




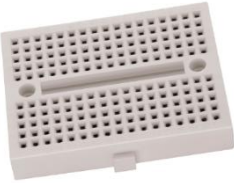



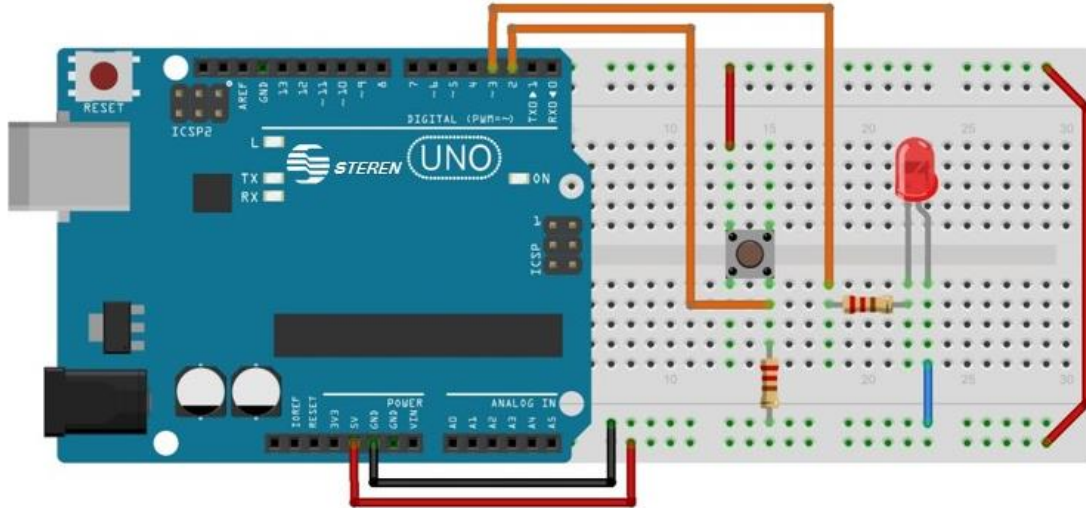
Practica 3. Lectura de un botón

Materiales		
Cantidad	Modelo	Foto
1	ARD-010 Arduino Uno	
1	USB-490 Cable USB A-B	
1	5/AZUL ULTRA LED de 5 mm, color azul	
1	R220 1/2 Resistencia de carbón, de 1/2 watt, al 5% de tolerancia, de 220 ohms	
1	ARD-310 Cables Dupont	
1	ARD-335 Mini Protoboard	
1	AU-101	

Objetivo General.

Conocer el funcionamiento del botón pulsador dentro de Arduino, mediante el encendido y apagado de un LED para conocer componentes que manipulen entrada y salida voltaje.

Diagrama de conexión



INSTRUCCIONES

1. Asignar los PINES dentro del código para manipular el LED con un botón pulsador.
2. Subir el código a la placa Arduino.
3. Conectar el botón pulsador, el LED y las resistencias, en la protoboard y la placa Arduino.
4. Verificar el funcionamiento de los componentes y la placa Arduino conectada.

Código

```
/*
```

```
PRACTICA No. 3 "Lectura de un botón"
```

```
*/
```

```
int boton = 2;    // Asignación del botón en el pin 2.
int led = 3;     // Asignación del led en el pin 3.
int lectura = 0; // Variable para guardar el estado (alto/bajo).

void setup() {
  pinMode (led, OUTPUT); // Declaración del pin como salida.
  pinMode (boton, INPUT); // Declaración del pin como entrada.
}

void loop() {
  lectura = digitalRead (boton); // Lee el estado del botón y lo guarda en la variable "lectura".
  if (lectura == HIGH) {        // Pregunta si el botón está en estado alto, o si está en estado bajo
    (presionado).
    digitalWrite (led, HIGH); // Si el botón está presionado, enciende el LED.
  }
  if (lectura == LOW) {        // Pregunta si el botón está en estado bajo, o no está en estado bajo
    (preciando).
    digitalWrite (led, LOW); // Si el botón no está presionado, se apaga el LED.
  }
}
```