



UNIVERSIDAD DE PAMPLONA
La academia al servicio de la vida

**IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y
TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL**

Autor:
Wilian Alberto Uribe Suarez

Director:
PhD. (c). Jesús Eduardo Ortíz Sandoval
Co-director:
PhD. Oscar Eduardo Gualdrón Guerrero

Ingeniería Electrónica
Departamento de Ingenierías Eléctrica, Electrónica, Sistemas y
Telecomunicaciones
Facultad de Ingenierías y Arquitectura
Universidad de Pamplona
Pamplona, 16 de Mayo de 2017

Para Ustedes

Teodoro y Josefina

Eterna Gratitud

ÍNDICE

Prefacio	VIII
Agradecimientos	XI
Resumen	XIII
1. INTRODUCCION	- 2 -
1.1 PROBLEMA	- 2 -
1.2 OBJETIVOS	- 3 -
1.3 PASOS METODOLÓGICOS	- 3 -
1.4 DISTRIBUCIÓN BIBLIOGRÁFICA.....	- 4 -
2. MARCO TEÓRICO	- 7 -
2.1 ANTECEDENTES	- 7 -
2.2 SISTEMA MULTISENSORIAL.....	- 9 -
2.2.1 OLFATO HUMANO	- 9 -
2.2.2 NARIZ ELECTRÓNICA.....	- 10 -
2.3 SENSORES.....	- 11 -
2.3.1 SENSORES DE GAS	- 11 -
3. DISEÑO DE COMPONENTES	- 13 -
3.1 DISEÑO PCB SENSORES	- 13 -
3.1.1 DISEÑO SCHEMATIC CAPTURE	- 14 -
3.1.2 DISEÑO PCB LAYOUT	- 16 -
3.2 DISEÑO PCB ARDUINO	- 18 -
3.3 DISEÑO PCB SHIELD SENSORES ARDUINO	- 18 -
3.4 IMPRESO DE LA SHIELD SENSORES MQ PARA ARDUINO	- 20 -
3.5 DISEÑO CAJA MODULO TRANSMISOR	- 21 -
3.6 DISEÑO CAJA MODULO RECEPTOR.....	- 22 -
3.7 COBERTURA RED MOVIL CLARO	- 23 -
4. PROGRAMACION Y DESARROLLO	- 27 -
4.1 TARJETA ARDUINO UNO	- 27 -
4.1.1 CARACTERÍSTICAS GENERALES DE ARDUINO UNO	- 27 -
4.2 MODULO GPRS SIM900.....	- 29 -
4.2.1 CARACTERÍSTICAS DEL MODULO SIM900.....	- 29 -

4.2.2 CONEXIÓN DEL MODULO SIM900.....	- 32 -
4.3 COMANDOS AT.....	- 33 -
4.4 VERIFICAR MUDULO SIM900 CON COMANDOS AT	- 36 -
4.5 EJEMPLOS CON EL MODULO SIM900 Y ARDUINO UNO	- 38 -
4.5.1 ENCENDER UN LED CON UNA LLAMADA TELEFONICA. [13].....	- 38 -
4.6 CALIBRACION DE LOS SENSORES MQ	- 41 -
4.6.1 SENSOR DE GAS MQ	- 41 -
4.6.2 ESCALANDO LOS SENSORES MQ	- 43 -
4.7 IMPLEMENTACION DEL SISTEMA DE MEDICION DE GAS METANO	- 47 -
4.7.1 MODULO TRANSMISOR (TX).....	- 49 -
4.7.1.1 DESCRIPCION DEL MODULO TRANSMISOR (TX).....	- 49 -
4.7.1.2 DIAGRAMA DE CONEXIÓN DEL MODULO (TX).....	- 50 -
4.7.1.3 PROGRAMACIÓN DEL MODULO TRANSMISOR (TX).....	- 50 -
4.7.2. MODULO RECEPTOR (RX).....	- 56 -
4.7.2.1 DESCRIPCION DEL MODULO RECEPTOR (RX).....	- 56 -
4.7.2.2 DIAGRAMA DE CONEXIÓN DEL MODULO (RX).....	- 57 -
4.7.2.3 PROGRAMACION DEL MODULO RECEPTOR (RX).....	- 58 -
4.8 DESARROLLO DE LA INTERFAZ GRAFICA EN MATLAB	- 61 -
4.8.1 PROGRAMACION ARDUINO Y MALATB.....	- 61 -
4.8.2 DISEÑO DE INTERFAZ GRAFICA EN GUIDE MATLAB	- 63 -
4.9 CREAR EJECUTABLE EN MATLAB.....	- 64 -
5. ANALISIS Y RESULTADOS.....	- 68 -
5.1 PRUEBA NARIZ E-NOSE V2.0	- 68 -
5.2 PRUEBA DEL SISTEMA DISEÑADO	- 69 -
5.3 LIMITANTES.....	- 70 -
5.4 ANÁLISIS ECONÓMICO	- 70 -
6. CONCLUSIONES FINALES	- 73 -
BIBLIOGRAFÍA.....	- 74 -

ÍNDICE DE FIGURAS

Fig. 1 Ciclo del proceso de diseño del instrumento de medición.....	- 4 -
Fig. 2 Conexiones entre la mucosa y el bulbo olfativo. [7].....	- 10 -
Fig. 3 Sensores de la familia Fígaro.....	- 11 -
Fig. 4 Diseño de un nuevo modelo esquemático del sensor MQ. [8].....	- 14 -
Fig. 5 Guardar el nuevo modelo esquemático creado. [8].....	- 15 -
Fig. 6 Utilización del nuevo componente en Schematic Capture. [8].....	- 16 -
Fig. 7 Asignación de la PCB para el modelo creado. [8].....	- 17 -
Fig. 8 PCB para la tarjeta Arduino uno. [8].....	- 18 -
Fig. 9 Diseño sensores arduino Schematic capture. [8].....	- 19 -
Fig. 10 Diseño sensores arduino PCB Layout. [8].....	- 19 -
Fig. 11 Máquina de circuitos impresos.	- 20 -
Fig. 12 Shield Sensores MQ para Arduino. [8].....	- 20 -
Fig. 13 Diseño caja modulo transmisor.....	- 21 -
Fig. 14 Caja modulo transmisor.	- 21 -
Fig. 15 Diseño de la Caja modulo receptor.	- 22 -
Fig. 16 Caja modulo receptor.	- 22 -
Fig. 17 Cobertura de red móvil Claro	- 23 -
Fig. 18 Distancia geográfica entre Leticia y Riohacha.	- 23 -
Fig. 19 Tarjeta Arduino uno. [9].....	- 27 -
Fig. 20 Descripción de la Tarjeta arduino uno. [10].....	- 28 -
Fig. 21 Diagrama de conexión GPRS y Arduino.....	- 29 -
Fig. 22 Diagrama funcional de SIM900. [11].....	- 32 -
Fig. 23 Diagrama de conexión GPRS y Arduino.	- 33 -
Fig. 24 Ubicación del interruptor que selecciona pines (Tx, Rx). [13].....	- 33 -
Fig. 25 Sketch para verificar funcionamiento del módulo SIM900. [15].....	- 36 -
Fig. 26 Monitor serial, comandos AT del módulo SIM900. [15].....	- 37 -
Fig. 27 Encender LED mediante llamada telefónica. [15].....	- 39 -
Fig. 28 Monitor serial encender LED mediante llamada.	- 40 -
Fig. 29 Función .indexOf	- 40 -
Fig. 30 Función identifica_llamada ().	- 41 -
Fig. 31 Sensores de gas MQ. [16].....	- 42 -
Fig. 32 Sensitivity characteristics of the MQ-4. [16].....	- 43 -
Fig. 33 Regresión y ecuación del sensor MQ-4.....	- 44 -
Fig. 34 Diagrama esquemático para leer el sensor MQ-4. [15].....	- 45 -
Fig. 35 Programa para leer el sensor MQ-4. [15].....	- 46 -
Fig. 36 Monitor serial valores obtenidos del sensor MQ-4. [15].....	- 47 -
Fig. 37 Diagrama de bloques del sistema de comunicación	- 48 -
Fig. 38 Sistema de comunicación inalámbrico	- 48 -

Fig. 39 Diagrama de bloques del módulo transmisor (Tx)	- 49 -
Fig. 40 Diagrama de conexión del módulo transmisor (Tx). [8]	- 50 -
Fig. 41 Programa principal del módulo transmisor (Tx)	- 56 -
Fig. 42 Diagrama de bloques del módulo receptor (Rx).....	- 57 -
Fig. 43 . Diagrama de conexión del módulo receptor (Rx). [8].....	- 57 -
Fig. 44 Programa del módulo receptor (Rx)	- 61 -
Fig. 45 Comunicación arduino & matlab.....	- 62 -
Fig. 46 Ventana inicio de la interfaz gráfica	- 63 -
Fig. 47 Ventana capture de la interfaz gráfica	- 64 -
Fig. 48 Crear ejecutable en matlab.....	- 65 -
Fig. 49 Archivos generados para el ejecutable de la aplicación creada en matlab	- 65 -
Fig. 50 Pruebas E-nose V2.0	- 68 -
Fig. 51 Resultado E-nose V2.0	- 68 -
Fig. 52 Resultado del sistema diseñado.....	- 69 -
Fig. 53 Presupuesto modulo Transmisor.....	- 70 -
Fig. 54 Presupuesto modulo Receptor.	- 71 -

ÍNDICE DE TABLAS

Tabla 1 Especificaciones técnicas. [9]	- 27 -
Tabla 2 Características Generales. [11]	- 30 -
Tabla 3 Especificaciones. [11].....	- 30 -
Tabla 4 Características Eléctricas. [11]	- 30 -
Tabla 5 Indicador LED NETSTATUS. [12]	- 31 -
Tabla 6 Configuraciones SIM900. [12].....	- 31 -
Tabla 7 Comandos AT Generales. [14]	- 34 -
Tabla 8 Comandos del servicio de red. [14].....	- 34 -
Tabla 9 Comandos de seguridad. [14].....	- 35 -
Tabla 10 Comandos para la agenda de teléfonos. [14]	- 35 -
Tabla 11 Comandos para SMS. [14].....	- 35 -
Tabla 12 Comandos AT y su respuesta. [14].....	- 38 -
Tabla 13 MQ series Semiconductor Gas Sensor. [16]	- 42 -
Tabla 14 Datos de Calibración del sensor MQ-4.....	- 45 -
Tabla 15 Descripción de las funciones utilizadas en el programa.....	- 50 -
Tabla 16 Códigos de las funciones utilizadas en el programa.....	- 52 -
Tabla 17 Descripción de las funciones utilizadas en el programa.....	- 58 -
Tabla 18 Códigos de las funciones utilizadas en el programa.....	- 59 -
Tabla 19 Funciones utilizadas en el programa matlab. [17]	- 62 -

Prefacio

Prefacio	VIII
Agradecimientos	XI
Resumen	XIII

“Entre las dificultades se esconde la oportunidad”

Albert Einstein

Agradecimientos

En primera medida quiero agradecer a Dios por darme la oportunidad y los dones necesarios de iniciar este proyecto de vida y llevarlo a feliz término

A mi madre, gracias a todos sus consejos, a la formación que desde muy niño fue inculcando en mi ser, por estar presente en todos los momentos que siempre necesite un abrazo o una palabra de apoyo en los momentos difíciles o cuando creía inalcanzable alguna meta ,por sacrificarse en toda medida para que yo siempre pudiera cumplir todas mis necesidades, porque gracias a ella aprendí a leer, escribir, sumar, restar y simplemente por ser la mejor madre del mundo, este triunfo es con todo mi corazón.

