

# ROBOTICS AND CODING

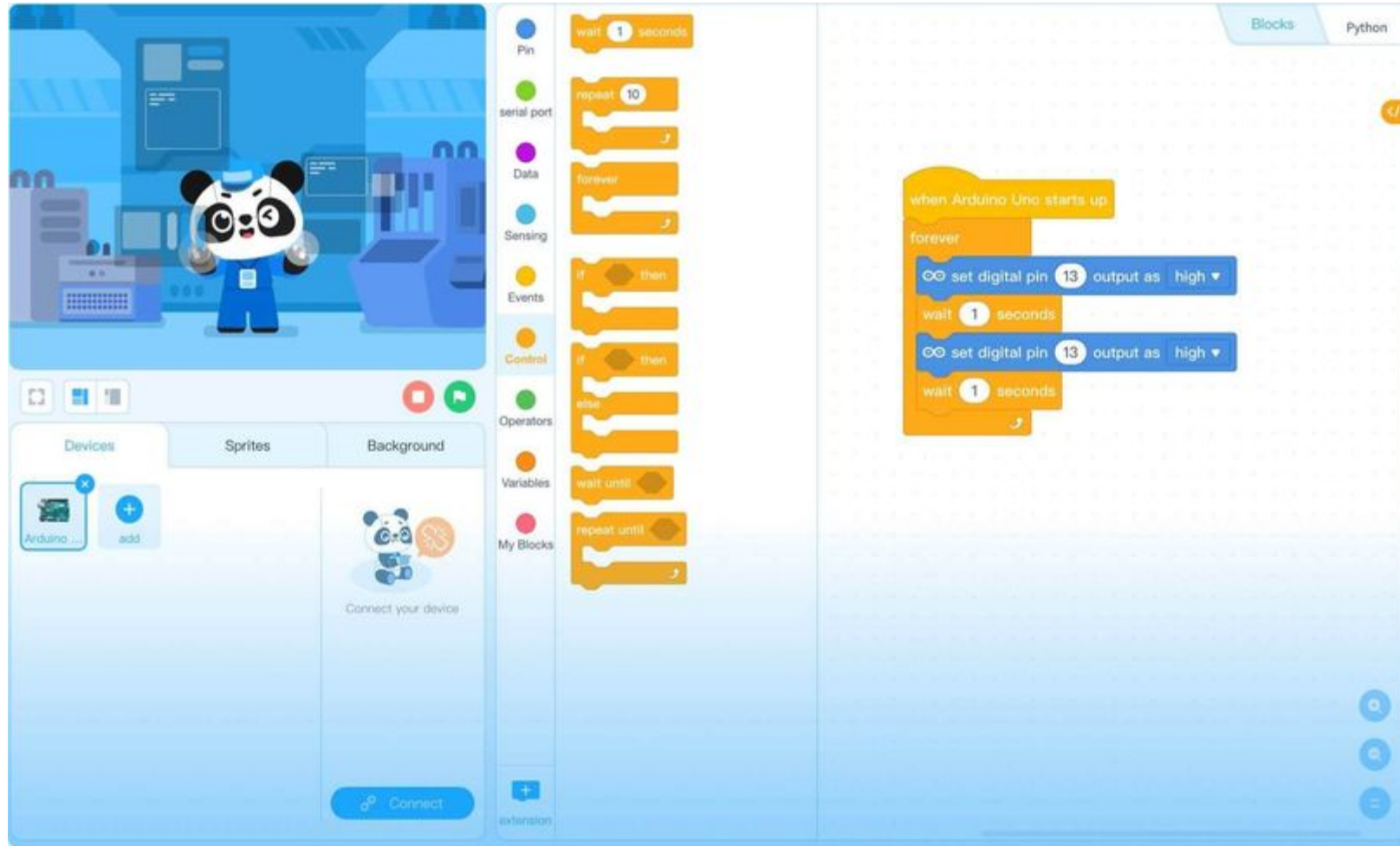
Erasmus Project

PROJECT NUMBER: Project 2020-1-FR01-KA229-079801\_1-ROBOTICS AND CODING



# MAKEBLOCK I MBLOCK

*Make with Code*





## Pedagogical Approach

**Makeblock Education primary school solutions** are designed to build students' interest in STEAM topics, cultivating their skills through project-based learning and collaboration, enhancing their analytical problem-solving skills. Computational thinking continues to be a key element for the development of the four key skills: decomposition, pattern recognition, abstraction and algorithms.



