

Design of a low-cost automatic weather station for capturing and improving the processing of variables obtained in real time

Santiago Rubiños, Doctor¹, Niko Alain Alarcón-Cueva, Doctor¹, Juan Grados, Doctor¹, Kevin Sedano, estudiante¹, Joshua Mayta, estudiante¹, Yesmi Ortega-Rojas, Doctor¹, Arcelia Rojas, Doctor¹

¹Universidad Nacional del Callao, Perú, slrubinosj@unac.edu.pe, naalarconc@unac.edu.pe, jhgradosg@unac.edu.pe, klsedanog@unac.edu.pe, jamaytat@unac.edu.pe, ykortegar@unac.edu.pe, aorojass@unac.edu.pe

Abstract.- The purpose of this scientist is to show the design of a system for obtaining meteorological variables through the use of sensors and a microcontroller capable of processing signals in real time in order to obtain useful reports in contrast to conventional high-speed methods. cost. In this way, the article presents the general concepts that encompass the research, as well as the current reality regarding meteorology companies. The procedures used for simulation and design are shown in the procedures. The results reflect the report published by the device and the representative graphs.

Keywords. - design, meteorological variables, low cost, automatic, processing

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2022.1.1.433>

ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

Diseño de una estación meteorológica automática de bajo costo para la captación y mejora del procesamiento de variables obtenidas en tiempo real

Santiago Rubiños, Doctor¹, Niko Alain Alarcón-Cueva, Doctor¹, Juan Grados, Doctor¹, Kevin Sedano, estudiante¹, Joshua Mayta, estudiante¹, Yesmi Ortega-Rojas, Doctor¹, Arcelia Rojas, Doctor¹

¹Universidad Nacional del Callao, Perú, slrubinosj@unac.edu.pe, naalarconc@unac.edu.pe, jhgradosg@unac.edu.pe, klsedanog@unac.edu.pe, jamaytat@unac.edu.pe, ykortegar@unac.edu.pe, aorojass@unac.edu.pe

Resumen.- El presente artículo científico tiene como propósito mostrar el diseño de un sistema de obtención de variables meteorológicas por medio del uso de sensores y microcontrolador capaces de procesar las señales en tiempo real a fin de obtener reportes útiles en contraste con los convencionales métodos de alto costo. De esta forma, el artículo presenta los conceptos generales que abarcan la investigación, así como la realidad actual respecto a las empresas de meteorología. En los procedimientos se muestran los procesos usados para la simulación y diseño. En los resultados se refleja el reporte emitido por el dispositivo y las gráficas representativas.

Palabras clave. - diseño, variables meteorológicas, bajo costo, automática, procesamiento

I. INTRODUCCIÓN

En la actualidad muchas entidades públicas y privadas se dedican a la actividad del muestreo de señales meteorológicas, con distintos fines, sean para prevenir fenómenos naturales como para anticipar fenómenos que traen consigo posibles catastrofes o indican condiciones adecuadas para aprovechamiento de energía. [1] La captura y procesamiento de dichas señales, muchas veces costosas para poder ser utilizada con fines personales o suponen un gran esfuerzo de inversión para empresas dedicadas. Ante ello, la presente investigación plantea el diseño de un sistema que captura señales meteorológicas, el sistema posee como base fundamental la placa Arduino Mega, hoy muy comercializada y con precio razonablemente bajo en el mercado, así como el uso del módulo ESP6288 de Arduino, con el cual podemos procesar las señales en tiempo real mediante en servidor web. De este modo la investigación que se presenta a continuación pretende mostrar el desarrollo de un sistema de señales meteorológicas, como temperatura, fuerza de viento, entre otras, con un costo de inversión bajo. [2]

La temperatura es una magnitud muy conocida cuando se realiza un estudio al estado de la atmósfera. En los medios de comunicación casi siempre se incluye un espacio dedicado a la temperatura. La temperatura está vinculada con la rapidez de las partículas de la materia al estar en movimiento, por lo tanto, mientras más movimiento exista, mayor será la temperatura. Para obtener datos de medición de la temperatura se tiene que conocer la relación del cambio de temperatura con las propiedades de la materia que se ven alteradas: la

resistencia eléctrica de algunos materiales, el volumen de un cuerpo, el color de un objeto, etc. [3]

