digital Inputs Pseudo Code

- **Before setup()**
  - Initialize the LED and switch pins
- **In setup()**
  - Attach pinMode
    - LED = OUTPUT
    - switch = INPUT
- **In loop()**
  - If switch is ON
    - LED = ON
  - else
    - LED = OFF

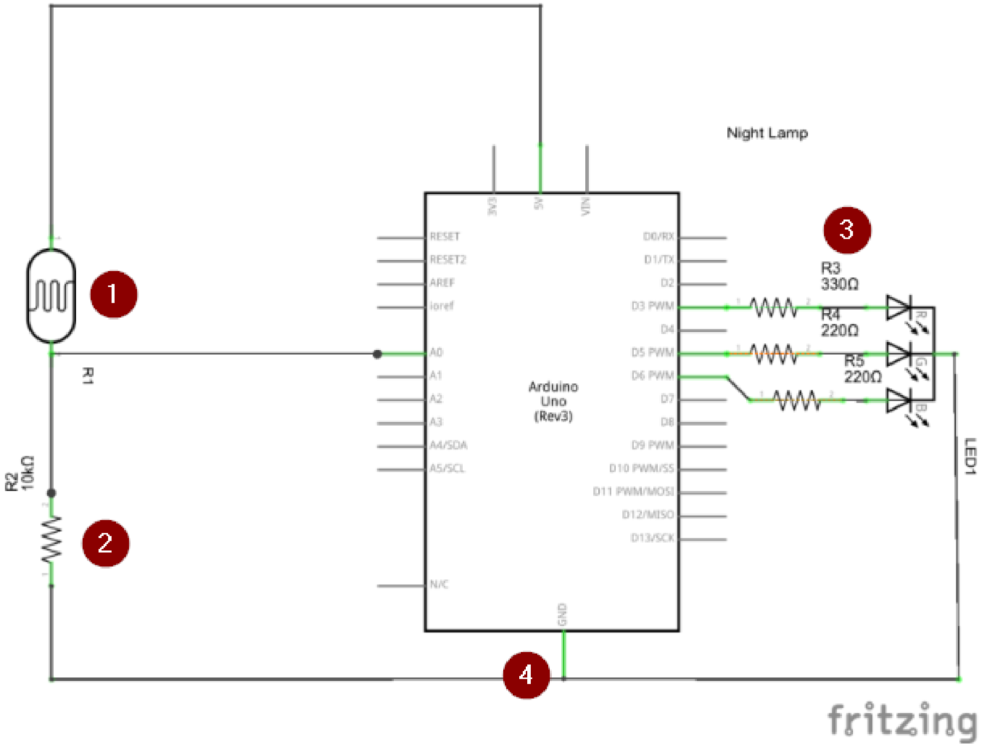
Project 3: Put it all together... and more!

- Photoresistor
  - Variable resistor that changes resistance based on light intensity
  - Direction doesn't matter
  - We can use Serial Monitor to check the ambient lighting condition!
    - This will help to pick the thresholds for our project!!



