

## ACTIVITIES WITH ARDUINO - Worksheet 1

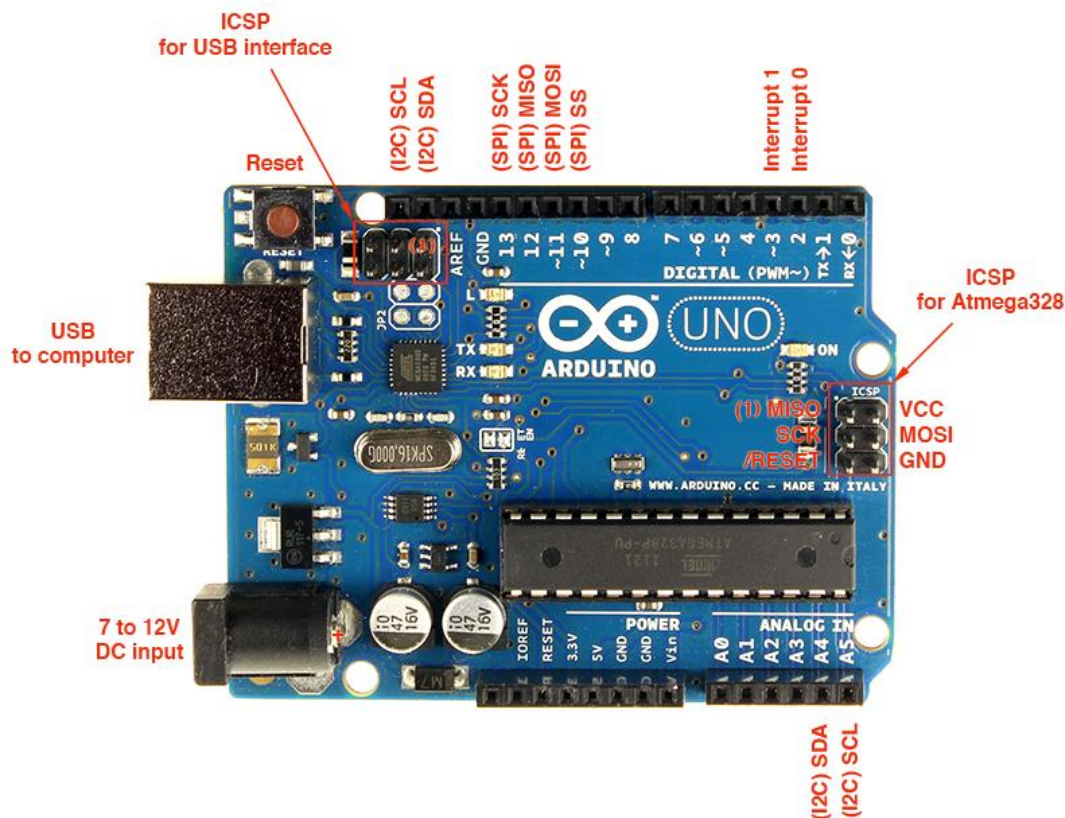
### Introduction to Arduino

#### Learning outcomes

- ✓ Arduino microcontroller
- ✓ Use of breadboard
- ✓ Materials: resistors, LED lights, jumper wires
- ✓ Basic structure of the program
- ✓ Arduino Digital output

