

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA POR DETECCIÓN DE LLAMA

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

FERNANDO JAVIER TIPANTIZA PACHECO

fernando.tipantiza@epn.edu.ec

DIRECTOR: ING. JAVIER ALEJANDRO ARMAS NAVARRETE

javier.armas@epn.edu.ec

DMQ, septiembre 2022

CERTIFICACIONES

Yo, Fernando Javier Tipantiza Pacheco declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



FERNANDO JAVIER TIPANTIZA PACHECO

fernando.tipantiza@epn.edu.ec

fertipantiza@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Fernando Javier Tipantiza Pacheco, bajo mi supervisión.



JAVIER ALEJANDRO ARMAS NAVARRETE
DIRECTOR

javier.armas@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

FERNANDO JAVIER TIPANTIZA PACHECO

DEDICATORIA

El proyecto realizado es dedicado a mis padres Viviana y Alonso, quienes me han apoyado en mi preparación personal y profesional de todas las formas posibles.

Además, dedico este proyecto a todos los docentes que me han guiado en todos estos años, que gracias a su esfuerzo me encuentro completando una nueva etapa en mi vida estudiantil.

Fernando Tipantiza

AGRADECIMIENTO

Al implementar este proyecto culmino otra etapa de mi vida estudiantil dando gracias a mis padres quienes fueron una guía desde mi nacimiento hasta la actualidad, y a pesar de los tropiezos nunca perdieron la fe en mí.

También a mis abuelitos, que supieron educarme de la mejor manera y gracias a ello tengo valores que me harán crecer aún más.

Y finalmente, al ingeniero Fernando Becerra, quien colaboró como mi tutor desde el primer semestre, por medio de reuniones de estudio para aprobar las materias que he recibido en todo este tiempo.

Fernando Tipantiza

ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VI
<i>ABSTRACT</i>	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	3
1.2 Objetivos específicos.....	3
1.3 Alcance	3
1.4 Marco Teórico	3
2 METODOLOGÍA.....	7
3 RESULTADOS	9
3.1 Establecimiento de requerimientos del prototipo.....	9
3.2 Selección de componentes.....	10
3.3 Diseño del prototipo	13
3.4 Implementación del prototipo	22
3.5 Pruebas de funcionamiento.....	32
4 CONCLUSIONES.....	34
5 RECOMENDACIONES	35
6 REFERENCIAS BIBLIOGRÁFICAS	37
7 ANEXOS.....	i
ANEXO I: Certificado de Originalidad	i
ANEXO II: Enlaces	ii
ANEXO III: Códigos Fuente	iii

RESUMEN

En este proyecto de titulación se plantea la implementación de un prototipo de sistema de alerta por detección de llama. En el primer capítulo se tiene una descripción de los componentes que se desea implementar, por ejemplo, se describe lo que es un módulo ESP32 GSM que incorpora una tarjeta SIM800L, un sensor KY-026, entre otros. También, se definen los objetivos generales y específicos que se desea cumplir para obtener el funcionamiento de un prototipo que envíe alertas al detectar una llama.

La metodología del proyecto se especifica en el segundo capítulo y brinda información sobre los requerimientos que se deben satisfacer para cumplir con los objetivos específicos. Incluye desde la problemática hasta la solución y se explica la función del prototipo tanto en *hardware* como en *software*.

El tercer capítulo se denomina resultados y detalla el proceso de creación del prototipo. Con la ayuda de los componentes seleccionados, investigaciones, cálculos y pruebas se determina el correcto funcionamiento del prototipo para poder ser implementado dentro de un chasis que proteja el circuito de fenómenos que provoquen daños a la placa y sus componentes.

En el cuarto capítulo se describen conclusiones y recomendaciones basadas en la experiencia adquirida en la realización del proyecto, brinda ayuda en la aplicación y rendimiento del prototipo al ser implementado en un evento de llama.

Finalmente, en los anexos se incluye el código de programación, así como el video que puede ser visto a través de un código QR con el objetivo de comprender el funcionamiento del prototipo a través de un manual.

PALABRAS CLAVE: Comunicación inalámbrica, longitud de onda, ESP32, fuente de energía, estructura.

ABSTRACT

In this degree project that arises the implementation of a prototype of a flame detection warning system. In the first chapter have been description of the components that wish to implement, for example, describe what it is ESP32 GSM module that incorporates a SIM800L card, a KY-026 sensor, among others. Also, in relation the general and specific objectives that wish to meet are defined, to obtain the operation of a prototype that sends alerts when detecting a flame.

The methodology of the project is specified in the second chapter, provides information on the requirement that must be satisfied to meet the specific objectives. It includes from a problem to the solution and explains the function of the prototype in both hardware and software.

The third chapter is denominated results and details the prototype process. With the help selecting components, research, calculations, and tests, you are determined the correct functioning of the prototype to be implemented into a chassis that protects the circuit from phenomena that cause damage to the board and components.

The fourth chapter you have conclusions and recommendations that are described based on the experience acquired in carrying out the project, provides help in the application and performance of the prototype when implemented in a flame event.

Finally, the annexes. This includes a programming code, that can be watched by QR codes with the objective to understand function that prototype through a manual.

KEYWORDS: *Wireless Communication, wavelength, ESP32, power source, structure.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El prototipo permite generar alertas al detectar un evento de llama, con el fin de proporcionar información hacia el usuario final quien debe tomar medidas respectivas, evitando que la catástrofe que se extienda a gran velocidad. El prototipo cumple con la función de alertar del evento al usuario registrado y gracias a su compartimento con su fuente de energía puede ser ubicado dentro de un ambiente con un gran registro del fenómeno de llama.

La actividad del prototipo se ve enfocada en la Figura 1.1.; el primer componente es un sensor de flama KY-026, el cual tiene la capacidad de recibir información analógica y digital al ser calibrado por el potenciómetro que lo integra, un módulo SIM800L quien genera una notificación al usuario final. Además, se tiene dos Diodos Emisores de Luz (LED), para informar el estado del prototipo y una batería que es recargable con la ayuda de un controlador de carga TP4056. Tras recibir información de un evento de llama se procesan los datos con la ayuda de un microcontrolador ubicado en la placa ESP32.

Un GPRS ESP32 es un módulo que facilita la comunicación inalámbrica con el usuario. Se emplea en redes 2G a las frecuencias de 850/900/1800/1900 (MHz). Al poseer un microcontrolador tiene la capacidad de Transmitir (Tx) y Recibir (Rx) información con la ayuda de los pines integrados. Este módulo integra una tarjeta SIM800L que ayuda enviando señales de alerta por medio de radio bases celular mediante la programación realizada dentro del módulo. Es muy útil y facilita la creación de prototipos como un sensor que detecta llama a ciertas distancias y en cual ambiente [1].

El sensor de flama KY-026 posee una resistencia térmica que permite oponerse al flujo de la temperatura, cuando se alcanza una temperatura dentro del umbral el LED que incorpora se enciende o se apaga, posee un potenciómetro que varía su señal en relación con la temperatura con el fin de calibrar su sensibilidad, contiene un comparador de voltaje LM393. Cuenta con 4 pines, el primer pin con el símbolo DO se conecta a un pin digital, el segundo ping con símbolo (+) representa a la tensión para que el sensor tenga voltaje, el tercer ping hace referencia a tierra (GND) y el cuarto ping A0 se dirige a la parte analógica. El sensor necesita de un ping análogo y digital para mejorar su precisión y sensibilidad al momento de obtener resultados [2].

El módulo SIM800L permite a un circuito actuar como un equipo móvil ya que implementa tecnología Sistema Global de Comunicaciones Móviles (GSM) y Servicio General de Paquetes Vía Radio (GPRS) para comunicación en redes celulares,

implementa 1 antena que nos permite acceder a la telefonía, Servicios de Mensajes Cortos (SMS), datos, voz ya que GSM puede usar estos servicios para la comunicación entre usuarios en redes 2G. Para su manejo se procede a configurar el número de teléfono a contactar por medio de códigos AT [3].

El prototipo debe contar con la antena direccional integrada al módulo ESP32 capaz de radiar la información a otra antena que se ubica en la radio base.

Una vez funcional el prototipo, el sensor al captar una señal infrarroja o de llama este envía información al módulo ESP32 donde el código se encarga de analizarlo y concluido su resultado pasa al módulo SIM800L, con la ayuda de la antena radia esta información hacia la radio base más cercana. Esta tarjeta SIM800L se encarga de generar un SMS hasta contactar con el número registrado dentro del módulo, informando los hechos que se presentan al usuario, aunque el usuario no se encuentre cercano al suceso.

Dicho esto, los elementos presentados se los implementará dentro de una placa electrónica que vuelve al prototipo funcional. Además, será contenido en un chasis encargado de proteger la función del circuito alejado de factores externos que produzcan daños y permita seguridad. Como punto final el prototipo debe funcionar mediante una fuente de energía que se la puede implementar por medio de una batería de lipo de 3.7 (V) que sea enlazada al prototipo.

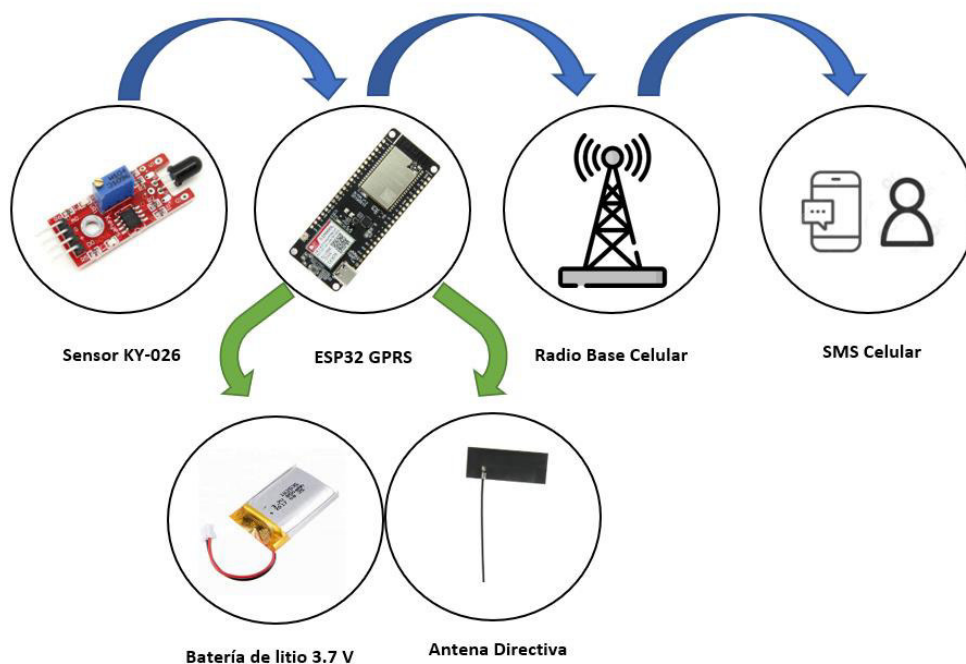


Figura 1.1 Función del prototipo [1] [2] [3]

1.1 Objetivo general

Implementar un prototipo de sistema de alerta por detección de llama.

1.2 Objetivos específicos

- Establecer los requerimientos del prototipo.
- Definir los componentes de hardware y software.
- Diseñar el prototipo.
- Implementar el prototipo.
- Realizar pruebas de funcionamiento.