A mi padre, por el ejemplo y la tenacidad que aprendí de él, porque gracias a ese ejemplo aprendí que de todas las dificultades siempre hay una oportunidad oculta y que por más grande que sea el problema si hay unión siempre hay solución, Porque el orgullo que el siente por mí me impulso a siempre corresponder con sacrificio y llevarle alegrías para poder iluminar el camino lleno de dificultades que muchas veces se tiene que transitar en el transcurso de la vida. Muchas gracias por enseñarme la fortaleza para enfrentar la vida.

Al Magister **Jesús Eduardo Ortiz Sandoval** mi director de trabajo de grado, por la confianza brindada para el desarrollo del proyecto, por todo el apoyo y confianza que siempre tuvo en mi persona para llevar a buen término la investigación, porque gracias a sus conocimientos he alcanzado una meta más, porque siempre estuvo al pendiente de la investigación y lo más importante en los momentos en que estuve bajo de ánimo siempre supo decir las palabras correctas para no desfallecer y continuar trabajando.

Al Doctor **Oscar Eduardo Gualdrón** ya que es muy bueno poder contar con la dirección de un excelente profesional que se preocupa por la excelencia de los trabajos y que le imprime integridad a las investigaciones buscando obtener los mejores resultados posibles.

Quiero también agradecerle de una forma muy especial al profesor **Jesús Eduardo Ortiz Sandoval** Director del programa de Ingeniería Electrónica, quien estuvo al pendiente de todo el desarrollo del proyecto y siempre tuvimos el apoyo incondicional en cuanto a las prestaciones de los servicios necesarios dentro de las instalaciones de la Universidad.

A los profesores del programa Ingeniería Electrónica, entre ellos los profesores designados como jurados de mi trabajo de grado el profesor **Julio Cesar Ospino**,

profesor **Carlos Arturo Vides Herrera**. Ya que tener como evaluadores a unos profesionales íntegros y capacitados represento en mí una obligación para hacer la investigación con excelencia y responsabilidad.

A mis amigos y compañeros de estudio durante toda etapa de Universidad, en especial un agradecimiento a aquellos que formaron parte de los grupos de trabajo y de investigación.

A mis hermanos **Uribe Suarez** de quien siempre he recibido el máximo apoyo tanto moral como económico para poder continuar con mi carrera, por ser un ejemplo de perseverancia

A mi familia, a mis profesores, a mis amigos, conocidos y a todas las personas que tal vez olvido en mencionar, mil y mil gracias. Gratitud Eterna.

1.

Introducción

1. Introducción.....	2
1.1. Problema.....	2
1.2. Objetivos.....	3
1.3. Pasos Metodológicos.....	3
1.4. Distribución Bibliográfica.....	4

1. INTRODUCCION

1.1 PROBLEMA

La contaminación del medio ambiente, se ha incrementado desde los años de la industrialización; los gases de efecto invernadero, hacen que la atmosfera se caliente y producen el calentamiento global no natural. En la Actualidad es un problema mundial, debido a la consecuencia que trae con esto, como los cambios climáticos que han dado muestra de su poder y han deteriorado la atmósfera terrestre, los océanos, aumentando la temperatura media de la tierra, lo cual conlleva a desastres naturales. Entre los gases de efecto de invernadero se encuentran diferentes tipos, uno de ellos es el gas metano CH₄ la cual es uno de los gases que más contamina y calienta la atmosfera.

Las emisiones de metano se ven incrementadas por actividades como la ganadería, industrias privadas que explotan recursos naturales, rellenos sanitarios; debidos a la acumulación y descomposición de desechos.

En la Universidad de Pamplona en la maestría de controles industriales se están haciendo procesos de investigación sobre el gas metano como fuente de energía la cual ellos requieren el diseño de un instrumento de medición que le detecte la presencia de estos gases y transmita su información a un punto de control para el procesamiento de la información.

El gas natural está compuesto de 97% metano, lo cual de claro que el metano es una fuente de energía muy útil. Para obtenerlo existen varias maneras, el metano puede ser extraído de los depósitos geológicos conocidos como campos de gas natural, y se forma por la descomposición anaeróbica de materia orgánica; también puede ser extraído de procesos en la digestión y defecación de animales (especialmente ganado). La digestión anaeróbica de la biomasa. Estos últimos han sido de gran aprovechamiento en países como Alemania, donde granjas se proveen del metano para su sostenibilidad y lo convierten en energía calórica y eléctrica. En Colombia los desarrollos de tecnología para energías alternativas se encuentran en una menor escala, una de las razones es que fuentes de energía como el metano son difíciles de medir y ubicar, por consiguiente implementar un sistema de medición y transmisión inalámbrica de gas metano sería de gran utilidad para el desarrollo de nuevas tecnologías y aprovechamiento de los recursos naturales.

1.2 OBJETIVOS

En el momento que se toma la decisión de realizar un proyecto de tal magnitud hay que examinar detenidamente el área sobre la que se va a desarrollar el proyecto para engranar perfectamente las actividades que se vayan a implementar pensando en llevar a buen destino los resultados de la investigación. Con estos argumentos se describe a continuación el objetivo principal de la investigación.

- *Implementación de un sistema de medición de gas metano y transmisión inalámbrica de los datos a un punto de control.*

También se estructuraron una serie de objetivos complementarios que facilitarían el proceso de investigación.

- *Realizar la ingeniería de detalle de la tarjeta de adquisición de datos inalámbrica.*
- *Estructurar Diseñar el sistema de medición del gas metano a partir de la tarjeta adquisición de datos.*
- *Diseñar una interfaz gráfica para visualizar la variable transmitida y poder generar reportes.*
- *Validar el funcionamiento del sistema desarrollado en una estación de control con la interfaz desarrollada.*

1.3 PASOS METODOLÓGICOS

Para dar comienzo con nuestro proyecto de implementar un sistema de medición de gas metano y transmisión inalámbrica de los datos a un punto de control se tuvo en cuenta en primera instancia la toma de muestra de datos de emisiones de gas metano en el basurero de Pamplona Norte de Santander, empleando la *E-nose versión 2.0* del grupo de investigación *GISM* para tener información concreta de la variable de medición y así poder diseñar un buen sistema.

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

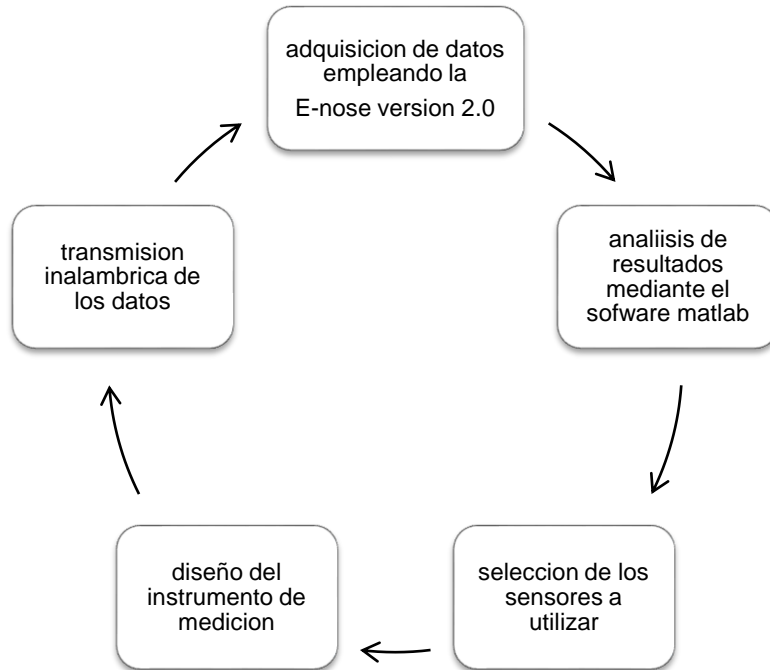


Fig. 1 Ciclo del proceso de diseño del instrumento de medición.

Todo el proceso que se describe en la figura 1, se realizan primero una recolección de información sobre la emisión de gases presentes en lugar mencionado anteriormente, la cual estos datos adquiridos se procesan y analizan en el software matlab, la cual nos permitió tener algunas bases para escoger los sensores adecuados para el correcto funcionamiento de nuestro sistema de medición de gas metano.

1.4 DISTRIBUCIÓN BIBLIOGRÁFICA

El libro se organiza mediante cinco capítulos concernientes a la Introducción, Marco Teórico, Diseño de Componentes, Programación y Desarrollo, Resultados y Conclusiones.

Cada capítulo tiene los ítems correspondientes al tema de interés.

2.

Marco Teórico

2. Marco Teórico.....	7
2.1. Antecedentes.....	7
2.2. Sistema Multisensorial.....	8
2.2.1. Olfato Humano	9
2.2.2. Nariz Electrónica.....	10
2.3. Sensores.....	11
2.3.1. Sensores de Gas.....	11

2. MARCO TEÓRICO

2.1 ANTECEDENTES

En los siguientes artículos científicos de investigación relacionados con este proyecto, la cual habla acerca de trabajos realizados en este campo.

Application of electronic nose for industrial odors and gaseous emissions measurement and monitoring. [1]

En este artículo trata del desarrollo del olfato artificial (nariz electrónica) a partir del olfato humano, analizando su funcionamiento y comportamiento. Realizan algunos estudios de emisión y monitoreo de gases en procesos industriales, implementando un sistema electrónico de olfato artificial (nariz electrónica) para determinar y clasificar estos olores.

A novel sensor selection using pattern recognition in electronic nose. [2]

Este artículo estudia la nariz electrónica portable (E-Nose) basado en sensor de gas MOS (Metal Oxide Semiconductor) para la detección de múltiples clases de olores contaminantes en el aire. Presenta características de portabilidad, bajo costo, múltiples tarjetas de detección y monitoreo. Benceno, tolueno, monóxido de carbón y amoníaco son las primeras tarjetas propuestas por E-Nose con beneficios desde las características de amplio espectro, reproducibilidad, sensibilidad y bajo costo de sensores de gas MOS.

