

A large teal rectangular overlay is positioned in the center of the page, partially covering a background image of an ebotics board. The text "PROYECTO Nº 7" and "LED RGB" is written in white, bold, sans-serif font within this overlay.

PROYECTO Nº 7

LED RGB

PROYECTO 7: LED RGB

Aprende a crear un programa para controlar el LED RGB.

ÍNDICE DE EJERCICIOS:

- Ejercicio 7.1: Enciende y apaga el LED RGB utilizando el pulsador
- Ejercicio 7.2: Cambia el color del LED RGB usando el botón
- Ejercicio 7.3: Cambia el color del LED RGB usando el potenciómetro y el pulsador

Ejercicio 7.1: Enciende y apaga el LED RGB utilizando el pulsador

Aprende a crear un programa para que al presionar el pulsador, el LED RGB se encienda y se ilumine de color violeta. Al soltar el pulsador, el LED RGB se apagará.

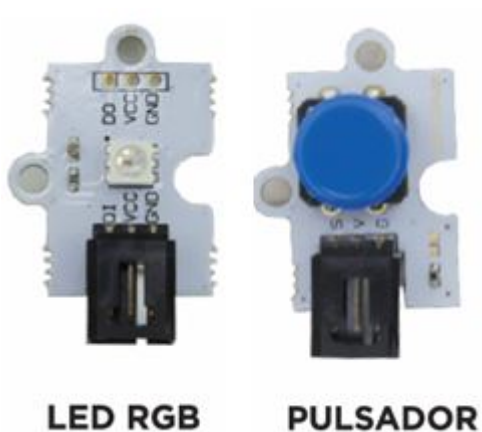
NIVEL DE DIFICULTAD: Avanzada.

DURACIÓN DEL EJERCICIO: 40 min.

MATERIALES:

- LED RGB
- Pulsador
- Ordenador

El kit Mini Lab deberá estar montado de acuerdo a las instrucciones indicadas en el manual.



CONEXIONES:

1. Conecta el LED RGB al puerto digital 6 de la placa Build&Code 4in1.
2. Conecta el pulsador al puerto digital 4 de la placa Build&Code 4in1.

CÓDIGO DE PROGRAMACIÓN

Realiza esta actividad utilizando Arduino o el *software* gráfico de programación por bloques compatible. A continuación, encontrarás el código de programación necesario.

Código Arduino

1. [Descarga el software Arduino](#) y realiza el proceso de instalación.
2. Para poder programar el LED RGB con ArduinoIDE en Windows, es necesario añadir una librería en el programa. Puedes hacerlo siguiendo los pasos indicados en [este enlace](#).

Una vez instalada la librería, introduce el siguiente código:

```
#include <Adafruit_NeoPixel.h> //LIBRERIA LED RGB
#define PIXEL_PIN 6 //PIN DEL LED RGB
#define PIXEL_COUNT 1 // NUMERO DE LEDS RGB

Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN,
NEO_GRB + NEO_KHZ800); // CONFIGURACION DEL LED RGB
int valuepuls = 0, pinpuls = 4; // VARIABLE Y PIN DEL PULSADOR

void setup() {
  // put your setup code here, to run once:
  //CONFIGURACION DEL LED RGB
  strip.begin();
  strip.show();
  //CONFIGURACIÓN DEL PULSADOR
  pinMode (pinpuls, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  // SI PULSAMOS EL PULSADOR SE ENCENDERA EL LED3
  valuepuls = digitalRead (pinpuls);
  if (valuepuls == HIGH) // PULSADOR NO PULSADO
  {
    colorWipe(strip.Color(0,0,0),0); // LED RGB = ON ; COLOR VIOLETA
    (RED=60, GREEN=0, BLUE=60) colorWipe(strip.Color(númRED, númGREEN, númBLUE),
    0)
  }
  else
  {
    colorWipe(strip.Color(60,0,60),0); // LED RGB = ON ; COLOR VIOLETA
    (RED=60, GREEN=0, BLUE=60) colorWipe(strip.Color(númRED, númGREEN, númBLUE),
    0)
  }
}

// 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);
  }
}
```

