

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DISEÑO Y DESARROLLO DE UN SISTEMA DE CONTROL Y
MONITOREO REMOTO DE GAS LICUADO DE PETRÓLEO (GLP)
PARA USO RESIDENCIAL.**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

JENNY LORENA FALCÓN BARRIONUEVO

jenny.falcon@epn.edu.ec

JHON SEBASTIÁN FLOREZ HENDE

jhon.florez@epn.edu.ec

DIRECTOR: ING. MONICA VINUEZA RHOR

monica.vinueza@epn.edu.ec

Quito, Julio 2019

DECLARACIÓN

Nosotros, Jenny Lorena Falcón Barrionuevo y Jhon Sebastián Florez Hende, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Jenny Falcón

Jhon Florez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por los señores Jenny Lorena Falcón Barrionuevo y Jhon Sebastián Florez Hende bajo mi supervisión.

Ing. Mónica Vinueza Rhor
DIRECTOR DE PROYECTO

ÍNDICE DE CONTENIDOS

1. INTRODUCCION

1.1	Marco Teórico	2
	El Gas Licuado de Petróleo (GLP).....	2
	Riesgos del GLP	3
	Control de Arduino mediante móvil.....	3
	Aplicaciones móviles.....	4
	Comunicación Wi-Fi.....	5

2. METODOLOGÍA.....5

3. RESULTADOS Y DISCUSIÓN.....8

3.1	Normas de Seguridad.....	8
3.2	Diagrama de Bloques.....	9
3.3	Sistema de Detección y Comunicación	10
3.4	Sistema de Programación	14
3.5	Sistema de Protección.....	19
3.6	Diagrama Circuital.....	20
3.7	Ubicación del dispositivo.....	21
3.8	Comunicación Inalámbrica y Aplicación.....	21
3.9	Construcción del prototipo.....	26
3.10	Pruebas de Funcionamiento.....	29

4. CONCLUSIONES Y RECOMENDACIONES.....29

4.1	Conclusiones.....	31
4.2	Recomendaciones.....	30

5. REFERENCIAS BIBLIOGRÁFICAS.....31

6. ANEXOS.....35

	Anexo A: Manual de Usuario.....	36
	Anexo B: Manual de Mantenimiento.....	43
	Anexo C: Fichas Técnicas de dispositivos.....	49
	Anexo D: Capacitación Bomberos.....	58

ÍNDICE DE FIGURAS

Figura 3. 1 Ejemplo de las normas	7
Figura 3. 2 Diagrama de Bloques sistema completo	8
Figura 3. 3 Ejemplo de Arduino Uno	9
Figura 3. 4 Partes de módulo ESP8266	10
Figura 3. 5 Distribución de pines ESP8266	11
Figura 3. 6 Tipos de sensores de la serie MQ	13
Figura 3. 7 Diagrama de flujo sensores de detección	13
Figura 3. 8 Programa de los sensores.	14
Figura 3. 9 Diagrama de flujo sensores <i>wifi</i> y app	15
Figura 3. 10 Programación del módulo wifi y app	16
Figura 3. 11 Diagrama para impresión	17
Figura 3. 12 Montaje del sensor de comunicación y sensores de detección	17
Figura 3. 13 Diagrama de bloques y conexiones	18
Figura 3. 14 Cargar librería desde Arduino	21
Figura 3. 15 Descarga de librerías para Arduino	21
Figura 3. 16 Crear nuevo proyecto	22
Figura 3. 17 Nombre de proyecto	22
Figura 3. 18 Seleccionar sensor	22
Figura 3. 19 Programa para ESP8266	23
Figura 3. 20 Compilación del programa correcto	24
Figura 3. 21 Sensores de detección de gas	25
Figura 3. 22 Placa Arduino Uno	25
Figura 3. 23 Sensor wifi	26
Figura 3. 24 Placa de protección	26
Figura 3. 25 Prueba de funcionamiento placa Arduino.....	27
Figura 3. 26 Verificación módulo conectividad	27
Figura 3. 27 Conectividad completa	28
Figura 3. 28 Detección de sensores	28

ÍNDICE DE TABLAS

Tabla 1. 1 Estándares de la tecnología de comunicación WiFi	5
Tabla 3. 1 Modos y valores de pines	20

RESUMEN

Inicialmente hay una introducción en la que se motiva a una lectura, desde el título en el cual se indica un problema social, en donde mediante un marco teórico se exponen conceptos básicos considerados para el desarrollo del proyecto que llame la atención del lector, siendo algunos de conocimiento general. Se maneja una metodología en la que se describe el problema planteado y la solución en base a la información adquirida.

Se incluyen principalmente los resultados y discusión en el que se refiere al proceso de desarrollo del sistema como tal, desde las normas de seguridad hasta las pruebas de funcionamiento del prototipo. Llegando a dar conclusiones a considerarse de acuerdo a la experiencia obtenida, durante el proceso de implementación se dan recomendaciones de todo el procedimiento así como de los resultados alcanzados.

Finalmente se adjuntaron anexos, como documentación de investigación; siendo este un recurso de utilidad para el progreso del proyecto, ya que de estos se tomaron conceptos y datos técnicos para la elaboración de cada sistema. Además que son respaldos del contenido expuesto en el documento.

ABSTRACT

Initially there is an introduction in which it is motivated to a reading, from the title in which a social problem is indicated, where by means of a theoretical framework basic concepts considered for the development of the project are exposed that calls the attention of the reader, being some of general knowledge. A methodology is used that describes the problem and the solution based on the information acquired.

It mainly includes the results and discussion in which it refers to the development process of the system as such, from the safety standards to the tests of operation of the prototype. In order to reach conclusions to be considered according to the experience obtained, during the implementation process recommendations are given for the whole procedure as well as for the results achieved.

Finally, annexes were attached, as research documentation; this being a useful resource for the progress of the project, since from these concepts and technical data were taken for the elaboration of each system. Also, they are backups of the content exposed in the document.

1. INTRODUCCIÓN.

Se diseña y se desarrolla un sistema de control y el monitoreo remoto de gas licuado de petróleo (GLP) para uso residencial, el cual detecte fugas de gas y la detección de monóxido de carbono (CO) que es producto de la combustión.

En el último año se reportaron 939 llamados de auxilio por fugas de gas según la página oficial de los Bomberos de Quito, lo que evidencia la problemática con este combustible en residencias el cual es muy inflamable, ya que la mayoría de personas no saben el manejo adecuado del cilindro de gas. [1]

Se registran llamados de auxilio por intoxicación de monóxido de carbono no tan común como las fugas de gas, que se dan con mayor frecuencia. El CO es producto de la combustión del gas de uso residencial, el cual puede llegar a ser mortal si no se trata a tiempo este afecta directamente al torrente sanguíneo ya que es incoloro, inodoro e insípido y es muy difícil de detectar.

Las fugas de gas son consideradas la segunda causa de incendios según fuentes oficiales, por ello recomiendan a la comunidad tomar las debidas medidas de seguridad, de evitar manipular los cilindros utilizados en hogares y comercios, estos son tan peligrosos que pueden explotar con una simple chispa.

La falta de conocimiento en cuanto a la correcta instalación del gas, también, ha causado muchos accidentes; esto puede dar origen a explosiones o asfixias. No es necesario que la concentración de gas llegue a producir fuego para que esta sea un problema, ya que muchos de los accidentes producidos se dan también por su inhalación en especial de CO.

Los dispositivos electrónicos del fabricante Arduino, han facilitado la compatibilidad entre varios sensores en este caso de GLP, CO y la conectividad inalámbrica *WiFi*, estos son los que se han usado en este proyecto. Adicionalmente, el *software* del mismo Arduino establece la comunicación entre ellos obteniendo un monitoreo de las variantes deseadas.

Para monitorear estas variables se aprovechó una aplicación móvil gratuita que pide un mínimo de datos de registro y configuración, del cual se pudo aprovechar la conectividad a redes inalámbricas para acceder a la comunicación con los módulos de detección que son los que van a recibir las señales de las variables.

Para desarrollar un sistema de control y monitoreo remoto de GLP de uso residencial, se planteó y realizó algunos propósitos, inicialmente se identificó las normas de seguridad

para la manipulación e instalación de GLP con el cual se determinó el *software* y *hardware* ya implementado en el proyecto en base a un estudio, por consiguiente se estableció la ubicación más eficiente del detector en el domicilio, se eligió una aplicación para la comunicación inalámbrica entre el detector y el dispositivo móvil, con todos estos elementos se logró construir el prototipo en función a los parámetros establecidos del diseño que finalmente permitió realizar pruebas al prototipo para validar su funcionamiento.

1.1 Marco Teórico.

▪ El Gas Licuado de Petróleo (GLP)

El consumo de GLP, denominado comúnmente como gas doméstico, del cual se deriva el monóxido de carbono (CO) como producto de combustión del gas; son los principales causantes de accidentes domésticos sea por mala instalación, manipulación o falta de prevención en cuanto a herramientas técnicas [2].

El GLP es un subproducto del petróleo, formado por una mezcla de gas butano (30 %), gas propano (68 %) y otros gases (2 %) [3].

Para que exista un posible peligro de inflamabilidad el gas debe tener contacto con el aire dentro de un cierto rango y ante alguna fuente de calor.

Las características más representativas del GLP son:

- Se puede comprimir a temperatura ambiente, esta característica permite disminuir considerablemente su volumen, logrando así almacenar una gran cantidad de gas en cilindro.
- Pesa más que el aire, en puro estado de gas pesa casi el doble que el aire. Por esto, no se eleva; sino que se desplaza al nivel del suelo.
- Como carece de olor, se le agrega un compuesto denominado mercaptano, que le da su olor característico, con el objeto de facilitar la detección de fugas, sino fuera imperceptible al sentido del olfato.
- Para que se inflame, es necesario una mezcla de GLP y aire, en presencia de una chispa, llama o fuente de calor. La mezcla debe encontrarse en cierta proporción en un rango de inflamabilidad, si la proporción de gas en el aire es menor que el límite inferior del rango, no se inflamará; lo mismo ocurre si se supera el valor máximo de la proporción.