“Artificial sniffing” based on induced temporary disturbance of gas sensor response. [3]

El olfateo natural es una fuente importante de inspiración en gas sensible. Los diferentes aspectos de este proceso son explotados para lograr la funcionalidad de las correspondientes soluciones técnicas. El mecanismo del sensor de gas semiconductor causa que la respuesta de este dispositivo dependa de: la concentración/presión parcial del gas y la interferencia de especies; las propiedades de las moléculas del gas (especialmente su peso molecular y forma, la afinidad del electrón, potencial de ionización, coeficiente de difusión). Estudiando estas características fundamentales se pueden desarrollar nuevas tecnologías en dispositivos de sensores de gases.

Methane emissions measurements of natural gas components using a utility terrain vehicle and portable methane quantification system. [4]

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

Las emisiones de gas de efecto de invernadero (GHG) son un problema de crecimiento en los estados unidos (US). El metano (CH₄) es un potente GHG (Greenhouse Gas) producido por severas estaciones del sector de gas natural. En un intento para cuantificar estas emisiones producidas por los vehículos convencionales, los investigadores modificaron un offroad el vehículo de terreno de utilidad(UTV) para incluir un sistema de muestreo del flujo máximo caudal FFS (Full Flow Sampling System) para la cuantificación del metano. Nuestra instrumentación de un UTV para incluir una detección del CH₄ y el sistema de cuantificación posibilitó localizar, acceso y medir las emisiones del CH₄ de varios sitios que fueron identificados como posibles emisores de CH₄.

La Nariz Electrónica: Estado del Arte. [5]

Este artículo menciona los orígenes de la nariz electrónica, las cuales se dan en los años 60, cuando la compañía Bacharach Inc., construyó un dispositivo conocido con el nombre de Sniffer, el cual constaba de un solo sensor de gas y por lo tanto, no se consideró una nariz electrónica. Krishna Persaud y George Dodd, realizaron la primera publicación relacionada con las narices electrónicas en 1982 (Persaud y Dodd, 1982), orientaron sus estudios en entender los procesos del olfato biológico, empleando un conjunto de sensores semiconductores de óxidos metálicos (MOS), y luego ampliaron su investigación a sensores basados en polímeros conductores. Las narices electrónicas fueron desarrolladas en un comienzo para reemplazar a los paneles de expertos en la clasificación de aromas, debido a que estos se consideran costosos, difíciles de transportar. En la actualidad las narices electrónicas son utilizadas para la detección de olores.

Organochlorine screening using food in fresh an electronic nose. [6]

En esta investigación proponen un método o alternativa para detectar a familia de pesticidas de organocloruros, organofosfatos en comidas frescas especialmente las frutas usando narices electrónicas o sistemas multisensoriales. Esta nariz electrónica construida, la meta es lograr una clasificación cualitativa en dos clases, la presencia de pesticidas y alimento orgánico, la cual pretende reducir la tasa de enfermedad por la contaminación excesiva de comida y crear una nueva metodología a través de un costo bajo. El equipo electrónico es una herramienta para agricultores pequeños y medianos.

2.2 SISTEMA MULTISENSORIAL

Cuando se habla de sistemas multisensoriales hay que tener en cuenta que son dispositivos diseñados para emular las características de los sentidos que poseemos los seres vivos. Estos sistemas pueden ser la Lengua Electrónica o el Olfato Electrónico que es el objeto de estudio del presente trabajo. Estos sentidos y como se explica a continuación son compuestos por múltiples membranas encargadas de percibir los patrones y enviar los correspondientes estímulos eléctricos al cerebro para que sea procesada la información. [7]

En los sistemas artificiales los encargados de hacer esa función son los diferentes sensores que igual que sus pares en el ser humano envían una señal eléctrica dependiendo del patrón presente en el ambiente.

2.2.1 OLFATO HUMANO

Antes de comenzar con la explicación del funcionamiento del olfato humano debemos remitirnos a definir que es un olor, y que hace posible su funcionamiento. El olor de cualquier material es una mezcla compleja de moléculas olorosas de distintos tipos, cada una tiene una concentración determinada lo que causa que al unirse todas se generen una muestra particular. Las moléculas que pueden ser detectadas por nuestro sentido del olfato deben tener ciertas propiedades, como por ejemplo ser lo suficientemente pequeñas como para ser volátiles (masa molecular relativa mayor de 30 a 300 g·mol⁻¹), de modo que puedan vaporizarse, alcanzar la nariz y luego disolverse en la mucosa olfativa. Otros factores importantes son las interacciones internas (moléculas polares o no polares), la distribución de átomos, la distribución de carga, y la posibilidad de que se produzcan rotaciones estructurales. Finalmente, las relaciones (fuerzas) intermoleculares también tienen mucha importancia, y determinan finalmente su volatilidad y solubilidad en agua, entre otros factores.

A través del aire aspirado por la nariz o la boca, las moléculas olorosas llegan a la cavidad nasal, y a través de los conductos nasales llegan a una zona en la parte superior interna de la nariz: el epitelio olfativo. La superficie de dicho epitelio olfativo (unos 5 cm² en los humanos) está cubierta por una fina capa mucosa, fundamentalmente compuesta de agua (y cuya misión principal es hacer de filtro mecánico y químico de grandes partículas).

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

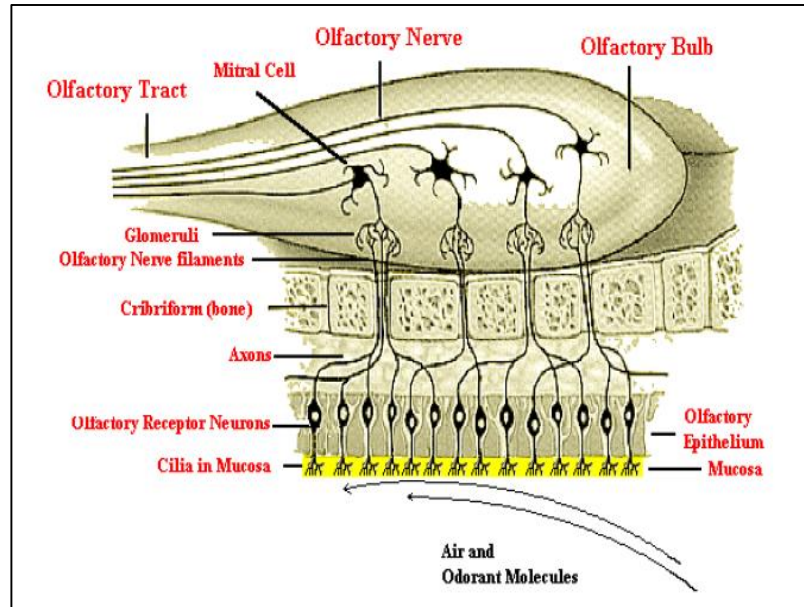


Fig. 2 Conexiones entre la mucosa y el bulbo olfativo. [7]

En la figura 2 se observa cómo se interconecta todos los elementos del sistema olfativo con ayuda de este gráfico se explicará como es el proceso químico básico:

- La estructura de la proteína receptora cambia cuando entra en contacto con la molécula olorosa, afectando a una proteína G asociada.
- La proteína G asociada cambia y activa una enzima (adenilato ciclasa)

2.2.2 NARIZ ELECTRÓNICA

Los sistemas multisensoriales se han ido diseñando y mejorando para que se acerquen poco a poco al funcionamiento de los sentidos en el ser humano, aunque es muy difícil predecir cuanto tiempo será necesario para alcanzar tal nivel de sensibilidad y eficiencia, lo que sí es verdad es que los investigadores trabajan sin cesar en mejorar cada uno de los elementos pertenecientes a los sistemas artificiales.

Una nariz electrónica es un instrumento que consiste en un conjunto de sensores electroquímicos con una especificidad parcial y un sistema adecuado de reconocimiento de patrones, capaz de reconocer olores simples o complejos.

2.3 SENSORES

Un sensor es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos aparatos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas.

Por ejemplo: existen sensores que se instalan en los vehículos y que detectan cuando la velocidad de desplazamiento supera la permitida. En esos casos, emiten un sonido que alerta al conductor y a los pasajeros.

Otro tipo de sensor muy habitual es aquel que se instala en la puerta de entrada de las viviendas y reacciona ante el movimiento. Si una persona se acerca al sensor, éste emite una señal y se enciende la iluminación. La utilización de estos sensores está vinculada a la seguridad, ya que evitan que alguien aproveche la oscuridad para ocultarse e ingrese en la casa sin ser advertido.

Los termómetros también son sensores ya que aprovechan la capacidad del mercurio para reaccionar ante la temperatura y, de este modo, permiten detectar si una persona tiene fiebre.

Los sensores, en definitiva, son artefactos que permiten obtener información del entorno e interactuar con ella. Así como los seres humanos apelan a su sistema sensorial para dicha tarea, las máquinas y los robots requieren de sensores para la interacción con el medio en el que se encuentran. [7]

2.3.1 SENSORES DE GAS

Los sensores químicos se pueden también clasificar en función del principio de operación que utilicen. Las propiedades más empleadas son conductividad, voltajes, capacidades, calentamientos, masas o constantes ópticas las cuales cambian al variar la concentración de las especies químicas que interactúan con el recubrimiento sensible del sensor. [7]



Fig. 3 Sensores de la familia Figaro

3.

Diseño de Componentes

3. DISEÑO DE COMPONENTES	14
3.1. Diseño PCB sensores.....	14
3.1.1. Diseño Schematic Capture	14
3.1.2. Diseño PCB Layout.....	16
3.2. Diseño PCB arduino.....	18
3.3. Diseño PCB Shield sensores arduino	18
3.4. Impreso de la Shield sensores MQ para arduino.....	20
3.5. Diseño caja modulo transmisor.....	21
3.6. Diseño caja modulo receptor.....	22

3. DISEÑO DE COMPONENTES

3.1 DISEÑO PCB SENSORES

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

Los circuitos impresos de los sensores para la E-nose metano, se diseñan con la ayuda del programa Proteus Design suite 8.4. La cual este programa cuenta con unas herramientas de trabajo, las cuales se emplean el área de diseño Schematic Capture y PCB Layout. Este software no cuenta con la PCB de este dispositivo electrónico (sensores MQ) dentro de su base de datos, la cual fue necesario crear su propia PCB, proteus permite la creación de nuevos modelos de dispositivos electrónicos. Este software es el que actualmente se utiliza en laboratorio de circuitos impresos del programa ingeniería electrónica de la universidad de pamplona.

3.1.1 DISEÑO SCHEMATIC CAPTURE

En la siguientes figura 4 y 5, podemos apreciar cómo se crea un nuevo modelo de un componente electrónico en el programa proteus.

