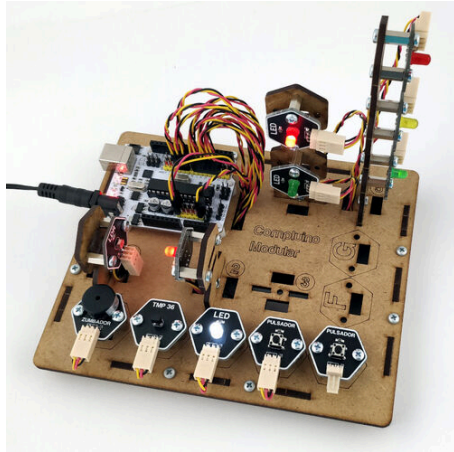
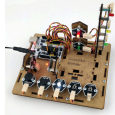




Kit Compluino Modular

Nueva versión

Completo kit de trabajo con Compluino

UNO, compatible Arduino

Incluye una amplia variedad de sensores y actuadores

Ideal para trabajar programación y proyectos STEAM en entornos educativos

EN STOCK (10)

OFERTA

-16% ANTES 90€

75 €

21.00% IVA incluido

- 1 +
unidades

AÑADIR A CESTA

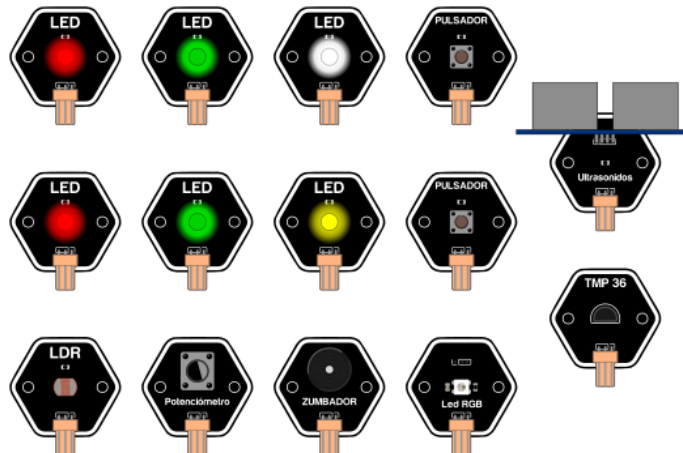
Características

Documentación

Contenido del kit

Este kit incluye los siguientes materiales:

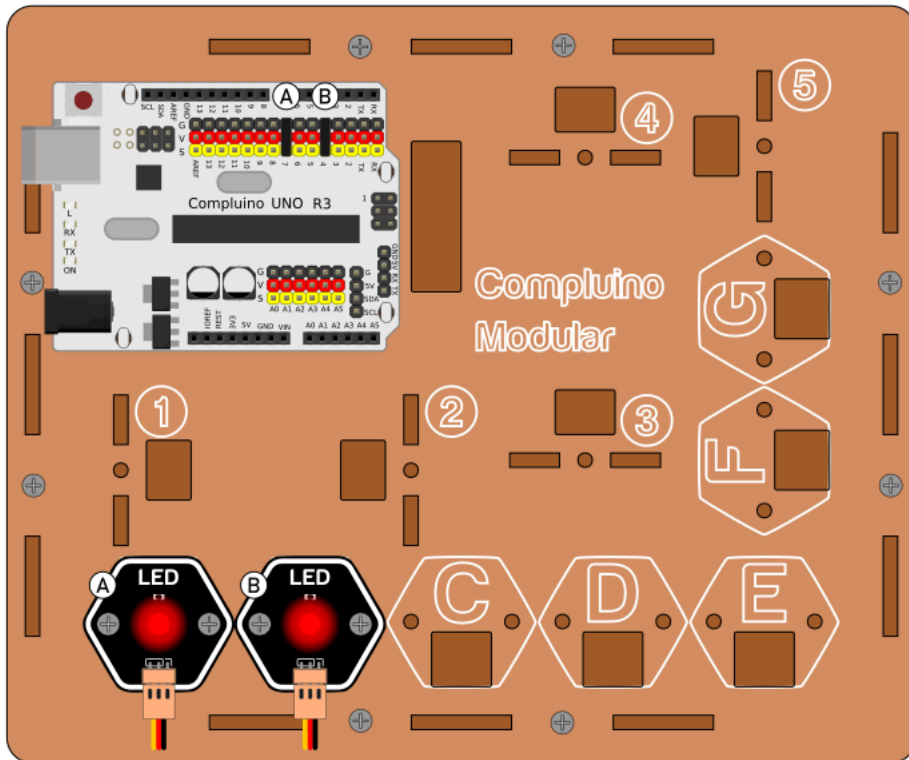
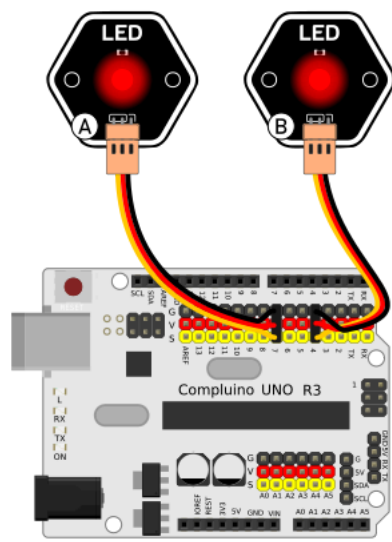
- 1 - Placa Compluino UNO
- 1 - Cable USB A – B.
- 1 - Chasis de madera + Tornillería + Separadores M3.
- 1 - Destornillador PH1.
- 2 - Módulo led rojo con conector Molex.
- 2 - Módulo led verde con conector Molex.
- 1 - Módulo led amarillo con conector Molex.
- 1 - Módulo led blanco con conector Molex.
- 2 - Módulo pulsador con conector Molex.
- 1 - Módulo LDR con conector Molex.
- 1 - Módulo potenciómetro con conector Molex.
- 1 - Módulo zumbador con conector Molex.
- 1 - Módulo TMP 36 con conector Molex.
- 1 - Módulo RGB Neopixel con protección y conector Molex.
- 1 - Módulo HC-SR04, en configuración "un hilo" y conector Molex.
- 7 - Cables de conexión Molex-Header trenzados.



