

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN



INTERNET DE LAS COSAS
PROYECTO FINAL
DETECTOR DE GAS

Proyecto Final IoT

Alumnos:

Chayña Batallanes Josnick
Mamani Sucacahua, Rodrigo Alexander

Docente:

Ramos Lovón, Wilber

Arequipa - Perú

28 de diciembre de 2022

Índice

Índice de figuras	1
1. Introducción	3
1.1. Objetivos	3
1.1.1. Objetivo General	3
1.1.2. Objetivos Específicos:	3
1.2. Motivación	3
2. Marco Teórico	3
2.1. Fuentes de condensación de gas	3
2.1.1. Biomasa	3
2.1.2. Fuentes móviles de contaminación	3
2.1.3. Calidad del aire en interiores de edificios	4
2.2. Microcontroladores	4
2.2.1. Arduino	4
2.2.2. Tipos de arduinos	5
2.3. Sensor de Gas	6
2.3.1. Principio de detección de gas	6
2.3.2. Tipos de sensores	7
2.3.3. Sensores MQ en el mercado	7
2.4. Internet Of Things	8
2.4.1. Visión Orientada a “Things”	8
2.4.2. Sensores y Actuadores	9
2.4.3. Sistema de Control	9
2.4.4. Fuentes de alimentación	9
3. Metodología	10
3.1. Propuesta de Solución	10
4. Componentes	10
5. Resultados	15
6. Conclusiones	16

Listings

1. Código fuente del Sistema Detector de Gas	20
--	----

Índice de figuras

1. Concentración de CO2 en locales en partículas por millón	4
2. Arduino UNO	6
3. Tabla de características	8
4. Arduino UNO	10
5. Módulo Bluetooth H-05	11
6. Sensor de gas MQ-6	11
7. Buzzer o zumbador	11
8. LED rojo, azul, verde.	12
9. Pantalla LCD 16	12
10. Resistencias	13
11. Cables	13
12. Protoboard	14
13. Encendedor	14

14.	Conexión del Arduino con la protoboard led.	15
15.	Conexión de LCD	16
16.	Diagrama esquemático en Tinkercat de la conexión del sensor de gas.	16
17.	Diagrama esquemático en Tinkercat de la conexión de los LEDs.	17
18.	Diagrama esquemático de Modulo Bluetooth H-05.	17
19.	Diagrama esquemático del buzzer.	18
20.	Diagrama esquemático de LCD.	18
21.	Código de Conexión al Bluetooth mediante la interfaz	19
22.	Código de Gestión del sensor mediante la interfaz	20
23.	Pantalla principal: Manejo del Bluetooth	22
24.	Pantalla secundaria: Gestión del sensor	23

Detector de Gas usando Arduino

1. Introducción

La arquitectura de conexión de arduino es una de las mas óptimas en cuanto a programación y automatización de trabajos, trabaja con sensores y microcontroladores para este proyecto.

Para la construcción de dicha arquitectura debería basarse en una problemática a resolver y conectar con los componentes que ayudarían a resolver este problema.

1.1. Objetivos

1.1.1. Objetivo General

- Implementar un sistema de alarma contra fuga de gas.

1.1.2. Objetivos Específicos:

- Utilizar la tecnología ARDUINO así como otros componentes electrónicos.
- Brindar una interfaz interactiva para el manejo del sensor.

1.2. Motivación

De acuerdo al último informe revelado por el Cuerpo General de Bomberos Voluntarios del Perú, sólo en el 2019 se registró 6.751 emergencias por fuga de gas (GLP) en todo el territorio nacional. Por lo que lo hace una de las emergencias más frecuentes en todo el Perú después de emergencias médicas, accidentes vehiculares e incendios.

2. Marco Teórico

2.1. Fuentes de condensación de gas

2.1.1. Biomasa

El término biomasa hace referencia a la materia orgánica que se produce en las plantas verdes a través del proceso de fotosíntesis, así como a la originada en los procesos de transformación de la primera, considerando tanto los que se producen de forma natural, como de forma artificial. La formación o transformación de la materia orgánica ha de ser reciente, lo que excluye del concepto a los combustibles fósiles (carbón, petróleo y gas natural), cuya formación tuvo lugar hace millones de años.

- Residuos agrícolas, forestales y cultivos energéticos.
- Residuos de industrias forestales y agroalimentarias
- Residuos urbanos
- Residuos ganaderos

2.1.2. Fuentes móviles de contaminación

ivado, y destacan por ir emitiendo contaminantes en su trayectoria. Podemos citar como ejemplo más conocido el automóvil, aunque hay otros como los autobuses, camiones, barcos, u otros tipos de maquinaria móvil. En la combustión, los combustibles se transforman básicamente en vapor de agua (H₂O), dióxido de carbono (CO₂), que no son nocivos, pero tenemos otros gases que se emiten en menores cantidades pero sí son nocivos, como el monóxido de carbono (CO), óxidos de nitrógeno (NO_x), óxidos de azufre (SO_x) e hidrocarburos no quemados (HC), además de Partículas y Compuestos Orgánicos Volátiles (COV's). Por ejemplo, la emisión de gases de los vehículos es uno de los puntos importantes que miran en las ITV (Inspección Técnica de Vehículos), pues según el Reglamento General de Vehículos no se deben superar ciertos

límites legales de emisión de humos, gases contaminantes, ruidos y compatibilidad electromagnética. Así, en la combustión de los motores de los vehículos tenemos hasta más de 100 gases distintos, unos tóxicos y otros no. Los gases contaminantes más importantes que se revisan en las ITV son:

- Monóxido de Carbono (CO), Dióxido de Carbono (CO₂), Hidrocarburos (HC), Óxidos de Nitrógeno (NO_x), Dióxido de Azufre (SO₂) y Material Particulado (MP). Todos estos gases son tóxicos y/o contaminantes

2.1.3. Calidad del aire en interiores de edificios

Es una aplicación interesante a tener en cuenta porque el diseño que se pretende realizar se adapta mejor a aplicaciones pequeñas que a las industriales. Una prestigiosa Universidad de Nueva York demostró que influye en el grado de inteligencia, rendimiento o que al final de la jornada podamos sentirnos agotados mentalmente. Además, una mala calidad del aire puede suponer un mayor riesgo de padecer enfermedades y/o alergias. Así, podemos citar que el aire exterior se puede considerar como “limpio” cuando tenemos 320 ppm de CO₂, 0.1 ppm de CO, 1.5 ppm de CH₄, 0.25 ppm de N₂O, 0.001 ppm de NO₂, 0.02 ppm de ozono, 0.0002 ppm de SO₂ y 0.01 ppm de NH₃, aunque hay que matizar que estos valores aumentan en el aire urbano, por tanto estos contaminantes exteriores que en su mayor parte proceden de combustiones de los motores de los coches, calefacciones, estufas, etc. también influirán en la calidad del aire de espacios interiores. Los principales gases contaminantes que pueden estar presentes en interiores son NO₂, NO, CO, CO₂, SO₂ y COV's (Compuestos Orgánicos Volátiles), además de partículas, aerosoles biológicos y el humo del tabaco en los lugares donde se fume.

Por ejemplo en el caso del CO₂, según el RITE (Reglamento de Instalaciones Térmicas en Edificios) podemos tener una calidad el aire u otra según el IDA aplicado:

Categoría	ppm(*)
IDA 1	350
IDA 2	500
IDA 3	800
IDA 4	1.200

Figura 1: Concentración de CO₂ en locales en partículas por millón

2.2. Microcontroladores

2.2.1. Arduino

Arduino es una plataforma de hardware libre, cuyas especificaciones y diagramas esquemáticos son de acceso público ya sea bajo algún tipo de pago o de forma gratuita, está basada en una placa, es decir, un circuito impreso que se utiliza para conectar eléctricamente a través de las pistas conductoras, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos con un microcontrolador y un entorno de desarrollo que normalmente consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador, todo esto diseñado para facilitar el uso de la electrónica en proyectos multidisciplinarios. Lo mejor de la plataforma de prototipado Arduino es que puede ser lo que queramos que sea. Podría ser un sistema de riego de plantas automático, un servidor Web o incluso un piloto automático de cuadricóptero. Arduino es una plataforma de desarrollo de microcontrolador unida a un lenguaje de programación intuitivo basado en C/C++ que permite desarrollar utilizando el entorno de desarrollo integrado (Integrated development environment, IDE) de Arduino. Equipando Arduino con sensores, activadores, luces, altavoces, módulos complementarios (denominados shields, que permiten añadir funciones como Ethernet, Wifi, Xbee, GPRS, . . .), y otros circuitos integrados, es posible convertir la plataforma en un “cerebro” programable para casi cualquier sistema de control.

