



Mini Lab

Project 7

LED RGB

PROJECT N° 7: RGB LED

Learn how to create a program to control the RGB LED.

EXERCISES INDEX:

- Exercise 7.1: Turn the RGB LED on and off by using the button
- Exercise 7.2: Change the color of the RGB LED by using the button
- Exercise 7.3: Change the color of the RGB LED by using the button and the potentiometer

Exercise 6.1: Turn the RGB LED on and off by using the button

Learn how to program so that, when pressing the button, the RGB LED turns on violet. When releasing the button, the RGB LED will turn off.

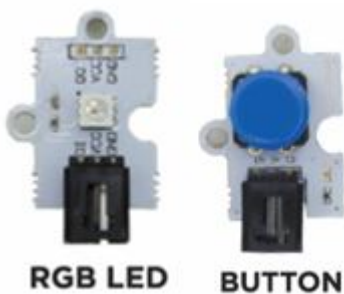
DIFFICULTY LEVEL: Advanced.

DURATION OF THE EXERCISE: 40 min.

MATERIALS:

- RGB LED
- Button
- Computer

The Mini Lab will have to be built according to the instructions manual.



CONNECTIONS:

1. Connect the RGB LED to the digital port 6 of the Build&Code 4in1 board.
2. Connect the button to the digital port 4 of the Build&Code 4in1 board.

PROGRAMMING CODE:

You can do this project using Arduino or a visual programming software by blocks compatible. Below you will find the necessary code.

Arduino Code

1. Download and install the Arduino IDE program.
2. In order to program the RGB LED with Arduino IDE in Windows, you need to add a library. You can do it by following the steps indicated in this link.
3. Once you have installed the library, copy the following program in it:

```
#include <Adafruit_NeoPixel.h> // RGB LED LIBRARY
#define PIXEL_PIN 6 // RGB LED PIN
#define PIXEL_COUNT 1 // LEDS RGB QUANTITY

Adafruit_NeoPixel strip =
Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, NEO_GRB +
```

```

NEO_KHZ800); // CONFIGURATION OF THE RGB LED
int valuepuls = 0, pinpuls = 4; // BUTTON VARIABLE
AND PIN

void setup() {
  // put your setup code here, to run once:
  // CONFIGURATION OF THE RGB LED
  strip.begin();
  strip.show();
  // CONFIGURATION OF THE BUTTON
  pinMode (pinpuls, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  // IF WE PRESS THE BUTTON THE LED3 WILL TURN ON
  valuepuls = digitalRead (pinpuls);
  if (valuepuls == HIGH) // BUTTON NOT PRESSED
  {
    colorWipe(strip.Color(0,0,0),0); // LED RGB = ON
    ; VIOLET COLOR (RED=60, GREEN=0, BLUE=60)
    colorWipe(strip.Color(numRED,numGREEN,numBLUE), 0)
  }
  else
  {
    colorWipe(strip.Color(60,0,60),0); // RGB LED =
    ON ; VIOLET COLOR (RED=60, GREEN=0, BLUE=60)
    colorWipe(strip.Color(numRED,numGREEN,numBLUE), 0)
  }
}

// FUNCTION OF THE RGB LED COLOR
void colorWipe(uint32_t c, uint8_t wait) {
  for(uint16_t i=0; i<strip.numPixels(); i++) {
    strip.setPixelColor(i, c);
    strip.show();
    delay(wait);
  }
}

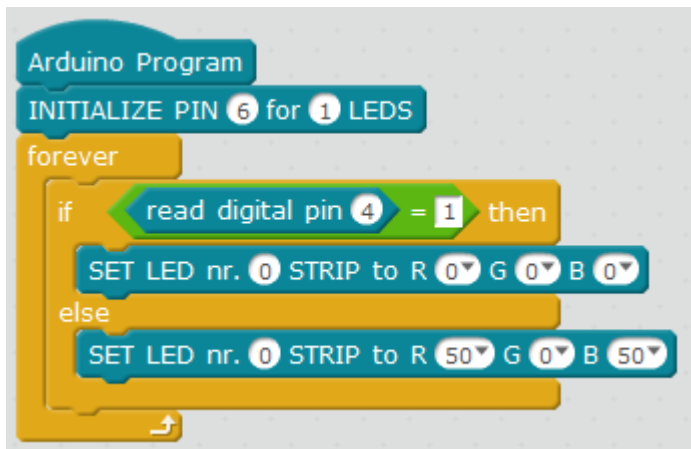
```

Code for the visual programming software by blocks compatible

1. Download and install the program.
2. In order to program the RGB LED with the visual programming software

by blocks you need to install the extension NeoPixel by Robokacija from Ivan Kunovi in the program. Follow the steps indicated in the following link.

3. Once you have installed the extension, copy the following code:



*Note that every time you charge a new program and the content on the screen changes, you will have to restart the Build&Code 4in1 board.