Montaje del chasis

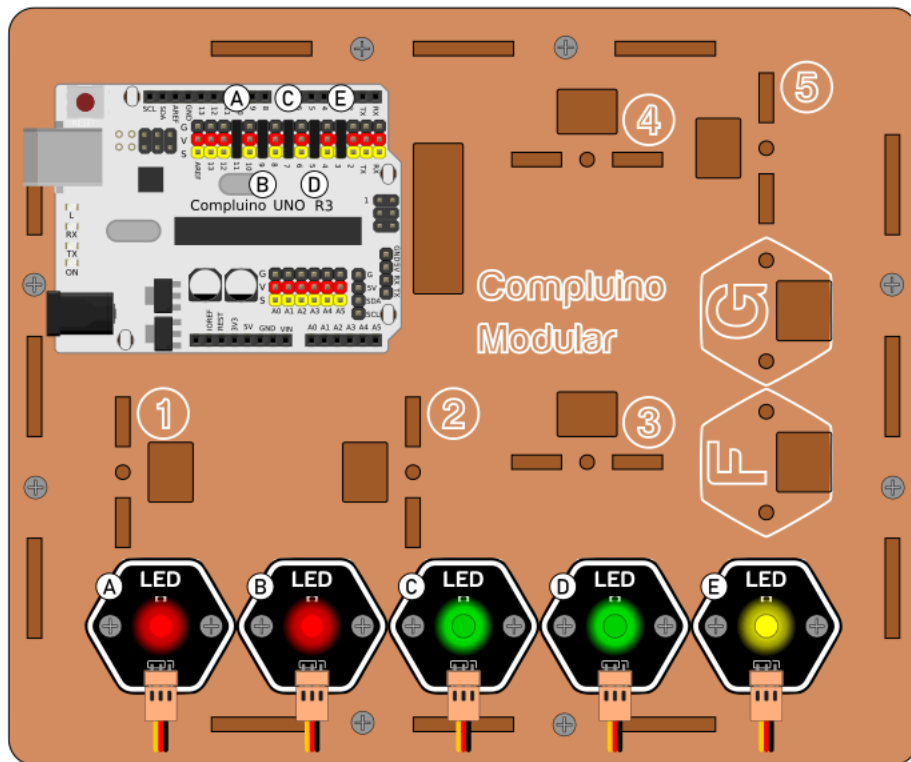
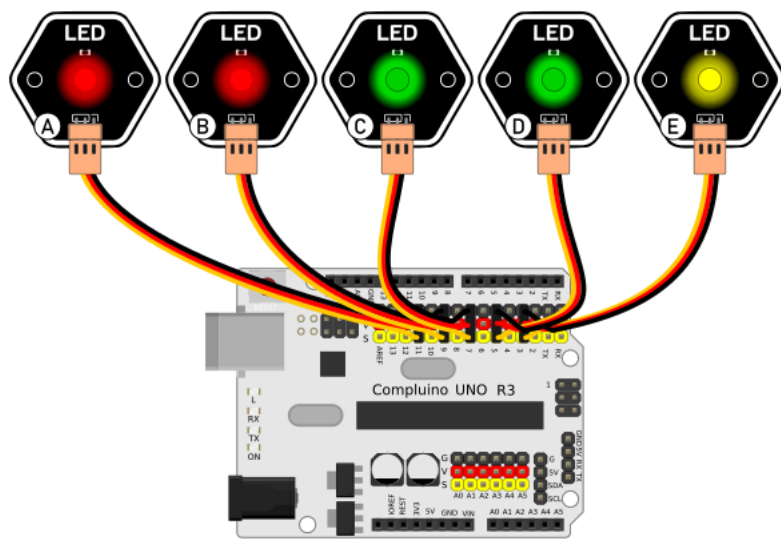
Ejemplos y actividades

Ejemplo 1: Parpadeo de dos ledes rojos cada 2000 milisegundos



```
// Ejemplo 1: parpadeo de dos ledes rojos.
#define LEDA 7
#define LEDB 4
void setup()
{
  pinMode(LEDA, OUTPUT); // Definir el pin 7 como salida.
  pinMode(LEDB, OUTPUT); // Definir el pin 4 como salida.
}
void loop()
{
  digitalWrite (LEDA, HIGH); // Encender el led A
  digitalWrite (LEDB, HIGH); // Encender el led B
  delay (2000); // Esperar dos segundos
  digitalWrite (LEDA, LOW); // Apagar el led A
  digitalWrite (LEDB, LOW); // Apagar el led B
  delay (2000); // Esperar dos segundos
}
```