- Descargar e Instalar el IDE de Arduino: debe instalarse el software en el ordenador. Podemos descargar la versión más reciente del IDE en la web oficial de Arduino en la pestaña Download. En el presente proyecto se trabaja con la versión Arduino 1.6.4, aunque en enero de 2016 ya existía la versión 1.6.7. Cuando se complete la descarga, descomprimirla. Dentro se encontrará el IDE de Ar-

duino. Ya existen versiones nuevas del IDE para Windows disponibles como instalador que se pueden descargar y ejecutar, en lugar de descargar el archivo ZIP.

- Ejecutar el IDE y conectar con la Arduino: Una vez tengamos el IDE descargado y listo para ejecutar, podemos conectar la Arduino al ordenador mediante USB. Los equipos Mac y Linux “casi” instalan los controladores de forma automática, pero si usamos una Arduino moderna en un ordenador con sistema operativo Windows, es probable que se necesite instalar controladores. Si no empleamos un ordenador Windows que requiera la instalación de controladores, podemos saltarnos las siguientes instrucciones. Si hemos instalado el IDE utilizando el instalador de Windows, estos pasos ya se han llevado a cabo. Si descargamos el archivo ZIP en el equipo e instalamos sus controladores necesarios.

2.2.2. Tipos de arduinos

Actualmente, igual que pasa con Linux, hay multitud de microcontroladores Arduino en el mercado, con características generales comunes pero a la vez cada uno pensado para un público en concreto o para una serie de tareas específicas. Existe una gran variedad de modelos oficiales, no oficiales y compatibles, por lo que es normal que la gente no sepa diferenciar con exactitud las características concretas de cada uno de ellos. En julio de 2014, ya existían más de 16 modelos oficiales de Arduino y sumando los modelos no oficiales y compatibles puede haber cientos de microcontroladores Arduino. El tipo de proyecto que se quiera realizar condiciona la elección de la placa Arduino, por ejemplo:

- Según el número de pines analógicos y digitales necesarios (normales y de tipo PWM para simular una salida analógica)
- Según el tamaño de código (nº de constantes y variables, que requerirán una mayor memoria flash para su almacenamiento)
- Según la RAM del microcontrolador, que afecta a la agilidad de procesamiento de Arduino
- Por último, el voltaje a nivel electrónico en muchos casos no es importante, y además con el uso de relés las placas podrían trabajar con periféricos a 230 V en alterna, aunque si se prescinde de fuente de alimentación externa debe tenerse en cuenta el voltaje que la placa puede manejar. NOTA: No hay que confundir el voltaje de trabajo del microcontrolador con el voltaje de funcionamiento de los elementos periféricos a la placa.

Arduino UNO es la primera placa Arduino que salió al mercado y la más extendida. Prácticamente todas las características de esta placa estarán implementadas en las demás placas Arduino. Es una placa robusta y adecuada para empezar con la electrónica y codificación de proyectos en Arduino. Sus principales características son:

- Microprocesador ATmega328P de 8 bits a 16 MHz que funciona a 5 V. Tiene una memoria flash de 32 KB (0.5 KB la usa el gestor de arranque), una SRAM de 2 KB y una EEPROM de 1 KB. NOTA: Si se aplican más de 500 mA al puerto USB se rompe automáticamente el fusible de conexión.
- 14 pines digitales de Entrada/Salida (6 de los cuales se pueden usar como salidas PWM).
- 6 entradas analógicas con 10 bits de resolución (Valores de 0 a 1023).
- Conector de alimentación con conexión USB, con un botón de reinicio.
- Voltaje de entrada recomendado 7-12 V y límite 6-20 V.
- Corriente DC por E/S Pin = 20 mA
- Corriente DC en el Pin de 3.3 V = 50 mA
- Largo, Ancho y Peso: (68.6 mm, 53.4 mm), 25 gr
- Serial 0 (RX) y Serial 1 (TX). Se utiliza para recibir (RX) y transmitir datos en serie (TX) TTL
- Interrupciones externas: 2 y 3. Estos pines pueden configurarse para activar una interrupción en un flanco ascendente o descendente, o para un cambio de nivel. Ver la función `attachInterrupt(...)` para más detalles.

- PWM: 3, 5, 6, 9, 10, y 11. proporcionar una salida PWM de 8 bits con la función `analogWrite(...)`.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan la comunicación SPI utilizando la librería SPI.
- LED: 13. Hay un LED incorporado impulsado por pin digital 13.
- TWI: pin A4 o SDA y A5 o pin SCL. Comunicación TWI. Como apoyo usar la librería Wire.

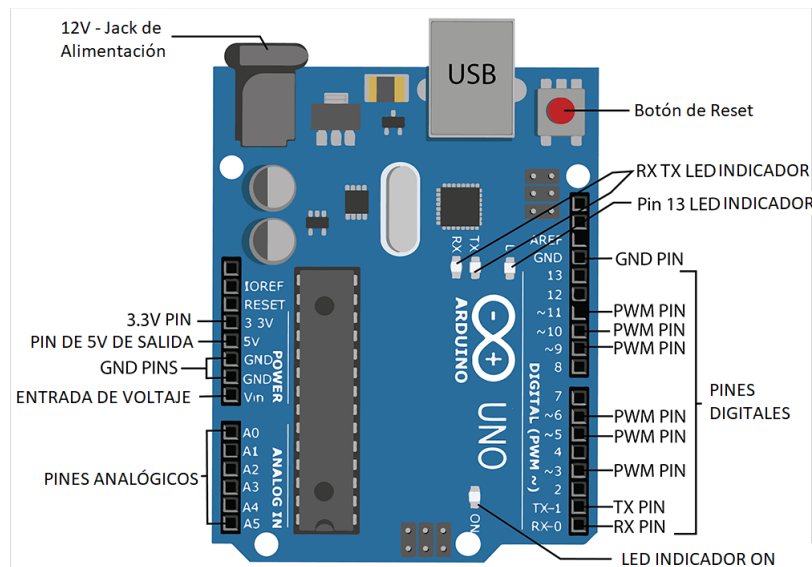


Figura 2: Arduino UNO

2.3. Sensor de Gas

Los sensores de gas de la serie MQ son candidatos atractivos en el área de detección de concentración de gas debido a su alta sensibilidad y bajo costo. Los sensores de gas semiconductor de óxido metálico (MOX), que comprenden la serie MQ y otros, tienen amplias aplicaciones en la detección y detección de concentraciones de gas debido a su alta sensibilidad y bajo costo (Baha y Dibi, 2009; Popa et al., 2019). El sensor de gas semiconductor MOX consta de un microtubo de cerámica Al_2O_3 , una capa sensible de dióxido de estaño (SnO_2) y aleaciones de níquel-cromo que sirven como bobina calentadora. Este sensor tiene 6 pines, 4 de los cuales son para señal y electrodos, mientras que los 2 restantes son para bobinas de calentamiento (Rumantri et al., 2018). El semiconductor de dióxido de estaño (SnO_2) es la porción sensible al gas del sensor (Kim et al., 2006) que tiene baja conductividad en aire limpio. El principio de funcionamiento de estos sensores se basa en la variación de su resistencia cuando entran en contacto con el gas a sensar. La magnitud de la señal de salida del sensor depende de la concentración y la naturaleza del gas y del tipo de óxido metálico utilizado para la superficie de detección del sensor (FEZARI et al., 2015). El sensor se compone de dos elementos, a saber, los elementos de calefacción y detección. Estos elementos normalmente se alimentan independientemente de la misma fuente de voltaje o de una fuente separada. El voltaje del calentador le permitirá generar el calor necesario para mantener el sensor en estado activo, mientras que el voltaje del sensor permitirá que el sensor convierta la concentración de gas detectada a un nivel de voltaje apropiado a través de la resistencia de carga conectada en serie con el elemento sensor (Kouda et al., 2018).

2.3.1. Principio de detección de gas

Los semiconductores permiten el intercambio de iones de un medio a otro. El flujo de iones induce una diferencia de potencial cuya intensidad podría medirse: tal es el principio utilizado en los sensores de gas (Nasef y Hegazy, 2004). La capa sensible en contacto con el gas en el momento de la detección impacta en la interacción fisicoquímica que conduce a las respuestas a través de pulsos eléctricos. El gas se detecta a través de dos enfoques principales, el hardware y el software, también conocidos como métodos no continuos y continuos, respectivamente. En las partes de hardware, se emplean humanos, perros y dispositivos en la

detección de fugas. Mientras que la parte continua, dividida en dos partes, el sistema de detección de fugas computarizado de tubería interna y externa.

