

Aulas Sustentables: Sistema de Automatización Basado en IoT para la Optimización de la Energía Eléctrica

Fassi, Lucas Jeremías

Moyano, José

Salazar Gisbert, Gabriel

Universidad Nacional del Sur, Departamento de Ciencias e Ingeniería de la Computación

Abstract

La optimización de la energía es un aspecto importante a tener en cuenta en todo tipo de sistema. Ayuda a reducir costos, aumentar la vida útil de equipamiento, y tiene un impacto favorable en el medio ambiente. En un entorno institucional, el objetivo fue reducir el consumo energético innecesario que se produce al mantener equipamiento activo en un ambiente que se encuentra desocupado. Por medio de un microcontrolador, diferentes sensores, y una interfaz web, es posible monitorear el entorno, interactuar remotamente y automatizar el comportamiento del equipamiento instalado.

Este documento describe el sistema desarrollado y los detalles de su implementación e instalación.

Palabras Clave

IoT, ESP32, Firebase, OTA.

Introducción

La motivación para realizar este trabajo se desprende naturalmente de dos perspectivas que se potencian, la necesidad de aplicar el conocimiento adquirido en la carrera, y la intención de generar un bien común. Este proyecto de *Internet of Things* (IoT) [9] se inspira en los esfuerzos realizados actualmente por la comunidad científica y tecnológica [8] para disminuir el uso de recursos no renovables utilizando IoT.

Las instituciones públicas nuclea diferentes grupos de personas que comparten el mismo espacio físico de estudio o trabajo. Esto da lugar a situaciones donde dicho espacio se encuentra desocupado, pero el equipamiento del mismo (luces, calefacción, etc) se mantiene activo innecesariamente.

En este trabajo se presenta la implementación de un sistema

automatizado que apaga los dispositivos en un aula sin uso.

En los **objetivos** se habla de que es lo que se quiere lograr, mencionando los objetivos principales así como también los secundarios.

En las **decisiones de diseño** se explican todas las elecciones tomadas al desarrollar el sistema.

Las secciones **procesador**, **plataforma de base de datos** y **dashboard web**, brindan una breve descripción de las tecnologías utilizadas y su razón.

La **arquitectura** explica cómo está implementado el sistema y cada uno de sus componentes con mayor detalle, **ESP32**, **firebase**, **dashboard web** y **Over-The-Air Programming** (OTA).

En las **capacidades del sistema** se indica resumidamente las características que el sistema tiene implementadas.

Ingreso al sistema

Para los que deseen observar el sistema en funcionamiento, pueden ingresar al sitio web de Aulas Sustentables [17].

Se puede ingresar con un usuario de prueba que solamente tiene permisos de lectura sobre la base de datos. Por lo tanto, sean libres de utilizar el sistema con tranquilidad. En la sección “Edificios”, el único aula que tiene un nodo asignado es el “Labo 4”.

Objetivos

El objetivo principal de este proyecto es optimizar la utilización de la energía eléctrica. Para esto, centramos la planificación, en principio, en las principales fuentes de consumo eléctrico comunes a un ambiente académico, la

iluminación, proyectores, y aire acondicionado. Si bien existen diferentes enfoques, soluciones y problemas vinculados a este objetivo, de nuestra solución se desprenden principalmente tres desafíos. El primero es determinar, con el mayor nivel de exactitud posible, cuándo un dispositivo está en funcionamiento. Esto implica el sensado de corriente, luminosidad, temperatura del ambiente, etc. Una vez que el sistema logra identificar qué dispositivo está activo, el segundo desafío es determinar si el mismo está siendo o no utilizado. Por ejemplo, si el sistema detecta que las luces están encendidas, debe determinar si el ambiente está siendo ocupado o se encuentra vacío. Por último, el desafío de mayor relevancia para este proyecto, es desactivar el dispositivo que no está siendo utilizado.

Este trabajo también tiene diferentes desafíos secundarios, que si bien no contribuyen directamente con el objetivo del proyecto, aportan facilidades. Uno de estos objetivos es que el sistema sea configurable, administrable y permita monitorear remotamente el estado de cada ambiente. Además que admita actualizaciones y correcciones de firmware sin la necesidad de intervenir físicamente. Finalmente, deseamos guardar los registros del comportamiento del sistema, como un historial, para verificar su correcto funcionamiento y realizar ajustes en caso de ser necesario.