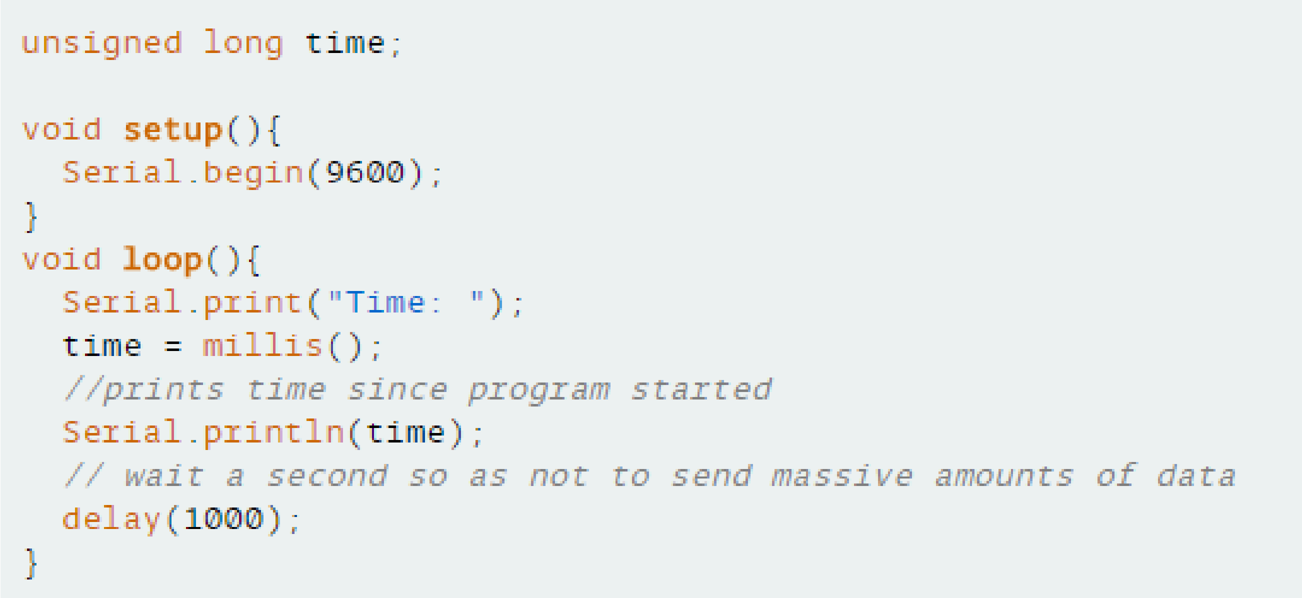
Project 3: Circuit

1. Connect Photoresistor to A0 and the other end to 5V
2. Connect a 10k Ω resistor from the rail from 1. to ground
3. Connect 3 pins of your choice to 3 LEDs of different colours
4. Connect the LEDs to ground



Project 3: Timer millis()

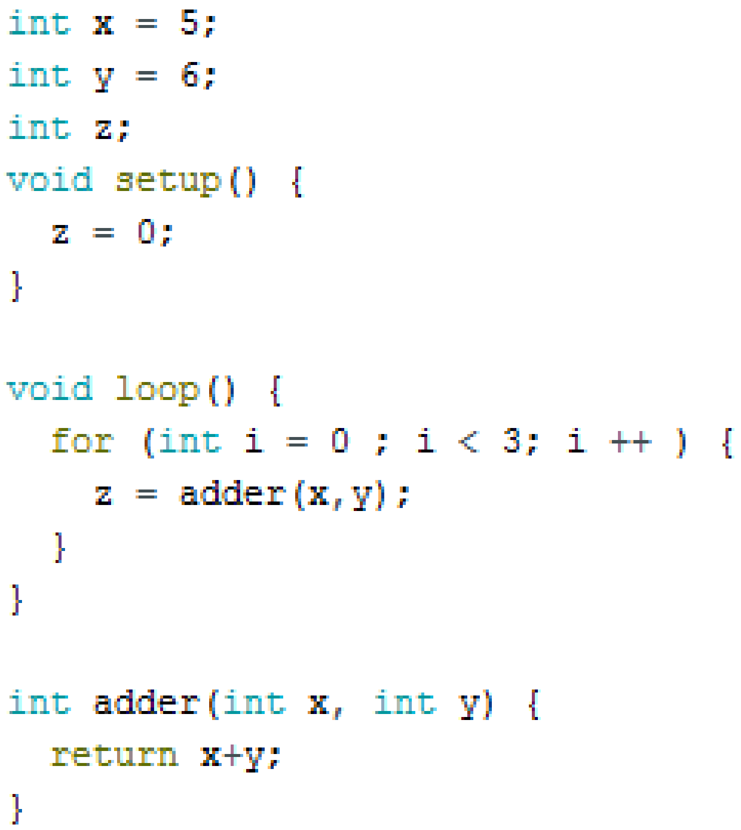
- Returns the time since the program started running



```
unsigned long time;

void setup(){
  Serial.begin(9600);
}
void loop(){
  Serial.print("Time: ");
  time = millis();
  //prints time since program started
  Serial.println(time);
  // wait a second so as not to send massive amounts of data
  delay(1000);
}
```

