



Mini estación meteorológica

Integrantes: Javier Carrere & Carlos Durán

Carrera: Ingeniería en informática

Grado: 2° año

Profesor: Christian Rivero

INDICE

INTRODUCCIÓN	1
Funcionamiento del proyecto	2
Objetivo del proyecto	2
Componentes utilizados	3
Sensor DHT11.....	3
Sensor BMP180.....	4
Modulo I2C para LCD	5
Pantalla LCD	6
Arduino Uno R3.....	7
ESP32 WROOM 32S.....	8
AVANCES.....	9
Desarrollo de “Proyecto mini estación meteorológica”	9
Avance: PANTALLA LCD I2C 16X2 – Modulo I2C	9
Avance: Sensor DHT11	10
Avance: Sensor BMP180	10
Avance: Etapa final.....	11
Prototipo de cartón	11
Migración a ESP32	13
Conexiones con ESP32	13
Mejora del modelado 3D y su impresión.....	14
Siguiente fase del modelado.....	15
CONCLUSIÓN	19

INTRODUCCIÓN

En este proyecto se exploró una nueva área de aprendizaje llamada Internet de las Cosas (Internet of Things – IoT), con el objetivo de diseñar y construir una mini estación meteorológica. Este sistema permite monitorear en tiempo real diversas condiciones atmosféricas, lo que permitió al equipo aplicar conocimientos teóricos en un contexto práctico. Durante dos meses, se trabajó en la organización del proyecto en distintas etapas, que incluyeron investigación, diseño, programación y pruebas, abordando y resolviendo desafíos en cada una de ellas.

La relevancia de esta experiencia radica en el crecimiento del IoT, una tecnología que está transformando sectores como la industria, la salud y el medio ambiente, y que ofrece amplias oportunidades de innovación. A través del desarrollo de esta estación meteorológica, se pudieron explorar las aplicaciones y beneficios del IoT, como el monitoreo remoto y en tiempo real de variables meteorológicas. En este informe, se detallan el proceso de desarrollo, los componentes utilizados y los conocimientos adquiridos, así como los logros alcanzados y los retos superados a lo largo de este proyecto.

Funcionamiento del proyecto

En este proyecto, se tomó la idea de una miniestación meteorológica que sería un modelo que llevaría dentro conexiones que nos mostraría a través de una pantalla LCD, los datos que se mostrarán en pantalla cuentan con una tecnología para poder tomar los siguientes parámetros a tiempo real:

Sensor DHT11

- Temperatura
- Humedad

Sensor BMP180

- Presión atmosférica (o nivel del mar)
- Altitud

Toda esta conexión con los sensores y la pantalla van unidos a un microcontrolador llamado Arduino Uno que irá conectado a una fuente de poder ya sea un computador (en nuestro caso), a un enchufe directamente o alguna batería para darnos el voltaje suficiente y poner en funcionamiento todos los componentes.

Objetivo del proyecto

Gracias a la tecnología del IoT, se puede utilizar los sensores mencionados anteriormente para poder obtener un seguimiento con los parámetros del clima a tiempo real ya dichos más arriba. La importancia de poder tener estos datos en el momento que uno requiera puede servir mucho ya que el tener ese conocimiento puede ayudar a las personas que sean conscientes de los cambios climáticos que podrían afectar tanto positiva como negativamente por ejemplo a los eventos por mal clima o algo más serio como los accidentes automovilísticos en las autopistas por la poca visibilidad por niebla o suelos resbaladizos por las lluvias.

