

Temperatura y Humedad con Arduino y Python

Este proyecto se basa en la obtención de datos de temperatura y humedad, mediante el uso del sensor DHT11 conectado a Arduino y el procesamiento de los mismos con Python.

Requisitos:

En cuanto a los requisitos, se emplearon los siguientes elementos:

- *Arduino Uno
- *Sensor DHT11
- *RTC DS1307
- *Protoboard
- *Cables de conexión
- *Cable USB.

Librerías:

Arduino:

- *RTCLib --> <https://github.com/adafruit/RTCLib>
- *DHT --> <https://github.com/adafruit/DHT-sensor-library>
- *Wire --> Gestor de librerías de Arduino.
- *Adafruit Unified Sensor --> Gestor de librerías de Arduino.

Python:

- *PySerial
- *csv
- *matplotlib

Códigos

Arduino:

A lo correspondiente al código, podemos hacer mención que el mismo se basa en la típica distribución de Arduino, dejando la importación de las librerías en la parte superior, seguida de las definiciones de pines y tipos de elementos, objetos y variables. En el bucle setup(), se inicializa la comunicación serial, el rtc y el sensor dht. A continuación se presenta uno de los segmentos más importantes del código:

```
In [ ]: while(tiempo == 0) {
        tiempo = Numerificacion(valorIngresado());
    }
```

El cual es un bucle, cuya función es la recibir el tiempo a esperar entre cada toma de datos. Esto se cumple gracias a la función valorIngresado()

```
In [ ]: String valorIngresado() {
        while(true) {
            if(Serial.available() > 0) {
                delay(20);
                String bufferString = "";

                while(Serial.available() > 0) {
                    bufferString += (char)Serial.read();
                }
                return bufferString;
            }
        }
    }
```

La cual espera y lee los valor ingresados por el usuario desde la terminal a través de python y la comunicación serial, almacenando los mismos para luego devolverlos en formato string, para luego ser transformados a valores del tipo entero, mediante la función numerificacion()

```
In [ ]: int Numerificacion(String numIngresado) {
        int numeroConvertido = numIngresado.toInt();
        return numeroConvertido;
    }
```

Una vez obtenido el intervalo de tiempo, el programa pasa al bucle loop(). El cual ejecuta solo dos funciones, la primera mostrarHora():

```
In [ ]: void mostrarHora() {
        Hora = rtc.now();

        Serial.print(Hora.hour());
        Serial.print(".");
        Serial.print(Hora.minute());
    }
```

La cual pregunta la hora actual al rtc y la envía al monitor serial y la función mostrarTempYHum():

```
In [ ]: void mostrarTempYHum() {
        float h = dht.readHumidity();
        float t = dht.readTemperature();

        Serial.print(" ");
        Serial.print(h);
        Serial.print(" ");
        Serial.print(t);
        Serial.println("");
    }
```

Que al igual que la anterior, toma los datos de temperatura y humedad provenientes del sensor y los envía al monitor serial.

Python:

Pasando a Python, el código es muy simple y se basa principalmente en funciones, las cuales son llamadas por el bloque principal del programa o por otras funciones.

En esta ocasión no se describirá la totalidad del código, ya que una de las principales virtudes de Python es que es un lenguaje con mucha claridad en la sintaxis, pero en cambio se describirán las principales partes del mismo.

Una de estas partes, es la función encargada de escribir los datos en un archivo .csv. Dicha función estará disponible solo si el usuario teclea "tomar" en la terminal, al inicio de la ejecución del código