2.3.2. Tipos de sensores

Para detección y medición de gases tóxicos:

- Sensores electroquímicos: Son los más usados para la detección de gases tóxicos como por ejemplo el monóxido de carbono, cloro y óxidos de nitrógeno. Funcionan por medio de señales de electrodos cuando se detecta el gas.
- Semiconductores de óxido de metal (o MOS): También se usan para detectar gases tóxicos (normalmente monóxido de carbono, aunque es capaz de detectar también sulfuro de hidrógeno, H₂S) y trabajan a través de una película sensible al gas, está compuesta de óxido de estaño o tungsteno. Estos sensores suelen considerarse eficientes por su capacidad para trabajar en rangos de baja humedad, además de ser capaces de detectar también gases combustibles.
- Sensores catalíticos: Representan un gran número de los dispositivos detectores de gas que se fabrican hoy en día. Se usan para detectar gases tales como hidrocarburos, funcionando a través de la oxidación catalítica. A pesar de que pueden ser calibrados para medir varios tipos de gases, el metano es el gas más común que suelen detectar. La salida de un sensor catalítico se mide mediante un dispositivo llamado puente de Wheatstone (usado comúnmente en circuitos eléctricos), que se usa para medir la resistencia eléctrica desconocida, que puede ser analizada para proporcionar la información adecuada en relación con los gases presentes
- Sensores de infrarrojos o detectores de IR: Funcionan a través de un sistema de transmisores y receptores para detectar gases combustibles (concretamente vapores de hidrocarburos). Si el gas está presente en la trayectoria óptica, se interfiere el poder de transmisión de luz entre el transmisor y el receptor. La alteración del estado de la luz determina si hay gas y además qué tipo de gas está presente. Estos sensores en lugar de depender de un puente de Wheatstone para medir la diferencia de la resistencia eléctrica, emplean una lámpara de fuente de infrarrojos y un filtro de longitud de onda

2.3.3. Sensores MQ en el mercado

Existe una gran cantidad de sensores MQ (MQ-2, MQ-3, MQ-4,...), son baratos y capaces de medir una gran cantidad de gases (por ejemplo Propano, Metano, Butano, LPG, Humo, Alcohol, Etanol, Gas natural, Hidrógeno, Monóxido de carbono, Gases inflamables, Ozono, Benceno, Sulfuro de hidrógeno, Amoníaco, Tolueno, Acetona, etc. . .). Diferentes sensores pueden medir el mismo tipo de gas (por ejemplo el alcohol lo pueden detectar los sensores MQ-2, MQ-3 y MQ-5) aunque se debe saber que cada uno detecta unos rangos distintos, tanto para este gas como para el resto de gases. Pese al abanico de modelos todos ellos se conectan y usan de la misma forma, por lo que habrá explicaciones que serán comunes en todos ellos. Son adecuados para detección de fugas de gas en casas, talleres, edificios comerciales, sistema de seguridad para detección contra incendios, monitorización de gases contaminantes, etcétera. Otras características importantes es que tienen alta sensibilidad, respuesta rápida, son capaces de detectar un amplio rango, su funcionamiento es estable, tienen larga vida y el circuito de accionamiento es simple. Pasamos a detallar algunos de ellos:

Sensor	Gases que puede detectar	Rango de detección (ppm)
MQ-2	Hidrógeno	300-5000
	LPG y Propano	200-5000
	Metano	5000-20000
	Monóxido de carbono	No especificado
	Alcohol	100-2000
MQ-3	Alcohol	25-500
	Benceno, Metano, Hexano, LPG y CO	No especificado
MQ-4	Metano	200-10000
	Hidrógeno, Monóxido de carbono, Alcohol y Humo	No especificado
MQ-5	LPG y Gas natural (butano, propano, gas ciudad)	200-10000
	Hidrógeno, Monóxido de carbono y Alcohol	No especificado
MQ-6	LPG	200-10000
	Hidrógeno, Metano, Monóxido de carbono y Alcohol	No especificado
MQ-7	Monóxido de Carbono	20-2000
	Hidrógeno, LPG, Metano y Alcohol	No especificado
MQ-8	Hidrógeno	100 - 10000
	LPG, Metano, Monóxido de carbono y Alcohol	No especificado
MQ-9	LPG	500-10000
	Monóxido de carbon	20-2000
	Metano	500-10000
MQ-131	NOx y CL2	No especificado
	O ₃	10ppb – 2 ppm
MQ-135	Amoníaco	100-300
	Alcohol	10-300
	Benceno	10-1000
	NOx, Humo y Dióxido de carbono	No especificado
MQ-136	Monóxido de carbono y Amonio	No especificado
	Sulfuro de hidrógeno	1-100
MQ-137	Monóxido de carbon	No especificado
	Amoníaco	5 - 200
MQ-138	Benceno y Alcohol	10 - 1000
	Amoníaco	10 - 3000
	Metano, Monóxido de carbono y Propano	No especificado

Figura 3: Tabla de características

2.4. Internet Of Things

En los próximos años de IoT, la conciencia del contexto une la interconexión entre el mundo físico y las entidades informáticas virtuales, e involucra metodologías de detección del entorno, comunicación en red y análisis de datos (Liu et al., 2020). El avance permite varias aplicaciones avanzadas de IoT, como sistemas de salud inteligentes, sistemas de transporte inteligentes, sistemas de energía inteligentes y edificios inteligentes. La arquitectura unificada de las redes de IoT incluye servicios de aplicaciones inteligentes basados en IoT y las redes de sensores de IoT subyacentes Percival y Walden, 2000. Además, el crecimiento mundial del mercado de IoT está impulsado por las tecnologías de redes inalámbricas y la adopción de tecnologías emergentes como las plataformas en la nube. Esta tendencia conduce a un aumento drástico en la demanda de servicios de aplicaciones y dispositivos IoT conectados.

2.4.1. Visión Orientada a “Things”

Las “things” consideradas eran elementos muy simples: Radio-Frequency IDentification (RFID). El término “Internet de las cosas” se atribuye, de hecho, a The Auto-ID Labs, una red mundial de laboratorios académicos de investigación en el campo de RFID en red y tecnologías de detección emergentes. Estas instituciones, desde su establecimiento, han sido el objetivo de diseñar el IoT, junto con EPCglobal. Su atención se ha

centrado principalmente en el desarrollo de Electronic Product Code™ (EPC) para respaldar el uso generalizado de RFID en las redes comerciales modernas de todo el mundo y crear los estándares globales impulsados por la industria para EPCglobal Network™. Estos estándares están diseñados principalmente para mejorar la visibilidad de los objetos (es decir, la trazabilidad de un objeto y el conocimiento de su estado, ubicación actual, etc.).

2.4.2. Sensores y Actuadores

Un mejor término para un sensor es un transductor. Un transductor es cualquier dispositivo físico que convierte una forma de energía en otra. Entonces, en el caso de un sensor, el transductor convierte algún fenómeno físico en un impulso eléctrico que luego puede interpretarse para determinar una lectura. Un micrófono es un sensor que toma energía vibracional (ondas de sonido) y la convierte en energía eléctrica de una manera útil para que otros componentes del sistema se correlacionen con el sonido original.

Otro tipo de transductor que se encuentra en muchos sistemas IoT es un actuador. En términos simples, un actuador opera en la dirección inversa de un sensor. Toma una entrada eléctrica y la convierte en acción física. Por ejemplo, un motor eléctrico, un sistema hidráulico y un sistema neumático son todos tipos diferentes de actuadores.

En los sistemas típicos de IoT, un sensor puede recopilar información y enrutarla a un centro de control donde se toma una decisión y se envía el comando correspondiente a un actuador en respuesta a esa entrada detectada.

Hay muchos tipos diferentes de sensores. Sensores de flujo, sensores de temperatura, sensores de voltaje, sensores de humedad, y la lista continúa. Además, hay múltiples formas de medir lo mismo. Por ejemplo, el flujo de aire puede medirse usando una pequeña hélice como la que vería en una estación meteorológica. Alternativamente, como en un vehículo que mide el aire a través del motor, el flujo de aire se mide calentando un elemento pequeño y midiendo la velocidad a la que se enfría el elemento.

2.4.3. Sistema de Control

Un dispositivo IoT puede ser más pequeño que una moneda o más grande que un refrigerador. Puede realizar una función de detección simple y enviar datos sin procesar a un centro de control. Puede combinar datos de muchos sensores, realizar análisis de datos locales y luego tomar medidas. Independientemente de la función, el entorno o la ubicación, el dispositivo IoT requiere dos componentes, un cerebro y conectividad. El “cerebro” proporciona control local (o toma de decisiones). La función del dispositivo determinará el tamaño y las capacidades del componente cerebral. Se necesita conectividad para comunicarse con el control externo. El entorno y la ubicación del dispositivo determinarán cómo se conecta. Lo más probable es que el dispositivo IoT utilice un microcontrolador como cerebro. Se puede pensar en un microcontrolador como una computadora pequeña con un núcleo de microprocesador, memoria y puertos de entrada/salida (E/S). El núcleo del microprocesador del microcontrolador es una unidad central de procesamiento. Maneja todo el procesamiento de números y la manipulación de datos locales y la toma de decisiones. La memoria incluye memoria de solo lectura (ROM) y memoria de acceso aleatorio (RAM). La ROM almacena el programa de software del microcontrolador. La RAM almacena y recibe datos al tiempo que admite el procesamiento de números. El componente final del microcontrolador, los puertos de E/S, pueden ser digitales o analógicos. Los puertos de entrada recopilan datos de sensores. Mientras que las salidas admiten cualquier actuación necesaria o control local en el dispositivo IoT. Por lo general, los microcontroladores controlan varios dispositivos o subsistemas dentro de las aplicaciones integradas. Al integrar el microprocesador, la memoria y las entradas/salidas, los microcontroladores reducen los costos y facilitan el desarrollo. Esto hace que sea más asequible y menos complicado controlar muchos dispositivos IoT.