Componentes utilizados

Sensor DHT11

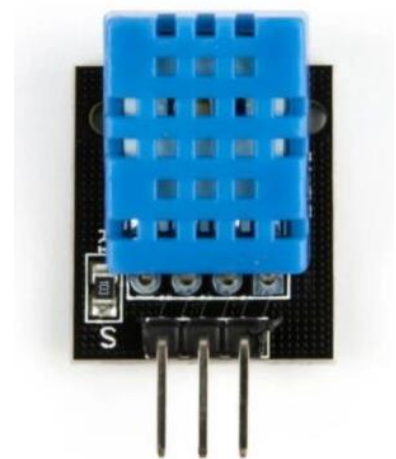
Este sensor higrómetro es un dispositivo que se utiliza para poder detectar y controlar el porcentaje de la humedad en el aire o de cualquier material o superficie en tiempo real. Un componente que suele usarse mucho en el aspecto de la meteorología.

Especificaciones técnicas:

- Voltaje de Operación: 3V – 5V DC
- Rango de medición de temperatura: 0 a 50 °C
- Precisión de medición de temperatura: ± 2.0 °C
- Rango de medición de humedad: 20% a 90% RH
- Precisión de medición de humedad: 5% RH
- Resolución Humedad: 1% RH

Pines:

- Alimentación: +5V (VCC)
- Datos (DATA)
- Tierra (GND)



Sensor BMP180

Un sensor barométrico se encarga de medir la temperatura, la altura respecto al nivel del mar basándose entre la presión del aire y la altitud. Un componente con alta precisión y bajo consumo de energía por lo que suele usarse mucho. Este fue diseñado para ser conectado directamente a un microcontrolador a través de un I2C. El uso de este destaca más en los sistemas de Autopiloto para Drones o UAVs.

Especificaciones técnicas:

- Voltaje de Operación: 1.8V – 6V DC
- Interfaz de comunicación: I2C (3.3V)
- Rango de Presión 300 a 1100 hPa
- Rango de altura medible: 0-9100 metros
- Medición de temperatura incluida
- Frecuencia de Muestreo: 120 Hz (máx.)

Pines:

- Alimentación: 3.3 – 5V DC (Recomendado 3.3V)
- Tierra (GND)
- Señal de Reloj (SCL)
- Señal de Datos (SDA)



Modulo I2C para LCD

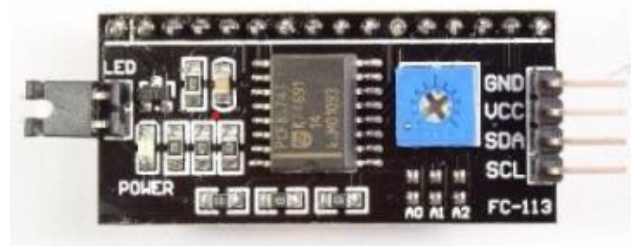
La función del módulo se encarga de manejar la pantalla LCD con mayor comodidad. Los recursos del Arduino Uno es limitados, pero gracias a la interfaz del I2C se reducen la cantidad de salidas para utilizar.

Especificaciones técnicas:

- Voltaje de Alimentación: 5V DC
- Controlador: PVCF8574
- Dirección I2C: 0X27
- Jumper para Luz de fondo
- Potenciómetro para ajuste de contraste

Pines:

- Tierra (GND)
- Alimentación: 5V (CVV)
- Señal de datos (SDA)
- Señal de Reloj (SCL)



Pantalla LCD

Es una pantalla de cristal líquido (de ahí la biblioteca que requiere llamada “Liquid Crystal”) que se utiliza para mostrar a desplegar información de tipo texto, números, símbolos o pequeñas figuras. Se puede trabajar con diversas tarjetas como Arduino, ESP32, etc. Es ideal para las interfaces hombre-máquina (HMI) para tener la información a la vista.

