



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Envío de datos desde un módulo ESP32 a Blynk con sensores SHT40 y Osram de proximidad y luminosidad

Omar Sánchez Gudiño.

MTRA EN COMPUTO APLICADO | INTERNET DE LAS COSAS.

8 de marzo de 2025

Resumen

En esta práctica, se desarrollará un módulo basado en ESP32 para la adquisición de datos de sensores de temperatura, humedad, proximidad y luminosidad. Los datos recopilados serán enviados a la plataforma Blynk mediante un protocolo de comunicación adecuado, permitiendo su monitoreo en tiempo real a través de una interfaz gráfica. Para lograrlo, se configurará y programará el ESP32 como microcontrolador principal, integrando los sensores correspondientes y utilizando las librerías necesarias. Además, se aplicarán conceptos de arquitectura IoT para garantizar una comunicación eficiente y confiable entre los dispositivos y la plataforma. Finalmente, se diseñará una interfaz en Blynk con widgets adecuados para la visualización dinámica de los datos.

ÍNDICE

1. <u>Objetivo.</u>	2
2. <u>Introducción.</u>	2
3. <u>Desarrollo.</u>	2
I. Configuración del ESP32 y conexión con Blynk.	2
II. Lectura y preprocesamiento de datos.	2
III. Integración de sensores	3
IV. Procesamiento y conversión de datos	3
V. Envío de datos a Blynk y control de Sensores	4
VI. Diseño de la Interfaz Gráfica en Blynk	4
4. <u>Resultados.</u>	4
5. <u>Conclusión.</u>	5
6. <u>Recursos.</u>	5
I. Repositorio en Github.	5
7. <u>Anexo.</u>	5
I. Caso de Uso: Monitoreo inteligente para un invernadero con IoT.	5
I.1. Situación y Problema	5
I.2. Solución con ESP32 y Blynk	5
I.3. Beneficios y Mejoras Posibles	6

1. OBJETIVO.

El objetivo de este artículo es diseñar, desarrollar e implementar un sistema de adquisición, transmisión y visualización de datos en tiempo real utilizando el microcontrolador ESP32 y la plataforma Blynk. Para ello, se integrarán una tarjeta con sensores capaces de medir temperatura, humedad, proximidad y luminosidad, permitiendo la captura y el procesamiento eficiente de la información. El proyecto busca establecer una comunicación confiable entre el ESP32 y Blynk mediante el uso de protocolos de comunicación adecuados, garantizando una transmisión de datos fluida y segura. A través de la implementación de conceptos fundamentales de la arquitectura IoT, se explorarán las mejores prácticas para la conectividad y gestión de dispositivos en entornos de monitoreo remoto. Además, se diseñará y configurará una interfaz gráfica interactiva en la plataforma Blynk, utilizando widgets adecuados para la representación visual de los datos en tiempo real. Esta interfaz permitirá a los usuarios supervisar las mediciones de los sensores de manera intuitiva, brindando una experiencia de usuario óptima y facilitando la interpretación de la información. Finalmente, este trabajo tiene como propósito fortalecer el conocimiento sobre la integración de microcontroladores con plataformas IoT, fomentando el desarrollo de aplicaciones en el ámbito del monitoreo ambiental, la automatización y el control remoto de dispositivos electrónicos.

2. INTRODUCCIÓN.

En la actualidad, el Internet de las Cosas (IoT) ha revolucionado la forma en que interactuamos con dispositivos electrónicos, permitiendo la recopilación, transmisión y análisis de datos en tiempo real. Gracias a la integración de sensores con microcontroladores y plataformas en la nube, es posible monitorear diversas variables ambientales de manera remota y eficiente. Este tipo de aplicaciones resulta especialmente útil en áreas como la domótica, la automatización industrial y la gestión de recursos energéticos. En este contexto, el presente trabajo se centra en el desarrollo de un sistema basado en el microcontrolador ESP32 para la adquisición de datos provenientes de sensores de temperatura, humedad, proximidad y luminosidad.