Decisiones de diseño

Para cumplir con los objetivos, el sistema debe contar con las siguientes propiedades:

- Costo bajo en relación a los ahorros logrados en energía y vida útil de los equipos monitoreados. A pesar de los beneficios al medio ambiente, un atractivo económico facilita justificar la decisión de incorporar estas tecnologías.
- Bajo consumo de energía.
- Instalación simple: Una instalación sencilla reduce los costos, y reduce la necesidad de conocimientos técnicos del instalador.

- Bajo impacto en la infraestructura edilicia: Para no comprometer tanto la estética como la infraestructura del edificio.

- Conectividad inalámbrica a internet: Facilitando el monitoreo y configuración remota del sistema.

- Uso de hardware estándar y de alta disponibilidad: Permite que el diseño tenga un ciclo de vida prolongado, y sea de fácil adquisición y distribución.

- Suficiente potencia de procesamiento.

- Soporte: Compiladores libres, variedad de bibliotecas provistas por terceros, comunidad de usuarios, diversidad de *frameworks*, *Integrated Desktop Environments* (IDE) y estrategias de desarrollo.

- Pocas o inexistentes restricciones de licencias o usos para las tecnologías.

- Soluciones genéricas que cubran varios tipos de dispositivos, de cualquier marca y modelo, así como también diferentes dimensiones de las aulas.

- La inclusión del sistema en las aulas no debe modificar el uso habitual de las mismas. Sobre todo al ocurrir una falla. Si el sistema se encuentra o no en las aulas, debe ser transparente para las personas que normalmente utilizan las instalaciones.

Procesador

Teniendo en cuenta las propiedades del sistema, en particular la de un bajo impacto en la instalación edilicia, es que se buscó dispositivos con conectividad inalámbrica. Entre ellos, se optó por utilizar el ESP32 [1]. El cual, se trata de un *system on a chip* (SoC) con conectividad WiFi y Bluetooth. Cuenta con un bajo costo y consumo de energía, gran disponibilidad en el mercado y una inmensa comunidad.

Algunas de las características a destacar incluyen: procesador 32-bit de dos núcleos de hasta 240Mhz, co-procesador de ultra baja energía, algoritmos criptográficos por hardware y conversores analógico-digital de alta precisión.

Plataforma de base de datos

Se decidió utilizar Firebase [2] como base de datos por sus características, las cuales cubren perfectamente varios de los requerimientos del sistema. Se trata de una plataforma en la nube provista por Google que, entre varias de sus funciones, permite alojar y disponer de los datos e información en tiempo real y enviar automáticamente eventos a las aplicaciones cuando los datos cambian. Además ofrece un hosting estático, así como también un almacenamiento para archivos.

Dichas funcionalidades permiten enfocar los esfuerzos y el tiempo en otros desafíos del proyecto. Vale la pena mencionar que si bien es un servicio pago, Firebase ofrece una capa inicial de uso gratuita que abarca ampliamente los alcances del sistema. Con el extra de tener garantizada la disponibilidad del servicio.

Dashboard Web

Para la visualización de los datos elegimos un template basado en Angular, desarrollado por Akveo [3]. Es un panel de control gratuito y *Open Source*, tanto para propósitos personales como comerciales y de fácil personalización. Cuenta con varios componentes visuales ya implementados, disponibles para su uso y auto adaptables a la resolución del dispositivo que lo esté reproduciendo.

Este template nos permitió, en muy poco tiempo, disponer de una aplicación web *responsive* sin la necesidad de comenzar a desarrollarla desde cero.

Determinar si el espacio está siendo ocupado

Para determinar si el aula se encuentra en uso o no vamos a utilizar un sensor de movimiento tradicional como el de las alarmas. Si bien este dispositivo tiene un porcentaje de error se compensa con su bajo costo, facilidad de instalación e implementación.

Determinar si un dispositivo está en uso

Determinamos la actividad de un dispositivo a través de medir la intensidad de la corriente eléctrica, utilizando un sensor de bajo costo, que internamente funciona con un sensor de efecto Hall, que detecta el campo magnético generado por la corriente que circula por la línea a medir.

