Lesson 2 – Sensor data collection (part I)

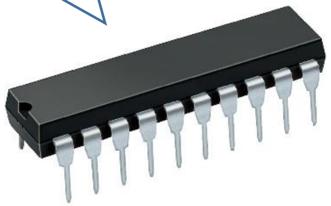
- S.P. Chong

Objectives

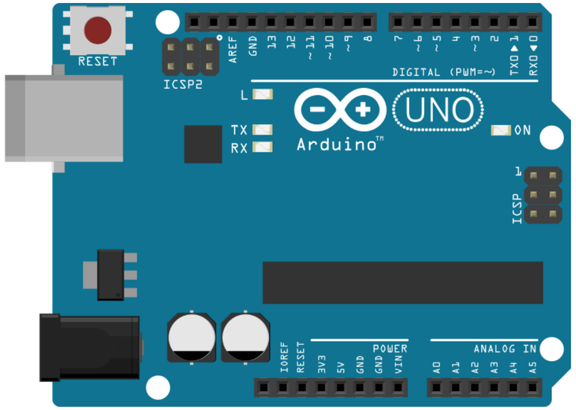
- In this lesson, you will learn to **program** an **Arduino UNO** (a microcontroller) to **collect sensor data**.
- You will learn the **characteristics** & **applications** of various **sensors** (& **actuators**).
- You will also learn basic **interfacing techniques** using **breadboard** & **wires** required to connect the microcontroller to the (**digital**/**analogue**) sensors & actuators.

What is a microcontroller?

Microcontroller IC.



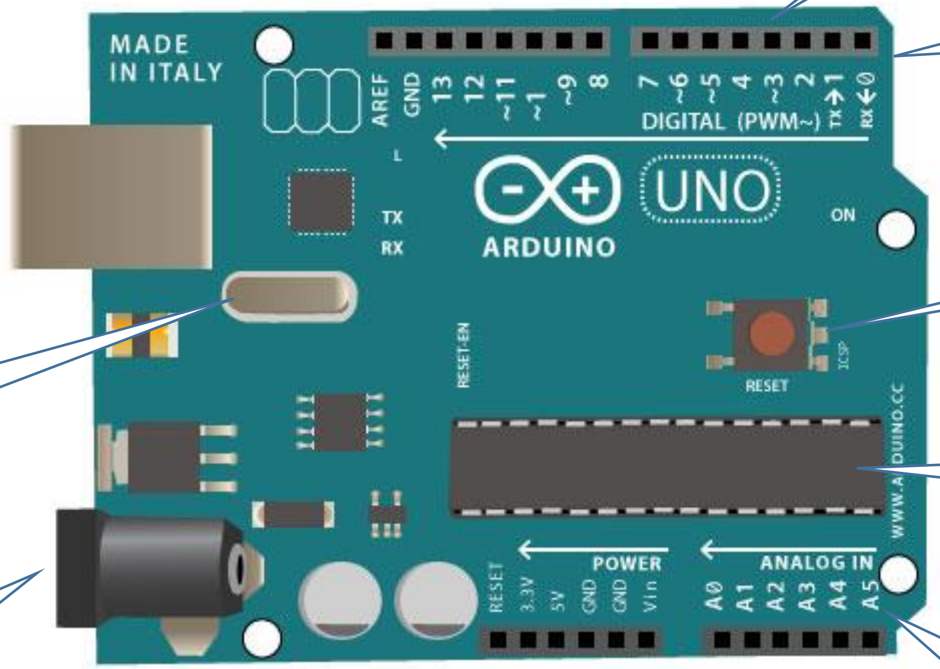
Microcontroller board: Arduino UNO.



- A microcontroller is a **small computer** on a **single IC** (Integrated Circuit).
- It has a **processing unit**, some **memory** for storing **program** & **data**, and some **I/O** (Input/Output) **pins** for connecting to the outside world.
- It is used in applications where **intelligent control** is needed.
- The microcontroller you will learn, **Arduino UNO**, is easy to use for 2 reasons:
  - Being **open-sourced**, many **sample codes** to perform various tasks can be found **online**, or even in the **IDE** (Integrated Development Environment – more on this later).
  - Many **shields** can be purchased to make **hardware connection** easy (e.g. MP3 player shield, GPS shield).

What is a microcontroller? (cont.)

- These are the **key features** of an Arduino UNO:



Current is 20mA.

0-13: 14 digital input / output pins.

Those with ~ (pins 3, 5, 6, 9, 10, 11) can be used as PWM output.

USB connection to PC, to program the UNO.

Reset button.

32KB flash, 2KB SRAM, 1KB EEPROM.

16 MHz crystal.

ATMEGA328P microcontroller.

DC power in.

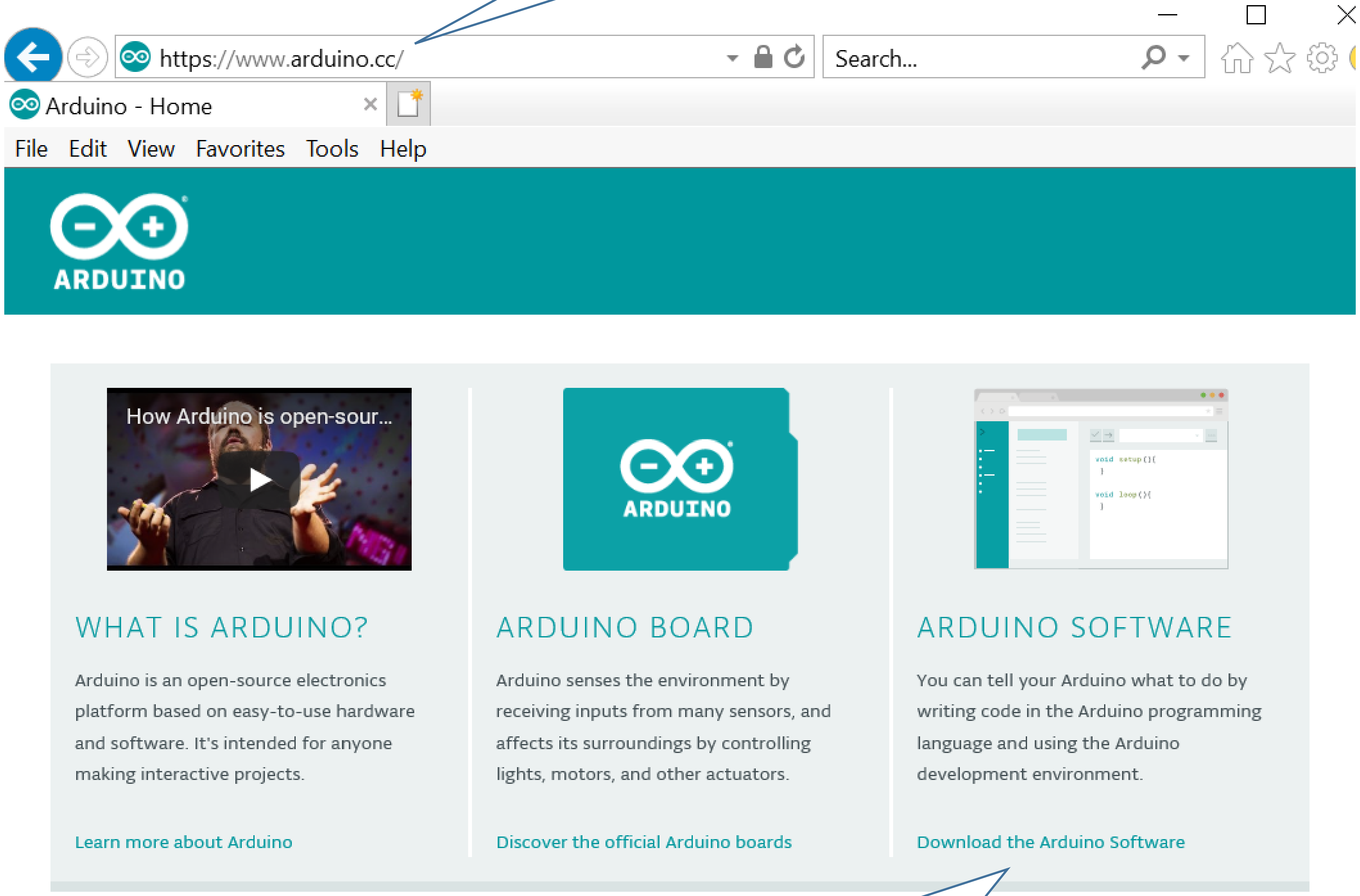
A0-A5: 6 analogue input pins.

5V operation.

How to use Arduino UNO?

- You can download and install the **Arduino IDE** (Integrated Development Environment) from www.arduino.cc
- This allows you to **write** a program (or sketch), **compile** & **download** to the Arduino UNO.

Official website.



WHAT IS ARDUINO?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

Learn more about Arduino

ARDUINO BOARD

Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

Discover the official Arduino boards

ARDUINO SOFTWARE

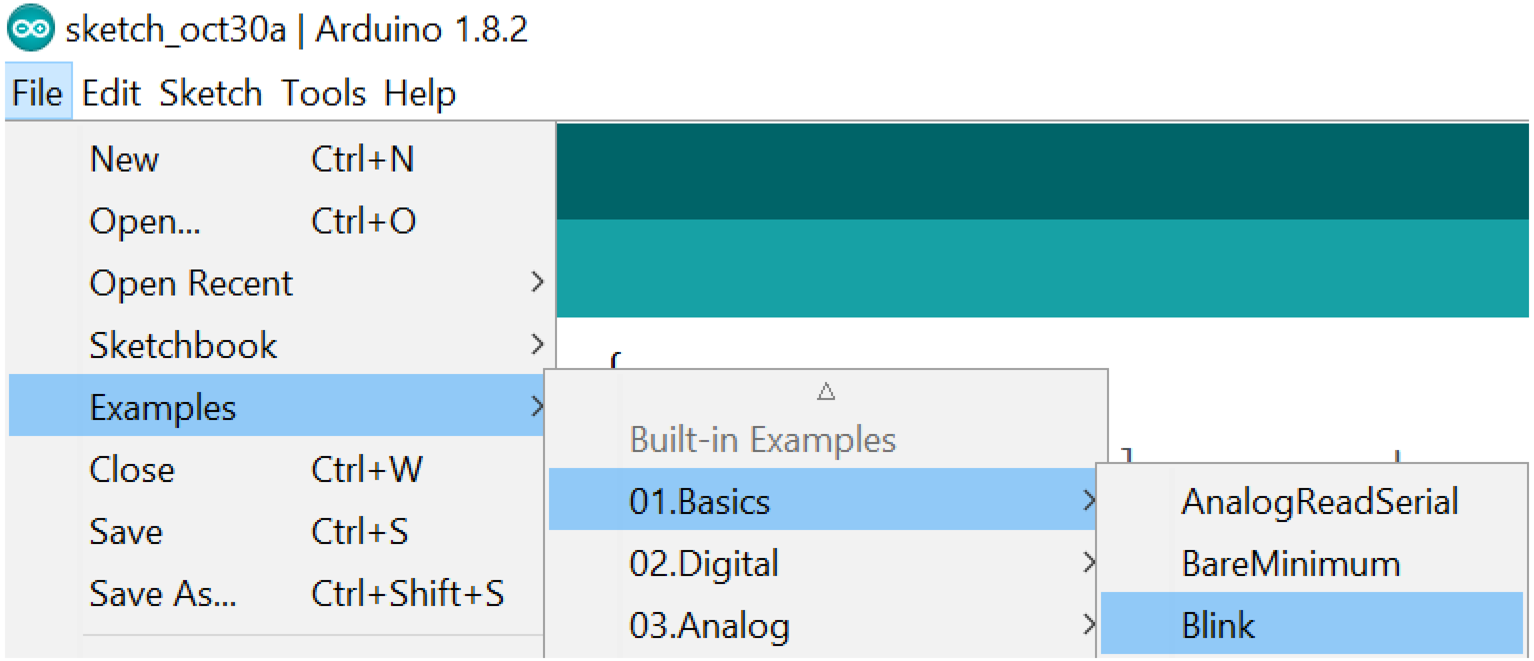
You can tell your Arduino what to do by writing code in the Arduino programming language and using the Arduino development environment.