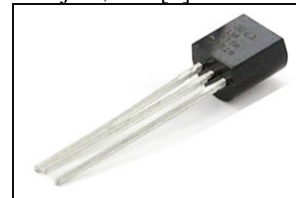


Fig. 1 Sensor de temperatura LM35

La humedad ambiental también es conocido como ratio de evapotranspiración, y se puede obtener mediciones con bastante aproximación si se relaciona la evapotranspiración potencial (ETP) y la precipitación. Esta variable se mide por medio del sensor de humedad DHT22 [4]

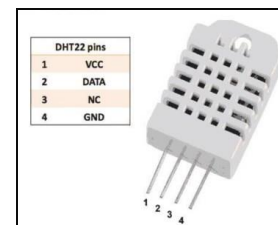


Fig. 2 Sensor de humedad DHT22

La radiación ultravioleta (UV) es una forma de radiación no ionizante que es emitida por el sol y fuentes artificiales, como las camas bronceadoras. Aunque ofrece algunos beneficios a las personas, como la producción de vitamina D, también puede causar riesgos para la salud. [5] El cual es medido por medio del sensor mostrado a continuación



Fig. 3 Sensor de UV EPD-440-0-1.4

La precipitación es la caída de agua desde la atmósfera hacia la superficie terrestre. La precipitación forma parte del ciclo del agua que mantiene el equilibrio y sustento de todos los ecosistemas.[6] La precipitación se genera por la condensación de agua, o sea, la acumulación de agua en la atmósfera creando nubes. El agua que se acumula en la atmósfera generalmente se encuentra en estado gaseoso. Cuando existe una cantidad considerable de agua gaseosa dentro de las nubes el agua pasa del estado gaseoso al líquido o al sólido. [7] [8]



Fig. 4 Sensor modelo RK400-04

II. PROCEDIMIENTO

La presente investigación desarrolla un sistema de estación meteorológica automática a fin de obtener una información del clima que sea confiable y precisa, utilizando dispositivos de bajo costo. Para la elaboración del proyecto se consideran los siguientes procesos en distintas etapas donde se realiza el acondicionamiento de la señal deseada a obtener:

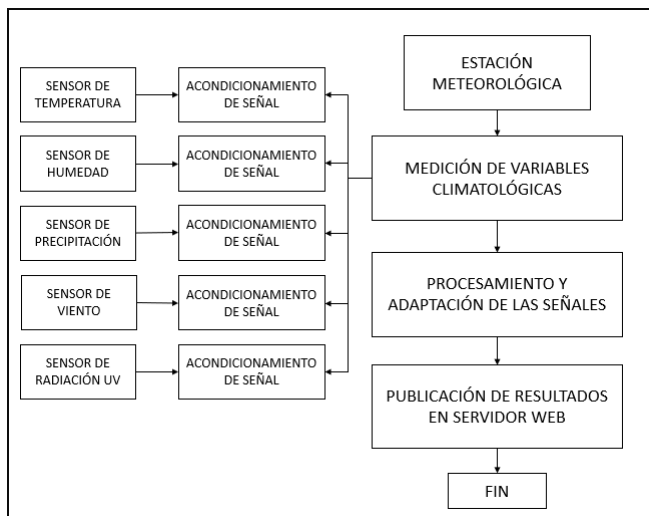


Fig.5 Diagrama de bloques del sistema

Para la obtención de la simulación del sensor de humedad podemos hacer uso del DHT22. Este sensor permite obtener tanto la temperatura y la humedad. A pesar de ello, la medida de la temperatura del DHT22 puede ser afectada pasando el tiempo. Por ello debemos colocar un sensor que añada robustez al sistema. Un sensor extra que podemos poner es el sensor LM35, este sensor permite una medida extra de

