

**Sistema de Monitoreo IoT para
Invernadero con ESP32**



ÍNDICE

MOTIVACIÓN.....	2
INTRODUCCIÓN.....	2
MATERIAL UTILIZADO.....	3
CONEXIONADO FÍSICO (Vídeo).....	4
CÓDIGO ARDUINO.....	5
RESULTADO (Vídeo completo del funcionamiento).....	6
FUTUROS DESARROLLOS.....	8
CONCLUSIÓN.....	9
PÁGINAS DE INTERÉS.....	9

MOTIVACIÓN

Este proyecto se ha realizado para demostrar los conocimientos de Arduino adquiridos en la asignatura de Sistemas Digitales. Lo que me motivó a elegir y desarrollar este proyecto fue la necesidad de monitorizar constantemente las condiciones de un invernadero.

Las plantas requieren un entorno específico de temperatura y humedad para crecer de forma óptima. Este sistema combina sensores de temperatura, humedad y movimiento, junto con un bot de Telegram, para proporcionar monitorización en tiempo real y control remoto del invernadero. La integración con ThingSpeak permite almacenar y analizar datos, lo que facilita el ajuste de las condiciones para lograr un entorno ideal. Además, el sistema de alertas por movimiento y alarma añade una capa de seguridad que ayuda a prevenir accesos no deseados. Este proyecto pretende simplificar el mantenimiento de las plantas, asegurando que sus necesidades se cumplan de manera efectiva y conveniente sin estar físicamente en el lugar.

INTRODUCCIÓN

Arduino es una plataforma de desarrollo de código abierto, utilizada para proyectos de automatización y control gracias a su facilidad de uso. La popularidad de Arduino se debe a su accesible combinación de microcontroladores y un entorno de programación intuitivo, que permite integrar y controlar diversos sensores y dispositivos de forma rápida.

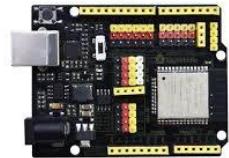
Este proyecto de vigilancia para un invernadero aprovecha las capacidades de la placa ESP32 (una versión avanzada compatible con Arduino, que incluye conectividad WiFi) para crear un sistema de seguimiento en tiempo real de las condiciones internas del cultivo. Mediante el sensor de temperatura y humedad DHT11, el sistema recopila datos ambientales que se muestran en una pantalla LCD y se envían a la plataforma en la nube ThingSpeak, lo que permite monitorizar y analizar estas variables a lo largo del tiempo. Además, se ha incluido un sensor de movimiento por infrarrojos, que alerta al usuario en caso de detectar actividad en el área del invernadero.

Para facilitar el acceso y la supervisión remota, el ESP32 se conecta a una red WiFi y utiliza un bot de Telegram que envía notificaciones de manera instantánea, permitiendo al usuario recibir alertas de movimiento o activar una alarma visual mediante el led rojo en caso de ser necesario. A su vez, un led azul indica que el usuario ha pedido en el bot de Telegram que se haga una foto.

De esta manera, el proyecto ofrece un sistema de monitorización completo y seguro del entorno del invernadero, brindando la posibilidad de actuar ante cambios o amenazas de manera rápida y eficiente. En conjunto, este sistema simplifica el mantenimiento de las plantas y asegura condiciones óptimas de cultivo, maximizando así el potencial del ESP32 en aplicaciones de agricultura inteligente y control remoto.

MATERIAL UTILIZADO

Para la creación del circuito físico del proyecto se han utilizado los siguientes componentes:

