



**LOS LIBERTADORES**

FUNDACIÓN UNIVERSITARIA

Acreditación Institucional en Alta Calidad

# **Informe de Implementación de Conectividad IoT en un Sistema Fotovoltaico Usando ESP32 y ThingSpeak**

Realizado por:

**Ing. Moisés Ibarra García**

**Ing. Jessica Moreno**

**Ing. David Velásquez**

**Maestría en Ingeniería**

**Facultad de Ingeniería y Ciencias Básicas**

**Fundación Universitaria Los**

**Libertadores**



**LOS LIBERTADORES**

FUNDACIÓN UNIVERSITARIA

Acreditación Institucional en Alta Calidad

## Tabla de contenido

### Contenido

Introducción .....	3
Objetivo de las Variables Medidas en el Proyecto.....	3
Medición de Temperatura Utilizando MAX6675: .....	3
Objetivo Técnico:.....	3
Aplicación Práctica: .....	4
Detección de Lluvia con el Sensor YL-83: .....	4
Objetivo Técnico:.....	4
Aplicación Práctica: .....	4
Conectividad WiFi Mediante ESP32: .....	4
Objetivo Técnico:.....	4
Aplicación Práctica: .....	4
Integración de Datos con ThingSpeak:.....	4
Objetivo Técnico:.....	4
Aplicación Práctica: .....	5
Conclusiones .....	5
Anexos: Evidencia de Ejecución de código:.....	6
Anexos: Montaje para la toma de datos:.....	7
Anexos: Código desarrollado en Arduino IDE: .....	8



## Introducción

En la era actual, donde la sostenibilidad y la eficiencia energética son de suma importancia, la integración de tecnologías del Internet de las Cosas (IoT) en sistemas fotovoltaicos representa un avance significativo. Este informe expone un proyecto que combina la tecnología IoT con un sistema fotovoltaico para mejorar su monitoreo y gestión. El proyecto utiliza la placa LILYGO® TTGO T-Display ESP32, conocida por sus capacidades avanzadas de conectividad y procesamiento, junto con el sensor de lluvia YL-83 para recopilar datos ambientales relevantes. Además, se emplea la plataforma ThingSpeak para el análisis y almacenamiento de datos, aprovechando las ventajas de la conectividad IoT para un monitoreo eficiente y en tiempo real.

La implementación se realiza a través del Arduino IDE, una herramienta accesible para programar microcontroladores, lo que demuestra la viabilidad de aplicar soluciones IoT complejas con recursos y herramientas accesibles. El objetivo principal es establecer un sistema capaz de recoger y transmitir datos críticos del entorno, como la temperatura y la presencia de lluvia, para optimizar el rendimiento de los sistemas fotovoltaicos. Estos datos no solo son cruciales para el mantenimiento preventivo y la reducción de costos operativos, sino también para garantizar la máxima eficiencia energética en condiciones cambiantes.

Este proyecto no solo refleja la importancia de integrar tecnologías emergentes en el campo de las energías renovables, sino que también destaca el potencial de IoT en la transformación de la gestión y supervisión de sistemas energéticos. La aplicación de estos avances en un sistema fotovoltaico abre nuevas posibilidades para la optimización de recursos, la gestión inteligente y la sostenibilidad en el sector energético.

## Objetivo de las Variables Medidas en el Proyecto

En el contexto de un proyecto de maestría que aborda la integración de tecnologías IoT en sistemas fotovoltaicos, las variables recopiladas mediante la implementación de sensores y microcontroladores son fundamentales para la operación y el mantenimiento eficiente de estos sistemas. A continuación, se detallan los objetivos específicos de cada variable medida:

### Medición de Temperatura Utilizando MAX6675:

#### Objetivo Técnico:

El sensor MAX6675 se emplea para medir con precisión la temperatura superficial de los paneles fotovoltaicos. Esta medición es crítica, dado que la eficiencia fotovoltaica puede verse significativamente comprometida por variaciones térmicas. La correlación entre la temperatura de la célula y la eficiencia del panel es un parámetro clave en la evaluación del rendimiento fotovoltaico.



