



# Arduino Lesson ( )

## Introduction to Arduino - LEDs

Name: \_\_\_\_\_ ( ) Class: \_\_\_\_\_ Date: \_\_\_\_\_

**Objectives:** At the end of this lesson, you should be able to

1. Understand what is an Arduino microcontroller
2. Apply basic circuitry to the Arduino microcontroller and other basic hardware
3. Write a basic programme using the Arduino software (IDE)

### Apparatus:

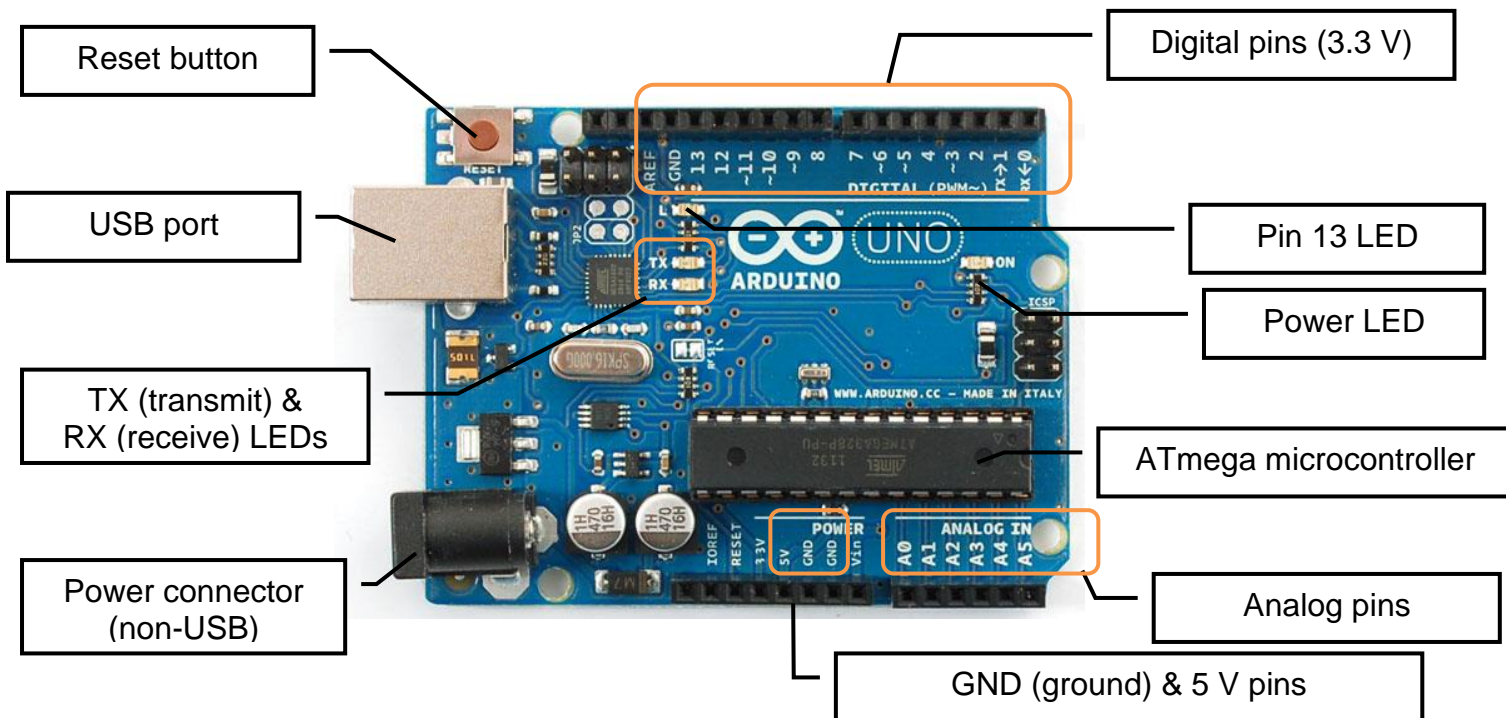
- 01 Arduino UNO microcontroller
- 01 USB cable
- 03 LEDs (red, green, yellow)
- 01 breadboard
- 06 jumper wires
- 03 resistors (1 kΩ)

### 1. About Arduino

- Main Arduino website: <https://www.arduino.cc/>
- History: 2005 in Italy
- Hardware: **Arduino microcontroller** to “receive information” (input) usually from sensors and “do something” (output) via other sensors or devices.
- Software: **sketch** (program) to tell the hardware what to do (involves C++ programming)

### 2. Getting to know Arduino board (hardware)

- **Parts** of Arduino board: refer details at <https://www.arduino.cc/en/Guide/BoardAnatomy>



### 3. Getting to know the Arduino software

- Install the Arduino software from <https://www.arduino.cc/en/Main/Software>

#### 3.1 Powering up the Arduino board (connecting to software)

- Plug the Arduino board into the USB port on the laptop.
- The onboard LED (pin 13) should blink regularly. This is because the *Blink* sketch (or program file) was already installed earlier by the manufacturer to verify that the board works.
- Click the Reset button. The LED should flicker momentarily.