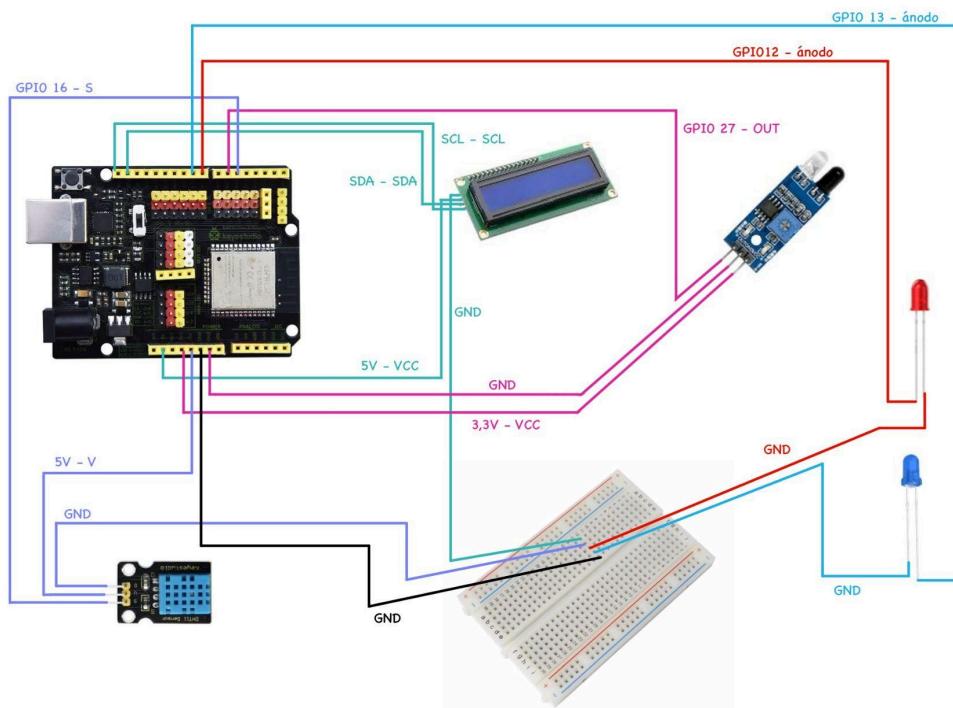
- **Placa ESP32 STEAMakers de Keyestudio:** Este es el microcontrolador principal del proyecto. Se encarga de gestionar todas las operaciones y comunicaciones del sistema. La ESP32 proporciona conectividad WiFi, lo que permite enviar los datos a plataformas en la nube como ThingSpeak y comunicarse con el bot de Telegram para recibir notificaciones y dar instrucciones en tiempo real.
- **LED Azul y LED Rojo:** Estos LEDs se utilizan como indicadores visuales de diferentes estados del sistema. El **LED azul** se enciende para indicar que el sistema ha recibido un comando específico desde el bot de Telegram. Por su parte, el **LED rojo** funciona como un indicador de alarma. Es decir, se enciende cuando recibe otro comando en Telegram, alertando de una posible intrusión en el invernadero.
- **Sensor de infrarrojos:** Este sensor de movimiento se utiliza para detectar la presencia de personas o animales en el área del invernadero. Cuando el sensor de movimiento por infrarrojos detecta movimiento, el sistema envía una alerta a través del bot de Telegram. Esto ayuda a proteger el invernadero y a evitar accesos no autorizados.
- **Sensor DHT11 (Keyestudio):** Este sensor mide la temperatura y la humedad del entorno dentro del invernadero. La placa ESP32 lee los datos del DHT11 periódicamente (cada 15 segundos) y los envía a ThingSpeak para crear gráficas de históricos, además de mostrarlos en la pantalla LCD para una consulta rápida en el propio invernadero.
- **Pantalla LCD:** La pantalla LCD se usa para mostrar en tiempo real los datos de temperatura y humedad, así como mensajes del sistema. Esto permite que cualquier persona cerca del invernadero pueda revisar las condiciones del entorno sin necesidad de conectarse a la red.

Cada uno de estos componentes se integra en el sistema para crear una solución de control y monitorización autónoma, con notificaciones en tiempo real y visualización local de los datos.

