Práctica 13. Implementación control por medio de mensajes del Arduino UNO

En esta práctica de implementación, consiste en controlar el encendido y apagado de un led por el envió de mensaje del Arduino UNO mediante el Monitor Serie del IDE de Arduino.

Objetivo

Controlar un actuador por medio de mensajes del Arduino UNO.

Equipo y Materiales

1 Arduino UNO

1 Protoboard

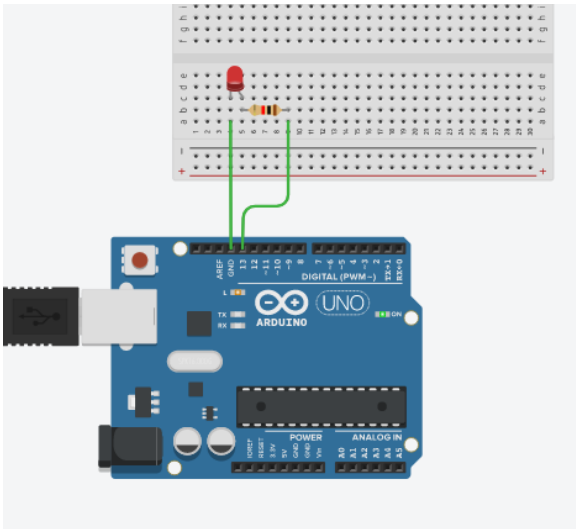
1 Resistencia de 330Ω o 220Ω

1 LED

IDE Arduino

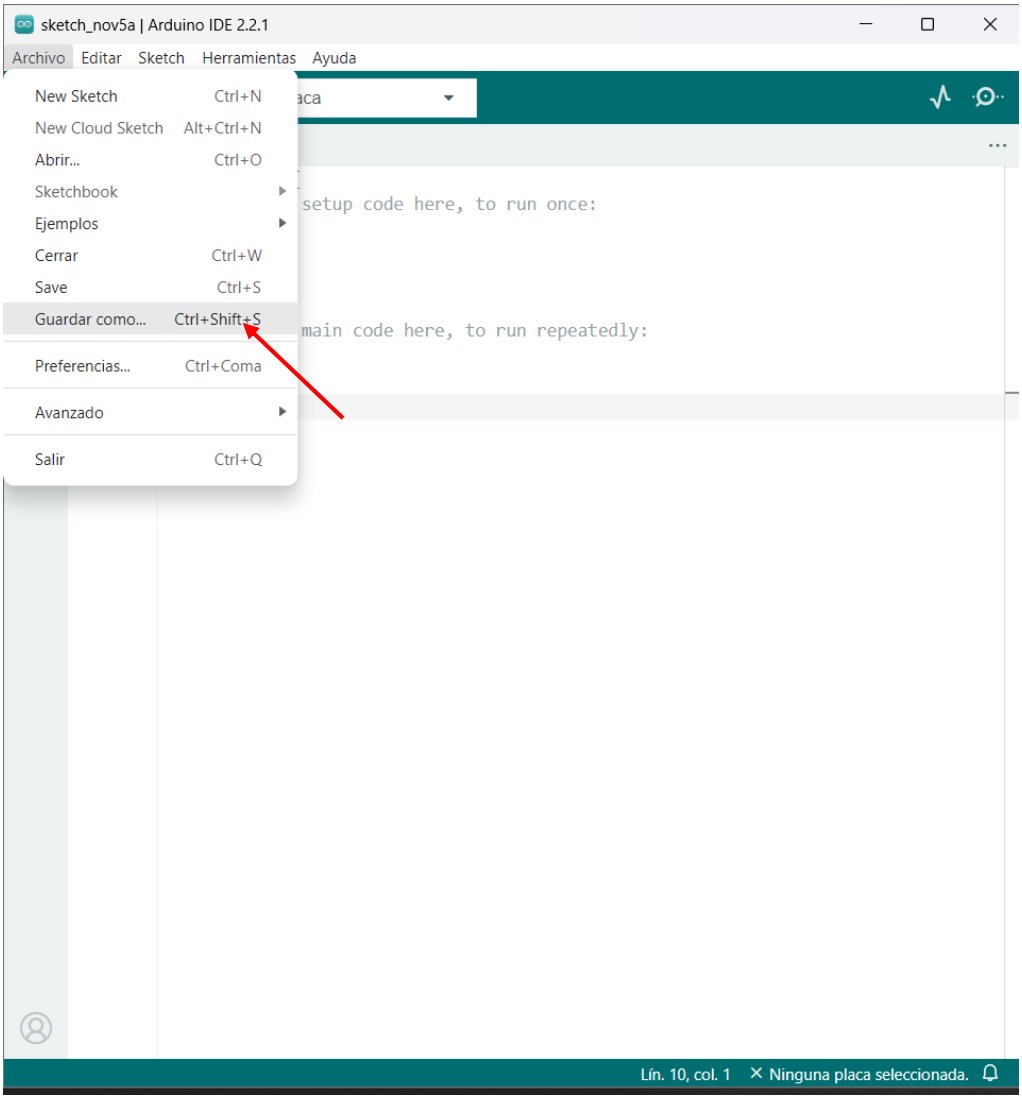
Procedimiento

PASO 1. Realice el circuito con sus componentes físicos como se muestra en el ejemplo:



Implementación

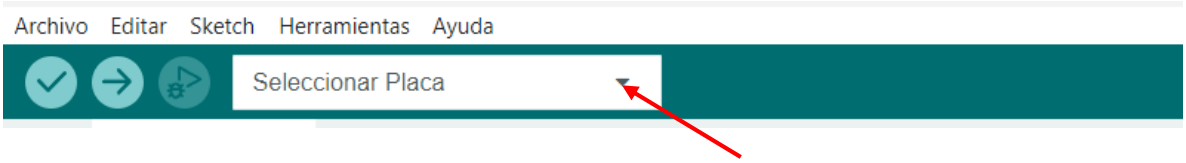
PASO 2. Diríjase al IDE Arduino. Siempre al iniciar deberá de nombrar su proyecto. En la parte superior de click sobre “Archivo” y en “guardar como”, borre el nombre predefinido y sustitúyalo por “blinkSerial2.ino”

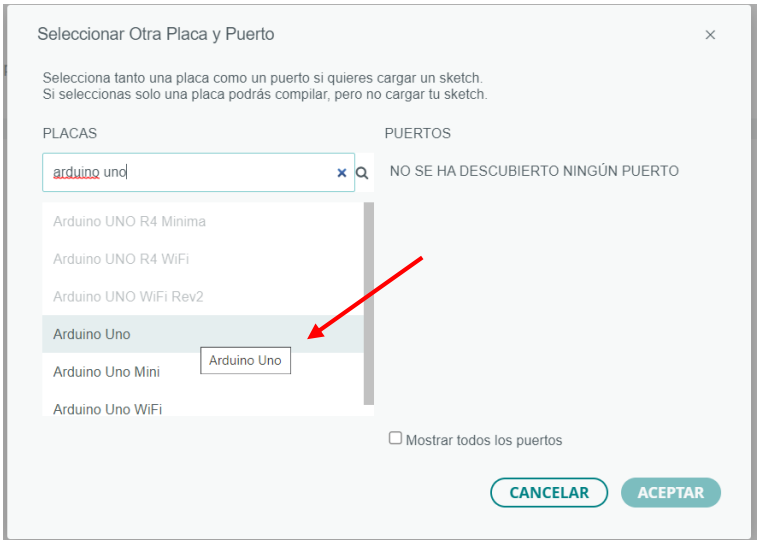


PASO 3. Una vez guardado el archivo, borre el contenido que le aparece y sustitúyalo por el siguiente código.

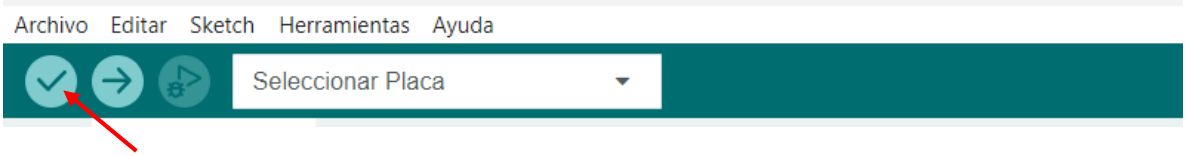
```
/*
* blinkSerial2.ino
*
* Este programa hace que el led de status del Arduino UNO o un led
* conectado al pin 13 y tierra, se encienda o apague controlado
* por comandos recibidos por el puerto serie
*/

#include <string.h>
const unsigned int PIN_LED = 13;
const unsigned int BAUD_RATE = 9600;

void setup() {
  // Establece el pin PIN_LED como de salida:
  pinMode(PIN_LED, OUTPUT);
  // Establece la velocidad de transmision del puerto serie al
  // valor BAUD_RATE
  Serial.begin(BAUD_RATE);
}

void loop() {
  char comando [6];
  // Si hay caracteres disponobles para lectura en el puerto serie
  if (Serial.available()>0){
    // Lee a lo mas 5 caracteres del puerto seie o hasta que se
    // presione la tecla Enter y los guarda en el arreglo comando
    int n = Serial.readBytesUntil('\n', comando, 5);
    // Todas las cadenas en c/c++ ternminan en el caracter de fin
    // de cadena, '\n'
    comando [n] = '\0';
    // Escribe el comando al puerto serie
    Serial.println(comando);

    // Si se lee el comando "on"
    if (!strcmp(comando, "on")){
        // Enciende el led conectado al pin PIN_LED
        digitalWrite(PIN_LED, HIGH);
        //Escribe al puerto serie
        Serial.println("LED encendido");
        }
    // Se lee el comando "off"
    else
    if(!strcmp(comando, "off")){
        // Apaga el led conectado al pin PIN_LED
        digitalWrite(PIN_LED, LOW);
        //Escribe al puerto serie
        Serial.println("LED apagado");
        }
    else{
        // Escribe al puerto serie
        Serial.print("Comnado desconocido: ");
        Serial.println(comando);
    }
  }
}

```

PASO 4. Seleccione la placa que está utilizando



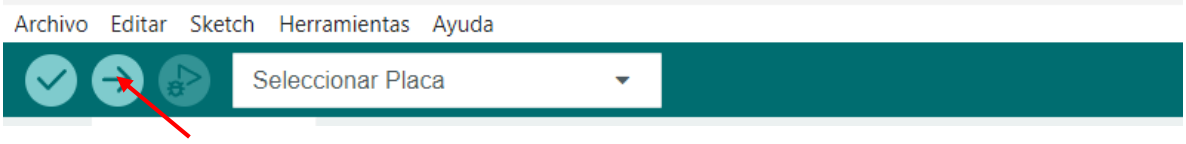


PASO 5. Verifique que su programa no tenga errores.



PASO 6. Conecte el cable del Arduino a este y al puerto USB de su computadora.

PASO 7. Cargue el programa



CUESTIONARIO

1. ¿Qué línea del código tengo que modificar para que el código de encendido y apagado sea diferente? Y ¿Cómo debería de cambiarlo?
2. ¿Para qué se utilizó \n en el programa de la práctica?
3. Quite el \n del programa y cárguelo de nuevo a la placa, ¿Qué es lo que hace diferente?