Trabajo Práctico Informática II

## Programación Orientada a Objetos  1. Simulación de Sensores con Clases:

***1.1 Crear una clase Sensor con atributos como valor, magnitud y métodos para leer  (leerMedición) y simular un valor de sensor (generarMedicion ).***

**1. Inclusión de la librería Arduino.h**

#include <Arduino.h>

* La librería Arduino.h contiene definiciones básicas para trabajar con Arduino, como analogRead, Serial, y constantes como HIGH y LOW.

**2. Definición de la clase Sensor**

La clase Sensor modela un sensor con atributos y métodos específicos.

**Atributos privados:**

private:

float valor; // Valor actual del sensor

String magnitud; // Tipo de magnitud que mide el sensor (ej. "Temperatura")

* valor: Almacena el valor actual que el sensor mide o simula.
* magnitud: Describe la magnitud física que mide el sensor (ejemplo: temperatura).

**Métodos públicos:**

Los métodos permiten interactuar con la clase.

**Constructor**

Sensor(String magnitudInicial) {

valor = 0.0;

magnitud = magnitudInicial;

}

* Se ejecuta al crear un objeto de tipo Sensor.
* Inicializa valor en 0.0 y asigna el texto recibido como magnitudInicial al atributo magnitud.

**Método leerMedicion**

float leerMedicion(int pin) {

int lectura = analogRead(pin);

valor = (lectura \* 5.0 / 1023.0 - 0.5) \* 100; // Conversión del TMP36

return valor;

}

* **analogRead(pin)**: Lee un valor analógico (entre 0 y 1023) del pin indicado.
* **Conversión del TMP36**: Transforma el valor analógico a temperatura en grados Celsius:
  + (lectura \* 5.0 / 1023.0): Convierte el valor analógico a voltaje (0V - 5V).
  + - 0.5: Ajusta el rango del TMP36 (centra la referencia a 0°C en 0.5V).
  + \* 100: Escala el valor en voltios a grados Celsius.
* Devuelve la temperatura calculada.

**Método generarMedicion**

void generarMedicion() {

valor = random(20, 30); // Simula un valor entre 20°C y 30°C

}

* Usa la función random(min, max) para generar un número aleatorio entre 20 y 30, simulando una medición ficticia.

**Método mostrarMedicion**

void mostrarMedicion() {

Serial.print("Medición: ");

Serial.print(valor);

Serial.print(" ");

Serial.println(magnitud);

}

* Muestra el valor del sensor (valor) y su magnitud (magnitud) en el monitor serie.

**3. Creación de un objeto de la clase Sensor**

Sensor sensorTemperatura("Temperatura");

* Declara un objeto llamado sensorTemperatura que representa un sensor de temperatura. Este objeto utilizará todos los métodos definidos en la clase Sensor.

**4. Configuración inicial en setup**

void setup() {

Serial.begin(9600);

randomSeed(analogRead(0)); // Inicializa el generador de números aleatorios

}

* **Serial.begin(9600)**: Configura la comunicación serial a 9600 baudios.
* **randomSeed**: Establece una semilla para los valores aleatorios, utilizando una lectura analógica inicial como punto de partida.

**5. Bucle principal en loop**

void loop() {

// Leer el valor del sensor TMP36

float temp = sensorTemperatura.leerMedicion(A0);

sensorTemperatura.mostrarMedicion();

// Simular un valor de medición

sensorTemperatura.generarMedicion();

sensorTemperatura.mostrarMedicion();

delay(2000); // Esperar 2 segundos

}

**Paso 1: Leer un valor real del sensor**

float temp = sensorTemperatura.leerMedicion(A0);

* Llama al método leerMedicion, pasando el pin A0 como argumento.
* Calcula y actualiza el atributo valor del sensor.

**Paso 2: Mostrar el valor real**

sensorTemperatura.mostrarMedicion();

* Imprime en el monitor serie el valor medido y la magnitud (temperatura).

**Paso 3: Simular un valor ficticio**

sensorTemperatura.generarMedicion();

* Cambia el atributo valor a un número aleatorio.

**Paso 4: Mostrar el valor simulado**

sensorTemperatura.mostrarMedicion();

* Imprime el valor simulado junto con la magnitud.

**Paso 5: Pausa de 2 segundos**

delay(2000);

* Detiene la ejecución por 2 segundos para evitar lecturas continuas.

**6. Ejemplo de salida en el monitor serie**

makefile

Copiar código

Medición: 25.35 Temperatura

Medición: 22.00 Temperatura

* La primera línea muestra un valor real leído del sensor.
* La segunda línea muestra un valor simulado.