Especificaciones técnicas:

- Voltaje de Operación: 5V
- Color de texto: Blanco
- Filas: 2
- Columnas: 16



Arduino Uno R3

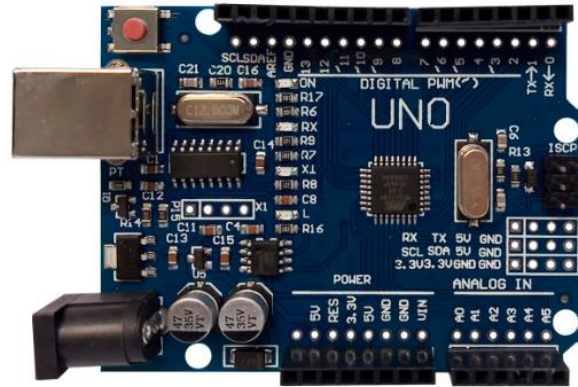
Esta placa de microcontrolador de tipo open-source con 20 pines de entrada y salida. En este dispositivo se pueden cargar programas desde la aplicación Arduino, es bastante práctico y sencillo de utilizar para empezar a trabajar con electrónica.

Especificaciones técnicas:

- Rango de alimentación: 7V – 12V

Pines:

- 6 pines de salida PWM
- 6 entradas analógicas
- 2 pines de alimentación 5V
- 1 pin de alimentación 3.3V
- 2 pines para tierra (GND)
- 1 pin SCL
- 1 pin SDA



ESP32 WROOM 32S

Es un microcontrolador dual core basado en un chip de Wi-Fi que trae Bluetooth. Este modulo se utiliza para prototipos de IoT. Este suele utilizarse en casos como el uso de una red de sensores, codificación de voz, transmisión de música y más.

Especificaciones técnicas

- Rango de alimentación: 5V (USB)
- Voltaje de entrada/salida: 3.3V
- Wi-fi 2.4 GHz hasta 150Mbit/s
- Desempeño hasta 600 DMIPS

Pines

- Pines digitales GPIO: 24 (Algunos solo entrada)
- Pines PWM: 16
- Pines analógicos ADC: 18



AVANCES

Desarrollo de “Proyecto mini estación meteorológica”

En la primera semana nos centramos en investigar sobre cómo funcionan las estaciones meteorológicas con Arduino, que librerías eran necesarias, que componentes son necesarios y el cómo realizar el circuito eléctrico para cumplir su función. De esa forma nos fuimos ordenando e investigando que cosas nos hacían falta para empezar a desarrollar nuestra mini estación meteorológica. Fuimos parte por parte, desde un principio primero investigamos como encender la pantalla que se encargaría de mostrar los datos que tomaríamos a través de los dos sensores, soldarla al modulo I2C para que funcione correctamente y como utilizar el sensor dht11 junto al bmp180 para poder obtener los datos a tiempo real que sería la función principal de este. A continuación, se explica más a detalle cada etapa del desarrollo

Avance: PANTALLA LCD I2C 16X2 – Modulo I2C

Una vez informados, pasamos a realizar pruebas pieza por pieza, empezando con la **pantalla LCD I2C** que era esencial para nuestro proyecto ya que necesitábamos mostrar los datos tomados de los sensores. Para poder realizar la conexión, utilizamos el **módulo I2C** para manejar la pantalla y la conexión de una forma más sencilla, ahorrando espacio con los cables y pines requeridos.

Complicaciones

Para lograr encender la pantalla y que llegara el voltaje suficiente era recomendable tener soldado el I2C junto a la pantalla, pero como no lo teníamos era complejo así que la conexión se hizo en la protoboard, sin embargo, aún era complicado porque los pines estaban sueltos y había que dejar la pantalla con un botón rojo por debajo para que mostrara el mensaje.

Avance: Sensor DHT11

Para trabajar con este sensor, se requirió incluir una librería **DHT11 sensor library by Adafruit** para poder añadir los parámetros **dht.readHumidity & dht.readTemperature** que son los que utilizaremos para tomar los datos de temperatura y humedad en tiempo real. Una vez instalada la librería desarrollamos el siguiente código que se encarga de imprimir en pantalla y en el monitor serial la humedad y temperatura.

