

Reporte Mensual de Tareas

Pasantía Profesional para
Ingeniería Electrónica

Luis Antonio Pérez Piedra

Periodo:

09 de Septiembre al 09 de Octubre 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 Septiembre del 2021 hasta el 09 de Octubre del mismo año por Luis Antonio Pérez Piedra. El informe está dividido en tres partes, una para cada trabajo realizado y sus respectivos progresos, así como una breve descripción de los cambios, mejoras y sugerencias para dichos trabajos.

Sistema de Monitoreo de Incendios Forestales

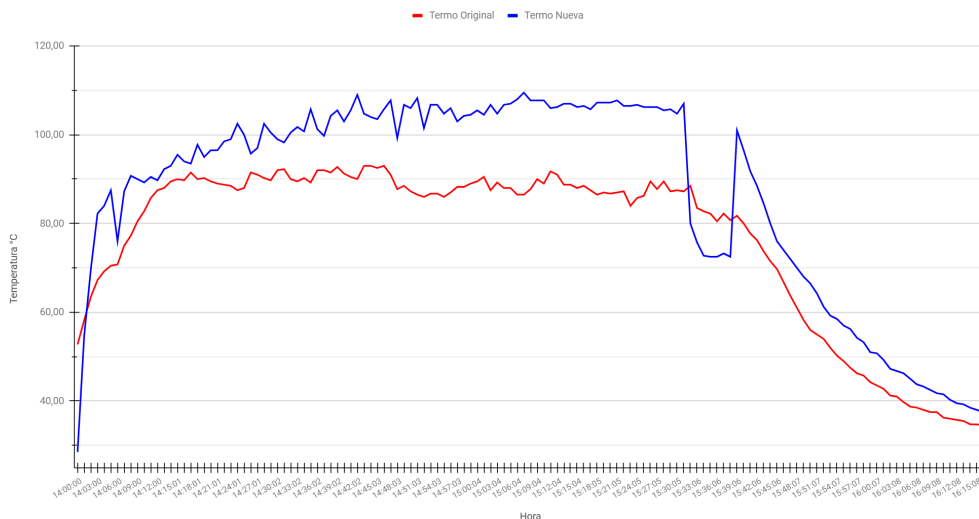
Prueba con Termocupla K Nueva



- Se utilizó el primer prototipo del Sistema para Incendios Forestales para realizar una prueba implementando un modelo de termocupla K distinto. Estas se colocaron en un disco eléctrico que se encuentra en la cocina del FabLab.

Resultados de Prueba Obtenidos

Temperaturas Registradas (09/09/2021)



- Se tomaron 137 datos en un periodo de 2 horas con 16 minutos. Por medio del archivo creado en la tarjeta MicroSD, se logró analizar los datos al crear una gráfica comparativa de la medición de ambas Termocuplas.

Cálculo de Precio del Sistema

Parte	Cantidad	Costo Unitario (\$)	Costo Total (\$)	Enlace
Arduino Uno	1	26,95	26,95	https://www.crcib
Módulo MicroSD	1	9,95	9,95	https://www.crcib
DS3231 (RTC)	1	5,95	5,95	https://www.crcib
MAX6675	1	9,95	9,95	https://www.crcib
Interruptor	1	2,25	2,25	https://cr.epaenl
Bateria 9V	1	3,75	3,75	https://cr.epaenl
9V a Barrel Jack	1	3,95	3,95	https://cr.epaenl
Femle Barrel Jack	1	0,95	0,95	https://www.crcib
Leds	2	0,35	0,70	https://www.crcib
Conector 3 Pins	1	5,95	5,95	https://www.crcib
Cables M-M (40)	1	2,49	2,49	https://www.crcib
Cables M-F (40)	1	2,49	2,49	https://www.crcib
Cables F-F (40)	1	2,49	2,49	https://www.crcib
			Total (\$)	77,82

- Se realizó una búsqueda en CrCibernetica de los componentes empleados para así tener una idea del costo de la parte electrónica del sistema.