### Activity 1 – Become familiar with the basic equipment and materials

#### Arduino UNO microcontroller

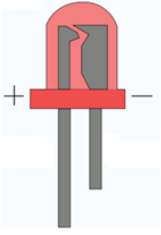


#### Sources

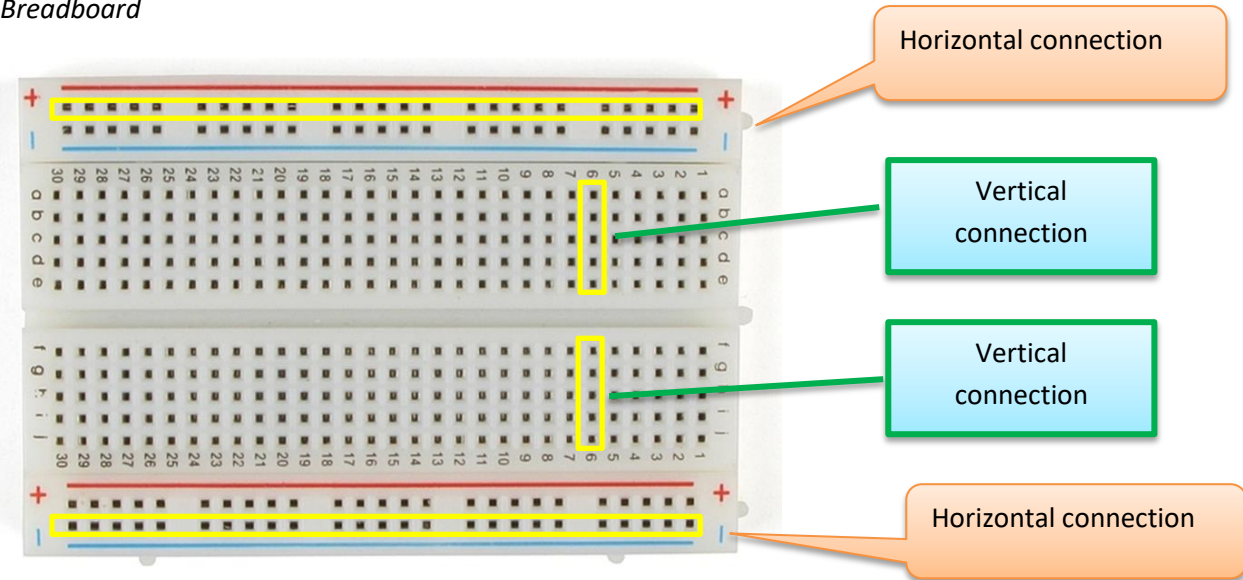
<http://learning.grobotronics.com/el/getting-started/arduino-uno/>

<https://github.com/ellak-monades-aristeias/CruiseCabinDigitalSignage/blob/master/Hardware/ArduinoUNO.md>

## LED light



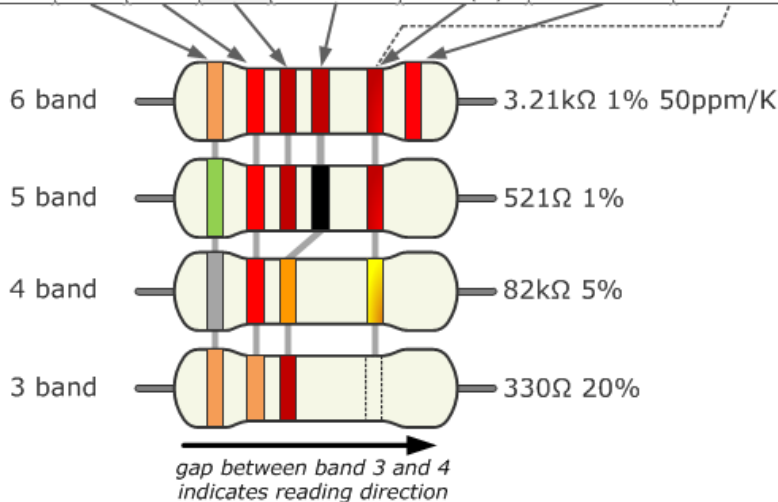
## Breadboard



## Resistor

www.resistorguide.com

	Color	Significant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		

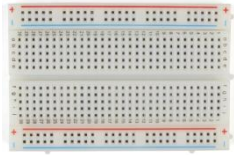


## Activity 2 – Create your first circuit

In this activity, we will create a circuit with an LED light that blinks every second.