1.3 Alcance

A través del establecimiento de requerimientos se analizará las necesidades que se tiene para que el prototipo sea funcional. La definición de componentes abarcará una investigación comparativa sobre los dispositivos a ser utilizados, concluyendo en la selección de los mismos. El diseño incluirá una vista en 3D del prototipo, los cálculos necesarios que justifiquen el circuito electrónico, así como el diagrama de flujo correspondiente. La implementación considerará la elaboración de una placa electrónica, así como la selección del chasis más adecuado que garantice la mayor estabilidad y estética para el prototipo. En las pruebas se imitará la presencia de los eventos correspondientes como lo es una llama, de modo que se verifique que una alerta sea enviada inalámbricamente al usuario, alertándolo para que tome las medidas correspondientes.

1.4 Marco Teórico

Evento de llama

Una llama es un suceso que se presenta en el fuego, este se produce cuando un gas combustible se encuentra afectado ante un intenso calor y sus partículas están en estado de excitación. Cuando la reacción se presenta, genera lo que se conoce como llama. Existen factores que se encargan de variar los resultados del fenómeno

mencionado; estos son, condiciones meteorológicas, alrededores, o en su defecto medio en el que actúa el suceso, entre otros [4] [5].

Una llama tiene usos en la parte doméstica, deterioro de materia, energía térmica, entre otros. A medida que la longitud de la llama aumenta, esta llega a un punto de descontrol hasta generar graves daños en la zona, alrededores, incluso daños físicos a personas cercanas al suceso. Tomar en cuenta que este evento puede controlarse al tener información de lugar, hora en la que se desarrolla, temperatura e intensidad. Con estos datos las personas pueden tomar medidas anticipadas hasta llegar a una solución [4].

Una de las características de la llama es que posee una longitud de onda que varía entre 185 a 260 (nm), se encuentra dentro de la luz ultravioleta (UV) visible y puede ser captada por diferentes dispositivos como los son los sensores de llama. Hay que tomar en cuenta que ondas se someten a ciertos fenómenos como; refracción, reflexión, difracción, entre otras, que ayuda de gran manera a captar estas longitudes de onda [6].

La reflexión se encarga de que la onda al momento de chocar con un objeto esta sea reflejada con el mismo ángulo en el que se proyectó con mayor o menor proporción. Mientras que la refracción pasa a través de un material con un cambio de dirección [7]. Por último, el fenómeno de la difracción, al tener un obstáculo semejante a la magnitud de longitud de onda, se la puede cruzar como una rejilla, no obstante, los obstáculos tienen que ser muy pequeños para aprovechar su función [8].

Módulo ESP32 GSM/GPRS

Un prototipo requiere de microcontroladores que se encargan de controlar el funcionamiento del circuito. El ESP32 es un módulo encargado de controlar el prototipo funcional, integra un microcontrolador para programar, transmitir (Tx) y Recibir (Rx) información por medio de pines serie. Este módulo cuenta con 7 pines analógico a digital, un conector USB-C para alimentar la placa, entre otros.

Existe una variedad de funciones que pueden ser implementadas dentro del módulo, uno es establecer la comunicación inalámbrica a través de Bluetooth, Wifi, Redes Celulares, entre otras. El ESP32 se lo implementa dentro del prototipo para el control total de los componentes, con la ayuda del microcontrolador y del software Arduino IDE es posible establecer parámetros que faciliten el manejo de módulo [9]. En la Figura 1.2 se establece las distribuciones de pines que operan a través de las instrucciones establecidas dentro del módulo.

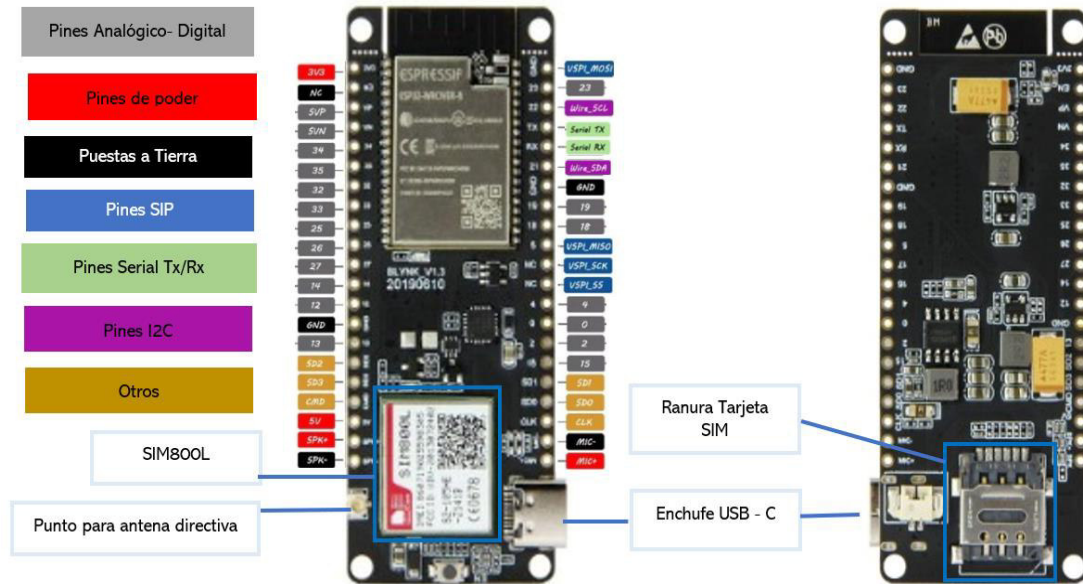


Figura 1.2 Distribución de pines del ESP32 GSM/GPRS [9]

SIM800L

El módulo ESP32 GSM/GPRS integra una tarjeta SIM800L que trabaja con la tecnología de Red Celular de Segunda Generación (2G). Para establecer la comunicación, esta tarjeta debe conectarse a una radio base (BS) con ayuda de una antena directa, misma que se encargará de radiar la información hasta el destino. La tarjeta SIM800L trabaja con una banda cuádruple a las frecuencias de 850/900/1800/1900 (MHz) y cuenta con un conector IPX donde se ubica la antena directa, tal como se muestra en la Figura 1.3. Además, cuenta con pines encargados de Tx y Rx para la comunicación con el módulo ESP32, también contiene un led de conexión a la red y cuenta con puntos para agregar circuitos relacionados al audio como micrófonos y parlantes [3].

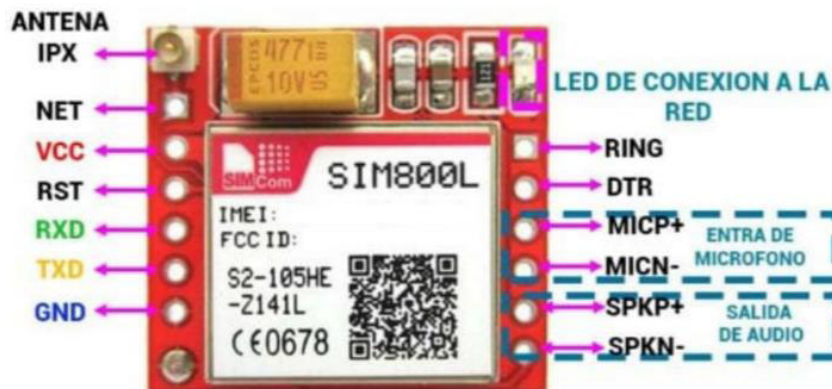


Figura 1.3 Distribución de pines de SIM800L [3]

Sensor de llama KY-026

Es un sensor que cuenta con 3 componentes que se encargan de captar, procesar y enviar información hacia el módulo ESP32. La primera unidad del sensor es encargada de medir el área y, al detectar una longitud de onda perteneciente a una llama, envía una señal analógica a la segunda unidad, que es el amplificador; este cuenta con un potenciómetro que se encarga de enviar otra señal amplificada al tercer módulo. En este último se establece un comparador encargado de cambiar la señal a una salida digital [10]. Posee 4 Pines que como se muestra en la Figura 1.4, para transmitir información al módulo. El primer pin envía señales digitales al módulo, el segundo se conecta a la fuente de energía, el tercer pin se dirige a GND y el cuarto pin envía señales analógicas al módulo.

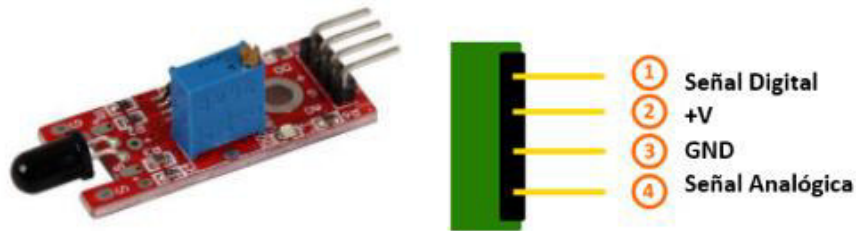


Figura 1.4 Distribución de pines de Sensor KY-026 [10]

El led infrarrojo puede detectar longitudes de onda dentro del rango de 760 a 1100 (nm). Hay que tomar en cuenta la sensibilidad del sensor ya que puede reaccionar a la luz, por lo que se regula el potenciómetro para adaptarse en la zona que se colocará el sensor [10].

Servicio GSM

La tecnología que implementa el Sistema Global de Comunicaciones Móviles (GSM), transmite y recibe Servicio de Mensajes Cortos (SMS), datos, red y correos. Su estructura funciona desde una Estación Móvil (MS) para acceder a los distintos servicios mencionados [11].

Luego se tiene una Estación Transceptora Base (BTS) muy útil para Transmitir y Recibir señales de radio que envían la MS. Las BTS deben ser organizadas y gestionadas por Controladores de Estación Base (BSC). El Centro de Conmutación Móvil (MSC) se encarga de conmutar los puntos de comunicación que se da en las BTS, con el objetivo de procesar el mensaje hacia su destino [11] [12].

2 METODOLOGÍA

Se establecieron cinco etapas como se muestra en la Figura 2.1 para la elaboración del proyecto, facilitando el desarrollo de los objetivos específicos con el fin de implementar el prototipo de forma sencilla.



Figura 2.1 Estructura de la Metodología

En base a un análisis a la problemática que se da ante un evento de llama se establecieron los requerimientos del prototipo, el proyecto cumple con el objetivo de detectar longitudes de ondas pertenecientes a una llama y alertar al usuario final. Se tomo en cuenta las condiciones del medio que rodean al prototipo, así como los componentes integrados dentro del proyecto. También, se estableció la fuente de energía recargable que permite movilidad con el fin de que el prototipo tenga comunicación celular con el usuario en cualquier lugar que tenga cobertura 2G con la tecnología GSM.

Los componentes cumplieron la función de comunicación entre sensor y usuario. Para lo cual, se consideraron diferentes factores como: precios de elementos, tamaño, durabilidad, calidad y uso en cada elemento.

Al comparar los componentes, se obtiene un análisis de cual es la mejor forma de brindar una mejor comunicación GSM/GPRS en una red celular y módulo ESP32. Este módulo se eligió en base a su costo, manejo y compatibilidad con los componentes necesarios para elaborar el proyecto. Además, el módulo cuenta con una tarjeta SIM800L como medio de comunicación entre el prototipo y el usuario. Para detectar longitudes de onda perteneciente a una llama se hizo uso del sensor KY-026 que capta valores analógicos y digitales en tiempo real.