**Makeblock Education secondary school solutions** are designed to develop students' higher-order thinking and coding skills. They include well-equipped teaching resources and elaborate programming projects, as well as fostering the development of critical thinking and problem solving. Students are encouraged to assume the role of innovators and entrepreneurs. The range of products is ideal for hands-on-project and inquiry-based learning with a natural progression into complex science and technology topics.



## Introduction of Mblock Program

### WHAT IS MBLOCK

**mBlock** is a software that is built on the graphical programming language called SCRATCH, which was developed by MIT (Massachusetts Institute of Technology) and tries to provide coding education for children with block-based programming logic, and makes learning coding fun with hardware and software integration.

There are ready-made code blocks in the mBlock program, and we can create code blocks like combining puzzle pieces by dragging these code blocks onto the stage with the mouse. The mBlock program has a compiler and converter that can convert the code blocks in it to C++ language. We can upload the codes created by these code blocks to the Arduino and run the electronic circuits we have prepared, either connected to the computer or independently of the computer.

The biggest difference between the Scratch program is that we can load the codes we designed in mBlock into the Arduino board and use it by feeding the board with external energy. In addition, the program supports many libraries within its own structure.

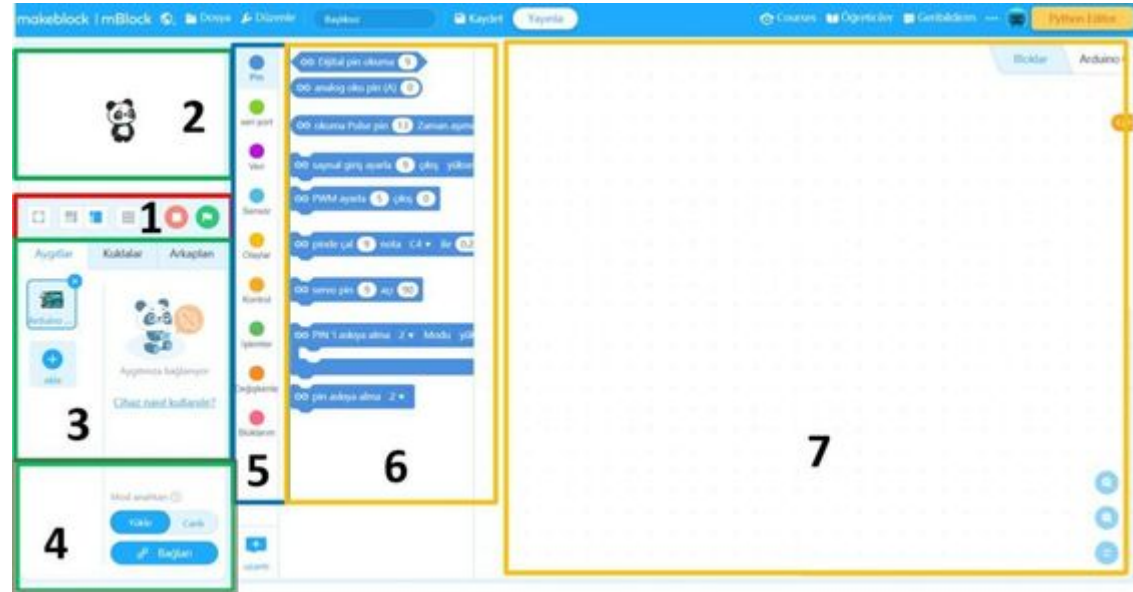
# INTRODUCTION OF THE MBLOCK PROGRAM

This section is the stage. Depending on the codes you have created, we can monitor the movement of the character on the stage from this area.

It is the part where the puppet icon on the stage is located. We can change the puppet by choosing one of the many characters in the mblock library, or we can design a puppet for the stage ourselves. We can also adjust the puppet costume and sound operations in this section. In this section, we can also select the device to be connected to the program.