#### 3.2 Arduino software

- Double click on the Arduino icon to start the software. The “sketch” will open in a separate window.
- Specify the type of Arduino board: from **Tools** menu > **Board** > Arduino Uno
- Specify the serial port connected to the Arduino board: the **Tools** menu > **Port**

**Note:** If the Arduino software could not find the serial port, open the Control Panel, select option to view Icons, and find Device Manager and click on 'Ports'

- Go to **File** > **Examples** > **01.Basics** > open the sketch *Blink*.

```
/*
Blink
Turns 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, it is attached to
digital pin 13. LED_BUILTIN is set to the correct LED pin independent of which board
is used. Also check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products
modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/
// 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
}
```

Multi-line comments start with /\* and end with \*/:  
e.g. shown: Name of sketch and description of what it does

Single line comments start with //

The setup routine

The loop routine

**Note:**

- The sketch (program) has key sections in its structure.
- The “comments” are not part of the sketch but are important part of the “**documentation**” to explain how a sketch works.
- It is important to type the characters with the **correct uppercase/lowercase, spacings & punctuations**.
- In delay(1000), “1000” means 1000 ms (milliseconds) = 1 s (second).

**Assignment 1:** Modify Blink

Modify the above **Blink** sketch to double the frequency of the blinking. Also update the comments.

Click  to verify the sketch and click  to upload it to into the Arduino board.

Observe any changes.


**Assignment 2:** Modify Blink

Write another sketch (modify previous version and **Save As** with new sketch name) to make the LED

- blink quickly twice and then
- blink slowly twice and repeats.

Use copy and paste to repeat instructions.

Verify and upload the new sketch and observe any changes.

Note: You may click  to both verify & upload the sketch.

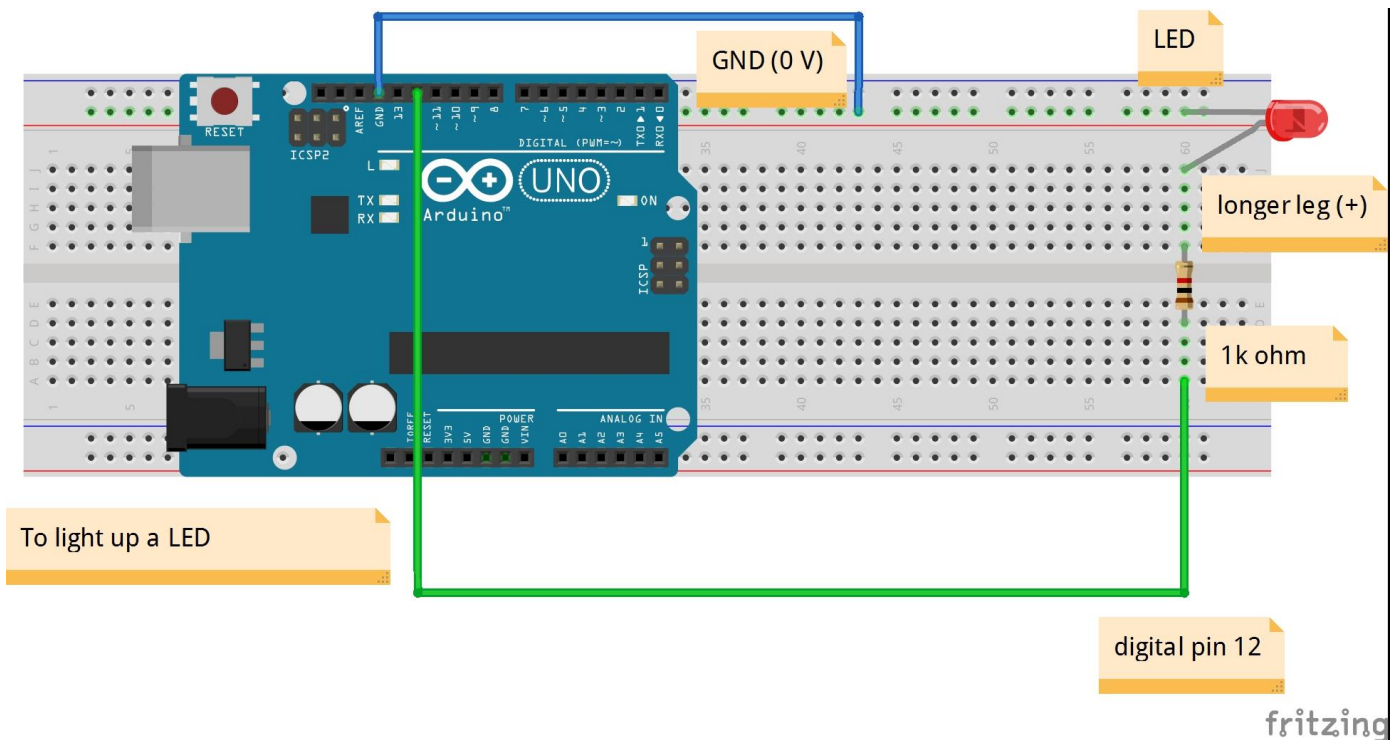
**Note:** The sketches are usually saved in Arduino folder under Libraries > Documents > “My Documents”.

