

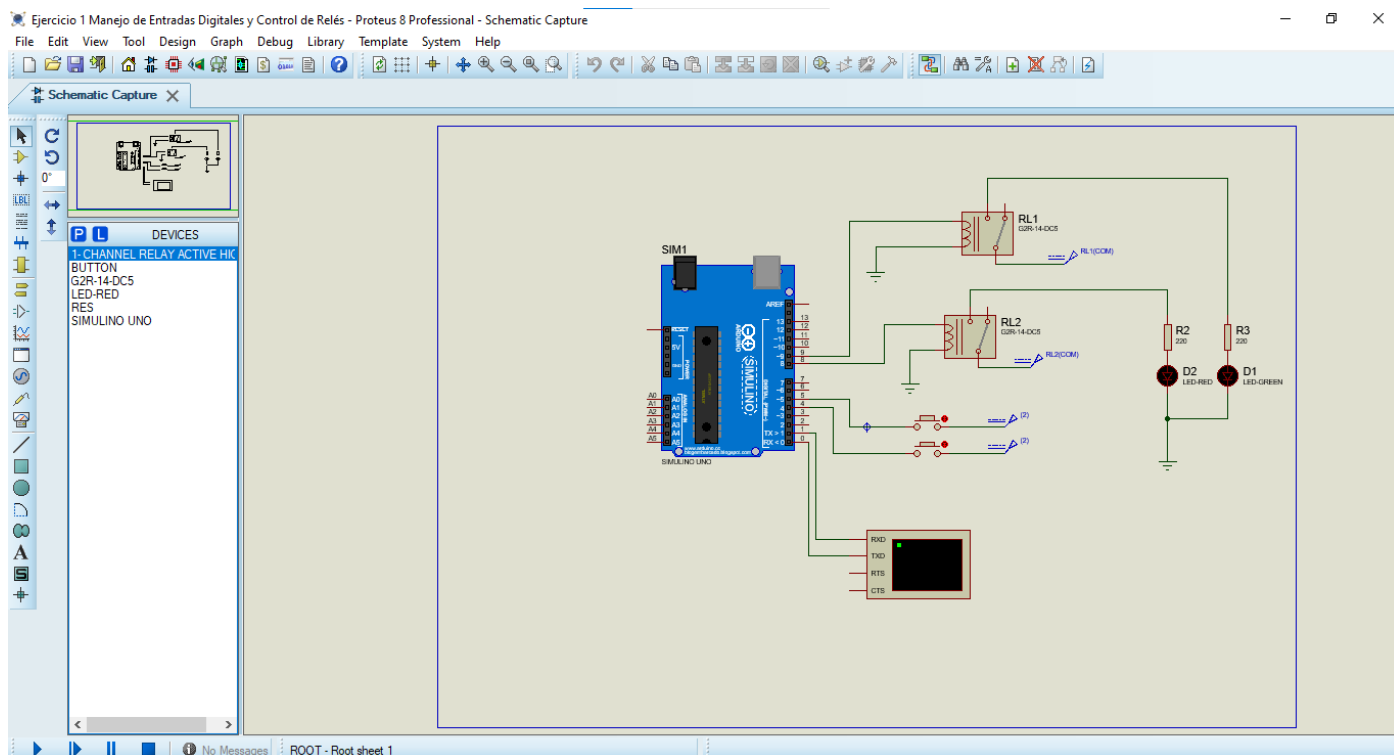
Ejercicio 1: Manejo de Entradas Digitales y Control de Relés

Objetivos:

1. Configurar y leer entradas digitales en Arduino.
2. Enviar y recibir datos a través del monitor serie.
3. Controlar relés como salidas digitales basados en la entrada de datos del monitor serie.
4. Implementar un sistema de control básico en Proteus utilizando botones y relés.

Desarrollo:

- Configura el Arduino para leer los estados de dos botones conectados a dos pines digitales.
- Establece una comunicación serial para recibir comandos que controlen un grupo de relés.
- Los relés deberán estar conectados a otros pines digitales y controlarán luces LED simuladas en Proteus.
- Programa el Arduino para que, al recibir ciertos comandos a través del monitor serie, active o desactive los relés correspondientes.