Modificar el estado de un dispositivo

Esta es una de las etapas más difíciles. Accionar sobre los dispositivos para apagarlos y encenderlos no es una tarea sencilla. Cada equipamiento es único y tiene su propio mecanismo de funcionamiento. Por lo que no existe una solución genérica para todos. Además, el hecho de que los dispositivos puedan estar ubicados en diferentes partes del aula dificulta el acceso a los mismos y por ende, la instalación. Es por esto que llegamos a la conclusión de utilizar un medio inalámbrico como es el infrarrojo. Gran parte de los equipamientos funcionan con un control remoto, como es el caso del proyector y el aire acondicionado. Esto resuelve los desafíos planteados y además independiza al sistema de los dispositivos. Ya no es necesario el acceso físico a los mismos por lo que el sistema no se ve afectado si un dispositivo deja de funcionar y es cambiado por otro, incluso si es de otra marca. Los nuevos desafíos ahora son detectar, almacenar y reproducir los diferentes tipos de comandos infrarrojos emitidos por cada marca y modelo.

En cuanto a las luminarias se optó por utilizar un relé para el encendido y apagado de las mismas. Dicho relé se encuentra acoplado al circuito eléctrico de las luces a través de un circuito escalera. Esto permite el uso normal de las mismas, por más que el sistema falle.

Arquitectura

Para el desarrollo del sistema se construyeron diferentes componentes que permiten, a través de una interfaz web, visualizar los diferentes parámetros

ambientales, la ocupación y la capacidad de accionar sobre el equipamiento del aula.

La arquitectura del presente proyecto se modula en los siguientes componentes:

1. Microcontrolador ESP32 [10] conectado a diferentes sensores y actuadores:

- Conexión Wifi a la red local.
- Sensado de los parámetros ambientales.
- Almacenamiento de los mismos en un servidor remoto.
- Capacidad de grabar en el servidor remoto comandos del protocolo NEC.
- Recupero y ejecución de los comandos del protocolo NEC.
- Accionamiento de los actuadores (led infrarrojo y relé).
- Actualizador remoto de firmware.
- Implementa su propio web server para la configuración de las credenciales de acceso a la red local.

2. Aplicación web y hosting:

- Crear, editar y eliminar edificios y aulas, asignándole a cada uno el módulo ESP32 que corresponda.
- Visualización del estado actual del aula.
- Capacidad de accionar: activar relé, grabar y disparar comando NEC.

3. Base de Datos: servicio gratuito provisto por Google.

- Base de datos alojada en la nube.
- Los datos se almacenan en formato JSON.
- Base no relacional, NoSQL.
- Sincronización en tiempo real con cada cliente conectado.

Firestore

Firestore de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil, y está disponible para distintas plataformas (iOS, Android y web).

Una de las herramientas que brinda esta plataforma, son las bases de datos en tiempo real, manteniendo actualizados los datos aunque el usuario no realice ninguna acción.

Firestore envía automáticamente eventos a las aplicaciones cuando los datos cambian. También ofrece un servidor para alojar las apps, esto es, un hosting estático y seguro. Proporciona certificados de seguridad SSL y HTTP2 de forma automática y gratuita para cada dominio, reafirmando la seguridad en la navegación.