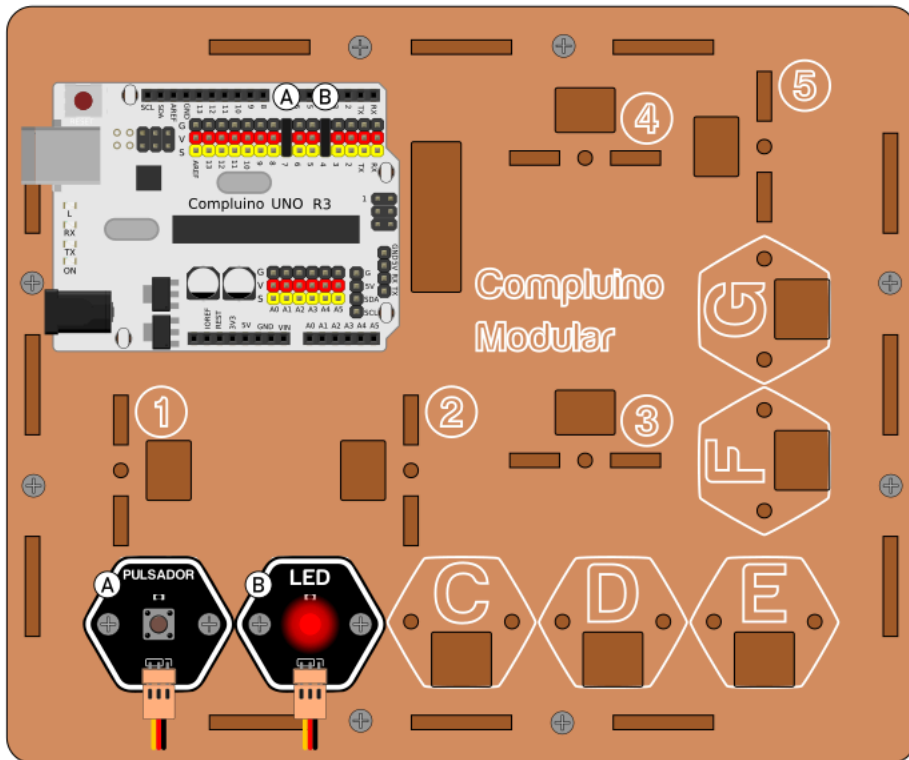
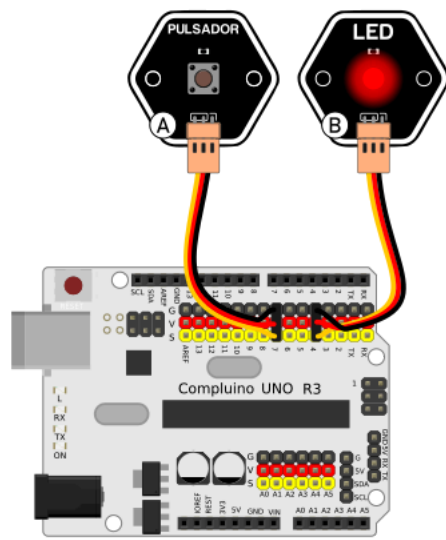
Ejemplo 2: iluminación secuencial de cinco ledes



// Ejemplo 2: iluminación secuencial de cinco ledes.

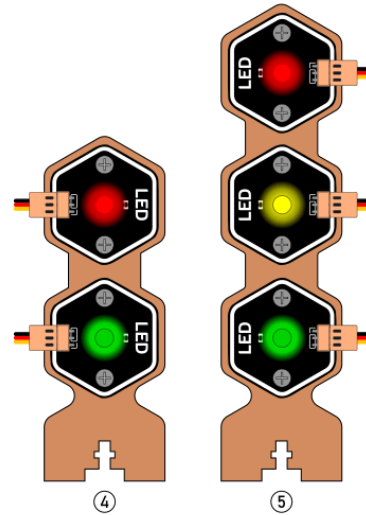
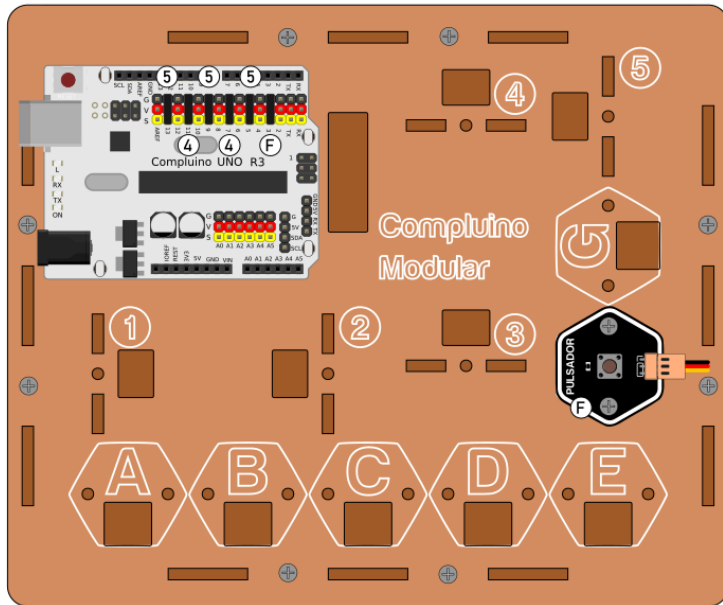
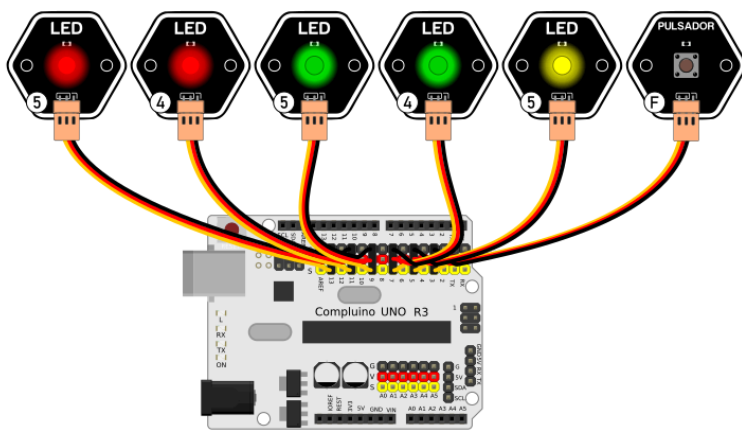
```
#define LEDA 11
#define LEDB 9
#define LEDC 7
#define LEDD 5
#define LEDE 3
#define TIEMPO 125
void setup()
{
  pinMode(LEDA, OUTPUT); // Definir el pin 11 como salida
  pinMode(LEDB, OUTPUT); // Definir el pin 9 como salida
  pinMode(LEDC, OUTPUT); // Definir el pin 7 como salida
  pinMode(LEDD, OUTPUT); // Definir el pin 5 como salida
  pinMode(LEDE, OUTPUT); // Definir el pin 3 como salida
}
void loop()
```