## **Aplicación Práctica:**

La información térmica obtenida permite al ingeniero o al técnico especializado en sistemas fotovoltaicos realizar ajustes operativos para optimizar la eficiencia del panel. Además, sirve como un indicador crítico para el mantenimiento preventivo, ayudando a evitar el deterioro acelerado del sistema debido a condiciones de estrés térmico.

## **Detección de Lluvia con el Sensor YL-83:**

### **Objetivo Técnico:**

El sensor YL-83 proporciona datos en tiempo real sobre la incidencia de precipitaciones. Este parámetro es esencial para comprender cómo las condiciones de lluvia afectan la acumulación de suciedad y los residuos en los paneles, lo cual puede reducir su capacidad de captación solar.

### **Aplicación Práctica:**

Desde la perspectiva de la gestión de sistemas fotovoltaicos, la información sobre la precipitación es utilizada para programar las operaciones de limpieza y mantenimiento de los paneles. Además, ayuda a los analistas de datos y a los ingenieros de sistemas a ajustar los modelos de predicción de rendimiento energético según las variaciones climáticas.

## **Conectividad WiFi Mediante ESP32:**

### **Objetivo Técnico:**

El uso del ESP32 para la conectividad WiFi es crucial en la transmisión de datos ambientales a plataformas de análisis en la nube. Esta conectividad inalámbrica facilita la recopilación y el envío de datos de los sensores a un sistema centralizado de monitoreo y análisis.

### **Aplicación Práctica:**

Los especialistas en sistemas de energía renovable utilizan esta conectividad para acceder de forma remota a los datos recopilados, permitiendo una supervisión continua y una respuesta rápida a cualquier anomalía detectada. Este enfoque mejora significativamente la gestión de activos y la eficiencia operativa del sistema fotovoltaico.

## **Integración de Datos con ThingSpeak:**

### **Objetivo Técnico:**

La integración de los datos recopilados con ThingSpeak permite su almacenamiento seguro, análisis avanzado y visualización. ThingSpeak ofrece una plataforma robusta para el manejo de datos de IoT, con capacidades para el procesamiento de grandes volúmenes de información y la generación de insights accionables.



## Aplicación Práctica:

Los investigadores y profesionales del sector de energías renovables utilizan ThingSpeak para analizar tendencias, realizar diagnósticos remotos y desarrollar estrategias de mantenimiento predictivo. La capacidad de visualizar y analizar estos datos facilita la toma de decisiones basada en evidencia y la optimización continua del sistema fotovoltaico.

## Conclusiones

El proyecto de la Electiva III ha demostrado con éxito la viabilidad y los beneficios de integrar tecnologías del Internet de las Cosas (IoT) en sistemas fotovoltaicos, empleando la innovadora placa LILYGO® TTGO T-Display ESP32 y el sensor de lluvia YL-83. Esta integración ha resultado en un avance significativo en la eficiencia operativa y el mantenimiento de los sistemas fotovoltaicos. Mediante la implementación de sensores precisos y conectividad avanzada, se ha logrado un monitoreo en tiempo real de variables ambientales críticas. Esta capacidad de monitoreo constante es esencial para optimizar el rendimiento de los paneles solares, adaptándose a las fluctuaciones térmicas y a las condiciones de humedad, lo que es vital para prevenir el sobrecalentamiento y los daños a largo plazo en el sistema. Además, la recopilación detallada de datos permite anticipar las necesidades de mantenimiento, reduciendo así los costos operativos y extendiendo la vida útil de los componentes del sistema.

Por otro lado, la utilización de ThingSpeak como plataforma de gestión de datos ha demostrado ser una herramienta valiosa en el ámbito del IoT aplicado a las energías renovables. Proporciona un medio eficiente para el almacenamiento, análisis y visualización de datos, lo cual es crucial para la toma de decisiones informadas y basadas en datos. Este enfoque analítico y basado en datos no solo mejora la gestión del día a día de los sistemas fotovoltaicos, sino que también abre nuevas posibilidades para la investigación y el desarrollo en el sector de la energía solar. El éxito de este proyecto pone de manifiesto el potencial de las tecnologías IoT para transformar la gestión y operación de los sistemas de energía renovable, señalando un camino prometedor hacia la innovación y la eficiencia en este campo.

