

Reporte Mensual de Tareas

Pasantía Profesional para
Ingeniería Electrónica

Luis Antonio Pérez Piedra

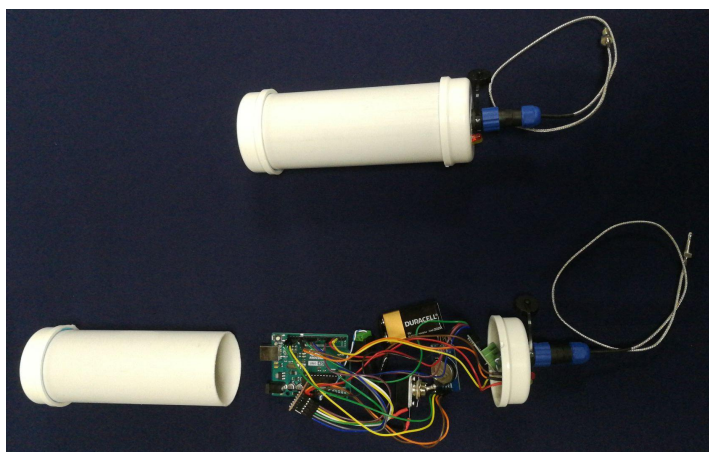
Periodo:

08 de Abril al 04 de Mayo 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 08 de Abril del 2021 hasta el 04 de Mayo del mismo año por Luis Antonio Pérez Piedra. El informe está dividido en dos partes, una para cada sistema realizado y sus respectivos progresos, así como una breve descripción de los cambios o mejoras que se le realizaron a dichos sistemas.

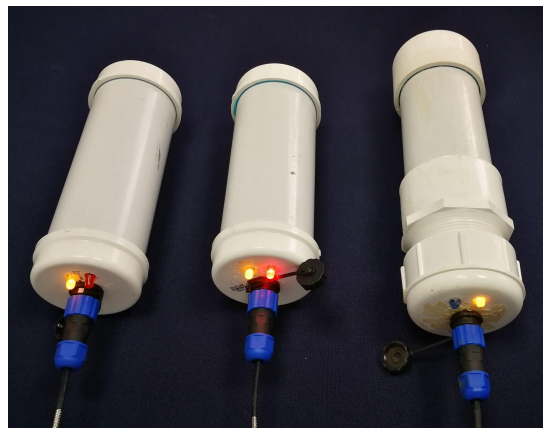
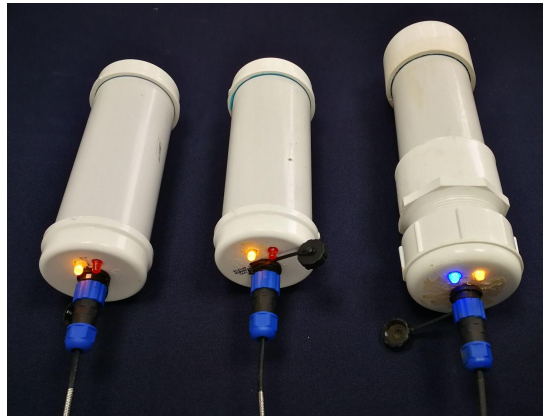
Sistema para Incendios Forestales

Realización de Sistemas Extra



- Se realizaron dos sistemas más para así tener un mejor monitoreo de la zona y además poder contrarrestar la pérdida de datos en caso de que alguno de los sistema falle por alguna razón.