Ejemplo 3: Encender o apagar un led dependiendo del estado de un pulsador



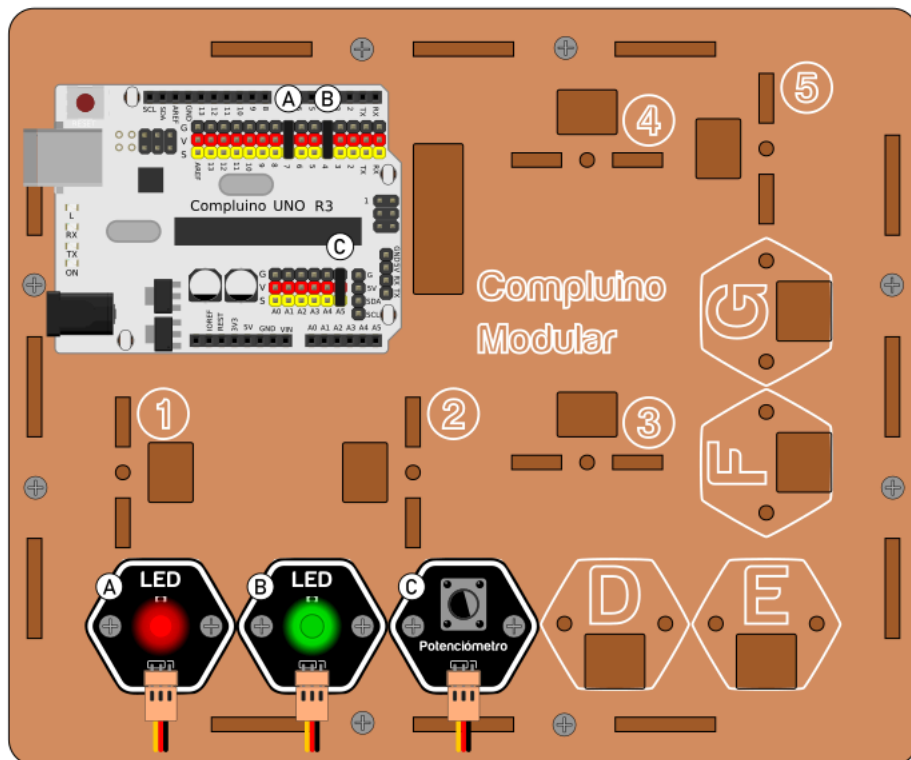
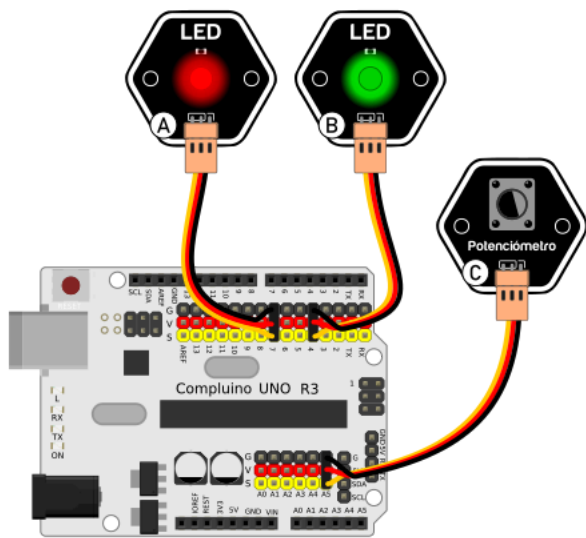
```
// Ejemplo 3: encender o apagar un led dependiendo del estado de un pulsador.
#define LED 4
#define PULSADOR 7
void setup()
{
  pinMode(LED, OUTPUT); // Definir el pin 4 como salida
  pinMode(PULSADOR, INPUT); // Definir el pin 7 como entrada
}
void loop()
{
  if (digitalRead(PULSADOR)==HIGH) //Comparar el estado que se lee en el pin del pulsador
  {
    digitalWrite(LED,HIGH); // Si se cumple la condición enciende el led
  }
  else
  {
  }
```