2.4.4. Fuentes de alimentación

- **Electricidad de red:** Para muchas aplicaciones, es posible que solo se necesite enchufar el dispositivo conectado. Las aplicaciones de automatización del hogar, como las bombillas de luz conectadas, pueden extraer energía del cableado existente en el hogar. Las aplicaciones industriales suelen ser dispositivos de ubicación fija, cableados a una red eléctrica.

- **Baterías:** Las baterías, que se encuentran en todo, desde dispositivos portátiles hasta herramientas, son una solución común para los dispositivos IoT portátiles. Su aplicación determinará si tiene sentido usar baterías recargables o de un solo uso (primarias).
- **Harvesting:** Se están volviendo populares varios enfoques para convertir la energía de las formas ambientales en electricidad. Los paneles solares, que convierten la luz en electricidad, son las más comunes. También es posible convertir el flujo de aire y fluido, el calor, el movimiento, la radiofrecuencia y la energía química en electricidad.

3. Metodología

3.1. Propuesta de Solución

Presentar un sistema el cual detecte la fuga de gas, en caso de fuga se sonará una alarma de manera continua y se prenderá un LED de color verde. Además de mostrar el estado en una pantalla LCD, se notificará a través de una notificación al celular la cual es mostrada por la aplicación conectada al prototipo.

4. Componentes

- Hardware
 - Arduino Uno

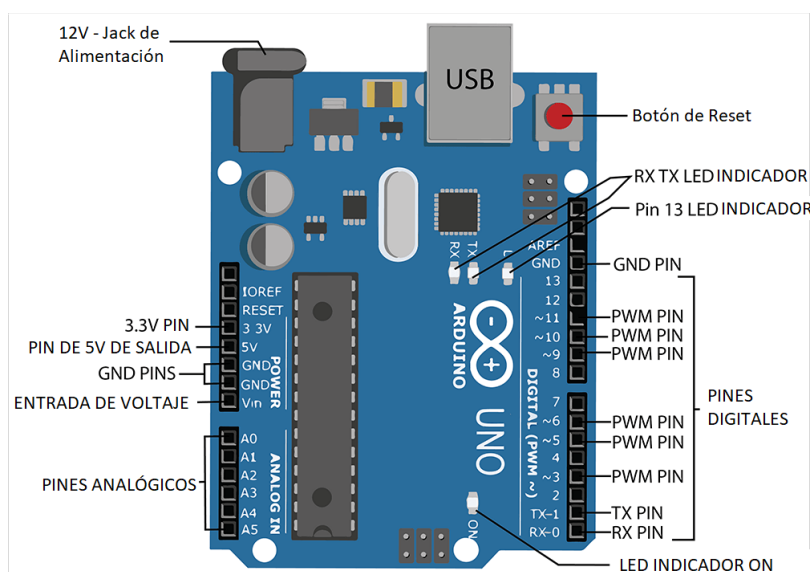


Figura 4: Arduino UNO

- Modulo Bluetooth H-05

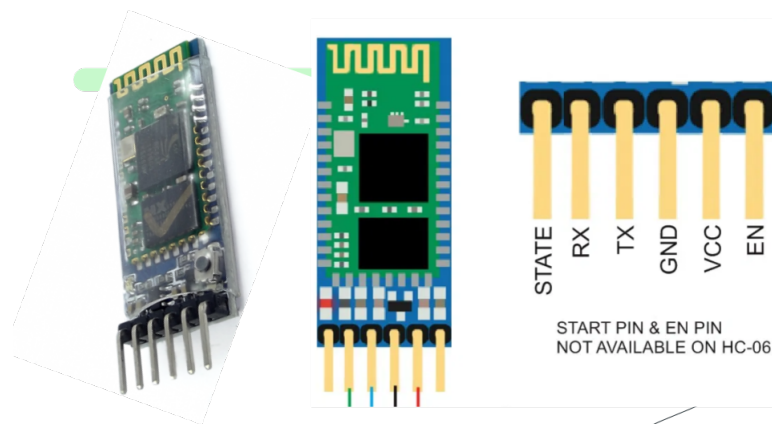


Figura 5: Modulo Bluetooth H-05

- Sensor de gas MQ-6

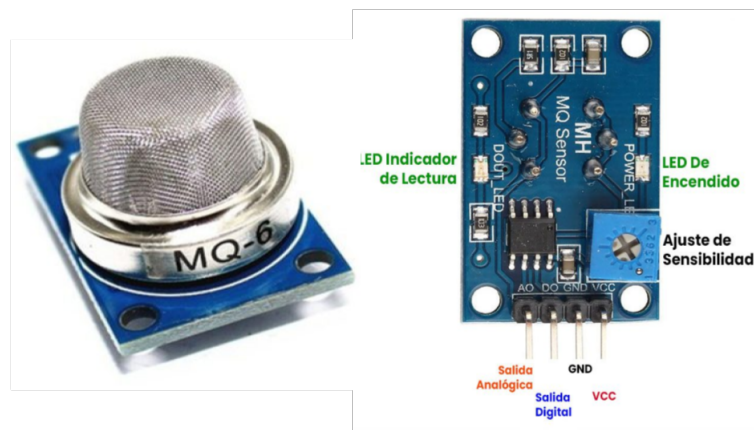


Figura 6: Sensor de gas MQ-6

- Buzzer

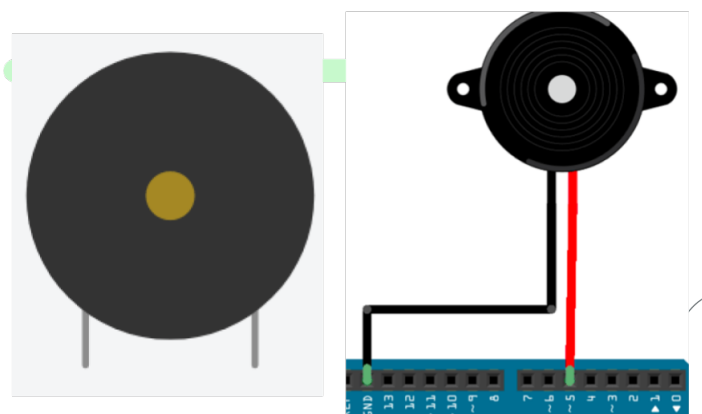


Figura 7: Buzzer o zumbador

- LEDs



Figura 8: LED rojo, azul, verde.