temperatura. a continuación, se muestra la simulación en proteus de los sensores

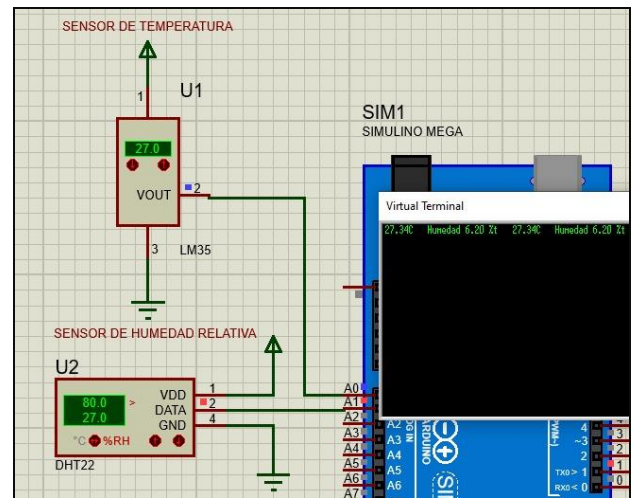


Fig. 6 Simulación del DHT22 y LM35

La simulación en Proteus nos puede ayudar para simular el acondicionamiento, ya sabiendo el acondicionamiento pasaremos a usar el Simulink de Matlab. Para cargar en tiempo real nuestro sistema debemos configurar el Simulink. Para ello realizamos primero una prueba en tiempo de real, la prueba consiste en prender y apagar un led a través de Simulink

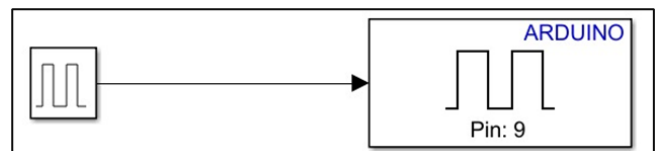


Fig. 7 Prueba de comunicación entre Simulink y Arduino

Configurando nuestro Arduino, el puerto a usar y los baudios utilizados, podremos realizar una buena conexión.

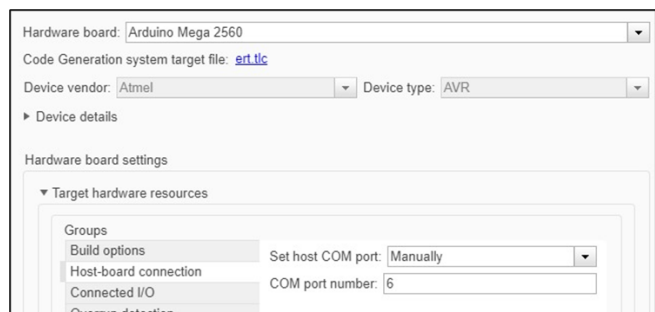


Fig. 8 Configuración de Hardware y puerto serial en Simulink

Se muestra que Simulink nos permite trabajar en tiempo real, además de programar directamente el Arduino al estar conectado. Con los siguientes softwares podremos obtener gráficas y realizar el acondicionamiento debido a los sensores

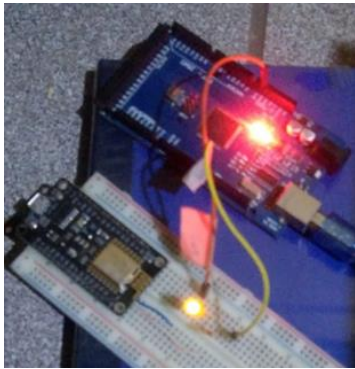


Fig. 9 Demostración de comunicación entre Simulink y Arduino

También es necesario realizar una comunicación serial entre el Arduino y el NodeMCU, ello lo hacemos ya que el módulo Wifi mandara las señales a la nube. El módulo de Wifi deberá adquirir todas las señales de nuestro sistema que le enviará el Arduino Mega y enviarlas a ThingSpeak para una monitorización exitosa. [9]

El módulo de Wifi deberá adquirir todas las señales de nuestro sistema que le enviará el Arduino Mega y enviarlas a ThingSpeak para una monitorización exitosa. [15]

En la imagen siguiente podremos ver la conexión entre el Arduino Mega y el NodeMcu.

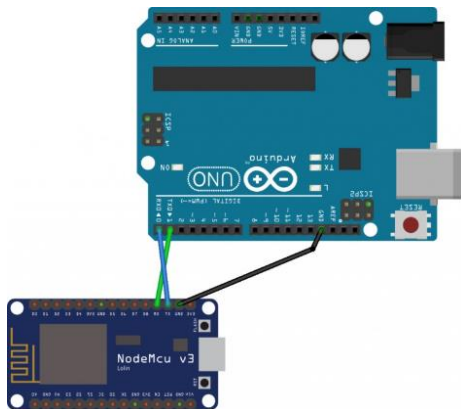


Fig. 10 Diagrama de conexión de ESP8266 y Arduino

También debe tenerse en cuenta que debemos enviar una cadena de caracteres por medio del Arduino Mega, el módulo Wifi lo recibirá en cadena de caracteres y luego deberemos separar los datos, A continuación, se muestra como sería este proceso



Fig. 11 Comunicación Serial entre Arduino y NodeMCU

Una vez comunicado el Arduino, Simulink , NodeMCU pasaremos a realizar la comunicación entre el NodeMCU y Ubidots. Para ello será necesario descargar la librería de Ubidots , además de configurar nuestro acceso a Ubidots para que se transmitan los datos.



