



Desarrollo de un Sistema IoT Básico para Monitoreo en Tiempo Real de Consumo y Demanda Energética Doméstica Usando Microcontrolador ESP32 y ThingSpeak de Matlab

Román-Reyes J.E.¹; González-Vargas, L. A.¹ ✉; Alvarado-Tovar, N.²; Manqueros-Avilés, V.E.¹

Datos de Adscripción:

¹ Tecnológico Nacional de México / Instituto Tecnológico Superior de Lerdo. Av. Tecnológico No.1555 Sur, Periférico Gómez-Lerdo Km. 14.5. Ciudad Lerdo, Estado de Durango. México. C.P. 35150.

² Tecnológico Nacional de México / Instituto Tecnológico de la Laguna. Blvd. Revolución y Av. Instituto Tecnológico de La Laguna, Torreón, Coahuila, C.P. 27000

✉ luis.gv@lerdo.tecnm.mx

Resumen – El artículo presenta la implementación detallada de un sistema IoT para monitorear en tiempo real la demanda energética doméstica. La aplicación de IoT en el monitoreo del consumo de energía en entornos domésticos es crucial para promover la eficiencia energética y reducir costos. Permite una supervisión continua y en tiempo real de los patrones de consumo, facilitando la identificación de áreas de mejora y la toma de decisiones informadas para optimizar el uso de la energía. En este proyecto, se utiliza un microcontrolador ESP32 junto con sensores de corriente (transformadores de corriente) para capturar datos sobre el consumo energético. Para la visualización remota de estos datos, se emplea ThingSpeak, una plataforma basada en MATLAB que permite trabajar con aplicaciones IoT. El proyecto se centra en la implementación de un sistema IoT básico para monitorear en tiempo real el consumo y la demanda energética doméstica. Se configura el hardware mencionado para recopilar datos de consumo de energía, que luego se transmiten a ThingSpeak para visualización local y remota. Los usuarios pueden acceder a esta información tanto localmente, a través de un display, como de manera remota utilizando la herramienta ThingSpeak. Se obtienen resultados significativos. Los usuarios pueden monitorear su consumo energético de forma precisa y en tiempo real, lo que les permite tomar medidas para mejorar la eficiencia y reducir el desperdicio de energía. La combinación de hardware y software seleccionados ha demostrado ser efectiva para este propósito, destacando el potencial de la tecnología IoT en la gestión energética residencial

Palabras Clave: Demanda, Energía, IoT, Microcontrolador ESP32, Monitoreo energético, ThingSpeak.

Abstract – The article presents the detailed implementation of an IoT system for real-time monitoring of household energy demand. The application of IoT in monitoring energy consumption in domestic environments is crucial for promoting energy efficiency and reducing costs. It enables continuous and real-time monitoring of consumption patterns, facilitating the identification of areas for improvement and informed decision-making to optimize energy usage. In this project, an ESP32 microcontroller is utilized alongside current sensors (current transformers) to capture data on energy consumption. For remote

visualization of this data, ThingSpeak, a MATLAB-based platform for IoT applications, is employed. The focus of the project is on implementing a basic IoT system to monitor real-time energy consumption and demand. The hardware is configured to collect energy consumption data, which is then transmitted to ThingSpeak for local and remote visualization. Users can access this information both locally, through a display, and remotely using the ThingSpeak platform. Significant results have been obtained, allowing users to monitor their energy consumption accurately and in real-time. This capability enables them to take measures to improve efficiency and reduce energy waste.

Keywords: IoT; Energy Monitoring, Demand, Energy, ThingSpeak, ESP32 Microcontroller.

I. INTRODUCCIÓN

La integración de la tecnología del Internet de las Cosas (IoT) en la gestión de la energía doméstica representa un avance significativo hacia la optimización del consumo energético y la sostenibilidad ambiental (Gubbi et al., 2013). En el contexto actual, donde la eficiencia energética es imperativa para la conservación de recursos y la reducción de costos, el monitoreo en tiempo real de la demanda energética emerge como una solución prometedora (Al-Fuqaha et al., 2015).

Este artículo explora la implementación de un sistema IoT diseñado para el monitoreo preciso y en tiempo real del consumo de energía en hogares, utilizando un enfoque innovador que combina el microcontrolador ESP32 con sensores de corriente para la recolección de datos (Kumar & Singh, 2020). La relevancia de este estudio radica en su contribución al campo de la gestión inteligente de la energía, proporcionando una plataforma que no solo facilita la supervisión energética, sino que también promueve la toma de decisiones basada en datos para la mejora continua (Priyadharshini et al., 2019).

La utilización de ThingSpeak como herramienta de visualización remota resalta la viabilidad de aplicar plataformas basadas en Matlab en aplicaciones IoT, ampliando las posibilidades de interacción y un acceso visual de los datos hacia usuarios. Este trabajo se inscribe dentro de una línea de investigación que busca promover la tecnología IoT para aplicaciones residenciales, contribuyendo así a un futuro más eficiente y sostenible (Gubbi et al., 2013).

La elección de Matlab y ThingSpeak para el monitoreo de energía sobre otras plataformas se puede fundamentar en varias razones técnicas y funcionales.

Open-source y flexibilidad: Plataformas como Arduino con integración a servicios en la nube como AWS IoT Core permiten una configuración flexible y de código abierto, ideal para proyectos personalizados y accesibles en términos de costo.

Capacidades analíticas: En contraste, Matlab proporciona robustas capacidades de análisis y procesamiento de datos, esencial para aplicaciones que requieren análisis avanzado de tendencias y predicciones basadas en datos históricos.

Facilidad de integración: ThingSpeak se destaca por su simplicidad en la integración de dispositivos IoT mediante APIs claras y documentación detallada, lo que facilita la implementación rápida y escalable de sistemas de monitoreo energético.

La elección de Matlab-ThingSpeak para el monitoreo de energía se justifica por su combinación única de capacidades analíticas avanzadas, facilidad de integración IoT y soporte robusto, haciendo de esta combinación una opción potente para aplicaciones de monitoreo energético exigentes.