- Pantalla LCD

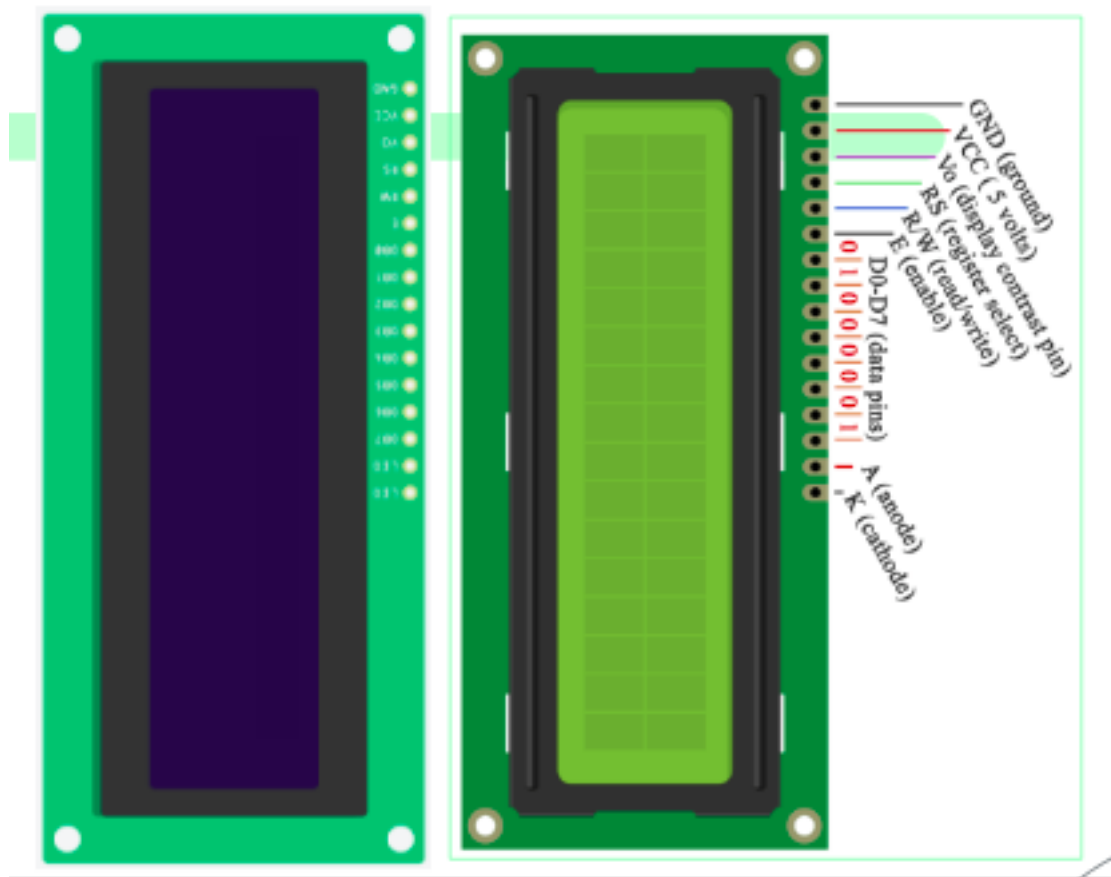


Figura 9: Pantalla LCD 16

- Resistencias

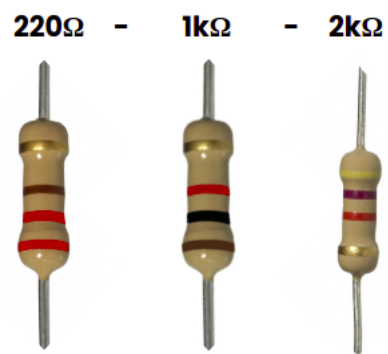


Figura 10: Resistencias

- Cables Jumper

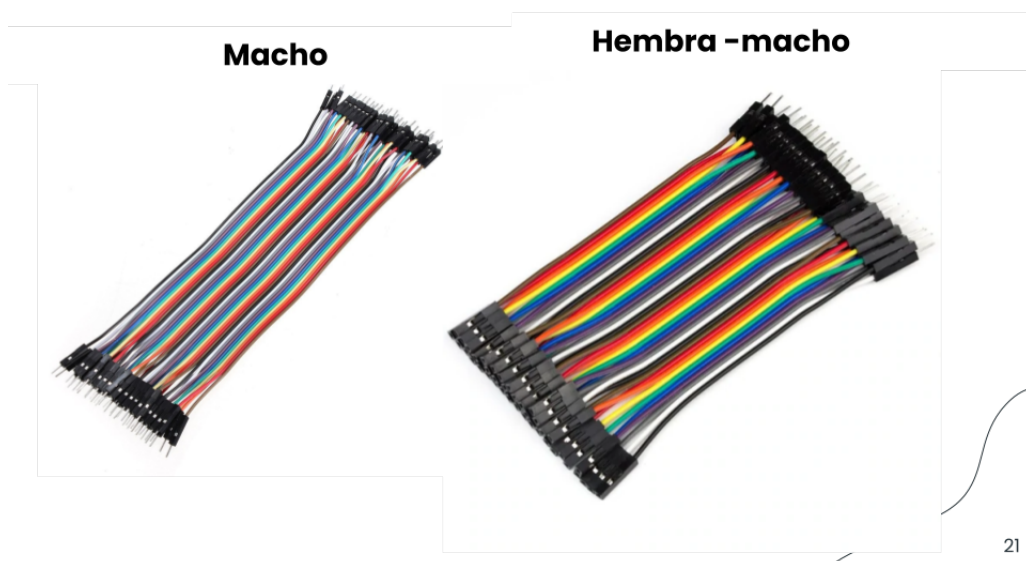


Figura 11: Cables

- Protoboard

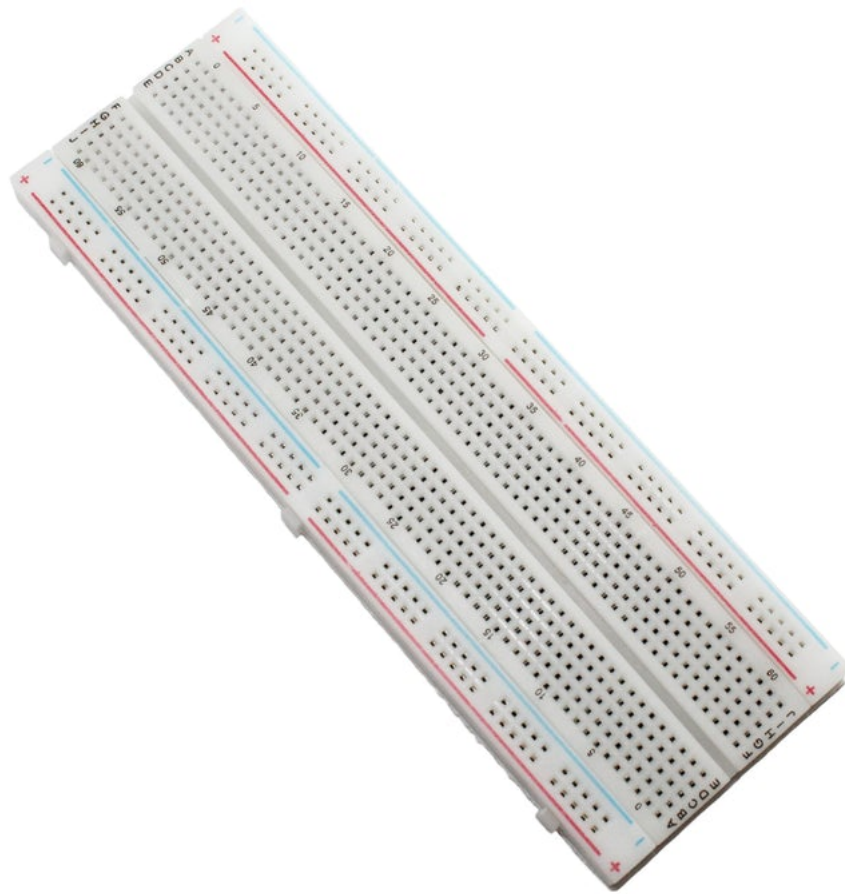


Figura 12: Protoboard

- Encendedor



Figura 13: Encendedor

- Laptop con un SO de Windows 10 o Ubuntu 20

- Smartphone con Android OS
- Software:
 - Arduino IDE 1.8.9
 - Compilador C/C++
 - MIT App Inventor
 - Controladores de USB para teléfonos móviles como OEM (marca del fabricante)

5. Resultados

Para los resultados de la conexión vemos que en la Figura 14 mostramos la conexión del arduino uno con la protoboard y los componentes principales conectando los leds rojo ver y azul los cuales serán explicados en la parte del código Apéndice A.

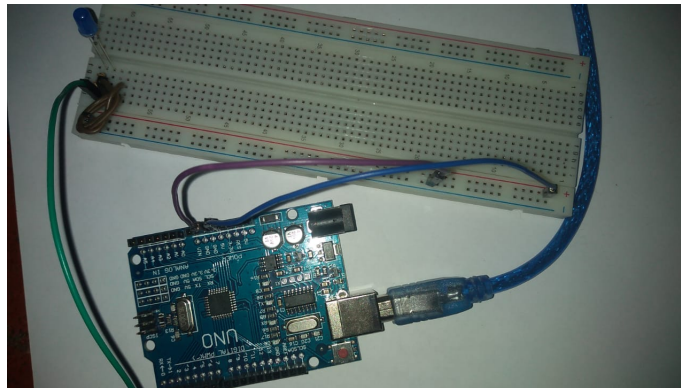


Figura 14: Conexión del Arduino con la protoboard led.

Para la conexión del LCD mostrado en la Figura 15 se recomienda soldar con el peine de cables macho-hembra para el completo contacto seguido de su potenciómetro regulador del LCD.

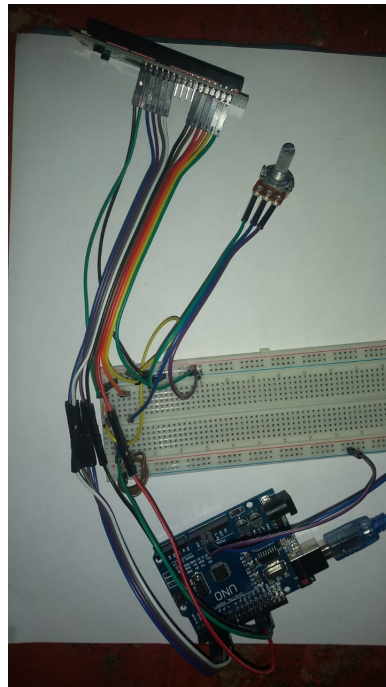


Figura 15: Conexión de LCD .