Kits de Arduino de Microsoft

Componentes Faltantes

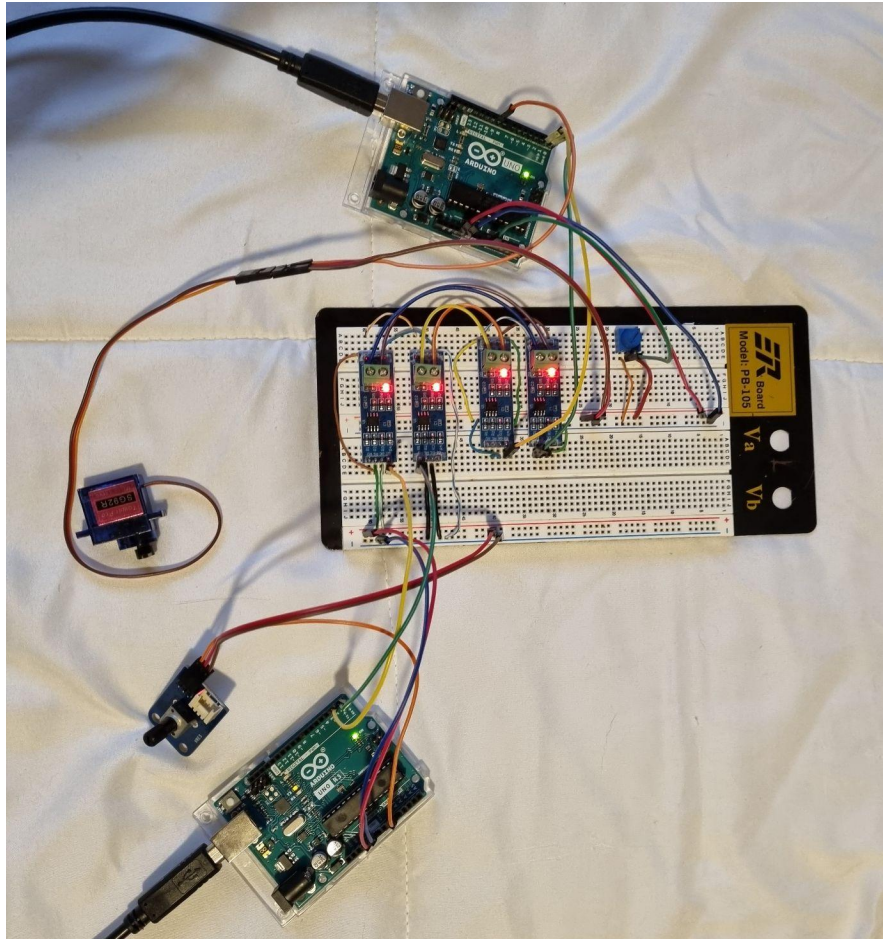


- Se agregaron los componentes que hacían falta para los kits donados por Microsoft.

Proyectos Universitarios

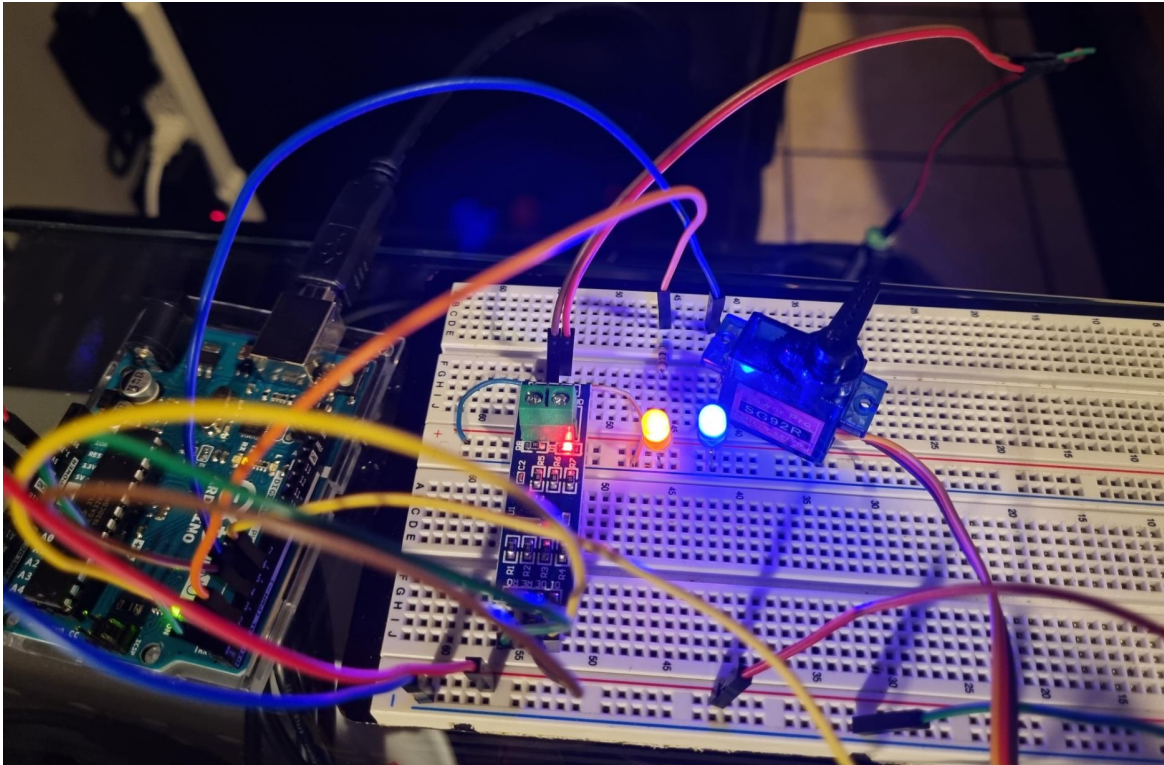
Del FabLab fueron prestados tres Arduinos UNO para la realización de dos proyectos universitarios para el curso de Redes Industriales para la Carrera de Ing. Electrónica en Automatización.