Complicaciones

La dificultad que tuvimos principalmente fue que el sensor no nos mostrara los datos en pantalla ni en el monitor serial. Fuimos descartando las posibles fallas, biblioteca incorrecta, error en el código, dispositivo defectuoso o dañado, etc. Pero resultó ser que la conexión de pines era incorrecta a lo que pensábamos. Muchas fuentes nos daban a entender que la conexión era S (5V) pero resultó ser que era para el dato, luego de cambiar eso logramos que funcionara.

Avance: Sensor BMP180

Al igual que el sensor DHT11 investigamos sobre como había que trabajar con este dispositivo y necesitaba la librería **Sparkfun BMP180** para utilizar los parámetros **bmp180.startPressure & bmp180.getPressure** que son necesarios para tomar los datos de presión atmosférica con la unidad de medida milibar y la altitud en metros.

Complicaciones:

Para este sensor hubo ciertos problemas con saber que voltaje era el ideal para evitar daños al componente, al igual que el sensor DHT11 se estuvo investigando como funcionaba para cubrir fallas y quemar el sensor. Durante un tiempo se utilizó la cantidad de 5V para ver como operaba este. Sin embargo, al momento conectarlo y hacer correr el código este se calentó demasiado y se quemó, así que se tomo la decisión de utilizar la cantidad de 3.3V en otro sensor que había y evitar más accidentes que retrasara el desarrollo de las conexiones.

Avance: Etapa final

Como etapa final del proyecto, fuimos recopilando cada información adquirida y con esto creamos un código en base a los 2 anteriores, para así poder concluir con el funcionamiento completo de nuestra estación meteorológica. Mezclando ambos parámetros (**bmp180.start**, **bmp180.get** y **dht.read**) para tomar datos logramos realizar un código funcional. (Ver en Anexo

Complicaciones

A decir verdad, al momento de implementar ambos códigos en uno solo no hubo problema alguno, algún que otro pequeño error al momento de hacer funcionar el código con los parámetros y su lógica pero que fueron resueltos en el momento.

Prototipo de cartón

Para esta versión del proyecto, se utilizaron materiales simples:

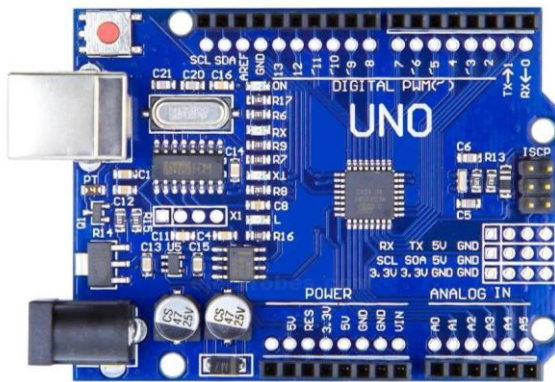
- Cartón
- Plástico
- Corchetera
- Wincha

Con esto se realizó una pequeña caja con un espaciado en la parte de arriba donde iría el plástico para que se pueda ver la conexión completa y la otra sería por el lado para poder meter toda la conexión con el sensor, poder interactuar con los cables en caso de que se suelten de sus pines y con los sensores para poder alterar la temperatura o humedad que se muestran en pantalla por un espacio que se hizo al frente. (Ver en Anexo 1 y 2)

MINI ESTACIÓN METEOROLOGICA

ETAPA 2

(Migración)



Migración a ESP32

Una vez completada la primera etapa en la que teníamos que hacer una simulación funcional con datos randomicos junto a un prototipo de cartón, iniciamos una nueva etapa en el proyecto en la cual utilizaríamos un módulo llamado ESP32, el cual nos ayudaría a guardar los datos tomados por los sensores en una base de datos que creamos y en una aplicación móvil MQTT gracias a un flujo de Node-RED creado en nuestro servidor.

