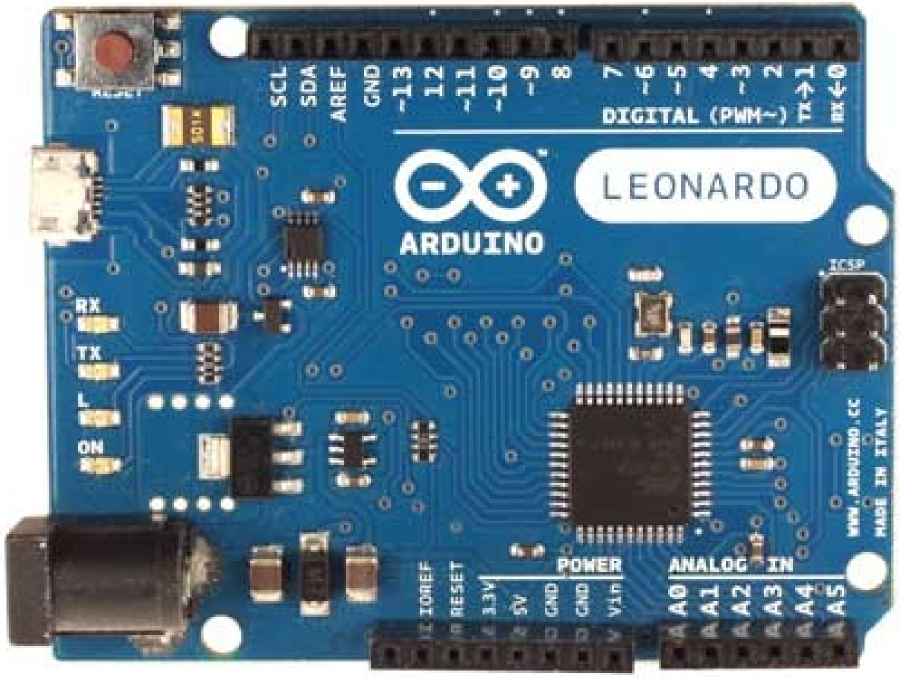
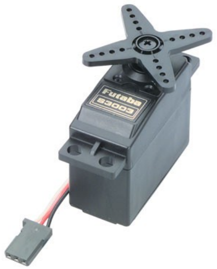
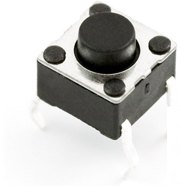
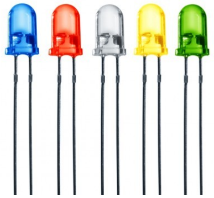
Arduino Electronics Introduction

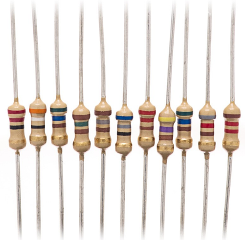
Welcome! This tutorial is designed to teach you the basics of making your own electronic circuits using the Arduino microcontroller, the blue thing on the left. You will learn how to safely connect different kinds of components to the Arduino and how to program it to interact with the outside world.