- Rango de Inflamabilidad del GLP, mínimo: 1,8 % C Máximo: 9,5 % de GLP en el aire [2].

- **Riesgos del GLP**

Los principales son:

- Explosión, una fuga de gas puede producir una mezcla explosiva en el ambiente, con los evidentes riesgos que esto implica. Es muy crítico en el caso del GLP ya que al ser un gas más pesado que el aire se acumula en el nivel del suelo, donde pueden caer chispas, cigarrillos u otros elementos que lo inflamen.
- Asfixia, al liberarse el GLP en un espacio cerrado o con deficiente ventilación, se acumulará y desplazará al aire, lo que puede causar la asfixia por falta de oxígeno de los ocupantes del lugar.
- El olor del mercaptano procura advertir oportunamente este riesgo. Otra posible causa de asfixia radica en la combustión de GLP, durante la cual se consume el oxígeno del aire, lo que puede producir una deficiencia de este vital gas y no permite respirar adecuadamente [2].

- **Control de Arduino mediante móvil**

Algunas aplicaciones de Arduino permiten controlar los componentes adicionales como sensores, desde un *Smartphone* o Tablet.

Arduino es una herramienta de programación en *hardware* y *software*, que permite controlar desde el ordenador, mediante una plataforma de desarrollo de computación física de código abierto, diseñado sobre una placa con un sencillo microcontrolador y un entorno de comandos para interactuar tanto física como virtualmente con el dispositivo móvil.

Se conoce como una herramienta electrónica de prototipos, que interactúa entre el dispositivo móvil y la placa programadora, leyendo datos de una gran variedad de interruptores, sensores, controles de varios tipos de luces, motores y otros actuadores físicos. En los proyectos con Arduino se pueden comunicar con una aplicación que se ejecute en el móvil, el software de desarrollo es abierto al público para modificaciones y distribución del mismo [7].

- **Aplicaciones móviles.**

La mayoría de teléfonos inteligentes son móviles que soportan la funcionalidad completa de un organizador personal. Una característica importante de casi todos los teléfonos inteligentes es que permiten la instalación de aplicaciones. Estas son desarrolladas por el fabricante del dispositivo, por el operador o por un tercero.

Android, al contrario de otros sistemas operativos para dispositivos móviles, se desarrolla de forma abierta y se puede acceder tanto al código fuente como al listado de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

La tienda de aplicaciones Android conocida como *Android market* remunera a los desarrolladores el 70% del precio de su aplicación. De esta forma, promueve el entusiasmo de los usuarios y desarrolladores por crear nuevas aplicaciones destinadas a brindar facilidades cada vez mayores.

El desarrollo y programación maneja muchas opciones y no requiere conocer lenguajes complejos de programación. Un conocimiento aceptable de *Java* y poseer el kit de desarrollo de *software*, el cual es provisto por *Google* y su descarga es gratuita [8].

- **Comunicación *Wi-Fi***

El *Wi-Fi* es el mecanismo mediante el cual los dispositivos inalámbricos acceden a una red *ethernet*. Permite muchas facilidades tanto para la transmisión y recepción de datos, como por ejemplo, movilidad, fácil instalación, flexibilidad para realizar cambios en la infraestructura de la red y la fácil ampliación de nuevos usuarios a la red sin el uso de nuevos cables y conexiones físicas.

Al ser un sistema de fácil implementación y configuración es posible ser usada la tecnología *Wi-Fi* en un sin número de dispositivos electrónicos. Sin embargo, es una tecnología de comunicación que carece o tiene deficiencia en ciertas características, que son más fuertes en redes cableadas, como por ejemplo, es menos segura, su alcance es limitado, solo dispone de ciertas frecuencias y velocidades de funcionamiento.

Como toda tecnología de comunicación se encuentra bajo estándares ya establecidos y regulados en este caso por la *Institute of Electrical and Electronics Engineers* (IEEE) y están distribuidos como se muestra en la *tabla 1*.

Tabla 1 Estándares de la tecnología de comunicación Wi-Fi [6]

Normas (capa física y de acceso al medio)	Velocidad transmisión máxima (Mbps)	Throughput máximo típico (Mbps)	Numero máximo de redes colocalizadas	Banda de frecuencia	Radio de cobertura típico (interior)	Radio de cobertura típico (exterior)
IEEE 802.11a/h	54 Mbps	22 Mbps	14 (5.7 GHz)	5 GHz	85 m	185 m
IEEE 802.11b	11 Mbps	6 Mbps	3	2.4 GHz	50 m	140 m
IEEE 802.11g	54 Mbps	22 Mbps	3	2.4 GHz	65 m	150 m
IEEE 802.11n (40 MHz)*	>300 Mbps	>100 Mbps	1 (2.4 GHz) 7 (5.7 GHz)	5 GHz	120 m	300 m
IEEE 802.11n (20 MHz)*	144 Mbps	74 Mbps	3 (2.4 GHz) 14 (5.7 GHz)	2.4 GHz y 5 GHz	120 m	300 m

2. METODOLOGÍA

Debido a la naturaleza del proyecto, en el primer paso se determinó el adecuado uso y normas para la manipulación del cilindro de gas. Además, se identificaron las causas por las que existen accidentes producto de las fugas de GLP en hogares y locales comerciales, por otro lado, se conoció los peligros que intensifican la probabilidad que una fuga provoque un accidente.

El proyecto se centró en la implementación de los dispositivos que analizan el ambiente siendo estos sensores de la familia MQ, que están en busca de fugas de GLP o CO, previamente programados en el *software* de Arduino; estos sensores que han sido calificados para el desarrollo del proyecto en el sistema de detección. Para el *hardware* necesario se incluyó una aplicación móvil que envía una alarma cuando haya gas en el aire de la cocina o lugar en el que exista un cilindro.

Un punto importante, al realizar la implementación del *hardware* y *software* de los dispositivos, se realizaron estudios y análisis de cada una de las características de los sensores y el comportamiento de estos gases para identificar cual será el lugar de la instalación en el cual se aproveche al máximo las propiedades de detección, debido a que es crítico el tiempo de respuesta.

En el bloque de comunicación que es el medio por el cual el usuario puede recibir las notificaciones a su dispositivo móvil, por medio de la interfaz *WiFi*. Se puede tener una alerta sin importar la distancia a la que se encuentre el usuario del prototipo.

Una vez completados los subsistemas y habiendo comprobado su funcionamiento, se tomó en cuenta futuros cambios por si necesita alguna modificación (otros umbrales de

alarma, otras localizaciones, comunicaciones etc.) o mejora, sin que esta afecte al resto de elementos que se mantengan en el sistema.

Una vez montado el sistema en su totalidad, a los módulos que hacen de emisores y receptores se les realizó pruebas para controlar y mejorar su comportamiento, de esa manera se validó el funcionamiento según especificaciones requeridas del proyecto. Se centra en el ajuste de la comunicación inalámbrica en este caso la comunicación por *WiFi* con el sistema de detección.

3. RESULTADOS Y DISCUSIÓN

3.1 Normas de Seguridad

Para identificar las normas de seguridad para la manipulación e instalación del cilindro de gas, el Cuerpo de Bomberos de Quito ayudó con información de las causas, efectos y manejo.

Las normas de seguridad que recomiendan a la ciudadanía son conceptos básicos acerca de los incendios, prevención, consejos y uso de extintores.

Consejos Útiles:

- No aceptar cilindros en mal estado.
- Las válvulas del cilindro deben estar bien selladas.
- Mantener los cilindros bajo protección.
- Tener mucho cuidado al cambiar el regulador del cilindro.
- No colocar los cilindros en lugares cercanos al calor.
- Colocar los cilindros en piso firme y ventilado.
- Para el encendido de la cocina, prenda primero el fósforo y luego la perilla.
- Si se enciende el cilindro no trate de apagarlo.

No se deben colocar los cilindros horizontalmente (acostados) ni tampoco inclinados