Download the Arduino Software

Download IDE.

How to use Arduino UNO? (cont.)

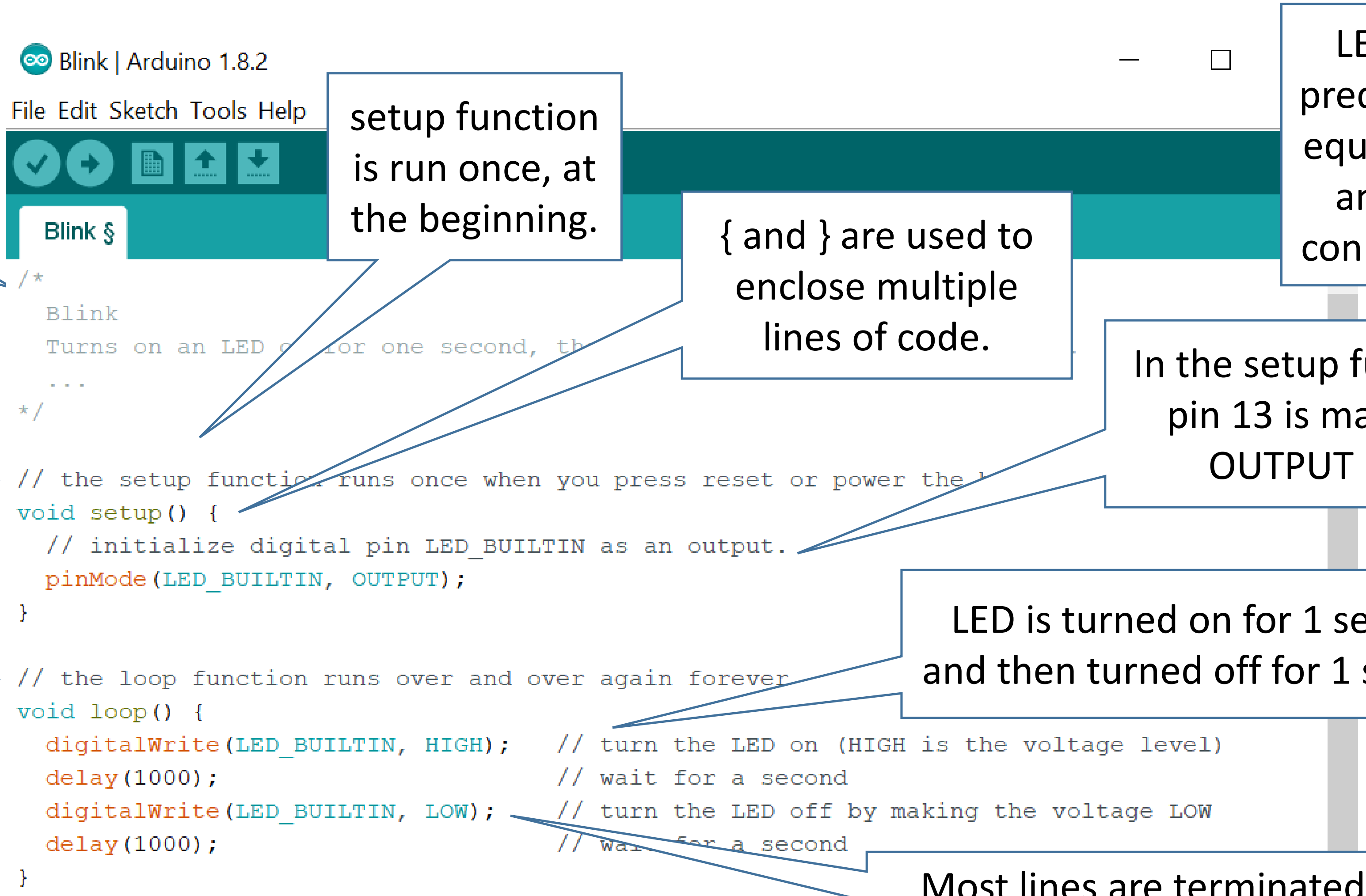
- As mentioned earlier, many **sample** codes (called “**sketches**”) are available, in the IDE.
- Click File -> Examples -> 01. Basics -> Blink to open the “**Blink**” sample sketch.



Blink sample sketch.

How to use Arduino UNO? (cont.)

- Let’s try to understand this sample sketch…



```
/*
  Blink
  Turns on an LED on for one second, th
  ...
*/

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

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

/* … */ enclose multi-line comment.

// precedes single-line comment.

loop function is run over and over again, after setup.

setup function is run once, at the beginning.

{ and } are used to enclose multiple lines of code.

LED_BUILTIN is a predefined constant, equal to 13. There is an on-board LED connected to pin 13.

In the setup function, pin 13 is made an OUTPUT pin.

LED is turned on for 1 sec, and then turned off for 1 sec.

Most lines are terminated with a semicolon.

How to use Arduino UNO? (cont.)

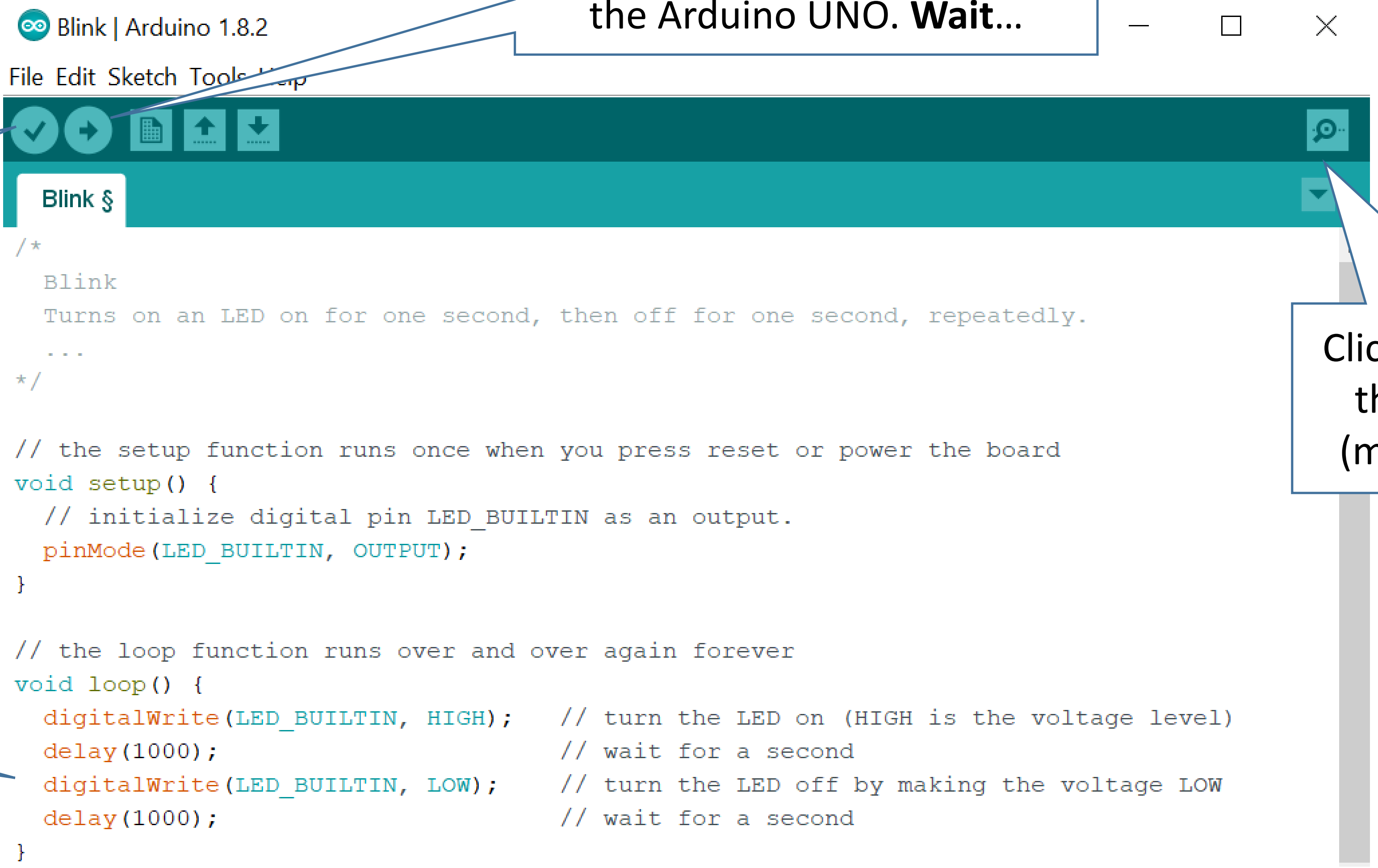
Question: why must the sketch be compiled?

Click this icon to download to the Arduino UNO. **Wait…**

Click this icon to compile the sketch i.e. verify the syntax.

Click this icon to open the Serial Monitor (more on this later).

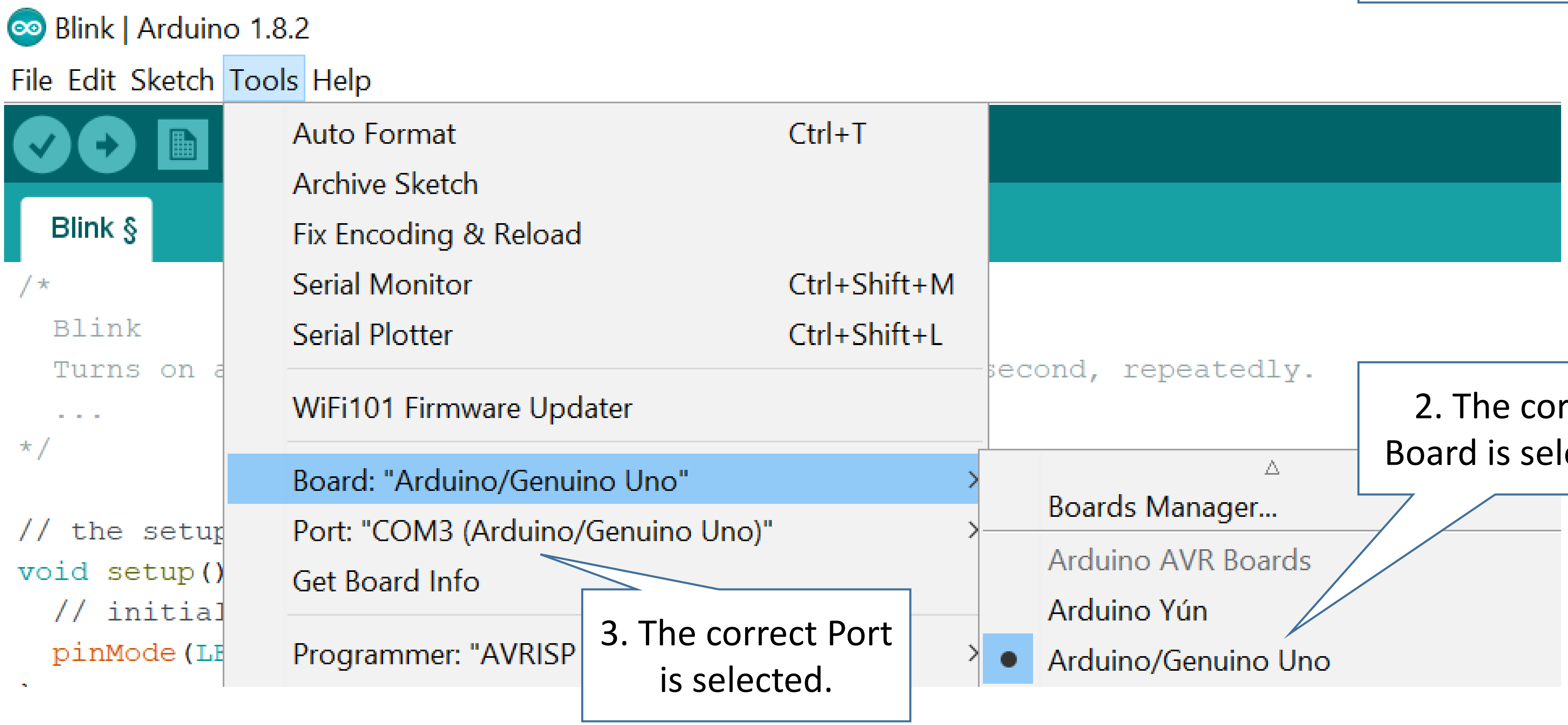
Question: How to make the LED blink twice as fast?



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

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

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

How to use Arduino UNO? (cont.)

Before you download, ensure that:
1. The Arduino UNO is connected to the laptop by a USB cable.



Board: "Arduino/Genuino Uno"

Port: "COM3 (Arduino/Genuino Uno)"

