

Práctica 1: Estación Meteorológica (20 Agosto 2021)

J. Wannan, M. Mata, D. López, W. Morales, E. Arnas y M. Chávez

Resumen— El documento contiene información sobre el desarrollo mediante la implementación de IoT y creación de una estación meteorológica la cual es capaz de obtener datos sobre la temperatura, humedad y velocidad y dirección del viento. Se dan especificaciones sobre el desarrollo, funcionamiento y conexión de dicha estación. La estación es controlada por el microcontrolador Arduino Mega, con visualización de datos por medio de Processing y sus datos son almacenados en una base de datos.

Palabras clave— anemómetro (*anemometer*), base de datos (*database*), comunicación serial (*serial communication*), estación meteorológica (*meteorological station*), IoT (*Internet of things*), medición (*measurement*).

I. NOMENCLATURA

II. INTRODUCCIÓN

ESTE documento proporciona la documentación del proyecto de estación meteorológica. Posee una vista al desarrollo del código para el funcionamiento de los diferentes sensores para el microprocesador *Arduino Mega* y la construcción de la misma estación. Así mismo posee una vista al desarrollo de la aplicación de visualización de los datos en *Processing* y la conexión de *Arduino* a la base de datos. Todo esto con el fin de desarrollar la estación con framework IoT.

III. DESARROLLO

A. Sensores

Para el desarrollo de la estación meteorológica fueron utilizados sensores de temperatura y emisores y receptores infrarrojos.

Fue usado un módulo sensor de DHT11. El cual es un sensor digital de temperatura y humedad el cual integra un sensor capacitivo de humedad y un termistor para medir el aire circundante y muestra los datos mediante una señal digital en el pin de datos.

Para la medición de velocidad del viento y su dirección fueron usados emisores y receptores infra rojos los cuales son comúnmente usados para detección de objetos por medio de la radiación electromagnética infrarroja.

B. Arduino [1]

Arduino es una compañía que crea microcontroladores (*hardware*) y programa (*software*) libres (*open-source*), así como un proyecto y comunidad de usuarios que diseña y manufactura microcontroladores de una sola placa, también

crea equipos (*kits*) para construir dispositivos digitales.

El programa de *Arduino* se encarga de la recepción de los datos proporcionados por los diferentes sensores conectados al microcontrolador. Según el tipo de sensor estos se dividen en:

1) Conexión Serial

La recepción de datos se realiza por medio del cable de *Arduino* al estar directamente conectada a el puerto USB de la computadora. *Arduino* posee la librería “*Serial*” con la cual se especifica la velocidad de transmisión y recepción de datos.

2) Sensor DHT11

El sensor DHT11 es conectado por medio del pin 7 de el microcontrolador y hace uso de la librería “*DHT*”. Con la cual con las funciones propias de la librería obtiene la temperatura y la almacena en variables para luego ser transmitida a base de datos y la aplicación de *Processing*.

3) Emisores y receptores infrarrojos

Los emisores y receptores infrarrojos son conectados por medio de los pines análogos A0-A6 se realiza la medición de la dirección del viento que, al ser detectado un receptor específico, se podrá saber la dirección del viento. De la misma forma, por medio del pin 50 del microcontrolador puede medir la frecuencia con la cual el emisor y el receptor se encuentran cuando el viento mueve el anemómetro.

C. Processing [2]

Processing es un programa flexible, útil para el aprendizaje de código. Creado en 2001 *Processing* ha sido remarcado por su uso de artes visuales y literatura visual.

La aplicación de *Processing* busca que la visualización de los datos recibidos desde los sensores del microcontrolador *Arduino*. Esta aplicación se divide en 5 partes que son:

1) Conexión Serial

La conexión serial es realizada entre la aplicación y el microcontrolador. Esto es realizado por medio de la librería “*Serial*” de *Processing*. Lee las cadenas enviadas y las modifica y procesa de tal forma que se pueda adquirir toda la información necesaria para los parámetros y variables.

2) Temperatura

La temperatura es recibida como parámetro y al ser leída su respectivo indicador muestra su representación numérica y su representación gráfica por medio de colores cambiantes conforme a la temperatura leída.