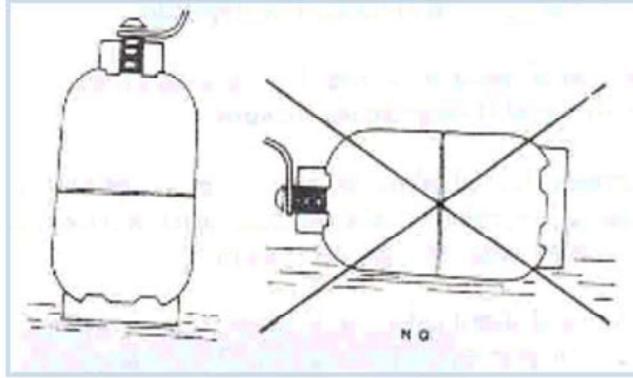


Figura 3.1 Ejemplo de las normas

3.2 Diagrama de Bloques

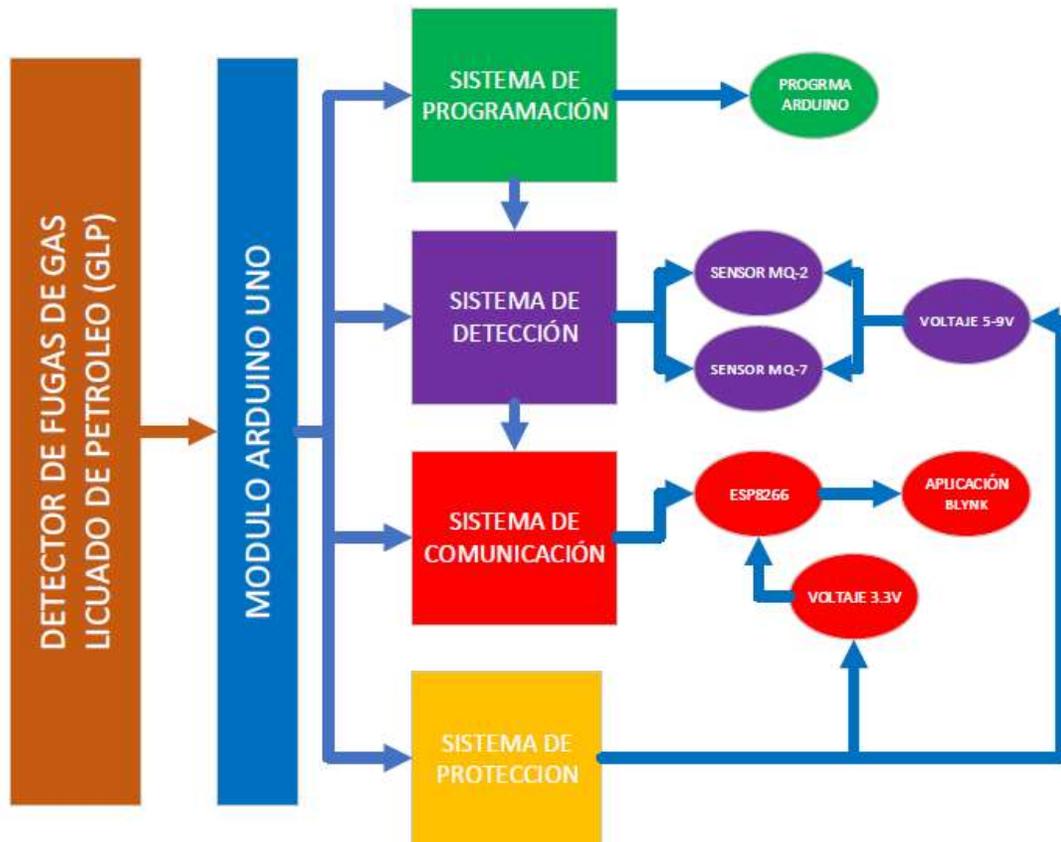


Figura 3.2 Diagrama de Bloques sistema completo

El detector de fugas de gas licuado de petróleo (GLP) es un dispositivo que básicamente depende del módulo Arduino, del que se derivan 4 subsistemas para su funcionamiento; de manera que todos y cada uno depende de esta placa base. Para empezar el sistema de programación se debe realizar primero la virtualización de la parte física con ayuda del *software* de Arduino que reflejará el resultado del sistema de detección, que es el que va a detectar el gas GLP o el CO. Estos módulos son de la familia MQ.

En el sistema de comunicación se verifica la alerta y mensaje entre el detector de GLP o CO y el dispositivo móvil. Con los resultados previos del sistema de detección. Para esto, el módulo *WiFi* ESP8266 con la programación y configuración, envía la señal a la aplicación *Blynk* instalada en el dispositivo móvil.

Finalmente el sistema de protección, tiene como función la de interconectar los diferentes sistemas, para proteger y proporcionar el voltaje necesario a el módulo ESP8266 por su sensibilidad.

3.3 Sistema de Detección y Comunicación

Para el *software* y *hardware* implementado en el proyecto se buscó sensores compatibles con Arduino Uno, estos requieren de una librería y configuración para su correcto funcionamiento.

En esta etapa del proyecto luego de haber investigado y definido los componentes a implementar, como la programación, características técnicas de los módulos y la funcionalidad; se logró la comunicación entre ellos para finalmente llegar con el mensaje de alerta a la aplicación móvil.

Arduino Uno

La placa de Arduino Uno es la mejor opción para iniciar con la programación y la experiencia electrónica, esta plataforma es la opción más robusta, usada y con mayor cantidad de documentación del fabricante.

Arduino UNO es una placa basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digitales (de los cuales 6 pueden ser usados con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión USB, conector *jack* de alimentación, terminales para conexión ICSP y un botón de reseteo. Tiene toda la electrónica necesaria para que el microcontrolador opere, simplemente hay que conectarlo a la energía por el puerto USB o con un transformador AC-DC.



Figura 3.3 Ejemplo de Arduino Uno

Módulo WiFi basado en ESP8266

Las primeras pruebas se realizaron con el sensor de conectividad, en este caso el modelo es un ESP-01, este módulo WiFi basado en ESP8266 que es un microcontrolador, previamente verificado en el *data sheet* y conexiones con la placa de Arduino, para poder receptar redes inalámbricas y acceder a internet.

Este módulo necesita de un *firmware* que es simplemente el *software* de bajo nivel que va a permitir controlar los circuitos eléctricos del mismo, por defecto, en el ESP-01 viene instalado la versión con la que se puede comunicar con el ESP8266 mediante comandos AT a través del puerto serie, que se convirtió en un estándar para la configuración y parametrización de *modems*.

Este tipo de comunicación permitirá hacer un puente entre la placa de Arduino y el módulo ESP8266, de esta manera poder conseguir que la placa se conecte a una red WiFi para enlazar la app con el móvil.

Partes de ESP-01 la placa microcontroladora

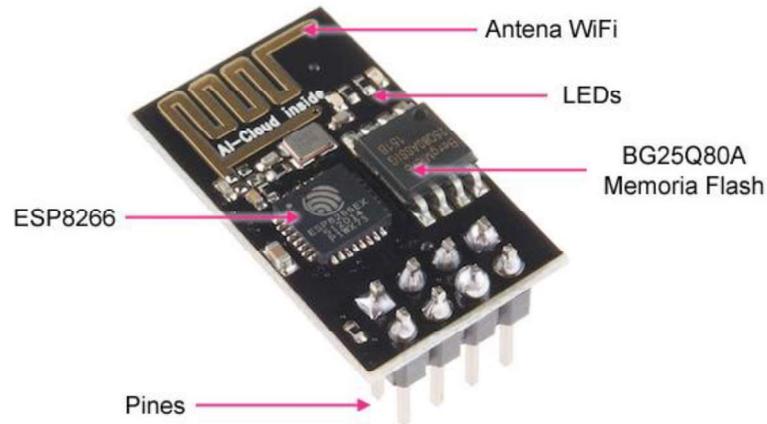


Figura 3.4 Partes de módulo ESP8266

- ESP8266 es el microcontrolador del módulo ESP-01.
- Pines donde se conecta la alimentación, sensores y transmisión del programa.
- BG25Q80A es la memoria *flash* donde residen los programas o *sketchs*. El ESP8266 no dispone de este tipo de memoria y por eso es un chip aparte.
- *LEDs* que informan si está encendido y la transmisión de datos (Tx y Rx).
- La antena *WiFi* para conectarse a una red/*Internet*.

Pines del ESP-01

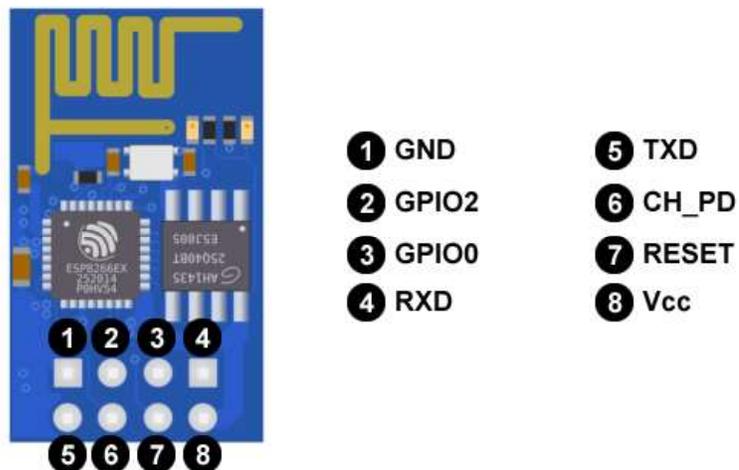


Figura 3.5 Distribución de pines ESP8266

1. GND es la toma de tierra.
2. GPIO2 es una entrada salida de propósito general, el pin digital número 2.
3. GPIO0 es una entrada salida de propósito general, el pin digital número 0.
4. RXD es el pin por donde se van a recibir los datos del puerto serie. Trabaja a 3,3 V. También se puede utilizar como pin digital GPIO: el número 3.
5. TXD es el pin por donde se van a transmitir los datos del puerto serie, trabaja a 3,3V. Se puede utilizar como pin digital GPIO: el número 1.
6. CH_PD pin para apagar y encender el ESP-01: si se pone a 0 V (LOW) se apaga, y a 3,3 V (HIGH) se enciende.
7. RESET pin para resetear el ESP-01: si se pone a 0 V (LOW) se resetea.
8. Vcc es por donde alimenta el ESP-01. Funciona a 3,3 V y admite un máximo de 3,6 V. La corriente suministrada debe ser mayor que 200 mA.

Sensores de gas MQ

Los sensores de gas son dispositivos que ayudan a detectar la presencia de algún gas específico, estos pueden configurarse o programarse para medir la concentración. Los usos más comunes son para prevenir la exposición a gases combustibles y tóxicos.

Existen varios tipos los cuales se diferencian en el tipo de tecnología que usan para detectar, entre los cuales se pueden destacar los siguientes:

- Sensores semiconductores, son sensores que trabajan de manera eficiente en ambientes húmedos, reaccionan químicamente cuando el gas hace contacto con el módulo provocando que la resistencia eléctrica decrezca.
- Sensores infrarrojos, el funcionamiento de estos sensores se basa en que el gas absorbe energía de la emisión, a una determinada longitud de onda -normalmente en el rango de los infrarrojos [4].
- Sensores ultrasónicos, este tipo de sensor es usado en la industria para detectar fugas en las tuberías ya que mediante el ultrasonido, se podrá encontrar una variación de frecuencia en el sonido producido por una fuga.