CONEXIONADO FÍSICO ([Vídeo](#))

Para este proyecto, se ha utilizado la placa ESP32 Steam Maker y varios sensores conectados directamente a esta ya que la placa provee las conexiones necesarias para controlar cada componente. Además se han añadido dos LEDs montados en una protoboard. El conexionado más específico se presenta a continuación:

1. Sensor de temperatura y humedad: El Pin de Datos se conecta al pin GPIO 16 de la ESP32. El pin de alimentación (VCC) se conecta al pin de 5V en la ESP32. Por último, el pin de tierra (GND) se conecta a la línea de GND de la protoboard.
2. Sensor de movimiento por infrarrojos: El pin de señal (OUT) se conecta al pin GPIO 27 de la ESP32. El pin de alimentación (VCC) se conecta al pin de 3,3V en la ESP32. Por último, el pin de tierra (GND) se conecta a la línea de GND de la protoboard.
3. LED azul: El ánodo (positivo) se conecta al GPIO 13 de la ESP32 y el cátodo (negativo) se conecta a la línea de GND de la protoboard.
4. LED rojo: El ánodo (positivo) se conecta al GPIO 12 de la ESP32 y el cátodo (negativo) se conecta a la línea de GND de la protoboard.
5. Pantalla LCD con interfaz I2C: La conexión SDA de la pantalla se conecta con el pin SDA en la ESP32. La conexión SCL de la pantalla se conecta con el pin SCL de la ESP32. El pin de alimentación (VCC) se conecta al pin de 5V en la ESP32. Por último, el pin de tierra (GND) se conecta a la línea de GND de la protoboard.
6. Conexión de la ESP32 a la protoboard: De la ESP32 sale una línea de tierra a la protoboard. Por otra parte, los pines GPIO 12 y GPIO 13 de la ESP32 también salen a la protoboard para permitir el conexionado de los LEDs.



CÓDIGO ARDUINO

El código completo de este proyecto, junto con esta misma explicación, se encuentra disponible en un repositorio de GitHub llamado [Invernadero_final](#).

La primera sección del código se centra en importar las bibliotecas necesarias y configurar los pines que la ESP32 utilizará para conectarse a los sensores y dispositivos externos. Estas bibliotecas permiten la comunicación con los sensores de temperatura, humedad, y movimiento, así como con las interfaces de red y de pantalla LCD.

En la siguiente sección, se establecen las variables de red y las APIs necesarias para la conectividad y la monitorización en remoto. Esto incluye las credenciales de WiFi para que el sistema pueda conectarse a Internet y las credenciales de las APIs para servicios externos, como Telegram y ThingSpeak. A través de la red WiFi, el proyecto puede enviar y recibir datos en tiempo real, logrando una integración completa con plataformas en la nube.

Después de definir las variables de red, se crean objetos específicos para controlar cada componente. Estos objetos son instancias de clases proporcionadas por las bibliotecas importadas y son esenciales para simplificar la interacción con los sensores y otros dispositivos. Por ejemplo, se crean objetos para el sensor de temperatura y humedad DHT, el sensor de movimiento por infrarrojos y la pantalla LCD. Cada objeto permite interactuar con los componentes mediante funciones sencillas, facilitando tanto la lectura de datos como la activación de dispositivos según se requiera.

A continuación, el código procede a la configuración inicial de los componentes, así como a la definición de variables de estado y de temporización. Estas variables ayudan a sincronizar el funcionamiento de los distintos sensores, permitiendo que el sistema lea datos de múltiples dispositivos al mismo tiempo sin conflictos. Las variables de estado también registran el último estado conocido de cada sensor o componente, mientras que las variables de temporización aseguran que las actualizaciones y lecturas se realicen en intervalos de tiempo definidos, optimizando el uso de la memoria y el rendimiento de la ESP32.

Seguidamente aparece la programación de las funciones necesarias para manejar tanto el bot de Telegram como cada sensor, de manera independiente:

1. **Función `handleNewMessages`**: Esta función está dedicada a procesar los mensajes entrantes en el bot de Telegram. Al recibir un mensaje, el sistema verifica que el *chat ID* del usuario coincida con el ID autorizado para garantizar que la comunicación es segura y está dirigida al bot correcto. Dependiendo del mensaje recibido, la función ejecuta diferentes acciones, que pueden incluir el control de dispositivos o la obtención de datos de sensores. Al final, envía una confirmación de vuelta al bot, permitiendo al usuario verificar que la acción solicitada se ha completado con éxito.