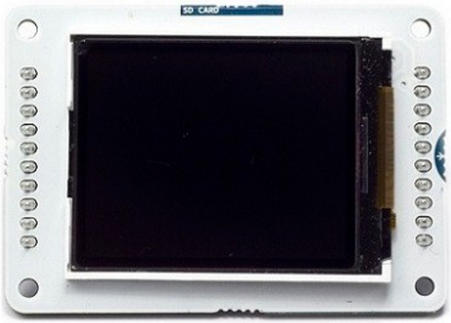






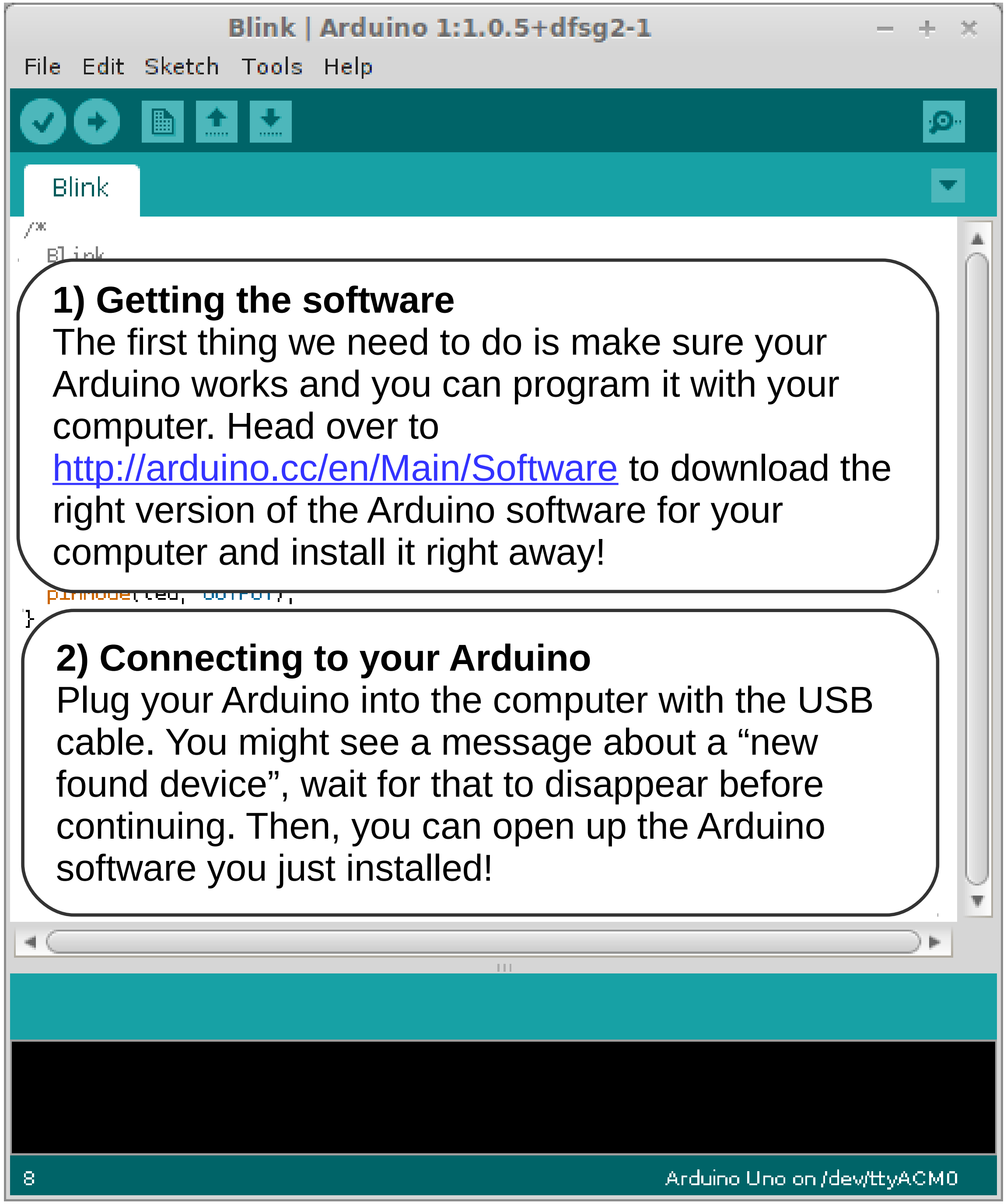


We will be using several different electronic components with the Arduino. Can you name the ones pictured on the page?



Don't worry if you have never programmed an Arduino before, it's the easiest thing in the world!
Have a look at: http://arduino.cc/en/Tutorial

Arduino Electronics Introduction



```
/*
  Blink
...
  pinMode(led, OUTPUT);
}
```

8 Arduino Uno on /dev/ttyACM0

1) Getting the software

The first thing we need to do is make sure your Arduino works and you can program it with your computer. Head over to http://arduino.cc/en/Main/Software to download the right version of the Arduino software for your computer and install it right away!

2) Connecting to your Arduino

Plug your Arduino into the computer with the USB cable. You might see a message about a “new found device”, wait for that to disappear before continuing. Then, you can open up the Arduino software you just installed!