Fig. 12 Selección de variable a usar en Ubidots

Nuestra programación de Arduino a través de Simulink se realizaría de la siguiente manera, primero se deberá seleccionar el bloque Analog Input y se tendrá que escoger el pin 0. Luego se elegirá el puerto Serial 3, que sería el puerto serial que se usará del Arduino para la comunicación serial.

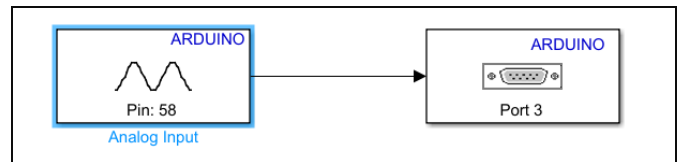


Fig. 13 Diagrama de bloques de Simulink

Podemos hacer un testeó rápido con este código, sin embargo, los datos que necesitaremos mandar al NodeMCU, necesitan ser números y por lo tanto necesitaremos solucionar dos problemas, el primero sería realizar la conversión de ASCII a número y usar un delay el cual permita leer todos los números que transmitimos. [14]

Para corregir el problema de los caracteres ASCII podemos hacer la conversión de entero a carácter usando (char) y crear un buffer para almacenar los caracteres que se están recibiendo, para que finalmente nos imprima el resultado del buffer cuando termine de recolectar la información. [10]

Se tiene que mencionar que el delay utilizado es referencial para esta aplicación, pero puede cambiar ya que guarda relación con la cantidad de caracteres que se desee leer.

Si deseamos enviar más de un numero de un dígito al NodeMCU , necesitaremos realizar una conversión de String a int, para esta acción podemos usar el comando to Int():. [11]

CÓDIGO PARA RECIBIR NÚMEROS DE MAS DÍGITOS POR MEDIO DEL NODEMCU.

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

void loop() {
  if (Serial.available() > 0) {
    //Delay especifico para la lectura de caracteres
    delay(10);

    //Creación de buffer
    String bufferString = "";

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

    //Conversión de buffer a numero
    int num = bufferString.toInt(); //Imprimimos el numero

    Serial.println(num); } }
```

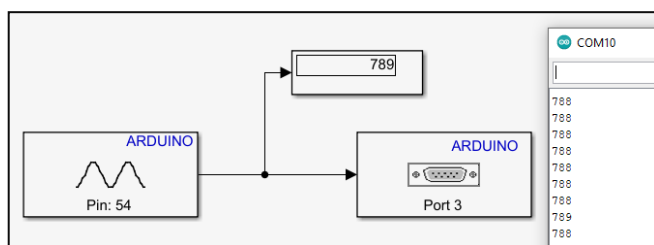


Fig 14. Envío de números a través del puerto serial desde el Arduino al NodeMCU.

Tener en cuenta que además de configurar la variable que usaremos, debemos copiar nuestro Ubidots Token, esto nos permitirá conectar el NodeMCU con la página. [12] [13]



Fig. 15 Obtención de Ubidots token a través de su página

A continuación, se muestra el código para la programación.

CÓDIGO PARA RECIBIR NÚMEROS DE MAS DÍGITOS POR MEDIO DEL NODEMCU