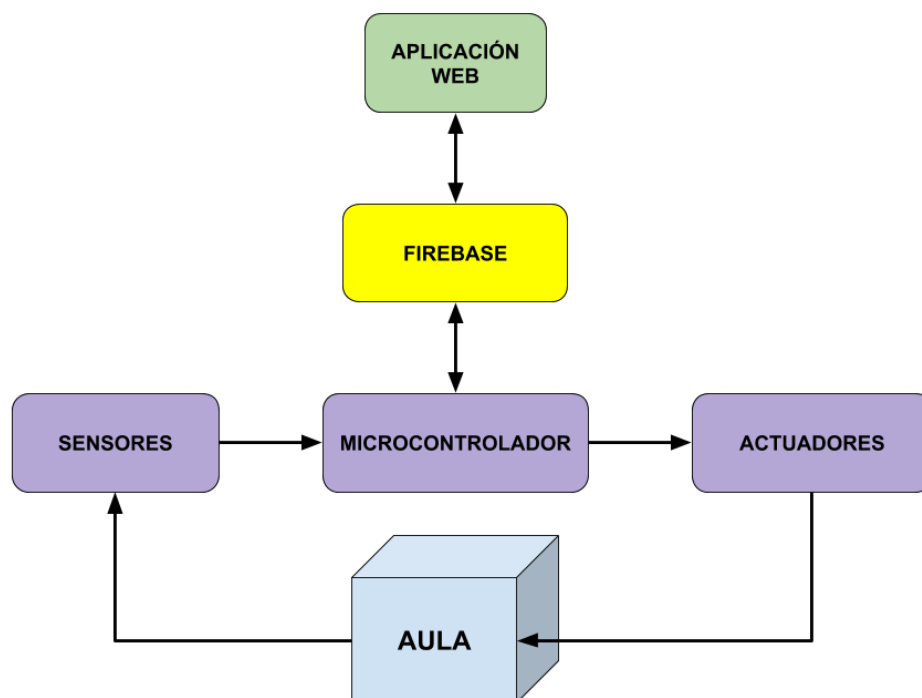


Figura 1. Relación entre los principales componentes.

Estas son solamente algunas de las funcionalidades que ofrece la plataforma y de las cuales más uso hemos hecho dado el tipo de sistema que desarrollamos.

ESP32

La placa de desarrollo o DEVKIT V1 NodeMCU-32 es una herramienta muy potente para el prototipado rápido de proyectos con IoT. Integra en una placa el SoM (*System on Module*) ESP-WROOM-32, que tiene como base al SoC ESP32, el conversor USB-serial CP2102, necesario para programar por USB al ESP32, memoria FLASH, reguladores de voltaje, leds indicadores y antena WiFi en PCB.

La plataforma ESP32 es la evolución del ESP8266 mejorando sus capacidades de comunicación y procesamiento computacional.

A nivel de conectividad, permite utilizar diversos protocolos de comunicación inalámbrica como: WiFi, Bluetooth y BLE. En cuanto a procesamiento, mejora su CPU 32-bit de dos núcleos de hasta 240Mhz que se pueden controlar independientemente.

Además incluye internamente una gran cantidad de periféricos para la conexión con: sensores táctiles capacitivos, sensor de efecto Hall, amplificadores de bajo ruido, interfaz para tarjeta SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C.

Puede alimentarse directamente del puerto micro-USB o utilizando una fuente externa de 5V o 3V dado que posee regulador de voltaje en placa. Sin embargo, los pines de entradas/salidas (GPIO) trabajan a 3.3V por lo que para conexión a sistemas de 5V es necesario utilizar convertidores de nivel.

El ESP32 puede ser programado utilizando el *framework* de Arduino, lo cual brinda un entorno de desarrollo sencillo, con una gran cantidad de librerías y proyectos disponibles para usar.

La comunidad de usuarios de Arduino es muy activa y da soporte a plataformas como el ESP32 y ESP8266.

Especificaciones técnicas:

- Procesador: Tensilica Xtensa 32-bit LX6 microprocessor.
- Núcleos: 2.
- Frecuencia: Hasta 240 MHz
- Ultra low power co-processor: permite realizar conversiones ADC, cálculos y detectar umbrales de nivel mientras se encuentra en modo *sleep*.
- Wi-Fi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz up to 150 Mbit/s).
- Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE).
- Algoritmos criptográficos por hardware (RSA, AES, SHA, RNG).
- ROM: 448 KiB.
- SRAM: 520 KiB.

La lógica implementada en el uCU ESP32, está modularizada en distintos componentes lógicos, basándose en las responsabilidades que cada uno debe cumplir:

Módulo Sensores: el mismo se encarga de inicializar los sensores, leer los valores de cada uno y entregarlos al módulo de Firebase para su almacenamiento en la base de datos. Vale aclarar que la detección de movimiento no es propiamente “medida o sensada”, sino que la detección de un cambio de tensión en el pin genera una interrupción.