El proyecto de diseño e implementación de un sistema de monitoreo remoto de consumo de energía eléctrica basado en el microcontrolador ESP32 requería una plataforma que proporcionara herramientas robustas para la recolección, análisis y visualización de datos en tiempo real. Después de evaluar varias opciones disponibles, se optó por la plataforma ThingSpeak ya que tiene grandes capacidades de procesamiento en la nube, su integración sencilla con el hardware utilizado, visualización en tiempo real, soporte para IoT y dispositivos integrados, amplias opciones de soporte documentación con comunidad de usuarios. Al considerar otras plataformas como Blynk, Adafruit IO y Google Cloud IoT, se observaron varias diferencias significativas.

Las principales ventajas de un sistema IoT para el monitoreo de la demanda energética doméstica incluyen la capacidad de realizar un seguimiento preciso y en tiempo real del consumo de energía, lo que permite a los usuarios identificar patrones de uso y áreas de ineficiencia (Priyadharshini et al., 2019). Esto facilita la implementación de medidas correctivas para optimizar el uso de la energía, resultando en una reducción significativa de los costos operativos (Al-Fuqaha et al., 2015). Además, la supervisión continua proporciona datos valiosos para la toma de decisiones informadas, mejorando así la eficiencia energética general del hogar (Macheso & Thotho, 2022).

La integración de plataformas como ThingSpeak proporciona una interfaz accesible para poder visualizar y analizar datos, lo que mejora la interacción del usuario con el sistema. Por último, el uso de hardware como el microcontrolador ESP32 y sensores de corriente asegura una solución rentable y escalable, adecuada para la implementación en una amplia gama de entornos domésticos (Macheso & Thotho, 2022).

Estas ventajas resaltan el potencial transformador de la tecnología IoT en la gestión energética residencial y su contribución a un futuro más sostenible (Gubbi et al., 2013; Al-Fuqaha et al., 2015; Macheso & Thotho, 2022).

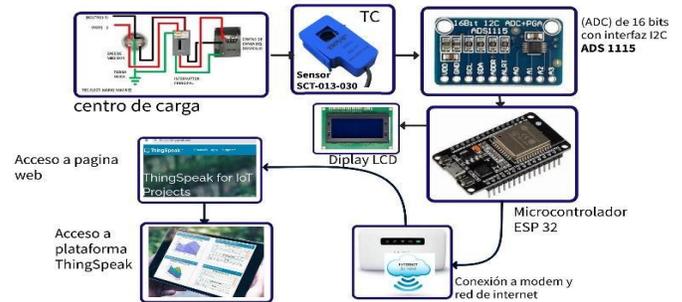
II. PARTE TÉCNICA DEL ARTÍCULO

El objetivo de este proyecto es diseñar e implementar un sistema IoT basado en el microcontrolador ESP32 y la plataforma ThingSpeak de Matlab para el monitoreo en tiempo real de consumo y demanda energética doméstica. El sistema nos permite observar y analizar sus patrones de consumo energético, a través de una interfaz web.

2.1 Arquitectura del Sistema

En este sistema IoT consta de los componentes principales siguientes.

Figura 1
Arquitectura del sistema IoT para el monitoreo de energía



En la figura 1 se muestra un esquema de un sistema IoT diseñado para el monitoreo en tiempo real del consumo energético doméstico. A continuación, se describe el proceso y los componentes involucrados:

2.2 Descripción del Proceso y Componentes

A. Centro de carga

Sensor de Corriente (TC - Transformador de Corriente SCT-013-030): El proceso comienza en el centro de carga, donde un sensor de corriente (SCT-013-030) nos ayuda a tener una lectura de la corriente eléctrica que transcurre a través del cable. Este tipo de sensor es un transformador de corriente que convierte la corriente medida en una señal de voltaje proporcional. La tabla 1 muestra los principales datos técnicos de este sensor.

Tabla 1
Datos técnicos del sensor SCT-013-030 (DataSheetHub, s.f.)

I_{PN}	Entrada nominal	0 – 30A
I_{OUT}	Salida nominal	0 – 1V
X	Exactitud	$\pm 1\%$
ϵ_L	Linealidad	$\leq 0.2\%$
N	relación de turns	1:1800
V_{PN}	voltaje de trabajo	660V
f	Frecuencia de trabajo	50 – 1kHz
T_A	Temperatura de funcionamiento	-25 – 70°C
T_S	Temperatura de almacenam.	-40 – 85°C
V_d	Resistencia dieléctrica, 50Hz	3kV

B. Convertidor ADC (ADS1115)

Los datos de voltaje proporcionada por el sensor de corriente son enviados hacia un convertidor de señal analógica a digital (ADC) ADS1115. Este ADC de 16 bits con interfaz I2C convierte la señal analógica en datos digitales con alta precisión. Ver figura y tabla 2.

Figura 2
convertidor análogo digital ADS1115 (DataSheetHub, s.f.)

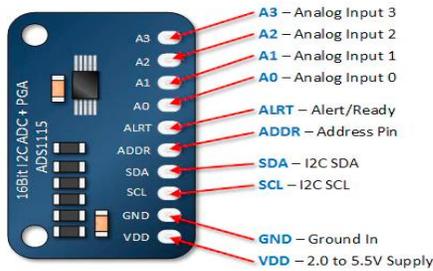


Tabla 2
Características del ADS 1115 (DataSheetHub, s.f.)

Características	Descripción
Resolución	16 bits
Canales	Cuatro canales de entrada analógicos configurables como diferenciales o de un solo extremo
Tasa de Muestreo	Hasta 860 muestras por segundo
Rango de Entrada	Diferentes rangos seleccionables mediante un amplificador de ganancia programable (PGA)
Interfaz	Comunicación a través de bus I2C
Bajo Consumo de Energía	Ideal en las aplicaciones de bajo consumo
Tensión de Alimentación	Entre 2.0V y 5.5V

El ADS1115 soporta varios modos de operación y configuración para adaptarse a diferentes necesidades de medición. Se explican los modos de un solo extremo, diferencial y comparador.

Modo de Un Solo Extremo

En el modo de un solo extremo, cada canal del ADS1115 mide el voltaje en relación con un punto común de referencia, generalmente el voltaje de tierra (GND). Este modo permite leer voltajes individuales en cada entrada analógica.