Project 3: Write your own function

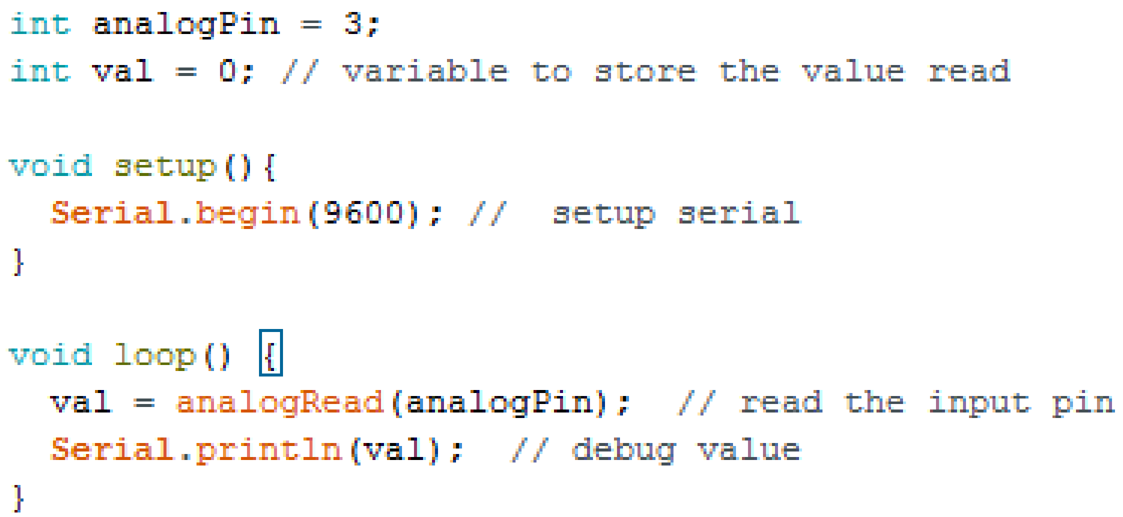
- Extract code from the main loop to keep it clean
- Avoid repeating the same line(s)
- 2 types of functions:
  - void: don’t return anything e.g. turnLEDOn()
  - int: function that returns an integer value e.g.: adder()



```
int x = 5;
int y = 6;
int z;
void setup() {
  z = 0;
}

void loop() {
  for (int i = 0 ; i < 3; i ++ ) {
    z = adder(x,y);
  }
}

int adder(int x, int y) {
  return x+y;
}
```

Project 3: Analog input/output

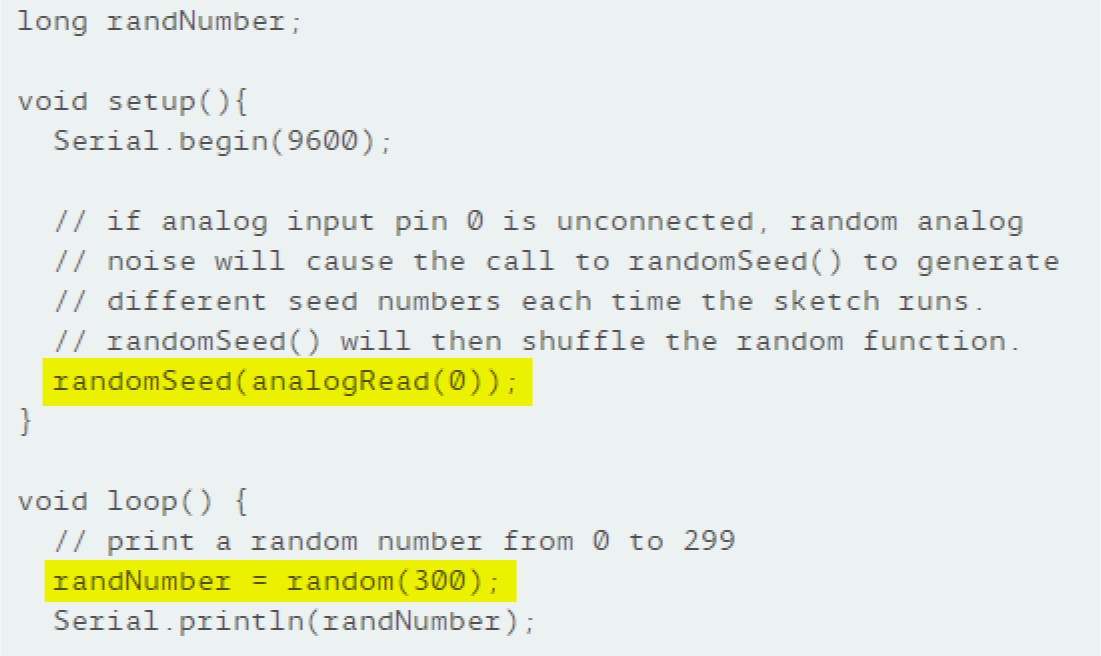
- Analog pins map input voltages between 0 and 5 volts into integer values between 0 and 1023
- Arduino Uno has analog pins A0 – A5
- Use analogRead(pin #); // to read from the analog pin
- Use analogWrite (pin #, duty cycle); // to write to the pin



```
int analogPin = 3;
int val = 0; // variable to store the value read

void setup(){
  Serial.begin(9600); //  setup serial
}

void loop() {
  val = analogRead(analogPin);  // read the input pin
  Serial.println(val);  // debug value
}
```