De modo que, estos componentes se configuraron con la ayuda de un software de código abierto donde digitalizamos un algoritmo fácil de comprender y manipular, lo cual permitió cumplir con el objetivo dar instrucciones a cada dispositivo por medio del microcontrolador. Se debe tomar en cuenta el tamaño de cada elemento, por lo que se ocupó el *software Fritzing* para que se generen las pistas del circuito. Una de las características dentro de las pistas que conectan a los componentes es que al cruzarse

con otra se originan cortocircuitos, provocando que el prototipo no tenga funcionalidad. Adicionalmente, se implementó el prototipo mediante un diseño en 3D con ayuda del *software SketchUP*, el cual contiene una vista tanto del chasis, placa y elementos que conforman al prototipo funcional.

No hay que olvidar que la seguridad y durabilidad del prototipo se analizaron para implementarse dentro del mismo. Para la energizar el prototipo se aplicaron fórmulas para definir el tipo de resistencias a usar, mientras que en durabilidad se estableció un tiempo de operación para la elección de una fuente de energía.

Una vez que se terminó con el diseño, se inició con la implementación del prototipo dentro de una placa de cobre. Este se sometió a un proceso de planchado, quemado, perforación y soldadura que son fundamentales para el funcionamiento del sensor. Este diseño se incorporó dentro de un chasis transparente con el fin de visualizar el funcionamiento del sistema. El chasis contiene ranuras que permiten al sensor captar las señales de llama, otra ranura para instalar un *switch* el cual controla el inicio o apagado del prototipo y, por último, una ranura para la salida de la antena vinculada al módulo.

Además, la programación de este prototipo se vio cargada dentro del módulo ESP32 que contiene un microcontrolador capaz de recibir información, analizarla y procesarla para controlar el prototipo. Este código cuenta con el número de usuario a quien va dirigido el SMS, variables para guardar información, librerías para generar un mensaje, entre otras cosas.

Luego de la implementación del prototipo se realizaron las respectivas pruebas de funcionamiento, así como la corrección de los errores presentados, logrando finalmente que el prototipo envíe al usuario una alerta mediante SMS cuando detecta ciertos niveles de llama.

3 RESULTADOS

En esta sección se detalla el proceso de establecimiento de los requerimientos para el funcionamiento del prototipo, así como la selección de los distintos módulos que se utilizaron en su implementación. También, se explica el proceso de diseño que se siguió. Además, se desarrolla la implementación del prototipo paso a paso para la obtención del producto final, empezando por la elaboración de la placa electrónica, la colocación de los distintos módulos de hardware, la programación del módulo ESP32 en el IDE de Arduino con la respectiva explicación del código desarrollado y la colocación del chasis de protección. Finalmente, se detallan las pruebas de funcionamiento realizadas.

El prototipo de sistema de alerta por detección de llama capta información a través del sensor KY-026, que detecta longitudes de onda propias de la llama y envía una señal hacia el módulo ESP32 con la información recolectada en tiempo real, donde es procesada y se genera una alerta que luego es transmitida con la ayuda de la tarjeta SIM800L a través de la red celular hacia el usuario final mediante SMS.

3.1 Establecimiento de requerimientos del prototipo

La función del prototipo es detectar, analizar y alertar al usuario sobre un evento de llama en tiempo real. Para cumplir con los requerimientos es necesario definir el sitio adecuado para el análisis de información de la llama. Gracias al sensor se estima que la mejor forma de captar un evento así es fijar el prototipo a una pared o techo donde exista la probabilidad de darse el fenómeno. No obstante, existen zonas externas que no tiene esta posibilidad por lo que se establece un sitio que proteja el prototipo de otros factores como lo es la lluvia.

Si el prototipo se sitúa dentro de un hogar lo más común es instalarlo cerca de dispositivos que de una u otra forma generen fuego. Se deben realizar pruebas de funcionamiento para evitar que el prototipo envíe alertas innecesarias debido a la operación normal de otros equipos, por lo que también es necesario tomar en cuenta la sensibilidad del sensor al momento de la configuración.

Al situar el prototipo externamente, se pueden tomar en cuenta zonas propensas a que ocurran eventos como incendios. Es necesario considerar que el sitio donde va a operar el prototipo debe encontrarse dentro del rango de cobertura de una red celular 2G, caso

contrario no se podrá realizar el envío de la alerta mediante SMS al usuario. El chasis que cubre al prototipo añade protección ante factores externos.

Dado que el proyecto necesita de la red celular para enviar la alerta, es decir, que la comunicación se hará de forma inalámbrica, es necesario considerar también una fuente de energía que se encarga brindar una tensión a los componentes dentro del prototipo.

Para cumplir con la comunicación inalámbrica es necesario una tarjeta SIM capaz de brindar el servicio SMS entre prototipo y usuario. No obstante, se tiene que generar un mensaje que notifique al usuario del evento de forma inmediata, por lo que se requiere de un microcontrolador ESP32 capaz de procesar información captada por el sensor para obtener una alerta llama.

3.2 Selección de componentes

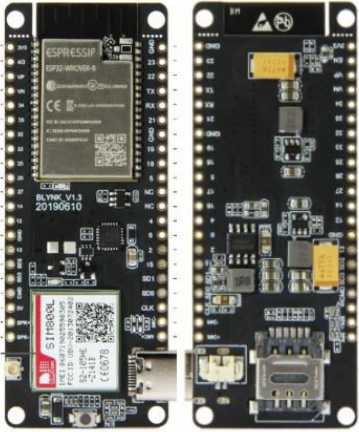

El prototipo requiere de una placa con todos los componentes incorporados, entre ellos una tarjeta microcontroladora, un sensor de detección de llama y un módulo para la comunicación inalámbrica.

Para la tarjeta microcontroladora se encontraron dos opciones, el ESP32 GSM/GPRS y el Arduino Uno, ambos de fácil programación utilizando el IDE de Arduino que, a su vez, es gratuito. Estos dos módulos presentaron una diferencia de componentes dentro de sus PCB.

Ambos módulos contienen conexiones de entrada y salida que son necesarios para el prototipo que tiene caminos hacia los componentes electrónicos. El ESP32 presentó una ventaja notoria ante Arduino Uno ya que este incorpora una tarjeta SIM800L que es un requerimiento importante en la comunicación SMS.


Una breve comparación de características entre el ESP32 GSM/GPRS y el Arduino Uno se muestra en la Tabla 3.1. Luego de analizar las características de ambos, se eligió el módulo ESP32 GSM/GPRS debido a su gran ventaja de tener incorporada una tarjeta para la comunicación inalámbrica que permitirá el envío de notificaciones a través de la red celular.

Tabla 3.1 Comparación entre Módulo ESP32 GSM/GPRS y Arduino Uno [9] [13]

Datos	ESP32 GSM/GPRS	Arduino Uno
Pines Energía	Vin; 3.3 (V), 5 (V), GND	Vin; 3.3 (V), 5 (V), GND
Pines Analógico – Digital	14	20
Pines Serial	1	1
Microcontrolador	ESP32-Wrover-B	ATmega328P y ATmega16u2
Voltaje de funcionamiento	2.7 – 3.6 (V)	7 – 12 (V)
Memoria RAM	520 (Kb)	32 (Kb)
Rango de temperatura	-40° C → +85° C	-40° C → +85° C
Observaciones	Tiene integrado una tarjeta SIM800L y una antena direccional reduciendo costos en componentes a implementar.	Los recursos de apoyo consumen una cantidad de corriente que apaga obliga al módulo a reiniciarse.
Imagen		



Dado que el módulo requerido para el envío de la alerta de forma inalámbrica ya está incorporado en el ESP32, no fue necesario adquirir otro. Este módulo, llamado SIM800L, cuenta con las características necesarias para una comunicación inalámbrica por medio de redes celulares 2G como se muestra en la Tabla 3.2. No obstante, y como se explicó anteriormente, la cobertura de la tecnología 2G dentro del área de funcionamiento del prototipo es de suma importancia para notificar al usuario por medio de un mensaje de texto cuando se genere un evento de llama.

Tabla 3.2 Características tarjeta SIM800L [3]

Datos	SIM800L
Banda de frecuencia	800/900/1800/1900 (MHz)
Rango de temperatura	-40°C → +85° C
Voltaje de funcionamiento	3.4 – 4.4 (V)
Corriente	0.7 (mA) modo sueño 500 (mA) modo trabajo
Velocidad de Tx	1.2 – 115.2 (Kbps)
Comandos	AT
Tamaño Tarjeta Chip	Microchip
Imagen	

En cuanto al sensor requerido para identificar longitudes de onda provenientes de la llama, se encontraron dos opciones de sensores con propiedades semejantes, el KY-026 y el FC-01_FL. La Tabla 3.3 muestra las características de cada uno de ellos. Es necesario considerar el ángulo y la distancia en la que puede trabajar el sensor. Luego del respectivo análisis comparativo, se decidió que el sensor KY-026 es la mejor opción, ya que tiene integrado un potenciómetro que regula la sensibilidad del sensor ante la llama, además incluye un indicador de funcionamiento y de salida de información dando a conocer que el sensor funciona correctamente [14].

Tabla 3.3 Comparación entre sensor KY-026 y FC-01_FL [10] [15]

Datos	KY-026	FC-01_FL
Voltaje de operación	3.3 – 5.5 (V)	3.3 – 5 (V)
Rango de operación	760 (nm) a 1100 (nm)	760 (nm) a 1100 (nm)
Ángulo de detección	60°	60°
Distancia	1 (m) pero varía de la magnitud de la llama	80 (cm) pero varía de la magnitud de la llama
Corriente de operación	15 (mA)	15 mA
Salidas	Pin analógico y Pin digital	Pin digital
Ajuste de sensibilidad	Fácil ajuste en el potenciómetro	Se necesita de una herramienta extra para su ajuste
Led de operación	Indica que el dispositivo funciona	No posee
Imagen		

3.3 Diseño del prototipo

Placa electrónica

Mediante el *software Fritzing* se diseñó una placa electrónica para el prototipo. Este *software* cuenta con dispositivos electrónicos que permiten simular una PCB de manera profesional. Además, posee con una interfaz esquemática, placa de pruebas, circuito impreso y código abierto. También, tiene una opción de añadir elementos que no se encuentran registrados en el programa, solo hace falta buscar las librerías para su simulación. Otra característica del *software* es que las pistas de la placa PCB pueden ser modificadas a un tamaño en el que el usuario pueda trabajar. Los caminos pueden ser colocados de manera automática, pero, al colocarlos de manera manual, se puede optimizar el espacio dentro de la placa. En el diseño de la placa es importante considerar

que las líneas de conexión no deben cruzarse entre sí, puesto que esto generaría un cortocircuito.

La Figura 3.1 muestra la placa electrónica diseñada, donde se pueden visualizar las líneas de conexión entre los distintos componentes. También, se puede observar el módulo ESP32, en el que cada pin maneja un voltaje de 3.3 (V), por lo que consideró que los LED estén conectados a una resistencia para controlar el flujo de corriente, impidiendo que estos se quemen.