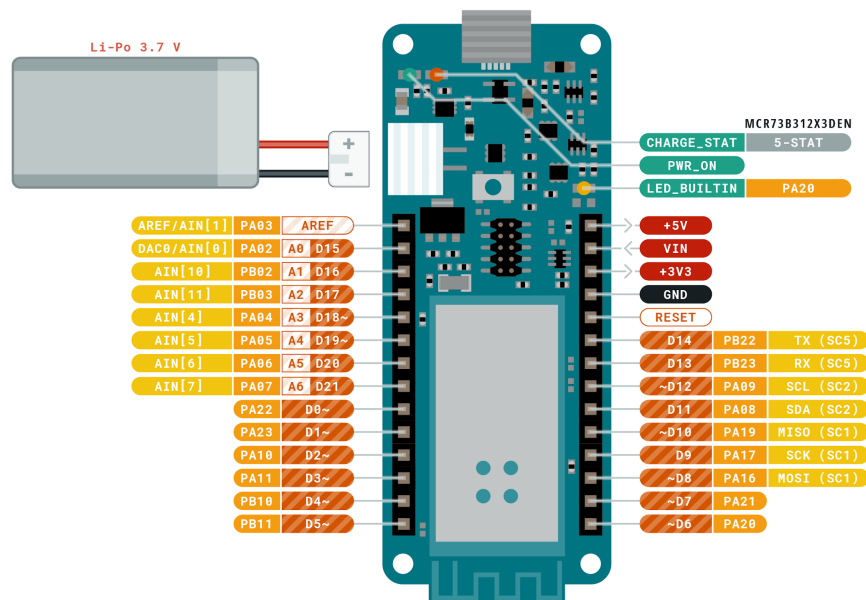
Sistemas Funcionando



- Se puede visualizar como las luces de cada sistema se encienden (de izquierda a derecha) cuando estos toman datos y los almacenan en sus respectivas tarjetas de memoria.
- Los dos sistemas nuevos son más compactos que el primero para así ahorrar tiempo a la hora de cavar para enterrar los sistemas.

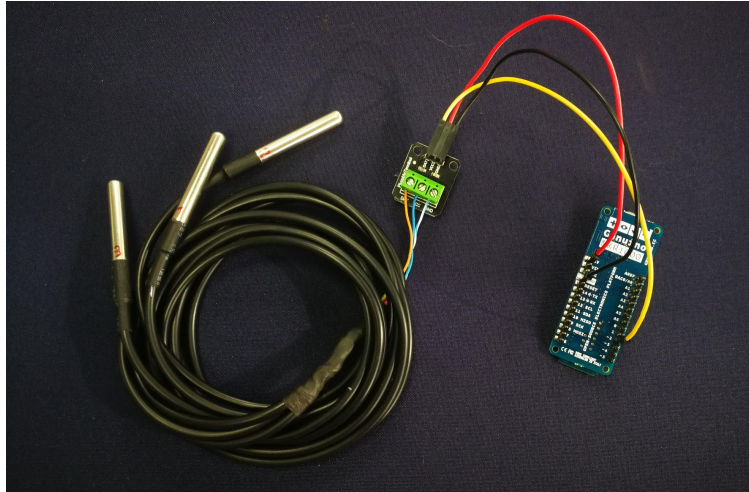
Sistema para Monitoreo de Nidos de Tortugas (MKR1000)

Diagrama del Sistema



- El circuito previo conocido como “Sistema de Múltiples Sensores de Temperatura” fue alterado.
- Por limitaciones de componentes de memoria, se decidió emplear el uso de una placa Arduino MKR1000 la cual se encarga de enviar datos por medio del internet.
- Se le adaptaron los mismos sensores de temperatura que al sistema previo y se agregó un sensor de humedad.

MKR1000 con Sensores de Temperatura



- El circuito previo conocido como “Sistema de Múltiples Sensores de Temperatura” fue alterado.
- Por limitaciones de componentes de memoria, se decidió emplear el uso de una placa Arduino MKR1000 la cual se encarga de enviar datos por medio del internet.

Código de Medición de Valores

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println(analogRead(A0));  
  delay(1000);  
}
```

- Con este simple código se miden los datos en dos casos: cuando el sensor esté completamente seco y cuando esté completamente mojado.
- Así se obtiene un valor máximo y un valor mínimo para calibrar mejor su medición.

Código de Porcentaje de Humedad

```
float ap = A0;
float valor;
float calculo;
float Porcentaje;

void setup() {

    Serial.begin(9600);
}

void loop() {
    valor = analogRead(ap);
    delay(1000);
    calculo = ((valor-330)*100L)/(650-330);
    Porcentaje = 100-calculo;
    Serial.print(" ");
    Serial.print(Porcentaje);
    Serial.println("%");
}
```

- Los valores máximos y mínimos medios previamente se cambian en la fórmula de “calculo” para así poder convertir estos valores en un porcentaje de humedad.

Código de Prueba de Todos los Sensores