Código para software de programación por bloques compatible

1. [Descarga el software](#) y realiza en proceso de instalación.
2. Para poder programar el LED RGB con el *software* de programación por bloques compatible es necesario añadir la extensión NeoPixel by Robokacija de Ivan Kunovi al programa. Accediendo a [este enlace](#) podrás consultar los pasos necesarios.
3. Una vez instalada la extensión, introduce el siguiente código:



*Ten en cuenta que cada vez que cargues un programa nuevo y el contenido de la pantalla cambie, debes de reiniciar la placa Build&Code 4in1 del Mini Lab.

RESULTADO DEL EJERCICIO:

Al presionar el pulsador, el LED RGB se encenderá y se iluminará de color violeta. Al soltar el pulsador, el LED RGB se apagará.

Ejercicio 7.2: Cambia el color del LED RGB usando el botón

Aprende a crear un programa para que, al presionar el pulsador, el LED RGB se encienda y cambie de color en intervalos de medio segundo.

NIVEL DE DIFICULTAD: Avanzada.

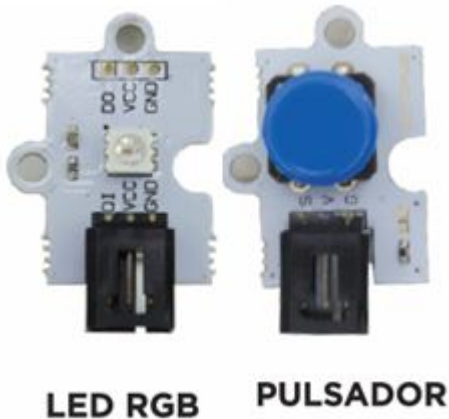
DURACIÓN DEL EJERCICIO: 40 min.

MATERIALES:

- LED RGB
- Pulsador

- Ordenador

El kit Mini Lab deberá estar montado de acuerdo a las instrucciones indicadas en el manual.



CONEXIONES:

1. Conecta el LED RGB al puerto digital 6 de la placa Build&Code 4in1.
2. Conecta el botón al puerto digital 4 de la placa Build&Code 4in1.

CÓDIGO DE PROGRAMACIÓN

Realiza esta actividad utilizando Arduino o el *software* gráfico de programación por bloques compatible. A continuación, encontrarás el código de programación necesario.

Código Aduino

1. [Descarga el software Arduino](#) y realiza el proceso de instalación.
2. Abre el programa Arduino y, una vez en él, copia el siguiente código:

```
#include <Adafruit_NeoPixel.h> //LIBRERIA LED RGB
#define PIXEL_PIN 6 //PIN DEL LED RGB
#define PIXEL_COUNT 1 // NUMERO DE LEDS RGB

Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN,
NEO_GRB + NEO_KHZ800); // CONFIGURACION DEL LED RGB
int valuepuls = 0, pinpuls = 4; // VARIABLE Y PIN DEL PULSADOR

void setup() {
  // put your setup code here, to run once:
  //CONFIGURACION DEL LED RGB
  strip.begin();
  strip.show();
}
```

```
//CONFIGURACIÓN DEL PULSADOR
pinMode (pinpuls, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  // SI PULSAMOS EL PULSADOR SE ENCENDERA EL LED3
  valuepuls = digitalRead (pinpuls);
  if (valuepuls == HIGH) // PULSADOR NO PULSADO
  {
    colorWipe(strip.Color(0,0,0),0); // LED RGB = ON ; COLOR
    VIOLETA (RED=60,GREEN=0,BLUE=60)
    colorWipe(strip.Color(númRED,númGREEN,númBLUE), 0)
  }
  else
  {
    colorWipe(strip.Color(60,0,0),0); // LED RGB = ON ; COLOR ROJO
    (RED=60,GREEN=0,BLUE=0)
    delay (500);
    colorWipe(strip.Color(60,60,0),0); // LED RGB = ON ; COLOR
    AMARILLO (RED=60,GREEN=60,BLUE=0)
    delay (500);
    colorWipe(strip.Color(60,40,0),0); // LED RGB = ON ; COLOR
    NARANJA (RED=60,GREEN=40,BLUE=60)
    delay (500);
    colorWipe(strip.Color(0,60,0),0); // LED RGB = ON ; COLOR VERDE
    (RED=0,GREEN=60,BLUE=60)
    delay (500);
    colorWipe(strip.Color(0,0,60),0); // LED RGB = ON ; COLOR AZUL
    (RED=0,GREEN=0,BLUE=60)
    delay (500);
    colorWipe(strip.Color(60,0,60),0); // LED RGB = ON ; COLOR
    VIOLETA (RED=60,GREEN=0,BLUE=60)
    delay (500);
    colorWipe(strip.Color(60,60,60),0); // LED RGB = ON ; COLOR
    BLANCO (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);
  }
}
```

```
}  
}
```