```
#include <Ubidots.h>
```

```
const char* UBIDOTS_TOKEN = "BBFF-
YSSI9lExO3xDgPc9BVK2V6jE5g1gZV"; // Ingrese el
Ubidots TOKEN
const char* WIFI_SSID = "_____"; // Ingrese el Wi-Fi
SSID
const char* WIFI_PASS = "_____"; // Ingrese el Wi-Fi
password
Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);
int var =0; //variable donde se almacenará la información
int var1=-2; //variable para solucionar errores de datos no
enviados
void setup() {
  Serial.begin(9600);
  ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);}
void loop() {
  while(var ==-2 || var1==0){
    if (Serial.available() > 0) {
      //Delay variable para cada aplicación
      delay(5);
      //Creación del buffer
      String bufferString = "";
      while (Serial.available() > 0) {
        bufferString += (char)Serial.read(); }
      var1 = bufferString.toInt();
      var = var1;
      if (var1!=0){
        var1 = 1; } } }
    ubidots.add("var", var); // Ingresamos la variable que
    manderemos a Ubidots
    bool bufferSent = false;
    bufferSent = ubidots.send(); // Envío de datos a una etiqueta
    de dispositivos que coincidan con el ID del dispositivo
    if (bufferSent) {
      // Mensaje para saber si los datos fueron enviados
      Serial.println("Valor enviado por el dispositivo"); }
    delay(1000); // delay usado
    var1=-2; }
```

Se puede ver a continuación como se programa el Arduino para la comunicación Serial con el NodeMCU, que a su vez se comunica con la página de Ubidots.

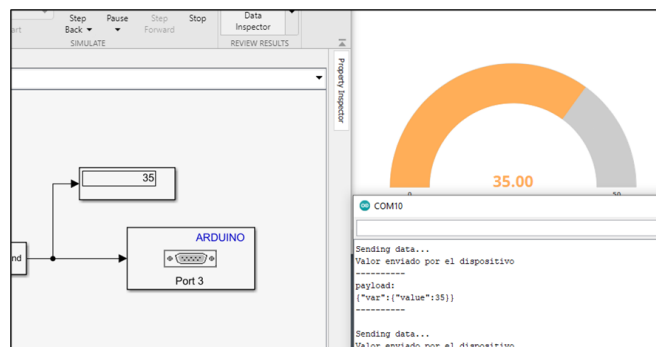


Fig. 16 Comunicación del NodeMCU con la página Ubidots

Habiendo mencionado lo anterior podemos apoyarnos de simulink para hacer el acondicionamiento de señales de los diversos sensores que tiene una estación meteorológica

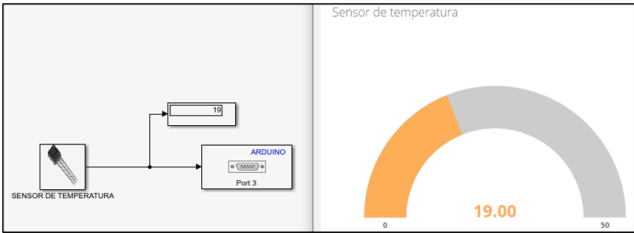


Fig. 17 Comunicación del sensor de temperatura y Ubidots

La programación la tendríamos que realizar en Simulink, e ingresar los datos de los sensores a través del Arduino para poder hacer el acondicionamiento.

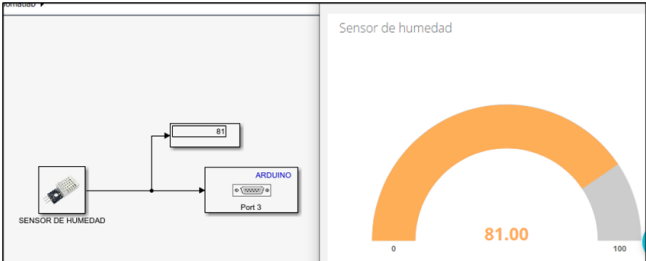


Fig. 18 Comunicación del sensor de DTH22 y Ubidots

Se puede acondicionar señales de otros sensores más avanzados, previo acondicionamiento de su alimentación. (esto no podría hacerlo en Simulink)

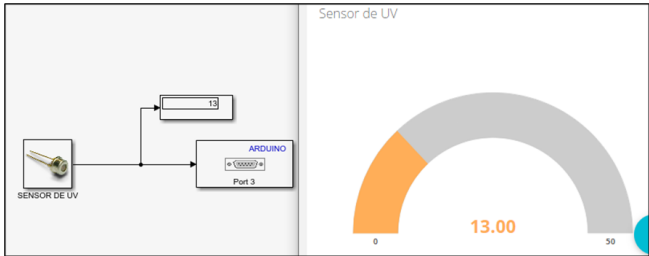


Fig. 19 Comunicación del sensor de UV y Ubidots

COMPARATIVA DE COSTO DE INVERSIÓN

Componentes:

	Precio
Arduino	\$ 12.00
Sensores	\$ 12.00
Módulo Wifi	\$ 2.00
Circuiteria	\$ 4.00
	\$ 30.00

Brinda un total de 30 dólares. en comparación de las estaciones meteorológicas nuevas cuyo precio es de 1800 dólares. [16]

III. RESULTADOS