#### 4. More about hardware

- Arduino microcontroller with CPU, RAM, ROM, inputs and outputs
- Breadboard
- Input devices: e.g. light dependent resistor (LDR), button
- Output devices: e.g. buzzer, light emitting diode (LEDs), organic light emitting diode (OLED)
- Basic circuitry
- Prototyping & soldering

#### 5. More about software

- Software: Integrated Development Environment (IDE) to create programs called sketches
- Sketches: e.g. Blink
- Programming structures: e.g. while-loop
- Libraries



### **Assignment 3:** Lighting up an LED

Connect the above circuit and write a sketch to light up an external red LED (max. 3.5 V).

- Draw the circuit diagram for the simple LED circuit.
- Connect the red LED in series with the resistor to GND (0 V) and pin 12 (to +ve terminal of LED).
- Modify the earlier **Blink** sketch by initialize digital pin 12 as an output (instead of pin 13) and change all pins (from 13) to 12:
 

```
pinMode(12, OUTPUT);
digitalWrite (12, HIGH);      or      digitalWrite (12, 1);
```
- Verify and upload the sketch and observe any changes.
- Also save this sketch using **Save As** with a new sketch name

#### **Assignment 4:** Traffic Light

Connect a circuit and write a sketch to control 2 LEDs to function as a simple traffic light.  
Draw the circuit diagram for two-LEDs circuit.

Write a sketch to control the LEDs such that:

- Green LED on for 3 seconds & Red LED off.
- Green LED blinks 2 times at 0.5 Hz.
- Green LED off & Red LED on for 4 seconds.
- Repeat.

Initialize digital pin 11 as an output for Green LED & pin 12 as an output for Red LED.

Verify and upload the sketch and observe any changes.

**Challenge:** add Yellow LED to create a “full” traffic light using pin 10.

**Note:** Frequency of blinking: e.g. 0.5 Hz =  $\frac{1}{2}$  Hz = 1 blinks / 2 s, so use a time delay of 2 s or 2000 ms.

#### **Further Readings:**

- Langbridge, J. A. (2015). Arduino Sketches: Tools and Techniques for Programming Wizardry.
- Schmidt, M. (2011). *Arduino*. Pragmatic Bookshelf.
- Website <https://www.arduino.cc>

#### **Acknowledgements**

- Materials adapted from 2015 NYGH Sec 2 PC Module on “*Coding & Electronics*” by Ms Wong-Tan Poh Yee, Nanyang Girls’ High School
- Materials adapted from 2015 Nov Teachers’ workshop on “*Introduction to Microcontrollers*” by Mr Wendell Wong, CRADLE, Singapore Science Centre

## Sample sketches

### Assignment 1: Modify Blink

Changes made:

*frequency = 1/ period*

*To double the frequency of the blinking, means to halve the period (time between blinks).*

*Reduce the time interval for “high” & “low” by half, from 1000 ms to 500 ms*

### Assignment 2: Modify Blink

Changes made:

- *copy and paste the void loop segment “blink” twice*
- *to blink slowly twice, double the time interval for “high” & “low” , from 1000 ms to 2000 ms*

### Assignment 3: Lighting up an LED

```
void setup() {
// initialize digital pin 12 as an output.
pinMode(12, OUTPUT);
}

// the loop function runs over and over again forever

void loop() {
digitalWrite(12, HIGH); // turn the LED on (HIGH is
the voltage level)
delay(1000);           // wait for a second
digitalWrite(12, LOW); // turn the LED off by
making the voltage LOW
delay(1000);           // wait for a second (1 s = 1000
ms)
}
```

### Assignment 4: Traffic light

```
void setup() {
// put your setup code here, to run once:
pinMode (11, OUTPUT); // green
pinMode (12, OUTPUT); // red
}

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

digitalWrite (11, 1); // green
digitalWrite (12, 0); // red
delay (3000);
digitalWrite (11, 0);
delay (500);

digitalWrite (11, 1);
delay (1000);
digitalWrite (11, 0);
delay (500);

digitalWrite (11, 1);
delay (1000);
digitalWrite (11, 0);

digitalWrite (11, 0);
digitalWrite (12, 1);
delay (4000);
digitalWrite (12, 0);
}
```