La información recopilada será transmitida a la plataforma Blynk, donde podrá visualizarse a través de una interfaz gráfica diseñada específicamente para este propósito. El ESP32 se ha convertido en una opción popular para proyectos de IoT debido a su bajo consumo energético, conectividad Wi-Fi y Bluetooth, y su capacidad de procesamiento eficiente. Su integración con sensores permite la captura de datos en tiempo real, mientras que la comunicación con Blynk facilita el almacenamiento y la representación de la información en un entorno accesible desde dispositivos móviles o computadoras. Para lograr estos objetivos, el sistema propuesto abordará distintos aspectos clave del desarrollo IoT, incluyendo la configuración y programación del ESP32, la integración de sensores, la implementación de protocolos de comunicación y el diseño de una interfaz de usuario intuitiva en Blynk. A través de este enfoque, se busca no solo demostrar la viabilidad de este tipo de sistemas en aplicaciones prácticas, sino también proporcionar un marco de referencia para futuros desarrollos en el ámbito de la automatización y el monitoreo remoto.

3. DESARROLLO.

I. Configuración del ESP32 y conexión con Blynk.

El sistema se basa en el microcontrolador ESP32, el cual se encarga de la adquisición y transmisión de datos de sensores a la plataforma Blynk. Para ello, en la fase de configuración inicial, se definen las credenciales de red Wi-Fi y el token de autenticación de Blynk, lo que permite la conexión del ESP32 a la nube.

II. Lectura y preprocesamiento de datos.

El primer paso en el procesamiento de la señal es la lectura de los datos desde el archivo CSV y su organización en función del tiempo. Para ello, se emplea la biblioteca pandas, que permite cargar los datos y convertir la columna de tiempo a un formato de fecha y hora. Posteriormente, los datos se ordenan cronológicamente para facilitar su análisis.

```
char ssid[] = "NOMBRE_RED";  
char pass[] = "CONTRASEÑA_RED";
```

```
#define BLYNK_AUTH_TOKEN "TOKEN"
```

En la función `setup()`, se inicializa la comunicación con Blynk utilizando estas credenciales:

```
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
```

Esta configuración permite que el ESP32 envíe los datos adquiridos por los sensores a la plataforma para su monitoreo en tiempo real.

III. Integración de sensores

El sistema implementa dos tipos de sensores:

- Sensor SHT4x: Se utiliza para medir temperatura y humedad. Su configuración se realiza dentro de la función `setup()`, donde se verifica la comunicación y se establecen parámetros como la precisión de medición y el estado del calentador integrado.

```
if (!sht4.begin())
{
  Serial.println("Couldn't find SHT4x");
  while (1) delay(1);
}
sht4.setPrecision(SHT4XHIGHPRECISION);
sht4.setHeater(SHT4X_NO_HEATER);
```

- Sensor SFH7779: Este sensor se encarga de medir proximidad y luminosidad. Su comunicación con el ESP32 se realiza a través del protocolo I2C. Dentro de `setup()`, se configura el sensor escribiendo valores específicos en sus registros mediante la función `Wire.write()`.

```
Wire.beginTransmission(SFH7779);
Wire.write(0x41);
Wire.write(0x09);
Wire.write(0x2B);
Wire.endTransmission();
```

Además, la función `ReadSensor()` lee los valores del sensor y almacena los datos en un arreglo de 6 bytes llamado `SensorValueCatch[]`:

```
void ReadSensor(void)
{
  Wire.beginTransmission(SFH7779);
  Wire.write(0x44);
  Wire.endTransmission();
  Wire.requestFrom(SFH7779, 6);
  uint8_t count1 = 0;
  while (Wire.available())
  {
    SensorValueCatch[count1]=Wire.read();
    count1++;
  }
}
```

IV. Procesamiento y conversión de datos

Para obtener la luminosidad en unidades de lux, se implementa la función `Lux_Value()`, la cual realiza una conversión basada en los valores de luz visible (VIS) e infrarroja (IR) obtenidos del sensor SFH7779.

```
float Lux_Value(uint16_t VIS, uint16_t IR)
{
  float LUX;
  float Div = float(IR) / float(VIS);
  if (Div < 0.109)
  {
    LUX = (1.534 * VIS) - (3.759 * IR);
  }
  else if (Div < 0.429)
  {
    LUX = (1.339 * VIS) - (1.972 * IR);
  }
  else if (Div < (1.3775))
  {
    LUX = (0.701 * VIS) - (0.483 * IR);
  }
  else if (Div < (2.175))
  {
    LUX = (1.402 * VIS) - (.56994 * IR);
  }
  else if (Div < (3.625))
  {
    LUX = (2.804 * VIS) - (.64239 * IR);
  }
  else
```

```

{
    LUX = (5.608 * VIS);
}
LUX = LUX / 64;
return (LUX);
}

```

Esta función se utiliza en `loop()` para calcular la luminosidad a partir de los datos crudos del sensor.

