

Reporte Mensual de Tareas

Pasantía Profesional para
Ingeniería Electrónica

Luis Antonio Pérez Piedra

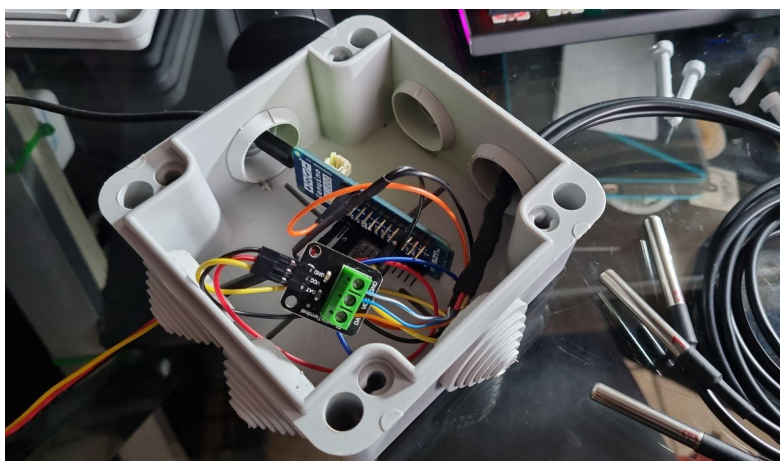
Periodo:

09 de Junio al 09 de Julio del 2021

El siguiente informe es una compilación de todos los avances hechos a nivel de proyectos y prototipos realizados en el FabLab desde el 09 de Junio del 2021 hasta el 09 de Julio del mismo año por Luis Antonio Pérez Piedra. El informe contiene una breve descripción de los cambios o mejoras que se le realizaron al sistema.

Sistema para el Monitoreo de Variables Críticas en el Proceso de Anidación de Tortugas Marinas en Peligro de Extinción

Encapsulado del Sistema



- Se realizó un encapsulado del sistema usando una caja plástica para conexiones en paredes ya que cuenta con los espacios para agregar más sensores en caso de ser necesario y con las medidas para guardar el Arduino MKR1000 y el módulo de los sensores de temperatura.

Simulación de Campo



- Para efectos de simulación, el sistema fue implementado para medir la temperatura de la tierra y la humedad de una maceta.
- Esto también fue realizado para analizar cómo el sistema era afectado por la distancia a la que se encontraba del dispositivo de Internet.
- Se encontró que al sistema perder la conexión a Internet, este no era capaz de volver a conectarse para enviar datos por lo que se tuvieron que realizar cambios al código.