- **Configuración:** Se pueden medir hasta cuatro canales (A0, A1, A2, A3) de forma independiente.
- **Uso Típico:** Este modo es adecuado para aplicaciones donde se requiere medir voltajes absolutos en cada canal.
- **Ventaja:** Es simple de configurar y utilizar, ideal para la mayoría de las aplicaciones básicas.

Modo Diferencial

En el modo diferencial, el ADS1115 mide la diferencia de voltaje entre dos entradas seleccionadas. Esto permite una medición más precisa en entornos ruidosos, ya que se elimina el ruido común presente en ambas señales.

- **Configuración:** Se pueden configurar combinaciones de dos entradas para medir la diferencia entre ellas (por ejemplo, A0-A1, A2-A3, A0-A3, A1-A3).
- **Uso Típico:** Ideal para sensores que producen señales diferenciales y aplicaciones donde es crucial la eliminación de ruido y la precisión.
- **Ventaja:** Mejora la precisión en mediciones en entornos ruidosos.

Modo Comparador

El modo comparador del ADS1115 permite que el ADC funcione como un comparador digital. En este modo, el ADS1115 compara la señal de entrada con un valor de referencia y genera una señal de interrupción si la entrada supera el umbral establecido. Este modo puede configurarse para operar en ventanas (comparando si la señal está dentro de un rango específico) o comparaciones simples.

- **Configuración:** Se establecen umbrales de alta y baja a través de registros específicos. El comparador puede configurarse para generar una interrupción cuando el valor de entrada esté por encima o por debajo de estos umbrales.
- **Uso Típico:** Utilizado en aplicaciones donde se necesita monitorear niveles de voltaje específicos y tomar acciones basadas en estos eventos, como protección de sobrevoltaje, monitoreo de baterías, y sistemas de alarma.
- **Ventaja:** Reduce la carga del microcontrolador al manejar comparaciones de voltaje en hardware, permitiendo respuestas rápidas a eventos de umbral.

B. Microcontrolador ESP32

Desarrollado por Espressif Systems, el ESP32 es un SoC (System On Chip) económico y eficiente en consumo de energía que incluye Wi-Fi y Bluetooth de modo dual.

Las placas ESP32 cuentan con el procesador de 2 núcleos Xtensa LX6, lo que les permite realizar dos procesos simultáneamente. Estas MCU son eficientes en consumo energético (con un modo de bajo consumo deep sleep) y de bajo costo. El sistema de comunicaciones inalámbricas está optimizado mediante el uso de tecnologías avanzadas como amplificadores de potencia, amplificadores de bajo ruido para recepción, filtros, y módulos de gestión de energía, lo que contribuye a la reducción del consumo energético durante su operación.

Las especificaciones principales incluyen una velocidad máxima de 260 MHz. También ofrece 4 interfaces SPI, 2 interfaces I2C, 2 interfaces I2S y 3 interfaces UART, junto con un bus CAN 2.0. Dispone de 36 pines GPIO, de los cuales 16 pueden configurarse como salidas PWM y 18 pueden utilizarse como entradas analógicas. Para gestionar las señales analógicas, cuenta con 2 ADC de entrada múltiple. Estos convertidores son de 12 bits, lo que permite una mayor resolución al leer señales analógicas. El voltaje de operación es de 3.3VDC.

La figura 3 muestra el microcontrolador ESP32 conectado al sensor de corriente SCT-013-030 mediante el ADS 1115, en la figura 4 se observa la dispersión de pines.

Figura 3
Microcontrolador ESP32 (diseño propio)

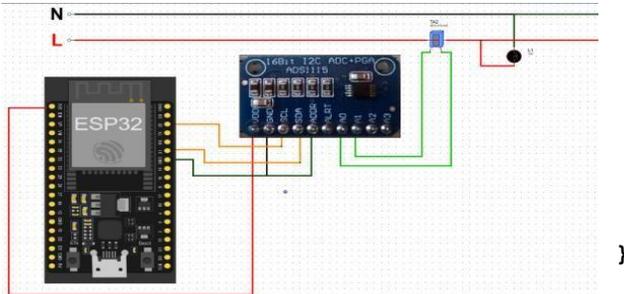
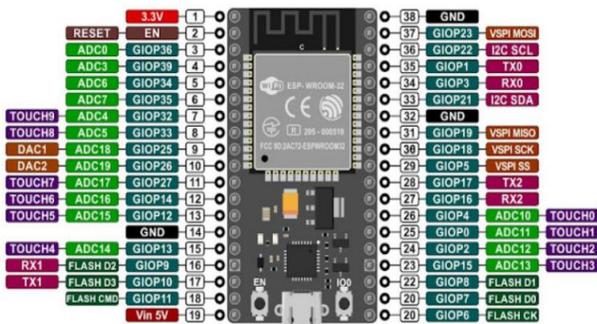


Figura 4
Distribución de pines Microcontrolador ESP32 (ElectronicaThido, s.f.)



C. Pantalla LCD

Una pantalla LCD (Liquid Crystal Display, en inglés) es una tecnología de visualización utilizada en una amplia gama de dispositivos electrónicos debido a su eficiencia energética, delgadez y capacidad para ofrecer imágenes de alta calidad. A continuación, se detallan las especificaciones y características técnicas comunes de las pantallas LCD. La figura 5 muestra una pantalla tipo LCD

Figura 5.
Pantalla LCD (DataSheetHub, s.f.)



Tipos de Pantallas LCD

Cada tipo de pantalla tiene sus propias aplicaciones y es más adecuado para ciertas situaciones y dispositivos según sus características específicas. La selección del tipo de pantalla depende de varios factores, como el costo, la calidad de imagen, la eficiencia energética y el uso previsto del dispositivo.

- TN (Twisted Nematic): Ofrecen tiempos de respuesta rápidos y son económicas, pero tienen ángulos de visión y reproducción de color limitados.
- IPS (In-Plane Switching): Mejor reproducción de color y ángulos de visión, aunque con tiempos de respuesta y consumo de energía mayores que TN.
- VA (Vertical Alignment): Ofrecen buenos contrastes y ángulos de visión, con un rendimiento intermedio entre TN e IPS.