```
/*
  Blink
  Turns on a                                          second, repeatedly.
  ...
*/

// the setup
void setup()
  // initial
  pinMode(LE
```

2. The correct Board is selected.

3. The correct Port is selected.

Digital output (a summary)

Pin must be made an OUTPUT pin.

Pin number: 0 to 13
Value: HIGH or LOW.

digitalWrite (pin number, value);

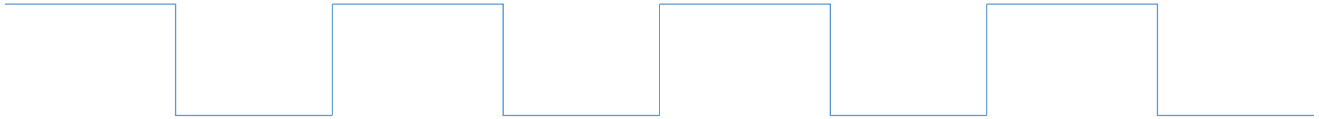
To turn on an LED at pin 13:

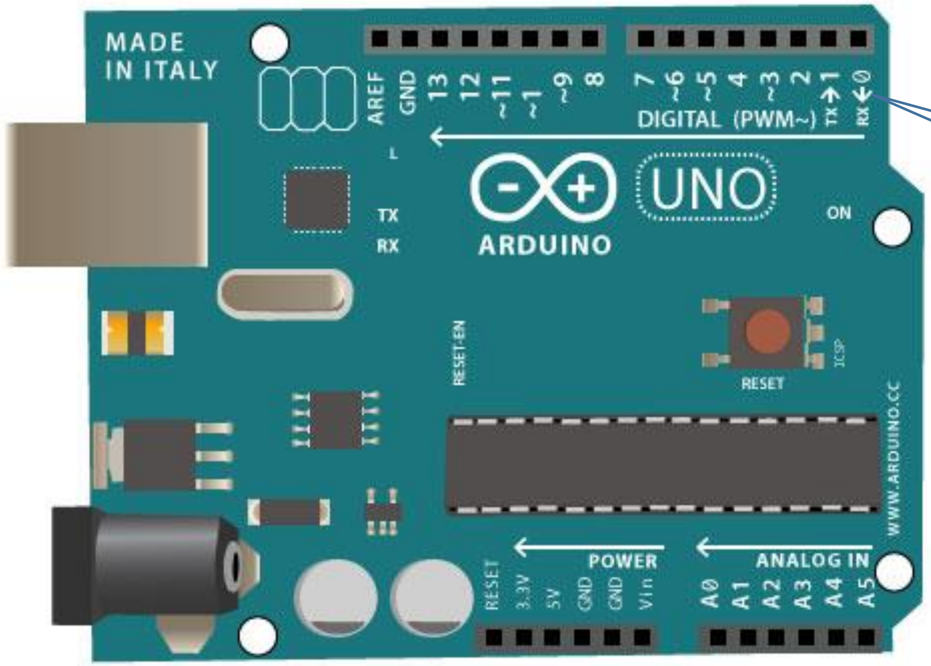
```
// in setup function
…
pinMode(13, OUTPUT);
…

// in loop function
…
digitalWrite (13, HIGH); // or LOW
…
```

Analogue / PWM output

- **PWM** (Pulse Width Modulation) is a method that allows a fast changing digital output to “**emulate**” an **analogue output**.

If the waveform is high 50% of the time, the average voltage is 2.5V.





Only those pins with ~ (3, 5, 6, 9, 10, 11) can be used as PWM output.

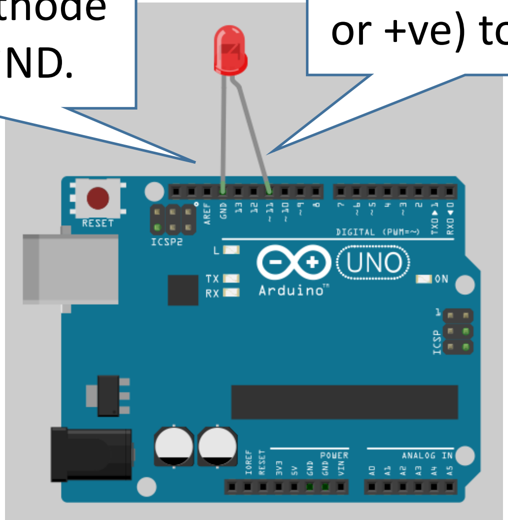
Analogue / PWM output (cont.)

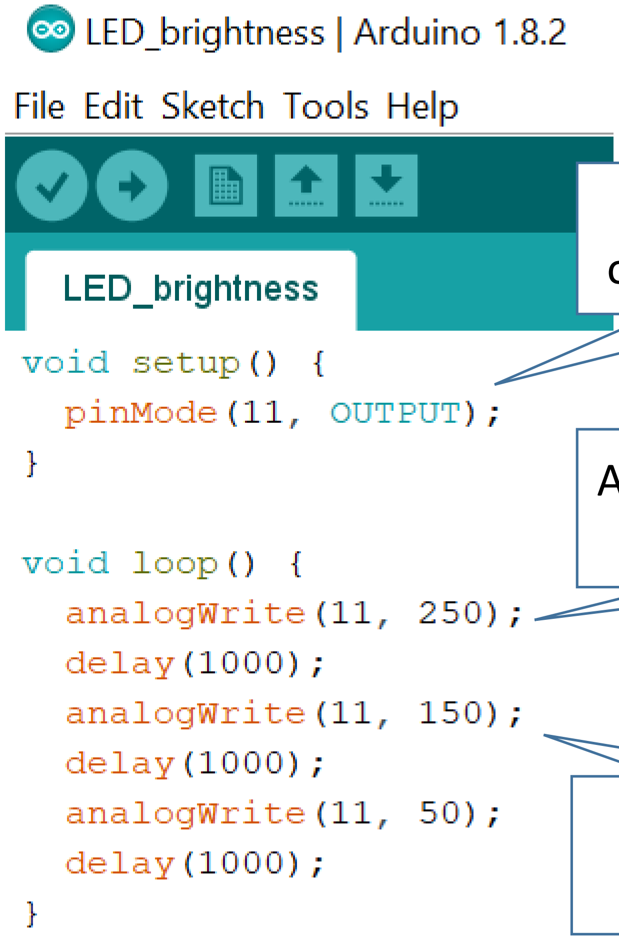
- Connect an **LED** to pin 11, and modify the Blink sketch to the LED_brightness sketch as shown.
- Upload the sketch and see what happens.

Short leg (cathode or -ve) to GND.

Long leg (anode or +ve) to pin 11.

Good practice to remove power, when changing connection.