Comunicación Maestro-Eslavo entre Arduinos con Protocolo RS485



- Este proyecto consistía en establecer la comunicación entre dos Arduinos, uno haciendo la función de Maestro y otro la de Esclavo. El Maestro enviaba una señal de ángulo por medio de un potenciómetro y el Esclavo recibía esta señal para mover un servomotor de los 0 a los 180 grados.
- También el Esclavo contaba con un trimpot y cada vez que el Maestro solicitaba una lectura de este, se enviaba una señal que variaba entre 0 y 1024 dependiendo del valor de resistencia.
- <https://youtu.be/YGi4g3xjDYI>

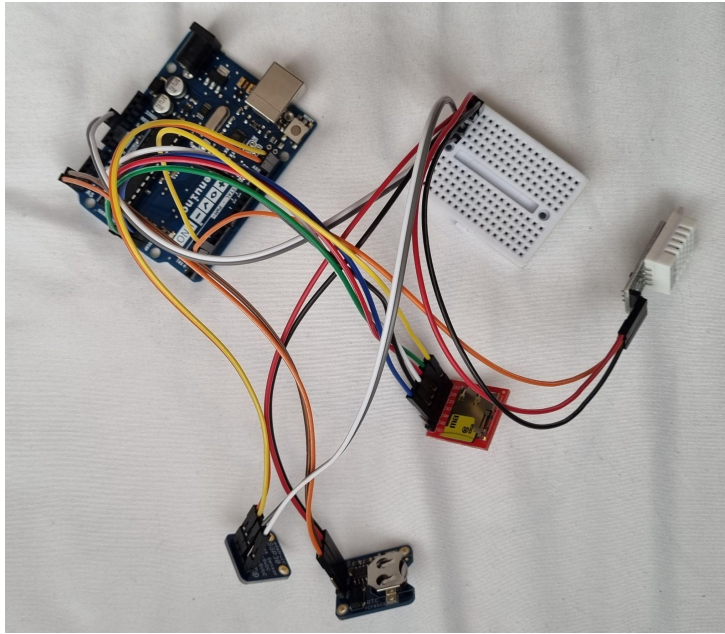
Comunicación con Arduino Esclavo usando Modbus



- Con este proyecto se pretendía establecer una comunicación entre una computadora actuando como Maestro y un Arduino como Esclavo usando el protocolo RS-485.
- Por medio de una aplicación realizada en LabVIEW y un convertidor de USB a RS485 es que la comunicación es posible entre la computadora y el Arduino. Desde dicha aplicación se establecen las 3 variables a manejar las cuales serían dos LEDs y un servomotor. El valor del servomotor se puede establecer entre 0 y 180, mientras que la señal que reciben los LEDs solo puede ser 0 o 1.
- Parte 1: <https://youtu.be/f1f8PUN2kf8>
- Parte 2: <https://youtu.be/6tYRqIcY4O4>

Sensor de Calidad de Aire

Prototipo con diferentes módulos



- Al sistema se le cambió el reloj de tiempo real del PCF8523 al DS3231 ya que el segundo produce una menor carga sobre la memoria del Arduino, problema que presentaba el primero ya que para su funcionamiento se necesitan muchas variables.