2. **Función *detectsMovement*:** La segunda función gestiona las interrupciones generadas por el sensor de movimiento por infrarrojos. Cada vez que el sensor detecta movimiento, esta función se activa automáticamente, permitiendo que el sistema responda de manera inmediata. Las interrupciones son útiles en este contexto, ya que permiten una detección rápida sin la necesidad de que el código esté constantemente monitoreando el estado del sensor.
3. **Función *leerdht1*:** La tercera función se encarga de leer los datos de temperatura y humedad del sensor DHT. Después de obtener estos valores, se envían a la plataforma ThingSpeak para su almacenamiento y análisis en la nube. Además, esta función también actualiza la pantalla LCD cada 15 segundos con las lecturas más recientes, proporcionando al usuario una visión en tiempo real del entorno monitoreado.

A continuación y tras las funciones se incluye el módulo *set up*, que lleva a cabo la configuración inicial de los periféricos y pines de la ESP32, incluyendo los pines asignados a los LEDs, el sensor de movimiento por infrarrojos y la pantalla LCD. Además, se establece la conexión WiFi, lo cual es esencial para el envío de datos y el control remoto. Durante esta fase, se configura la seguridad para la conexión de Telegram, asegurando que todas las comunicaciones con el bot estén protegidas. Finalmente, el sistema envía un mensaje de inicio a través de Telegram y ThingSpeak, confirmando que la conexión ha sido establecida y que el sistema está operativo.

Por último, se encuentra el módulo *loop*, es decir, el código entra en una fase de operación continua donde monitoriza activamente los sensores, recibe comandos a través de Telegram y actualiza ThingSpeak periódicamente con datos de temperatura y humedad. Este registro continuo garantiza que el sistema responda a eventos en tiempo real, como la detección de movimiento o solicitudes de datos del usuario a través de Telegram. La integración con ThingSpeak también permite al usuario consultar el historial de datos ambientales en la plataforma en la nube o en la aplicación disponible para dispositivos Android.

RESULTADO ([Vídeo completo del funcionamiento](#))

El resultado de este proyecto es un sistema de monitorización y control automatizado diseñado para supervisar y gestionar parámetros ambientales en tiempo real en un invernadero. Usando una ESP32 como placa de control, el sistema recopila datos de sensores de temperatura, humedad y movimiento, permitiendo obtener un análisis completo de las condiciones del entorno en el que está instalado, como un invernadero.