```

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};

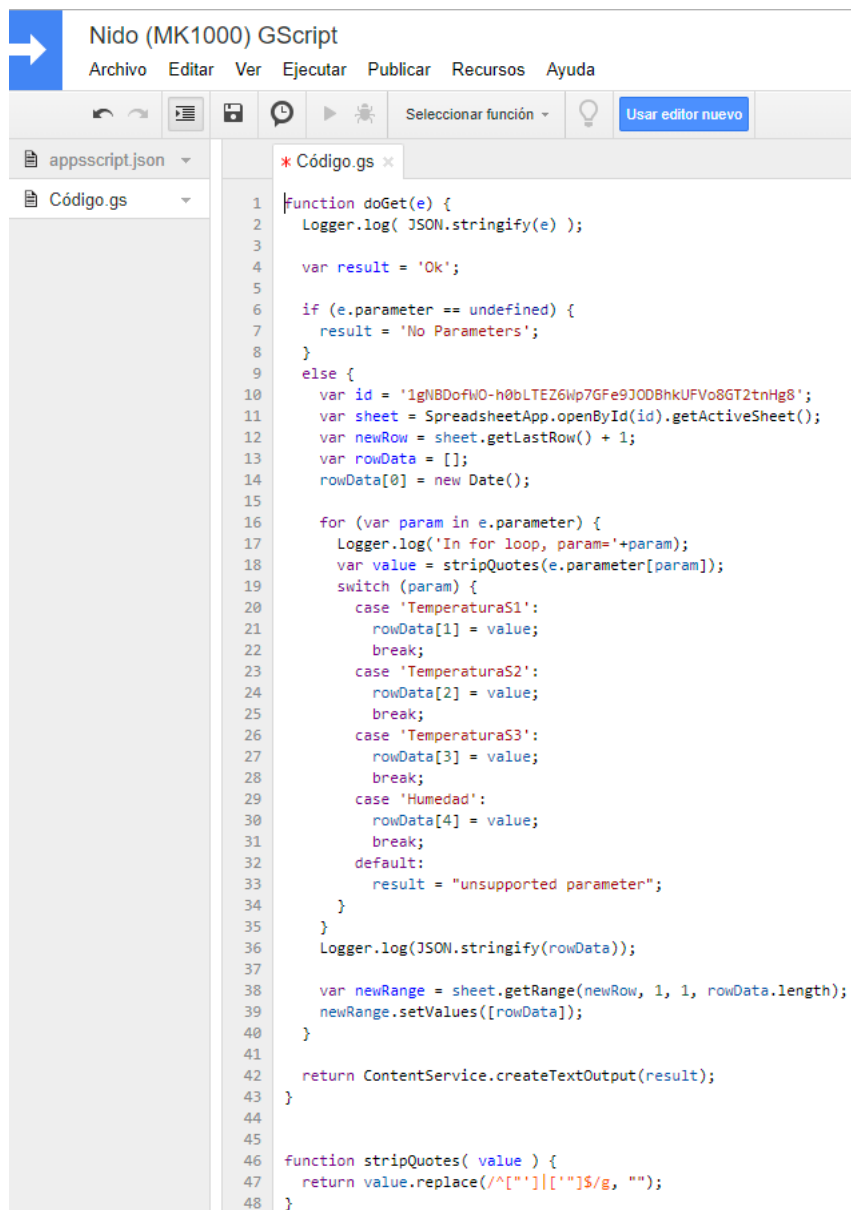
float ap = A0;
float valor;
float calculo;
float Porcentaje;

void setup() {
  Serial.begin(9600);
  sensors.begin();
}

void loop() {
  delay(1000);
  Serial.println(".....");
  valor = analogRead(ap);
  calculo = ((valor-330)*100L)/(650-330);
  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("%");
  }
}
  
```

- Se realizó un código de prueba en el que se comprobó el correcto funcionamiento de todos los sensores para así luego enviar los datos por medio de internet a un documento en Google Sheets.

Código Google Script