Código para software de programación por bloques compatible

1. [Descarga el software](#) y realiza en proceso de instalación.
2. Para poder programar el LED RGB con el *software* de programación por bloques compatible es necesario añadir la extensión Neopixel by Robokacija de Ivan Kunovi al programa. Accediendo a [este enlace](#) podrás consultar los pasos necesarios.
3. Una vez instalada la extensión, introduce el siguiente código:

```
Programa de Arduino  
INITIALIZE PIN 6 for 1 LEDS  
por siempre  
  si leer pin digital 4 = 1 entonces  
    SET LED nr. 0 STRIP to R 0 G 0 B 0  
  si no  
    SET LED nr. 0 STRIP to R 60 G 0 B 0  
    esperar 0.5 segundos  
    SET LED nr. 0 STRIP to R 60 G 60 B 0  
    esperar 0.5 segundos  
    SET LED nr. 0 STRIP to R 60 G 40 B 0  
    esperar 0.5 segundos  
    SET LED nr. 0 STRIP to R 0 G 60 B 0  
    esperar 0.5 segundos  
    SET LED nr. 0 STRIP to R 0 G 0 B 60  
    esperar 0.5 segundos  
    SET LED nr. 0 STRIP to R 60 G 0 B 60  
    esperar 0.5 segundos  
    SET LED nr. 0 STRIP to R 60 G 60 B 60  
    esperar 0.5 segundos
```

RESULTADO DEL EJERCICIO:

Al presionar el pulsador, el LED RGB se encenderá y cambiará de color en intervalos de medio segundo.

Ejercicio 7.3: Cambia el color del LED RGB usando el potenciómetro y el pulsador

Aprende a crear un programa para que, cuando el botón no esté pulsado, el LED RGB emita colores cálidos con el giro del potenciómetro. Cuando el botón esté pulsado, el LED RGB emitirá colores fríos con el giro del potenciómetro.

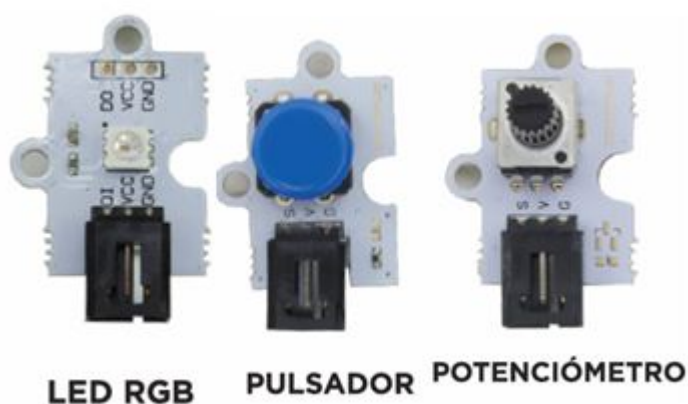
NIVEL DE DIFICULTAD: Avanzada.