Código:

Ejercicio_1_Manejo_de_Entradas_Digitales_y_Control_de_Rel_s

```
#include <Arduino.h>
int salida1 = 8;
int salida2 = 9;
int salida3 = 10;
int salida4 = 11;
int entrada1 = 4;
int entrada2 = 5;
int entrada3 = 6;
int entrada4 = 7;
int valorIngresado = 0; // Variable to store the entered value

void setup() {
  pinMode(salida1, OUTPUT);
  pinMode(salida2, OUTPUT);
  pinMode(salida3, OUTPUT);
  pinMode(salida4, OUTPUT);
  pinMode(entrada1, INPUT);
  pinMode(entrada2, INPUT);
  pinMode(entrada3, INPUT);
  pinMode(entrada4, INPUT);
  Serial.begin(9600); // Initialize serial communication at 9600 baud
  Serial.println("Sistema Ready:");
  Serial.println("1 Encendido rele 1:");
  Serial.println("2 Encendido rele 2:");
  Serial.println("3 Apagado rele 1:");
  Serial.println("4 Apagado rele 2:");
  Serial.println("esperando el ingreso de numero:"); // espera
}

void loop() {
  if (Serial.available() > 0) {

    valorIngresado = Serial.parseInt();

    Serial.print("numero ingresado: ");
    Serial.println(valorIngresado);
    digitalWrite(valorIngresado - 1, HIGH);

    // Clear the serial buffer
    while (Serial.available() > 0) {
      Serial.read();
    }
  }
  if(valorIngresado == 1 || entrada1 == HIGH){
    Serial.println(" encendiendo rele 1 ");
    digitalWrite(salida1, HIGH);
  } else if(valorIngresado == 3 || entrada3 == HIGH){
    Serial.println(" Apagando rele 1 ");
    digitalWrite(salida1, LOW);
  }
  if(valorIngresado == 2 || entrada2 == HIGH){
    Serial.println(" encendiendo rele 2 ");
    digitalWrite(salida2, HIGH);
  } else if(valorIngresado == 4 || entrada4 == HIGH){
    Serial.println(" Apagando rele 2 ");
    digitalWrite(salida2, LOW);
  }
}
```

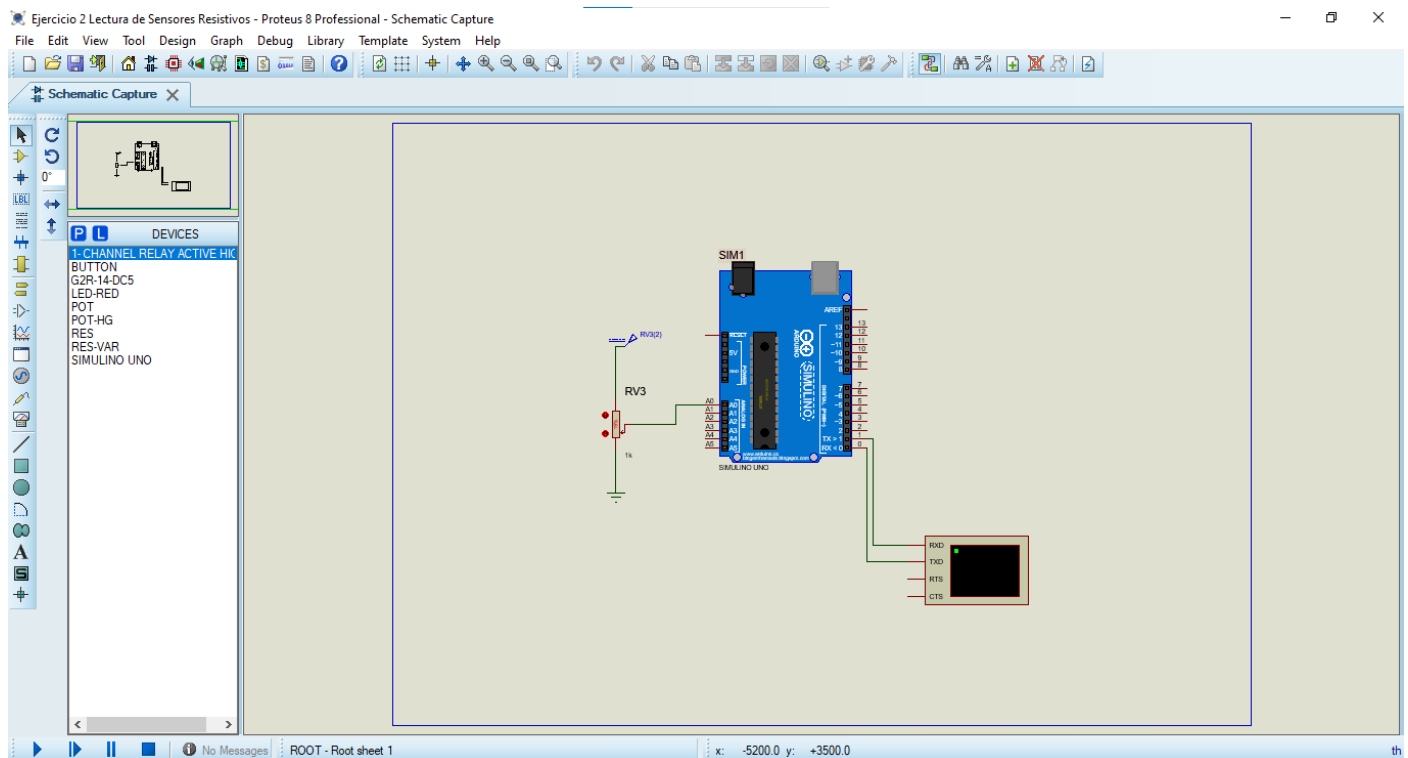
Ejercicio 2: Lectura de Sensores Resistivos