```

1 function doGet(e) {
2   Logger.log( JSON.stringify(e) );
3
4   var result = 'Ok';
5
6   if (e.parameter == undefined) {
7     result = 'No Parameters';
8   }
9   else {
10    var id = '1gNBDoFW0-h0bL7E6Wp7GFe9J0DBhkUFVo8GT2tnHg8';
11    var sheet = SpreadsheetApp.openById(id).getActiveSheet();
12    var newRow = sheet.getLastRow() + 1;
13    var rowData = [];
14    rowData[0] = new Date();
15
16    for (var param in e.parameter) {
17      Logger.log('In for loop, param='+param);
18      var value = stripQuotes(e.parameter[param]);
19      switch (param) {
20        case 'TemperaturaS1':
21          rowData[1] = value;
22          break;
23        case 'TemperaturaS2':
24          rowData[2] = value;
25          break;
26        case 'TemperaturaS3':
27          rowData[3] = value;
28          break;
29        case 'Humedad':
30          rowData[4] = value;
31          break;
32        default:
33          result = "unsupported parameter";
34      }
35    }
36    Logger.log(JSON.stringify(rowData));
37
38    var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
39    newRange.setValues([rowData]);
40  }
41
42  return ContentService.createTextOutput(result);
43 }
44
45
46 function stripQuotes( value ) {
47   return value.replace(/^["]|[""]$/g, "");
48 }
  
```

- Este código es el que permite la comunicación del Arduino con Google Sheets para así poder enviar los datos recolectados por los sensores.

Archivo generado por el arduino en la MicroSD

```
#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";
const String devid = "v00DB1FF6420FBAE";
const char* MY_SSID = "FABLAB";
const char* MY_PWD = "#$IICA2019$#";

int status = WL_IDLE_STATUS;

void setup() {
  Serial.begin(9600);
  sensors.begin();
  if (WiFi.status() == WL_NO_SHIELD) {
    while (true);
  }
  while (status != WL_CONNECTED)
  {
    Serial.print("Intentando conectarse al dispositivo de internet: ");
    Serial.println(MY_SSID);
    status = WiFi.begin(MY_SSID, MY_PWD);
    delay(5000);
  }
  Serial.println("\nConectado!!!\n");
  printWifiStatus();
}

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\n");
}
```


```

void loop() {
  delay(900000);
  Serial.println(".....\n");
  humedad = analogRead(EA);
  calculo = (humedad-330)*100L/(650-330);
  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");
    client.stop();
  }
}

```

- Este es el código que se encarga de enviar los datos recolectados por el Arduino y sus sensores a Google Sheets cada 15 minutos.

Datos en Google Sheets


Datos del Nido (MKR1000) ☆ 📁 ☁

Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda Última modificación hace 7 días

100% | € % .0 .00 123 | Predetermi... | 10 | B I S A | 📏 📐 📊 📈 📉 📉 📉 | 📄 📄 📄 📄 | 📄 📄 📄 📄 |

	A	B	C	D	E	F	G	H
1	Fecha/Hora	Temperaturas (°C)			% Humedad			
2		Sensor 1	Sensor 2	Sensor 3				
3	27/04/2021 9:59:15	24.38	23.88	25.06	0.20			
4	27/04/2021 10:14:15	24.12	23.81	24.88	0.61			
5	27/04/2021 10:29:17	25.00	24.69	25.94	0.61			
6	27/04/2021 10:44:15	24.25	24.19	24.94	1.02			
7	27/04/2021 10:59:13	24.31	24.38	25.00	3.46			
8	27/04/2021 11:14:10	23.94	23.69	24.62	3.25			
9	27/04/2021 11:29:09	24.50	24.25	25.12	6.71			
10	27/04/2021 13:58:59	24.25	23.62	25.00	0.61			
11	27/04/2021 14:13:58	24.31	23.69	25.06	2.24			
12	27/04/2021 14:28:58	24.31	23.62	26.69	0.20			
13	27/04/2021 14:43:57	24.62	23.88	25.25	0.00			
14	27/04/2021 14:58:56	24.44	23.88	25.12	99.39			
15	27/04/2021 15:13:55	23.31	22.88	24.19	99.19			
16								

- Impresión de los datos recolectados por los sensores en Google Sheets.

Pequeña Maqueta de Prueba



- Pequeña maqueta donde se pueden hacer unas cuantas pruebas.
- El sensor de humedad se puede visualizar a la izquierda.