In this section, when we plug our Arduino device into the computer, we can connect the program to the device, send a code to the device or disconnect it.

In this section, the blocks that we use for programming are classified. There are different blocks in each class.





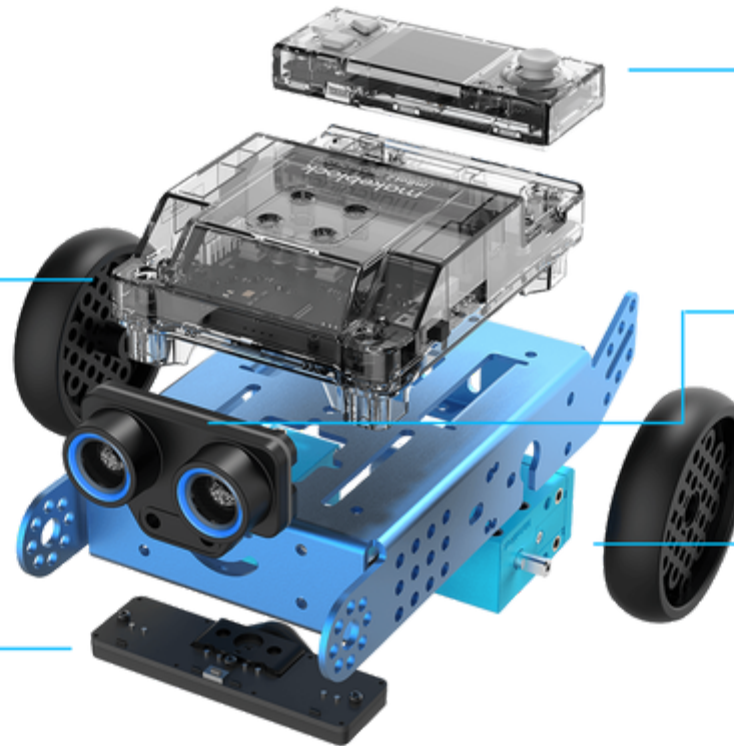
# mBot

## mBot2 shield

compatible with a variety of external components ,and includes a built-in lithium-ion battery.

## Quad RGB sensor

Four sensor probes support color recognition ,as well as basic and advanced line detection programs.



## CyberPi

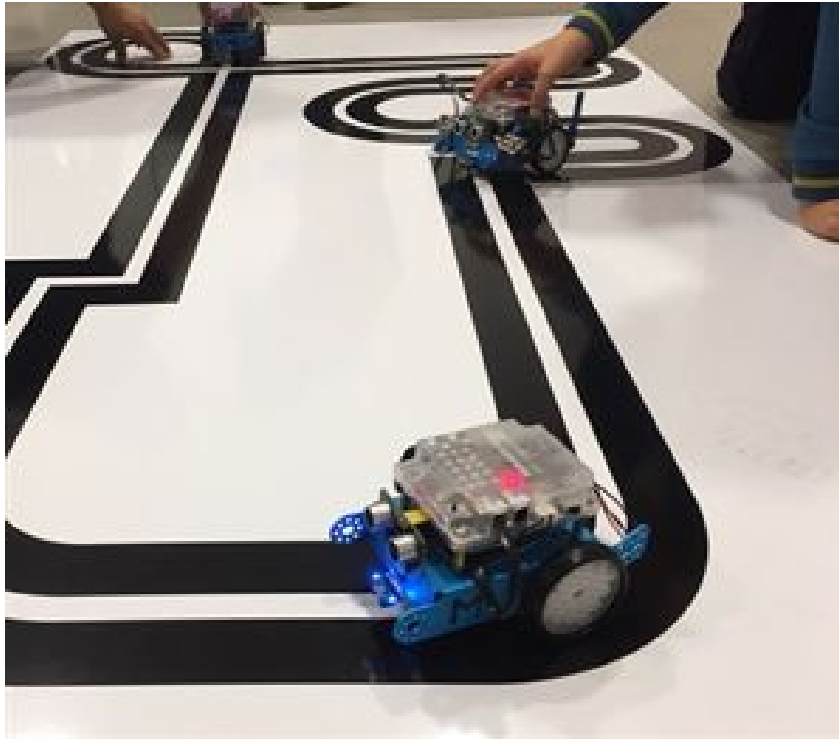
ESP32 microprocessor for wireless communication ,and compatibility with block-based and Python coding.