El acondicionamiento de sensores de bajo costo para la obtención de señales aceptables en una estación meteorológica automática fue exitoso. Obtuvimos una señal limpia de ruido y los datos fueron enviados a internet para su posterior monitoreo. De esta manera corroboramos nuestra hipótesis y verificamos los resultados obtenidos con resultados de los mismos parámetros de diversas investigaciones y páginas meteorológicas. A continuación, se muestran los resultados de datos mostrados, obteniendose una gráfica ascendente

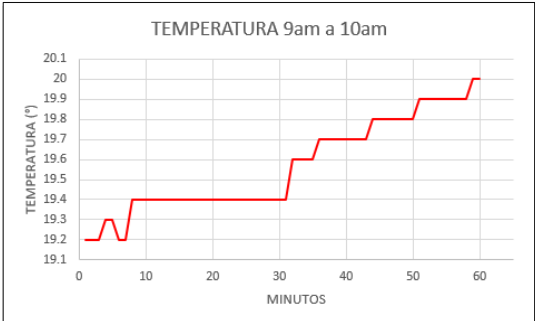


Fig 20. Gráfica de temperatura en 1 hora de medición

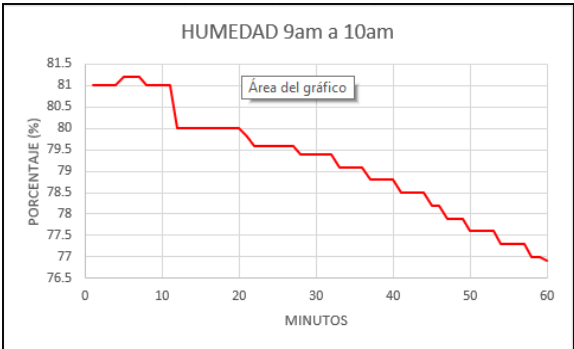


Fig 21. Gráfica de humedad en 1 hora de medición

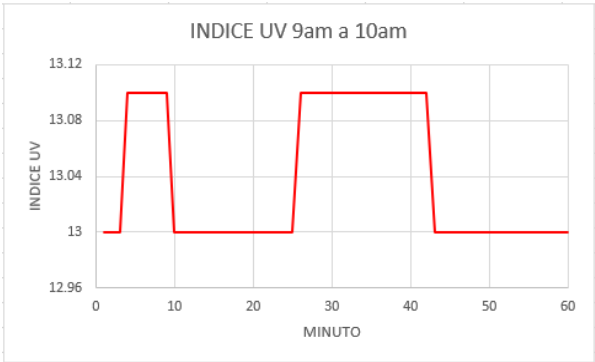


Fig 22. Gráfica de índice UV en 1 hora de medición

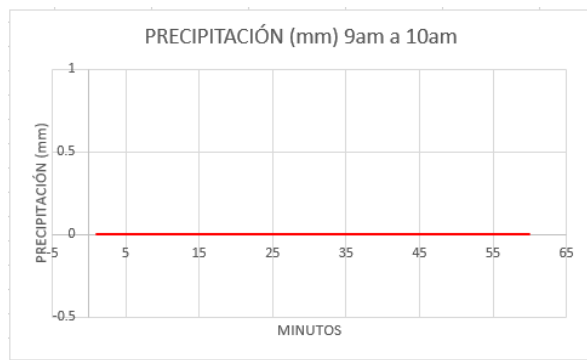


Fig 23. Gráfica de precipitaciones en 1 hora de medición

IV. CONCLUSIONES

Se diseñó un acondicionamiento de señales para sensores de bajo costo para una estación meteorológica automática, permitiendo demostrar que un eficiente acondicionamiento para la eliminación del ruido de la señal puede reducir costos en operación. A su vez, se diseñó un método de visualización por medio de la plataforma UBIDOTS en donde los datos se muestran en tiempo real. De esta manera se concluye que el dispositivo es funcional, mejorable y eficiente para medir variables a tiempo real.

RECOMENDACIONES

- A través de la comunicación serial se puede transmitir caracteres en vez de números y para poder ser usados, todo tipo de dato debe ser convertido a formato numérico.
- En la relación de Simulink con Arduino, se recomendaría usar un tiempo de muestreo de 0.02
- Tanto en Simulink como Arduino y ESP se debe usar el mismo formato de tipo de variable, o usar el bloque Round que permite redondear todo decimal a entero
- Los datos obtenidos en la toma de resultados deben ser corroborados con páginas de meteorología para verificar la veracidad y precisión de los valores obtenidos de sensores baratos con acondicionamiento.

REFERENCES

- [1] Arteaga Ramirez, R., Cervantes Osornio, R., Vásquez Peña, M., & Ojeda Bustamante, W. (2017). Estación meteorológica convencional versus automática sus diferencias evaluadas con índices estadísticos . Revista Mexicana de ciencias agrícolas, 1301-1314.
- [2] Aybar Camacho, C., & Lavado Casimiro, W. (2017). Atlas de zonas de vida del Perú : guía explicativa. Lima: Servicio Nacional de Meteorología e Hidrología del Perú - SENAMHI.
- [3] Cárdenas rojas, j. (2017). implementación de un sistema de gestión de información de las estaciones meteorológicas, en la cuenca del rio cachi, región ayacucho. tesis. universidad peruana los andes, huancayo - Perú.
- [4] Ferrer Sanabria, R. D. (2017). Desarrollo de una estación meteorológica autónoma de bajo costo. Tesis. Universidad Santo Tomás Bucaramanga, Bucaramanga.