6. Conclusiones

Este proyecto nos ha enseñado a experimentar con la tecnología ARDUINO, en un inicio tuvimos complicaciones con las conexiones del modulo Bluetooth H-05, sin embargo pudimos superar este obstáculo, ha sido una experiencia interesante trabajar físicamente con esta tecnología, desde encender un LED hasta realizar el sistema completo del detector de gas.

Apendice A: Diagramas esquemáticos de circuitos

Conexión de Sensor de Gas MQ-6

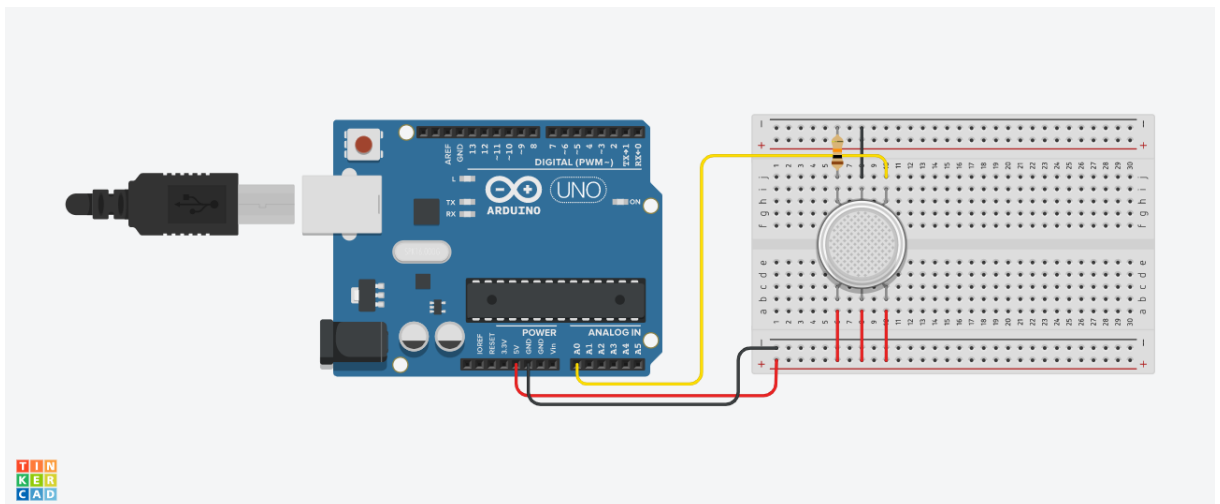


Figura 16: Diagrama esquemático en Tinkercat de la conexión del sensor de gas.

Conexión de LEDs

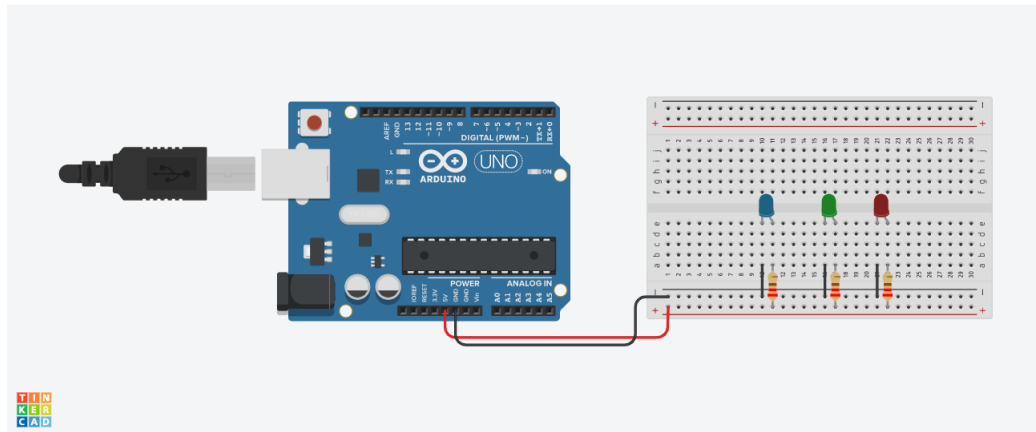


Figura 17: Diagrama esquemático en Tinkercat de la conexión de los LEDs.

Conexión del Modulo Bluetooth H-05

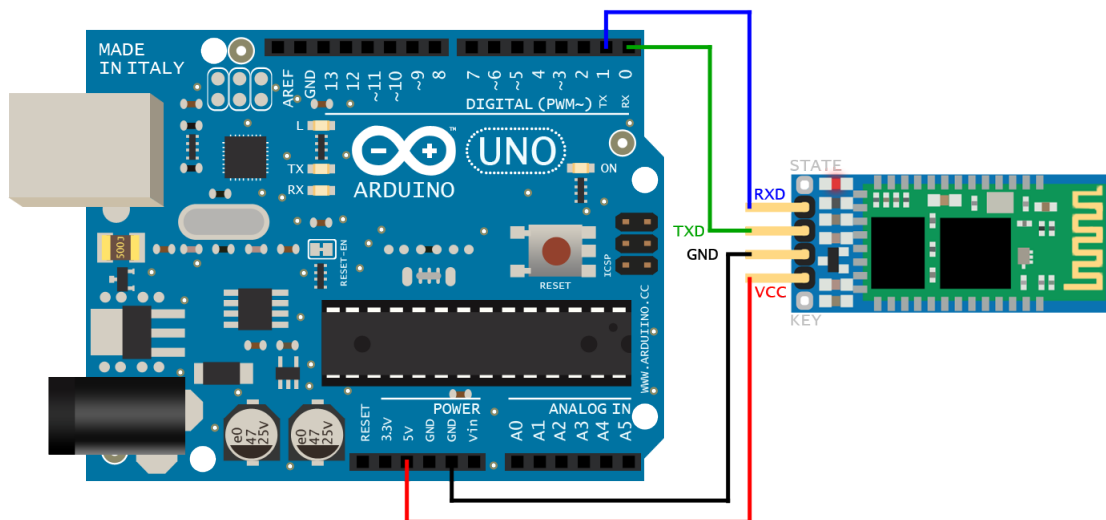


Figura 18: Diagrama esquemático de Modulo Bluetooth H-05.

Conexión del Buzzer o Zumbador

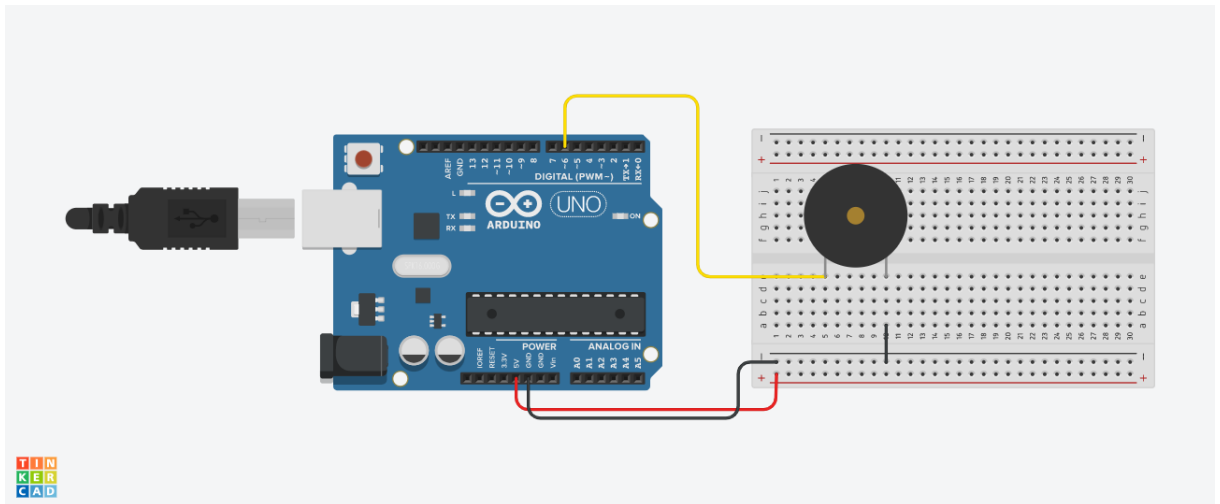


Figura 19: Diagrama esquemático del buzzer.