```
void setup() {
  pinMode(11, OUTPUT);
}

void loop() {
  analogWrite(11, 250);
  delay(1000);
  analogWrite(11, 150);
  delay(1000);
  analogWrite(11, 50);
  delay(1000);
}
```

Make pin 11 an output pin (optional).

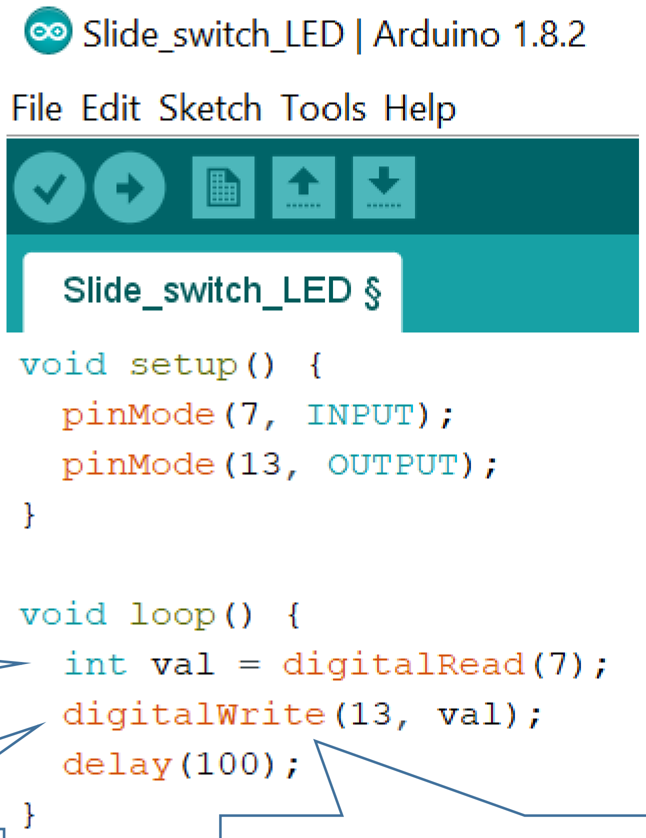
Analog values range from 0 to 255 (i.e. 8-bit).

3 levels of brightness: bright, medium, dim…

Digital input

- Connect a **slide switch** to pin 7 and an LED to pin 13, and modify any sketch to the Slide_switch_LED sketch as shown.
- Upload the sketch and see what happens, when you change the slide switch position.

If you have never used a breadboard & wires to connect up a circuit, the lecturer will explain to you how this is done.

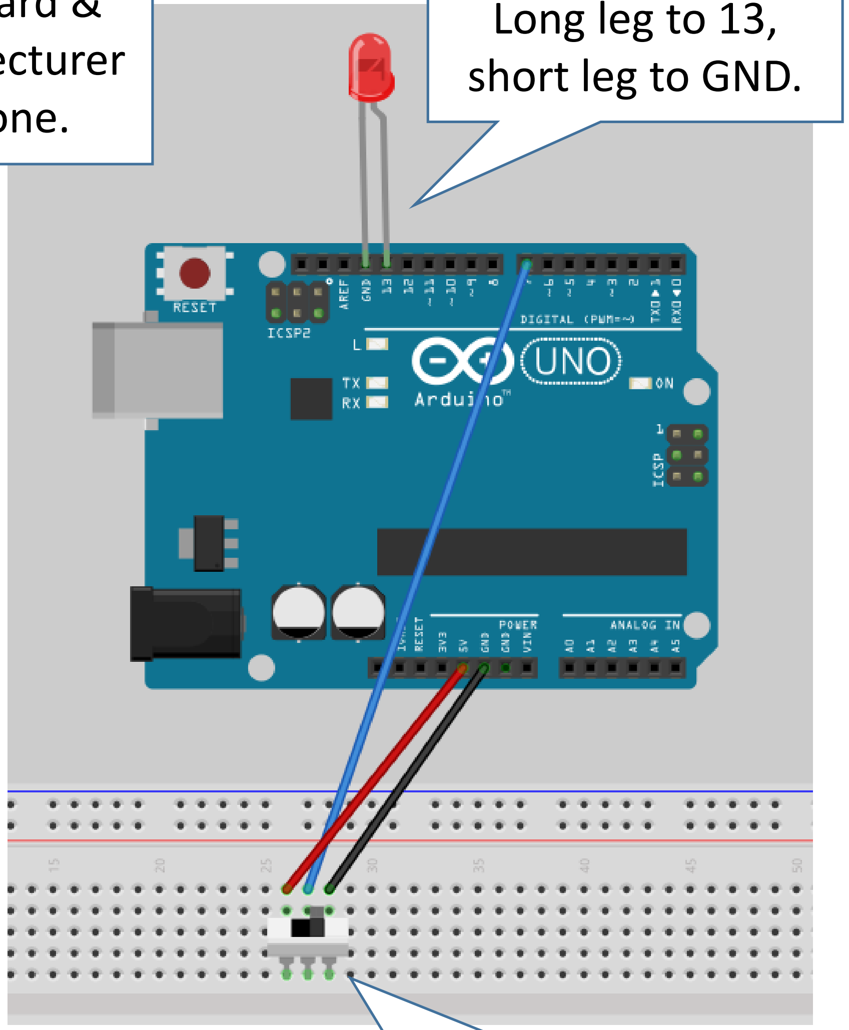


```
void setup() {
  pinMode(7, INPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  int val = digitalRead(7);
  digitalWrite(13, val);
  delay(100);
}
```

The slide switch output is read into an integer variable (val), which is then written to the LED pin.

If the slide switch connects 5V to pin 7, digitalRead gives HIGH.

If the slide switch connects GND to pin 7, digitalRead gives LOW.



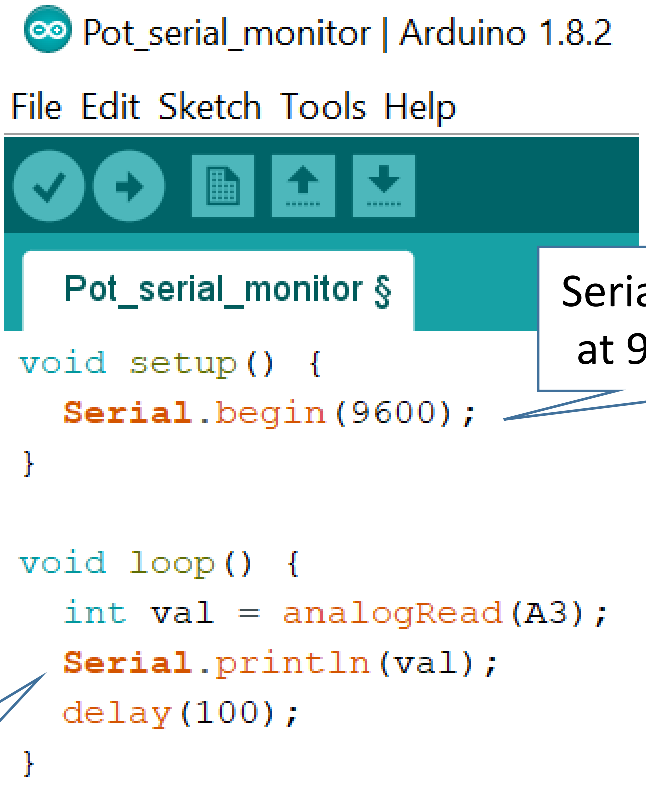
Long leg to 13, short leg to GND.

5V to left, pin 7 to centre lead, GND to right.

Analogue input

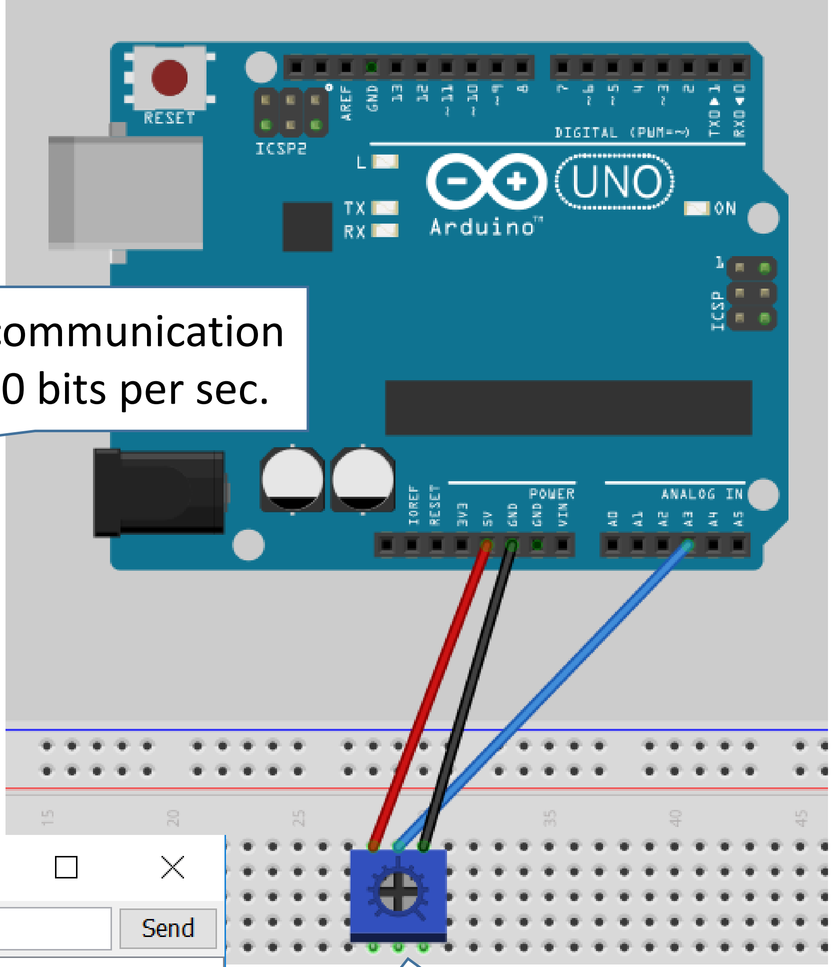
- Connect a **potentiometer** output to pin A3, and modify any sketch to the Slide_switch_LED sketch as shown.
- Upload the sketch, open the Serial Monitor, and see what happens when you turn the potentiometer.

The potentiometer output is read into an integer variable (val), which is then printed to the Serial Monitor.

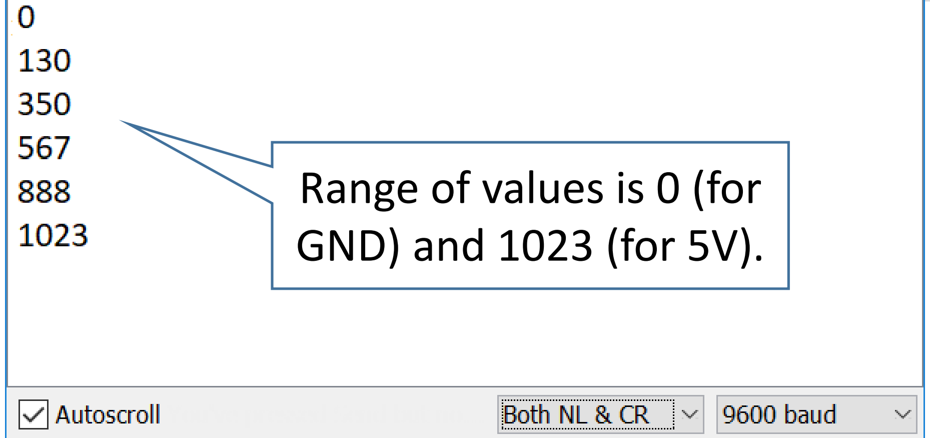

Pot_serial_monitor | Arduino 1.8.2
File Edit Sketch Tools Help
Pot_serial_monitor §

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

void loop() {
  int val = analogRead(A3);
  Serial.println(val);
  delay(100);
}
```

Serial communication at 9600 bits per sec.