- Sensores electroquímicos, el gas se propaga hacia el sensor a través de una membrana hasta llegar al electrodo de trabajo. Cuando el gas alcanza este electrodo, se produce una reacción electroquímica: una oxidación o una reducción, según el tipo de gas. Por ejemplo, el monóxido de carbono se oxida y se forma dióxido de carbono, y el oxígeno puede reducirse a agua [5].
- Sensores catalíticos, estos sensores son los más comunes en la industria ya que son los más usados y fáciles de configurar, están compuestos por dos bobinas de platino, ambas encapsuladas en un material cerámico [4].

Su funcionamiento consiste en la oxidación del gas en la parte del catalítico, esto produce una elevación en la temperatura ocasionando una variación en la resistencia eléctrica y generando una señal característica para cada tipo de gas.

Por ser los más comunes en la industria, y los de mejor configuración para determinados tipos de gases, existen una gran variedad en el mercado que pertenecen a la serie de los MQ, ya que son de fácil adaptación a dispositivos de programación como lo son los Arduino y PICS, y se muestran en la *figura 3.6*.



Figura 3.6 Tipos de Sensores de la serie MQ [4].

3.4 Sistema de Programación

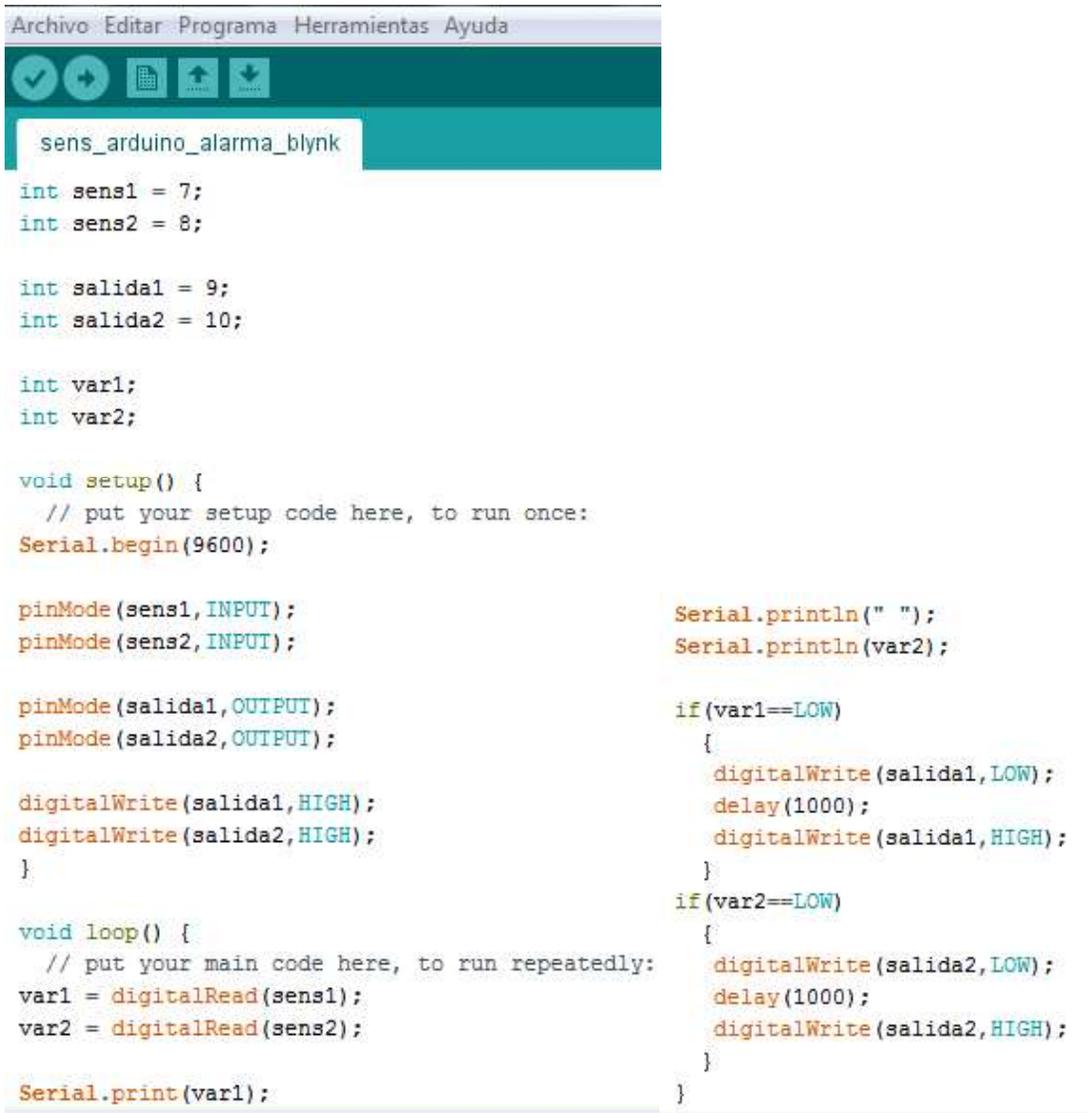
Para este sistema se partirá con la explicación de los diagramas de flujo para el sistema de detección y comunicación, los cuales tienen una programación por separado como se expone a continuación en la *figura 3.7* y *figura 3.9* respectivamente.



Figura 3.7 Diagrama de flujo sensores de detección

Se presenta el diagrama de la programación de los sensores de detección de GLP y CO, se realiza una lectura de valores digitales, es decir, de 1 o 0, el mismo que será escrito en el programa para enviar posteriormente al sistema de comunicación.

A continuación se describe el programa en sí, ya desarrollado en el *software* de Arduino las sentencias anteriormente descritas en el diagrama.

The image shows a screenshot of an IDE window titled "sens_arduino_alarma_blynk". The code is written in C++ and is designed to read two digital sensors and control two digital outputs based on their states. The code includes variable declarations for sensor pins (sens1, sens2) and output pins (salida1, salida2), as well as temporary variables (var1, var2). The setup function initializes the pins and starts the serial communication. The loop function reads the sensor values, prints them to the serial monitor, and then uses conditional logic to set the output pins to HIGH or LOW, with a 1000ms delay between each output change.

```
Archivo Editar Programa Herramientas Ayuda
sens_arduino_alarma_blynk
int sens1 = 7;
int sens2 = 8;

int salida1 = 9;
int salida2 = 10;

int var1;
int var2;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  pinMode(sens1, INPUT);
  pinMode(sens2, INPUT);

  pinMode(salida1, OUTPUT);
  pinMode(salida2, OUTPUT);

  digitalWrite(salida1, HIGH);
  digitalWrite(salida2, HIGH);
}

void loop() {
  // put your main code here, to run repeatedly:
  var1 = digitalRead(sens1);
  var2 = digitalRead(sens2);

  Serial.print(var1);

  Serial.println(" ");
  Serial.println(var2);

  if(var1==LOW)
  {
    digitalWrite(salida1, LOW);
    delay(1000);
    digitalWrite(salida1, HIGH);
  }
  if(var2==LOW)
  {
    digitalWrite(salida2, LOW);
    delay(1000);
    digitalWrite(salida2, HIGH);
  }
}
```

Figura 3.8 Programación de los sensores

Como se observa en las líneas de código se declara variables de los pines de entrada, salida y las asignadas a los valores que se imprimirán. Estos valores son los que entrarán en un bucle de sentencia dependiendo si el valor es alto o bajo; siendo este 1L o 0L por lo que son señales digitales.

Se escribirán o imprimirán en el programa de detección como resultado, para posteriormente enviar al sistemas de comunicación.