3) Load An Example Program

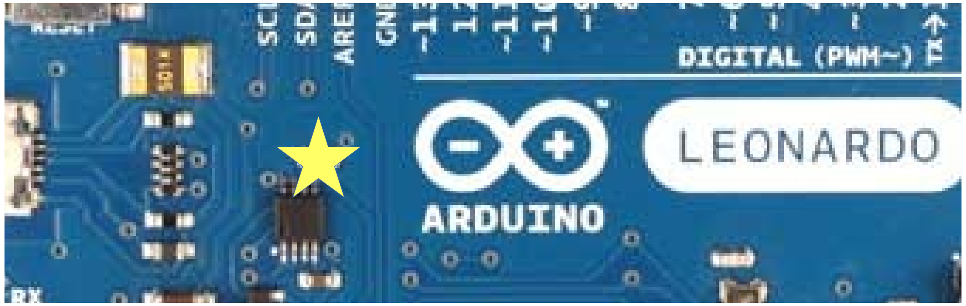
When the software opens, click on: File -> Examples -> 01 Basics -> Blink. You will see another window open with some example code. Don't worry if you don't understand yet, we'll get to that!

4) Configure your board

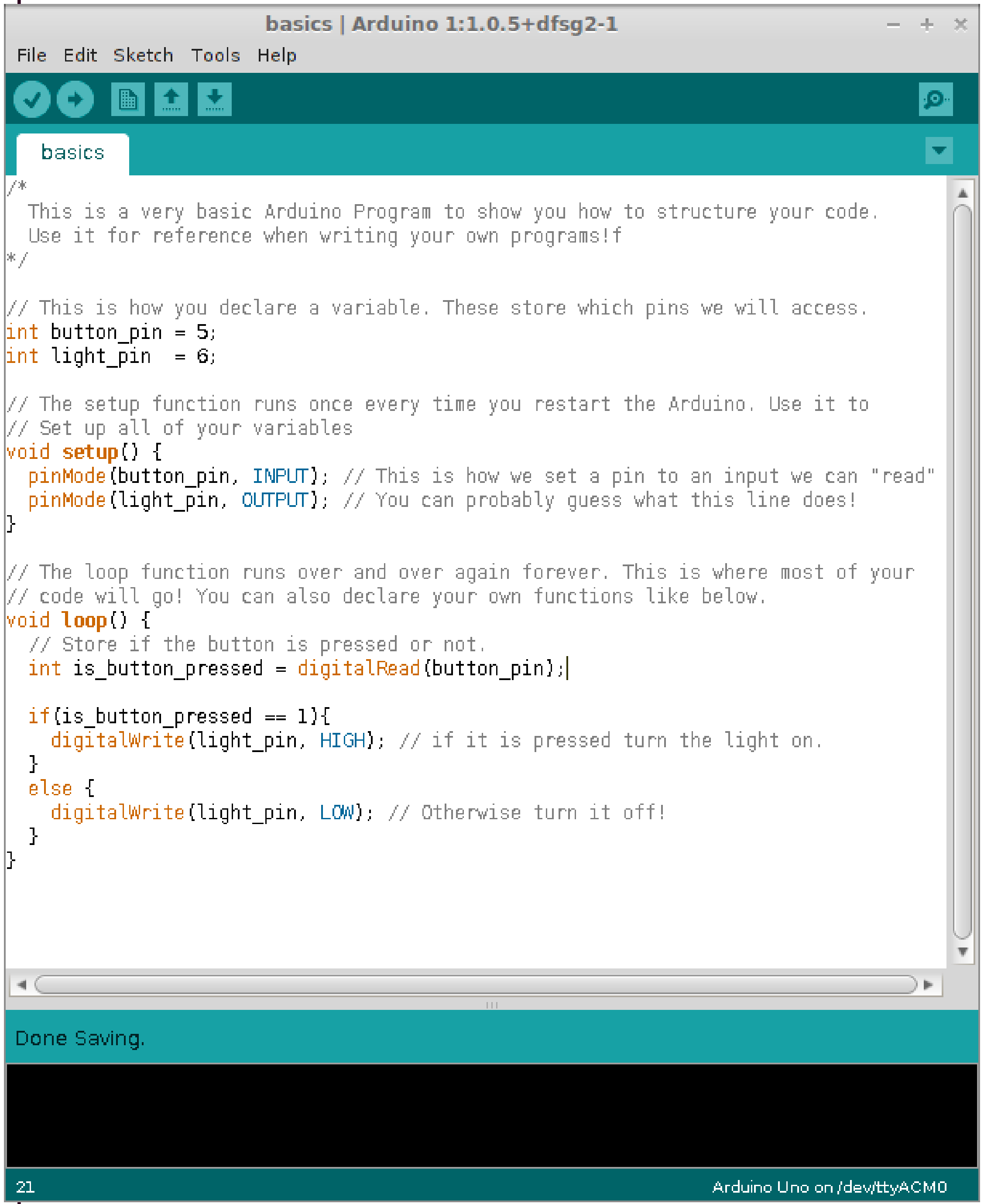
We are using an “Arduino Leonardo”. So under Tools -> Board Select the Arduino Leonardo! You will also need to select the right “port” under Tools -> Port. It is usually the one with the biggest number at the end!