Módulo Actuadores: se encarga desde lo más simple como manejar un led de notificación, hasta la lógica que implementa toda la comunicación infrarroja. La misma está basada en la librería IRremoteESP8266. Sus funciones principales son:

- Inicializar el receptor y el transmisor infrarrojos.
- Capturar, decodificar y almacenar los comandos en la base de datos.
- Recuperar comandos de la base de datos.
- Disparar el comando.

Módulo Firebase: implementa la comunicación con la base de datos. Todos los datos que se almacenan en la base o se recuperan de la misma, son manejados por este módulo.

Módulo **DBStream**: este es el principal canal de comunicación entre el ESP32 y el usuario. Todas las acciones que realiza el ESP32, encender o apagar las luces, apagar el aire acondicionado, subir y bajar los comandos infrarrojos, actualización del firmware y refrescar las configuraciones del aula, se ejecutan gracias a este módulo. Se implementa por medio de una *streamCallback*, una funcionalidad brindada por Firebase que permite monitorear cambios (eventos) en un determinado campo o path de la base.

Módulo **OTA**: módulo encargado de las actualizaciones del *firmware*. Cuando el ESP32 recibe el evento indicado, ejecuta el chequeo de versiones de firmware entre, una cargada en memoria (generada en la última compilación) y la versión que se encuentra almacenada en Firebase. Si las versiones son diferentes, el nodo inicia la descarga de la nueva versión, la almacena temporalmente en memoria (para esto tuvimos que redimensionar las tablas de la memoria FLASH) y cuando completa la

descarga, se reinicia ejecutando la nueva versión.

Módulo **Autómata**: las condiciones que deben cumplirse para apagar o no las luces y el aire acondicionado, se evalúan en este módulo. El mismo se encarga de chequear constantemente un rango horario, definido por el usuario, que le indica los lapsos de tiempo en los cuales es evidente que los componentes del aula deben permanecer apagados.

Módulo **APWebServer**: con el fin de poder modificar fácilmente las credenciales de la red a la que se conecta el módulo ESP32, se implementó un pequeño WebServer que acepte como input los datos necesarios para establecer conexión con la red deseada. Al inicio del setup principal, se valida que las credenciales guardadas en memoria no estén vacías. Si este es el caso, el dispositivo cambia su funcionamiento a modo AP+STA (*Access Point and Station*) e inicializa la ejecución de dicho webserver. Este modo de funcionamiento del módulo ESP32 genera una nueva red wifi cuyas credenciales son:

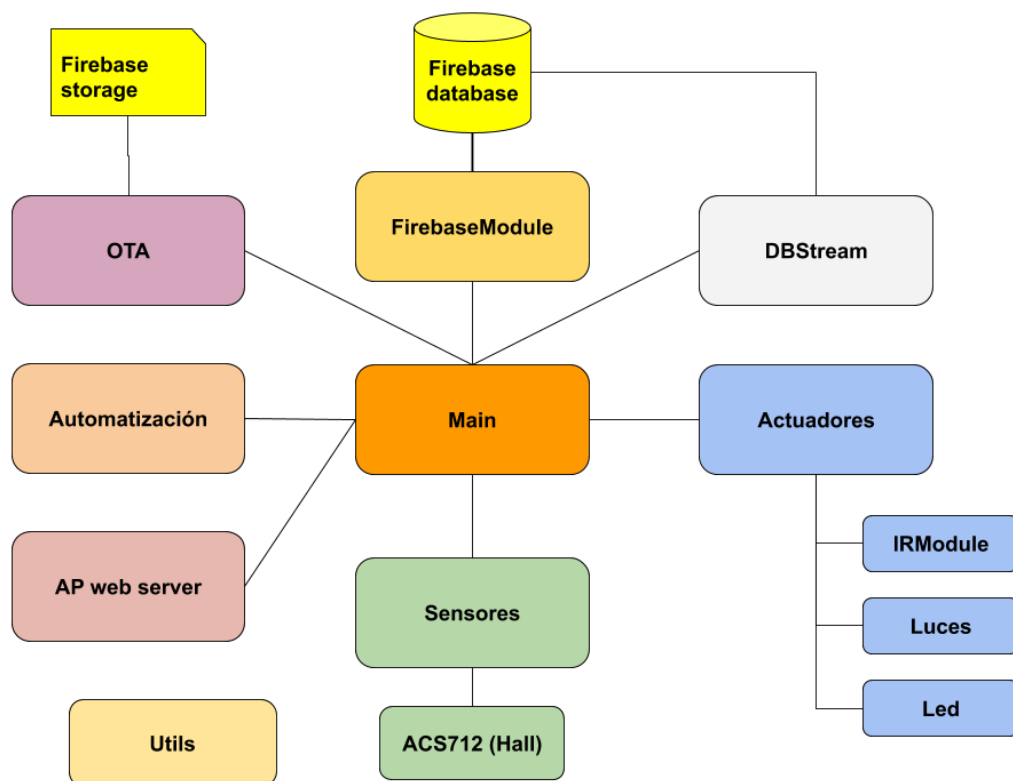


Fig. 2. Resumen de la modularización del software.

- **SSID:** “Aulas Sustentables (AA:BB:CC)”. Donde AA:BB:CC son los 24 bits de la parte inferior de la dirección MAC del ESP32.
- **Pass:** “aulassustentables”.

Finalmente, para poder acceder al server se ingresa a la siguiente dirección desde cualquier navegador: <https://192.168.4.1/>
Módulo Utils: módulo con funciones utilitarias.

Sensores

Un sensor es un dispositivo que está capacitado para detectar acciones o estímulos externos y enviar información a un dispositivo electrónico. Estos aparatos pueden transformar las magnitudes físicas o químicas en señales eléctricas.

Para poder obtener el estado del aula, el microcontrolador necesita recuperar información de su entorno, y para ello necesita contar con diferentes tipos de sensores conectados. En este proyecto utilizamos los siguientes sensores / receptores:

- Sensor de movimiento DSC LC-100-PI.
- Sensor de luz BH1750.
- Sensor de temperatura y humedad DHT22.
- Sensor de corriente ACS712.
- Receptor infrarrojo TL1838.

Actuadores

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un controlador y en función a ella genera la orden para activar un elemento final de control, como por ejemplo un relé. Son los elementos que influyen directamente en la señal de salida del automatismo, modificando su magnitud según las instrucciones que reciben de la unidad de control. En este proyecto utilizamos los siguientes actuadores:

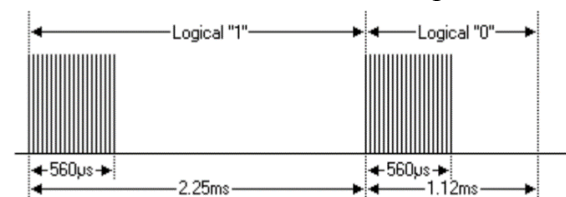
- Led de Notificación.
- Relé 3.3V - 220V.

- Fototransistor L51P3C.
- Led infrarrojo 38kHz.

Protocolo NEC

Se trata de uno de los protocolos más extendidos en todo tipo de aplicaciones remotas a través de infrarrojos, aunque no el único, siendo también comunes otros como el Sharp Protocol, Sony SIRC o Philips RC-5. Los principales rasgos que caracterizan a las señales asociadas a este protocolo son las siguientes:

- 8 bits de dirección.
- 8 bits de datos.
- Tanto los bits de dirección como los de datos son enviados por duplicado (la segunda vez invertidos, haciendo así constante la duración de la trama) para aumentar la fiabilidad.
- Frecuencia portadora de 38kHz.
- La modulación que diferencia entre un “0” y un “1” lógicos, a través de la distinta duración del bit, es de 1.125ms y 2.25ms respectivamente, siendo 560us el ancho de pulso:



- La señal enviada al pulsar un cierto botón estará formada, además de por los citados bytes de dirección y datos por duplicado, por un pulso de 9ms indicador de la llegada, seguido de una pausa de 4.5ms. El conjunto total tendrá, por tanto, la siguiente forma:



Over-The-Air Programming (OTA)

La programación OTA [11] se refiere a varios métodos para distribuir software nuevo a través de una comunicación inalámbrica y remota. En el proyecto, incluimos un pequeño módulo encargado de realizar la tarea de actualizar el firmware

del ESP32. La aplicación web se encarga de recibir un archivo binario, subirlo al almacenamiento de Firebase, escribir la URL obtenida en la base de datos, e indicarle a cada ESP32 que debe chequear si la versión que tiene cargada en memoria difiere de la nueva versión indicada por la aplicación web. Si esto es así, inicia la descarga de la nueva versión, la cual es almacenada temporalmente hasta que se complete la descarga, verificando previamente que se cuente con la memoria suficiente.