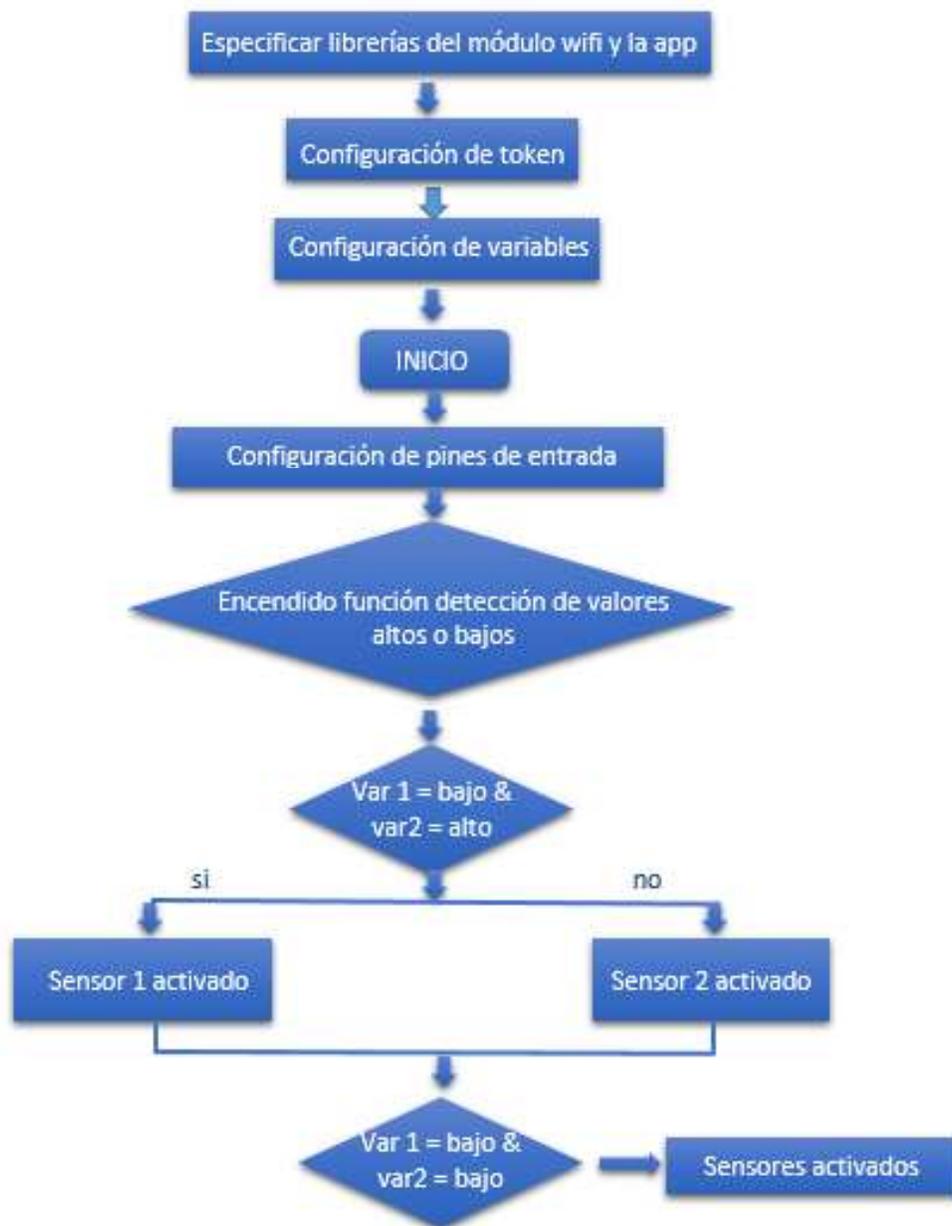


Figura 3.9 Diagrama de flujo sensores wifi y app

Este diagrama explica la compilación del programa, del sensor de *WiFi* y de la aplicación móvil, se realizó en un programa aparte ya que daba problemas con las variables de los módulos de detección, estos luego se enviarán a este sistema para las lecturas.

Los valores del sistema anterior entrarán en el bucle de la sentencia que determinará el texto que debe mostrar en la aplicación móvil en conjunto con la alarma indicando que alguno de los módulos se activó o si son ambos.

```

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "token";

char ssid[] = "usuario";
char pass[] = "clave";

int sens1 = 0;
int sens2 = 2;

int var1;
int var2;

void setup() {
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

  pinMode(sens1, INPUT_PULLUP);
  pinMode(sens2, INPUT_PULLUP);
}

void loop() {
  var1 = digitalRead(sens1);
  var2 = digitalRead(sens2);

  Serial.print(var1);
  Serial.print(" ");
  Serial.println(var2);

  if (var1 == LOW && var2 == HIGH) {
    Serial.println("sensor 1 activado");
    Blynk.notify("sensor 1 activo");
  }

  else if (var2 == LOW && var1 == HIGH) {
    Serial.println("sensor 2 activado");
    Blynk.notify("sensor 2 activo");
  }
}

```

Figura 3.10 Programación del módulo wifi y la app

En esta figura se muestra la programación del sensor de *WiFi* y la configuración de la aplicación móvil *Blynk* con sus respectivas librerías, desde los parámetros del *token* hasta las sentencias de las variables de detección, de acuerdo a estas lecturas, adquiridas anteriormente se obtendrá un mensaje y alarma en al app instalada en el móvil.

3.5 Sistema de Protección

En este sistema se adiciona una placa que fue realizada en baquelita, en donde se puso los pines usados del Arduino y unos optoacopladores para la protección del sensor *WiFi* que es muy sensible a voltajes mayores a 3.3V recomendados por el fabricante y los mensajes que pueden dejar el sistema de detección; también la facilidad de montaje y desmontaje de los sensores de detección si requiere de algún reemplazo por daño, como parte de las pruebas de funcionamiento o mantenimiento.

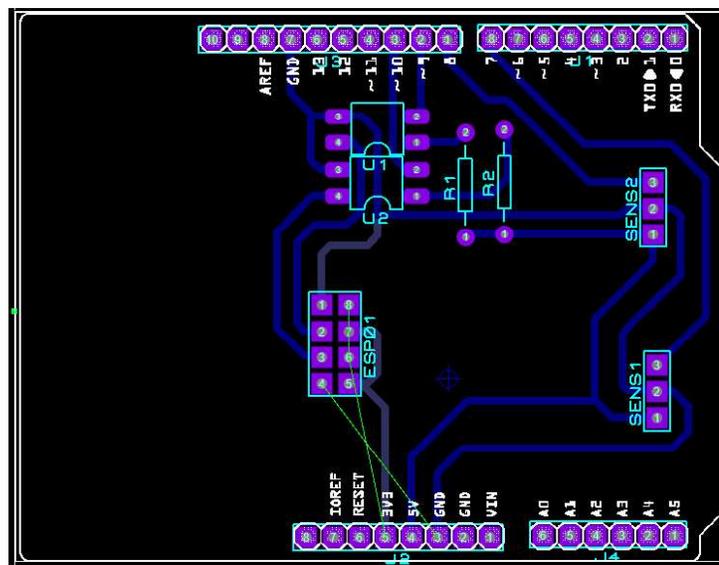


Figura 3.11 Diagrama para impresión

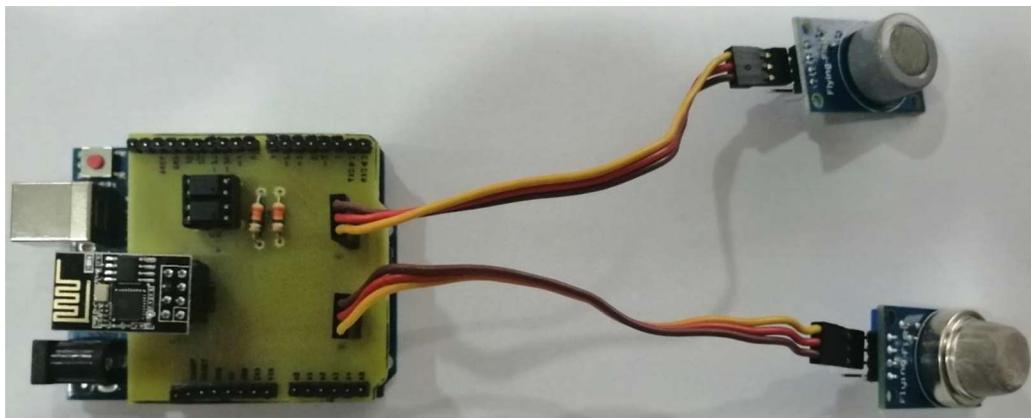


Figura 3.12 Montaje del sensor de comunicación y sensores de detección

3.6 Diagrama Circuital

Para que sea posible ser implementado físicamente el circuito del detector de fugas de gas licuado de petróleo (GLP) y CO, se realizó previamente esta simulación, que funciona de la siguiente manera; primero es necesario tener en cuenta que el módulo Arduino Uno posee una programación establecida para su funcionamiento, así se procede a indicar que la detección de los gases se realiza por medio de los sensores MQ, los cuales emiten una señal por su salida digital hacia el Arduino.

El módulo Arduino realiza su proceso con la lectura digital recibida y a su vez emite una señal digital de salida la cual pasa por los optoacopladores hacia el módulo *WIFI* ESP8266, los optoacopladores permiten que las señales no se mezclen en la comunicación, evitando un mensaje erróneo o que no llegue ningún mensaje a la aplicación *BLYNK*, su forma de evitar esto, es dejando pasar una señal a la vez, ya que puede existir la posibilidad que los dos sensores se activen al mismo tiempo.

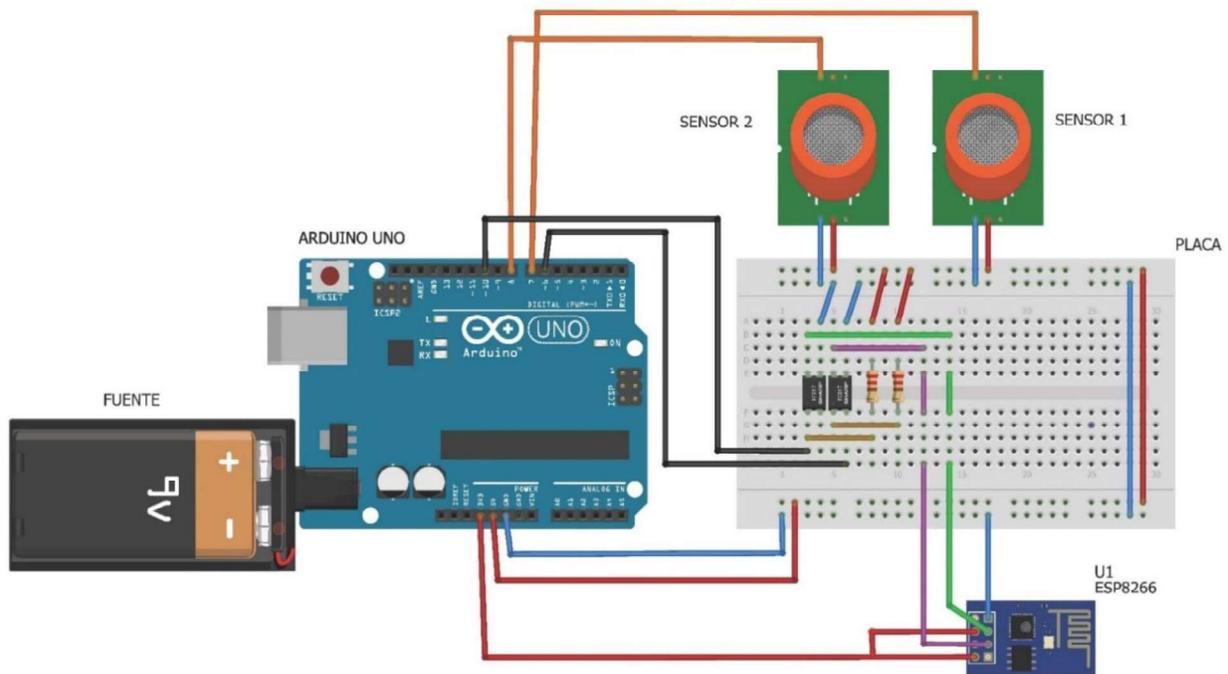


Figura 3.13 Diagrama de bloques y conexiones

3.7 Ubicación dispositivo

El estudio para determinar la ubicación más eficiente de los detectores en el domicilio, depende básicamente de la ubicación del cilindro de gas; y si es para la cocina o calefón que es lo más común, se recomienda en un lugar ventilado y si es posible afuera, a su vez que el sensor este cerca del suelo ya que este gas pesa más que el aire por ende tiende a quedarse en el suelo en una fuga; el sensor MQ2 que es de GLP. El detector de CO el sensor MQ7 cercano a válvulas o tuberías que es donde se puede concentrar la combustión de gas que también dependerá de la calibración que se le dé por su sensibilidad.