Apart from the microcontroller Arduino UNO, we will need:

- 1 x breadboard



- 1 x LED light



- 1 x 220 Ohm resistor



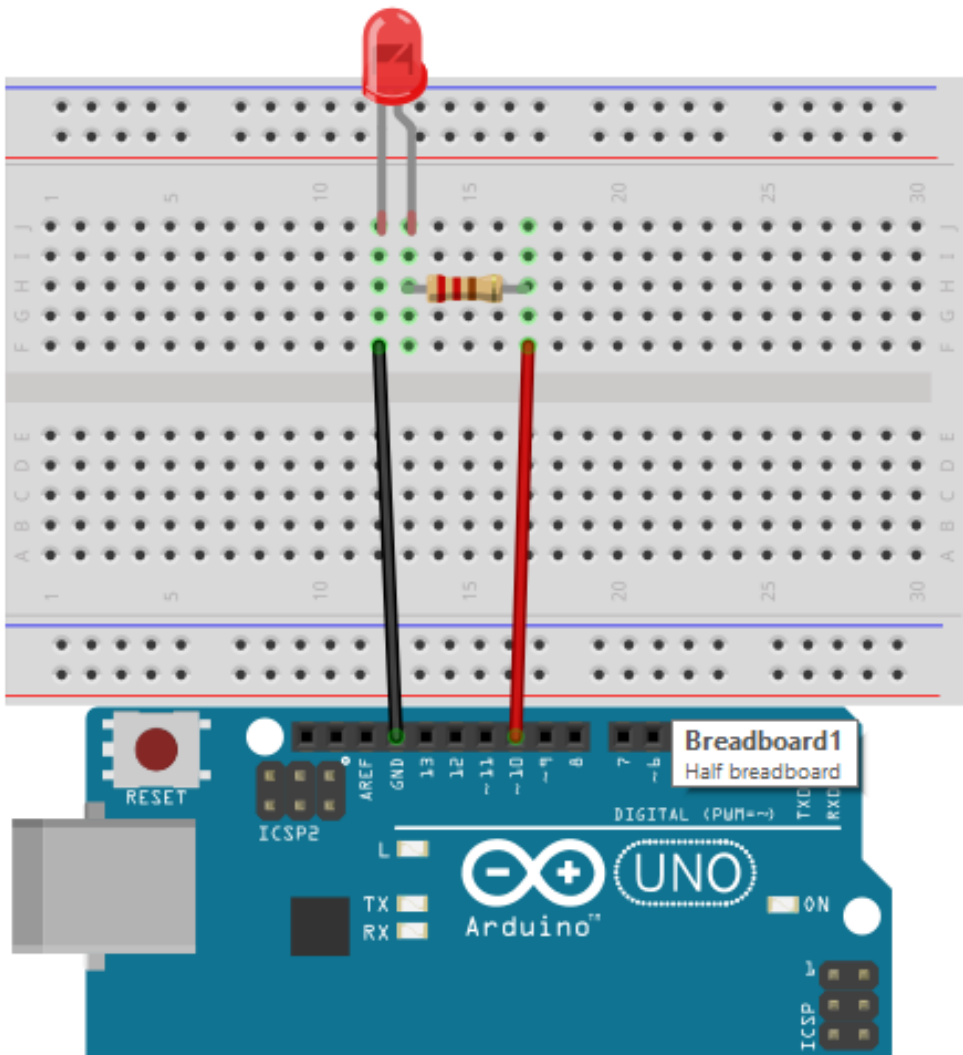
- 2 x jumper wires male-male



Find the materials in your box and place them in front of you on your desk.

### Circuit

Next, create the following circuit.



## Programming environment

For the programming part, you will use the Arduino IDE programming environment, which is already installed on your computer. Open the Arduino IDE.



## Basic structure of the program

A screenshot of the Arduino IDE interface. The window title is "sketch\_jan27a | Arduino 1.6.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar shows icons for opening, saving, and running. The main text area contains the following code:

```
1 void setup() {  
2  
3  
4 }  
5  
6 void loop() {  
7  
8  
9 }
```

The status bar at the bottom indicates "3" and "Arduino/Genuino Uno on COM3".

It is executed once when the program starts (when power is supplied) or after the Arduino board is reset.

It is executed repeatedly.