V. Envío de datos a Blynk y control de Sensores

Dentro de `loop()`, se ejecuta `Blynk.run()` para mantener la conexión con la plataforma. Luego, si el usuario activa la variable `switchState` desde la aplicación en Blynk, el ESP32 toma las mediciones de los sensores y las envía a los pines virtuales de la plataforma.

```

if(switchState == 1)
{
    sensors_event_t humidity, temp;
    uint16_t error, prox;
    uint32_t timestamp = millis();
    float lum;
    ReadSensor();
    lum = (Lux_Value((SensorValue)));
    prox = ((SensorValueCatch));
    timestamp = millis() - timestamp;
    Blynk.virtualWrite(V0, temperature);
    Blynk.virtualWrite(V1, humidity);
    Blynk.virtualWrite(V2, lum);
    Blynk.virtualWrite(V3, prox);
}

```

Si la luminosidad supera un umbral de 100 lux, se genera un evento en Blynk para alertar sobre este cambio:

```

if (lum > 100)
{
    Blynk.logEvent("luminosidad_alta");
    Serial.println("Se ejecutó");
}

```

Cuando `switchState` está desactivado, se envían valores de cero a la plataforma:

```

else
{
    Blynk.virtualWrite(V0, 0);
    Blynk.virtualWrite(V1, 0);
    Blynk.virtualWrite(V2, 0);
    Blynk.virtualWrite(V3, 0);
}

```

VI. Diseño de la Interfaz Gráfica en Blynk

En la aplicación de Blynk, se crean widgets para visualizar los datos de cada sensor en tiempo real:

- Temperatura (V0) y Humedad (V1): Se representan mediante gráficas o etiquetas de texto.
- Luminosidad (V2): Se muestra en un medidor tipo "Gauge".
- Proximidad (V3): Puede visualizarse en una barra de progreso o un indicador de estado.

Además, se incorpora un botón para activar y desactivar la medición (V4), lo que permite controlar el flujo de datos desde la aplicación.

4. RESULTADOS.

Como resultado, tenemos un código funcional el cual nos ayuda a alimentar nuestra GUI realizada en Blynk la cual luce de la siguiente manera

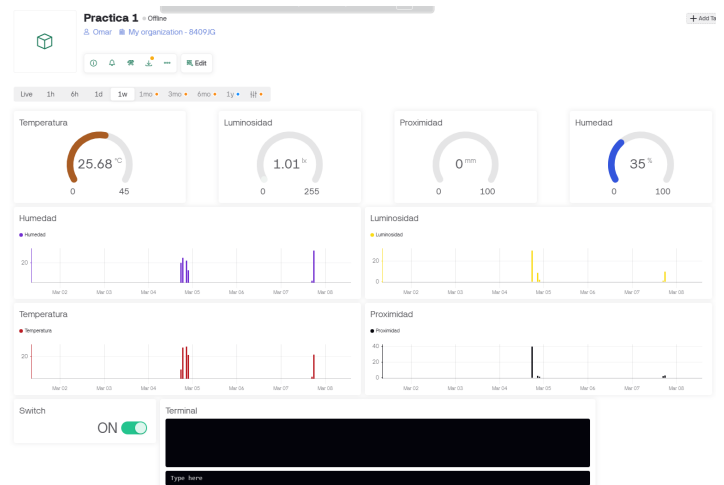


Figura 1: Los datos mostrados son de la semana de trabajo en la que se realizaron pruebas.

La implementación de la interfaz gráfica en Blynk permite la visualización en tiempo real de las variables monitoreadas por el sistema basado en ESP32. La plataforma facilita la representación intuitiva de datos a través de widgets interactivos, proporcionando un monitoreo eficiente del ambiente. Como se puede apreciar los datos se adquieren de manera adecuada y se muestran de forma muy intuitiva para que el usuario pueda entender como es que el sensor estuvo operando.