5) Program the board!

Click on the button with the arrow pointing right under “Edit”. This will compile your code and “upload” it to the Arduino! When it finishes, you should see a single LED on the board flashing on and off. Congratulations, you now know how to program and Arduino!



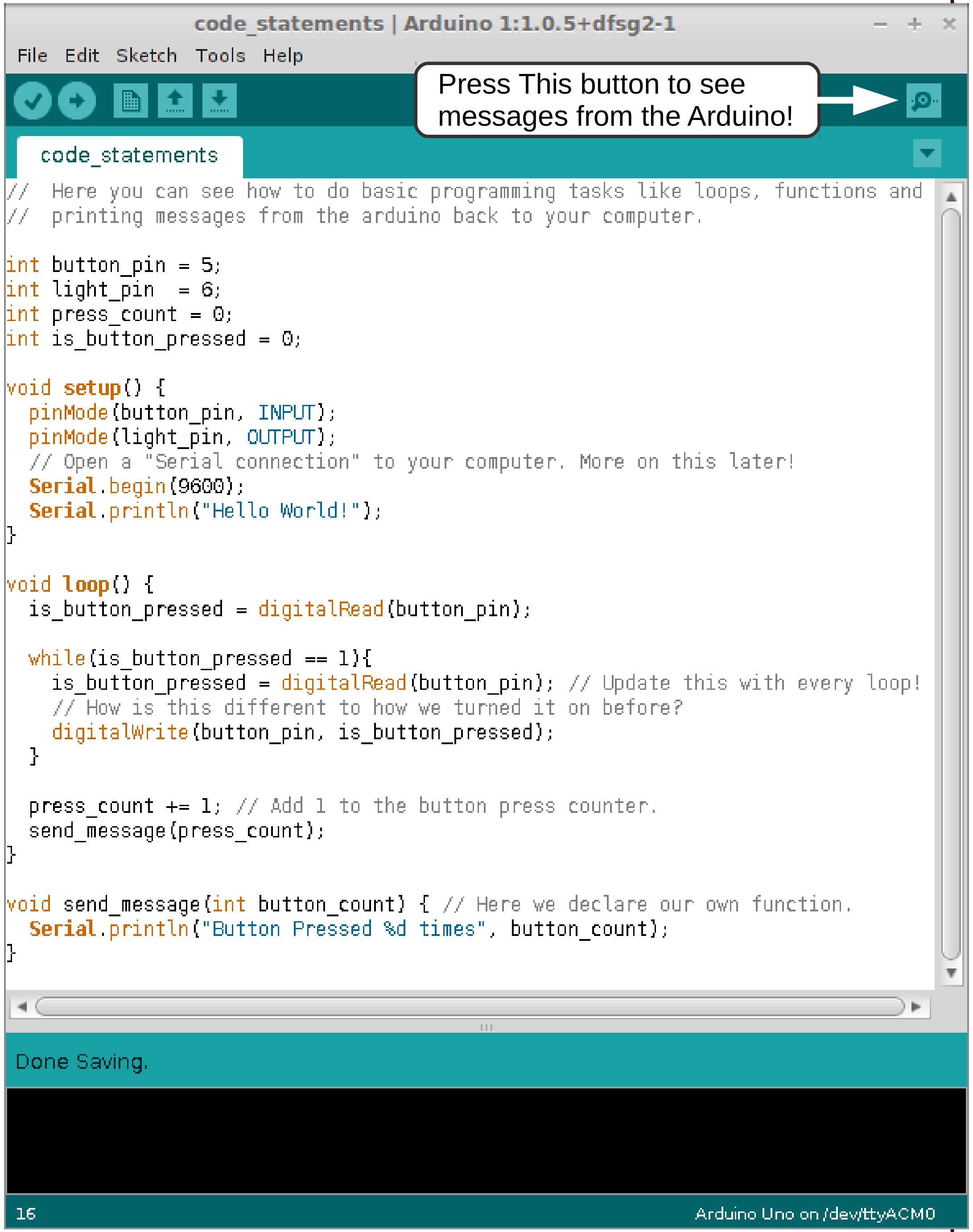
Arduino Electronics Introduction



```
/*
  This is a very basic Arduino Program to show you how to structure your code.
  Use it for reference when writing your own programs!f
*/

// This is how you declare a variable. These store which pins we will access.
int button_pin = 5;
int light_pin  = 6;

// The setup function runs once every time you restart the Arduino. Use it to
// Set up all of your variables
void setup() {
  pinMode(button_pin, INPUT); // This is how we set a pin to an input we can "read"
  pinMode(light_pin, OUTPUT); // You can probably guess what this line does!
}

// The loop function runs over and over again forever. This is where most of your
// code will go! You can also declare your own functions like below.
void loop() {
  // Store if the button is pressed or not.
  int is_button_pressed = digitalRead(button_pin);

  if(is_button_pressed == 1){
    digitalWrite(light_pin, HIGH); // if it is pressed turn the light on.
  }
  else {
    digitalWrite(light_pin, LOW); // Otherwise turn it off!
  }
}
```