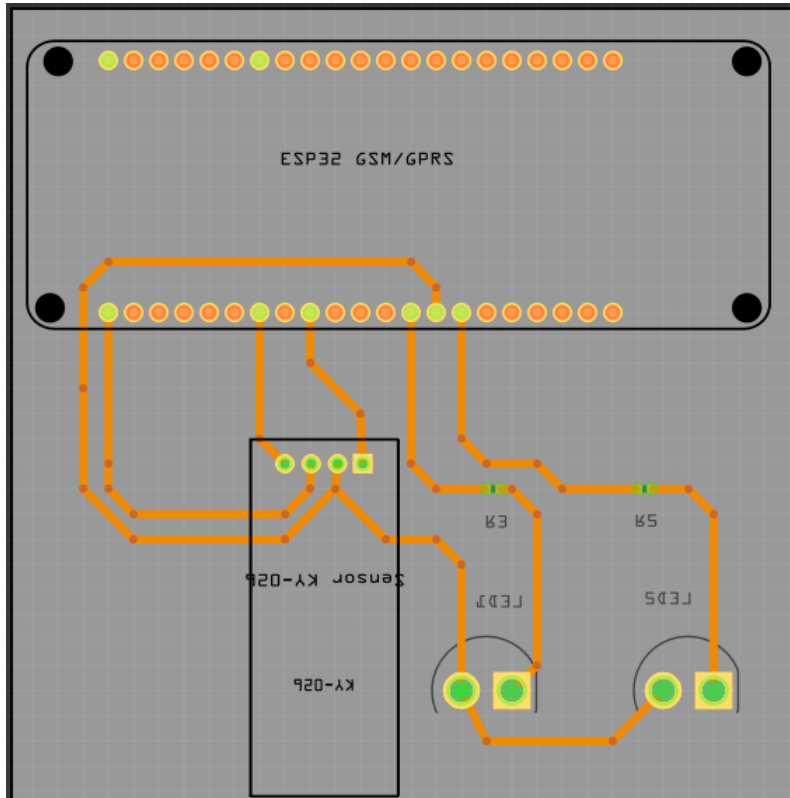


Figura 3.1 Líneas de conexión en *Fritzing*

Estos datos se obtendrán mediante la Ecuación 3.1 presentada a continuación:

$$V = I \cdot R$$

Ecuación 3.1 Ley de Ohm

Los LED implementados se denominan indicadores y trabajan en corrientes de 15 (mA) hasta 30 (mA). Se decidió trabajar con una corriente de 27 (mA), una corriente cercana al máximo para que la luz del LED sea distinguible.

Una vez obtenidos los datos de voltaje y corriente para los LED, se procede a aplicar la ley de Ohm como se muestra en la Ecuación 3.2:

$$R_{LED} = \frac{V_{LED}}{I}$$

Ecuación 3.2 Ley de Ohm para el cálculo de la resistencia

Donde:

V_{LED} : 3.3 (V) voltaje de salida de pines ESP32

I : 27 (A) corriente

$$R_{LED} = \frac{3.3 (V)}{27 (mA)} = 120 (\Omega)$$

De la expresión anterior se puede determinar que a cada LED se le debe aplicar una resistencia de 120 (Ω), limitando el flujo de corriente recibida del módulo ESP32 para impedir que estos se quemem.

Consumo de energía

El módulo tiene la posibilidad de recibir 5 (V) a través del conector USB-C, y 3.7 (V) a través de una batería de tipo LiPo que se conecta al conector JST. El consumo eléctrico se distribuye en el módulo, cada pin brinda un voltaje de 3.3 (V) que se dirige al resto de componentes del prototipo. En la

Tabla **3.4** se muestra el voltaje de funcionamiento de cada componente implementado en el prototipo, con excepción de los LED.

Tabla 3.4 Voltaje de cada componente

	ESP32 GSM/GPRS	SIM 800L	Sensor KY-026
Voltaje (V)	2.7 (V) a 3.6 (V)	3.4 (V) a 4.4 (V)	3.3 (V) a 5.5 (V)

No hay que dejar de lado la corriente que manejan los componentes dentro del prototipo. Debido a que se tiene que mantener los componentes seguros, se establece la corriente de operación dentro de la Tabla 3.5:

Tabla 3.5 Corriente de operación de cada componente

	ESP32 GSM/GPRS	SIM 800L	Sensor KY-026
Corriente (mA)	70 (mA) a 500 (mA)	0.7 (mA) a 500 (mA)	15 (mA)

Mediante los valores presentados se calcula el consumo de corriente del prototipo por medio de la Ecuación 3.3. Algunos valores presentados son corrientes muy bajas, es

decir que el componente se encuentra en modo *sleep* por lo que ocupa muy poca energía.

$$I_{prot} = I_{ESP32} + I_{SIM800L} + I_{Sensor}$$

Ecuación 3.3 Corriente del prototipo

Donde:

I_{ESP32} : 70 (mA) corriente mínima y 500 (mA) corriente máxima ESP32

$I_{SIM800L}$: 500 (mA) corriente de funcionamiento SIM800L

I_{Sensor} : 15 (mA) corriente sensor KY-026

Por lo tanto:

$$I_{prot} = 70 (mA) + 500 (mA) + 15 (mA) = 585 (mA)$$

Con el valor obtenido mediante la ecuación anterior, se procede a realizar el cálculo de la capacidad que tendrá la batería a implementar dentro del prototipo. Se desea que el prototipo funcione ininterrumpidamente durante un tiempo considerable y que su consumo de energía sea bajo cuando los componentes se encuentren en modo *sleep*. La Ecuación 3.4 permite calcular la capacidad de la batería que necesita el prototipo para un tiempo de uso de 2 a 3 horas.

$$C_{Bat} = I_{prot} * T$$

Ecuación 3.4 Capacidad de la batería

Donde:

C_{Bat} : Capacidad de la batería

I_{prot} : 585 (mA) corriente del prototipo

T : 2 - 3 (h) tiempo de funcionamiento

Por lo tanto:

$$C_{Bat} = 585 (mA) * 2 (h) = 1170 (mAh)$$

$$C_{Bat} = 585 (mA) * 3 (h) = 1755 (mAh)$$

De las expresiones anteriores se puede determinar que la batería necesita una capacidad de 1170 (mAh) a 1755 (mAh) para que el prototipo funcione en las condiciones deseadas.

Justificación de rangos de operación del prototipo

La operación del sensor KY-026 depende de las longitudes de ondas captadas y la distancia en la que se encuentra el evento de llama; estos factores determinan el rango de funcionamiento del prototipo. Al detectar una longitud de onda pertenecientes a un evento de llama, cambia el estado funcional de los componentes que se encontraban en modo *sleep* a modo *up*. Mientras que al presentarse longitudes de onda diferentes al de un evento de llama el prototipo vuelve a su modo *sleep* evitando un alto consumo de energía. Estos valores se verán reflejados en los pines analógico y digital del sensor KY-026 hacia el módulo ESP32[16].

Al encontrarse apagados el resto de los pines, se tiene un ahorro de energía. Mientras que el sensor contiene un comparador LM393 el cual opera un suministro de energía simple o doble [17]. El LM393 nos permite tener una salida analógica y digital, esta salida digital se conecta al potenciómetro incorporado que permite configurar la sensibilidad del módulo KY-026 [18].

La potencia que debe salir por el pin análogo depende de la resistencia, sensibilidad y la tensión que se da en el led infrarrojo, se realiza un análisis como se muestra en la Tabla 3.6; sobre cómo opera el sensor en un rango de longitudes de ondas [19].

Tabla 3.6 Rango de operación del sensor

Datos (nm)	Clasificación de longitud de onda
760 - 1100	Operable
> 1100	No operable

Los datos presentados se dividen en dos partes ya que al ser un sensor infrarrojo detecta varios tipos de luces. Por otra parte, la llama trabaja en un rango de onda Ultravioleta, luz visible e infrarrojo como se muestra en la Tabla 3.7; estos valores se encuentran dentro de las longitudes que puede detectar el sensor KY-026.

Tabla 3.7 Longitud de onda perteneciente a una llama [19] [20]

Radiación	Longitud de onda	Longitud de onda de una llama
Ultravioleta	100 – 400 (nm)	< 0.3 (um)
Luz visible	400 – 780 (nm)	0.7 – 1.1 (um)
Infrarroja	780 (nm) – 1 (mm)	> 1.1 (um)

En la banda de onda ultravioleta cuando la llama no es visible hacia el usuario, el sensor tendrá un tiempo de reacción de 3 a 4 (ms), hay que tener en cuenta que se da un retraso por si detecta una luz no perteneciente a una llama. Cuando pasa a ser una onda visible se somete a un análisis de radiación, una vez analizado pasa por varios canales con el fin de crear una imagen de llama. Una vez que se extiende la longitud de onda a infrarrojo pasa a ser considerado como un incendio [19].

Una vez analizada la información de la Tabla 3.6 y Tabla 3.7 se establecen los datos que fueron ocupados dentro del proyecto. Al considerar que es un prototipo de llama nos centramos hasta el valor de 1.1 (um). En la Tabla 3.8 se establece el rango de operación del prototipo, este tendrá dos resultados a presentar si hay una llama o si no hay ningún evento de llama.

Tabla 3.8 Rango de operación del Prototipo

Valores para pin digital	Valores para pin analógico	Clasificación de longitud de onda
0.7 - 1.1	100 – 199	Detectado
0 – 0.69	> 200	No Detectado

Diagrama de flujo

El diagrama de flujo que se da a conocer en la Figura 3.2 muestra el desarrollo del código de programación dentro de Arduino IDE. El diagrama de flujo inicia declarando variables para los distintos pines a utilizar con el objetivo de Tx y Rx datos al microcontrolador. Todas las variables son afectadas por las librerías incluidas dentro del código.

Con el objetivo de generar un mensaje SMS, se hace uso de comandos AT como una salida de las variables a mostrar. Dentro del diagrama de flujo de la Figura 3.2. Se brinda la estructura de como inicia el código, configuración de pines, tipo de módulo, configuración de entradas y salidas tanto para analógico como digital.

Luego se hace uso de condiciones *if else* ya que permite establecer el rango de operación del prototipo permitiendo la activación de la alerta de llama. Esta condición

es la encargada de activar o desactivar la notificación que debe ser plasmado en el SMS destinado al usuario.

- Si el valor de sensor infrarrojo detecta una llama cerca, enviará valores menores a 200 cambiando el estado del LED rojo a *HIGH* y el LED verde a *LOW*. No obstante, esta orden solo se acepta si el número PIN añadido anteriormente es perteneciente al Chip integrado dentro del módulo ESP32.
- Si el número PIN es correcto procede a enviar una notificación al usuario registrado acerca del evento que ocurre cerca del prototipo. Caso contrario se imprime un mensaje en el puerto serial "Mensaje no enviado simPIN incorrecto".
- Finalmente, si se presentan valores mayores a 200 el LED rojo cambia de estado a *LOW*, mientras que el LED verde pasa a *HIGH*.

Los valores captados cumplen un ciclo que consumen energía, por lo que se establece un tiempo de espera para volver a captar otra información. En el caso de registrar un evento de llama el tiempo de espera es de 10000 (ms). Mientras que el tiempo de espera en el caso de no registrar un evento de llama es de 5000 (ms).

El proceso de extracción de información es repetitivo cuando el prototipo se encuentra en funcionamiento por lo que solo se puede llegar al fin del código apagándolo o que la fuente de energía se haya terminado.