Objetivos:

1. Aprender a leer valores analógicos de un potenciómetro.
2. Convertir valores analógicos en datos digitales comprensibles.
3. Visualizar y monitorear cambios en el valor del potenciómetro a través del monitor serie.
4. Simular la conexión y lectura de un sensor resistivo en Proteus.

Desarrollo:

- Conecta un potenciómetro a un pin analógico del Arduino.
- Utiliza la función `analogRead()` para leer los valores del potenciómetro.
- Envía los valores leídos al monitor serie para visualizarlos.
- Implementa un sistema de visualización en Proteus para representar los valores leídos del potenciómetro.



Código:

Ejercicio_2_Lectura_de_Sensores_Resistivos Arduino 1.8.19 (Windows Store 1.8.57.0)

Archivo Editar Programa Herramientas Ayuda

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  // Leer el valor del potenciómetro  
  int valorsensor = analogRead(A0);  
  
  // Mostrar los valores en el monitor serie  
  Serial.print("Valor analógico: ");  
  Serial.println(valorsensor);  
  
  delay(1000); // Esperar 100 milisegundos antes de la siguiente lectura  
}
```

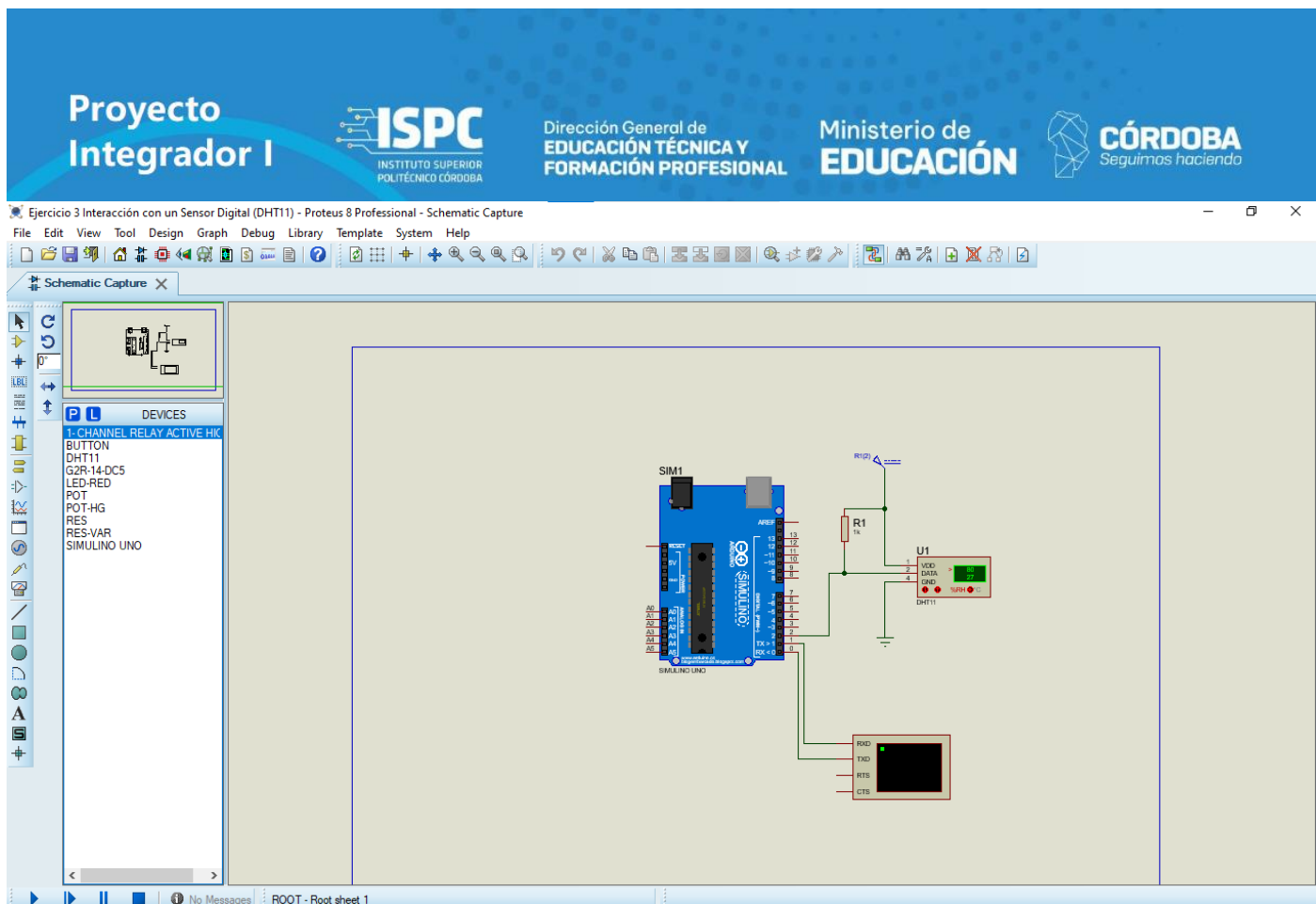
Ejercicio 3: Interacción con un Sensor Digital (DHT11)

Objetivos:

1. Integrar y configurar el sensor de temperatura y humedad DHT11 con Arduino.
2. Leer datos digitales de un sensor ambiental.