Done Saving.

21
Arduino Uno on /dev/ttyACM0


code_statements | Arduino 1:1.0.5+dfsg2-1
File Edit Sketch Tools Help
code_statements

Press This button to see messages from the Arduino!

```
//  Here you can see how to do basic programming tasks like loops, functions and
//  printing messages from the arduino back to your computer.

int button_pin = 5;
int light_pin  = 6;
int press_count = 0;
int is_button_pressed = 0;

void setup() {
  pinMode(button_pin, INPUT);
  pinMode(light_pin, OUTPUT);
  // Open a "Serial connection" to your computer. More on this later!
  Serial.begin(9600);
  Serial.println("Hello World!");
}

void loop() {
  is_button_pressed = digitalRead(button_pin);

  while(is_button_pressed == 1){
    is_button_pressed = digitalRead(button_pin); // Update this with every loop!
    // How is this different to how we turned it on before?
    digitalWrite(button_pin, is_button_pressed);
  }

  press_count += 1; // Add 1 to the button press counter.
  send_message(press_count);
}

void send_message(int button_count) { // Here we declare our own function.
  Serial.println("Button Pressed %d times", button_count);
}
```

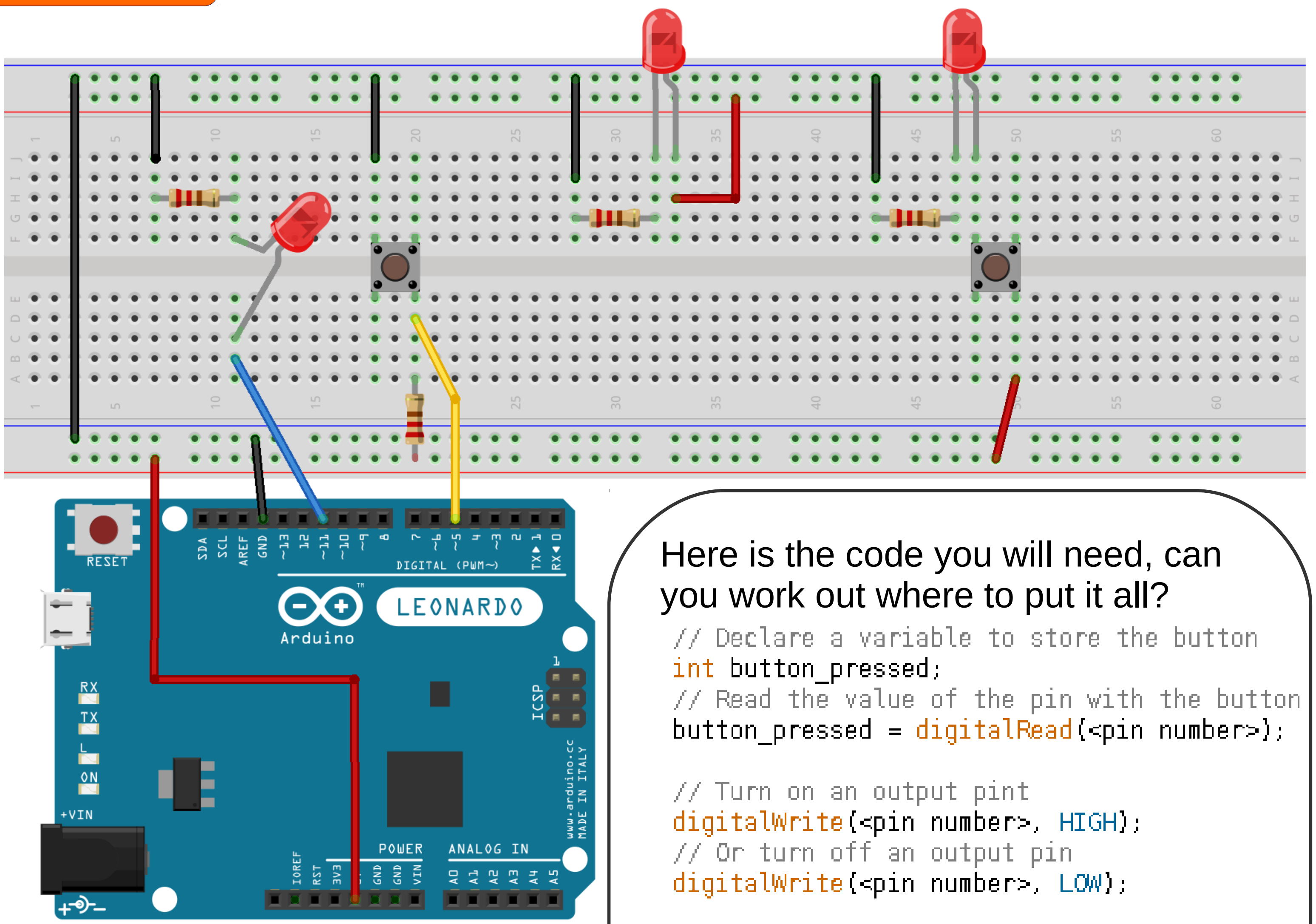
Done Saving.

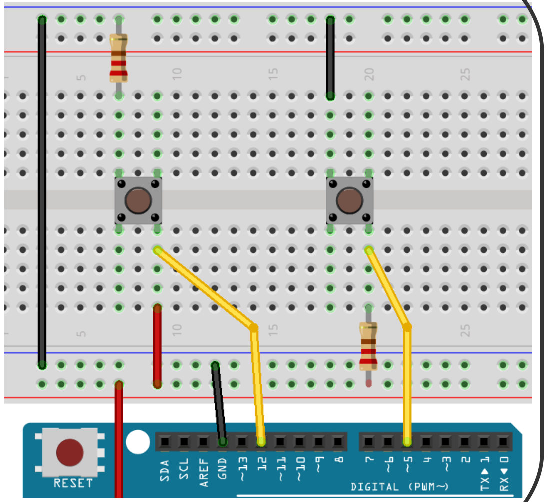
Arduino Electronics Introduction

LED's and Resistors

These are the two most basic components we can use. Try copying this circuit and controlling the LED with the Button and the Arduino

We **always use a resistor with an LED.** This makes sure it doesn't draw too much power and burn out!





Try building the circuit on the left. What do you notice is different about the values the buttons normally take? Print them out so you can see them.

Here is the code you will need, can you work out where to put it all?

```
// Declare a variable to store the button
int button_pressed;
// Read the value of the pin with the button
button_pressed = digitalRead(<pin number>);

// Turn on an output pint
digitalWrite(<pin number>, HIGH);
// Or turn off an output pin
digitalWrite(<pin number>, LOW);

// if statement
if(<my variable> == <some value>){
    // DO something!
}
else {
    // DO something ELSE!
}
```

Potentiometers
+
Servos

LCD Screen

Arduino Reference

TEMPLATE PAGE