Código del Sistema

```
#include <WiFi101.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SD.h>
#include <SPI.h>

OneWire ourWire(2);
DallasTemperature sensors(&ourWire);

DeviceAddress address1 = {0x28, 0x1C, 0x8F, 0x79, 0xA2, 0x0, 0x3, 0xFC};
DeviceAddress address2 = {0x28, 0x89, 0xC5, 0x79, 0xA2, 0x0, 0x3, 0xF2};
DeviceAddress address3 = {0x28, 0xD7, 0x47, 0x79, 0xA2, 0x0, 0x3, 0x87};

float EA = A0;
float humedad;
float calculo;
float porcentaje;

const char WEBSITE[] = "api.pushingbox.com"; //pushingbox API server
const String devid = "v00DB1FF6420FBAE"; // ID del dispositivo en Pushingbox
const char* MY_SSID = " "; // nombre del dispositivo de internet
const char* MY_PWD = " "; // contraseña del dispositivo de internet

int status = WL_IDLE_STATUS;
// si no se quiere usar DNS
// es mejor usar el IP numerico en vez del nombre del servidor:
// IPAddress server(X,X,X,X); // IP numerico de Google (no DNS)

void setup() {
  Serial.begin(9600);
  sensors.begin();
  pinMode(4, OUTPUT);
  digitalWrite(4, HIGH);
}

void printWifiStatus() {
  Serial.print("(SSID: ");
  Serial.print(WiFi.SSID());
  Serial.print(" / ");
  IPAddress ip = WiFi.localIP();
  Serial.print("Dirección IP: ");
  Serial.print(ip);
  Serial.print(" / ");
  long rssi = WiFi.RSSI();
  Serial.print("Fuerza de la señal (RSSI): ");
  Serial.print(rssi);
  Serial.println("dBm");
}
```

```
void loop() {
  // Establecer la conexion a internet:
  Serial.println("Conectandose al dispositivo: ");
  Serial.print(MY_SSID);
  status = WiFi.begin(MY_SSID, MY_PWD);
  delay(5000);
  Serial.println("\nConectado!!!\n");
  printWifiStatus();
  Serial.println("\n");
  humedad = analogRead(EA);
  calculo = (humedad-505)*100L/(952-505);
  porcentaje = 100-calculo;
  sensors.requestTemperatures();
  float temp1= sensors.getTempC(address1);
  float temp2= sensors.getTempC(address2);
  float temp3= sensors.getTempC(address3);
  Serial.print("Temperatura Sensor 1: ");
  Serial.print(temp1);
  Serial.println(" °C");
  Serial.print("Temperatura Sensor 2: ");
  Serial.print(temp2);
  Serial.println(" °C");
  Serial.print("Temperatura Sensor 3: ");
  Serial.print(temp3);
  Serial.println(" °C");
  Serial.print("Humedad: ");
  if (porcentaje > 100) {
    Serial.println("100%");
  } else if (porcentaje < 0) {
    Serial.println("0%");
  } else if (porcentaje > 0 && porcentaje < 100) {
    Serial.print(porcentaje);
    Serial.println("%");
  }
}

WiFiClient client;
if (client.connect(WEBSITE,80)){
  client.print("GET /pushingbox?devid=" + devid
    + "&TemperaturaS1=" + (String) temp1
    + "&TemperaturaS2=" + (String) temp2
    + "&TemperaturaS3=" + (String) temp3
    + "&Humedad=" + (String) porcentaje
    );
  client.println(" HTTP/1.1");
  client.print("Host: ");
  client.println(WEBSITE);
  client.println("User-Agent: MKR1000/1.0");
  client.println();
  Serial.println("Datos Enviados!!!\n");
  pinMode(5, OUTPUT);
  digitalWrite(5, HIGH);
}
```

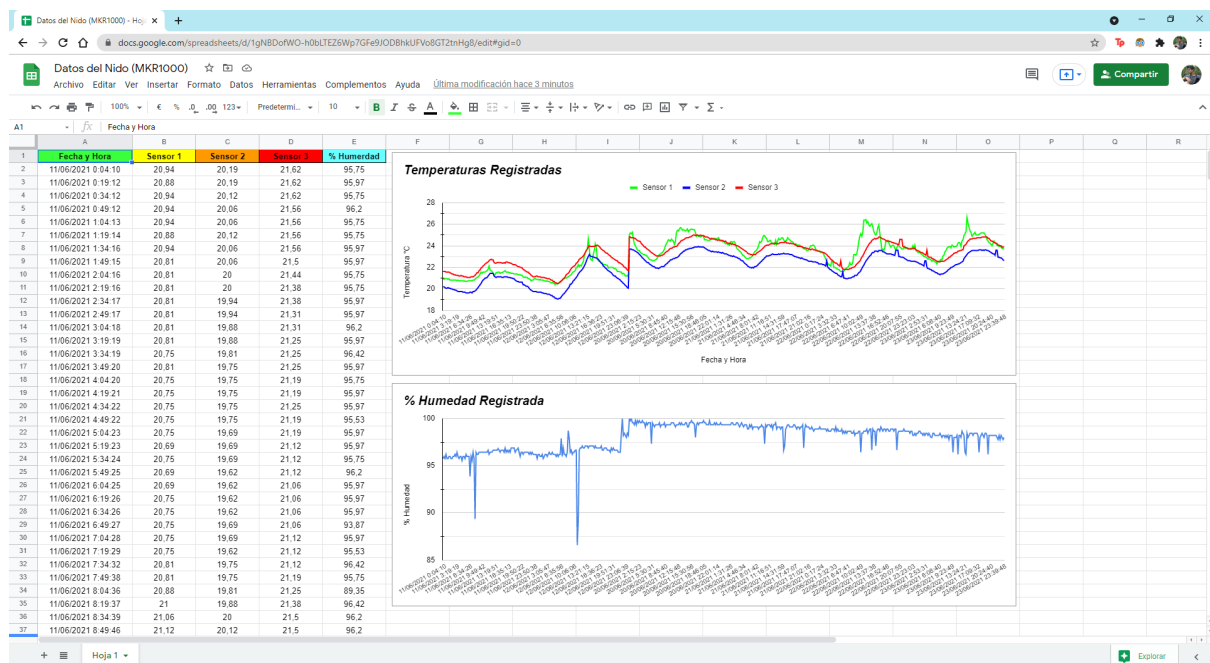
```

delay(5000);
client.stop();
digitalWrite(5, LOW);
delay(890000);
}
}