COM3 (Arduino/Genuino Uno)



```
0
130
350
567
888
1023
```

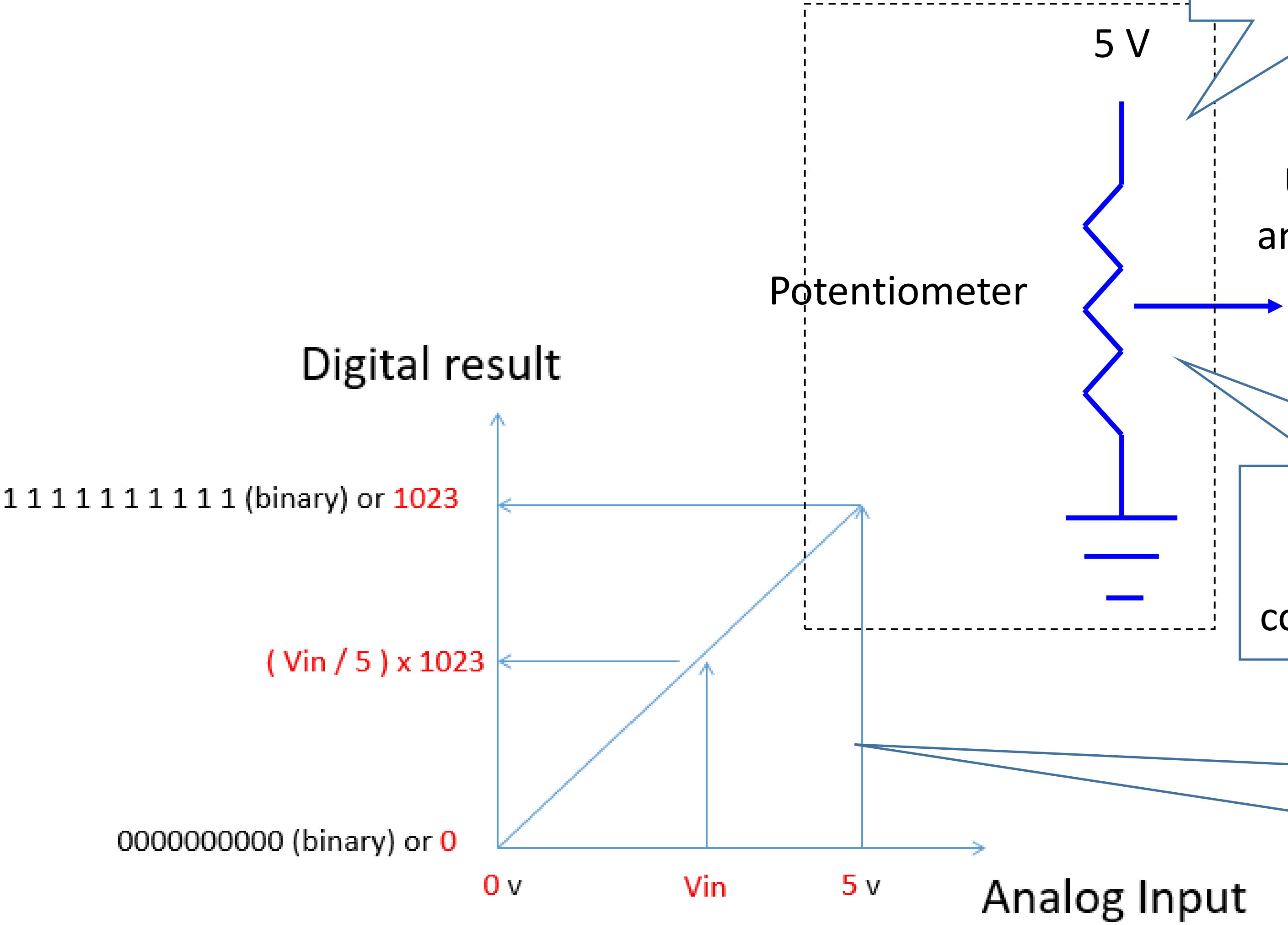
☑ Autoscroll Both NL & CR 9600 baud

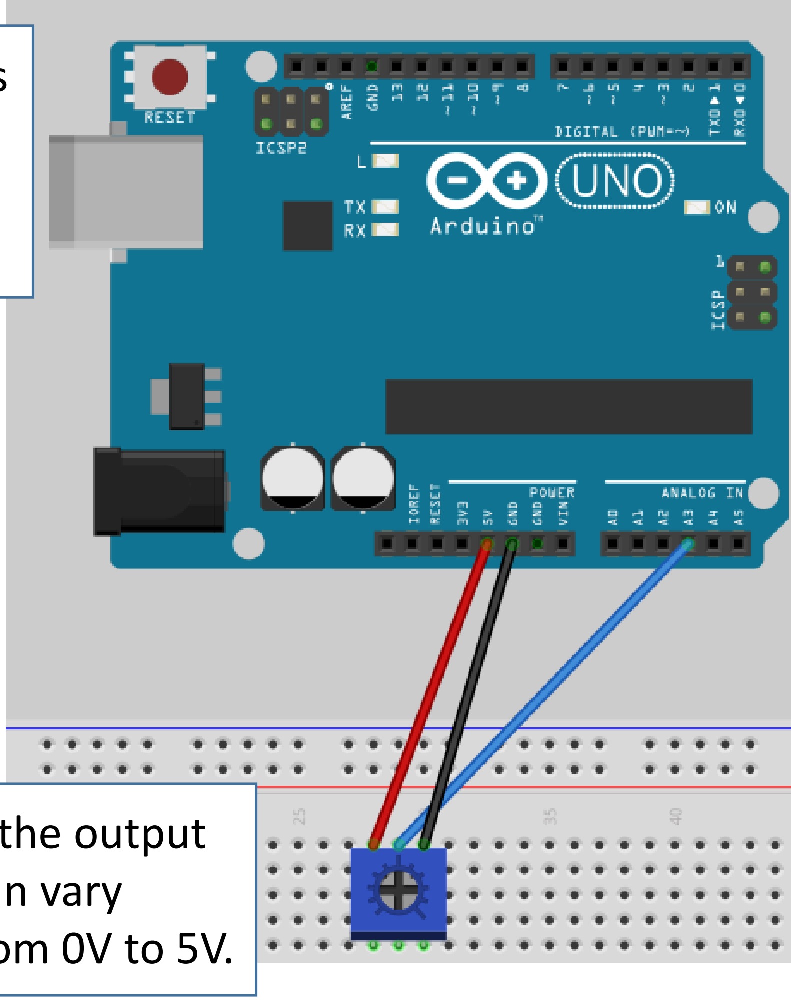
Range of values is 0 (for GND) and 1023 (for 5V).

5V to left, pin A3 to centre lead, GND to right.

Analogue input (cont.)

A potentiometer is a variable resistor, connected to 5V and GND.





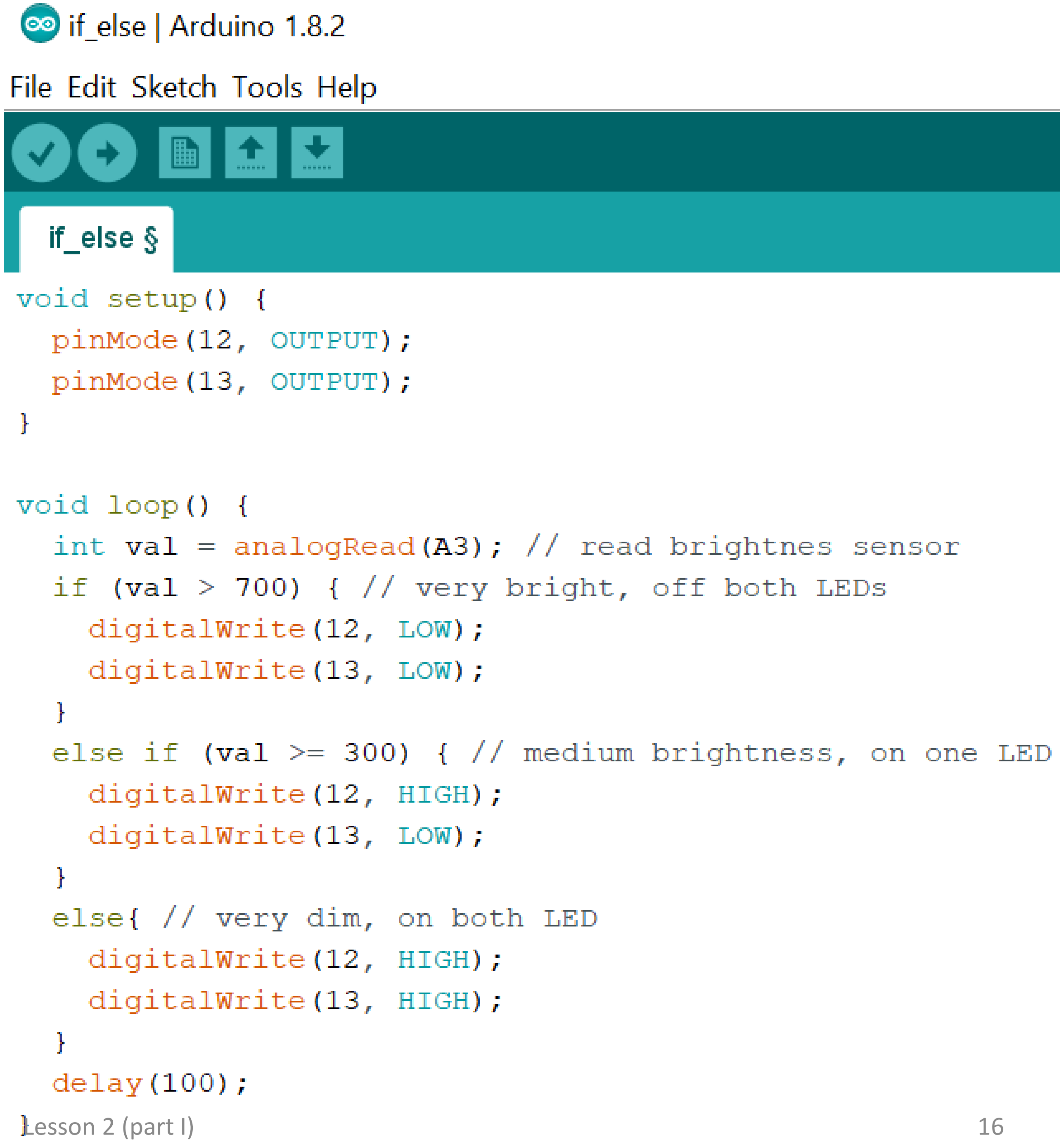
By rotating it, the output voltage can vary continuously from 0V to 5V.

analogRead converts this into a 10-bit number, from 0 (for GND) to 1023 (for 5V).

if else

- **If-else** is used to control the flow of the program i.e. which of a few alternative sets of code get executed.