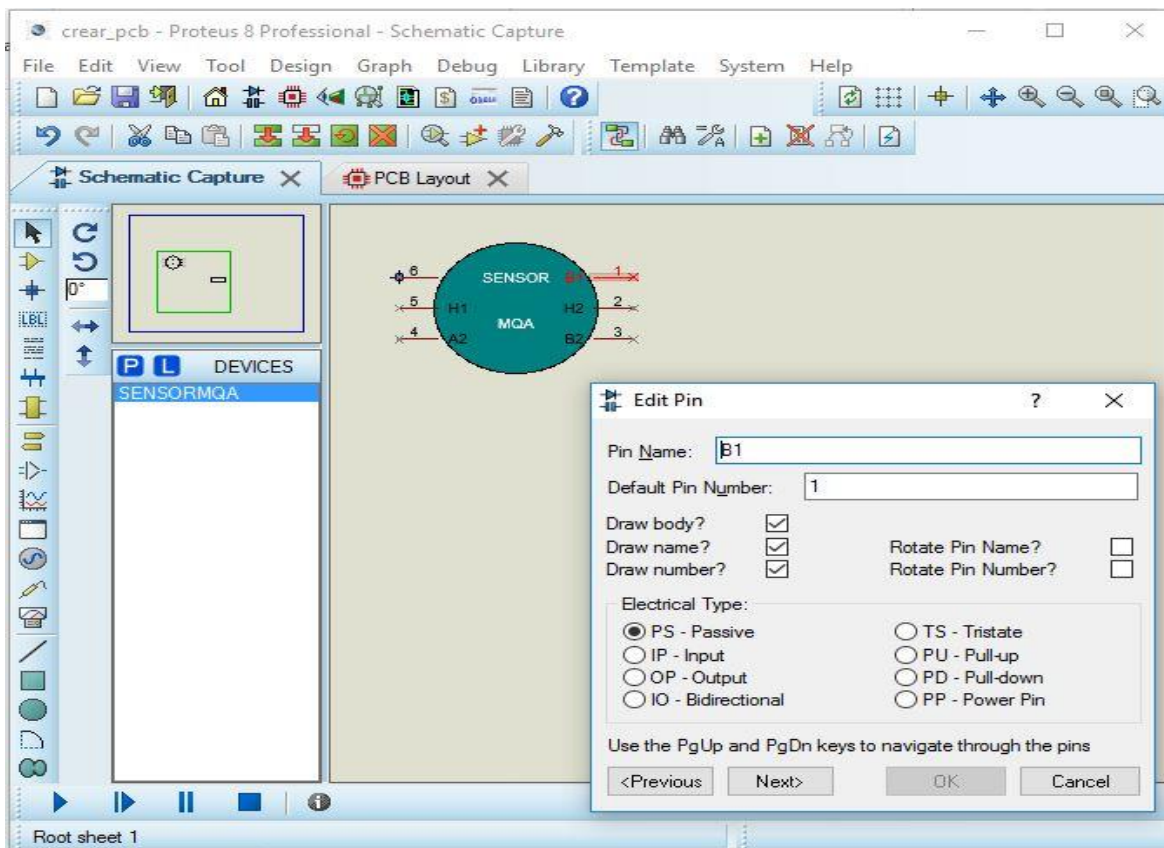


Fig. 4 Diseño de un nuevo modelo esquemático del sensor MQ. [8]

Haciendo clic sobre cada pin creado, se desprende una ventada llamada Edit Pin, donde se configuran los pines a utilizar. En la siguiente figura 5 muestra cómo se guarda el nuevo componente creado, haciendo click en el icono Make Device, donde

se desprende una ventana, se realizan los pasos correspondientes y se llenan los campos necesarios para luego guardarlo en la librería *USERDVC*.

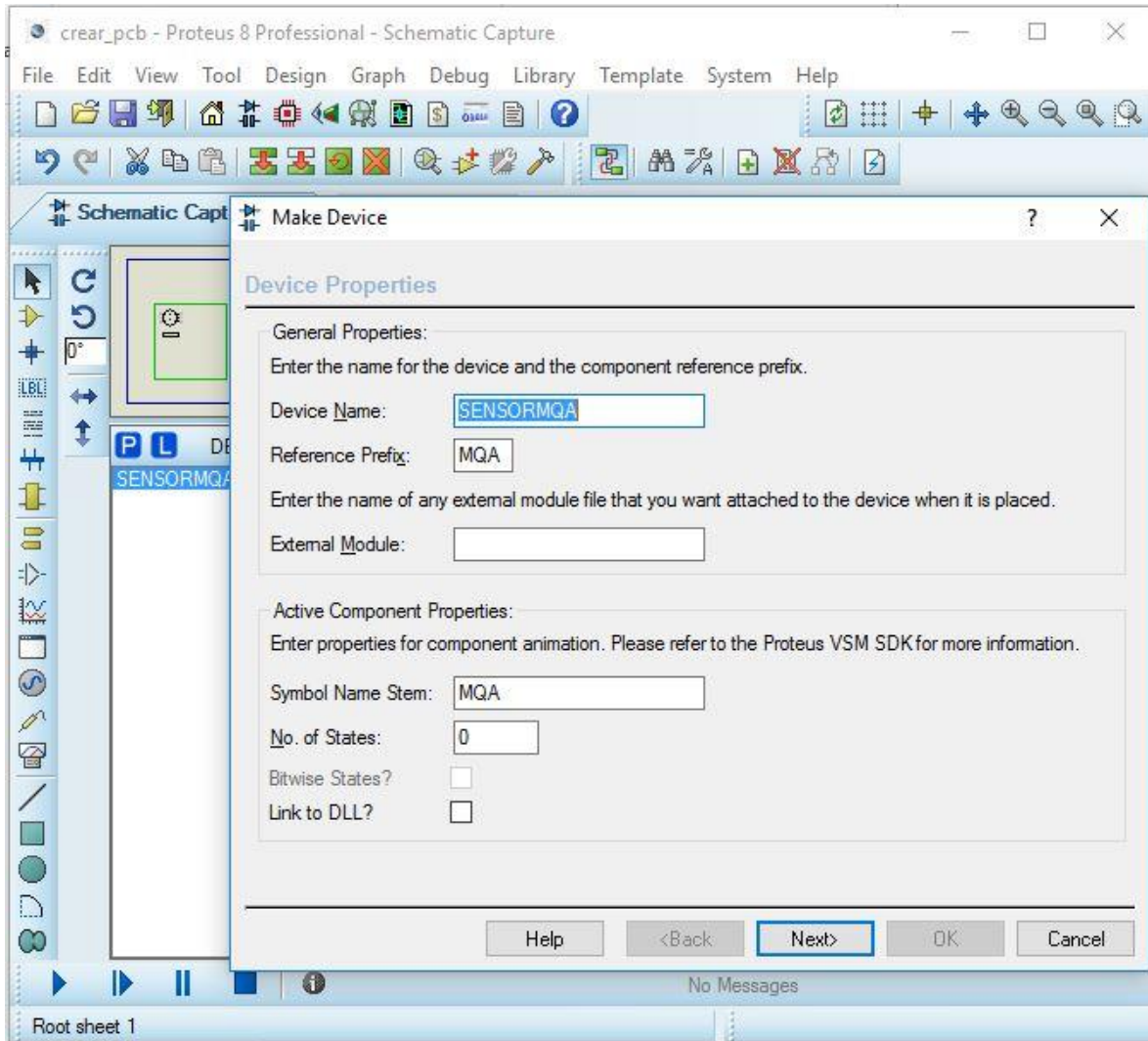


Fig. 5 Guardar el nuevo modelo esquemático creado. [8]

Una vez realizado todos los pasos anteriores de crear el nuevo componente, podemos utilizar nuestro modelo en cualquier otro diseño que deseemos ejecutar. En la figura 6, se puede observar como utilizamos nuestro componente creado.

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALAMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

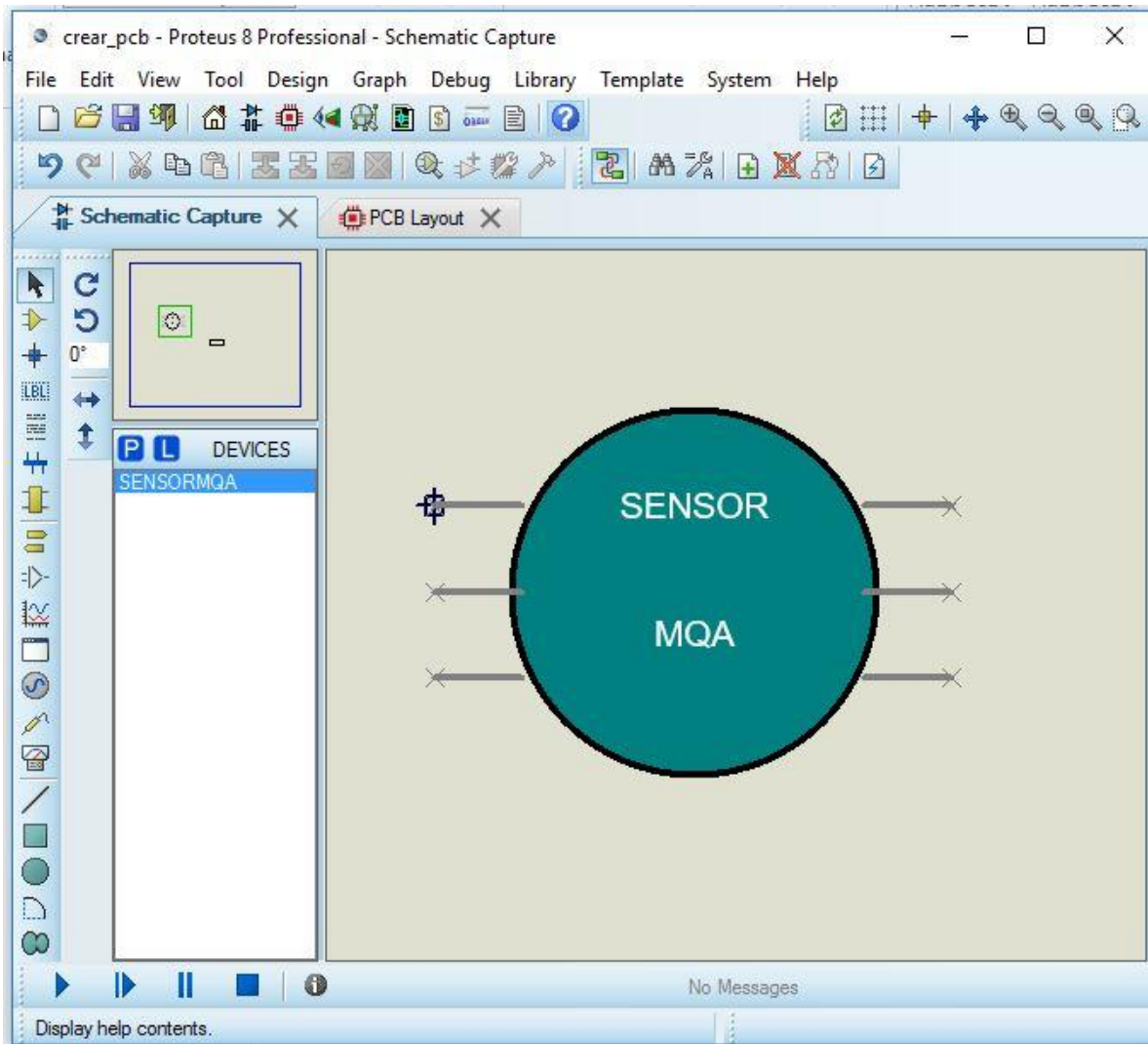


Fig. 6 Utilización del nuevo componente en Schematic Capture. [8]

3.1.2 DISEÑO PCB LAYOUT

Una vez creado nuestro componente, debemos asignarle el paquete de PCB, para poder realizar nuestro Shield de los sensores. En la siguiente figura 7, se analiza cómo se crea la PCB para el componente creado anteriormente.

