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>
```

- Proporciona funciones básicas de Arduino como `analogRead`, `Serial`, `delay`, etc.
- Necesaria cuando trabajamos fuera del IDE de Arduino (por ejemplo, con plataformas como PlatformIO).

---

2. Definición de la clase `Sensor`

Atributos privados

```
private:
    float valor;       // Almacena el valor actual medido o simulado
    String magnitud;   // Describe la magnitud física (ej. "Temperatura")
```

- **`valor`**: Almacena el resultado de una lectura o simulación.
- **`magnitud`**: Es un texto que describe qué mide el sensor (temperatura, presión, etc.).

Métodos públicos

Los métodos permiten interactuar con los atributos.

Constructor

```
Sensor(String magnitudInicial);
```

- Es el encargado de inicializar el objeto.
- En este caso, se inicializa:
  - `valor` a `0.0` (por defecto).
  - `magnitud` con el texto que se le pasa al constructor al crear el objeto.

leerMedicion

```
float leerMedicion(int pin);
```

- Lee un valor analógico desde el pin especificado.
- Convierte la lectura del sensor TMP36 a una temperatura en grados Celsius.

generarMedicion

```
void generarMedicion();
```

- Simula un valor ficticio (en este caso, un rango aleatorio entre 20 y 30).

mostrarMedicion

```
void mostrarMedicion();
```

- Imprime el valor y la magnitud en el monitor serie.

---

3. Implementación de los métodos

Constructor

```
Sensor::Sensor(String magnitudInicial) {
    valor = 0.0;
    magnitud = magnitudInicial;
}
```

- Se ejecuta automáticamente cuando se crea un objeto `Sensor`.
- Inicializa el valor a `0.0` y asigna el texto recibido a `magnitud`.

Método `leerMedicion`

```
float Sensor::leerMedicion(int pin) {
    int lectura = analogRead(pin); // Lee un valor analógico (0-1023)
    valor = (lectura * 5.0 / 1023.0 - 0.5) * 100; // Conversión TMP36
    return valor;
}
```

- **`analogRead(pin)`**: Captura un valor entre 0 y 1023 del pin analógico.
- **Conversión a temperatura**:
    1. **`lectura * 5.0 / 1023.0`**: Transforma la lectura en un voltaje entre 0 y 5V.
    2. **`- 0.5`**: Ajusta el rango del TMP36 (donde 0°C corresponde a 0.5V).
    3. **`* 100`**: Convierte el voltaje ajustado a grados Celsius.

Método `generarMedicion`

```
void Sensor::generarMedicion() {
    valor = random(20, 30); // Simula un valor aleatorio entre 20 y 30
}
```

- La función `random(min, max)` genera un número entero aleatorio entre `min` (inclusive) y `max-1` (por lo que aquí es entre 20 y 29).

Método `mostrarMedicion`

```
void Sensor::mostrarMedicion() {
    Serial.print("Medición: ");
    Serial.print(valor);
    Serial.print(" ");
    Serial.println(magnitud);
}
```

- Imprime en el monitor serie:
  - `valor`: El valor actual del sensor.
  - `magnitud`: La descripción de lo que mide.

---

4. Creación del objeto `sensorTemperatura`

```
Sensor sensorTemperatura("Temperatura");
```

- Declara un objeto llamado `sensorTemperatura` de la clase `Sensor`.
- La magnitud que mide este sensor es "Temperatura".

---

5. Configuración en `setup`

```
void setup() {
    Serial.begin(9600);          // Inicia la comunicación serial
    randomSeed(analogRead(0));   // Inicializa el generador de números aleatorios
}
```

- **`Serial.begin(9600)`**: Configura el puerto serie para comunicación a 9600 baudios.
- **`randomSeed`**: Mejora la aleatoriedad del generador `random` usando una lectura analógica inicial.

---

6. 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
}
```

El bucle se repite continuamente y realiza las siguientes acciones:

Paso 1: Leer un valor del sensor real

```
cpp
Copiar código
float temp = sensorTemperatura.leerMedicion(A0);
```

- Llama al método `leerMedicion` pasando el pin `A0`.
- Calcula el valor en grados Celsius y lo asigna a `temp`.

Paso 2: Mostrar el valor real

```
cpp
Copiar código
sensorTemperatura.mostrarMedicion();
```

- Imprime el valor real del sensor y su magnitud en el monitor serie.

Paso 3: Generar un valor simulado

```
cpp
Copiar código
sensorTemperatura.generarMedicion();
```

- Cambia el atributo `valor` a un número aleatorio entre 20 y 30.

Paso 4: Mostrar el valor simulado

```
cpp
Copiar código
sensorTemperatura.mostrarMedicion();
```

- Imprime el valor simulado junto con la magnitud.

Paso 5: Pausa

```
cpp
Copiar código
delay(2000);
```

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

---

7. Ejemplo de salida en el monitor serie

```
makefile
Copiar código
Medición: 23.45 Temperatura
Medición: 25.00 Temperatura
```

- La primera línea muestra el valor real leído del TMP36.
- La segunda línea muestra el valor simulado.

Resumen del flujo:

1. Configura el sensor y la comunicación serial.
2. En cada iteración:
    - Lee un valor real del sensor.
    - Lo muestra en el monitor serie.
    - Genera un valor simulado.
    - Lo muestra también.
3. Espera 2 segundos y repite.