3.8 Comunicación Inalámbrica y Aplicación

En la implementación de la aplicación para la comunicación inalámbrica entre el detector y el dispositivo móvil se debe conocer algunos conceptos y también especificaciones del tipo de comunicación a usar en este caso *WiFi*.

Programar el ESP-01

Inicialmente se debe configurar el controlador de este módulo en el *software* de Arduino, lo hace desde el mismo menú del IDE del programa, una vez ya conectado a la placa. Para cargar el programa a través del puerto serie, que utiliza los pines RX y TX para transmitir los datos a la memoria *Flash*, donde se almacenará el *sketch*.

Al trabajar con Arduino, no se puede decir si se cargará un programa o si debe ejecutarlo: ya que todo esto lo hace de forma interna y automática, el microcontrolador que viene previamente programado de fábrica.

Con el ESP-01 no sucede así, se debe configurar de forma manual, donde se pueda activar con diferentes modos. Existen dos modos de operación y se configuran a través de los puertos GPIO0 y GPIO2: el modo *Flash* y el modo UART.

Modo **UART**, carga de programa en la memoria del ESP-01: cuando se carga un programa en el ESP-01 hay que encender o resetear teniendo el pin GPIO0 a nivel bajo (LOW = 0 V = GND) y el GPIO2 a nivel alto (HIGH = 3,3 V = Vcc). El ESP8266 trabaja con niveles lógicos de 3,3 V, el pin GPIO2 está por defecto a *HIGH*, ya que tiene un *pull-up* interno, por lo que se suele dejar simplemente desconectado.

Modo **Flash**, ejecución de programa en el ESP-01: una vez cargado el programa hay que ejecutarlo, esto se logra teniendo el GPIO0 y el GPIO2 a nivel alto (3,3 V). Tanto el GPIO0 como el GPIO2 están por defecto a *HIGH*, ya que ambos tienen un *pull-up* interno, por lo que se puede dejar simplemente desconectados.

Tabla 3.1 Modos y valores de pines

	GPIO0	GPIO2
Modo UART (carga programa)	LOW	HIGH (desconectado)
Modo Flash (ejecuta programa)	HIGH (desconectado)	HIGH (desconectado)

Para cargar el programa al módulo, hay que tener en cuenta:

- RX y TX se utiliza para cargar el programa. Una vez finalizada la carga se puede usar como pines de entrada y salida digitales.
- Los modos de trabajo se indican a través de los pines 0 y 2 cuando se resetea o reinicia la placa. Una vez que el modo de ejecución esté funcionando (funcionamiento normal), ya se conecta cualquier componente a estos pines. [9]

Programación del módulo ESP8266 (Wi-Fi)

Para descargar el programador Arduino se puede realizar desde la página oficial de Arduino la cual proporciona un acceso y descarga gratuita.

Se debe descargar las librerías necesarias para poder proceder con la programación en este caso necesitaremos las librerías de Arduino para el ESP8266 como se muestra en la figura 3.5 y para completar la comunicación también se deberá descargar las librerías para la aplicación *BLYNK*. [10]

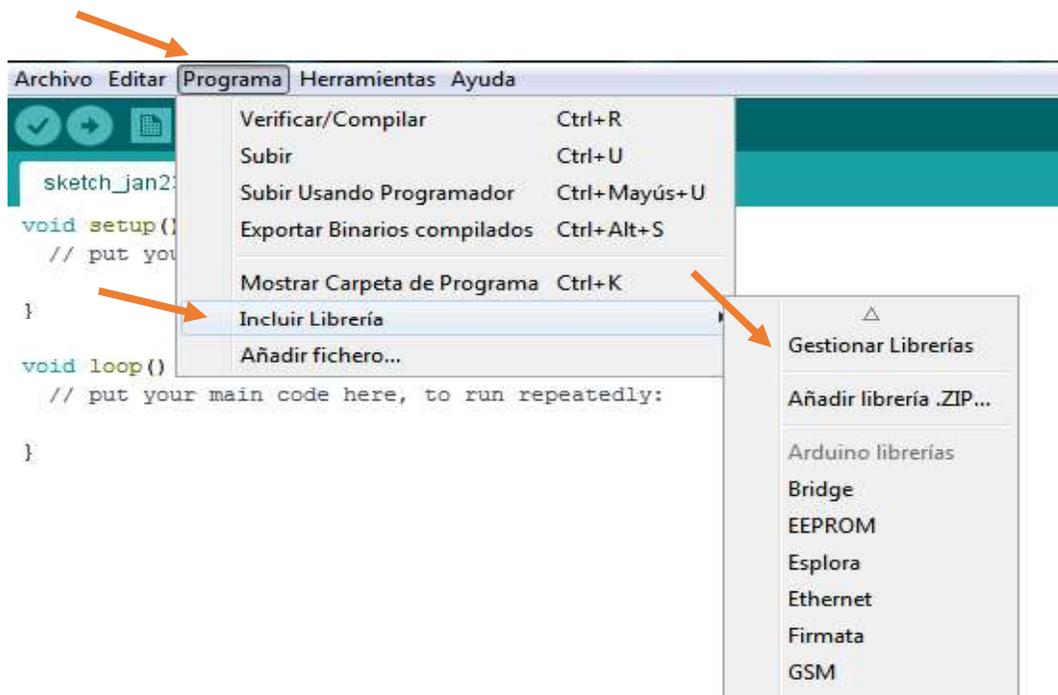


Figura 3.14 Cargar librería desde Arduino

Se abre la siguiente ventana y se escoge la librería señalada en este caso el sensor de WiFi escogido y se procede a que se instale software.

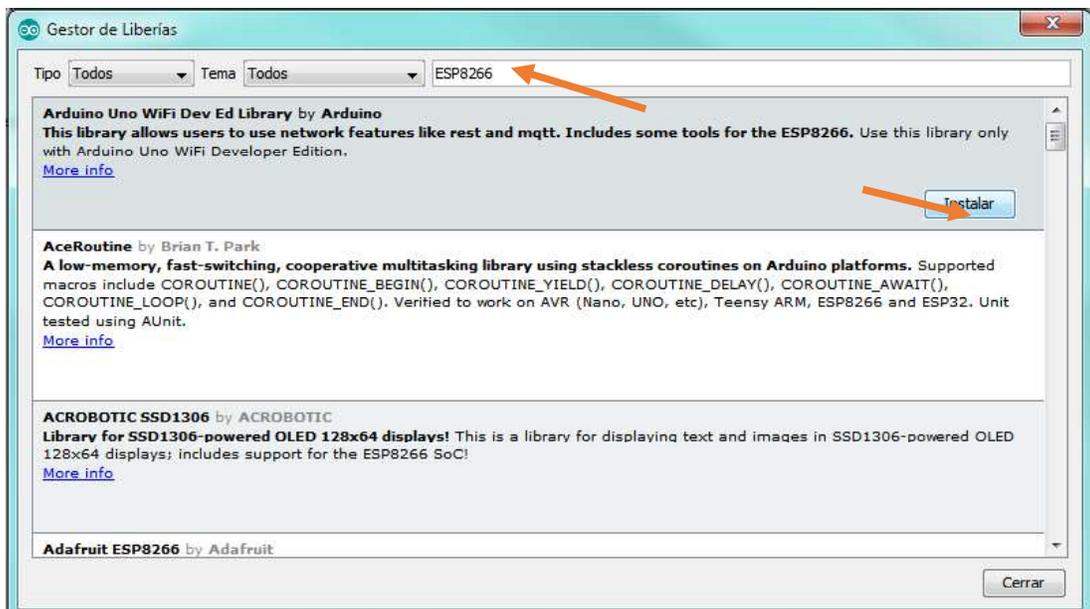


Figura 3.15 Descarga de librerías para Arduino

- **Ejecución del programa para el módulo ESP8266.**

Antes de ejecutar el programa se debe descargar la aplicación *BLYNK* en el *Smartphone*, Tablet o dispositivo móvil para la comunicación. [11]

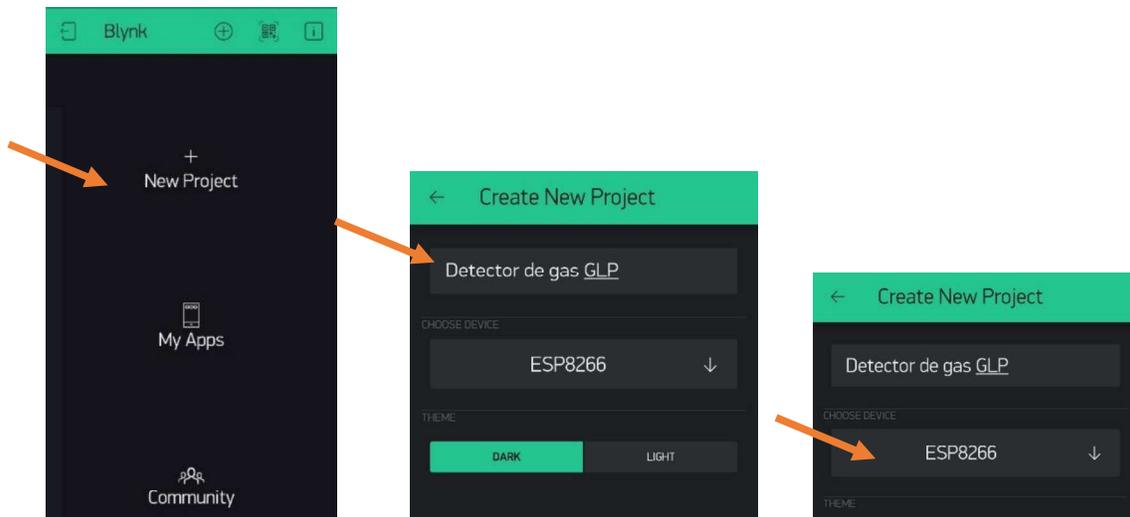


Figura 3.16 Crear nuevo proyecto

Figura 3.17 Nombre de proyecto

Figura 3.18 Seleccionar sensor

En las *figuras 3.16, 3.17 y 3.18*, se verifica la interfaz de la aplicación ya en el móvil.