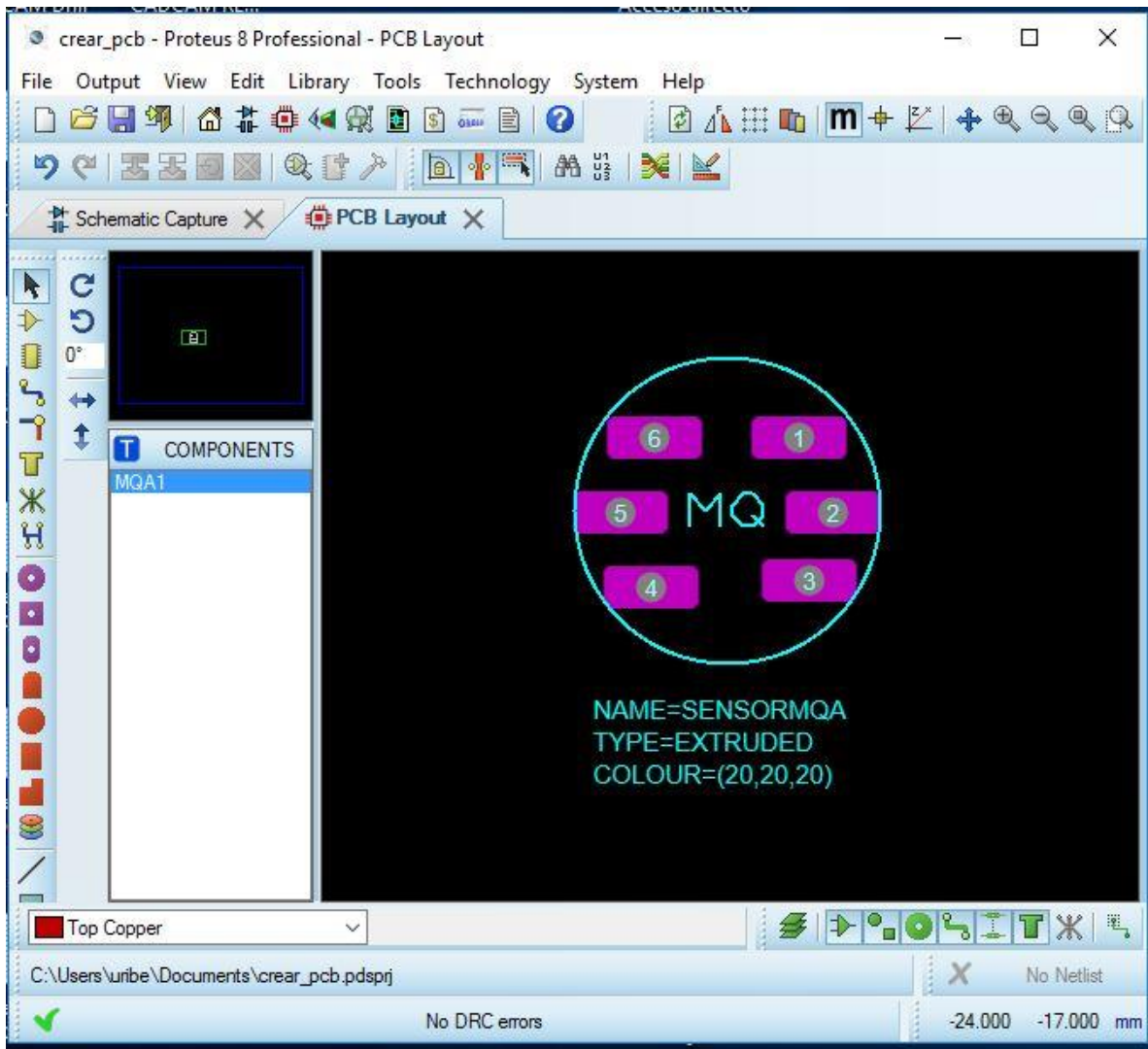


Fig. 7 Asignación de la PCB para el modelo creado. [8]

3.2 DISEÑO PCB ARDUINO

En la figura 8 se observa el modelo creado de la PCB para la tarjeta arduino uno.

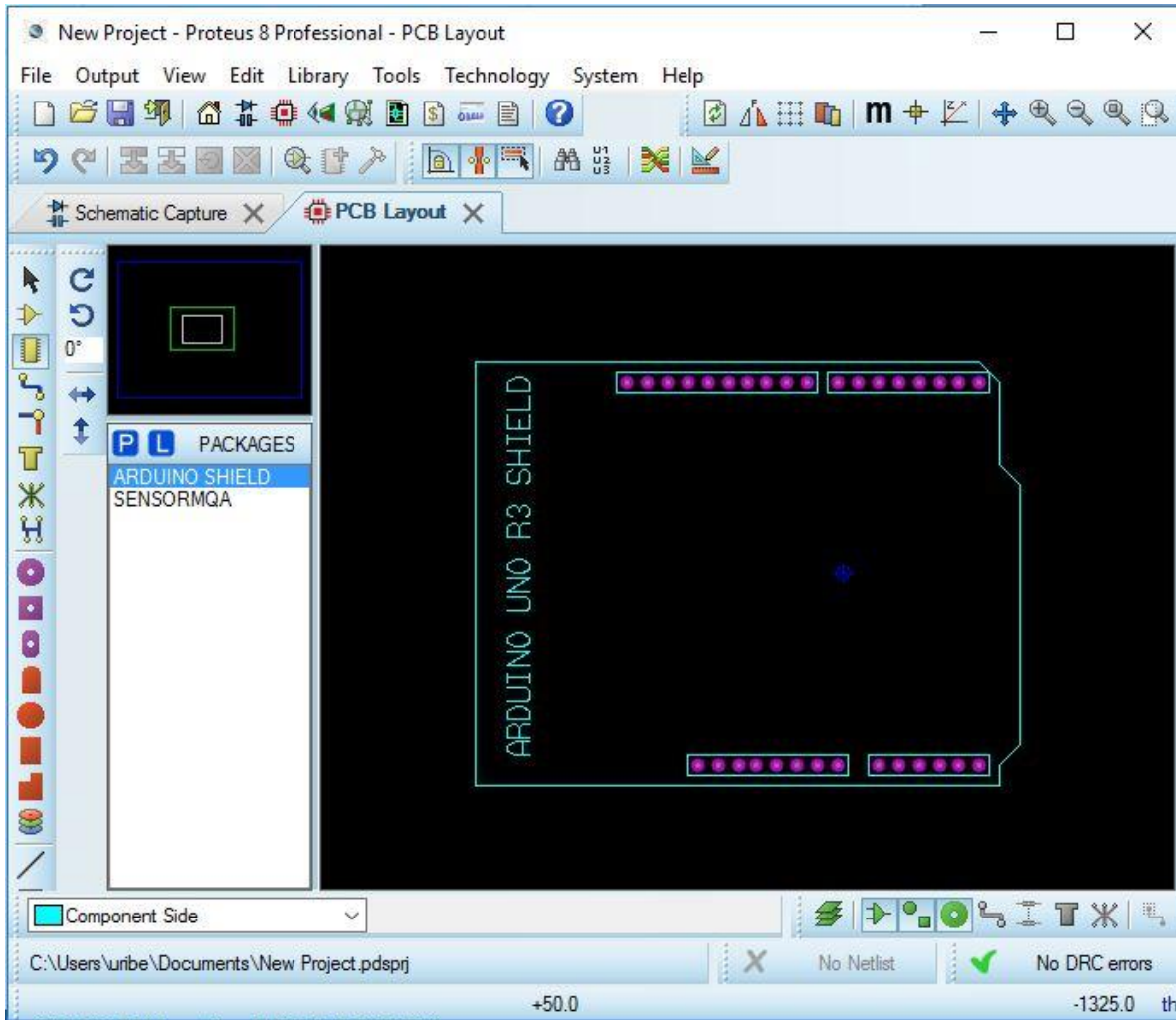


Fig. 8 PCB para la tarjeta Arduino uno. [8]

3.3 DISEÑO PCB SHIELD SENSORES ARDUINO

El circuito completo del sistema de la Shield sensores arduino se puede apreciar en las siguientes figuras 9 y 10, el *Schematic capture* y la *PCB Layout*.

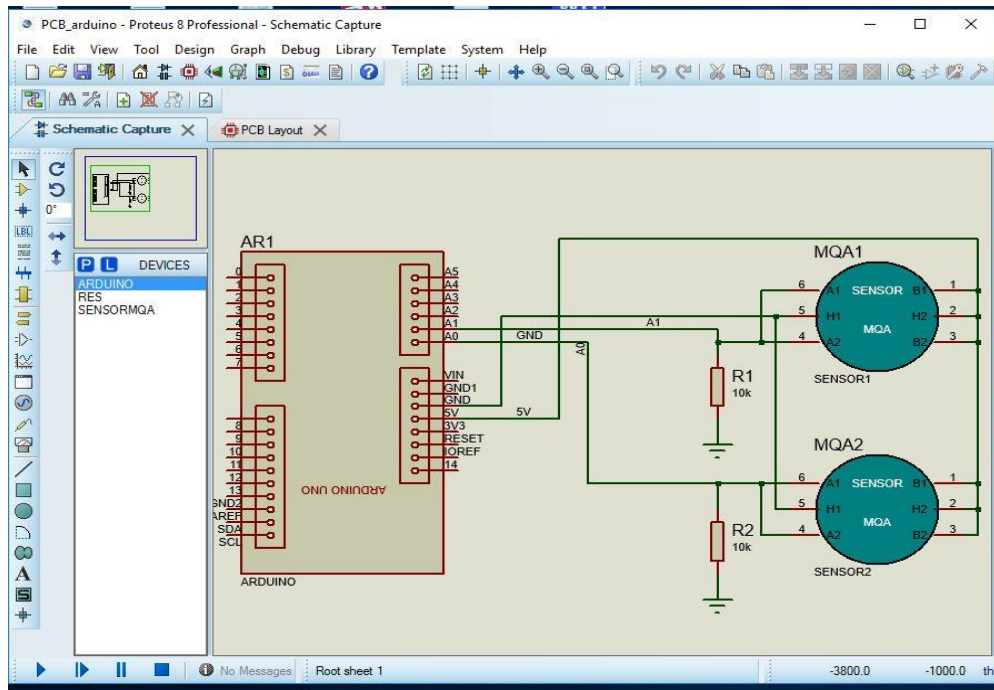


Fig. 9 Diseño sensores arduino Schematic capture. [8]