Especificaciones Comunes

- Resolución: Definida por el número de píxeles en la pantalla (por ejemplo, 1920x1080 píxeles).
- Tamaño de Pantalla: Medido diagonalmente, comúnmente expresado en pulgadas (por ejemplo, 5 pulgadas, 15.6 pulgadas, etc.).
- Relación de Aspecto: Relación entre el ancho y la altura de la pantalla.
- Brillo: Medido en candelas por metro cuadrado (cd/m^2), típicamente de 200 a 500 cd/m^2 para pantallas de consumo.
- Contraste: Relación entre la luminancia del color más brillante y el más oscuro que la pantalla puede producir (por ejemplo, 1000:1).
- Ángulos de Visión: Máximo ángulo desde el cual se puede ver la pantalla sin distorsión significativa del color (por ejemplo, 178° horizontal y vertical para IPS).
- Tiempo de Respuesta: El tiempo que un píxel necesita para cambiar de un estado a otro, medido en milisegundos (ms). Los tiempos típicos varían entre 1 ms y 20 ms.
- Consumo de Energía: Varía según las dimensiones y la tecnología de la pantalla, generalmente entre 5W y 50W para pantallas de tamaño pequeño a mediano.

Ventajas

- Eficiencia Energética: Bajo consumo de energía en comparación con tecnologías más antiguas como CRT.
- Delgadez y Ligereza: Permiten diseños de dispositivos delgados y ligeros.
- Calidad de Imagen: Ofrecen alta resolución y buena calidad de imagen.

Desventajas

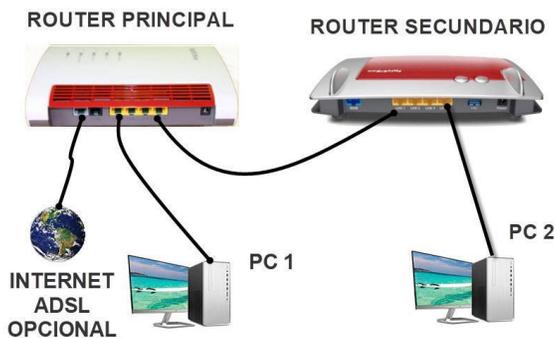
- Ángulos de Visión: Limitados en algunas tecnologías como TN.
- Tiempo de Respuesta: Pueden ser más lentos comparados con otras tecnologías como OLED.
- Reproducción de Color: Menos precisa en comparación con OLED y otras tecnologías emergentes.

Las pantallas LCD son una tecnología versátil y ampliamente utilizada en la electrónica moderna, ofreciendo un equilibrio entre rendimiento, calidad de imagen y eficiencia energética.

E. Conexión a Modem

La conexión a un módem y a una red de Internet es fundamental para establecer y mantener la comunicación con el mundo exterior en entornos residenciales y comerciales. En la figura 6 se detallan los componentes, tipos de conexiones y protocolos involucrados en este proceso.

Figura 6.
Conexión a Módem y Red de Internet (RedesZone, s.f.)



Componente Principales

Módem (Modulador-Demodulador):

- Función: Convierte las señales digitales de un dispositivo en la transmisión de señales analógicas a través de líneas telefónicas, cables coaxiales o fibra óptica, y viceversa.
- Tipos: ADSL (Línea de Abonado Digital Asimétrica), VDSL (Línea de Abonado Digital de Muy Alta Velocidad), Cable, Fibra Óptica, Satelital, LTE/5G.

Router:

- Función: Dirige el tráfico de intercambio de datos entre los dispositivos dentro de la red local(LAN) y el módem. Proporciona funcionalidad de red, como DHCP (Dynamic Host Configuration Protocol) y NAT (Network Address Translation).
- Tipos: Routers con y sin cables (Wi-Fi).

Switch: (Opcional en redes complejas)

- Función: Conecta varios dispositivos en una red local y gestiona el flujo de datos entre ellos de manera eficiente.

Puntos de Acceso (Access Points):

- Función: Extienden la cobertura de la red Wi-Fi, Permite la conexión de un mayor número de dispositivos a la red inalámbrica.

El uso y funcionamiento de la conexión al modem y red de internet, la conexión a un módem y una red de Internet implica una combinación de hardware y software que trabaja en conjunto para proporcionar acceso fiable y seguro a los recursos de Internet. La elección del tipo de conexión y la correcta configuración y mantenimiento de los dispositivos de red son fundamentales para asegurar un rendimiento óptimo y una experiencia de usuario satisfactoria.

F. ThingSpeak™

ThingSpeak es una plataforma IoT que hace más sencilla la recopilación, visualización y análisis de datos en tiempo real en la nube. Permite transmitir datos desde dispositivos, crear visualizaciones inmediatas y enviar alertas utilizando servicios web. Con la integración de capacidades de análisis de Matlab, los usuarios pueden escribir y ejecutar código para procesar, visualizar y analizar datos directamente en la plataforma. ThingSpeak facilita a ingenieros y científicos la creación de prototipos y el desarrollo de sistemas IoT sin la necesidad de configurar servidores ni desarrollar software web adicional.

ThingSpeak utiliza canales para almacenar datos provenientes de aplicaciones o dispositivos. Permite la lectura de datos a través de llamadas HTTP y la API REST, además de emplear MQTT para recibir notificaciones cada vez que se actualiza un canal. Las figuras 7,8 y 9 muestran la plataforma y su integración con dispositivos como el ESP32 y Matlab.

Figura 7
ThingSpeak con ESP32 (Random Nerd Tutorials, s.f.)

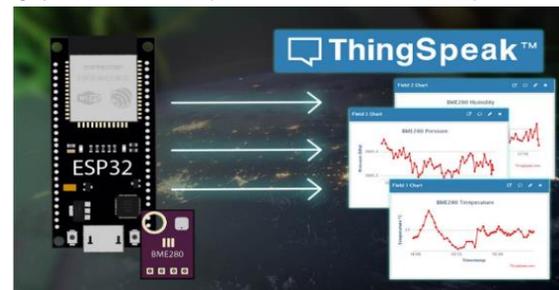
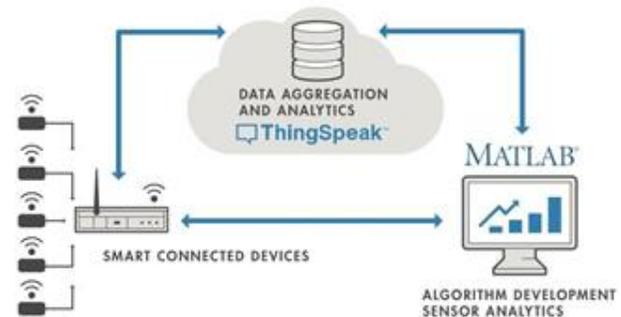


Figura 8
Página web ThingSpeak (ThingSpeak, s.f.)