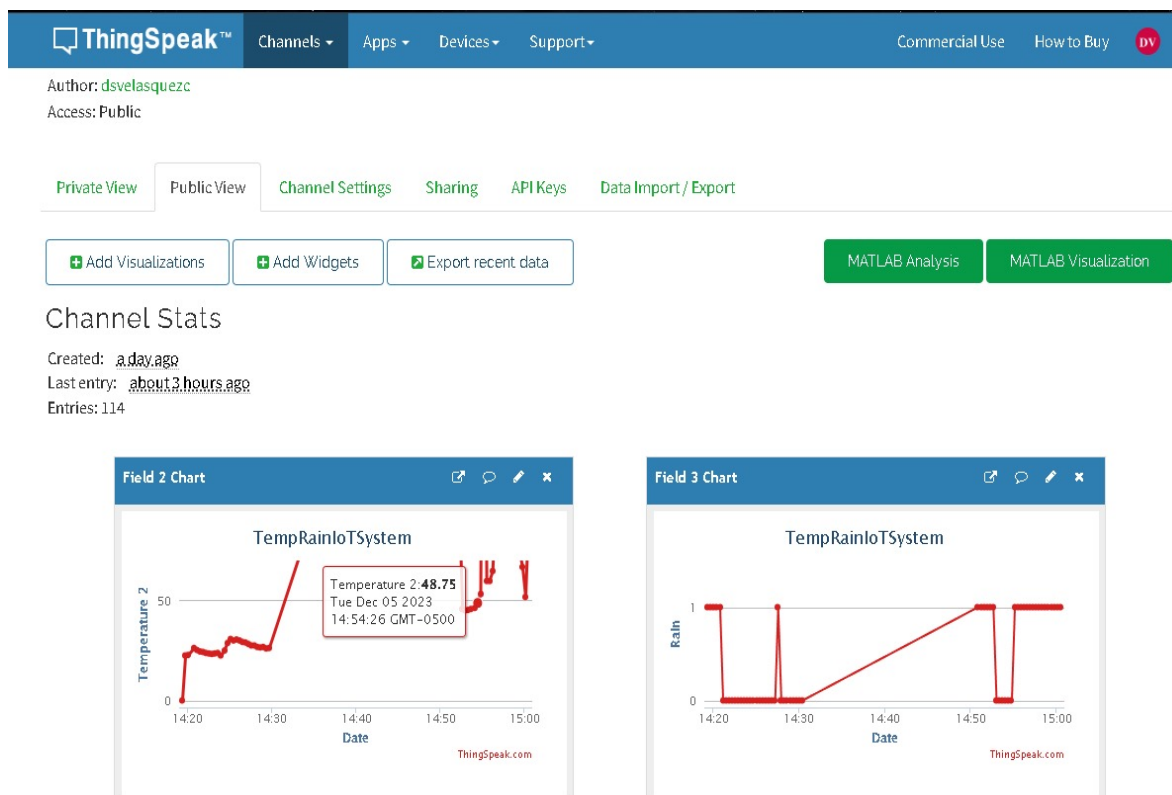


LOS LIBERTADORES

FUNDACIÓN UNIVERSITARIA

Acreditación Institucional en Alta Calidad

## Anexos: Evidencia de Ejecución de código:



**Figura. 1.** Visualización de los datos de las variables de temperatura y lluvia en la plataforma ThingSpeak.





**LOS LIBERTADORES**

FUNDACIÓN UNIVERSITARIA

Acreditación Institucional en Alta Calidad

## **Anexos: Montaje para la toma de datos:**



**Figura. 2.** *Instalación de los sensores de temperatura y sensor de lluvia.*



## Anexos: Código desarrollado en Arduino IDE:

```
#include <WiFi.h> // Incluye la biblioteca WiFi para
conectar el ESP32 a una red inalámbrica
#include <ThingSpeak.h> // Incluye la biblioteca ThingSpeak
para enviar datos a ThingSpeak
#include <SPI.h> // Incluye la biblioteca para la
comunicación SPI (Serial Peripheral Interface)
#include "max6675.h" // Incluye la biblioteca para el
sensor de temperatura MAX6675

// Credenciales WiFi
const char* ssid = "iPhoneDV"; // Define el SSID (nombre) de la red
WiFi a la que se conectará
const char* password = "David12345"; // Define la contraseña de la red
WiFi

// Credenciales de ThingSpeak
unsigned long myChannelNumber = 2368512; // Define el número de canal de
ThingSpeak para enviar datos
const char * myWriteAPIKey = "BOJZOWKDXML541IN"; // Define la clave de API
de escritura para ThingSpeak

// Configuración MAX6675
int thermoSCLK = 19; // Define el pin de reloj para los
sensores MAX6675
int thermoCS1 = 18; // Define el pin de selección de
chip para la primera termocupla
int thermoCS2 = 5; // Define el pin de selección de
chip para la segunda termocupla
int thermoMISO = 23; // Define el pin de salida de datos
maestro para los sensores MAX6675

MAX6675 termocupla1(thermoSCLK, thermoCS1, thermoMISO); // Inicializa la
primera termocupla MAX6675
MAX6675 termocupla2(thermoSCLK, thermoCS2, thermoMISO); // Inicializa la
segunda termocupla MAX6675

// Configuración del sensor de lluvia
const int pinSensorLluvia = 34; // Define el pin al que está
conectado el sensor de lluvia
```





```
WiFiClient cliente; // Crea un objeto cliente para la
conexión WiFi

void setup() {
  Serial.begin(9600); // Inicia la comunicación serial a
9600 baudios

  // Intento de conexión a WiFi