Conexión de LCD(16x2)

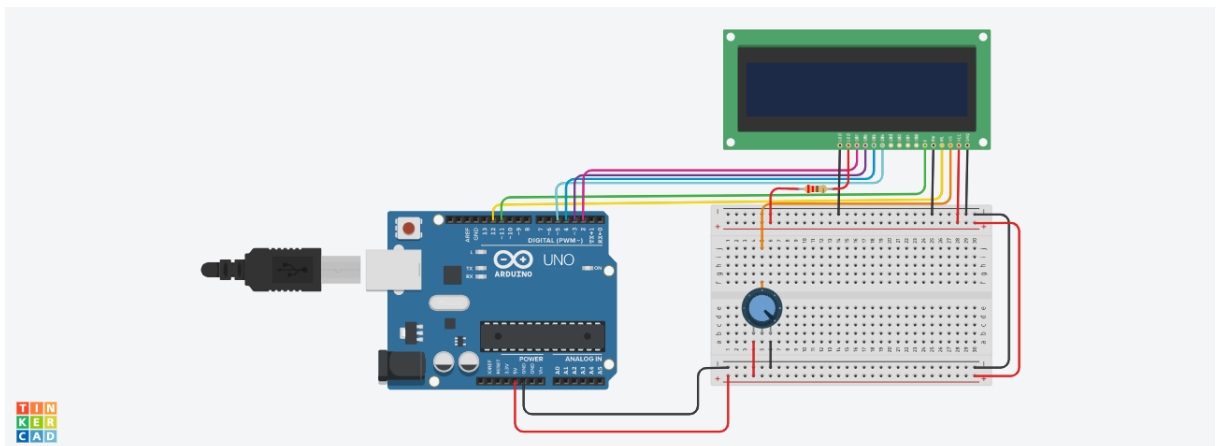


Figura 20: Diagrama esquemático de LCD.

Apendice B: Código

Código de la interfaz

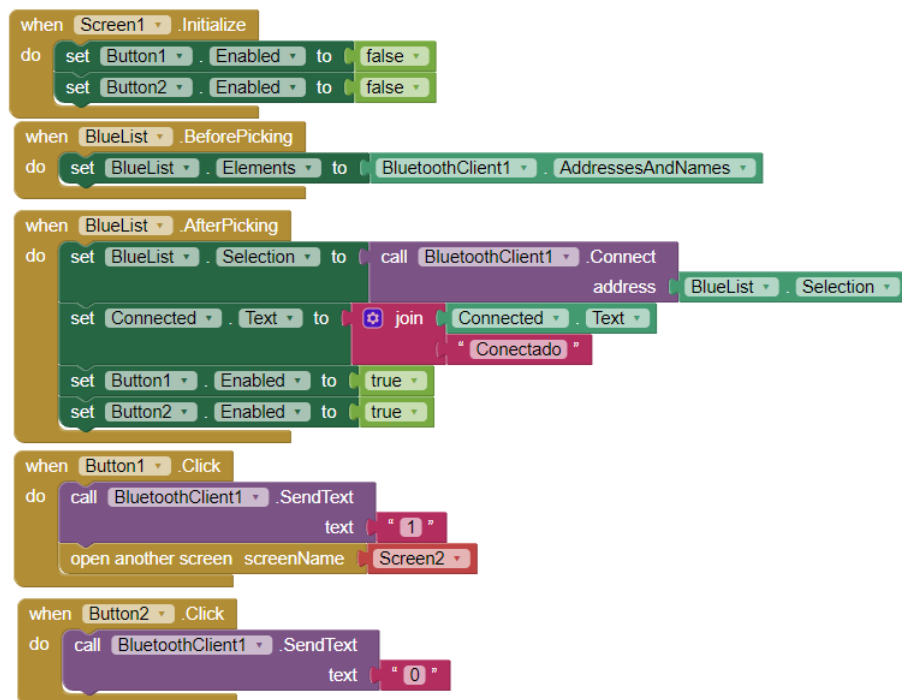


Figura 21: Código de Conexión al Bluetooth mediante la interfaz

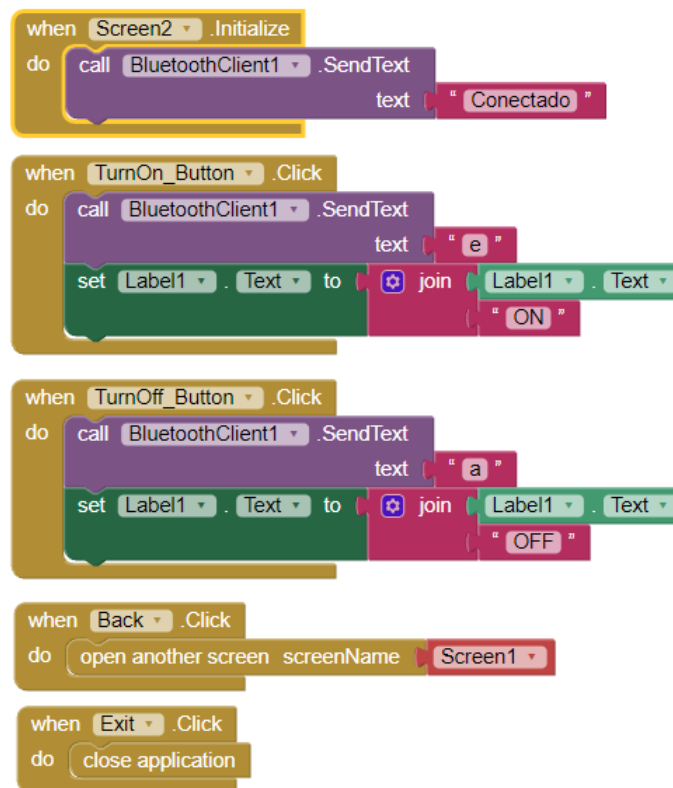


Figura 22: Código de Gestión del sensor mediante la interfaz

Código en Arduino

```

1  /* DEFINIENDO PINES DE DETECTOR DE GAS*/
2  const int pinGas = A0; //pin analogico para el sensor de gas
3  const int pinLed = 13; //pin digital para indicar escape de gas (HIGH=escape de gas y LOW=
    sin escape)
4  const int pinBuzzer = 10; //pin pwm del buzzer
5  // VARIABLES : sensor de gas
6  int lecturaGas = 0; // guarda la lectura del sensor
7  // VARIABLES : alarma
8  float sinVal; // guarda valor de sen
9  int toneVal;
10
11 /* MODULO BLUETOOTH */
12 const int pinLedB = 9; // indicara el inicio de la configuracion
13 const int pinBT = 8; //pin vcc de modulo bluetooth
14 const int pinEA = 7; // encendido y apagado del sensor
15 // VARIABLES: bluetooth
16 const char nombreBT[11] = "HC-05";
17 const char pin[5] = "0000"; // pin para asignar al bluetooth
18 const char velocidad = '4'; // constante 4 = 9600
19 const char mode = '1'; // 0 esclavo y 1 maestro
20
21 /* FUNCIONES */
22 void on_off_led(int pin){
23     digitalWrite(pin,HIGH);
24     delay(100);
25     digitalWrite(pin,LOW);
26     delay(100);
27     return ;
28 }
29 void alarma(){
30     for(int i = 0; i < 180 ; i++){

```

```

31     sinVal = (sin(i*(3.1412/180)));
32     toneVal = 2000 + int((sinVal*1000));
33     tone(pinBuzzer,toneVal);
34     delay(2);
35 }
36 }
37
38 void conf_BT(){
39
40     pinMode(pinLedB, OUTPUT); // azul
41     pinMode(pinBT, OUTPUT);
42     Serial.begin(38400);
43     digitalWrite(pinLedB, HIGH);
44     delay(4000);
45     digitalWrite(pinLedB, LOW);
46     digitalWrite(pinBT, HIGH);
47     delay(3000);
48     Serial.print("AT\r\n");
49     Serial.print("AT+NAME:");
50     Serial.print(nombreBT);
51     Serial.print("\r\n");
52     Serial.print("AT+PIN:");
53     Serial.print(pin);
54     Serial.print("\r\n");
55     Serial.print("AT+BAUD:");
56     Serial.print(velocidad);
57     Serial.print("\r\n");
58     Serial.print("AT+MODE:");
59     Serial.print(mode);
60     Serial.print("\r\n");
61     digitalWrite(pinLedB, HIGH);
62     digitalWrite(pinLedB, LOW);
63 }
64
65 bool Conect_android(){
66     if(Serial.available()){
67         char data=Serial.read();
68         if(data=='1'){ //conectado
69             digitalWrite(pinEA, HIGH);
70             return true;
71         }else if(data == '0'){ //desconectado
72             return false;
73         }
74     }
75 }
76
77 void iniciar_sensor(){
78     digitalWrite(pinEA, HIGH);
79     lecturaGas = analogRead(pinGas);
80     Serial.println(lecturaGas, DEC);
81     if(lecturaGas > 300 && lecturaGas < 400){
82         alarma();
83         on_off_led(pinLed);
84         Serial.println("!Cuidado, escape de gas!");
85     }else if(lecturaGas > 400){
86         alarma();
87         on_off_led(pinLedB); //pin azul
88         on_off_led(pinLed); // pin rojo
89         Serial.println("!Peligro, salga de casa!");
90     }else{
91         noTone(pinBuzzer);
92         digitalWrite(pinLed, LOW);
93         Serial.println("Todo tranquilo :D");
94     }
95     delay(1000);
96 }
97 void setup() {
98     pinMode(pinGas, INPUT );
99     pinMode(pinLed, OUTPUT );
100    pinMode(pinBuzzer, OUTPUT);

```

```

101 pinMode(pinLedB, OUTPUT );
102 pinMode(pinEA, OUTPUT );
103 conf_BT();
104 Serial.begin(9600);
105
106 }
107
108 void loop() {
109   Serial.println("Realizando la configuracion ...");
110   if(Conect_android() == true){
111     Serial.println("Sensor prendido");
112     iniciar_sensor();
113   }else{
114     Serial.println("Sensor apagado");
115     digitalWrite(pinEA, LOW);
116   }
117   delay(1000);
118   // Obtenemos la lectura del sensor de gas
119
120
121 }

```

Listing 1: Código fuente del Sistema Detector de Gas

Apéndice C: Interfaz de la aplicación

La interfaz fue desarrollada en el entorno de desarrollo para aplicaciones móviles llamado **MIT App Inventor**, en la cual se pueden desarrollar una serie de aplicaciones móviles. Tal como se menciona en su página oficial es un ambiente de desarrollo de programación intuitivo y carácter visual para el desarrollo de aplicaciones móviles en Android y iOS.