## Commands

- Set the digital pin as output:  
**pinMode(pin\_number, OUTPUT);**
- Set the state of the digital pin:  
**digitalWrite(pin\_number, state);**  
(where **state** → HIGH ḡ LOW)
- Time delay, wait:  
**delay(number);**  
(where **number** → number in milliseconds)

## Code

Write the following code:

```
void setup() {  
  pinMode(10, OUTPUT);  
}  
void loop() {  
  digitalWrite(10, HIGH);  
  delay(1000);  
  digitalWrite(10, LOW);  
  delay(1000);  
}
```

Connect the Arduino microcontroller to your computer using the USB cable. From the Tools > Board menu, make sure the board is set to Arduino Uno. From the Tools > Port menu, select the port for your board, e.g., COM4 (Arduino Uno).

Validate the code by clicking the button ✓ and save the file. Upload the code to arduino microcontroller by clicking the button ⇨ .

### Activity 3 – Experiment your circuit

In the circuit from Activity 2, also try the following:

- Program the LED light to blink every 0.5 seconds.
- Program the LED light to stay on for 2 seconds and off for 1 second.
- Connect an additional LED that will blink (i) together with the other LED, and (ii) alternately with the other LED.

### Activity 4 – Problem solving

In this activity, you are asked to apply the knowledge you have gained so far and create a circuit that simulates a traffic light. The circuit will have three LEDs in green, orange, and red. The color changes occur just like in a traffic light (from green to orange, from orange to red, and from red back to green). You can freely choose the duration (in seconds) that each LED will stay on.



### Activity 5 – Expand your circuit

Based on the circuit you created in the previous activity, add an additional traffic light to simulate an intersection (crossroad). That is, when one traffic light is green, the other should be red. You can freely choose the duration (in seconds) for each color on each traffic light, as long as it does not cause vehicle collisions!

## ACTIVITIES WITH ARDUINO - Worksheet 2

### Automations with Arduino: Buzzer

#### Learning outcomes

- ✓ Material: Buzzer
- ✓ Create a simple circuit using a buzzer
- ✓ Program and create a short tune/melody

#### What is a buzzer?



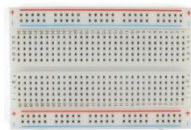
Buzzer

- A simple electronic device that produces sound.
- It has 2 pins.
- When you apply voltage to the pins, the buzzer produces sound.
- You can change the frequency of the sound depending on the time that you apply the voltage.

#### Activity 1 – Create a simple circuit using a buzzer

Apart from the Arduino UNO microcontroller, we will need:

- 1 x breadboard



- 1 x LED light



- 1 x 220 Ohm resistor



- 1x Buzzer



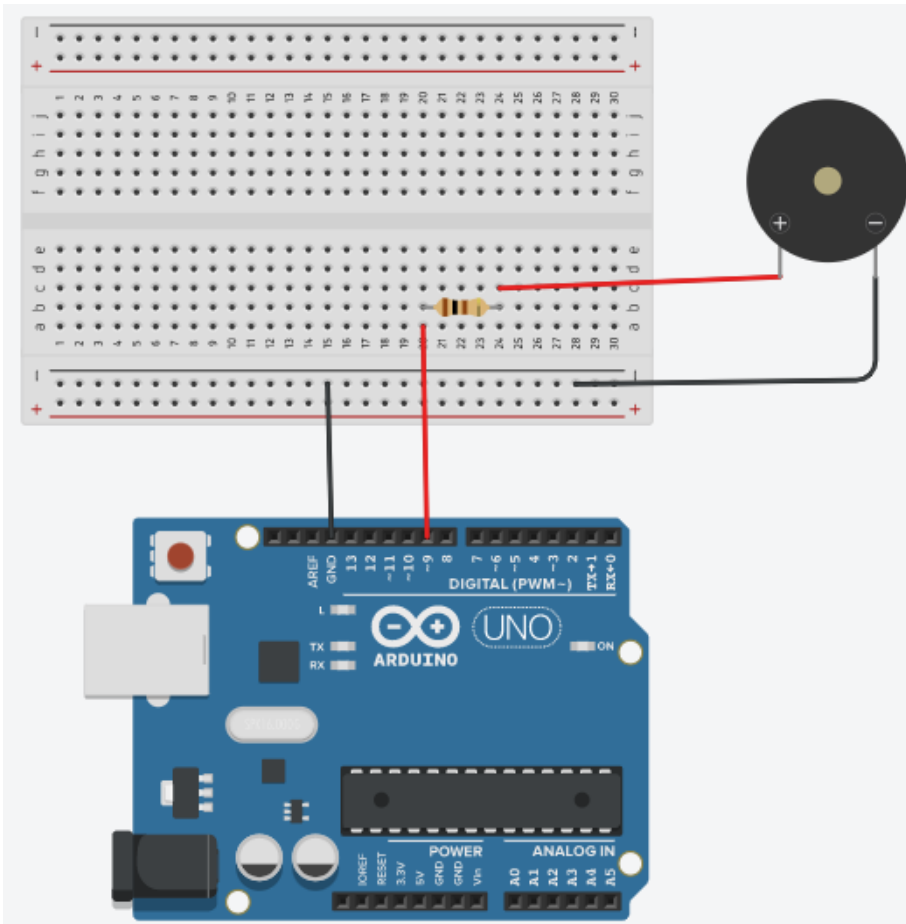
- 2 x jumper wires male-male



Find the materials in your box and place them in front of you on your desk.

## Circuit

Next, create the following circuit.



## Commands

- **Activate buzzer**  
`tone(pin_number, frequency)`  
where: `pin_number` the pin where you connected the buzzer  
and `frequency` the frequency in Hertz
- **Deactivate buzzer**  
`noTone(pin_number)`  
where: `pin_number` the pin where you connected the buzzer

## Code

```
void setup() {  
  pinMode(9, OUTPUT);  
}  
void loop() {  
  tone(9, 500);  
  delay(200);  
  noTone(9);  
  delay(200);  
}
```

## Activity 2

Experiment on the frequencies in your code, to make a short melody.

Note	Frequency in Hz
C	261
D	293
E	329
F	349
G	392

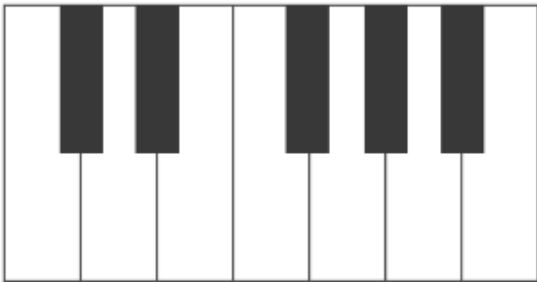
You can also use the command:

```
tone(pin, frequency, duration)
```

where: pin the pin where you connected the buzzer

frequency the frequency we want the buzzer to produce in Hertz

duration the duration we want the buzzer to produce the frequency in milliseconds



TONE		FREQUENCY (in hertz)		DELAY VALUE
Do (C)	→	261.63	→	1915 $\mu$ s
Re (D)	→	293.66	→	1700 $\mu$ s
Mi (E)	→	329.63	→	1519 $\mu$ s
Fa (F)	→	349.23	→	1432 $\mu$ s
Sol (G)	→	392.00	→	1275 $\mu$ s
La (A)	→	440.00	→	1136 $\mu$ s
Si (B)	→	493.88	→	1014 $\mu$ s

# Musical Notes / Frequencies

Note	Frequency (Hz)
C <sub>4</sub>	261
C <sup>#</sup> <sub>4</sub> /D <sup>b</sup> <sub>4</sub>	277
D <sub>4</sub>	293
D <sup>#</sup> <sub>4</sub> /E <sup>b</sup> <sub>4</sub>	311
E <sub>4</sub>	329
F <sub>4</sub>	349
F <sup>#</sup> <sub>4</sub> /G <sup>b</sup> <sub>4</sub>	369
G <sub>4</sub>	392
G <sup>#</sup> <sub>4</sub> /A <sup>b</sup> <sub>4</sub>	415
A <sub>4</sub>	440
A <sup>#</sup> <sub>4</sub> /B <sup>b</sup> <sub>4</sub>	466
B <sub>4</sub>	493