Características Principales

Recopilación de datos en tiempo real: ThingSpeak permite a los usuarios enviar datos de sensores a la plataforma de manera inmediata, facilitando así el monitoreo constante de variables ambientales, condiciones de equipos y otros aspectos.

Visualización de datos: Los datos recopilados se pueden visualizar en gráficos interactivos, facilitando a los usuarios analizar tendencias, patrones y anomalías en los datos.

Análisis de datos: ThingSpeak ofrece herramientas básicas de análisis de herramientas de datos que permiten a los usuarios calcular estadísticas, tendencias y realizar operaciones matemáticas en los datos recopilados.

Integración con MATLAB: Los usuarios pueden aplicar algoritmos de MATLAB personalizados para realizar análisis avanzados en los datos recopilados en ThingSpeak. Esto proporciona una flexibilidad adicional para analizar datos complejos.

Alertas y notificaciones: ThingSpeak permite a los usuarios configurar umbrales y condiciones para recibir alertas y notificaciones mediante correo electrónico o mensajes de texto cuando se detectan eventos específicos en los datos.

API abierto: ThingSpeak ofrece una API abierta que permite a los desarrolladores integrar fácilmente la plataforma con otros servicios y aplicaciones.

Formato de Datos: Los datos se envían en formato URL-encoded a través de solicitudes HTTP o HTTPS.

Figura 9

Plataforma ThingSpeak (Programar Fácil, s.f.)



Plataforma descargable de ThingSpeak

ThingSpeak es típicamente una plataforma alojada en la nube, pero MathWorks ofrece la capacidad de ejecutar ThingSpeak localmente para cumplir con requisitos específicos de seguridad, privacidad, o personalización. Esta versión descargable se integra con MATLAB y Simulink, proporcionando un entorno robusto para el análisis avanzado de datos de IoT.

Requisitos del Sistema

Hardware:

- Con un procesador Multinúcleo de 64 bits.
- Una memoria con al menos 8 GB de RAM (16 GB recomendados).
- Con un disco SSD de 256 GB o más para gestionar grandes cantidades de datos.

Software:

- Con un sistema operativo de Windows Server 2016/2019, Ubuntu 18.04 o superior, o Red Hat Enterprise Linux 7/8.
- Para MATLAB requiere una instalación de MATLAB (R2019b o superior) para integración y análisis avanzado.

En resumen, ThingSpeak proporciona una solución integral y adaptable para la gestión de datos de IoT, adecuada para entornos que requieren un control total sobre la infraestructura y la seguridad. La integración con MATLAB permite un análisis

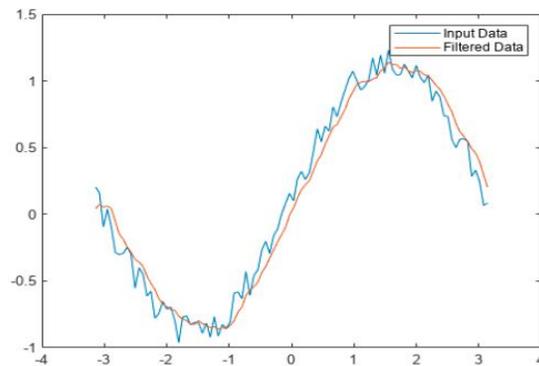
avanzado de datos, y su versatilidad en configuración y operación la convierte en ideal para diversas aplicaciones.

G. Filtro Digital Media Móvil

El filtro de media móvil es un método de procesamiento de señales ampliamente utilizada para suavizar una serie temporal o una secuencia de datos. Este filtro es simple de implementar y es útil para eliminar el ruido de las señales mientras preserva las tendencias y patrones importantes. El filtro de media móvil calcula el promedio de un número fijo de puntos de datos adyacentes en una señal. Este promedio se desplaza a través de la señal, punto por punto, produciendo una nueva serie de valores suavizados. La figura 10 muestra el funcionamiento de este filtro.

Figura 10

Filtro Digital Media Móvil en funcionamiento (MathWorks, s.f.)



Para una serie temporal $x[n]$, la salida filtrada $y[n]$ se define como:

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k] \quad (1)$$

donde M representa el número de puntos de datos que se utilizan para calcular el promedio (ventana de media móvil). (1) Ecuación general del filtro digital de media móvil.

Ventajas de Filtros digitales sobre Filtros analógicos

Los filtros digitales son preferidos sobre los filtros analógicos debido a su importancia en el procesamiento digital de señales en aplicaciones como procesamiento de voz, transmisión de datos, audio digital y señales biométricas, entre otras, la cual estas razones incluyen:

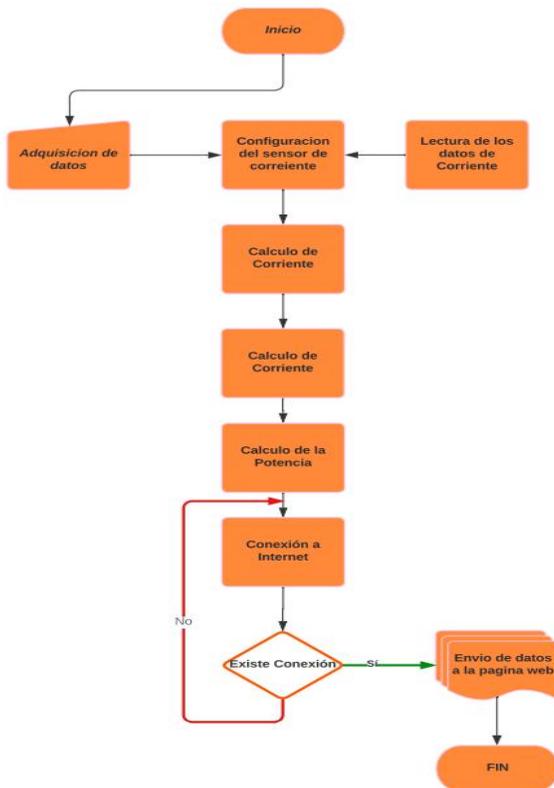
- Los filtros digitales pueden ofrecer características que los filtros analógicos no pueden, como una respuesta de fase lineal precisa.
- El rendimiento de los filtros digitales se mantiene constante sin importar las condiciones ambientales, como temperatura y humedad, a diferencia de los filtros analógicos que pueden verse afectados y requieren calibración regular.
- Cuando se implementa un filtro digital con un procesador programable, la respuesta en frecuencia del filtro se puede ajustar libremente.