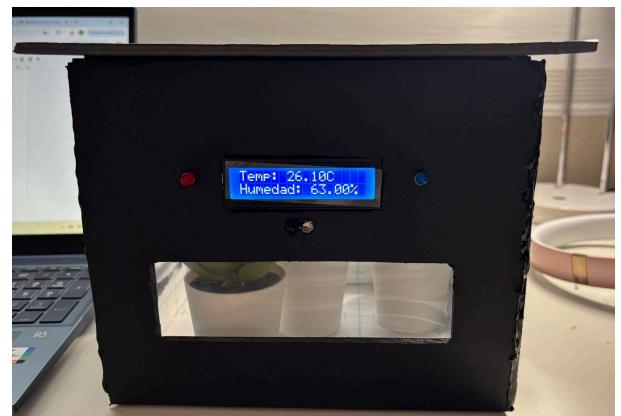
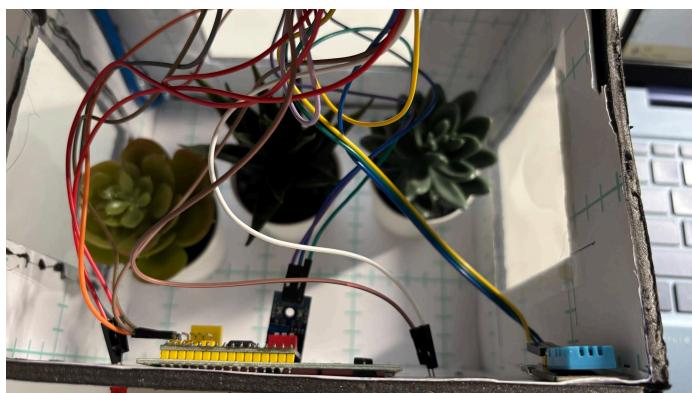
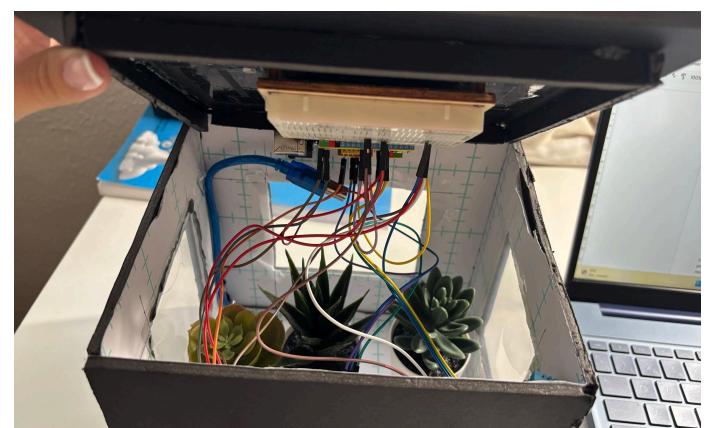
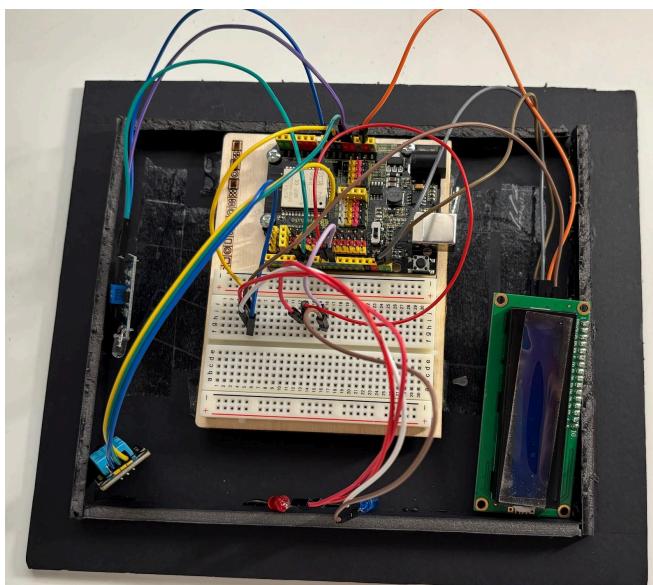
Los datos de temperatura y humedad son registrados y enviados a la plataforma en la nube ThingSpeak, donde pueden visualizarse y analizarse a lo largo del tiempo, permitiendo una mejor comprensión de las fluctuaciones y patrones ambientales. La pantalla LCD proporciona una

lectura visual directa de estos valores, actualizándose automáticamente cada 15 segundos con las mediciones más recientes.

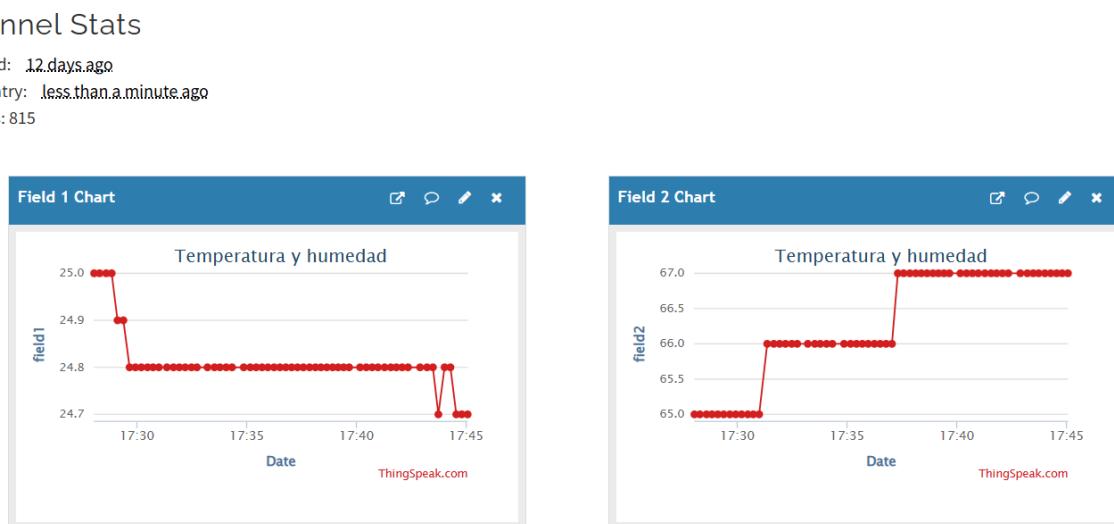
Además, el sistema permite la interacción a distancia mediante un bot de Telegram, ofreciendo al usuario la posibilidad de recibir notificaciones y enviar comandos desde cualquier ubicación con conexión a Internet. Este bot puede manejar mensajes del usuario para verificar el estado de los sensores, activar alertas, o recibir confirmaciones de acciones realizadas, como el envío de datos o activación de dispositivos de control.