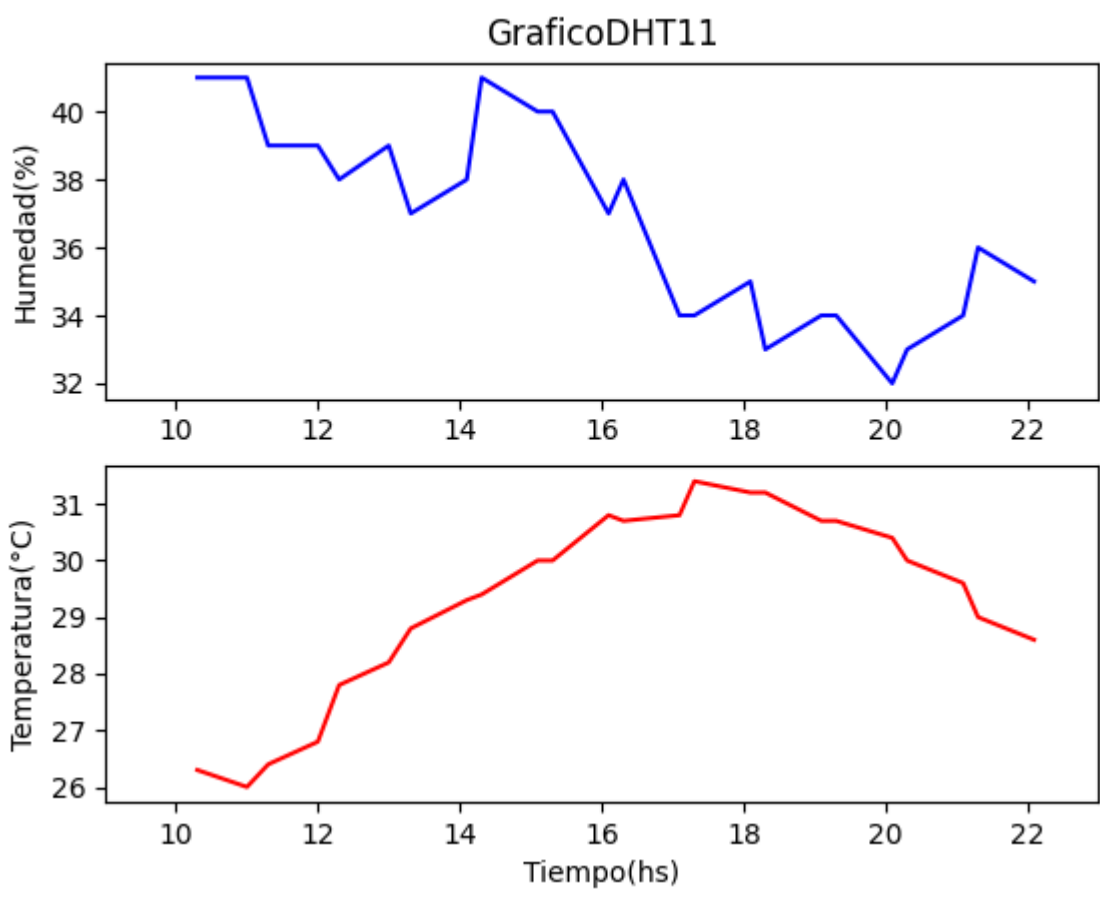
```
In [3]: def EscribirArchivo(listValores):
        'Abre o crea un archivo .csv, donde se guardan los valores enviados por Arduino'
        Archivo = open("")
        Ingrese el nombre del archivo en el cual se almacenaran los datos,
        En caso, de que el nombre no exista o no exista el archivo, se creará uno.
        No olvide escribir la terminación ".csv"
        Nombre: "", "w") #Se abre el archivo en modo escritura.
        escribir = csv.writer(Archivo)

        #Se escriben los datos en formato de filas.
        for renglon in listValores:
            escribir.writerow(renglon)
        Archivo.close()
```

El archivo en cuestión se abrirá o creará si es que no existe uno con el nombre designado por el usuario. En este caso se creó un archivo con el nombre: TemperaturaHumedadPrimeroEnero12Horas.csv En este caso se probó el código con la toma de 24 datos, los cuales se tomaron con un intervalo de 30 minutos, empezando a las 10:30 hs y finalizando a las 22:1 hs, quedando constituido de la siguiente manera:

TemperaturaHumedadPrimeroEnero12Horas.csv - LibreOffice Calc								
File Edit View Insert Format Styles Sheet Data Tools Window Help								
H49								
	A	B	C	D	E	F	G	H
1	10.30	41.00	26.30					
2								
3	11.0	41.00	26.00					
4								
5	11.30	39.00	26.40					
6								
7	12.0	39.00	26.80					
8								
9	12.30	38.00	27.80					
10								
11	13.0	39.00	28.20					
12								
13	13.31	37.00	28.80					
14								
15	14.1	38.00	29.30					
16								
17	14.31	41.00	29.40					
18								
19	15.1	40.00	30.00					
20								
21	15.31	40.00	30.00					
22								
23	16.1	37.00	30.80					
24								
25	16.31	38.00	30.70					
26								
27	17.1	34.00	30.80					
28								
29	17.31	34.00	31.40					
30								
31	18.1	35.00	31.20					
32								
33	18.31	33.00	31.20					
34								
35	19.1	34.00	30.70					
36								
37	19.31	34.00	30.70					
38								
39	20.1	32.00	30.40					
40								
41	20.31	33.00	30.00					
42								
43	21.1	34.00	29.60					
44								
45	21.31	36.00	29.00					
46								
47	22.1	35.00	28.60					
48								

Una vez obtenidos los datos, debemos volver a iniciar el código y teclear “analizar” para poder leer los datos de un archivo .csv elegido por el usuario, graficarlos (elegido el nombre que se desee para el título del gráfico) y obtener los valores máximos y mínimos.



Importante:

Para que el sistema funcione de la manera en la que se pensó, es necesario que cada conjunto de datos sea enviado por Arduino en la misma línea y separados por un espacio o tabulación, quedando de la siguiente manera:

16.0	15	25.3
16.10	15	25.6
16.20	15	26

Siendo la primera columna la hora, la segunda la humedad en porcentaje y la última corresponde a la temperatura. Estos datos se leen de la siguiente manera:

“A las 16 hs, hubo una humedad del 15% e hizo 25.3 °C”

Otra noción a tener en cuenta es que no se debe tener abierto el monitor serie de Arduino, ya que esto ocasiona problemas en la comunicación y no dejará ejecutar el programa en Python. Al mismo tiempo, no es necesario tener abierto el IDE de Arduino, solo es

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js SB y tener cargado el sketch en la placa Arduino.