## Ultrasonic Sensor 2

Object detection is accompanied with 8 programmable LEDs for an enhanced interaction.

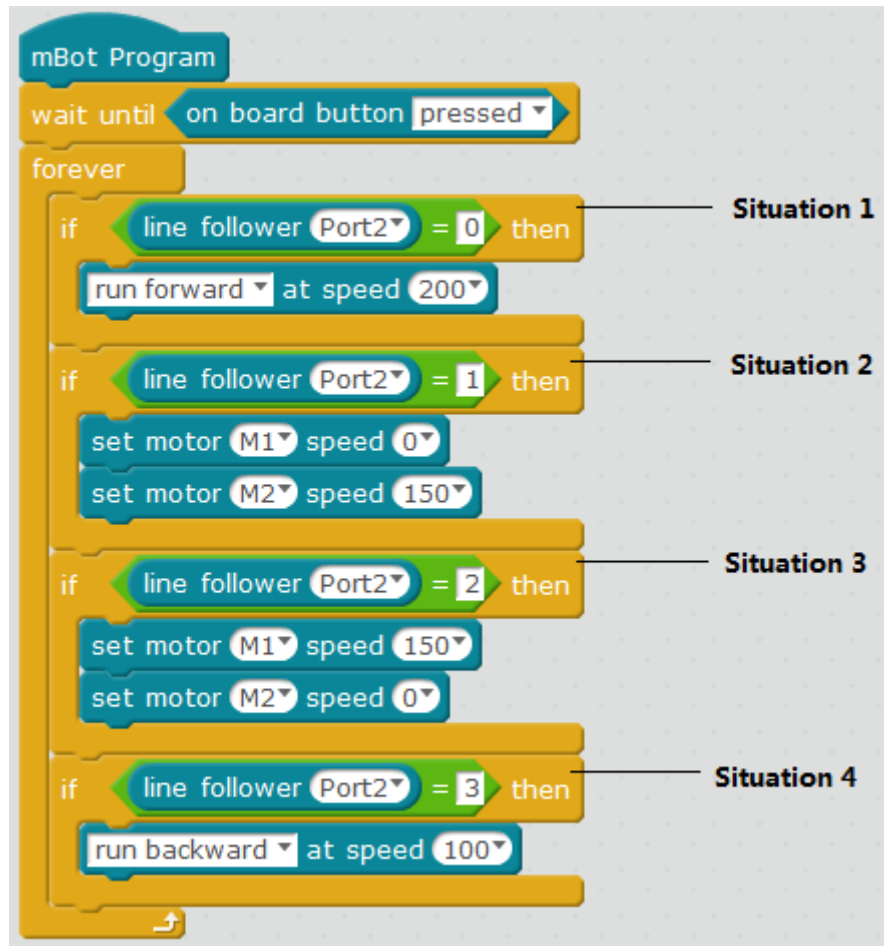
## Encoder Motors

1 degree detection accuracy ,distance traveled, and up to 200 RPM can be precisely controlled.



## Line Follower Robot

The line follower sensor can distinguish black and white colors. When you examine the sensor, you will see that it consists of a combination of 2 sensors. Sensor 1 – The reason why there are 2 sensors called Sensor 2; It is to understand which way it goes from the black colored road. Our mBot will detect sensor 1 if it leaves the road from the right side, and sensor 2 if it leaves from the left side and give a warning.



We created a variable named line and assigned it the value read by the line sensor at gate 2. I have used nested if commands. We made the line go forward if the value it takes is 0 (if both sensors are black), if the line value is 1 (only the left sensor detects black), it turns left, and if the line value is 2 (only the right sensor detects black), it turns right

With these codes, you can easily make your robot move on the black path you will draw.

A new block has been included with the mBlock 3.45 model. Thanks to this block, there is no need for synchronization to values such as 1, 2.