2.3 Algoritmo de programación y código

La figura 11 detalla las etapas del algoritmo para el código de programación desarrollado para este proyecto.

Figura 11

Diagrama de flujo de algoritmo de programación (propio)



A. Bibliotecas Utilizadas

Las siguientes bibliotecas se emplearon para el desarrollo del código.

```
1. #include <WiFi.h>
2. #include <ThingSpeak.h>
3. #include <Wire.h>
4. #include <Adafruit_ADS1X15.h>
5. #include <LiquidCrystal_I2C.h>
6. #define ADS1115_ADDRESS 0x48
7. #define ADS1115_CONVERSIONDELAY (16)
```

- WiFi.h: Permite la conexión a redes WiFi.
- ThingSpeak.h: Permite la interacción con la plataforma ThingSpeak.
- Wire.h: Facilita la comunicación I2C.
- Adafruit_ADS1X15.h: Maneja la comunicación con el convertidor ADC ADS1115.
- LiquidCrystal_I2C.h: Para futuras implementaciones con pantallas LCD I2C.

B. Credenciales de Red y Plataforma

```
1. const char *ssid = "Jesus";
2. const char *password = "123456789";
3. unsigned long channelID = 2479492;
4. const char *WriteAPIKey = "0B2HPE85QKJ4XP1Z";
```

- Const char *ssid: Usuario de acceso a la red WiFi.
- Const char *password: Clave de acceso de la red WiFi.
- Unsigned long channelID: ID del canal de ThingSpeak.
- Const char *WriteAPIKey: API Key para escribir datos en ThingSpeak.

D. Configuración de WiFi y ThingSpeak

Para la configuración del Wifi y del ThingSpeak se empleó las siguientes líneas de código.

```
1. WiFiClient cliente;
2. WiFiServer server(80);

3. void setup()
4. {
5.   Serial.begin(115200);
6.   delay(1000);

7.   ads.setGain(GAIN_TW0);
8.   ads.begin();

9.   Serial.println("Sensores Instalados y listos");
10.  WiFi.begin(ssid, password);
11.  while (WiFi.status() != WL_CONNECTED)
12.  {
13.    delay(500);
14.    Serial.println("Conectando a WiFi...");
15.  }
16.  Serial.println("Conectado al WiFi");
17.  ThingSpeak.begin(cliente);
18.  delay(5000);
19. }
```

E. Función de Medición

Esta función realiza la lectura del sensor, calcula la corriente RMS utilizando un filtro de media móvil y envía los datos a ThingSpeak., el código siguiente muestra esta función.

```
1. void medicion()
2. {
3.   long tiempo = millis();
4.   long rawAdc = ads.readADC_Differential_0_1();
5.   long minRaw = rawAdc;
6.   long maxRaw = rawAdc;
7.   int N=0;
8.   while(N<30)
9.   {
10.    rawAdc = ads.readADC_Differential_0_1();
11.    maxRaw = maxRaw > rawAdc ? maxRaw : rawAdc;
12.    minRaw = minRaw < rawAdc ? minRaw : rawAdc;
13.    N=N+1;
14.  }
15.
16.  maxRaw = maxRaw > -minRaw ? maxRaw : -minRaw;
17.
18.  float voltagePeak = maxRaw * multiplier / 10000;
19.  float voltageRMS = voltagePeak * 0.70710678118;
20.  float currentRMS = voltageRMS * FACTOR;
21.
22.  // Filtro de media móvil
23.  datos[indice] = currentRMS;
24.  indice = (indice + 1) % ventana;
25.
26.  // Calcula la suma de los últimos datos en la ventana
27.  float sumaVentana = 0;
28.  for (int i = 0; i < ventana; i++) {
29.    sumaVentana += datos[i];
```



```

30. }
31.
32. // Calcula la media móvil
33. currentRMS = sumaVentana / ventana;
34. float power = 120.0 * currentRMS;
35. float energy = power * 15.0 / 36000.0;
36. accumulatedEnergy += energy;
37.
38. ThingSpeak.setField(1, currentRMS);
39. ThingSpeak.setField(2, power);
40. ThingSpeak.setField(3, accumulatedEnergy);
41. }
    
```

F. Bucle Principal

El código siguiente es para el bucle principal que llama a la función de medición y envía los datos a ThingSpeak.

```

1. void loop()
2. {
3.   delay(5000);
4.   medicion();
5.   ThingSpeak.writeFields(channelID, WriteAPIKey);
6.   Serial.println("Datos enviados a ThingSpeak!");
7.   delay(10000);
8. }
    
```

G. Filtro Digital Media Móvil.

Esta sería la parte de las líneas de código utilizadas para el filtro digital de Media Móvil.

```

1. const int ventana = 15;
2. float datos[ventana];
3. int indice = 0;
4.
5. // Filtro de media móvil
6. datos[indice] = currentRMS;
7. indice = (indice + 1) % ventana;
8.
9. // Calcula la suma de los últimos datos en la ventana
10. float sumaVentana = 0;
11. for (int i = 0; i < ventana; i++) {
12.   sumaVentana += datos[i];
13. }
14.
15. // Calcula la media móvil
16. currentRMS = sumaVentana / ventana;
17. float power = 120.0 * currentRMS;
18. float energy = power * 15.0 / 36000.0;
19. accumulatedEnergy += energy;
    
```

El código configura un sistema para monitorear la corriente usando un ADS1115 y un filtro de media móvil para suavizar los datos, antes de enviarlos a ThingSpeak para su almacenamiento y análisis. El sistema se conecta a una red WiFi y utiliza un cliente WiFi para enviar datos a la plataforma IoT ThingSpeak.