Ejemplo 4: Semáforo para peatones y vehículos con pulsador



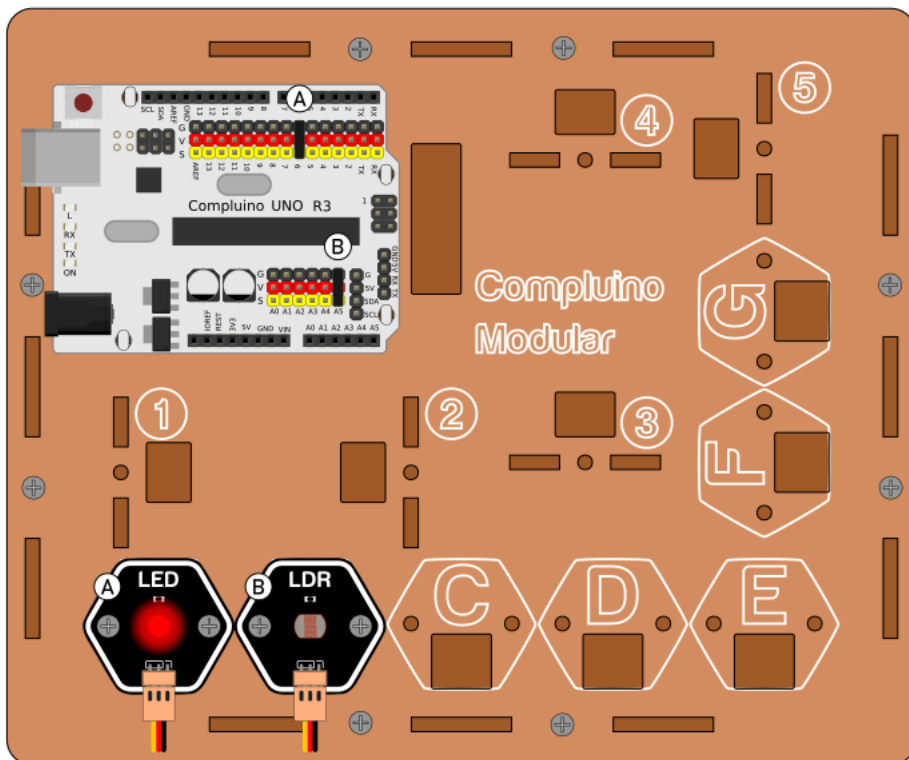
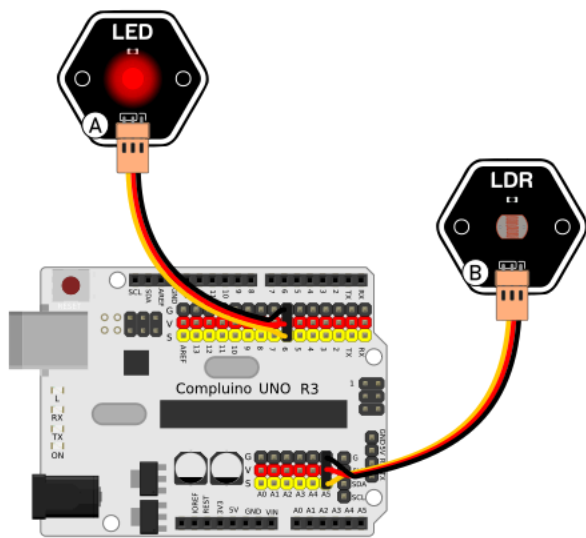
```
// Ejemplo 4: Semáforo peatones y vehículos con pulsador
#define C_ROJO 13
#define C_AMBAR 5
#define C_VERDE 9
#define P_ROJO 11
#define P_VERDE 7
#define PULSADOR 3
void setup()
{
  pinMode(C_ROJO, OUTPUT); // Definir el pin 13 como salida
  pinMode(C_AMBAR, OUTPUT); // Definir el pin 5 como salida
  pinMode(C_VERDE, OUTPUT); // Definir el pin 9 como salida
  pinMode(P_ROJO, OUTPUT); // Definir el pin 11 como salida
  pinMode(P_VERDE, OUTPUT); // Definir el pin 7 como salida
  pinMode(PULSADOR, INPUT); // Definir el pin 3 como entrada
}
```

Ejemplo 5: Encender el led verde o el rojo dependiendo de si el valor del potenciómetro es menor o mayor de 512