Note	Frequency (Hz)
C <sub>5</sub>	523
C <sup>#</sup> <sub>5</sub> /D <sup>b</sup> <sub>5</sub>	554
D <sub>5</sub>	587
D <sup>#</sup> <sub>5</sub> /E <sup>b</sup> <sub>5</sub>	622
E <sub>5</sub>	659
F <sub>5</sub>	698
F <sup>#</sup> <sub>5</sub> /G <sup>b</sup> <sub>5</sub>	739
G <sub>5</sub>	783
G <sup>#</sup> <sub>5</sub> /A <sup>b</sup> <sub>5</sub>	830
A <sub>5</sub>	880
A <sup>#</sup> <sub>5</sub> /B <sup>b</sup> <sub>5</sub>	932
B <sub>5</sub>	987

The frequencies of basic notes of a piano are as follows:

- c - 261 Hz
- d - 294 Hz
- e - 329 Hz
- f - 349 Hz
- g - 392 Hz
- a - 440 Hz
- b - 493 Hz
- C - 523 Hz

## Sources

<https://www.circuitar.com/projects/controlling-the-buzzer/index.html>

<https://www.instructables.com/How-to-use-a-Buzzer-Arduino-Tutorial/>

<https://klinikrobot.com/arduino-tone-indicator-sound-bunyi-tangga-nada-not-balok-dan-not-angka-menggunakan-software-mblock/>

<https://www.instructables.com/Arduino-As-a-Piano/>

## ACTIVITIES WITH ARDUINO - Worksheet 3

### Automations with Arduino: Using the Serial Port

#### *Learning outcomes*

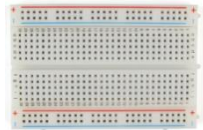
- ✓ Use the serial port of Arduino
- ✓ Input from keyboard
- ✓ Display output on the screen

#### **Activity 1**

In this activity, we will create a circuit with an LED light that we will control through the keyboard.

Apart from the microcontroller Arduino UNO, we will need:

- 1 x breadboard



- 1 x LED light



- 1 x 220 Ohm resistor



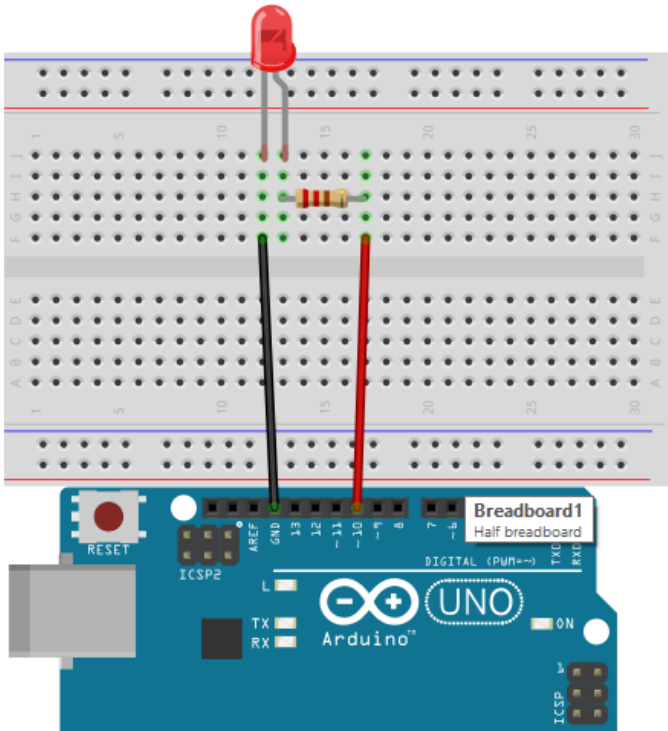
- 2 x jumper wires male-male



Find the materials in your box and place them in front of you on your desk.

#### *Circuit*

Next, create the following circuit.