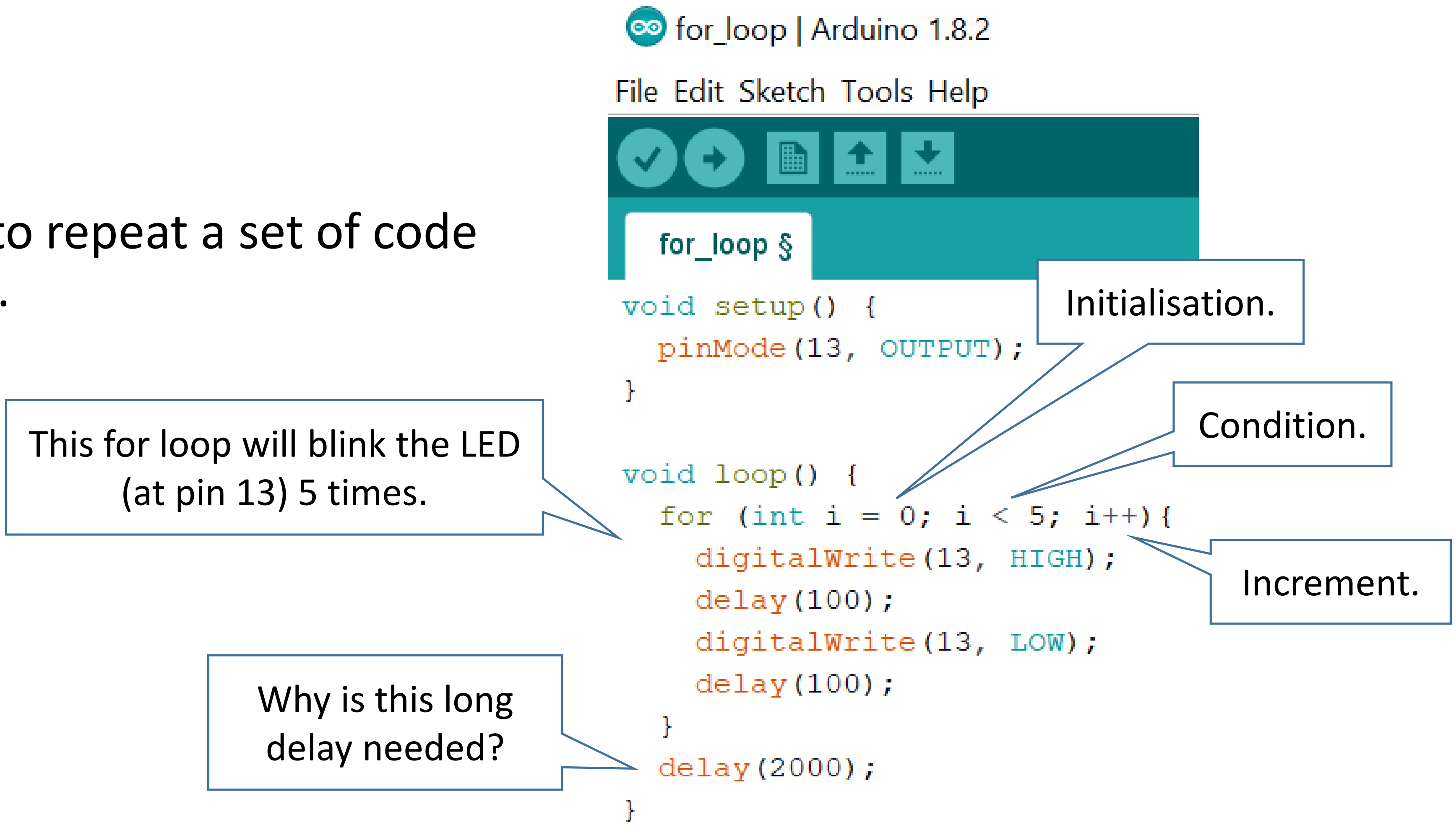
In some cases, the "else if" and "else" branches need not be included.



```
void setup() {
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  int val = analogRead(A3); // read brightnes sensor
  if (val > 700) { // very bright, off both LEDs
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
  }
  else if (val >= 300) { // medium brightness, on one LED
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
  }
  else{ // very dim, on both LED
    digitalWrite(12, HIGH);
    digitalWrite(13, HIGH);
  }
  delay(100);
}
```

for loop

- A **for loop** is used to repeat a set of code a number of times.



```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  for (int i = 0; i < 5; i++){
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);
  }
  delay(2000);
}
```

Initialisation.

Condition.

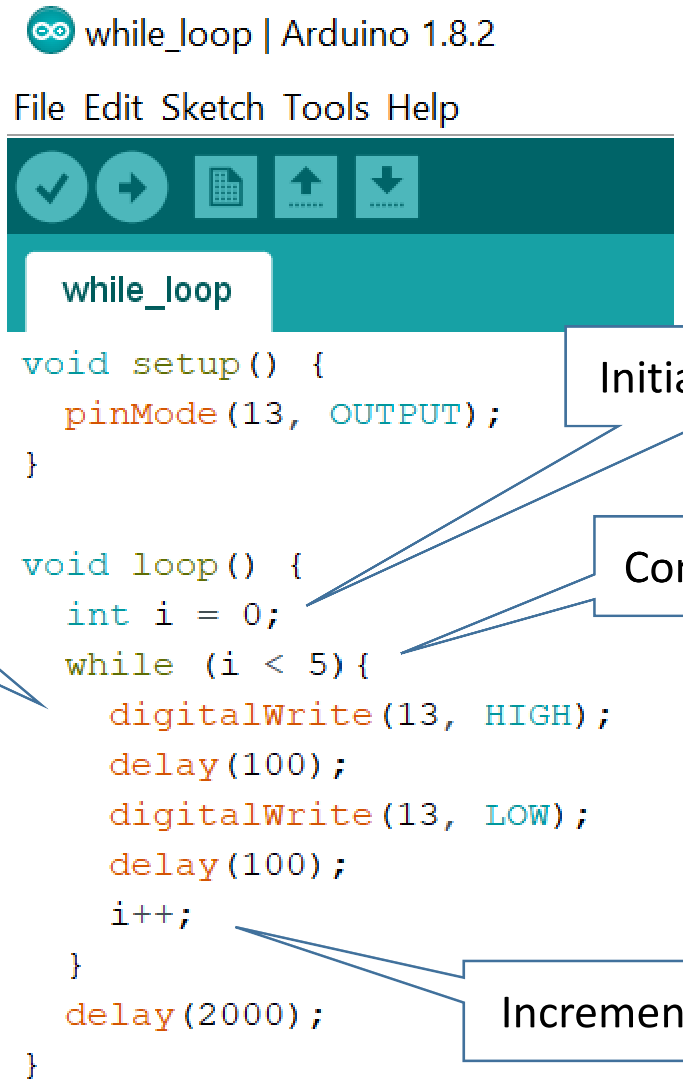
Increment.

This for loop will blink the LED (at pin 13) 5 times.

Why is this long delay needed?

while loop

- A **while loop** is also used to repeat a set of code a number of times.



```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  int i = 0;
  while (i < 5){
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);
    i++;
  }
  delay(2000);
}
```

Initialisation.

Condition.

Increment.

This will behave exactly like the for loop, which blinks the LED (at pin 13) 5 times.

Under what condition do you use for loop? How about while loop?

function

- **Functions** are building blocks of a program.
- Examples of function: setup and loop.

A function to blink the LED (at pin 13) "count" times.

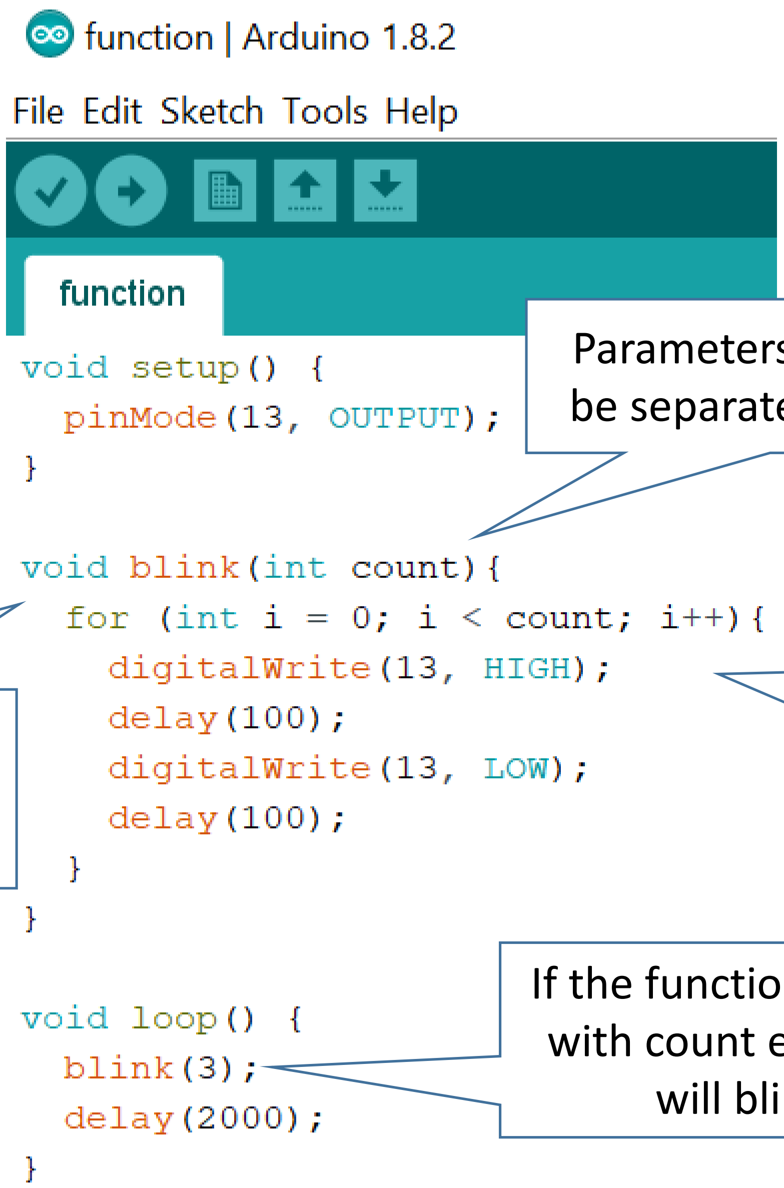
When a function has no return statement (such as return answer;), the return type is void.

Why should function be written?

Parameters passed in can be separated by commas.