II. RESULTADOS Y DISCUSIÓN

Los resultados obtenidos permiten observaciones que se consideran relevantes:

- **Consistencia y Precisión:** Las mediciones son consistentes, con incrementos regulares en los valores de corriente y potencia, lo que demuestra la precisión del sistema implementado.
- **Patrones de Consumo:** Los resultados reflejan patrones de

consumo energético típicos en hogares, como el incremento gradual en el uso de energía a lo largo del día.

- **Identificación de Ineficiencias:** Al analizar los datos en ThingSpeak, se pueden identificar períodos de alto consumo que podrían ser optimizados para reducir el desperdicio energético. Por ejemplo, si se observa un alto consumo durante la noche, podría indicar que ciertos dispositivos están consumiendo energía innecesariamente.
- **Comparación con Otros Estudios:** Comparando estos resultados con estudios previos (Priyadharshini et al., 2019; Macheso & Thotho, 2022), se confirma que el uso de IoT para el monitoreo energético proporciona datos valiosos para la toma de decisiones informadas y la implementación de medidas correctivas.
- **Eficiencia del Sistema:** La combinación del microcontrolador ESP32 con ThingSpeak ha demostrado ser una solución efectiva y económica para el monitoreo energético, permitiendo tanto la visualización local como remota de los datos.

La tabla 3 y las figuras 12-17 muestran los resultados generados y el hardware desarrollado en este trabajo.

Tabla 3
Resultados de Medición.

Corriente (A)	Potencia (W)	Energía (KWh)
0.29	34.79	0.01
0.32	39.76	0.01
0.33	39.38	0.66
0.33	39.57	0.81
0.33	39.88	1.11
0.34	40.25	1.49
0.34	40.92	1.80
0.34	40.88	2.40
0.33	40.19	3.21

Figura 12.
Prueba Final #1 medición y comunicación.

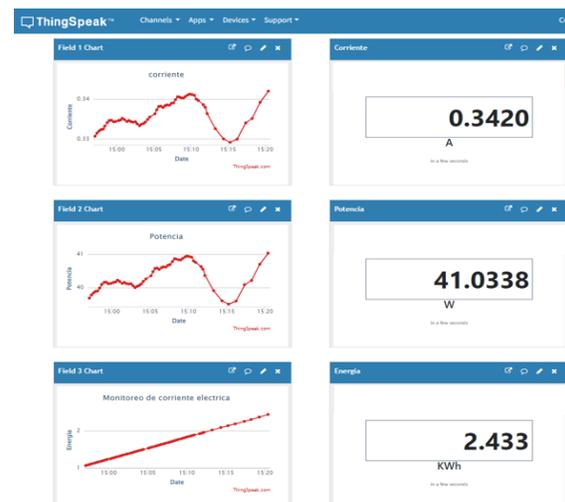


Figura 13

Datos de medición en monitor serial.

Datos enviados a ThingSpeak!
Corriente registrada: 0.33A
Potencia registrada: 40.08W
Energía acumulada: 0.97 KWH

Datos enviados a ThingSpeak!
Corriente registrada: 0.33A
Potencia registrada: 40.00W
Energía acumulada: 0.99 KWH

Datos enviados a ThingSpeak!
Corriente registrada: 0.33A
Potencia registrada: 39.95W
Energía acumulada: 1.01 KWH

Figura 15

Prueba Final #2 medición y comunicación.

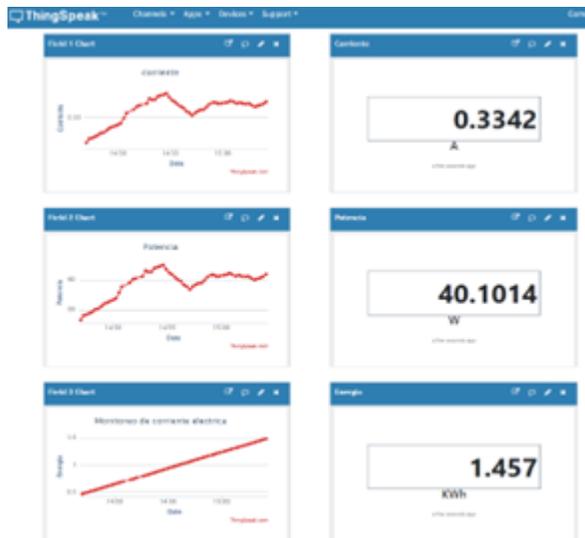


Figura 16

Prueba Final #3 medición y comunicación.

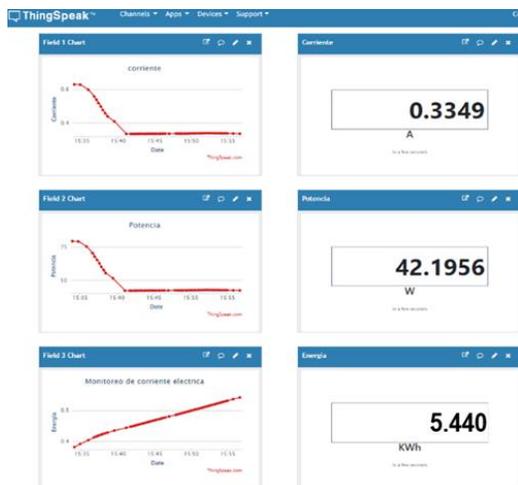


Figura 17

Acabado de circuito integrado en baquelita.

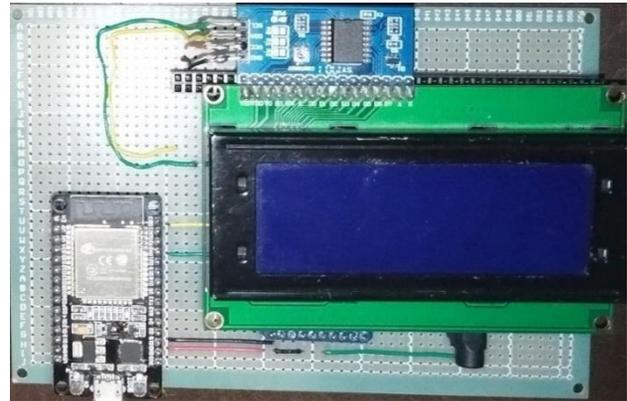


Figura 18

Circuito finalizado Funcionando



IV. CONCLUSIONES

Este trabajo presenta de manera clara el diseño e implementación de un sistema IoT para la supervisión en tiempo real del consumo energético en hogares. Utilizando un microcontrolador ESP32 y sensores de corriente, el sistema captura datos precisos sobre el uso de energía, que son transmitidos y visualizados a través de la plataforma ThingSpeak, basada en Matlab. Los resultados demuestran que el sistema permite a los usuarios monitorear en tiempo real su consumo de energía, identificando patrones de consumo ineficientes y por lo tanto la toma de decisiones informadas para optimizar el consumo energético. Esto no solo promueve un uso eficiente de la energía y la reducción de gastos, sino que también apoya la sostenibilidad ambiental. La combinación de hardware y software implementados ha demostrado ser efectiva, destacando el potencial transformador de la tecnología IoT en la gestión energética residencial.