Con esta implementación, el proyecto logra una solución para la monitorización ambiental remota y el control en tiempo real, ideal para aplicaciones de automatización en agricultura, domótica, y otros escenarios donde es esencial mantener ciertas condiciones de manera constante.

El resultado del maquetado físico sería:



Por otra parte, las gráficas históricas de temperatura y humedad que se vería en la página web de ThingSpeak o en la propia aplicación del móvil son:



FUTUROS DESARROLLOS

Para futuros desarrollos, el proyecto podría ampliarse y agregar funcionalidades o mejorar la capacidad de monitoreo visual y la interacción remota.

La primera idea sería la integración de una ESP32-CAM para control visual. Esto consistiría en cambiar el LED azul por una ESP32-CAM que permitiría al sistema capturar y transmitir imágenes del entorno en tiempo real. Esta integración proporcionaría al usuario una visual del espacio controlado. La ESP32-CAM también podría configurarse para capturar imágenes periódicamente o en respuesta a la detección de movimiento, ofreciendo alertas visuales en caso de actividad inesperada.

Por otra parte, se podría ampliar la funcionalidad del bot de Telegram: Actualmente, el bot de Telegram permite la comunicación básica para obtener lecturas de sensores y confirmar acciones. Un desarrollo futuro podría incluir funciones adicionales, como ofrecer un menú en el bot que permita al usuario acceder a distintas opciones, como activar dispositivos como ventiladores o rociadores de agua, etc. Estos dispositivos podrían activarse manualmente o se podría fijar una temperatura desde el propio bot para que se activen automáticamente. Además podría enviar alertas si se superan ciertos umbrales fijados o si se pierde la conexión.

CONCLUSIÓN

Este proyecto de invernadero inteligente permite desarrollar un sistema de monitorización ambiental que combina sensores de temperatura, humedad y movimiento con una interfaz visual para el usuario. Utilizando la placa ESP32 Steam Maker, se ha logrado integrar todos los componentes de forma eficiente, proporcionando conectividad y capacidad de procesamiento suficiente para operar el sistema de manera autónoma y confiable.

Los resultados han demostrado que el sistema es capaz de medir y mostrar condiciones ambientales en tiempo real, además de activar alarmas cuando se indica.

En conclusión, este proyecto no solo demuestra la viabilidad de un invernadero inteligente, sino que también proporciona una base para futuras ampliaciones, como el control remoto o la automatización de riego y ventilación, elevando el nivel de control y eficiencia en la agricultura bajo condiciones controladas.

PÁGINAS DE INTERÉS

Para la realización de este proyecto, he consultado diversas páginas web y vídeos explicativos en YouTube. El enlace a todas ellas se encuentra a continuación. La última visita a estas páginas se realizó el 10 de noviembre de 2024.

[Vídeo del sensor de movimiento IOT.](#)

[Vídeo de la creación del bot de Telegram.](#)

[Primer foro de Arduino.](#)

[Segundo foro de Arduino.](#)

[ChatGpt.](#)

[GitHub para la importación de bibliotecas.](#)

Por último veo importante añadir la página oficial de [ThingSpeak](#) ya que ha sido utilizada para la monitorización de los datos de temperatura y humedad.

La idea para la realización de este proyecto la he obtenido de diversos libros de tecnología de la Editorial Donostiarra.