Código del Sistema

```
#include <SD.h>
#include <SPI.h>
#include <DHT.h>
#include <Wire.h>
#include <DS3231.h>
#include "Adafruit_SGP30.h"
```

```
Adafruit_SGP30 sgp;
```

```
#define DHTPIN 3
#define DHTTYPE DHT22
```

```
DS3231 rtc(SDA, SCL);
DHT dht = DHT(DHTPIN, DHTTYPE);
File archivo;

int CS_pin = 4;
float t = 0;
float h = 0;

uint32_t getAbsoluteHumidity(float temperature, float humidity) {
    const float absoluteHumidity = 216.7f * ((humidity / 100.0f) * 6.112f * exp((17.62f *
temperature) / (243.12f + temperature)) / (273.15f + temperature)); // [g/m^3]
    const uint32_t absoluteHumidityScaled = static_cast<uint32_t>(1000.0f * absoluteHumidity);
    // [mg/m^3]
    return absoluteHumidityScaled;
}

void setup() {
    Serial.begin(115200);
    pinMode(CS_pin, OUTPUT);
    rtc.begin();
    rtc.setTime(11, 50, 00);
    rtc.setDate(23, 9, 2021);
    dht.begin();

    if (!sgp.begin()){
        Serial.println("...No se encuentra el sensor...");
        while (1);
    }
    Serial.print("SGP30 serial #");
    Serial.print(sgp.serialnumber[0], HEX);
    Serial.print(sgp.serialnumber[1], HEX);
    Serial.println(sgp.serialnumber[2], HEX);

    while (!Serial) {
        ;
    }
    Serial.print("---Iniciando tarjeta SD---\n");
    if (!SD.begin(4)) {
        Serial.print("...No se pudo iniciar la tarjeta SD...");
        while (1);
    }
    archivo.println(" ");
    Serial.println("!!!Tarjeta SD iniciada con éxito!!!");
```



```
}

void loop() {
  Serial.println("-----");
  Serial.print("FECHA --> ");
  Serial.println(rtc.getDateStr());
  Serial.print("HORA --> ");
  Serial.println(rtc.getTimeStr());
  Serial.print("TEMPERATURA --> ");
  t = dht.readTemperature();
  Serial.print(t);
  Serial.println("°C");
  Serial.print("HUMEDAD --> ");
  h = dht.readHumidity();
  Serial.print(h);
  Serial.println("%");

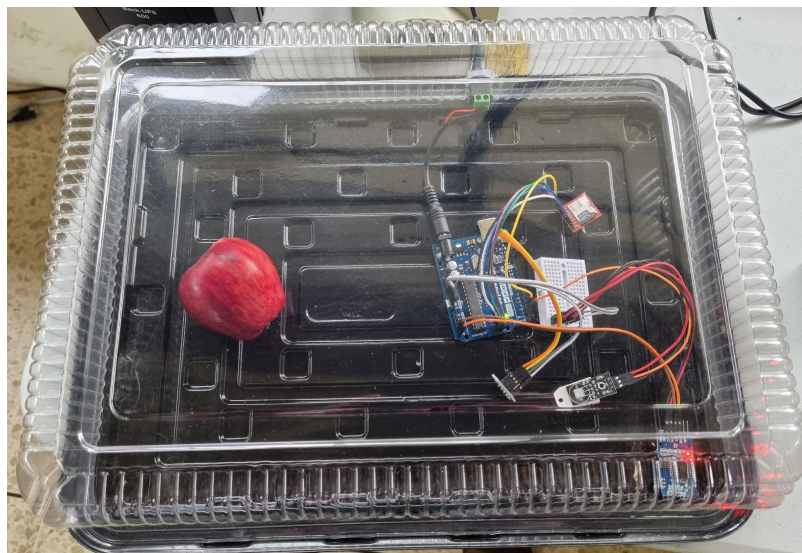
  if (! sgp.IAQmeasure()) {
    Serial.println("...Error al realizar medida...");
    return;
  }
  Serial.print("TVOC --> ");
  Serial.print(sgp.TVOC);
  Serial.println(" ppb");
  Serial.print("eCO2 --> ");
  Serial.print(sgp.eCO2);
  Serial.println(" ppm");

  if (! sgp.IAQmeasureRaw()) {
    Serial.println("...Error al realizar medidas raw...");
    return;
  }
  Serial.print("RAW H2 --> ");
  Serial.print(sgp.rawH2);
  Serial.println(" ");
  Serial.print("RAW ETHANOL --> ");
  Serial.print(sgp.rawEthanol);
  Serial.println(" ");

  archivo = SD.open("datos.txt", FILE_WRITE);
  if (archivo) {
    archivo.print(rtc.getTimeStr());
    archivo.print(" ");
    archivo.print(rtc.getDateStr());
```

```
archivo.print(",");  
t = dht.readTemperature();  
archivo.print(t);  
archivo.print(",");  
h = dht.readHumidity();  
archivo.print(h);  
archivo.print(",");  
archivo.print(sgp.TVOC);  
archivo.print(",");  
archivo.print(sgp.eCO2);  
archivo.print(",");  
archivo.print(sgp.rawH2);  
archivo.print(",");  
archivo.print(sgp.rawEthanol);  
archivo.println(" ");  
archivo.close();  
} else {  
  Serial.println("!!!Error al abrir el archivo!!!");  
}  
delay (600000);  
}
```

Monitoreo de Funcionamiento



- Se realizó un encapsulado para el sistema (el cual se programó para tomar datos cada 5 minutos) junto con una fruta para ver el comportamiento del SGP30 en un ambiente cerrado.