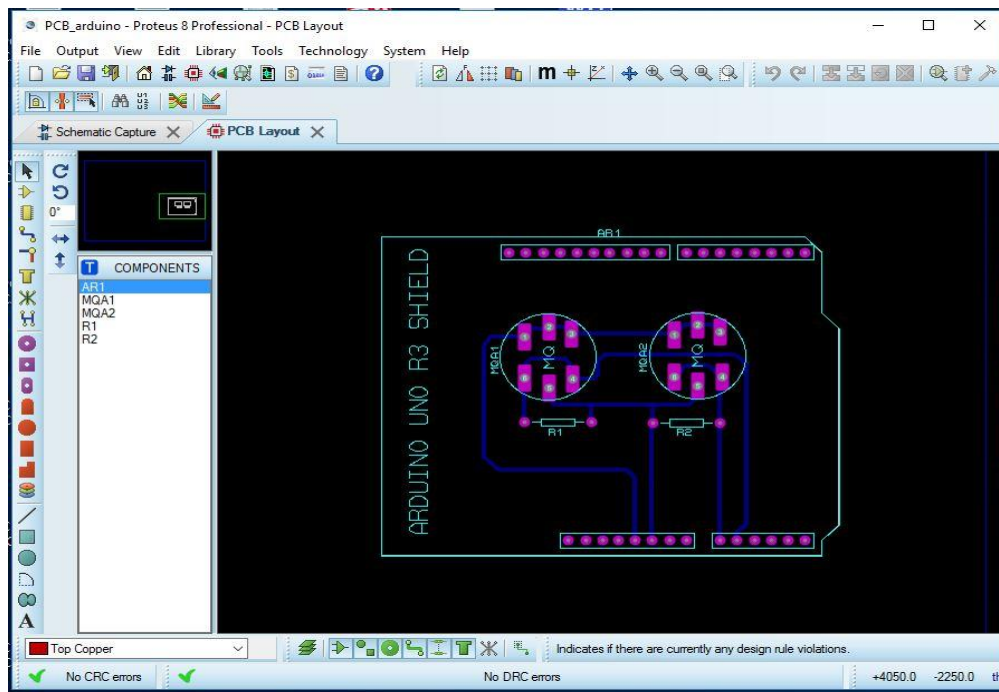


Fig. 10 Diseño sensores arduino PCB Layout. [8]

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

3.4 IMPRESO DE LA SHIELD SENSORES MQ PARA ARDUINO

Después de haber realizado nuestro diseño del circuito impreso del Shield sensores para arduino, este se ejecuta en el laboratorio de circuitos impresos del programa ingeniería electrónica de la universidad de pamplona. En la siguiente figura 11 se puede observar esta herramienta y en la figura 12 se observa la tarjeta de circuito impreso diseñado.



Fig. 11 Máquina de circuitos impresos.

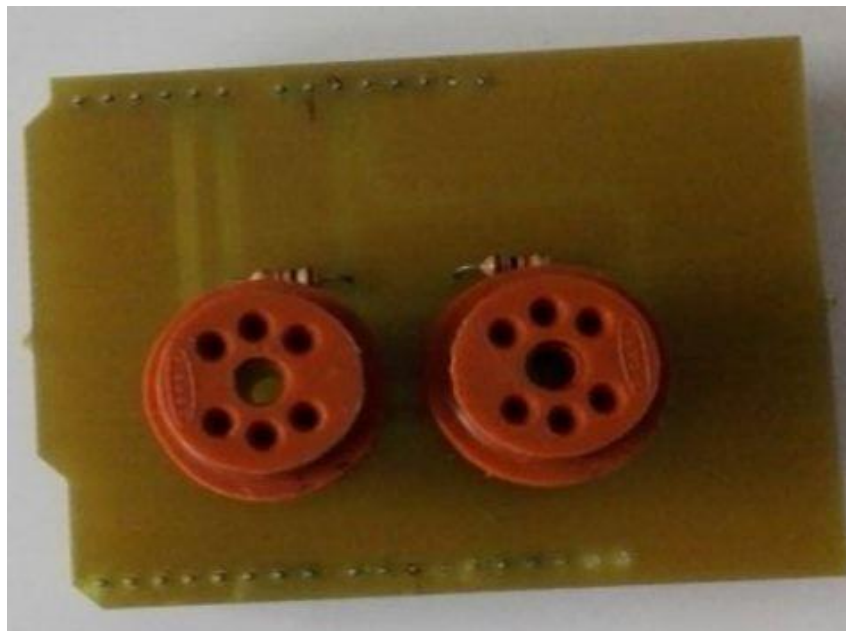


Fig. 12 Shield Sensores MQ para Arduino. [8]

3.5 DISEÑO CAJA MODULO TRANSMISOR

El diseño de esta caja, se puede ver en la siguiente figura 13, la cual se hace con el objetivo de proteger los módulos del transmisor.

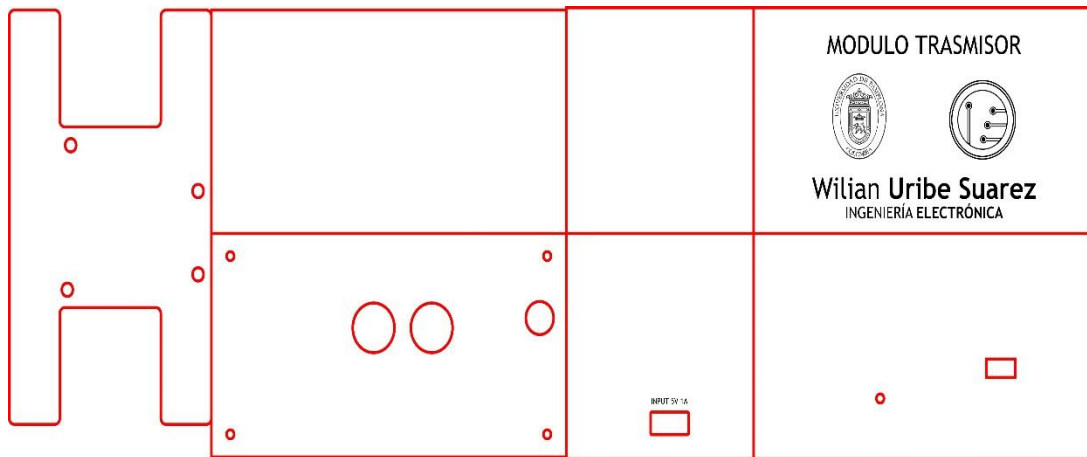


Fig. 13 Diseño caja modulo transmisor.

La caja del módulo transmisor ya construida se puede apreciar en la siguiente figurara 14.



Fig. 14 Caja modulo transmisor.

3.6 DISEÑO CAJA MODULO RECEPTOR

El diseño de la caja del módulo receptor, se puede apreciar en la siguiente figura 15.

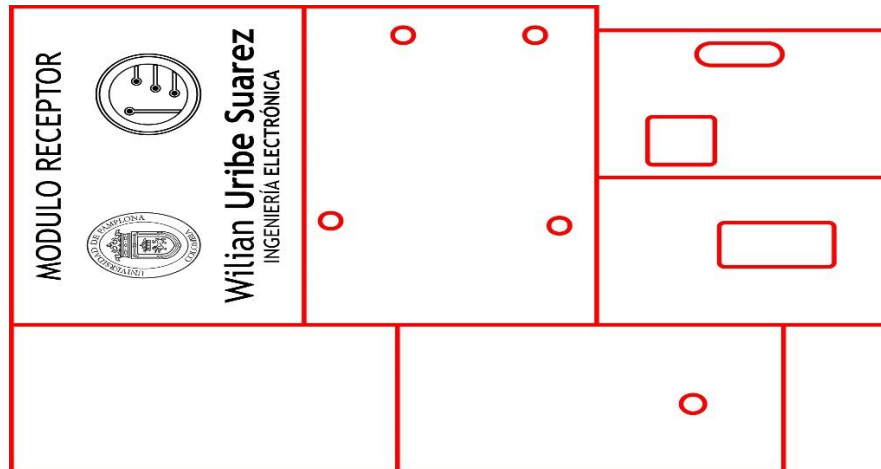


Fig. 15 Diseño de la Caja modulo receptor.

La caja del módulo receptor ya construida con los materiales correspondientes se puede observar en la siguiente figura 16.



Fig. 16 Caja modulo receptor.

3.7 COBERTURA RED MOVIL CLARO

El módulo de comunicación GPRS utiliza la SIM CAR del servicio de operador Claro, la cual la distancia entre el modulo transmisor y el modulo receptor para realizar una comunicación inalámbrica está determinada por la cobertura que ofrece esta red móvil. En la siguiente figura muestra el mapa de cobertura del operador claro y la calidad de la señal.

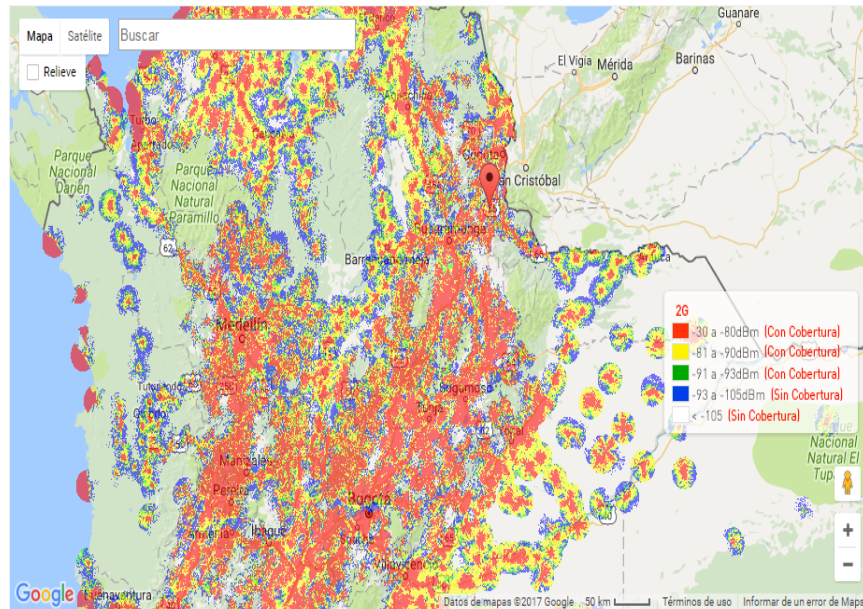


Fig. 17 Cobertura de red móvil Claro

La distancia geográfica (ruta aérea) entre Leticia y Riohacha es **1781 km**. La siguiente figura se puede observar dicha longitud.