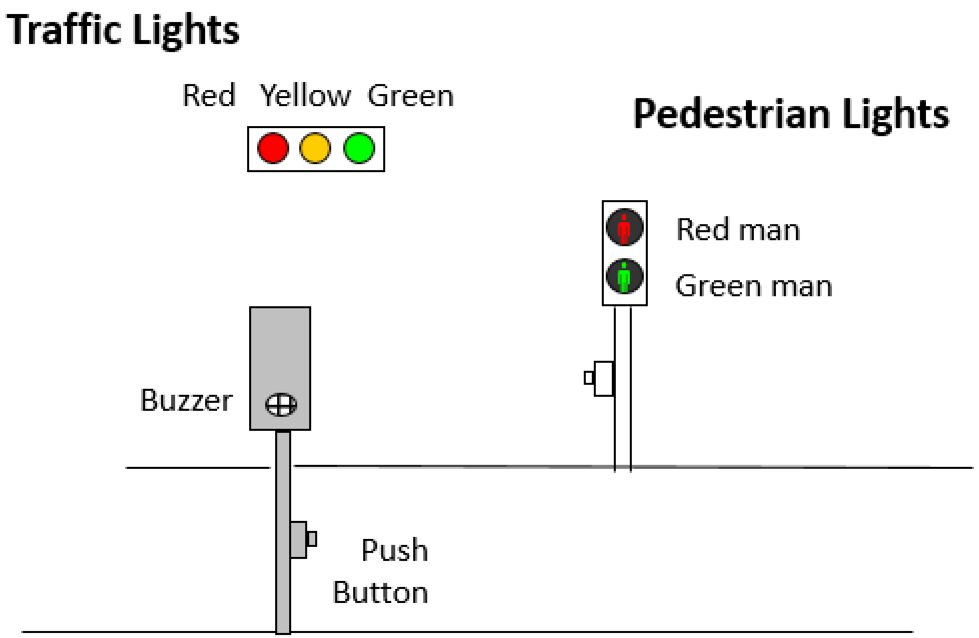
The function can be placed anywhere in the sketch.

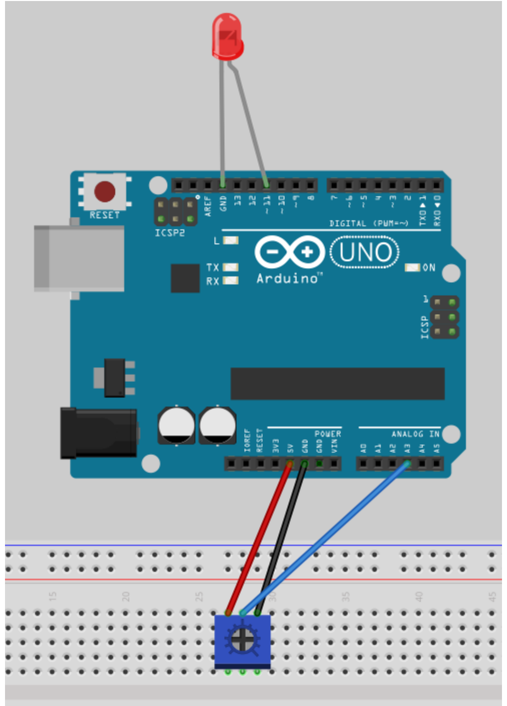
If the function "blink" is called with count equals 3, the LED will blink 3 times.



```
void setup() {
  pinMode(13, OUTPUT);
}

void blink(int count){
  for (int i = 0; i < count; i++){
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);
  }
}

void loop() {
  blink(3);
  delay(2000);
}
```

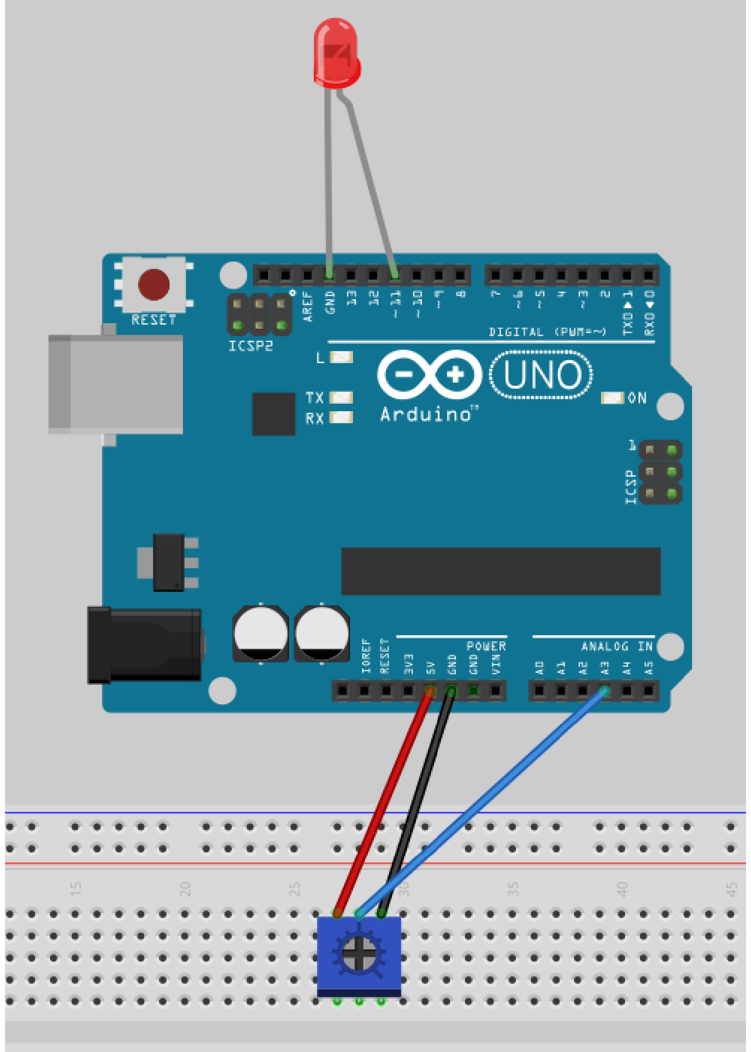
Lab Exercises



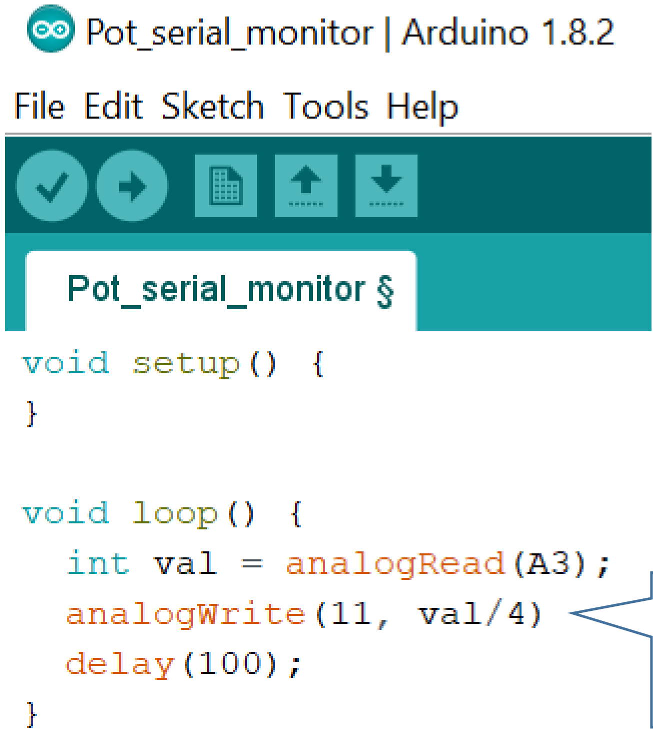


- Exercise 2.1 – Dimmer lamp
- Exercise 2.2 – Two sets of lamp, 3 levels of brightness
- Exercise 2.3 – OMG! Someone needs help!
- Exercise 2.4 – Traffic light controller

Exercise 2.1 – Dimmer lamp



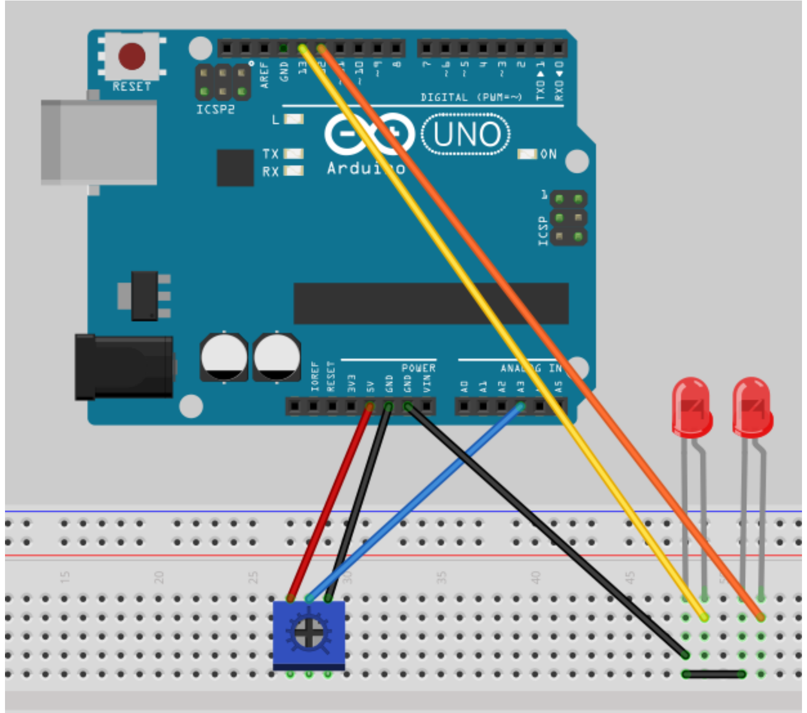
Connect an LED to pin 11, and a potentiometer to pin A3. Modify any existing sketch into the Dimmer_lamp sketch below and run it. What happens when the potentiometer is turned?


Pot_serial_monitor | Arduino 1.8.2
File Edit Sketch Tools Help
Pot_serial_monitor §

```
void setup() {
}

void loop() {
  int val = analogRead(A3);
  analogWrite(11, val/4);
  delay(100);
}
```

Why is there a need to divide by 4?

Exercise 2.2 – Two sets of lamp, 3 levels of brightness



Connect two LEDs to pins 12 & 13, and a potentiometer to pin A3.

The potentiometer is used to emulate a brightness sensor and the LEDs will be turned on or off, depending on the brightness level:

- If it is very bright i.e. analogRead of A3 gives value above 700, both LEDs are turned off.
- If A3 gives value between 300 and 699, only one LED is turned on.
- If it is very dim i.e. A3 gives value below 300, both LEDs are turned on.