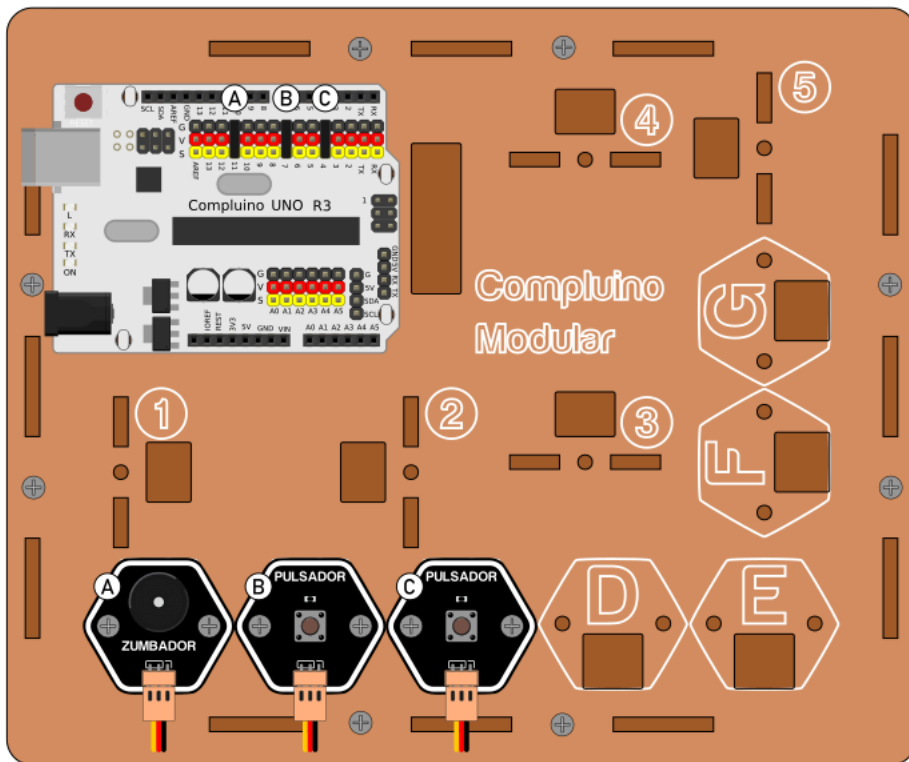
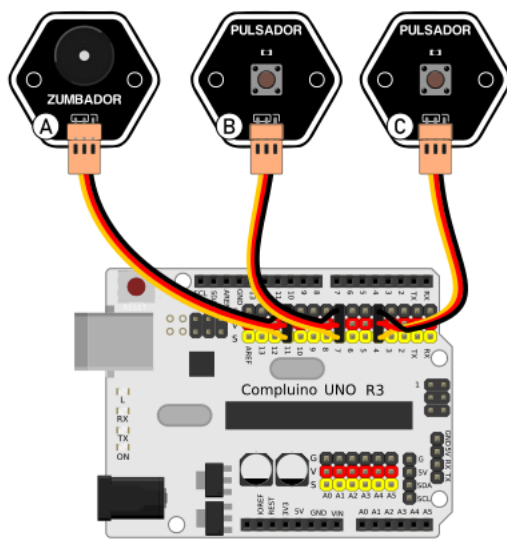
```
// Ejemplo 5: encender el led verde o el rojo dependiendo de si el valor del potenciómetro es menor o mayor de 512
#define POT A5
#define LEDA 7
#define LEDB 4
int lectura_pot = 0;
void setup()
{
  pinMode(LEDA, OUTPUT); // Definir el pin 7 como salida
  pinMode(LEDB, OUTPUT); // Definir el pin 4 como salida
}
void loop()
{
  lectura_pot = analogRead(POT); //Guardar el valor leído en el pin A5 en una variable
  if (lectura_pot >= 512) // Comparar la lectura del potenciómetro con un valor de corte
  {
```

Ejemplo 6: controlar el brillo de un led con la LDR



```
// Ejemplo 6: controlar el brillo de un led con la LDR.
#define LDR A5
#define LED 6
int lectura_ldr = 0;
int brillo = 0;
void setup()
{
  pinMode(LED, OUTPUT);
}
void loop()
{
  lectura_ldr = analogRead(LDR); //Guardar el valor leído en el pin A5 en una variable
  brillo = map(lectura_ldr, 0, 1023, 0, 255); //Convertir el valor a un rango de 0 a 255
  analogWrite(LED, brillo);
}
```

