#include <WiFi.h>

#include <WiFiUdp.h>

#include "time.h"

WiFi.h: Para manejar la conexión WiFi.

WiFiUdp.h: Para manejar la comunicación UDP.

time.h: Para manejar la obtención y manipulación del tiempo.

const char\* ssid = "iPhone de Roberto";

const char\* password = "12345678";

ssid: El nombre de la red WiFi.

password: La contraseña de la red WiFi.

const char\* ntpServer = "pool.ntp.org";

const long gmtOffset\_sec = 3600;

const int daylightOffset\_sec = 3600;

ntpServer: El servidor NTP que se utilizará para obtener la hora.

gmtOffset\_sec: El offset de la hora GMT en segundos (3600 segundos = 1 hora).

daylightOffset\_sec: El offset adicional para el horario de verano en segundos (3600 segundos = 1 hora).

const char\* serverIP = "10.211.55.3";

const int serverPort = 21;

serverIP: La dirección IP del servidor.

serverPort: El puerto del servidor.

WiFiUDP udp;

Declara un objeto WiFiUDP llamado udp que se usará para la comunicación UDP.

bool sendTime = false; // Variable para controlar el envío de la hora

Declara una variable booleana sendTime que controla si se envía la hora o no.

void sendTimeToServer(time\_t timeinfo) {

char buffer[20];

snprintf(buffer, sizeof(buffer), "%ld", timeinfo);

udp.beginPacket(serverIP, serverPort);

udp.write((uint8\_t\*)buffer, strlen(buffer));

udp.endPacket();

}

Declara un buffer char de 20 caracteres.

Usa snprintf para convertir el tiempo timeinfo en una cadena y guardarla en el buffer.

Usa udp.beginPacket para iniciar un paquete UDP dirigido a serverIP y serverPort.

Usa udp.write para escribir los datos del buffer en el paquete UDP.

Usa udp.endPacket para enviar el paquete UDP.

void printLocalTime() {

struct tm timeinfo;

if(!getLocalTime(&timeinfo)){

Serial.println("Failed to obtain time");

return;

}

char formattedTime[50];

strftime(formattedTime, sizeof(formattedTime), "%A, %B %d %Y %H:%M", &timeinfo);

Serial.println(formattedTime);

// Envía la hora local al servidor

sendTimeToServer(timeinfo.tm\_sec);

}

Declara una estructura tm llamada timeinfo para almacenar la información de tiempo.

Llama a getLocalTime(&timeinfo) para obtener la hora local. Si falla, imprime "Failed to obtain time" y sale de la función.

Declara un buffer char de 50 caracteres.

Usa strftime para formatear la hora en una cadena y guardarla en el buffer.

Imprime la hora formateada en el monitor serial.

Llama a sendTimeToServer para enviar la hora al servidor.

void setup() {

Serial.begin(115200);

// Conectar a WiFi

Serial.printf("Conectando a %s ", ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println(" CONECTADO");

// Inicializar y obtener la hora

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

}

Imprime un mensaje indicando que se está conectando a la red WiFi.

Llama a WiFi.begin(ssid, password) para iniciar la conexión a la red WiFi con las credenciales proporcionadas.

En un bucle while, comprueba el estado de la conexión con WiFi.status() != WL\_CONNECTED y espera 500 ms, imprimiendo un punto cada vez, hasta que se establezca la conexión.

Cuando la conexión se establece, imprime "CONECTADO".

configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer);

Llama a configTime(gmtOffset\_sec, daylightOffset\_sec, ntpServer) para configurar la hora utilizando el servidor NTP especificado y los offsets proporcionados.

void loop() {

if (Serial.available() > 0) {

String command = Serial.readStringUntil('\n');

command.trim();

if (command.equals("start")) {

sendTime = true;

Serial.println("Envío de hora iniciado.");

} else if (command.equals("stop")) {

sendTime = false;

Serial.println("Envío de hora detenido.");

}

}

delay(1000);

if (sendTime) {

printLocalTime();

}

}

Comprueba si hay datos disponibles en el puerto serial.

Si hay datos disponibles, los lee hasta encontrar un salto de línea y los almacena en la variable command.

Usa command.trim() para eliminar espacios en blanco al principio y al final.

Si command es igual a "start", establece sendTime a true y imprime "Envío de hora iniciado."

Si command es igual a "stop", establece sendTime a false y imprime "Envío de hora detenido."

Espera 1 segundo (delay(1000)).

Si sendTime es true, llama a printLocalTime() para obtener y mostrar la hora local, y enviarla al servidor.