Nuevo diseño de proyecto (Ver en anexo 3)

Para el nuevo modelo de nuestra estación meteorológica tomamos la decisión de utilizar un modelado 3D, para eso nuestro jefe de carrera nos mostró como hacer un diseño 3D en la página www.tinkercad.com. Una vez comprendido el uso de tinkercad, comenzamos con el nuevo diseño para nuestro proyecto el cual se basó en un personaje llamado BMO de una serie animada.

Conexiones con ESP32

Para realizar las conexiones con el módulo ESP32 se reemplazaron ciertos pines ya utilizados con el Arduino Uno los cuales fueron:

- SCL por D22
- SDA por D21
- Pin2 por Pin13

Con estos cambios se logró una conexión exitosa con el ESP32 y los otros componentes para la estación meteorológica. Se utilizó un código tomado de GitHub para realizar la conexión por código hacia una red Wi-Fi para así continuar con el envío de parámetros hacia la base de datos.

Mejora del modelado 3D y su impresión

En el diseño 3D del personaje BMO, hubo varias imperfecciones por corregir. Estas imperfecciones fueron las medidas del modelo que estaban distintas a las acordadas, los botones de la parte frontal no estaban pegados a la tapa como correspondía, el espacio en el modelo 3D para la pantalla estaba mal hecho por medidas, y no se hizo un espacio para el cable que va conectado al ESP32 para brindarle voltaje. Luego de corregir cada una de esas imperfecciones se pasó a la aplicación cura y se dividió la impresión en 3 partes diferentes.

Primera parte (Ver en Anexo 4)

Se imprimió la base completa por los lados junto al nombre del personaje en el que se inspiró el diseño 3D (BMO). Dentro de esta caja iría toda la conexión de la miniestación meteorológica.

Segunda parte (Ver en Anexo 5)

Esto se imprimió aparte de todo el modelo, sería la parte frontal o la cara del diseño la cual se utilizaría como una tapa para poder guardar la conexión dentro, también se hizo un espacio de forma rectangular que sería el lugar donde iría la pantalla LCD que mostraría los datos.

Tercera parte (Ver en Anexo 6)

La última parte de la impresión serían las extremidades del personaje, brazos y piernas para dar por finalizada la impresión 3D del modelo.

Siguiente fase del modelado

Para seguir con el desarrollo del modelo 3D, se buscaron los materiales adecuados para hacerlo lo más parecido posible, buscar los colores correctos, que tipo de pintura se utilizaría para colorear al BMO. Los materiales utilizados fueron los siguientes:

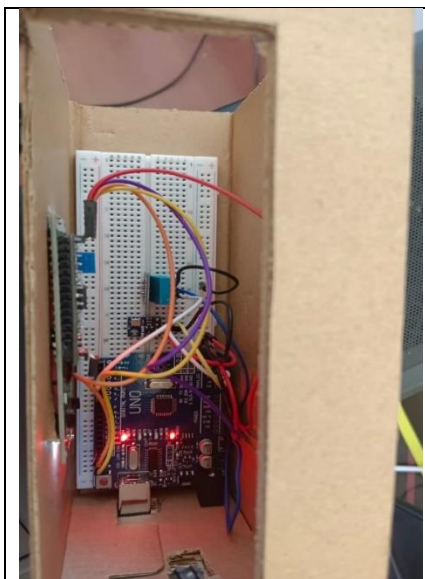
- Pintura de lata spray de color celeste
- Acrílicos
- Pegote
- Aluza
- Cartonero
- Gotita (Pegamento)

Como se realizó el coloreado:

Se utilizó una pintura en lata de color celeste para la caja completa solo por fuera y rociarla con unas 2-3 capas de pintura con espera de 30 minutos de secado de cada una para que agarre un tono fuerte (Ver en Anexo 7). Luego se repitió el mismo proceso con la parte frontal, cubriendo con pegote en cada botón y la cara. Luego de esperar el secado de la pintura celeste, se utilizó aluza para cubrir alrededor de los botones donde se pintaría con acrílico para evitar manchas por encima de la capa de pintura principal, ya que si llegase a ocurrir esto sería complejo retirar el acrílico sin pasar a dañar la capa celeste. (Ver en Anexo 8)