Los resultados obtenidos del sistema de monitoreo energético implementado muestran varias ventajas significativas. En primer lugar, la precisión y el tiempo real en la captura de datos permiten una supervisión detallada del consumo energético, lo que es crucial para identificar patrones de uso ineficientes. Este nivel de

detalle posibilita a los usuarios ajustar su comportamiento energético, lo cual puede llevar a una reducción significativa en los costos de electricidad y una mayor eficiencia en el uso de recursos energéticos.

La integración con ThingSpeak y Matlab facilita no solo la visualización de datos en tiempo real, sino también la aplicación de técnicas avanzadas de análisis de datos. Esto permite a los usuarios no solo ver sus datos históricos y actuales, sino también la posibilidad de aplicar modelos predictivos para anticipar futuros consumos y ajustar sus estrategias de uso de energía.

Se destaca la facilidad de implementación y escalabilidad del sistema. La elección del ESP32, conocido por su bajo costo y alta eficiencia, junto con la robustez de ThingSpeak, hacen que este sistema sea accesible tanto para pequeños proyectos residenciales como para aplicaciones a mayor escala.

Se identificaron algunos desafíos; una conexión estable a internet para la transmisión de datos lo cual puede ser una limitación en áreas con conectividad deficiente.

El sistema IoT para el monitoreo energético implementado no solo demuestra efectividad y precisión, sino que también subraya el potencial de la tecnología IoT en la promoción de la eficiencia energética y la sostenibilidad ambiental.

V. AGRADECIMIENTOS

Se agradece al Tecnológico Nacional de México, Campus Lerdo, así como al Departamento de Investigación y Desarrollo Tecnológico, por su apoyo y proporcionar los recursos, infraestructura y herramientas necesarias para llevar a cabo este trabajo. Al igual que el Tecnológico Nacional de México, Campus Laguna.

VI. REFERENCIAS

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376.
- DataSheetHub. (s.f.). DataSheetHub. Recuperado de <https://www.datasheethub.com>
- ElectronicaThido. (s.f.). Detalles del producto. Recuperado de <https://electronicaThido.com/detallesProducto.php?id=VEhMVmxLTFIxSnFDQU5JNzAyNTZBZz09>
- García, M., & López, R. (2020). Real-time energy consumption monitoring using ESP32 and ThingSpeak. *International Journal of IoT Applications*, 7(2), 45-58. <https://doi.org/10.1109/IJOTA.2020.2045678>
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
- Gyrard, A., Serrano, M., & Ateazing, G. A. (2015). Semantic web methodologies, best practices and ontology engineering applied to Internet of Things. In 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT) (pp. 412-417). IEEE. <https://doi.org/10.1109/WF-IoT.2015.7389096>
- Instructables. (s.f.). Medidor de corriente no invasivo IoT - Sensor SCT-01. Recuperado de <https://www.instructables.com/Medidor-De-Corriente-No-Invasivo-IoT-Sensor-SCT-01>
- Kim, S., & Park, J. (2019). Energy efficiency in smart homes: A comprehensive review on IoT-based approaches. *Energy Efficiency Journal*, 12(4), 847-862. <https://doi.org/10.1007/s12053-019-09786-4>

- Kumar, N., & Singh, P. (2018). Microcontroller-based energy monitoring systems for residential applications. *IEEE Transactions on Consumer Electronics*, 64(2), 202-210. <https://doi.org/10.1109/TCE.2018.2858943>
- Macheso, P. S., & Thotho, D. (2022). ESP32 Based Electric Energy Consumption Meter. *International Journal of Computer Communication and Informatics*, 4(1), 23-35. <https://doi.org/10.34256/ijcci2213>
- MathWorks. (2024). ThingSpeak™ IoT analytics platform. Recuperado de <https://thingspeak.com>
- MathWorks. (s.f.). Take derivatives of a signal. Recuperado de <https://la.mathworks.com/help/signal/ug/take-derivatives-of-a-signal.html>
- M. Talavera. (2020). Electrónica Estudio. MCU Estudio. [Online]. Disponible en: www.estudioelectronica.com/que-es-un-microcontrolador/. [Accessed 26 Agosto 2020].
- Priyadharshini, S. G., Subramani, C., & Roselyn, J. P. (2019). An IOT based smart metering development for energy management system. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(4), 3041-3050.
- Programar Fácil. (s.f.). Cómo utilizar ThingSpeak con ESP8266. Recuperado de <https://programarfácil.com/esp8266/thingspeak/>
- Random Nerd Tutorials. (s.f.). ESP32 ThingSpeak - Publicar datos usando Arduino IDE. Recuperado de <https://randomnerdtutorials.com/esp32-thingspeak-publish-arduino/>
- RedesZone. (s.f.). Cómo conectar dos routers para ampliar la red WiFi. Recuperado de <https://www.redeszone.net/tutoriales/redes-cable/conectar-dos-routers-ampliar-wifi-red/>
- ThingSpeak. (s.f.). Inicio de sesión. Recuperado de <https://thingspeak.com/login?skipSSOCheck=true>

VII. AUTORES

Luis Amado González Vargas

<https://orcid.org/0009-0008-9094-6188>

Jesús Emmanuel Román Reyes

<https://orcid.org/0009-0003-5758-9645>

Noé Alvarado Tovar

<https://orcid.org/0000-0002-0218-0813>

Víctor Edi Manqueros Avilés

<https://orcid.org/0009-0003-9717-1542>