1. Configure and upload the code, following the indications on the Mini Lab First Steps guide.
2. Check that the BTL/USB switch on the Build&Code 4in1 board is set to USB, to upload the code correctly.

RESULT OF THE EXERCISE:

When pressing the button, the RGB LED will turn on violet. When you stop pressing the button, the RGB LED will turn off.

Exercise 7.2: Change the color of the RGB LED by using the button

Learn how to create a program so that, when pressing the button, the RGB LED turn on and changes its color every half second.

DIFFICULTY LEVEL: Advanced.

DURATION OF THE EXERCISE: 40 min.

MATERIALS:

- RGB LED

- Button
- Computer

The Mini Lab will have to be built according to the instructions manual.



CONNECTIONS:

1. Connect the RGB LED to the digital port 6 of the Build&Code 4in1 board.
2. Connect the button to the digital port 4 of the Build&Code 4in1 board.

PROGRAMMING CODE:

You can do this project using Arduino or a visual programming software by blocks compatible. Below you will find the necessary code.

Arduino Code

1. Download and install the Arduino IDE program. It is available for Windows, Mac OS and Linux.
2. Open the Arduino program and copy the following program in it:

```
#include <Adafruit_NeoPixel.h> //RGB LED LIBRARY
#define PIXEL_PIN 6 //RGB LED PIN
#define PIXEL_COUNT 1 // RGB LEDS QUANTITY

Adafruit_NeoPixel strip =
Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, NEO_GRB +
NEO_KHZ800); // CONFIGURATION OF THE RGB LED
int valuepuls = 0, pinpuls = 4; // VARIABLE AND PIN
OF THE BUTTON

void setup() {
// put your setup code here, to run once:
// CONFIGURATION OF THE RGB LED
strip.begin();
```

```

strip.show();
// CONFIGURATION OF THE BUTTON
pinMode (pinpuls, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  // IF WE PRESS THE BUTTON, THE LED3 WILL TURN ON
  valuepuls = digitalRead (pinpuls);
  if (valuepuls == HIGH) // BUTTON NOT PRESSED
  {
    colorWipe(strip.Color(0,0,0),0); // RGB LED = ON
    ; VIOLET COLOR (RED=60, GREEN=0, BLUE=60)
    colorWipe(strip.Color(numRED,numGREEN,numBLUE), 0)
  }
  else
  {
    colorWipe(strip.Color(60,0,0),0); // RGB LED = ON
    ; RED COLOR (RED=60, GREEN=0, BLUE=0)
    delay (500);
    colorWipe(strip.Color(60,60,0),0); // RGB LED =
    ON ; YELLOW COLOR (RED=60, GREEN=60, BLUE=0)
    delay (500);
    colorWipe(strip.Color(60,40,0),0); // RGB LED =
    ON ; ORANGE COLOR (RED=60, GREEN=40, BLUE=60)
    delay (500);
    colorWipe(strip.Color(0,60,0),0); // RGB LED = ON
    ; GREEN COLOR (RED=0, GREEN=60, BLUE=60)
    delay (500);
    colorWipe(strip.Color(0,0,60),0); // RGB LED = ON
    ; BLUE COLOR (RED=0, GREEN=0, BLUE=60)
    delay (500);
    colorWipe(strip.Color(60,0,60),0); // RGB LED =
    ON ; VIOLET COLOR (RED=60, GREEN=0, BLUE=60)
    delay (500);
    colorWipe(strip.Color(60,60,60),0); // RGB LED =
    ON ; WHITE COLOR (RED=60, GREEN=60, BLUE=60)
    delay (500);
  }
}

// FUNCION PARA EL COLOR DEL LED RGB
void colorWipe(uint32_t c, uint8_t wait) {
  for(uint16_t i=0; i<strip.numPixels(); i++) {

```

```

    strip.setPixelColor(i, c);
    strip.show();
    delay(wait);
  }
}

```

Code for the visual programming software by blocks compatible

1. Download and install the program.
2. In order to program the RGB LED with the visual programming software by blocks you need to install the extension NeoPixel by Robokacija from Ivan Kunovi in the program. Follow the steps indicated in the following link.
3. Once you have installed the extension, copy the following code:



*Note that every time you charge a new program and the content on the screen changes, you will have to restart the Build&Code 4in1 board.

1. Configure and upload the code, following the indications on the Mini Lab

First Steps guide.

2. Check that the BTL/USB switch on the Build&Code 4in1 board is set to USB, to upload the code correctly.

RESULT OF THE EXERCISE:

When pressing the button, the RGB LED will turn on and change its color every half second.

Exercise 7.3: Change the color of the RGB LED by using the button

Learn how to create a program so that, when the button is not pressed, the RGB LED emits warm colors with the turn of the potentiometer. When the button is pressed, the RGB LED will emit cold colors with the turn of the potentiometer.