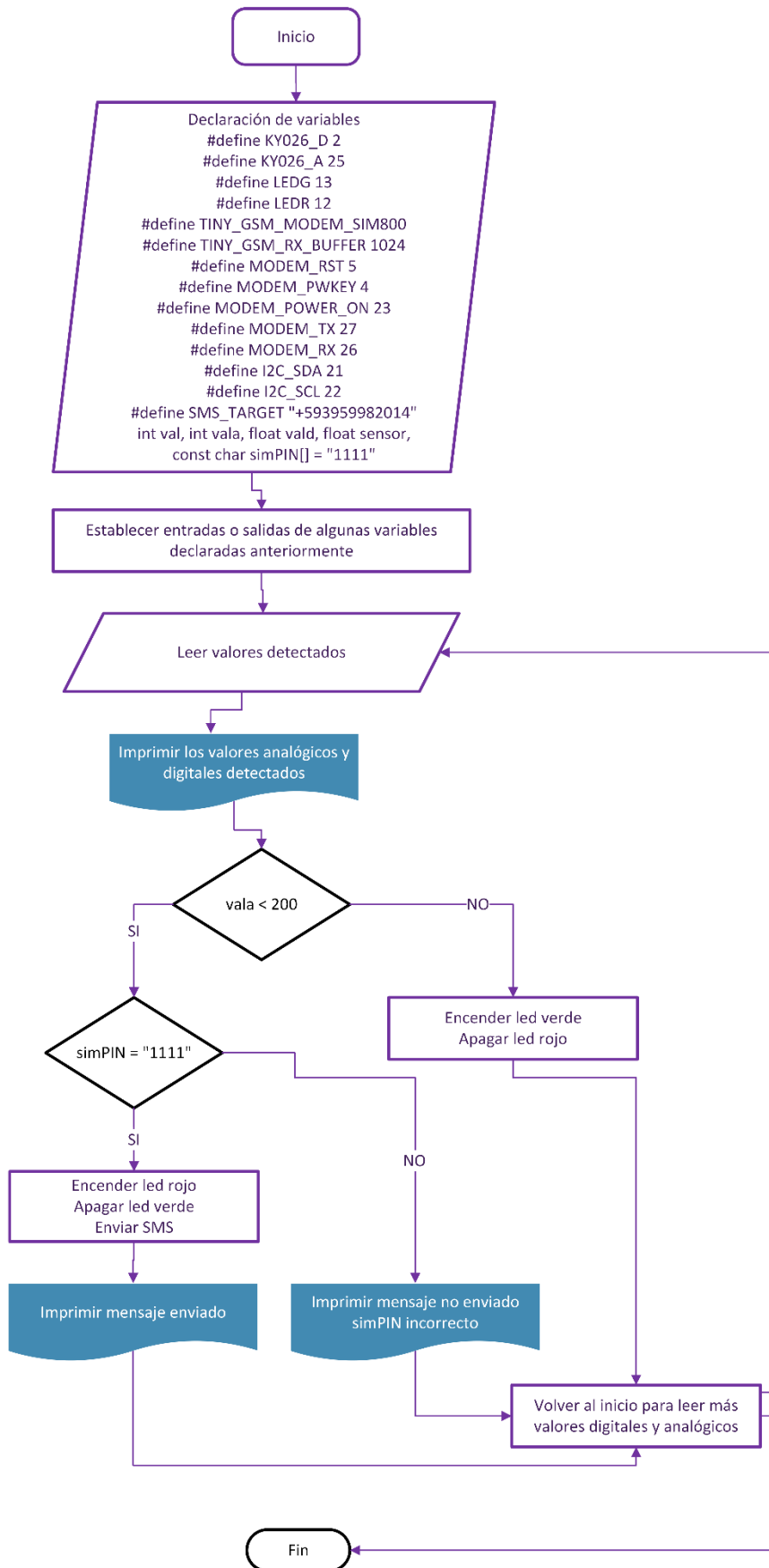


Figura 3.2 Diagrama de Flujo

Vista 3D del prototipo

En la Figura 3.3 se puede visualizar una representación en 3D del prototipo desarrollado. Este esquema fue diseñado mediante el *software SketchUp*, y permite mostrar los componentes que conforman el prototipo. *SketchUp* posee una interfaz amigable con el usuario y permite la edición de los archivos descargados. Por ejemplo, el módulo ESP32 requiere de una antena direccional que no estaba dentro del archivo descargado, por lo que fue necesario modificarlo después.

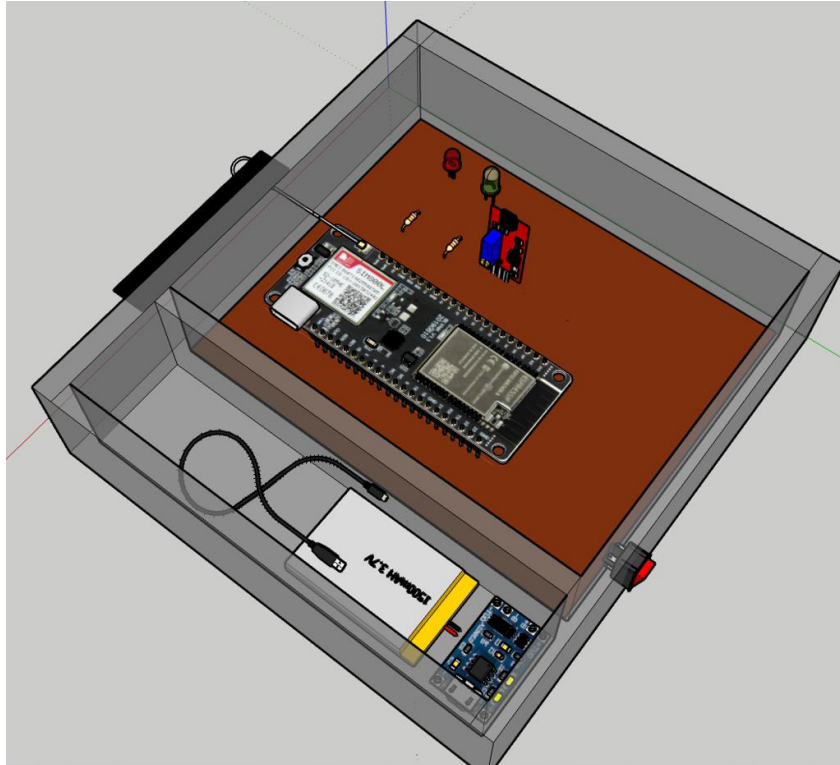


Figura 3.3 Prototipo 3D en *SketchUp*

La Figura 3.4 muestra una vista superior del prototipo. En la misma, se aprecian los compartimentos donde se ubica la placa y la batería, así como los distintos componentes que forman parte del prototipo.

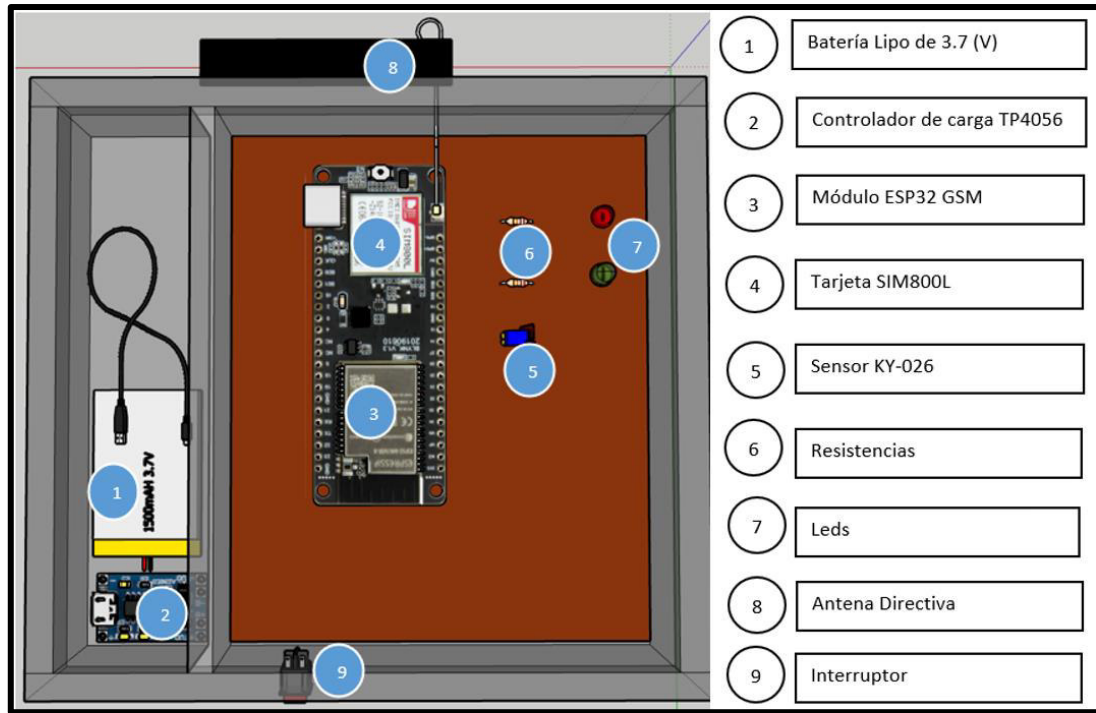


Figura 3.4 Componentes instalados dentro del prototipo

3.4 Implementación del prototipo

Elaboración de la placa electrónica

Mediante el *software Fritzing* se lleva a cabo la elaboración del circuito impreso que se implementa en la placa PCB. Al extraer el archivo de la Figura 3.5, se determina la ubicación y camino para cada componente dentro del circuito, así como las medidas requeridas que debe tener la placa.

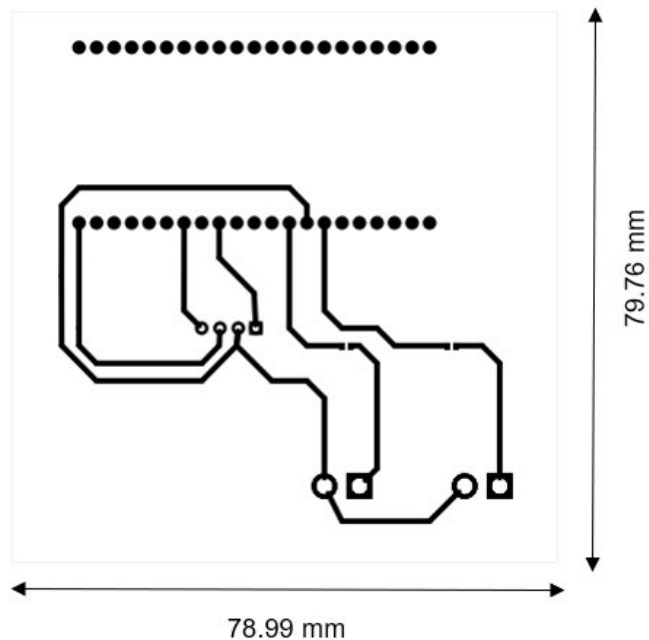


Figura 3.5 Pistas del circuito para la placa

Todos estos materiales que se encuentran en la Tabla 3.9 ayudaron a la creación del prototipo mediante los siguientes pasos.