# Arduino Blink (Blinking Led – Blinker)

## CIRCUIT DIAGRAM:

### CIRCUIT SETUP:

Materials:

Arduino Uno

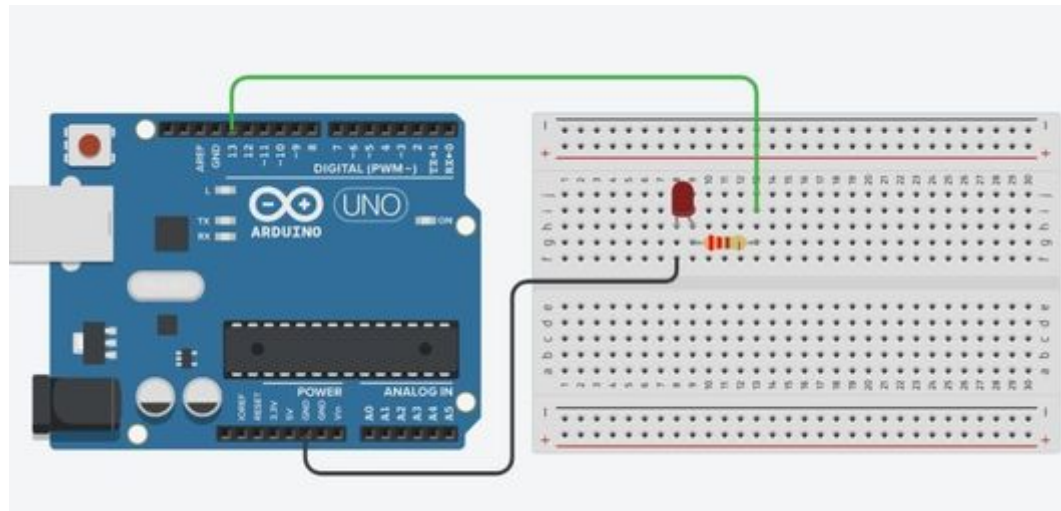
Breadboard

1 pc Led Diode

1 pc 220  $\Omega$  resistor

2 pcs male both ends jumper cable

LED diode is placed on the breadboard. A resistor is connected in series with the LED diode. The purpose of the resistor here is to prevent the led diode from being damaged because the arduino output voltage and current is more than the led diode operating voltage and current. One end of the resistor is connected to the (+) leg (long leg) of the led diode, and one end is connected to the 13th pin of the Arduino via the board. (Here it does not have to be pin 13. We can connect it to any of the pins 0 to 13. It is enough to replace the pin number with the pin number we connected to whichever pin number we connected in the code section.) The (-) leg (short leg) of the Led Diode is connected to the board of Arduino. We complete the circuit by connecting it to the GND pin.



## EXPLANATIONAL CODE SECTION

```
// Flashing led diode application
// The "void setup" section below works once
when you press the reset button or power the
board.
void setup() {
pinMode(13, OUTPUT); // Initializes digital pin 13
as output
}
// The "void loop" section below is the loop
function. It works by repetitively.
void loop() {
digitalWrite(13, HIGH); // sends energy to pin 13
delay(1000); // waits 1000 milliseconds (1 second)
digitalWrite(13, LOW); // cuts off the power on
pin 13
delay(1000); // waits 1000 milliseconds (1 second)
}
```

.....