3. Interpretar y procesar datos de temperatura y humedad.
4. Mostrar los datos procesados en el monitor serie y simular esta configuración en Proteus.

Desarrollo:

- Conecta el sensor DHT11 a un pin digital en el Arduino.
- Utiliza una biblioteca adecuada para leer los datos de temperatura y humedad del sensor.
- Programa el Arduino para enviar estos datos al monitor serie, mostrando la temperatura y la humedad ambiental.
- Configura el simulador Proteus para modelar esta interacción, utilizando representaciones visuales de los datos recibidos del DHT11.



Código:

Ejercicio_3_Interacci_n_con_un_Sensor_Digital_DHT11_pdsprj Arduino 1.8.19 (Windows Store 1.8.57.0)

Archivo Editar Programa Herramientas Ayuda

```
#include <DHT.h>

#define DHTTYPE DHT11 // DHT 11 (AM2302)

DHT dht(DHTPIN, DHTTYPE);

//Variables
int chk;
float hum; //humedad
float temp; //Stores temperature value

void setup()
{
  Serial.begin(9600);
  Serial.println(" Midiendo temperatura inicial :");
  dht.begin();
}

void loop()
{
  hum = dht.readHumidity();
  temp= dht.readTemperature();
  //muestra en serial monitor los valores de temperatura
  Serial.print("Humedad: ");
  Serial.print(hum);
  Serial.print(" % , Temperaruta: ");
  Serial.print(temp);
  Serial.println(" Celsius");
  delay(2000); //Delay 2 sec.
}
```

12

Arduino Uno en COM7