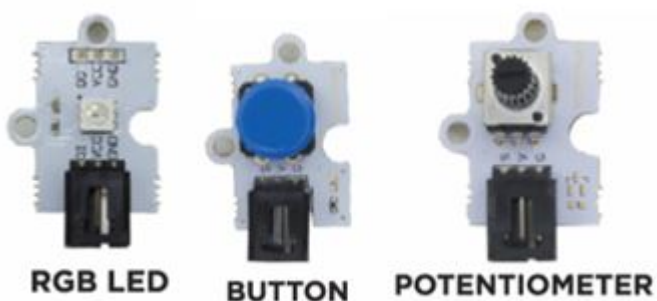
DIFFICULTY LEVEL: Advanced.

DURATION OF THE EXERCISE: 40 min.

MATERIALS:

- RGB LED
- Button
- Potentiometer
- Computer

The Mini Lab will have to be built according to the instructions manual.



CONNECTIONS:

1. Connect the RGB LED to the digital port 6 of the Build&Code 4in1 board.

2. Connect the button to the digital port 4 of the Build&Code 4in1 board.
3. Connect the potentiometer to the digital port A1 of the Build&Code 4in1 board.

PROGRAMMING CODE:

You can do this project using Arduino or a visual programming software by blocks compatible. Below you will find the necessary code.

Arduino Code

1. Download and install the Arduino IDE program. It is available for Windows, Mac OS and Linux.
2. Open the Arduino program and copy the following program in it:

```
#include <Adafruit_NeoPixel.h> //RGB LED LIBRARY

#define PIXEL_PIN 6 //RGB PIN LIBRARY
#define PIXEL_COUNT 1 // RGB LEDS UANTITY
Adafruit_NeoPixel strip =
Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, NEO_GRB +
NEO_KHZ800); // CONFIGURATION OF THE RGB LED
int valuepuls = 0, pinpuls = 4; // VARIABLE AND PIN
OF THE BUTTON
int pot = 0, pinpot = A1; //VARIABLES AND PINS FOR
THE POTENTIOMETER
int R = 0, G = 0, B = 0; // VARIABLES FOR THE RED
(R), GREEN (G) AND BLUE (B) COLORS

void setup() {
  // put your setup code here, to run once:
  // CONFIGURATION OF THE RGB LED
  strip.begin();
  strip.show();
  // CONFIGURATION OF THE BUTTON
  pinMode (pinpuls, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  pot = analogRead (pinpot); // READING OF THE
POTENTIOMETER VALUE
  valuepuls = digitalRead (pinpuls); // READING OF
THE BUTTON
```

```

    if (valuepuls == HIGH) // BUTTON NOT PRESSED
    {
        R = map (pot, 0,1023, 0, 255); // VALUES RELATION
        OF THE POTENTIOMETER FOR THE RED COLOR
        G = map (pot, 1023,0, 0, 255); // VALUES
        RELATION OF THE POTENTIOMETER FOR THE GREEN COLOR
        colorWipe(strip.Color(R,G,0),0); // LED RGB = ON
        ; VIOLET COLOR (RED=60, GREEN=0, BLUE=60)
        colorWipe(strip.Color(numRED,numGREEN,numBLUE), 0)
    }
    else
    {
        R = map (pot, 1023,0, 0, 255); // VALUES RELATION
        OF THE POTENTIOMETER FOR THE RED COLOR
        B = map (pot, 0,1023, 0, 255); // VALUES RELATION
        OF THE POTENTIOMETER FOR THE RED COLOR
        colorWipe(strip.Color(R,0,B),0); // LED RGB = ON
        ; RED COLOR
        (RED=60, GREEN=0, BLUE=0)
    }
}

// COLOR FUNCTION FOR THE RGB LED
void colorWipe(uint32_t c, uint8_t wait) {
    for(uint16_t i=0; i<strip.numPixels(); i++) {
        strip.setPixelColor(i, c);
        strip.show();
        delay(wait);
    }
}

```

Code for the visual programming software by blocks compatible

1. Download and install the program.
2. In order to program the RGB LED with the visual programming software by blocks you need to install the extension NeoPixel by Robokacija from Ivan Kunovi in the program. Follow the steps indicated in the following link.
3. Once you have installed the extension, copy the following code:

```
Arduino Program
INITIALIZE PIN 6 for 1 LEDS
forever
  if read digital pin 4 = 1 then
    SET LED nr. 0 STRIP to R read analog pin (A) 1 / 4 G 255 - read analog pin (A) 1 / 4 B 0
  else
    SET LED nr. 0 STRIP to R read analog pin (A) 1 / 4 G 0 B 255 - read analog pin (A) 1 / 4
```

*Note that every time you charge a new program and the content on the screen changes, you will have to restart the Build&Code 4in1 board.

4. Configure and upload the code, following the indications on the Mini Lab First Steps guide.
5. Check that the BTL/USB switch on the Build&Code 4in1 board is set to USB, to upload the code correctly.

RESULT OF THE EXERCISE:

When the button is not pressed, the RGB LED will emit warm colors with the turn of the potentiometer. When the button is pressed, the RGB LED will emit cold colors with the turn of the potentiometer.