3) Humedad

La humedad es recibida en una variable y al ser leída su respectivo indicador muestra su representación numérica, así como una representación simbólica de cantidad de partículas en el ambiente.

4) Velocidad del Viento

La velocidad del viento es representada por la una imagen reconocible, es recibida por su respectivo indicador y muestra la imagen girando dependiendo de la velocidad obtenida.

5) Dirección del Viento

La dirección del viento es obtenida por medio de cadenas y dependiendo del indicador activado, la imagen de una flecha apuntará la dirección de donde viene el viento, la cola de dicha flecha indica la dirección del viento

La figura 3 representa como se ve la aplicación

D. Base de Datos [3]

La base de datos seleccionada es MongoDB el cual es una base de datos de fuente disponible (*source-available*), multiplataforma y orientada a documentos. Clasificado como un programa de base de datos NoSQL, MongoDB usa documentos estilo JSON con esquemas opcionales.

Luego de obtener los datos por medio de comunicación serial desde *Arduino* la aplicación de *Processing* envía los dichos datos por medio de una API a la base de datos para poder ser almacenados.

E. Gráficos



Figura 1: Boceto de prototipo de estación meteorológica



Figura 2. Prototipo completo de estación meteorológica

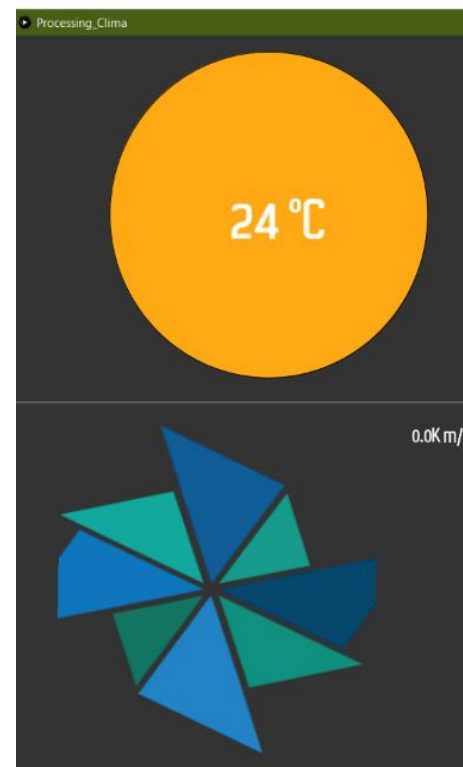


Figura 3: Aplicación de Processing

```

> db.data.remove({});
WriteResult({ "nRemoved" : 510 })
> db.data.remove({});
WriteResult({ "nRemoved" : 164 })
> db.data.find({});
{ "_id" : ObjectId("6120808cfc7fb490f85657f1"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("6120808efc7fb490f85657f2"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("6120808ffc7fb490f85657f3"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208090fc7fb490f85657f4"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208091fc7fb490f85657f5"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208092fc7fb490f85657f6"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208093fc7fb490f85657f7"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208094fc7fb490f85657f8"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208095fc7fb490f85657f9"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208096fc7fb490f85657fa"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208097fc7fb490f85657fb"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208098fc7fb490f85657fc"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("61208099fc7fb490f85657fd"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("6120809efc7fb490f85657fe"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("6120809ffc7fb490f85657ff"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("612080a0fc7fb490f8565800"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("612080a1fc7fb490f8565801"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }
{ "_id" : ObjectId("612080a2fc7fb490f8565802"), "Id" : "[object Promise]1", "Temperatura" : 24.4, "Humedad" : 70, "Viento" : 0, "Direccion" : "N" }

```

Figura 4: Muestra de información en base de datos

IV. CONCLUSIONES

Como conclusión se puede resaltar que existen componentes que generan problemas a la hora de su obtención, debido a su escasez o inexistencia localmente. Como solución se da usar componentes diferentes para poder adaptar el funcionamiento y realizar mediciones consiguiendo el efecto deseado. Un ejemplo de esto es la medición de la velocidad y dirección del viento los cuales sus sensores específicos no están en existencia en el país.

REFERENCIAS

Web Pages (Páginas Web):

- [1] Available: <https://www.arduino.cc/>
- [2] Available: <https://processing.org/>
- [3] Available: <https://www.mongodb.com/>