Una vez terminado todo el coloreado de la caja y su tapa, se procedió con pintar las extremidades, con gotitas se le agregaron las extremidades a cada lado encima de la O, se le pegaron los brazos, luego las piernas por debajo de los botones para finalizar el coloreado. Y ya para terminar por completo todo, con pegote se le puso en cada esquina y por dentro de la tapa simulando una bisagra para poder sostener la tapa. (Ver en Anexo 9)

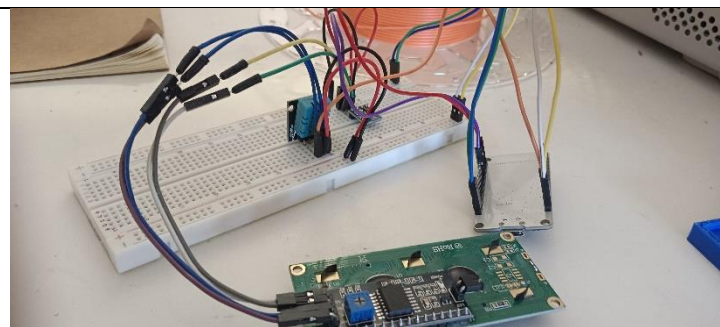
Anexos



Anexo 1 (Modelo de cartón por dentro)



Anexo 2 (Modelo de cartón por fuera)



Anexo 3 (Conexiones con ESP32)



Anexo 4 Impresión 3D (Primera parte)



Anexo 5 Impresión 3D (Segunda parte)



Anexo 6 Impresión 3D (Tercera parte)



Anexo 7 (Coloreado caja)



Anexo 8 (Coloreado tapa)



Anexo 9 (Modelo terminado)

LINKOGRAFÍA

Apoyo de IA

<https://chatgpt.com/>

Driver para Arduino Uno

<https://www.geekfactory.mx/tutoriales-arduino/driver-ch340-para-arduino-chinos-o-genericos/>

Sensor DHT11

https://www.youtube.com/watch?v=huljJV2B8sA&t=102s&ab_channel=JohannPerezE

<https://www.automatizacionparatodos.com/sensor-dht11-arduino/>

Sensor BMP180

https://github.com/sparkfun/BMP180_Breakout_Arduino_Library

<https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-/installing-the-arduino-library>

https://www.youtube.com/watch?v=Qxe-LSXtqo8&ab_channel=BioMakersIndustries

Modulo ESP32

<https://www.mechatronicstore.cl/esp32/?srsltid=AfmBOorVSZCFuHmtgokCVmUYPeSRoUvkRhvFh8nryUawR3curH8XCJLI>

<https://samueladesola.medium.com/how-to-set-up-esp32-wroom-32-b2100060470c>

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Modelado 3D

<https://www.tinkercad.com/dashboard>

Flujo Node-Red

<https://github.com/CarlosDur4n/FlujoNodeRedIoT.git>

Código versión final

<https://github.com/CarlosDur4n/CodigoArduino.git>

CONCLUSIÓN

El trabajar en equipo nos permitió llevar a cabo exitosamente el proyecto de la mini estación meteorológica, utilizando tecnología de sensores, Arduino, ESP32 y MQTT. A lo largo de las distintas etapas de desarrollo del prototipo, se adquirieron distintos conocimientos y habilidades prácticas, lo que convirtió esta experiencia en una base valiosa para futuros proyectos. La construcción de este sistema no solo fue una experiencia importante y recordada, sino también una oportunidad para aplicar conceptos de manera estructurada, logrando así una base sólida en el ámbito del Internet de las Cosas y en el manejo de herramientas tecnológicas innovadoras.