Project 3: Random generator

- Generate a random number (integer, long, etc) by reading noise from unused analog pin
- In setup, we need to create a randomSeed (analogRead(pin #));
- In loop, we generate number by doing random (min# , max #);



```
long randNumber;

void setup(){
  Serial.begin(9600);

  // if analog input pin 0 is unconnected, random analog
  // noise will cause the call to randomSeed() to generate
  // different seed numbers each time the sketch runs.
  // randomSeed() will then shuffle the random function.
  randomSeed(analogRead(0));
}

void loop() {
  // print a random number from 0 to 299
  randNumber = random(300);
  Serial.println(randNumber);
```

Project 3: Night lamp Pseudo Code

- **variables:**
  - int: pins and constants
  - unsigned long timer
  - boolean variable
- **setup():**
  - Set pinMode
  - Serial Motor
  - Random generator

Project 3: Night Lamp Pseudo Code

- **loop():**
  - Get the value from the checkBrightness() function
  - If the brightness < threshold
    - If led is on
      - turn on led by calling function
      - Update the Boolean variable
      - Update the timer
    - Else if it is time to change colour
      - Turn the led on again by calling function
  - Else
    - Turn off the LED
    - Update Boolean variable

Project 3: Functions Pseudo Code

- int **getBrightness** (): return the analog value from the photoresistor

- void **turnLEDOff**()
  - analogWrite(pin #, 0); // to turn off the LEDs

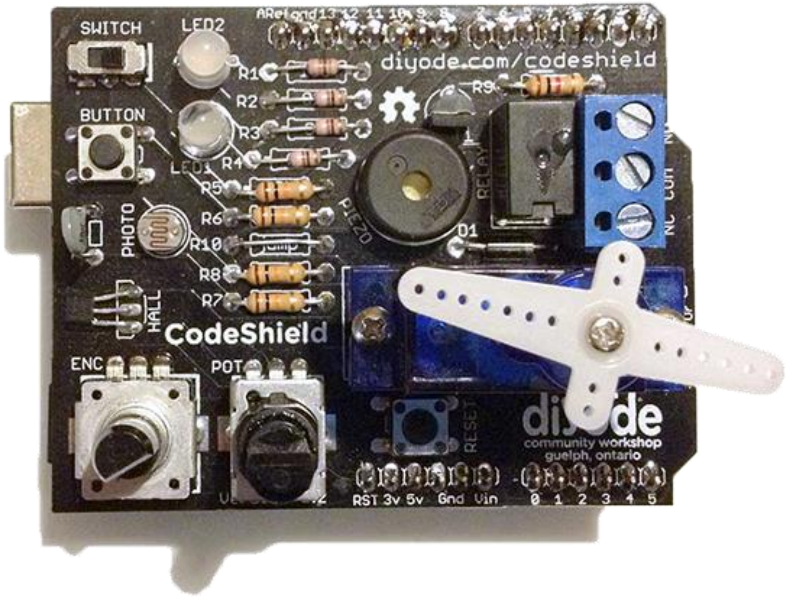
- void **turnLEDOn**()
  - Generate random values
  - analogWrite(pin #, randomValue); // to set intensity of the LEDs

Switch Gears

Let’s now look at the servo motor on Diyode CodeShield

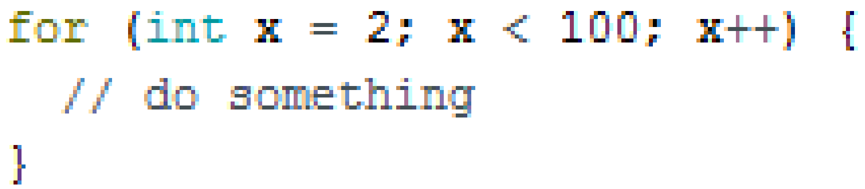
This piece of hardware handles the wiring for us!