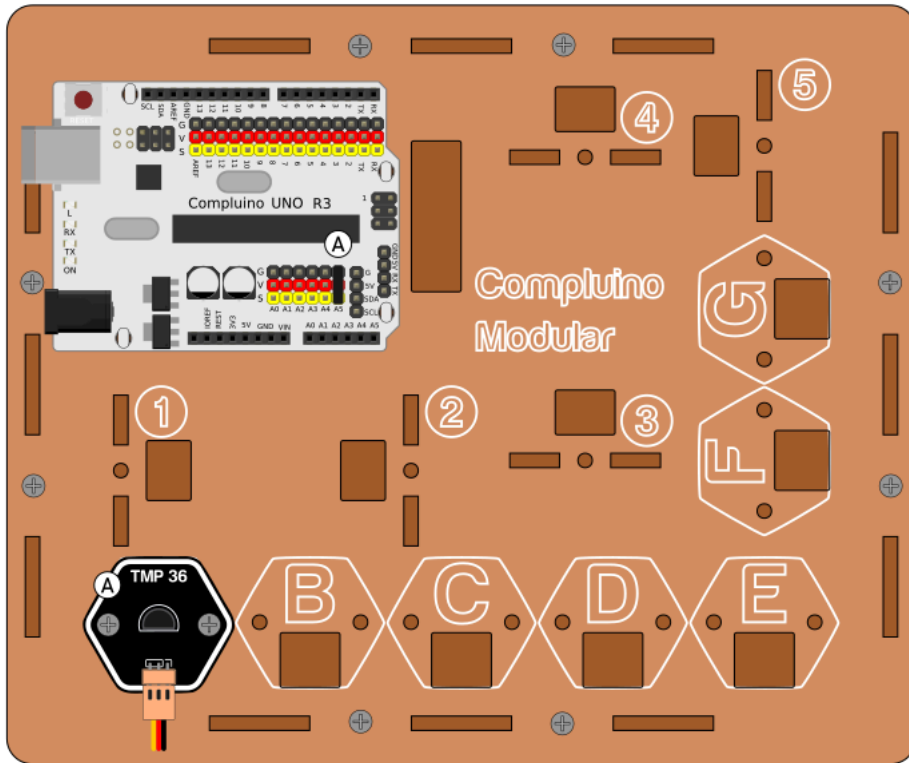
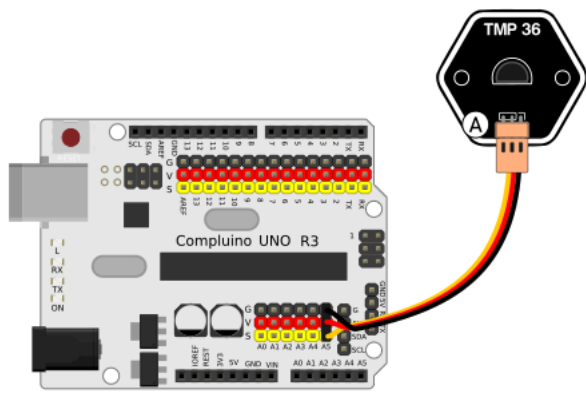
Ejemplo 7: dos pulsadores activan tonos diferentes en el zumbador



// Ejemplo 7: dos pulsadores activan tonos diferentes en el zumbador

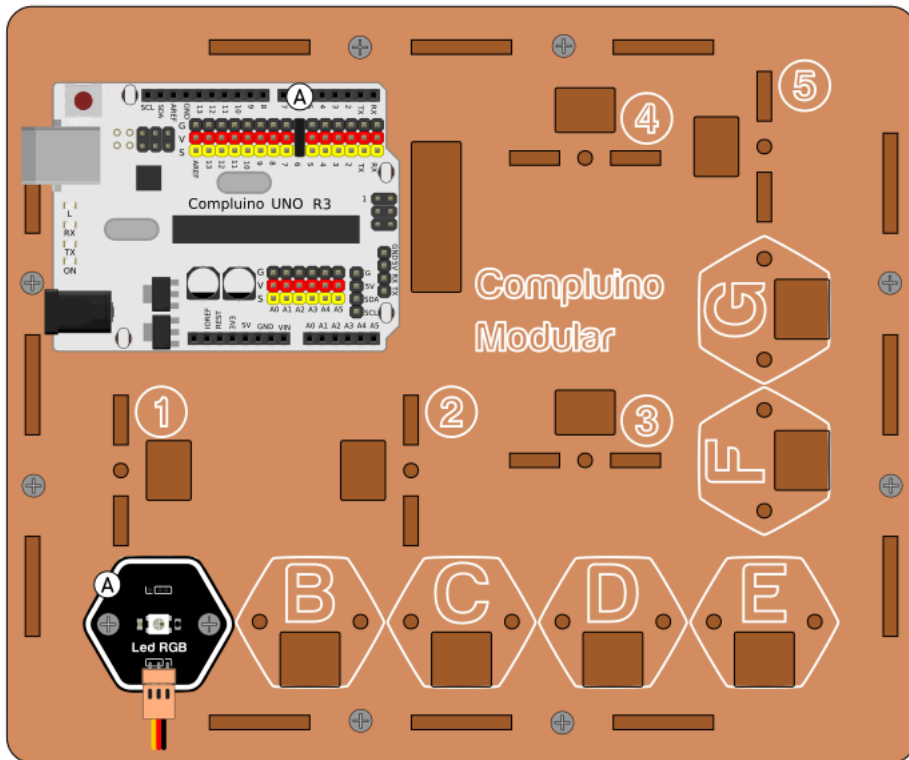
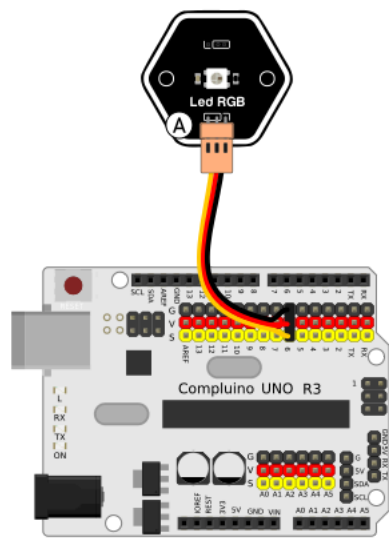
```
#define PULSADORB 7
#define PULSADORC 4
#define ZUMBADOR 11
void setup()
{
  pinMode(PULSADORB, INPUT);
  pinMode(PULSADORC, INPUT);
  pinMode(ZUMBADOR, OUTPUT);
  tone(ZUMBADOR, 293); //Nota RE
  delay(200);
  tone(ZUMBADOR, 440); //Nota LA
  delay(200);
  noTone(ZUMBADOR);
}
void loop()
```