```

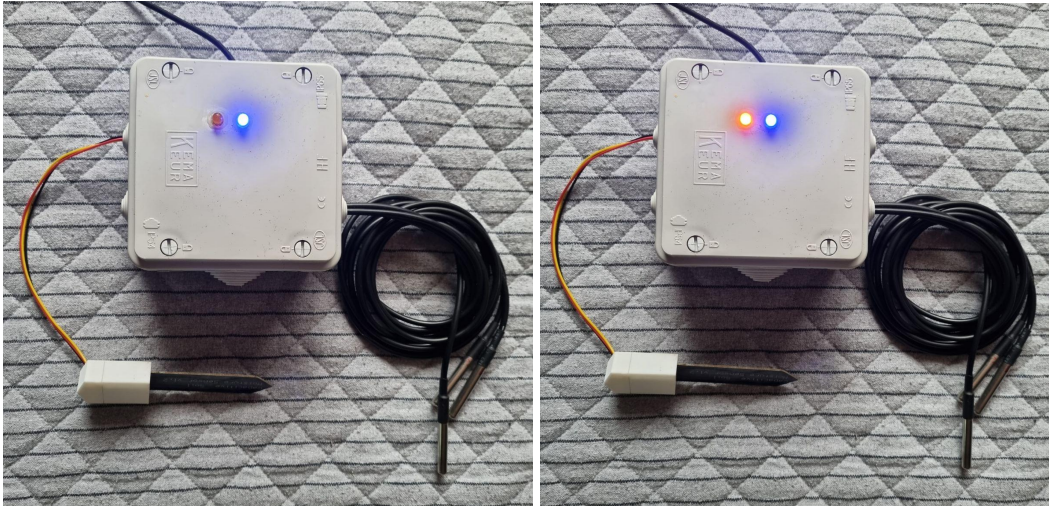
- Dos cambios fueron realizados en el código:
 - El primer cambio es que el sistema ahora cada 15 minutos se encarga de volver a establecer nuevamente la conexión con el dispositivo de internet a diferencia de antes que solo se establecía la conexión una vez.
 - El segundo cambio fue la implementación de comandos para controlar dos LEDs posicionados fuera del encapsulado que se encargan de mostrarle al usuario cuando el sistema se encuentra encendido y cuando datos han sido enviados al archivo de Google Sheets.

Documento de Google Sheets



- Se creó un respaldo de los datos recopilados anteriormente por el sistema y el original fue limpiado para así llevar un nuevo registro de los datos medidos durante la simulación de campo.

Encapsulado Final con LEDs



- El LED azul indica que el sistema se encuentra encendido mientras que el naranja indica cuando un dato fue enviado a través del servidor API de PushingBox.
- Luego de varias pruebas al reiniciar el dispositivo de internet para que así el Arduino perdiera la conexión y la volviera a establecer cuando el dispositivo de internet estuviera listo, se encontró que los cambios realizados al código fueron efectivos y el problema fue solucionado.