Fig. 18 Distancia geográfica entre Leticia y Riohacha. [9]

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

Programación y Desarrollo

4. PROGRAMACION Y DESAROLLO	26
4.1. Tarjeta arduino uno.....	26
4.1.1. Características generales de arduino uno.....	26
4.2. Módulo GPRS SIM900.....	28
4.2.1. Características del módulo SIM900	28
4.2.2. Conexión del módulo SIM900.....	31
4.3. Comandos AT.....	32
4.4. Verificar módulo SIM900 con comandos AT.....	34
4.5. Ejemplos con el módulo SIM900 y arduino uno.....	37
4.5.1. Encender un LED con una llamada telefónica.....	37
4.6. Calibración de los sensores MQ.....	40
4.6.1. Sensor de gas MQ.....	40
4.6.2. Escalando los sensores MQ.....	41
4.7. Implementación del sistema de medición.....	46
4.7.1. Modulo Transmisor (Tx).....	48
4.7.1.1. Descripción del módulo transmisor (Tx).....	48
4.7.1.2. Diagrama de conexión del módulo (Tx).....	49
4.7.1.3. Programación del módulo transmisor (Tx).....	49
4.7.2. Modulo receptor (Rx).....	55
4.7.2.1. Descripción del módulo receptor (Rx).....	55
4.7.2.2. Diagrama de conexión del módulo (Rx).....	56
4.7.2.3. Programación del módulo receptor (Rx).....	57
4.8. Desarrollo de la interfaz gráfica en matlab.....	60
4.8.1. Programación arduino y matlab.....	60

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALAMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

4.8.2. Diseño de interfaz gráfica en guide matlab.....	62
4.9. Crear ejecutable en matlab.....	63

4. PROGRAMACION Y DESARROLLO

4.1 TARJETA ARDUINO UNO

La tarjeta arduino uno está basado en el microcontrolador ATmega328. Este tiene 14 pines digitales input/output (6 de estos pueden usarse como PWM outputs), 6 inputs análogas, un cristal de oscilación de 16 MHz, un conector tipo USB, un jack power, un botón reset. En la siguiente figura 17 vemos su aspecto físico de la tarjeta.

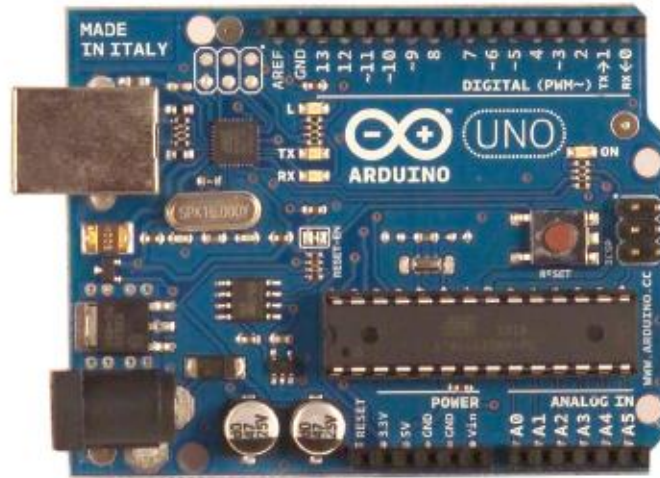


Fig. 19 Tarjeta Arduino uno. [10]

4.1.1 CARACTERÍSTICAS GENERALES DE ARDUINO UNO

En la siguiente tabla 1 se muestra las especificaciones técnicas más importantes de la tarjeta arduino uno.

Tabla 1 Especificaciones técnicas. [10]

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
<hr/>	
Clock Speed	16 MHz

Como se puede ver en la figura 18 se muestran una descripción de las partes más fundamentales que conforma la tarjeta arduino uno.

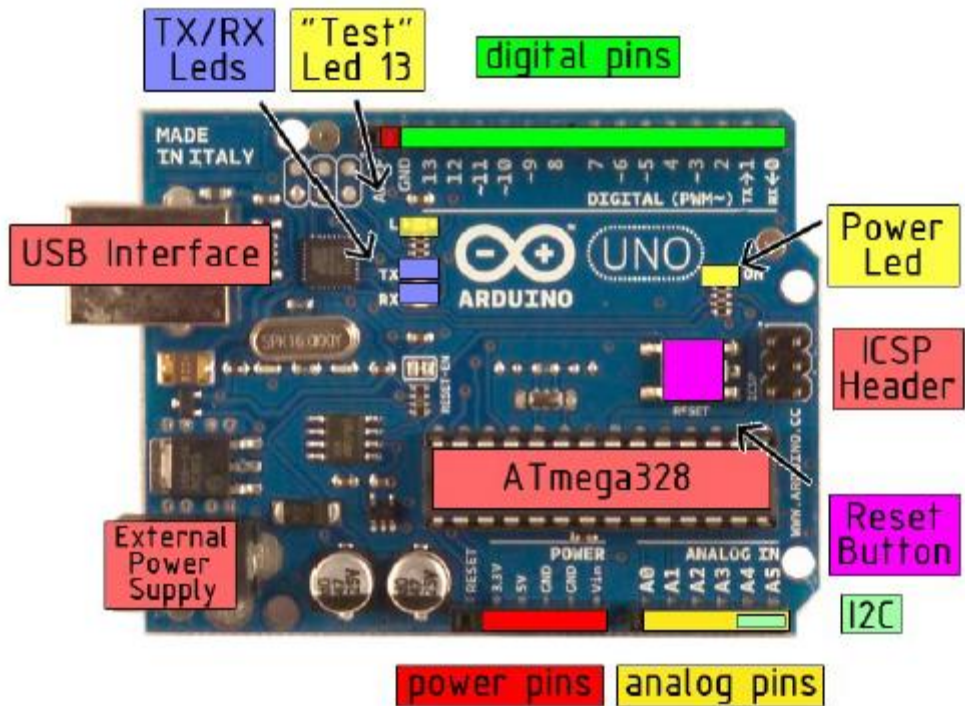


Fig. 20 Descripción de la Tarjeta arduino uno. [11]

4.2 MODULO GPRS SIM900



Fig. 21 Diagrama de conexión GPRS y Arduino.

El **GSM** (*Global System for Mobile Communications*) es el sistema de comunicaciones que más se utiliza en teléfonos móviles y es un estándar en Europa. Su principal función de esta tecnología es la de transmisión de voz, pero también transmite datos (SMS, Internet), a una velocidad muy baja de 9kb/s.

El **GPRS** (*General Packet Radio Service*) es una extensión del GSM basada en la transmisión por paquetes que ofrece un servicio más eficiente para las comunicaciones de datos, especialmente en el caso de acceso a internet. La velocidad máxima (en teoría) del GPRS es de 171kb/s.

4.2.1 CARACTERÍSTICAS DEL MODULO SIM900.

En las siguientes tablas se mencionan las características generales más importantes del módulo Shield GPRS SIM900. La cual es un módulo que permite el diseño de proyectos de comunicación y control inalámbrico con microcontroladores, por vía de la red celular e internet, ya sea utilizando mensajes cortos SMS o enlaces por vía GPRS.

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

Tabla 2 Características Generales. [12]

Quad-Band 850/900/1800/1900MHz
GPRS multi slot class 10/8
GPRS mobile station class B
Class 4 (2W@850/900MHz)
Class 1 (1W@1800/1900MHz)
Control via commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands)
Power on/off and reset function supported by Arduino interface

Tabla 3 Especificaciones. [12]

PCB size	77.2mm X 66.0mm X 1.6mm
Indicators	PWR, status LED, net status LED
Communication Protocol	UART

Tabla 4 Características Eléctricas. [12]

Specification	Min	Type	Max	Unit
Power Voltage(Vlogic)	4.5	5	5.5	VDC
Power Voltage(Vsupply)	9	-	20	VDC
Input Voltage VH:	4.5	5	5.5	V
Input Voltage VL:	-0.3	0	0.5	V
Current Consumption(pulse)	-	-	2000	mA
Current Consumption(Continues)			500	mA
Baud rate		9600		bps

Tabla 5 Indicador LED NETSTATUS_[13]

Estado	Descripción
Off	SIM900 is not running
64ms On/800ms Off	SIM900 not registered the network
64ms On/3000ms Off	SIM900 registered to the network
64ms On/300ms Off	GPRS communication is established

Tabla 6 Configuraciones SIM900. [13]

	Descripción
STATUS	Power status of SIM900.
PWR	Power status of IcomSat.
SIM900-POWER	After the IcomSat power on, you need to press the SIM900-POWER button for a moment to power on the SIM900 module. The pin 9 of Arduino interface is connecting to PWRKEY, and a high pulse with 400us wide can power on/off it
SIM90-RST	Reset the SIM900. The pin 10 of Arduino interface is connecting to RESRT of SIM900, and a high pulse with 400us wide can power on/off it.
ARDUINO-RST	Reset the Arduino

La siguiente figura 20 muestra un diagrama funcional de la SIM900:

- La GSM baseban engine
- Flash
- La GSM radio frecuencia
- La antena interface
- Otras interfaces

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

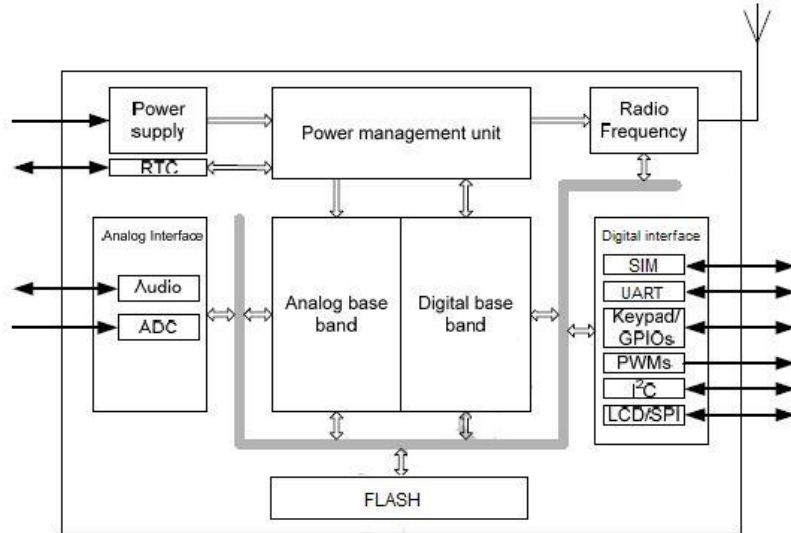


Fig. 22 Diagrama funcional de SIM900. [12]

4.2.2 CONEXIÓN DEL MODULO SIM900

En la siguiente figura 21, se observa el diagrama de conexión del módulo GPRS y el arduino uno, la cuales emplean los pines (Tx, Rx) la para la comunicación de los datos.