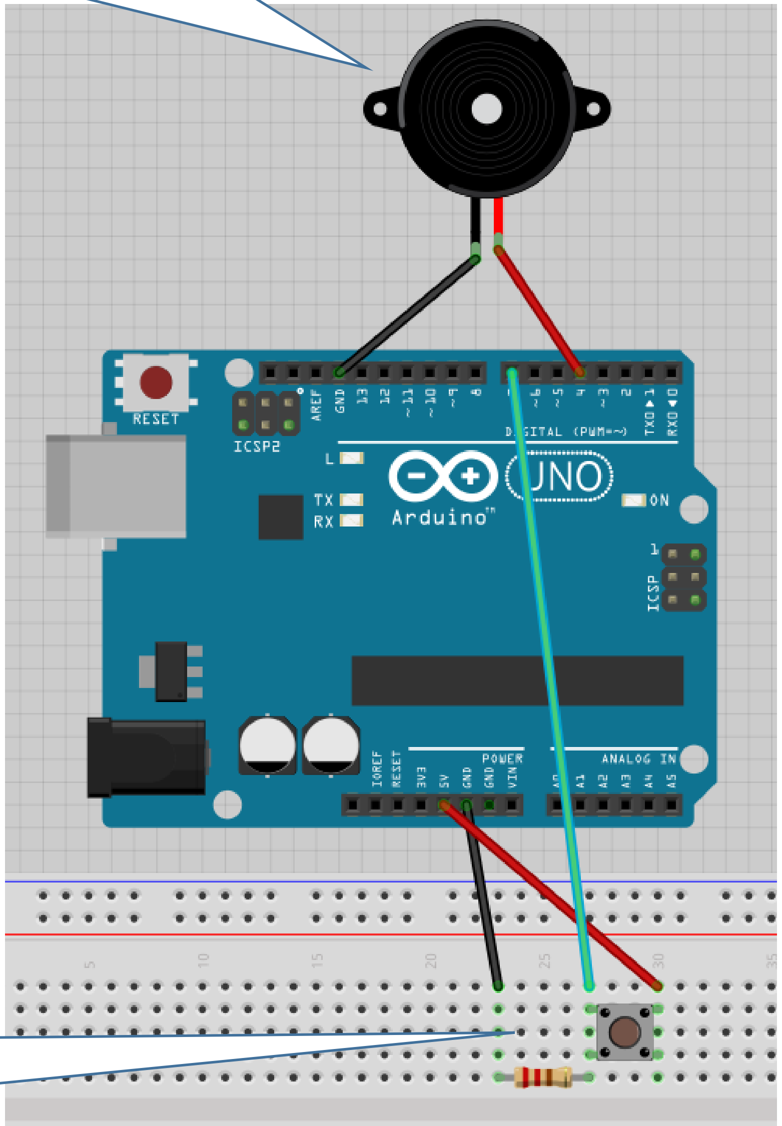
Write the sketch required using if-else.

Exercise 2.3 – OMG! Someone needs help!

Connect a buzzer to pin 4, and a push button switch to pin 7.

If the push button is pressed, the buzzer will beep 3 times. Otherwise, it will be turned off.

Write the sketch required. Hint: use if & a for / while loop.



Long leg to pin 4, short leg to GND.

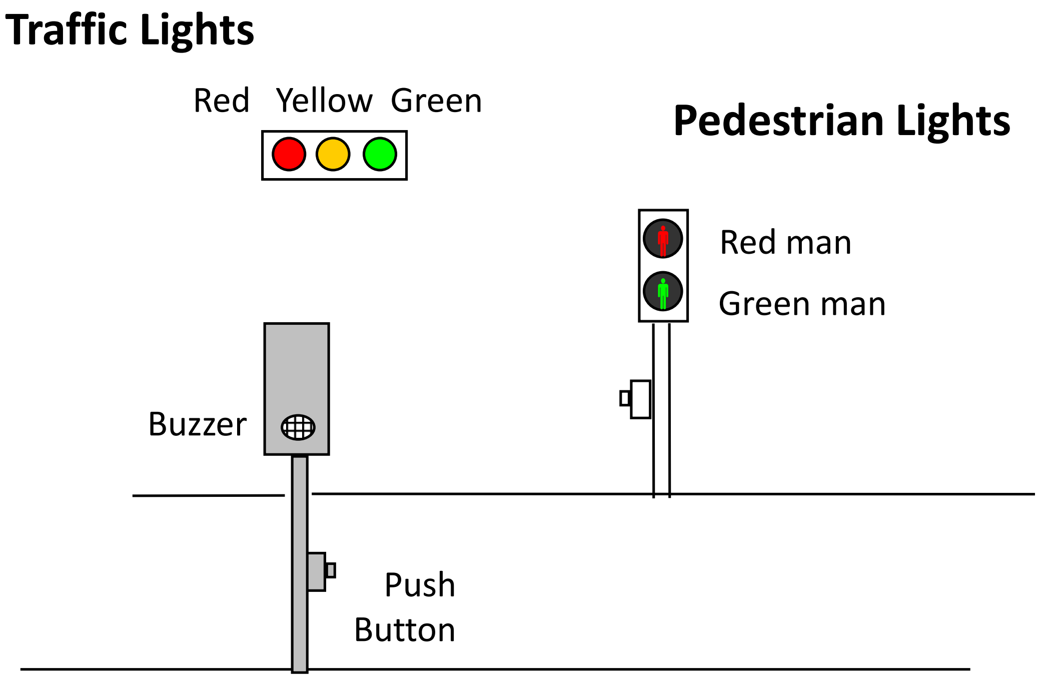
Ask you lecturer to explain how a push button switch can be connected.

Exercise 2.4 – Traffic light controller

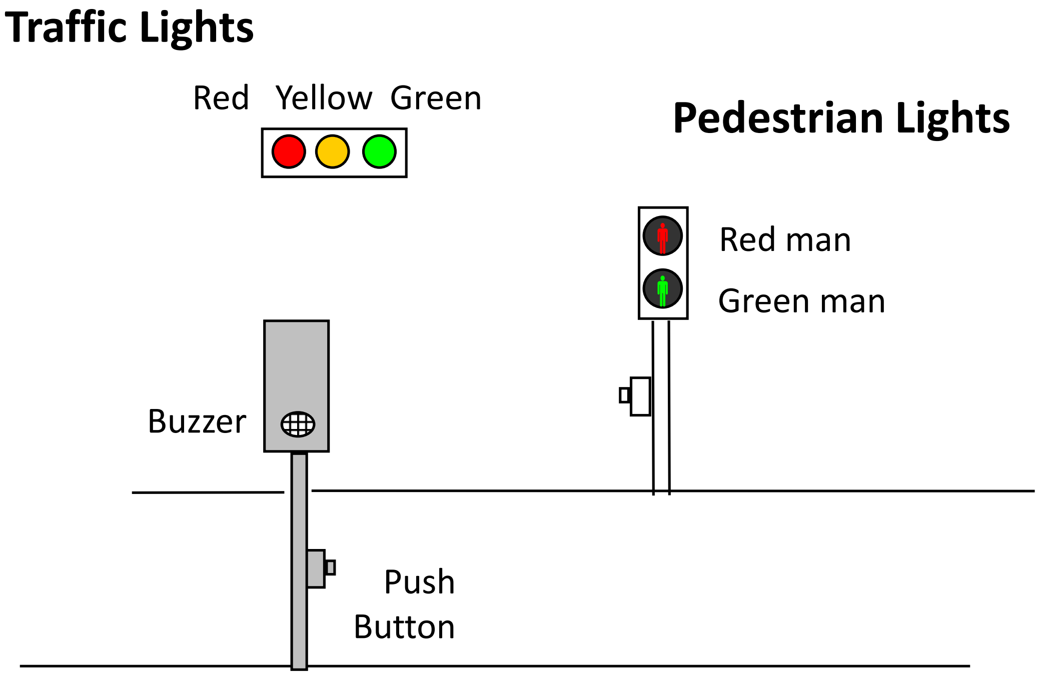
Connect five LEDs (2 red, 1 yellow, 2 green), a buzzer and a push button switch to 7 pins of an Arduino UNO.

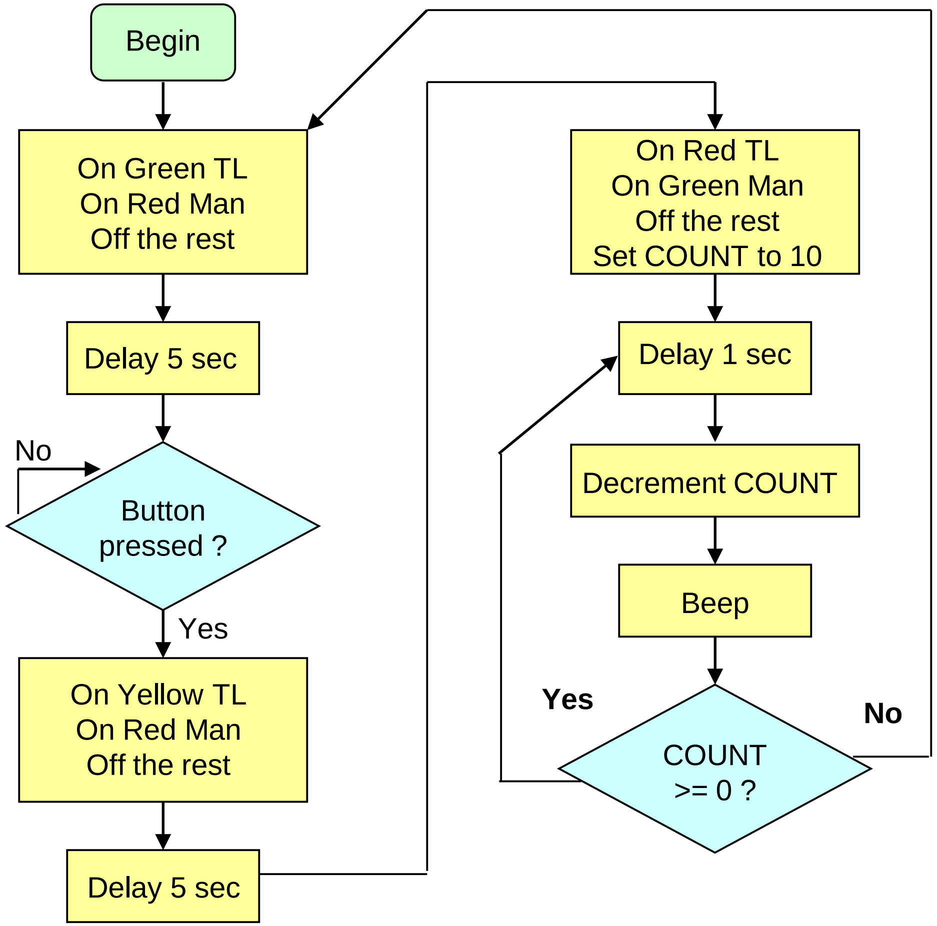
Write a sketch to control the traffic & pedestrian lights.

You can follow the flowchart on the next page when writing your sketch.



Exercise 2.4 – Traffic light controller (cont.)





Thank You