Además, al mantener en memoria el firmware actual y el nuevo, si la instalación fallara se vuelve a la versión original. De no haber errores, el ESP32 se reinicia descartando la versión vieja y ejecutando la nueva.

Dashboard Web

Funcionalidades principales:

- Visualizar las aulas, agrupadas por edificios.
- Visualizar el estado actual de un aula.
- Accionar remotamente sobre las luminarias y el aire acondicionado de un aula.
- Configurar la automatización de un aula. Puede establecerse el intervalo de tiempo en el cual se le permite al sistema tomar decisiones. Por si se quiere evitar que el sistema realice alguna acción durante el uso normal de las aulas.
- Grabar comandos de un control infrarrojo en un aula específica. El microcontrolador de dicha aula se coloca en modo escucha para recibir el comando.
- Editar comandos infrarrojos (marca y descripción).
- Actualizar el firmware mediante OTA. Permite la subida de un archivo compilado que será retransmitido a todos los microcontroladores del sistema para que actualicen su firmware.
- Visualización del estado en el que se encuentra un microcontrolador (logs). Se puede visualizar en tiempo real la actividad de cada microcontrolador en funcionamiento.

Alimentación

Al ser un sistema pensado para funcionar las 24hs del día y sin restricciones energéticas, el uso de baterías para su funcionamiento no es algo viable. La solución fue conectar el sistema directamente a la red eléctrica del edificio donde se instaló. Para esto utilizamos fuentes de alimentación que convierten los 220V de la red a los 3.3V que necesita el ESP32 para funcionar. Lo mismo hicimos para el sensor de movimiento pero con una fuente de 220V a 12V.

Capacidades del sistema

- Lectura de la intensidad de corriente del circuito eléctrico de las luminarias.
- Lectura de la luminosidad y temperatura de un aula.
- Determinar si un aula se encuentra o no en uso, mediante la detección de movimiento.
- Encender y apagar las luminarias de un aula.
- Lectura de comandos infrarrojos emitidos por cualquier control remoto.
- Almacenamiento y reproducción de dichos comandos.
- Apagado del aire acondicionado.
- Actualización del firmware de cada nodo del sistema remotamente.
- WebServer en el ESP32 para el ingreso de las credenciales WiFi.
- Dashboard web.

Trabajos Relacionados

Existe un interés creciente en mejorar la eficiencia del uso de la energía, vinculado a políticas de sustentabilidad y protección del medio ambiente [15], y en función de mejorar la seguridad y comodidad de los usuarios [16]. Esto es particularmente cierto en los edificios [14]. Aunque en construcciones nuevas se pueden tener en cuenta, en etapas de diseño, la reducción del consumo de energía incorporando tecnologías, es necesario ofrecer soluciones a las edificaciones existentes; y estas soluciones deben poder incorporarse con el menor costo y daño posible.

Esta realidad es abordada en [12] y [13], donde se exploran los beneficios y las opciones de utilizar tecnologías de automatización e IoT para reducir el consumo de energía. Este trabajo aporta el caso particular del aula y el ambiente educativo, especialmente en las universidades públicas, donde los espacios son compartidos por todos.

Conclusión y Trabajos Futuros

En la universidad, las luminarias y los aires acondicionados son las principales fuentes de consumo de energía eléctrica, por lo tanto reducir su consumo a lo estrictamente necesario es un desafío que se debe tomar en cuenta. Utilizando un sistema de control basado en IoT puede potencialmente mejorar la eficiencia energética de la institución, reduciendo el consumo de energía eléctrica, aumentando la vida útil de los componentes y claramente reduciendo los costos económicos asociados.

En este documento, presentamos un sistema basado en IoT, con interfaz web y comunicación inalámbrica capaz de, medir la energía eléctrica que circula por la red, detectar si un ambiente se encuentra o no ocupado, accionar sobre las luminarias y el aire acondicionado de manera manual o automatizada, mantener un registro de la actividad y actualizar el firmware de los microcontroladores remotamente, es decir, sin la necesidad de intervenir físicamente cada nodo que conforme el sistema. Además el sistema se incorpora a cualquier aula con muy pocas modificaciones en las mismas y sin interrumpir el funcionamiento normal.