DURACIÓN DEL EJERCICIO: 40 min.

MATERIALES:

- LED RGB
- Pulsador
- Potenciómetro
- Ordenador

El kit Mini Lab deberá estar montado de acuerdo a las instrucciones indicadas en el manual.



CONEXIONES:

1. Conecta el LED RGB al puerto digital 6 de la placa Build&Code 4in1.
2. Conecta el botón al puerto digital 4 de la placa Build&Code 4in1.
3. Conecta el potenciómetro al puerto analógico A1 de la placa Build&Code 4in1.

CÓDIGO DE PROGRAMACIÓN

Realiza esta actividad utilizando Arduino o el *software* gráfico de programación por bloques compatible. A continuación, encontrarás el código de programación necesario.

Código Aduino

1. [Descarga el software Arduino](#) y realiza el proceso de instalación.
2. Abre el programa Arduino y, una vez en él, copia el siguiente código:

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

#define PIXEL_PIN 6 //PIN DEL LED RGB
#define PIXEL_COUNT 1 // NUMERO DE LEDS RGB
Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN,
NEO_GRB + NEO_KHZ800); // CONFIGURACION DEL LED RGB
int valuepuls = 0, pinpuls = 4; // VARIABLE Y PIN DEL PULSADOR
int pot = 0, pinpot = A1; //VARIABLES PINS DEL POTENCIOMETRO
int R = 0, G = 0, B = 0; // VARIABLES PARA EL COLOR ROJO (R), VERDE
(G) Y AZUL (B)

void setup() {
  // put your setup code here, to run once:
  //CONFIGURACION DEL LED RGB
  strip.begin();
  strip.show();
  //CONFIGURACIÓN DEL PULSADOR
  pinMode (pinpuls, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  pot = analogRead (pinpot); // LECTURA DEL VALOR DEL POTENCIOMETRO
  valuepuls = digitalRead (pinpuls); // LECTURA DEL PULSADOR
  if (valuepuls == HIGH) // PULSADOR NO PULSADO
  {
    R = map (pot, 0,1023, 0, 255); // RELACIÓN DE VALORES DEL
POTENCIOMETRO PARA EL COLOR ROJO
    G = map (pot, 1023,0, 0, 255); // RELACIÓN DE VALORES DEL
POTENCIOMETRO PARA EL COLOR VERDE
    colorWipe(strip.Color(R,G,0),0); // LED RGB = ON ; COLOR
VIOLETA (RED=60,GREEN=0,BLUE=60)
    colorWipe(strip.Color(númRED,númGREEN,númBLUE), 0)
  }
  else
  {
    R = map (pot, 1023,0, 0, 255); // RELACIÓN DE VALORES DEL
POTENCIOMETRO PARA EL COLOR ROJO
    B = map (pot, 0,1023, 0, 255); // RELACIÓN DE VALORES DEL
POTENCIOMETRO PARA EL COLOR ROJO
```

```

        colorWipe(strip.Color(R,0,B),0); // LED RGB = ON ; COLOR ROJO
        (RED=60, GREEN=0, BLUE=0)
    }
}

// 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);
    }
}

```

Código para software de programación por bloques compatible

1. [Descarga el software](#) y realiza en proceso de instalación.
2. Para poder programar el LED RGB con el *software* de programación por bloques compatible es necesario añadir la extensión NeoPixel by Robokacija de Ivan Kunovi al programa. Accediendo a [este enlace](#) podrás consultar los pasos necesarios.
3. Una vez instalada la extensión, introduce el siguiente código:



RESULTADO DEL EJERCICIO:

Cuando el botón no esté pulsado, el LED RGB emitirá colores cálidos con el giro del potenciómetro. Cuando el botón esté pulsado, el LED RGB emitirá colores fríos con el giro del potenciómetro.