Tabla 3.9 Materiales para creación del prototipo

Material
Placa de cobre
Cloruro férrico
Trozo de papel lija
Minitaladro con sus brocas
Plancha
Cinta adhesiva
Papel digital para estampado
Estaño
Grasa para soldar
Cautín
Multímetro
Impresora digital

Primero, se debe elegir la placa de cobre para realizar el respectivo corte. Una vez cortado, se inicia la limpieza de impurezas, luego se aplica el papel lija por toda la placa,

incluyendo los bordes, hasta que quede lisa y, con ayuda de alcohol y un pañuelo, retiramos los residuos de cobre. Finalmente, se obtiene una placa brillante.

Posteriormente, se realiza el proceso de estampado. Con la ayuda de la cinta adhesiva fijamos la placa con el papel digital que contiene el circuito impreso. No hay que olvidar que el estampado se realiza donde se encuentra el cobre. Una vez sujeto el papel con la placa, se procede a pasar una plancha por un tiempo estimado de 10 minutos.

Luego del estampado, se debe someter a la placa a un proceso de enfriamiento colocándola en un recipiente con agua, tomando las debidas precauciones para evitar quemaduras en la piel. Una vez que el circuito se ha enfriado, lo retiramos del agua y procedemos a retirar el papel digital.

A continuación, la placa debe colocarse dentro de un depósito que contiene cloruro férrico. Este ácido debe cubrir totalmente la baquelita y, por un tiempo determinado, debemos dejar la placa sumergida sin derramar el ácido, puesto que es muy corrosivo. La placa debe permanecer sumergida dentro del cloruro férrico hasta que lo único que quede de cobre en ella sea el estampado realizado.

Una vez que la placa está limpia, con la ayuda de un multímetro se procede a comprobar si hay continuidad en las pistas para verificar el correcto funcionamiento de la misma.

Finalmente, los componentes se ubican dentro de la placa como se muestra en la Figura 3.6. Como se puede ver, el módulo ESP32 se ubica junto a el sensor KY-026 seguido de resistencias y LEDs. Todos ellos se encontrarán conectados a una misma puesta a Tierra (GND).

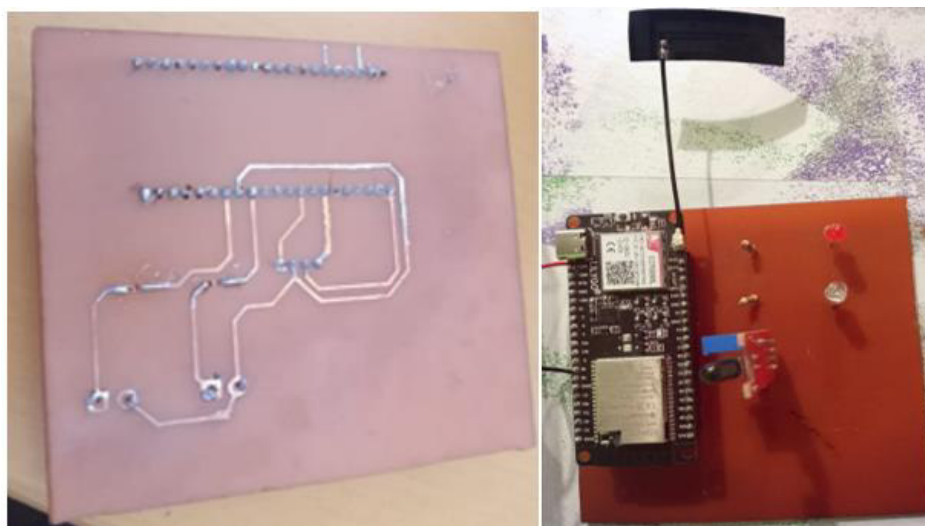


Figura 3.6 Circuito soldado con sus componentes

Fuente de energía del prototipo

La batería requiere de un controlador de carga TP4056 para recargar la energía. La alimentación de energía se encuentra conectada directamente al módulo ESP32, y este se encarga de distribuir la energía por todos los pines conectados. Por otra parte, la programación implementada dentro del módulo se encargará de que la energía no sea consumida cuando el prototipo se encuentre en estado *sleep*.

La batería de lipo se debe unir a un conector JST para energizar el prototipo, esto se da con la ayuda de dos pines I2C dentro del módulo con el fin de mantener la energía. Estos pines I2C permiten la comunicación serial entre maestro-esclavo, facilitando el intercambio de datos (SDA) y empleando una señal de reloj (SCL) [21].

Al tener un fuente de alimentación por el terminal USB-C, se aplica la misma distribución que con la batería, la única diferencia es que no ocupa los pines mencionados anteriormente ya que esta fuente se distribuye por todo el módulo.

Programación del módulo ESP32 en Arduino IDE

La programación del módulo se la realizó en la plataforma Arduino IDE, en el Anexo III se muestra de forma general cómo se lo implementa. En la Figura 3.7 se muestra la selección del módulo a utilizar dentro del IDE de Arduino.

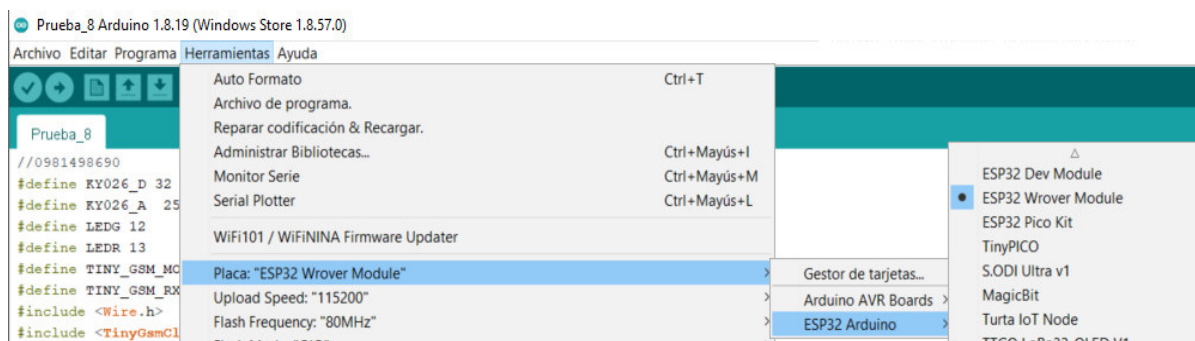


Figura 3.7 Selección de placa en Arduino IDE

Es necesario importar dos librerías para garantizar el funcionamiento adecuado del proyecto, tal como se muestra en la Figura 3.8:

```
#include <Wire.h>
#include <TinyGsmClient.h>
```

Figura 3.8 Librerías añadidas

Las librerías añadidas son:

- Librería *Wire*: Nos permite que los pines I2C puedan tener comunicación con los componentes conectados dentro de la placa. Activa la función maestro – esclavo, lo que permite el intercambio de datos y la señal de reloj.
- Librería *TinyGsmClient*: Contiene funciones para establecer una comunicación GSM por medio del módulo ESP32. Solo permite enviar SMS a un usuario definido, sin embargo, no permite recibirlos. Ocupa un espacio de memoria corto y es compatible con el módulo SIM800L.

Es necesario definir el modelo SIM que incorpora el módulo en nuestro caso es el SIM800L. También, por medio de esta librería se establece el *buffer* en 1 (Kb) que ocupa el módulo para el almacenamiento de información, como se muestra en la Figura 3.9.

```
#define TINY_GSM_MODEM_SIM800
#define TINY_GSM_RX_BUFFER 1024
```

Figura 3.9 Declaración de espacio y tipo de tarjeta SIM

Para establecer una comunicación entre el módulo ESP32 y el usuario, se declara una variable y se define el número de destino como se muestran en la Figura 3.10. La variable de tipo *char* contiene la contraseña o número PIN necesario para que el módulo pueda acceder al microchip integrado. Este número puede ser visualizado al adquirir el microchip en un punto de venta. Para contactar al usuario se asigna el número telefónico de destino, este contiene el prefijo telefónico del país seguido del número celular al que se desea notificar.

```
const char simPIN[] = "1111";
#define SMS_TARGET "+593959982014"
```

Figura 3.10 Variables para la comunicación GSM

Luego, se definen los pines que se comunican con los componentes electrónicos, el sensor y los LED ocupan la función *define* seguida de un nombre y por último el pin, como se muestra en la Figura 3.11.

```

#define KY026_D 32
#define KY026_A 25
#define LEDG 12
#define LEDR 13

```

Figura 3.11 Declaración de pines para el sensor y leds

En la Figura 3.12 muestra cómo se definen los pines pertenecientes al módulo ESP32, que permiten la comunicación serial. Estas variables son configuradas automáticamente por las librerías añadidas.

```

#define MODEM_RST          5
#define MODEM_PWKEY       4
#define MODEM_POWER_ON    23
#define MODEM_TX           27
#define MODEM_RX           26
#define I2C_SDA            21
#define I2C_SCL            22

```

Figura 3.12 Declaración de pines ESP3

Todos los valores recibidos por el sensor son almacenados en las variables que se muestran en la Figura 3.13. El sensor posee un pin analógico y otro digital, por lo que se requieren dos variables para captar la información y otros dos para presentar esa información en el puerto serial.

```

int val;
int vala;
float sensor;
float vald;

```

Figura 3.13 Variables para guardar información del sensor

También, se definen la consola serial y el uso de comandos *Attention* (AT) como se puede visualizar en la Figura 3.14. Al unir estas variables con las librerías y los comandos AT, permite la comunicación GSM y crea los mensajes que se envían por SMS al usuario destino.

```

#define SerialMon Serial
#define SerialAT Serial1
TinyGsm modem(SerialAT);

```

Figura 3.14 Declaración para salida de información y uso de comandos AT

La función `setup()` permite establecer la velocidad de transmisión que maneja la consola. Asimismo, dentro de la función se hacen configuraciones necesarias para los pines I2C, quienes se encargan de mantener la energía del módulo al conectarse a una batería como su fuente de energía, como se muestra en la Figura 3.15.

```

void setup()
{
  SerialMon.begin(115200);
  Wire.begin(I2C_SDA, I2C_SCL);
}

```

Figura 3.15 Velocidad de la consola y pines I2C

Luego, se habilitan los pines de alimentación del módulo ESP32 que se encuentran en uso, los pines que van conectados al sensor son los únicos que se encuentran como entrada, el resto de los pines son de salida. El pin de encendido y reinicio deben encontrarse en estado *HIGH* para el paso de energía, lo que permite estar en alto cuando se inicializa el prototipo. El resto de los actuadores deben estar en estado *LOW*. Además, para la comunicación AT quien establece comandos de texto para generar un mensaje se establece a la misma velocidad asignada anteriormente a los pines de comunicación GSM, tal como se muestra en la Figura 3.16.

```

//Habilitar pines de alimentación
pinMode(MODEM_PWKEY, OUTPUT);
pinMode(MODEM_RST, OUTPUT);
pinMode(MODEM_POWER_ON, OUTPUT);
pinMode(LEDG, OUTPUT);
pinMode(LEDR, OUTPUT);
pinMode(KY026_D, INPUT);
pinMode(KY026_A, INPUT);
digitalWrite(MODEM_PWKEY, LOW);
digitalWrite(MODEM_RST, HIGH);
digitalWrite(MODEM_POWER_ON, HIGH);

SerialAT.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX);
delay(3000);