- Servo motor comes with encoders, which allows us to identify the position of the motor
- Include these lines before your setup()
  - #define SERVO 5
  - #include <Servo.h>



For loops

- Used when you need to repeat something for a known number of times



```
for (int x = 2; x < 100; x++) {
  // do something
}
```

Arduino: Servo Class

1. To control a servo using Arduino, we need to import the servo motor class:
   - #include <Servo.h>
2. In our setup(), attach the servo pin to the Arduino
   - myservo.attach(SERVO); //SERVO is the pin defined in the previous slide: *#define SERVO 5*
3. In loop(), to tell the motor to turn, use the function
   - myservo.write(position); //position is an int variable, telling the motor how much to turn
   - Stick a delay() function right after write()!
   - delay(15); // in milliseconds

Project 4: Turn 180° then switch direction

Using the servo class (myServo.attach(), myServo.write()), delay(), and for loop, write a program

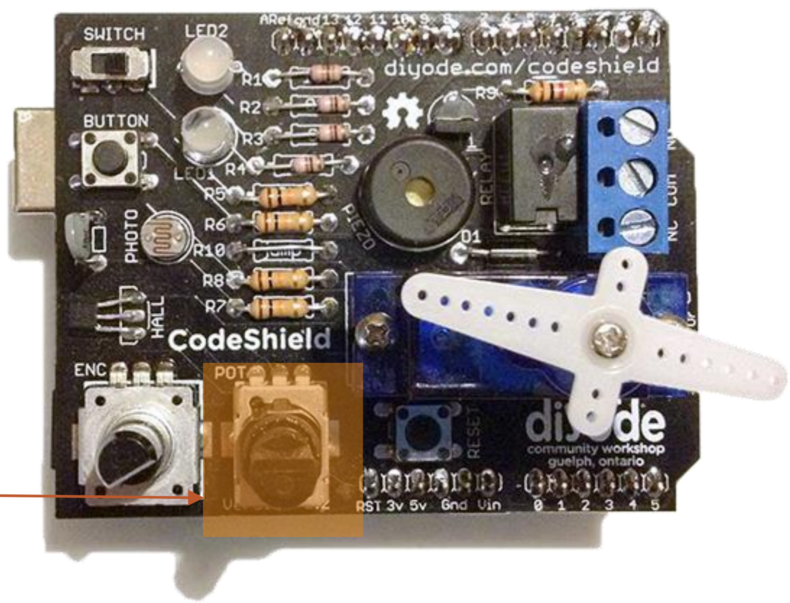
- Turn the motor from 0° to 180°
- Once reached 180 °, turn the motor back to 0°

How do you manipulate the for loop to go from 180°- 0° ?

Project 5: Servo Control - Potentiometer

- Potentiometer
  - A variable resistor
  - Resistance value changes based on the contact with the rotatable shaft
  - E.g.: volume control, etc