### Programming environment

For the programming part, you will use the Arduino IDE programming environment, which is already installed on your computer. Open the Arduino IDE. Sets the data rate in bits per second (baud) for serial data transmission



### Commands

- Activate the Serial port:  
**Serial.begin(number);**  
where **number** is the data rate in bits per second (baud) for serial data transmission → 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 or 115200
- Output (no line change):  
**Serial.print('message');**  
where **message** → the message we want to display on the screen
- Output (with line change):  
**Serial.println('message');**
- Input data:  
**Serial.read( );**  
a character from the keyboard is read (input)
- Conditional statement:  
**if (condition)**  
{  
  **commands;**  
}

where **condition** → check a condition if it is true, e.g. `if(x=='a')`

Variations of the `if()` statement:

if	if-else	if-else if	nested if
<pre>if (condition) {   commands; }</pre>	<pre>if (condition) {   commands 1; } else {   commands 2; }</pre>	<pre>if (condition 1) {   commands 1; } else if (condition 2) {   commands 2; } ... else {   commands n; }</pre>	<pre>if (condition 1) {   commands 1;   if (condition 2)   {     commands 2;   }   commands 3; }</pre>

- Variable declaration:

```
char x;
int number = 0;
```

It is mandatory that you declare the variables used in the program. Examples of variable types: **int** for an integer, **char** for a character. Optionally, you can set an initial value for the variable.

*Code*

Control the LED light (1 turns it ON, 2 turns it OFF)

Write the following code:

```
void setup( ) {
  pinMode(10, OUTPUT);
  Serial.begin(9600);
  Serial.println("Press 1 to turn the LED light on and 2 to turn it off.");
}

void loop( ) {
  if (Serial.available()>0) {
    char x = Serial.read(); //declare a character variable
    if(x=='1')
    {
      digitalWrite(10, HIGH);
      Serial.println("LED ON!");
    }
  }
}
```

```

if(x=='2')
  {
    digitalWrite(10, LOW);
    Serial.println("LED OFF!");
  }
}
}

```

Connect the Arduino microcontroller to your computer using the USB cable. From the Tools > Board menu, make sure the board is set to Arduino Uno. From the Tools > Port menu, select the port for your board, e.g., COM4 (Arduino Uno).

Validate the code by clicking the button ✓ and save the file. Upload the code to arduino microcontroller by clicking the button ⇨ .

### Activity 2 – Expand your circuit

In the circuit from Activity 2, also try to connect two more LED lights and control them all from the keyboard.

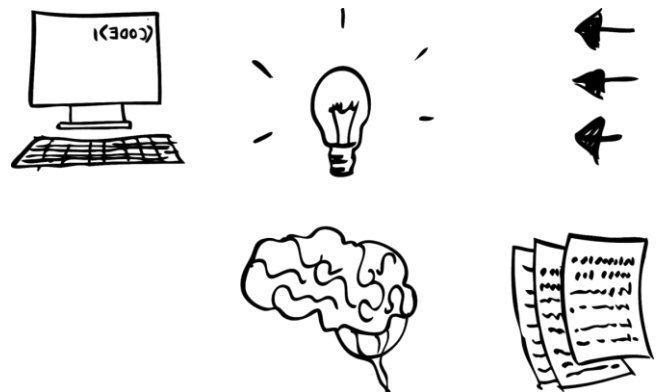
### Activity 3 – Combine it all together

In this activity, you are asked to apply the knowledge you have gained so far and create a circuit that combines everything: LED lights, buzzer and Serial port input/output.

### Activity 4 – Environmental problem solving

Imagine that you work for an IT company that develops cutting-edge automation systems to address environmental problems, ranging from simple (small-scale) to more complex (large-scale) solutions.

Your team is asked to envision the company’s new product that will tackle an environmental issue. Imagine and design a prototype for this system by creating a poster with a short presentation, describing the benefits of using it. You can search for automation materials online (e.g. in e-shops).



Materials examples: (a) motors, (b) sensors: distance, touch, temperature, pressure, gas, light, wind speed, wind direction, rainfall, humidity, soil moisture, colour, sound, movement detection, accelerometer, gyroscope, etc.

You will present your idea in class.