En el orden que se visualiza se debe escoger un nuevo proyecto, en la siguiente opción se le debe dar un nombre al nuevo proyecto y finalmente escoger la elección del dispositivo con el que se requiere la conexión.

Se genera un TOKEN automáticamente desde el fabricante de la app, el cual será enviado al correo electrónico registrado previamente, con este TOKEN el nombre de la RED y la contraseña de la misma se procede a programar el ESP8266 en el *software* de Arduino como se especifica a continuación con las siguientes imágenes de las líneas de código, al final se verifica que ya haya conectividad del sensor con el programa así se comprueba si le llega las señales emitidas por los sensores, de esta manera se confirma si hay comunicación entre la app el sensor y la programación.

```
Archivo Editar Programa Herramientas
[Icons]
arduino_esp01_alarma_blynk
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "token";

char ssid[] = "usuario";
char pass[] = "clave";

int sens1 = 0;
int sens2 = 2;

int var1;
int var2;

void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(sens1, INPUT_PULLUP);
  pinMode(sens2, INPUT_PULLUP);
}

void loop() {
  var1 = digitalRead(sens1);
  var2 = digitalRead(sens2);

  Serial.print(var1);
  Serial.print(" ");
  Serial.println(var2);

  if (var1 == LOW && var2 == HIGH) {
    Serial.println("sensor 1 activado");
    Blynk.notify("sensor 1 activo");
  }

  else if (var2 == LOW && var1 == HIGH) {
    Serial.println("sensor 2 activado");
    Blynk.notify("sensor 2 activo");
  }
}
```

Figura 3.19 Programa para ESP8266

A continuación esta imagen donde se compila el programa con los datos sugeridos por el aplicativo y se comprueba al 100% que están bien las líneas de código.

La comunicación de la aplicación móvil con el Arduino a través del sensor móvil también es realizada sin problema ya que esta es la principal función del proyecto para recibir la alerta y también muestra un texto con el sensor que se activado que podrá a su vez ser editado como el usuario desee.

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "5837b3710ea9464d82663696b8086efe";

char ssid[] = "Tvcable_FLOREZ";
char pass[] = "1750443887";

int sens1 = 0;
int sens2 = 2;

int var1;
int var2;