  Serial.println("Conectando a WiFi..."); // Imprime un mensaje en la
consola serial
  WiFi.begin(ssid, password); // Inicia la conexión WiFi con las
credenciales definidas

  // Esperar hasta que se conecte a WiFi
  while (WiFi.status() != WL_CONNECTED) { // Mientras la conexión WiFi no
esté establecida
    delay(500); // Espera 500 milisegundos
    Serial.print("."); // Imprime un punto en la consola
serial
  }

  Serial.println("\nWiFi Conectado."); // Una vez conectado, imprime un
mensaje en la consola serial
  Serial.print("Dirección IP: "); // Imprime un mensaje solicitando
la dirección IP
  Serial.println(WiFi.localIP()); // Imprime la dirección IP asignada
al ESP32

  ThingSpeak.begin(cliente); // Inicia la conexión con
ThingSpeak
  pinMode(pinSensorLluvia, INPUT); // Configura el pin del sensor de
lluvia como entrada
}

void loop() {
  // Leer temperaturas
  float temp1 = termocupla1.readCelsius(); // Lee la temperatura en Celsius
de la primera termocupla
  float temp2 = termocupla2.readCelsius(); // Lee la temperatura en Celsius
de la segunda termocupla

  // Leer sensor de lluvia
  int estadoLluvia = digitalRead(pinSensorLluvia); // Lee el estado del
sensor de lluvia (alto/bajo)
```



```
// Imprimir en Serial (para depuración)
Serial.print("Temperatura 1: ");          // Imprime un mensaje sobre la
primera temperatura
Serial.println(temp1);                    // Imprime la primera temperatura
Serial.print("Temperatura 2: ");          // Imprime un mensaje sobre la
segunda temperatura
Serial.println(temp2);                    // Imprime la segunda temperatura
Serial.print("Estado de Lluvia: ");       // Imprime un mensaje sobre el
estado del sensor de lluvia
Serial.println(estadoLluvia);             // Imprime el estado del sensor de
lluvia

// Enviar a ThingSpeak
ThingSpeak.setField(1, temp1);            // Asigna la primera temperatura al
campo 1 de ThingSpeak
ThingSpeak.setField(2, temp2);            // Asigna la segunda temperatura al
campo 2 de ThingSpeak
ThingSpeak.setField(3, estadoLluvia);     // Asigna el estado de lluvia al
campo 3 de ThingSpeak

ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey); // Envía los datos
a ThingSpeak

delay(10000);                             // Espera 10 segundos (10000
milisegundos) antes de repetir el loop
}
```