Ejemplo 8: medir la temperatura y mostrarla en el monitor serie



```
// Ejemplo 8: medir temperatura y mostrarla en el monitor serie
#define TEMP A5
float temperatura = 0, medida_sensor = 0, voltaje = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  medida_sensor = analogRead(TEMP); //Actualizamos el valor del sensor de temperatura
  voltaje = medida_sensor * 5 / 1024; //convierte cuentas a voltios
  temperatura = (voltaje * 100) - 50; //convierte de voltios a grados C°
  Serial.print("La temperatura actual es: ");
  Serial.print(temperatura);
  Serial.println(" grados");
  delay(500);
}
```

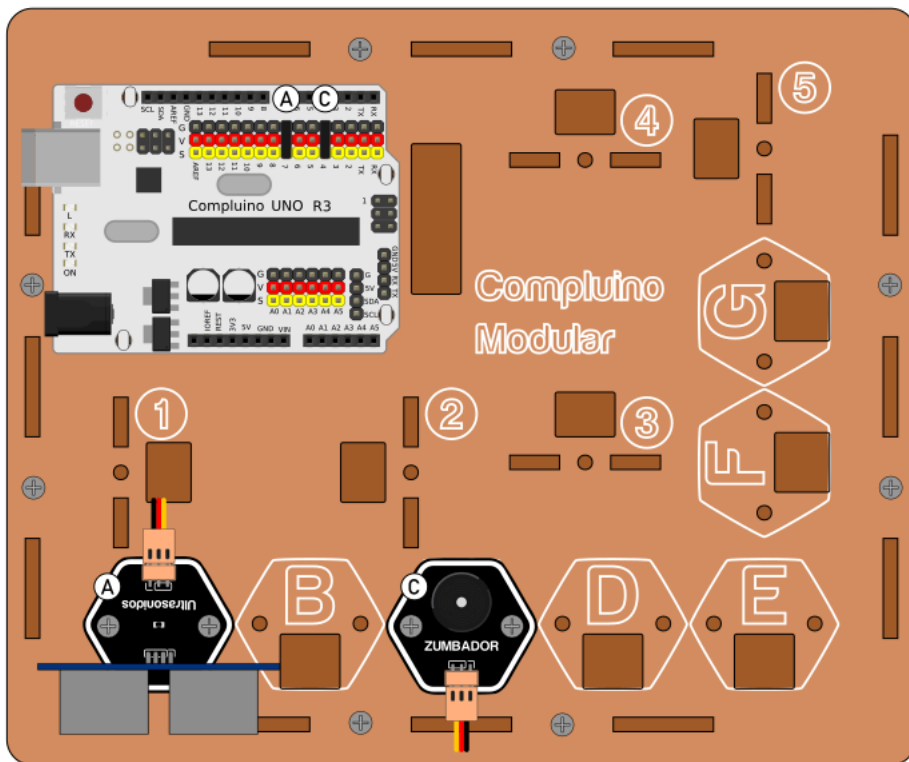
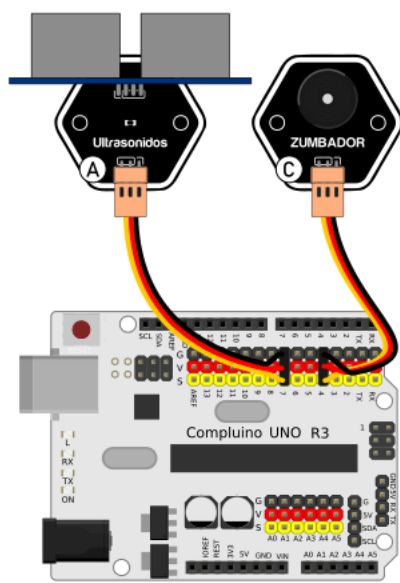
Ejemplo 9: secuencia de ocho colores con el led RGB y la librería Neopixel



```
// Ejemplo 9: 8 colores con el led RGB y la librería de Neopixel
#include <Adafruit_NeoPixel.h>
#define PIN_RGB 6 //Pin del rgb
Adafruit_NeoPixel rgb(1, PIN_RGB, NEO_GRB + NEO_KHZ400); //Configura 1 pixel RGB
void setup()
{
  rgb.begin();
}
void loop()
{
  rgb.setPixelColor(0, rgb.Color(255, 0, 0)); //Muestra el color rojo
  rgb.show();
  delay(2000);

  rgb.setPixelColor(0, rgb.Color(0, 255, 0)); //Muestra el color verde
  rgb.show();
}
```

Ejemplo 10: medir distancia con el sensor ultrasónico y utilizarlas para modificar el sonido del zombador



```
// Ejemplo 10: medir distancia con el HC SR04 y variar la frecuencia de pitido de un zumbador
#define PIN_US 7 //Pin del ultrasonido
#define ZUMBADOR 4 //Pin del zumbador
float tiempo; //Variable para guardar el tiempo
float distancia; //Variable para guardar la distancia
void setup()
{
  pinMode(ZUMBADOR,OUTPUT); //Configuramos el zumbador como salida
}
void loop()
{
  pinMode(PIN_US,OUTPUT); //Configuramos el pin como salida
  digitalWrite(PIN_US,LOW);
  delayMicroseconds(5);
  digitalWrite(PIN_US,HIGH); //Emitimos un echo por el ultrasonidos
  delayMicroseconds(10);
```

Etapas educativas

Plataforma

Educación Secundaria

Compluino

Productos relacionados