5. CONCLUSIÓN.

El desarrollo de este sistema basado en ESP32 y la plataforma Blynk demuestra la integración efectiva de tecnologías IoT para la adquisición, procesamiento y monitoreo de datos en tiempo real. A través de la implementación de sensores de temperatura, humedad, luminosidad y proximidad, se logra un sistema capaz de recopilar información ambiental y transmitirla a la nube, donde puede visualizarse de manera intuitiva en una interfaz gráfica. El uso de protocolos de comunicación adecuados, como I2C para la lectura de sensores y Wi-Fi para la conexión con Blynk, garantiza un flujo de datos eficiente y en tiempo real. Además, la estructura del código permite una fácil escalabilidad y adaptación a otros proyectos, lo que abre la posibilidad de integrar nuevos sensores o funcionalidades avanzadas, como la automatización basada en umbrales predefinidos. Desde el punto de vista práctico, esta implementación puede aplicarse en diversas áreas, como el monitoreo de condiciones ambientales en invernaderos, sistemas de control de iluminación inteligente o dispositivos de seguridad con sensores de proximidad. La combinación de hardware y software en este proyecto demuestra la versatilidad del ESP32 y su potencial en soluciones IoT. Finalmente, este trabajo no solo refuerza el conocimiento sobre el desarrollo de sistemas embebidos, sino que también brinda una base sólida para futuras investigaciones y mejoras. Optimizar el consumo energético del sistema, mejorar la precisión de los sensores o integrar modelos de inteligencia artificial para la predicción de condiciones ambientales son algunas de las posibles extensiones que podrían llevar este proyecto a un nivel más avanzado.

6. RECURSOS.

1. Repositorio en Github.

En el siguiente enlace pueden encontrar el script generado para este experimento: https://github.com/OmarGudi/IoT_ESP32-Blynk-SHT40-0sram

7. ANEXO.

1. Caso de Uso: Monitoreo inteligente para un invernadero con IoT.

1.1. Situación y Problema

En un invernadero, es fundamental mantener condiciones ambientales óptimas para el crecimiento de las plantas. Factores como la temperatura, la humedad y la luz afectan directamente el desarrollo de los cultivos. Si estos parámetros no se controlan adecuadamente, pueden generarse problemas como deshidratación, enfermedades fúngicas o falta de luz suficiente para la fotosíntesis. Además, la presencia de personal en ciertas áreas puede influir en los ajustes del sistema, por lo que un sensor de proximidad podría ayudar a activar o desactivar ciertos mecanismos automáticamente.

1.2. Solución con ESP32 y Blynk

Un sistema basado en ESP32, junto con los sensores SHT40 y Osram de proximidad y luminosidad, podría implementarse en el invernadero para monitorear las condiciones en tiempo real y enviar los datos a la plataforma Blynk.

- Temperatura y humedad: El sensor SHT40 mide ambos valores, permitiendo al sistema activar ventiladores o humidificadores automáticamente si los niveles no son adecuados.
- Luminosidad: Con el sensor de Osram, el sistema puede determinar si la luz natural es suficiente o si es necesario encender luces artificiales para mantener la fotosíntesis óptima.
- Proximidad: Si el sensor detecta movimiento en ciertas áreas del invernadero, puede activar iluminación o alertar a los trabajadores si están en una zona de riesgo.

Los datos recolectados se envían a la nube a través de Wi-Fi y se visualizan en tiempo real en la aplicación de Blynk, lo que permite a los agricultores supervisar las condiciones desde cualquier lugar y recibir alertas si algún parámetro sale de los rangos establecidos.

1.3. Beneficios y Mejoras Posibles

Beneficios:

- Automatización del control climático en el invernadero.
- Reducción de desperdicio de energía al encender sistemas solo cuando es necesario.
- Acceso remoto a los datos en tiempo real, mejorando la toma de decisiones.

- Reducción de la intervención manual, permitiendo a los agricultores enfocarse en otras tareas.

Mejoras Posibles:

- Implementar inteligencia artificial para predecir cambios ambientales y ajustar los sistemas con anticipación.
- Agregar más sensores, como medidores de CO₂ o humedad del suelo, para una supervisión más completa.
- Integrar baterías o paneles solares para hacer el sistema más eficiente energéticamente.