void setup() {
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

  pinMode(sens1, INPUT_PULLUP);
  pinMode(sens2, INPUT_PULLUP);
}

Subido
Archiving built core (caching) in: C:\Users\JHON\AppData\Local\Temp\arduino_cache_417664\
El Sketch usa 319324 bytes (63%) del espacio de almacenamiento de programa. El máximo es
Las variables Globales usan 29840 bytes (36%) de la memoria dinámica, dejando 52080 bytes
Uploading 323472 bytes from C:\Users\JHON\AppData\Local\Temp\arduino_build_195758\arduino
..... [ 25% ]
..... [ 50% ]
..... [ 75% ]
..... [ 100% ]
```

Figura 3.20 Compilación del programa correcta

3.9 Construcción del Prototipo

Al construir el prototipo en función a los parámetros establecidos del diseño inicial se programó los sensores para que reciban y envíe señales digitales para que puedan ser receptadas por el módulo de comunicación que a su vez enviará un uno o cero lógicos para que suene la alarma.

Materiales para construcción:

Sensores de la familia MQ para GLP el MQ-2 y para CO el MQ-7, módulos de fabricante Arduino compatibles con *hardware* y *software*.



Figura 3.21 Sensores detección de gas

La placa base Arduino Uno para armar de los dispositivos físicos que viene con su propio software de programación.

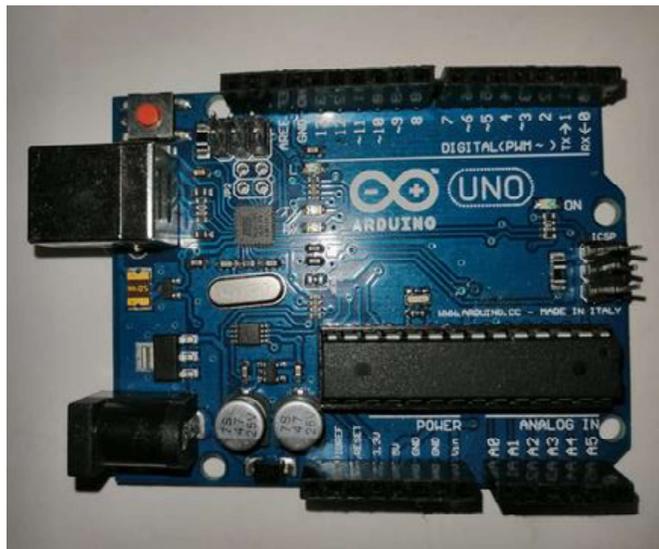


Figura 3.22 Placa Arduino Uno

En la elección del sensor de *WiFi* de igual manera se buscó el que es compatible con la placa del fabricante y con mejor alcance para la conectividad.

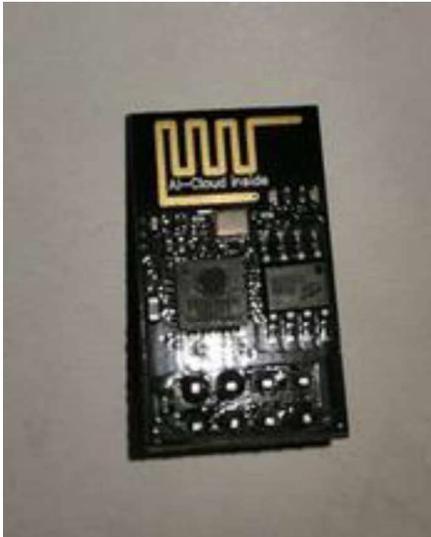


Figura 3.23 Sensor wifi

Se construyó una placa adicional para protección del sensor *WiFi* por su sensibilidad a altos voltajes y para el montaje de los sensores de detección.

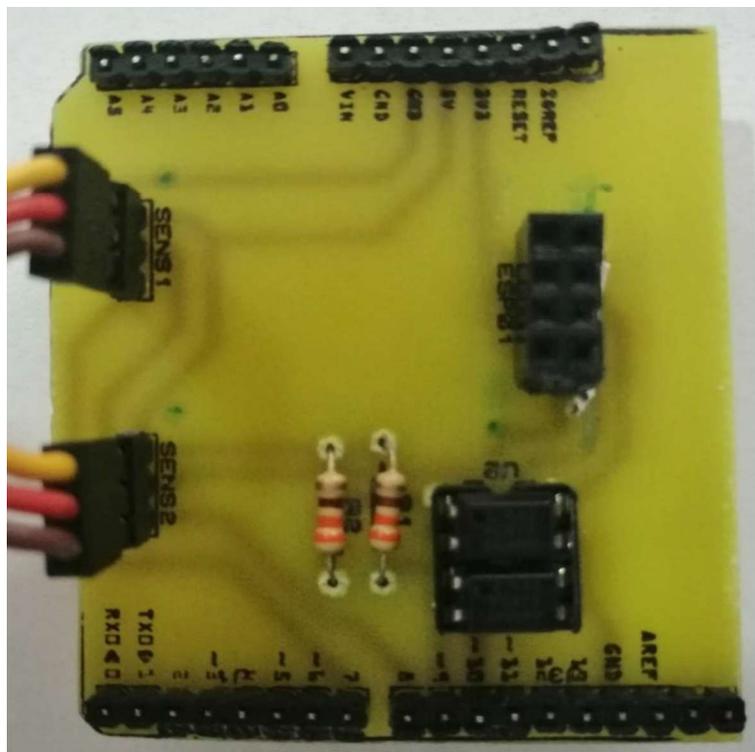


Figura 3.24 Placa de protección

3.10 Pruebas de Funcionamiento

Al realizar pruebas al prototipo para validar su funcionamiento, una vez cargado el programa el dispositivo comunica su correcta compilación con un encendido de los leds y obviamente también en el programa se observará.

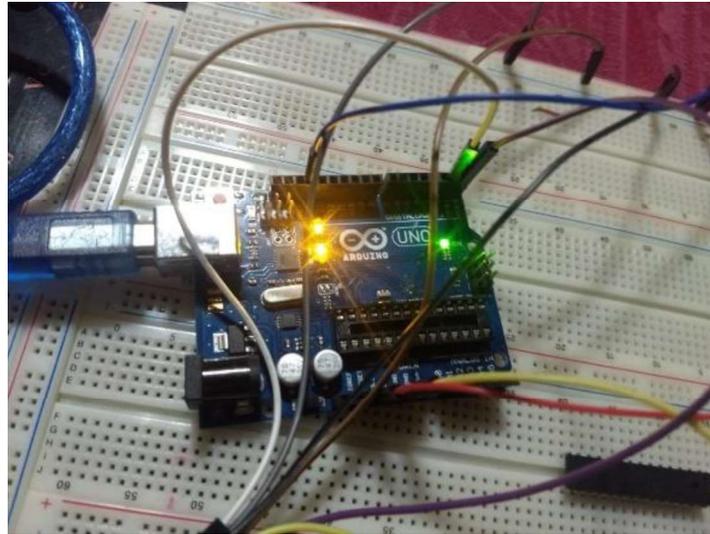


Figura 3.25 Prueba de funcionamiento placa Arduino

Se verifica comunicación con modulo WiFi, se enciende led al recibir conectividad.

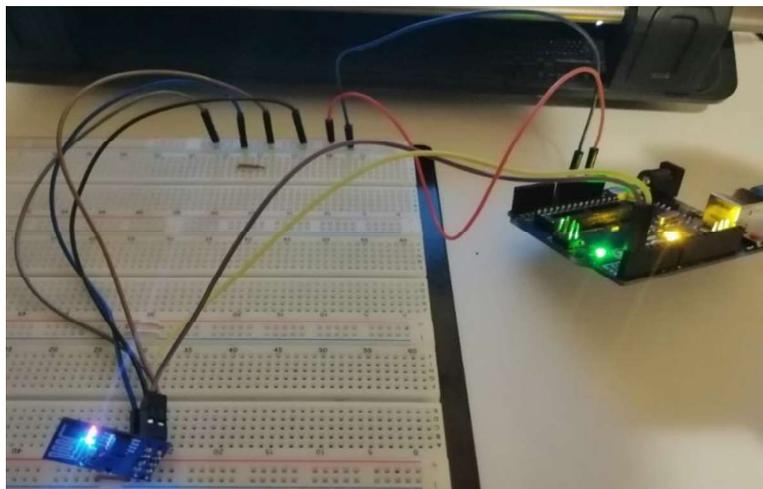


Figura 3.26 Verificación modulo conectividad

La luz azul del sensor indica que tiene conectividad con el dispositivo móvil y la app a su vez.

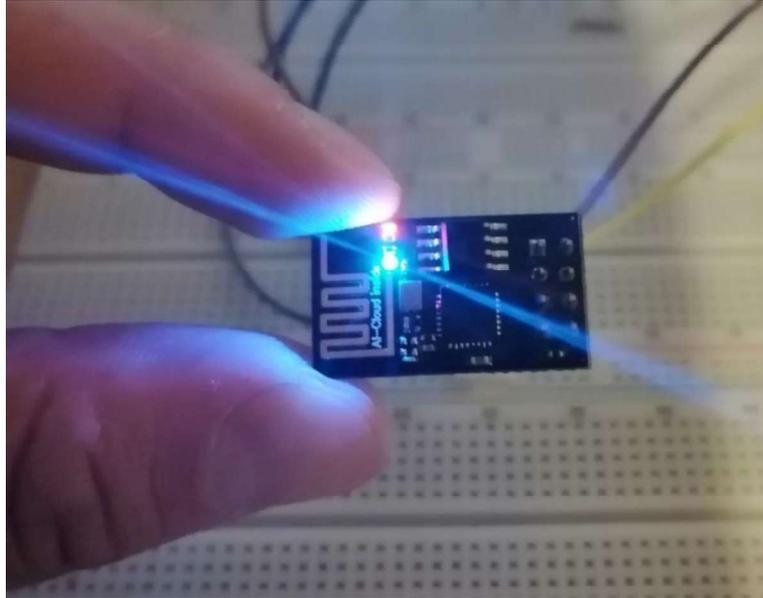


Figura 3.27 Conectividad completa

Se verifica funcionamiento del sensor de GLP con el gas de la fosforera y el sensor de CO con el humo que desprende el cautín al contacto con la pomada.



Figura 3.28 Pruebas con los sensores

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones:

- En el manejo de los cilindros de gas pocas personas se informan como instalarlo y manipularlo, es por eso que, ha habido muchos accidentes. Con acercarse al Cuerpo de Bomberos se obtuvo fácilmente una capacitación básica, adicional en internet que es un recurso ilimitado de información, de este resultado dependa la ubicación de los sensores y cuántos cilindros usen en el hogar.
- En el caso del hardware y software la elección fue fácil, debido a los conocimientos previos del fabricante Arduino y los módulos compatibles. Adicionalmente la programación para cada sensor e incluso para la comunicación se la pudo encontrar en páginas oficiales del fabricante que ayudó a precisar de acuerdo a la necesidad del proyecto.
- Para la aplicación móvil la elección se basó en la tienda en línea de *play store* que sean compatibles con Arduino. La seleccionada contempla los requerimientos del proyecto, la app escogida es muy versátil en cuanto a la configuración, porque se puede comunicar con varios dispositivos electrónicos.
- En la construcción del prototipo se usó una placa adicional de respaldo para la conexión de los módulos que facilitaron el cableado, y en las pruebas de funcionamiento los sensores detectores de los gases se pudieron probar con alcohol y el humo de cautín en ambos casos; ya que de esta manera se ahorró la manipulación de un cilindro de gas que por su tamaño es complicado realizar las pruebas en cualquier lugar, principalmente en la demostración.
- Los sensores usados en este caso MQ-2(sensor de glp) y MQ-7(sensor de CO) poseen un mecanismo en el cual se puede calibrar la sensibilidad de cada uno, esto ayuda a una mejor lectura de la fuga de gas concentrada en el ambiente.
- En el caso del módulo *WiFi* ESP8266 no se debe conectar a un voltaje superior a 3.3V ya que su característica principal es el rango de operación en ese voltaje, al ser expuesto a un voltaje superior al antes mencionado el módulo se quemará. Es

por esto que se determinó usar una placa en la cual el módulo pueda ser retirado y remplazado en caso de una falla.

- Este proyecto puede ser implementado en residencias, que ayudaría significativamente a prevenir incendios por fugas de gas o intoxicaciones por monóxido de carbono en ambientes cerrados, con una aplicación móvil que se puede instalar en cualquier dispositivo celular y que tenga acceso a una red inalámbrica.

4.2 Recomendaciones:

- Este prototipo detector de fugas de GLP es recomendable tanto para uso en el sector residencial como en el sector industrial.
- Es recomendable el uso de un dispositivo receptor que siempre esté enganchado a una red de internet ya sea inalámbrico o cableada.
- Para el uso adecuado del prototipo hay que tomar en cuenta las recomendaciones previas en cuanto al uso y manipulación del cilindro de gas, así como la instalación de los mismos.
- Revisar previamente el funcionamiento de los sensores ya que de ello dependerá el resultado deseado y el uso dado, principalmente si la variable a detectar es digital o analógica, cambiará la programación.
- Para programar los sensores se debe entender previamente las sentencias a usar, realizar muchas pruebas hasta obtener el resultado deseado con el dispositivo físico ya que no es lo mismo que una simulación.
- El diseño del prototipo podría variar durante la elaboración, por lo que es recomendable generalizar los objetivos, a que es más factible realizar cambios del mismo y no tener inconvenientes con el resultado final.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] B. Quito, «Bomberosquito,» 14 1 2018. [En línea]. Available: <https://www.bomberosquito.gob.ec/balance2018/>. [Último acceso: 23 1 2019].
- [2] Alhejandro, «SlideShare,» 8 12 2009. [En línea]. Available: <https://es.slideshare.net/Alhejandro/el-gas-y-el-monxido-de-carbono-en-viviendas>. [Último acceso: 17 01 2018].
- [3] O. Colmera, «Biblioteca USB,» 2007. [En línea]. Available: <http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/38906.pdf>. [Último acceso: 31 01 2018].
- [4] «<http://blog.330ohms.com>,» 11 07 2016. [En línea]. Available: <http://blog.330ohms.com/2016/07/11/como-funcionan-los-sensores-de-gas/>. [Último acceso: 06 06 2018].
- [5] «Industrial Scientific,» The gas Detection People, [En línea]. Available: <http://www.indsci.es/LEGACY-electrochemicalsensors/>. [Último acceso: 05 06 2018].
- [6] «CNMC blog,» 05 2010. [En línea]. Available: <https://blog.cnmc.es/2010/05/28/conceptos-basicos-de-telecos-redes-inalambricas-fijas-y-en-bandas-de-uso-comun/>. [Último acceso: 06 06 2018].
- [7] U. d. Cadiz. [En línea]. Available: http://www.uca.es/recursos/doc/Unidades/Unidad_Innovacion/Innovacion_Docente/ANEX_OS_2011_2012/22232441_310201212102.pdf. [Último acceso: 01 07 2018].
- [8] F. L. M. Gonzáles, 2011. [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/11538/Memoria.pdf>. [Último acceso: 24 06 2018].
- [9] L. d. V. Hernandez, «Programa facil,» 03 2018. [En línea]. Available: <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>. [Último acceso: 20 07 2018].
- [10] L. d. V. Hernández. [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/instalar-una-libreria-de-arduino/>. [Último acceso: 8 2 2019].
- [11] V. García, 10 12 2016. [En línea]. Available: <https://www.diarioelectronicohoy.com/blog/alarma-wifi-con-blynk>. [Último acceso: 8 2 2019].
- [12] F. Esteves, «Uva,» 2015. [En línea]. Available: <https://uvadoc.uva.es/bitstream/10324/14057/1/TFG-P-302.pdf>.
- [13] Arduino, «Arduino,» 2019. [En línea]. Available: <https://www.arduino.cc/>. [Último acceso: 8 2 2019].