## PLAIN CODE SECTION

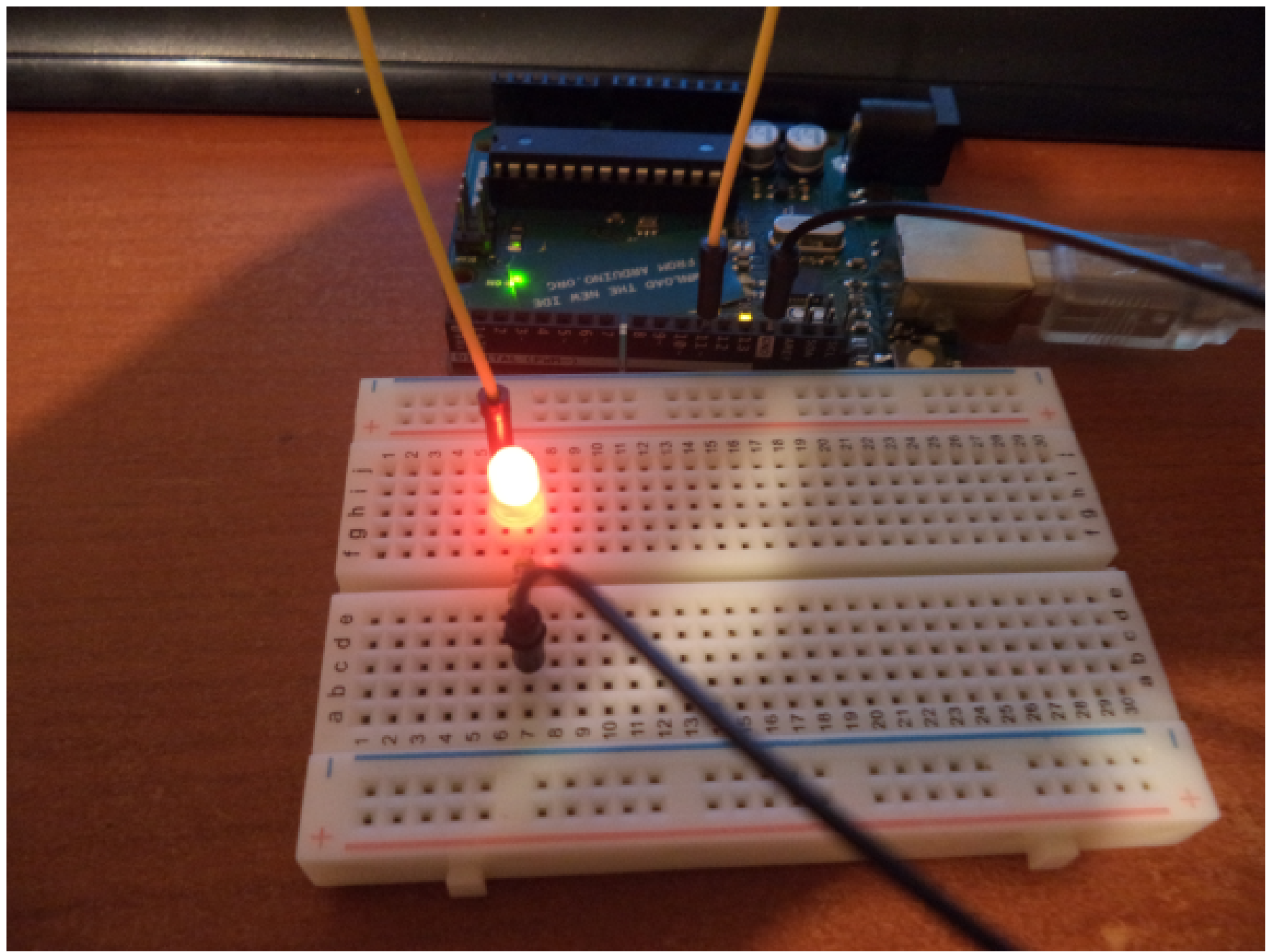
```
void setup() {
pinMode(13, OUTPUT);
}

void loop() {
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

.....

Both code tables are the same. Whether you paste the first code table or the second code table into the Arduino, both will run the same circuit. Only the first code table contains descriptions of what action to take on the coding line. These explanations will allow you to better understand which phases are written into the line of code while running the circuit.

The code section is designed using pin 13 on our Arduino board. During the operation of the circuit, pin 13 is programmed to give energy (volts) for 1 second and not to give energy (volt) for 1 second.



# REFERENCE

<https://egitim.ahmetcandemir.com.tr/arduino-blink-buzzer-yanip-sonen-led/>

<http://temrinlerim.org/index.php?topic=209.0>

<https://kodlamayap.com/2017/11/28/mblock-ile-arduino-kodlama/>

<http://www.kodlakafa.com/mblock/mblock-ve-arduino-ile-led-kontrolu-blink-nasil-yapilir/>

<https://education.makeblock.com/secondary/>