```

Figura 3.16 Habilitación de pines de alimentación

Posteriormente, la estructura inicial dentro del monitor serial, este brinda un mensaje como se muestra en la Figura 3.17. Este mensaje será enviado con el fin de advertir al usuario que el prototipo se está encendiendo o reiniciando. A continuación, se lee el número PIN añadido anteriormente en la Figura 3.10 para desbloquear el microchip y enviar una notificación al destino por comandos AT. Si el número es incorrecto, el microchip no se desbloqueará, impidiendo enviar un SMS.

```
SerialMon.println("Iniciando prototipo....");
modem.restart();

// Desbloquear la tarjeta SIM con un PIN si es necesario
if (strlen(simPIN) && modem.getSimStatus() != 3 )
{
    modem.simUnlock(simPIN);
}
}
```

Figura 3.17 Estructura de inicio en el monitor serial y lectura del PIN

La función *loop()*, contiene la definición de las variables declaradas anteriormente en la Figura 3.13. Tal como se muestra en la Figura 3.18; estas variables reciben la información enviada por el sensor y la transmiten al monitor serial. Todos estos valores tienen un retardo de 1000 (ms).

```
void loop()
{
    sensor = analogRead(KY026_A);
    vala = analogRead(KY026_A);
    SerialMon.println(sensor);
    val = digitalRead(KY026_D);
    vald = digitalRead(KY026_D);
    SerialMon.println(vald);
    delay(1000);
}
```

Figura 3.18 Lectura de valores entregados por el sensor

A continuación, dentro del condicional, se procede a comparar los valores obtenidos del sensor. En caso de que el valor analógico sea menor a 200 y el valor digital igual a 1, se procede con el envío del mensaje. Se debe tomar en cuenta que si el desbloqueo de la tarjeta SIM con el número PIN no se completó como se muestra en la Figura 3.17, se

mostrará un mensaje en la consola serial de mensaje no enviado. Pero, si se completó correctamente, el LED verde pasa del estado *HIGH* a *LOW*, mientras que el LED naranja cambia de *LOW* a *HIGH*. Todos estos pasos se pueden visualizar en la Figura 3.19.

```

if (vala < 200 && vald ==1) //Valor cuando se detecta una llama
{
//Estructura del mensaje SMS a enviar al usuario configurado
String smsMessage = "Advertencia una llama se encuentra cerca del prototipo longitud de onda de la llama de "+String(vala)+" nm";

if(modem.sendSMS(SMS_TARGET, smsMessage))
{
SerialMon.println(smsMessage);
SerialMon.println("Mensaje Enviado");
digitalWrite(LEDG, LOW);
digitalWrite(LEDOR, HIGH);
}
else
{
SerialMon.println("Mensaje no enviado simPIN incorrecto");
}
delay(10000);
}

```

Figura 3.19 Envío de notificación cuando el sensor detecta una llama

Al final, cuando no se detecta ningún valor de llama. Se establece que el LED verde vuelva a su estado *HIGH* y el LED rojo regrese a *LOW*, como se muestra en la Figura 3.20.

```

else
{
digitalWrite(LEDG, HIGH);
digitalWrite(LEDOR, LOW);
}
delay(5000);
}

```

Figura 3.20 Desactivar alerta cuando no hay una llama cerca

Implementación de un chasis en el prototipo

El prototipo requiere de una protección ante factores externos que pueden dañar los componentes, por esa razón, se añadió un chasis. El chasis tiene dos compartimentos, en el primero se ubica la placa PCB, esta tiene orificios para colocar un *switch* que controle el paso de energía de la batería al circuito. También, contiene un orificio para que el LED infrarrojo pueda salir y captar las longitudes de onda de la llama, este orificio se encuentra en la parte superior del chasis. Por otro lado, la antena directiva requiere de un orificio que permita radiar información a la radiobase más cercana.

En el segundo compartimento se implementa la batería. En la Figura 3.21 se puede ver el chasis de protección del prototipo con sus respectivos compartimentos.

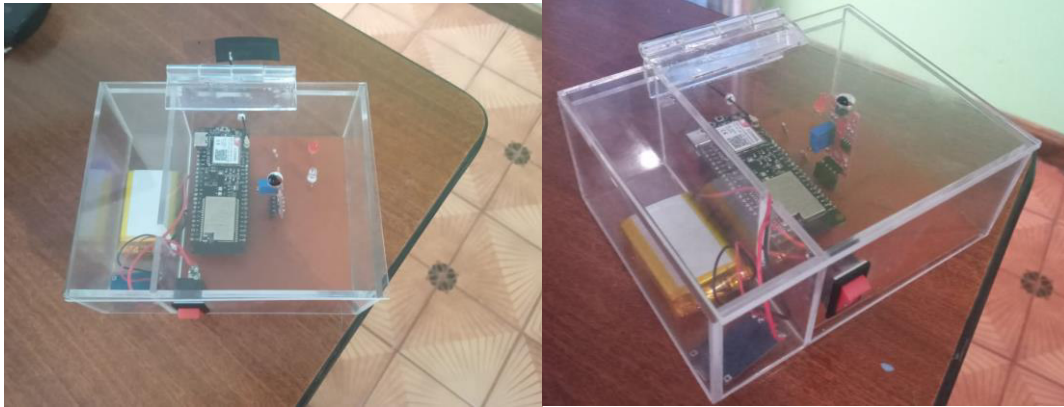


Figura 3.21 Prototipo protegido por un chasis

Costo de materiales

El costo aproximado de implementación del prototipo se muestra en la Tabla 3.10, donde se detalla una lista de los elementos requeridos en la implementación, junto con los precios en los que fueron adquiridos.

Tabla 3.10 Costo total

Material o Componente	Cantidad	Costo por unidad	Total
TTGO T-CALL ESP32	1 (u)	\$ 36	\$ 36
Sensor KY-026	1 (u)	\$ 3	\$ 3
Placa de cobre	1 (u)	\$ 1.80	\$ 1.80
Cloruro Férrico en polvo	1 (u)	\$ 1	\$ 1
LEDs	2 (u)	\$ 0.08	\$ 0.16
Cable de Estaño 1 (m)	1 (m)	\$ 1	\$ 1
Switch	1 (u)	\$ 0.28	\$ 0.28
Batería de Lipo 3.7 (V)	1 (u)	\$ 17.25	\$ 17.25
Caja de acrílico	1 (u)	\$ 25	\$ 25
Chip Claro	1 (u)	\$ 5	\$ 5
Mini taladro	1 (u)	\$ 28	\$ 28
Broca 1 (mm)	1 (u)	\$ 1	\$ 1
Cautín	1 (u)	\$ 10	\$ 10
Grasa para soldar	1 (u)	\$ 2.65	\$ 2.65
Resistencia 120 Ω	20 (u)	\$ 1	\$ 1
Impresión Digital	1 (u)	\$ 1	\$ 1
Adaptador de pila	1 (u)	\$ 1.38	\$ 1.38
		Total	\$ 135.52

3.5 Pruebas de funcionamiento

Con el fin de garantizar la correcta operación del prototipo se realizaron pruebas de funcionamiento.

Al iniciar el prototipo se enciende un LED perteneciente al sensor de llama seguido por el LED de inicio del módulo ESP32. Para comprobar el funcionamiento de la tarjeta SIM800L se tiene un LED rojo que se enciende cada cierto tiempo, esto confirma que existe conectividad con una radio base. Por último, se enciende un LED verde, que indica que no hay alerta de llama cerca del prototipo, como se puede visualizar en la Figura 3.22.

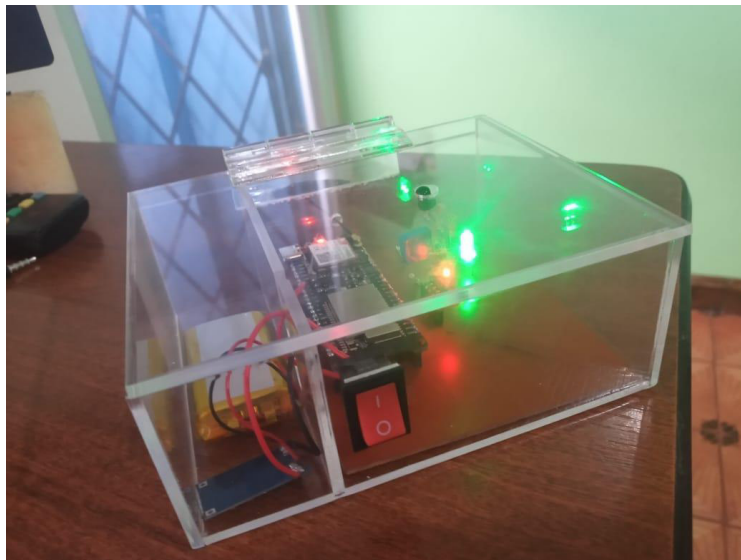


Figura 3.22 Inicio del prototipo

Utilizando un encendedor, se procede a generar una llama para comprobar que el sensor es capaz de detectarla. Dado que el encendedor genera una llama muy baja, la longitud de onda tiene un alcance muy corto por lo que es necesario acercarla al prototipo para ser detectado por el sensor. Cuando esto ocurre, automáticamente el LED verde se apaga e inicia el funcionamiento del LED rojo como se muestra en la Figura 3.23.

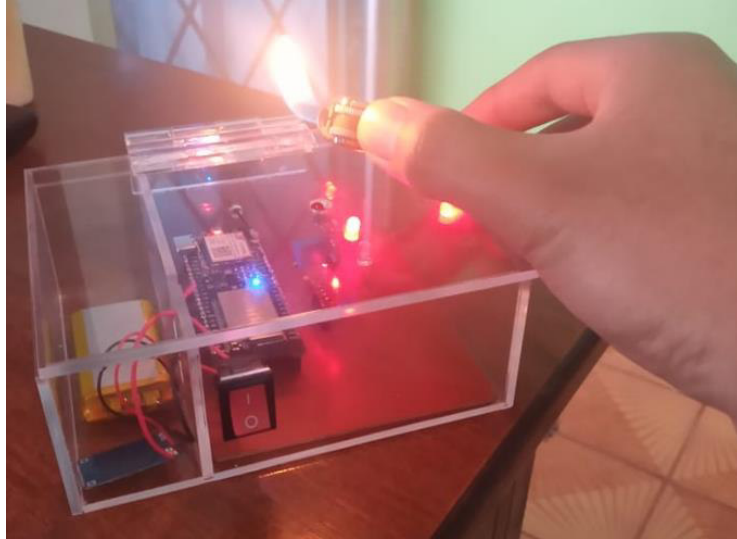


Figura 3.23 Prototipo sometido a una llama

Con la ayuda de la consola serial podemos verificar que los valores que se reflejan son los calculados en la Tabla 3.8. Al analizar la situación se verifica que cuando el prototipo no tiene una llama, presenta valores mayores a 200 y cuando se lo somete a la llama brinda datos menores a 200 como se muestra en la Figura 3.24.

```
COM8
Iniciando prototipo...
4095.00
0.00
4095.00
0.00
165.00
1.00
Advertencia una llama se encuentra cerca del prototipo
Mensaje Enviado
```

Valores analógico y digital cuando no hay una llama cerca

Valores analógico y digital cuando hay una llama cerca

Figura 3.24 Valores de la consola serial

Cuando los valores son menores a 200, es decir, que existe una llama cerca del prototipo, se genera un mensaje de advertencia y se envía una notificación al usuario. En la Figura 3.25 se refleja el envío de la notificación por SMS al usuario destino.



