TECNICATURA SUPERIOR

Telecomunicaciones

Proyecto Integrador

CAPA DE PROCESAMIENTO

ISPC INSTITUTO SUPERIOR POLITÉCNICO CÓRDOBA

Dirección General de EDUCACIÓN TÉCNICA Y FORMACIÓN PROFESIONAL — Ministerio de EDUCACIÓN — CÓRDOBA entre todos

Profesor: Gonzalo Vera

Explicación del código para el sensor BH1750:

1. **Bibliotecas Importadas:**

```
#include <Wire.h>
#include <BH1750.h>
#include <WiFi.h>
#include <HTTPClient.h>
```

- Wire.h: Habilita la comunicación I2C entre el ESP32 y el sensor BH1750.
- BH1750.h: Esta biblioteca proporciona funciones para interactuar con el sensor de luz BH1750, el cual mide la intensidad de la luz en lux.
- WiFi.h: Facilita la conexión del ESP32 a una red Wi-Fi.
- HTTPClient.h: Esta biblioteca permite enviar datos a través de solicitudes HTTP, como las que se envían a una API RESTful.

2. **Credenciales Wi-Fi:**

```
const char* ssid = "Tu_SSID";
const char* password = "Tu_CONTRASEÑA";
```

- Aquí se define el SSID (nombre) y la contraseña de la red Wi-Fi a la cual el ESP32 se conectará.

3. **URL de la API:**

```
const char* serverName = "https://api.gonaiot.com/plata/datos_dispositivos";
```

- La URL del servidor donde se enviarán los datos en formato JSON. Este es el endpoint que recibe los datos procesados por el ESP32.

4. **Inicialización del sensor BH1750:**

```
BH1750 lightMeter;
```

- Se crea un objeto lightMeter que representa el sensor de luz BH1750 y permite acceder a sus funciones.

5. **Configuración Inicial (setup()):**

```
void setup() {
  Serial.begin(115200);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando a WiFi...");
  }
  Serial.println("Conectado a WiFi");

  Wire.begin();
  if (!lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE)) {
    Serial.println("Error al inicializar BH1750");
    while (1);
  } else {
    Serial.println("BH1750 iniciado correctamente");
  }
}
```

- Serial.begin(115200);: Inicializa la comunicación serial para monitorear el progreso desde un monitor serie.
- WiFi.begin(ssid, password);: Intenta conectar el ESP32 a la red Wi-Fi usando las credenciales especificadas.
- Wire.begin();: Inicializa la comunicación I2C para poder leer datos del sensor.

- lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE);: Configura el BH1750 en el modo de alta resolución continua, donde el sensor mide continuamente la intensidad de luz y proporciona valores precisos.

6. **Lectura de la Intensidad de Luz en el Bucle Principal (loop()):**

```
void loop() {
  float lux = lightMeter.readLightLevel();

  Serial.print("Intensidad de luz: ");
  Serial.print(lux);
  Serial.println(" lx");

  sendData(lux);

  delay(60000); // Esperar 1 minuto antes de la próxima lectura
}
```

- lightMeter.readLightLevel();: Lee el nivel de luz actual medido por el sensor en lux.
- sendData(lux);: Llama a la función sendData() para enviar el valor de luz a la API.
- delay(60000);: Espera un minuto antes de repetir el proceso.

7. Función sendData() para Enviar Datos a la API:

```
void sendData(float lux) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(serverName);
    http.addHeader("Content-Type", "application/json");

    String jsonPayload = "{\"sensor\":\"bh1750\",\"intensidad_luz\":" + String(lux) +

    int httpResponseCode = http.POST(jsonPayload);

    if (httpResponseCode > 0) {
      String response = http.getString();
      Serial.println(httpResponseCode);
      Serial.println(response);
    } else {
      Serial.print("Error en la conexión: ");
      Serial.println(httpResponseCode);
    }

    http.end();
  } else {
    Serial.println("Error en la conexión WiFi");
  }
}
```

- if (WiFi.status() == WL_CONNECTED): Verifica si el ESP32 está conectado a la red Wi-Fi antes de intentar enviar datos.
- HTTPClient http;: Crea un cliente HTTP para gestionar las solicitudes a la API.
- http.begin(serverName);: Inicia una solicitud HTTP hacia la URL del servidor.
- http.addHeader("Content-Type", "application/json");: Añade un encabezado HTTP para indicar que el contenido que estamos enviando es en formato JSON.
- String jsonPayload = ...: Crea un string en formato JSON que contiene los datos del sensor BH1750, específicamente la intensidad de luz en lux.
- int httpResponseCode = http.POST(jsonPayload);: Envía una solicitud HTTP POST con el JSON. El código de respuesta se almacena en httpResponseCode.
    - Si la solicitud es exitosa (código > 0), imprime el código de respuesta y el contenido de la respuesta.
    - Si hay un error, se imprime el código de error.
- http.end();: Cierra la conexión HTTP.

Resumen General:

1. El ESP32 se conecta a la red Wi-Fi.
2. Se inicializa el sensor de luz BH1750.
3. En el bucle principal, se leen los datos de intensidad de luz en lux.
4. Se envían los datos a la API en formato JSON.
5. El ESP32 espera un minuto antes de realizar una nueva medición.

Este proceso asegura que los datos de luz medidos por el sensor BH1750 se envíen correctamente al servidor configurado.