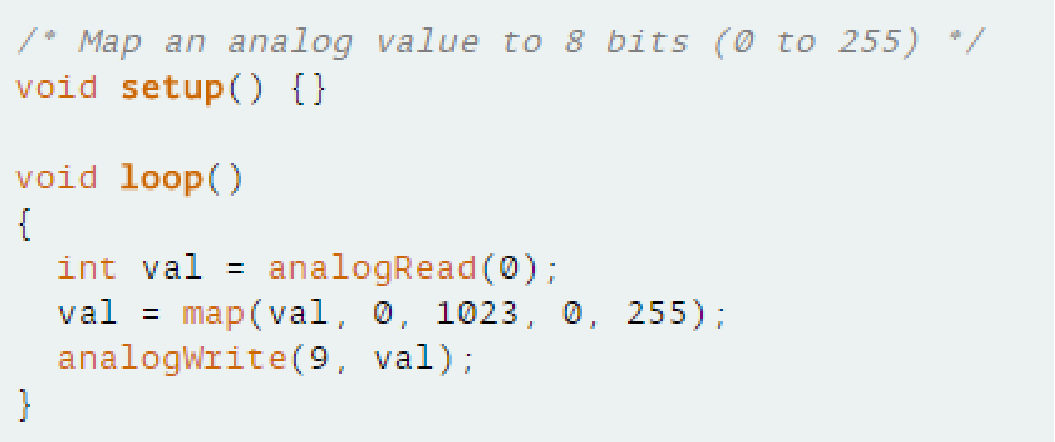
- On the CodeShield
  - #define POT 2



Potentiometer (POT)

Project 5: Map(variable, a, b, c, d);

- The values from the potentiometer exceeds the range of the servo motor
- Use map() to scale up/down the range between a-b to fit the range of c-d
- For our purpose:
  - variable = input (i.e. the analog value from the potentiometer)
  - a = min value from the potentiometer (0 A)
  - b = max value from the potentiometer (1023 A)
  - c = min value to be mapped to (0 ° for the servo)
  - d = max value to be mapped to (e.g.: 180 °)



```
/* Map an analog value to 8 bits (0 to 255) */
void setup() {}

void loop()
{
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
  analogWrite(9, val);
}
```

Project 5: Servo Control Pseudo Code

- **Before setup:**
  - Include the servo library
  - Create a servo object
  - Create an integer variable to store the input from the potentiometer
- **setup():**
  - Attach servo to the pin
- **loop():**
  - Read the analog input from the potentiometer; store it in the integer
  - Map the input range to the range of the motor
  - Write the position to the motor
  - Delay()!!!

Bonus

- Want to create an Android Application communicate with Bluetooth check this out:
  - Guide on How to Use App Inventor with Arduino
  - App Inventor

- Want to create a Matlab program/GUI using Arduino:
  - Matlab: You want to get the hardware support package
  - Simulink: Get this support package

- Interested about how the processing core works
  - You want to check out: assembly languages (low-level programming language)
  - Here is a neat tutorial on assembly language

Bonus

- Arduino Built-in Functions:
  - http://arduino.cc/en/Reference/HomePage

- Interested about how the processing core works
  - You want to check out: assembly languages (low-level programming language)
  - Here is a neat tutorial on assembly language

Bonus – Python with Arduino

- For those that wanna see how micro-controllers can be used to communicate with the computer. Install Python 2 (in specific Python 2.7.3)
- On Windows: using Python in the command prompt:

  1. Go to the Control panel in the start menu
  2. Click on System Properties control
  3. Go to “Environment Variables”
  4. Select "Path", and then in the bottom section (Systems Variables) select "Edit"
  5. At the end of the “Variable Value” without deleting any of the text already there, add the text: ";C:\Python27"

- Install PySerial
  - PySerial allows access to serial ports and automatically selects the appropriate back-end. For information on PySerial is available here.
  - For all OS, download the .tar.gz install package for PySerial 2.6 from https://pypi.python.org/pypi/pyserial This will give you a file called: pyserial-2.6.tar.gz
- Decompress the folder:
- Windows: Install 7zip to decompress the file.
- Mac or Linux: Open a Terminal session, and go to where you've downloaded pyserial-2.6.tar.gz and then issue the following command to unpack the installation folder.

Image References

http://images.memes.com/meme/830710

https://cdn.instructables.com/F6R/IPAP/HQF9H5IO/F6RIPAPHQF9H5IO.MEDIUM.jpg

http://image.slidesharecdn.com/introductiontoembeddedsystems-091122110735-phpapp01/95/introduction-to-embedded-systems-30-728.jpg?cb=1258888068

http://memecrunch.com/meme/1TG1/arduino-like-a-boss/image.jpg

http://i.stack.imgur.com/QJ9mt.jpg

http://circuitdigest.com/sites/default/files/circuitdiagram_mic/Arduino-LED-Circuit.gif

https://www.arduino.cc/en/uploads/Tutorial/ExampleCircuit_sch.png

https://cdn.instructables.com/FOH/F83V/IAMCFJPJ/FOHF83VIAMCFJPJ.MEDIUM.jpg

http://codeshield.diyode.com/igg/images/codeshield.png