Utiliza la programación basada en bloques así como el uso de código fuente. Este fue desarrollado por el grupo de CSAIL y estudiantes auspiciado por Google Labs MIT App Inventor, 2020. Por tal motivo, a continuación se muestra la interfaz que contiene dicho aplicativo.

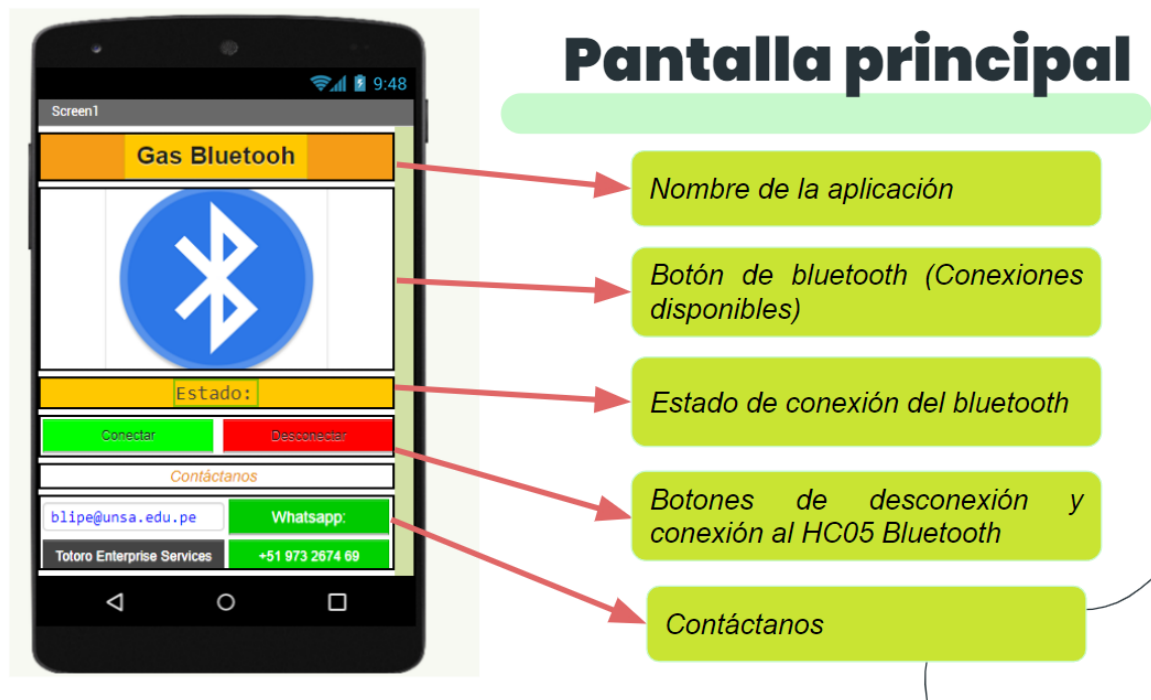


Figura 23: Pantalla principal: Manejo del Bluetooth

Como se observa en la figura 23 se denotan las partes fundamentales que contiene la aplicación, siendo las más importantes las de Botón de Bluetooth, Estado de conexión y los botones de gestión de conexión. Básicamente con estos 3 elementos se logra realizar la conexión del HC05 Bluetooth con el teléfono móvil a utilizar con el prototipo. Luego, de estar conectado pasa a la siguiente pantalla, la cual se muestra en la figura 24. Esta se encarga del manejo del sensor de gas (MQ 6), el cual lo desactiva/activa cuando el usuario lo desee, mientras esté en uso la aplicación.

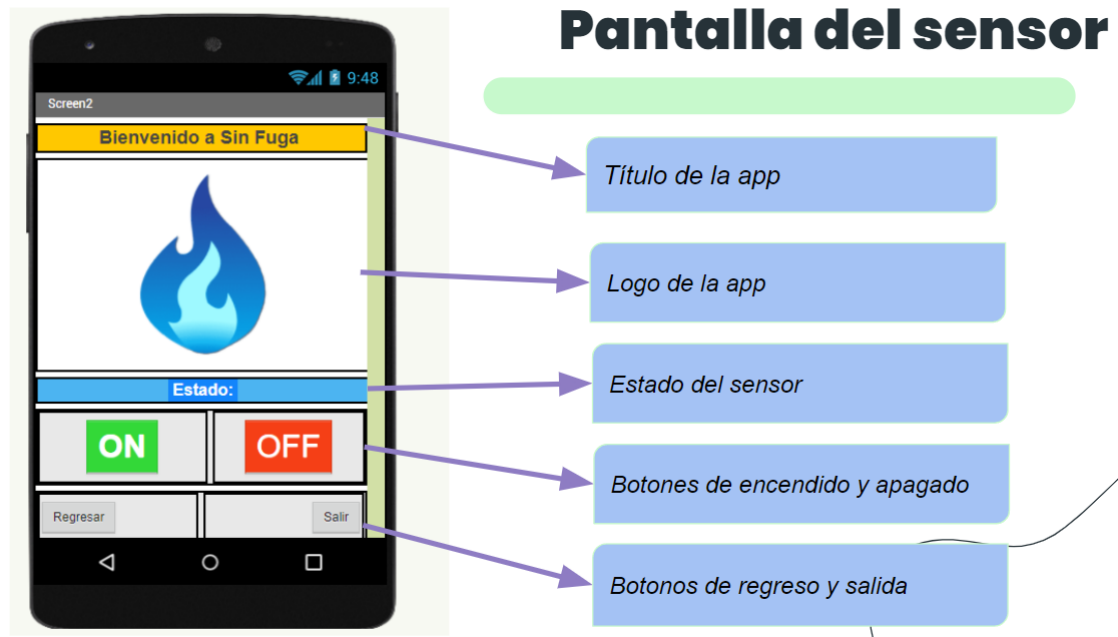


Figura 24: Pantalla secundaria: Gestión del sensor

6. Referencias

- Baha, H., & Dibi, Z. (2009). A novel neural network-based technique for smart gas sensors operating in a dynamic environment. *Sensors*, 9(11), 8944-8960.
- FEZARI, M., HATTAB, R., & AL-DAHOU, A. (2015). Oak Ridge Air Quality Index Computation: a way for Monitoring Pollutions in Annaba City. *Conference: International Arab Conference on Information Technology (ACIT)*.
- Kim, J. Y., Kang, S. W., Shin, T. Z., Yang, M. K., & Lee, K. S. (2006). Design of a smart gas sensor system for room air-cleaner of automobile-thick-film metal oxide semiconductor gas sensor. *2006 International Forum on Strategic Technology*, 72-75.
- Kouda, S., Dendouga, A., Barra, S., & Bendib, T. (2018). Design of a selective smart gas sensor based on ANN-FL hybrid modeling.
- Liu, Y., Dillon, T., Yu, W., Rahayu, W., & Mostafa, F. (2020). Missing value imputation for industrial IoT sensor data with large gaps. *IEEE Internet of Things Journal*, 7(8), 6855-6867.
- MIT App Inventor. (2020). MIT App Inventor [Accessed: 2021-12-28].
- Nasef, M. M., & Hegazy, E.-S. A. (2004). Preparation and applications of ion exchange membranes by radiation-induced graft copolymerization of polar monomers onto non-polar films. *Progress in Polymer Science*, 29(6), 499-561.
- Percival, D. B., & Walden, A. T. (2000). *Wavelet methods for time series analysis* (Vol. 4). Cambridge university press.
- Popa, A., Hnatiuc, M., Paun, M., Geman, O., Hemanth, D. J., Dorcea, D., Son, L. H., & Ghita, S. (2019). An intelligent IoT-based food quality monitoring approach using low-cost sensors. *Symmetry*, 11(3), 374.

Rumantri, R., Khakim, M., & Iskandar, I. (2018). Design and characterization of low-cost sensors for air quality monitoring system. *Jurnal Pendidikan IPA Indonesia*, 7(3), 347-354.