Figura 3.25 Notificación SMS

Una vez terminadas las pruebas de funcionamiento, se ubica el equipo en las zonas con mayor probabilidad de generarse una llama. Hay que tomar en cuenta que el microchip integrado dentro del módulo ESP32, debe contar con un plan de datos para el envío de la notificación.

4 CONCLUSIONES

- Un evento de llama es un fenómeno que puede presentarse de manera inesperada dentro del hogar o en exteriores. Por lo que instalar un prototipo detector de llama en las zonas más propensas es de mucha ayuda, ya que notifica al usuario del evento que ocurre en tiempo real para que tome las medidas respectivas.
- Al analizar los componentes para la comunicación GSM, se obtiene la mejor forma de notificar al usuario sobre los acontecimientos en tiempo real. Se tomó en cuenta costos, capacidad, confiabilidad que brinda cada componente instalada dentro del prototipo para obtener buenos resultados.
- Para determinar el tipo de batería necesaria para la operación del prototipo en las condiciones deseadas, se realizaron cálculos considerando la distribución de energía en el módulo ESP32, llegando a la conclusión de que la mejor opción es una batería de 3.7 (V) de tipo LiPo.

- Gracias a la modelización en 3D del prototipo realizada antes de su implementación, se logró una correcta distribución de los componentes dentro del chasis.
- La utilización de librerías externas dentro del código de programación facilitó el desarrollo del mismo. En el caso del módulo ESP32 que incorpora la tarjeta SIM800L, se usó la librería *TinyGSMClient*, que facilita la generación y envío de mensajes destinados al receptor.
- Dividir el chasis en compartimentos ayuda a distribuir los componentes del prototipo de forma más ordenada. El chasis es un refuerzo que protege al circuito interno ante factores externos como la lluvia.
- Dado que el envío de la notificación al usuario únicamente se realiza cuando el sensor detecta una llama cerca de él, se garantiza un uso eficiente de energía, prolongando la duración de la batería.

5 RECOMENDACIONES

- Es recomendable hacer pruebas de funcionamiento cada cierto periodo de tiempo, con el fin de comprobar que las lecturas del sensor KY-026 sean las adecuadas. Esto ayuda a evitar falsas notificaciones de llama.
- Al escoger el lugar donde se debe instalar el prototipo se tiene que calibrar el potenciómetro del sensor, esto lo vuelve más o menos sensible a longitudes de ondas perteneciente a una llama. Si se calibra el sensor de forma antihoraria lo vuelve menos sensible. Por el contrario, al calibrarlo en forma horaria, se vuelve más sensible.
- Al momento de usar el software *Fritzing* se pudo observar que las pistas del circuito eran muy delgadas, una recomendación es ampliar la anchura de las pistas para evitar al momento del estampado en la baquelita zonas no impresas.

- El prototipo requiere de una cantidad de energía necesaria para notificar al usuario y, si no cuenta con esta energía, puede que el circuito se reinicie continuamente sin enviar ninguna alerta, por lo que es recomendable tomar mediciones de voltaje de la batería al notar que los componentes se ven apagados.
- El prototipo cumple con el objetivo de informar al usuario, pero es recomendable ubicarlo en una zona con acceso a redes celulares de segunda generación ya que la tarjeta SIM800L solo trabaja con este tipo de tecnología.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] N. Kolban, «Kolban's Book on ESP32,» 2017.
- [2] Proserquisa, «Tutorial 26 Módulo sensor de llama KY-026,» 12 Abril 2018. [En línea]. Available: <https://es.scribd.com/document/376215287/Tutorial-26-Modulo-sensor-de-llama-KY-026-pdf>. [Último acceso: 13 Junio 2022].
- [3] ElectroPro, «electropro,» [En línea]. Available: <http://electropro.pe/image/data/imgProductos/140.%20M%C3%B3dulo%20GSM%20SIM800/SIM800L.pdf>. [Último acceso: 13 Junio 2022].
- [4] L. G. Zárate López, «Estudio de las características físicas y geométricas de la llama en los incendios forestales,» Universidad Politécnica de Catalunya, Barcelona, 2004.
- [5] Sawakinome, «Sawakinome,» [En línea]. Available: <https://es.sawakinome.com/articles/words/difference-between-fire-and-flame.html>. [Último acceso: 13 Junio 2022].
- [6] B. Fontal, «El espectro electromagnético y sus aplicaciones,» ULA.
- [7] I. Torres Álvarez, «SlidePlayer,» Junta de Andalucía, [En línea]. Available: <https://slideplayer.es/amp/3952142/I>. [Último acceso: 17 08 2022].
- [8] Z. Moran, «SlidePlayer,» 2015. [En línea]. Available: <https://slideplayer.es/slide/3887485/>. [Último acceso: 17 08 2022].
- [9] Botnroll, «Botnroll,» [En línea]. Available: <https://www.botnroll.com/pt/gsm/3955-esp32-c-m-dulo-gsm-gprs-sim800l.html>. [Último acceso: 13 Junio 2022].
- [10] Joy-it, «Moviltronics,» 16 Junio 2017. [En línea]. Available: <https://moviltronics.com/wp-content/uploads/2019/10/KY-026.pdf>. [Último acceso: 13 Junio 2022].
- [11] K. J. Limones Calle y F. E. Tutasí Bahamonde, «Diseño e implementación de una radio base celular de emergencia para el envío de mensajería utilizando radio definidas por software,» Universidad Politécnica Salesiana, Guayaquil, 2017.

- [12] S. A. Laura Nata, «Sistema de interferencia de señal celular en dispositivos con tecnología GSM,» Universidad Técnica de Ambato, Ambato, 2015.
- [13] Components101, «Components101,» 19 Julio 2021. [En línea]. Available: <https://components101.com/microcontrollers/arduino-uno>. [Último acceso: 16 Junio 2022].
- [14] D. Electrónicas, «Modulo sensor detector de flama,» [En línea]. Available: <https://www.didacticaselectronicas.com/index.php/sensores/flama-ollama/modulo-sensor-detector-sensores-detectores-de-llama-flama-infrarrojo-fuego-detail>. [Último acceso: 20 Junio 2022].
- [15] Aliexpress, «AliExpress,» [En línea]. Available: <https://es.aliexpress.com/item/32733583110.html>. [Último acceso: 20 Junio 2022].
- [16] F. Electronik, «Módulo Sensor de Llama KY-026,» [En línea]. Available: <https://fcelectronik.com/modulo-sensor-llama/>. [Último acceso: 20 Junio 2022].
- [17] M. Electrónica, «Lm393 Comparador De Voltaje De Precisión,» [En línea]. Available: <https://mvelectronica.com/producto/lm393-comparador-de-voltaje-de-precision>. [Último acceso: 20 Junio 2022].
- [18] TECmikro, «SENSOR INFRARROJO DETECTOR DE LLAMA,» [En línea]. Available: <https://tecmikro.com/sensores/544-sensor-infrarrojo-detector-de-llama.html>. [Último acceso: 20 Junio 2022].
- [19] E. osciloscopio, «Tutorial sobre el sensor de llama KY-026 para Arduino, ESP8266 y ESP32,» [En línea]. Available: <https://elosciloscopio.com/tutorial-sensor-de-llama-ky-026-para-arduino-esp8266-y-esp32/>. [Último acceso: 22 Junio 2022].
- [20] D. Asturias, «Longitud de Onda,» Linternas Profesionales, [En línea]. Available: <https://www.linternasprofesionales.com/longitud-de-onda>. [Último acceso: 22 Junio 2022].
- [21] J. Guerra Carmenate, «Programarfacil,» Arduino, [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/comunicacion-i2c-con-arduino/>. [Último acceso: 27 Junio 2022].

7 ANEXOS

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 11 de septiembre de 2022

De mi consideración:

Yo, **JAVIER ALEJANDRO ARMAS NAVARRETE**, en calidad de Director del Trabajo de Integración Curricular titulado **IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA POR DETECCIÓN DE LLAMA** elaborado por el estudiante **FERNANDO JAVIER TIPANTIZA PACHECO** de la carrera **TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES**, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

LINK: [Informe Turnitin - TIC - Fernando Tipantiza.pdf](#)

Atentamente,



JAVIER ALEJANDRO ARMAS NAVARRETE

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces

Anexo II.I Código QR del video de implementación y pruebas de funcionamiento del prototipo



ANEXO III: Códigos Fuente

```
#define KY026_D 32 //Pin digital conectado al sensor
#define KY026_A 25 //Pin analógico conectado al sensor
#define LEDG 12
#define LEDR 13
#define TINY_GSM_MODEM_SIM800 // El modem es SIM800L
#define TINY_GSM_RX_BUFFER 1024 // Establecer el bufer Rx en 1Kb

#include <Wire.h>
#include <TinyGsmClient.h>

float vald;
int vala;
int val;
float sensor;
const char simPIN[] = "1111";
#define SMS_TARGET "+593959982014"

//Pines para ESP32
#define MODEM_RST 5
#define MODEM_PWKEY 4
#define MODEM_POWER_ON 23
#define MODEM_TX 27
#define MODEM_RX 26
#define I2C_SDA 21 //pines de bateria 21, 22 (mantienen la energía)
#define I2C_SCL 22

#define SerialMon Serial //Establecer serie para consola SerialMon
#define SerialAT Serial1 //Establecer serie para comandos AT
TinyGsm modem(SerialAT);

void setup()
{
  SerialMon.begin(115200); //velocidad de la consola en baudios
  Wire.begin(I2C_SDA, I2C_SCL); //Mantener la energía cuando se
  ejecuta desde la batería
```

```

//Habilitar pines de alimentación
pinMode(MODEM_PWKEY, OUTPUT);
pinMode(MODEM_RST, OUTPUT);
pinMode(MODEM_POWER_ON, OUTPUT);
pinMode(LEDG, OUTPUT);
pinMode(LEDOR, OUTPUT);
pinMode(KY026_D, INPUT);
pinMode(KY026_A, INPUT);
digitalWrite(MODEM_PWKEY, LOW);
digitalWrite(MODEM_RST, HIGH);
digitalWrite(MODEM_POWER_ON, HIGH);

SerialAT.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX); //Establecer
la tasa de baudios del módulo GSM y los pines UART
delay(3000);

SerialMon.println("Iniciando prototipo....");
modem.restart();

// Desbloquear la tarjeta SIM con un PIN si es necesario
if (strlen(simPIN) && modem.getSimStatus() != 3 )
{
  modem.simUnlock(simPIN);
}
}

void loop()
{
  sensor = analogRead(KY026_A);
  vala = analogRead(KY026_A);
  SerialMon.println(sensor);
  val = digitalRead(KY026_D);
  vald = digitalRead(KY026_D);
  SerialMon.println(vald);
  delay(1000);

  if (vala < 200 && vald ==1)
  {

```

```
String smsMessage = "Advertencia una llama se encuentra cerca del prototipo  
longitud de onda de la llama de "+String(vala)+" nm";;
```

```
if(modem.sendSMS(SMS_TARGET, smsMessage))  
{  
  SerialMon.println(smsMessage);  
  SerialMon.println("Mensaje Enviado");  
  digitalWrite(LEDG, LOW);  
  digitalWrite(LEDG, HIGH);  
}  
else  
{  
  SerialMon.println("Mensaje no enviado simPIN incorrecto");  
}  
delay(10000);  
}  
else  
{  
  digitalWrite(LEDG, HIGH);  
  digitalWrite(LEDG, LOW);  
}  
delay(5000);  
}
```