Respecto a los trabajos futuros, quisiéramos mencionar que al tratarse de un proyecto que apunta a resolver una problemática energética transversal a muchas instituciones públicas, deseáramos que los datos obtenidos estén almacenados y sean administrados localmente. Nuestro proyecto utiliza los servicios gratuitos de Google por cuestiones de tiempos y facilidad de implementación, pero se podría pensar en

reemplazar este componente por una base de datos almacenada en los servidores de la universidad.

Otra funcionalidad interesante es la de incorporar métricas y estadísticas de consumo y gastos. Por ejemplo, si se guarda las noches que se detecta luces o aire acondicionado encendido se puede calcular el ahorro mensual o anual producto de que el sistema apague dichos dispositivos. Si las luces se apagaron un viernes a la noche son aproximadamente 60hs hasta el lunes a la mañana.

También es una tarea pendiente detectar el estado en el que se encuentran los dispositivos accionados por medio de comandos infrarrojos. Algunos dispositivos utilizan el mismo comando para el encendido y apagado (modo *toggle*). Por lo que, a la hora de apagar el dispositivo, es necesario conocer el estado en el que se encuentra para decidir si emitir o no el comando.

Agradecimientos

- Guillermo Dominela, quien nos propuso la problemática y financió el proyecto.
- Liet Czertok, prácticamente un autor más.
- Virginia Salazar por sus diseños del logo.
- Familiares y amigos por su apoyo incondicional.

Referencias

- [1] <http://esp32.net/>
- [2] <https://firebase.google.com/>
- [3] <https://github.com/akveo/ngx-admin>
- [4] <https://github.com/mobizt/Firebase-ESP8266>
- [5] <https://cms.dsc.com/download.php?t=1&id=14149>
- [6] <https://github.com/crankyoldgit/IRremoteESP8266>
- [7] <https://www.investigacionyciencia.es/blog/s/tecnologia/50/posts/cunto-resistimos-12851>
- [8] K. Moser, J. Harder and S. G. M. Koo, "Internet of things in home automation and energy efficient smart home technologies," *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, San Diego, CA, 2014, pp. 1260-1265, doi: 10.1109/SMC.2014.6974087.

- [9] Lu Tan and Neng Wang, "Future internet: The Internet of Things," *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Chengdu, 2010, pp. V5-376-V5-380, doi: 10.1109/ICACTE.2010.5579543.
- [10] Espressif Systems, "ESP32 Datasheet", October 8, 2016.
- [11] https://en.wikipedia.org/wiki/Over-the-air_programming
- [12] A. Saha, M. Kuzlu and M. Pipattanasomporn, "Demonstration of a home energy management system with smart thermostat control", *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1-8, Feb 2013.
- [13] N. H. Motlagh, S. H. Khajavi, A. Jaribion and J. Holmstrom, "An IoT-based automation system for older homes: a use case for lighting system," *2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA)*, Paris, 2018, pp. 1-6, doi: 10.1109/SOCA.2018.8645771.
- [14] L. Prez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008.
- [15] S. Cox, "Cooling a warming planet: a global air conditioning surge," *Yale Environment* 360, Features section, May 2012.
- [16] E. Nemethova, W. Stutterecker and T. Schoberer, "Thermal comfort and energy consumption using different radiant heating/cooling systems in a modern office building", *Slovak Journal of Civil Engineering*, vol. 25, no. 2, pp. 33-38, 2017.
- [17] <https://aulas-sustentables.firebaseio.com>
Cuenta demo: aulas.sustentables@gmail.com
Contraseña: [aulas.sustentables](https://aulas-sustentables.firebaseio.com)

Datos de Contacto:

Gabriel Salazar Gisbert. Universidad Nacional del Sur (UNS). gsalazargisbert@gmail.com

Lucas Jeremías Fassi. Universidad Nacional del Sur (UNS). jeremiasfassi@gmail.com

José Hipólito Moyano. Universidad Nacional del Sur (UNS). jose.moyano@cs.uns.edu.ar