- [5] Gómez de 31 años (2017). Niveles de radiación ultravioleta (UV) en la ciudad de ayacucho periodo 2006 - 2011 e incidencia de cáncer de piel en la población. Tesis. Universidad Nacional de San Cristóbal de Huamanga, Ayacucho – Perú.
- [6] Guillén Fernández, g. m. (2018). identificación espacial de zonas vulnerables ante lluvias extremas en lima metropolitana y el callao, a partir de los impactos reportados por la inusual lluvia del 15 de enero de 1970 y otros eventos. tesis. Pontificia Universidad Católica del Perú, lima.
- [7] Medina Molina, c. a., & ramirez galvis, w. f. (2019). diseño y construcción de estación meteorologica, ubicada en la vereda el callejon, ricaurte-cundinamarca. Tesis. Universidad Católica de Colombia, Bogotá.
- [8] Medina, L., Ramón, V., & Fresdo, V. (2017). Comparación de la temperatura, precipitación, velocidad media y dirección del viento medidas por dos estaciones meteorológicas, ubicadas en Punta de Piedras, isla de Margarita. Memoria de la Fundación La Sala de Ciencias Naturales, 61-80.
- [9] Murcia Almanza, D. S., & Rojas Paternina, A. (2015). Diseño e implementación de un prototipo de estación meteorológica remota de bajo costo empleando el enfoque de internet de las cosas. Tesis. Universidad Piloto de Colombia, Bogotá D.C.
- [10] Organización Mundial de la Salud. (2002). Índice UV solar mundial: Guía práctica.
- [11] Polo Bravo, C., Torres Muro, H., Ríos Marquezado, J., Bernabé Coaquera, R., & De la Gala Contreras, A. (2017). Diseño, construcción y sistematización de una estación metereológica de registro automático y procesamiento de tres parámetros basado en la implementación de hardware libre. Ciencias, 30-37.
- [12] H. Trojer, «Meteorología y climatología de la vertiente del Pacífico colombiano», Rev. Acad. Colomb. Cienc. Ex. Fis. Nat., vol. 41, n.º Suplemento, pp. 467–490, dic. 2017.
- [13] Vargas Seminario, F. (2015). Sistema de información en línea para el seguimiento del fenómeno El Niño en la costa norte del Perú. Tesis. Universidad de Piura, Piura.
- [14] Cantú Silva, Israel, & Yáñez Díaz, María Inés. (2018). Efecto del cambio de uso de suelo en el contenido del carbono orgánico y nitrógeno del suelo. Revista mexicana de ciencias forestales, 9(45), 122-151. <https://doi.org/10.29298/rmcf.v9i45.138>
- [15] Ruiz-Ayala, Daniel Camilo, Vides-Herrera, Carlos Arturo, & Pardo-García, Aldo. (2018). Monitoreo de variables meteorológicas a través de un sistema inalámbrico de adquisición de datos. Revista de Investigación, Desarrollo e Innovación, 8(2), 333-341. <https://doi.org/10.19053/20278306.v8.n2.2018.7971>
- [16] Zuñiga Vinueza, W., Polanco Risquet, A., & Nagy, A. S. (2016). Prototipo de mini estación meteorológica automática inalámbrica. . Publicando, 20-32