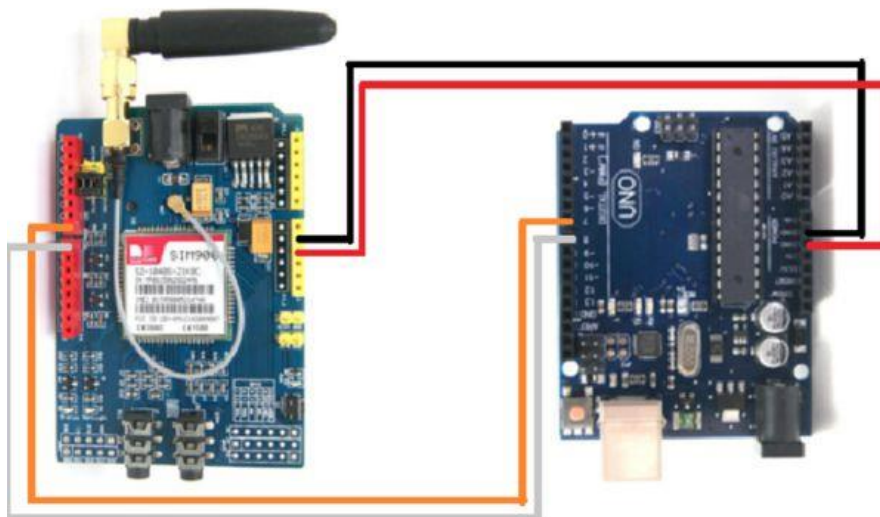


Fig. 23 Diagrama de conexión GPRS y Arduino.

El módulo SIM900, tiene un interruptor de dos posiciones, el cual permite la seleccionar los pines que vamos a utilizar para realizar la comunicación serial con nuestro arduino, en la siguiente imagen podemos observar el círculo rojo el cual indica su ubicación en la tarjeta.

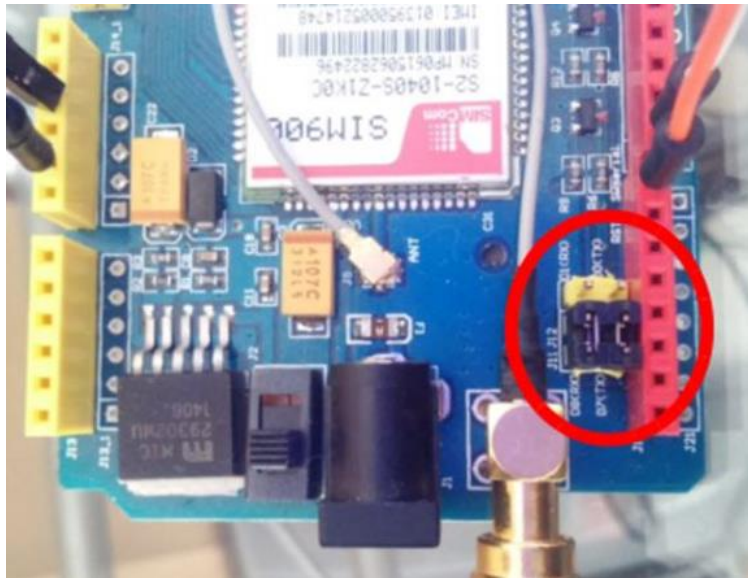


Fig. 24 Ubicación del interruptor que selecciona pines (Tx, Rx). [14]

La comunicación serial por defecto viene configurada a una velocidad de 19200 bps, 8-N-1, estos datos se pueden modificar a través de los comandos AT, la cual se explicaran más adelante.

4.3 COMANDOS AT

Los comandos AT, también conocidos como comandos Hayes (en honor a su desarrollador Dennis Hayes), son una serie de instrucciones que conforman un interfaz de comunicación entre usuario y modem. Estos comandos nos permiten ejecutar acciones como: marcar un número de teléfono, enviar o leer un SMS, consultar el estado de conexión a la red, leer o escribir en la agenda de contactos, etc. Al enviar un comando AT el SIM deberá contestarnos con un OK. Algunos de estos comandos se mencionan en las siguientes tablas.

Tabla 7 Comandos AT Generales. [15]

AT	DESCRIPCIÓN
AT+CGMI	Identificación del fabricante
ATI	Información del producto
AT+CIMI	Obtener el IMSI
AT+CGSN	Visualizar el IMEI del chip utilizado
AT+IPR=?	Preguntar el Baud Rate que opera la SIM
AT+IPR?	Preguntar Baud Rate actual
AT+IPR=XXXX	Configuramos a la frecuencia deseada
AT+COPS?	Nombre de la compañía telefónica
ATA	Contestar una llamada
ATH	Colgar una llamada
AT+CPAS	Leer estado del modem

Tabla 8 Comandos del servicio de red. [15]

AT	DESCRIPCIÓN
AT+CSQ	Obtener calidad de la señal
AT+COPS	Selección de un operador
AT+CREG	Registrarse en una red
AT+WOPN	Leer el nombre del operador
AT+GSV	Identificación producto revisión
ATH	Colgar una llamada

Tabla 9 Comandos de seguridad. [15]

AT	DESCRIPCIÓN
AT+CPIN	Introducir el PIN
AT+CPINC	Obtener el número de reintentos que quedan
AT+CPWD	Cambiar contraseña

Tabla 10 Comandos para la agenda de teléfonos. [15]

AT	DESCRIPCIÓN
AT+CPBR	Leer todas las entradas
AT+CPBF	Encontrar una entrada
AT+CPBW	Almacenar una entrada
AT+CPBS	Buscar una entrada

Tabla 11 Comandos para SMS. [15]

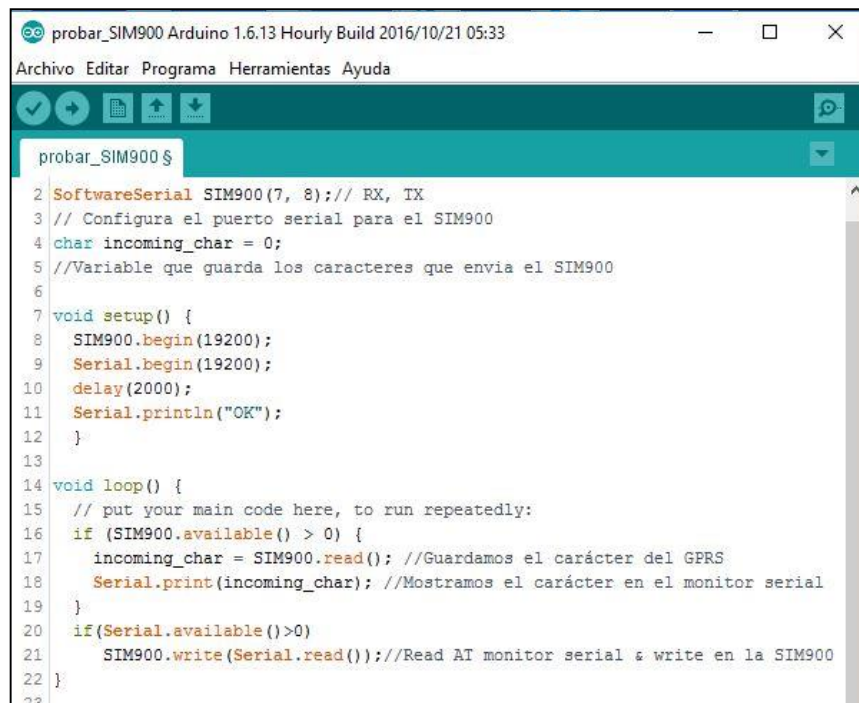
AT	DESCRIPCIÓN
AT+CPMS	Seleccionar lugar de almacenamiento de los SMS
AT+CMGF	Seleccionar formato de los mensajes SMS Modo = 0: indica formato de mensajes en modo PDU Modo = 1: indica formato de mensajes en modo TEXTO
AT+CMGR	Leer un mensaje SMS almacenado
AT+CMGL	Listar los mensajes almacenados
AT+CMGS	Enviar mensajes SMS
AT+CMGW	Almacenar mensaje en memoria
AT+ CMSS	Enviar mensaje almacenado
AT+CSCA	Establecer el Centro de mensajes a usar
AT+ WMSC	Modificar el estado de un mensaje
AT+CMGD	Borrar un SMS

4.4 VERIFICAR MÓDULO SIM900 CON COMANDOS AT

Materiales Necesarios:

- Arduino uno
- Shield GPRS SIM900
- Cable de datos

Para comprobar el correcto funcionamiento de nuestro módulo SIM900, se debe ejecutar el siguiente código en el IDE (Entornos de Desarrollo Integrado) de arduino, para poder realizar la interface de comunicación de datos entre arduino y el módulo SIM900. En la siguiente figura se aprecia el código.



```
probar_SIM900 $
2 SoftwareSerial SIM900(7, 8); // RX, TX
3 // Configura el puerto serial para el SIM900
4 char incoming_char = 0;
5 //Variable que guarda los caracteres que envia el SIM900
6
7 void setup() {
8   SIM900.begin(19200);
9   Serial.begin(19200);
10  delay(2000);
11  Serial.println("OK");
12 }
13
14 void loop() {
15  // put your main code here, to run repeatedly:
16  if (SIM900.available() > 0) {
17    incoming_char = SIM900.read(); //Guardamos el carácter del GPRS
18    Serial.print(incoming_char); //Mostramos el carácter en el monitor serial
19  }
20  if (Serial.available() > 0)
21    SIM900.write(Serial.read()); //Read AT monitor serial & write en la SIM900
22 }
23
```

Fig. 25 Sketch para verificar funcionamiento del módulo SIM900. [16]

En el código anterior de la figura 23 describe lo siguiente:

- Se configura el puerto serial para la SIM900, en la función *SoftwareSerial_nombre (RX, TX)*, asignan los pines para la comunicación serial entre la SIM900 y el arduino.
- Debemos colocar los jumpers de la SIM900 en los pines (D7, D8).
- En el void setup, se configura la Baud rate del monitor serial y la SIM900.

- En el void loop es la función que se encarga de repite el código, con estas líneas, mostramos los datos de la SIM900 en el monitor serial y leemos los comandos AT ingresados a través del monitor serial.
- El arduino se comunica con el PC a través de los pines digitales (PIN_0_Tx y PIN_1_Rx).

En la siguiente figura 24, vemos como se ingresan los comandos AT en el monitor serial y su repuesta respectiva. Lo único que debemos hacer ahora es abrir la ventana correspondiente al Monitor Serial, seleccionar la opción para el envío de caracteres de fin de línea y retorno de carro (*Ambos NL & CR*), la opción baudios y comenzar a transmitir los comandos.

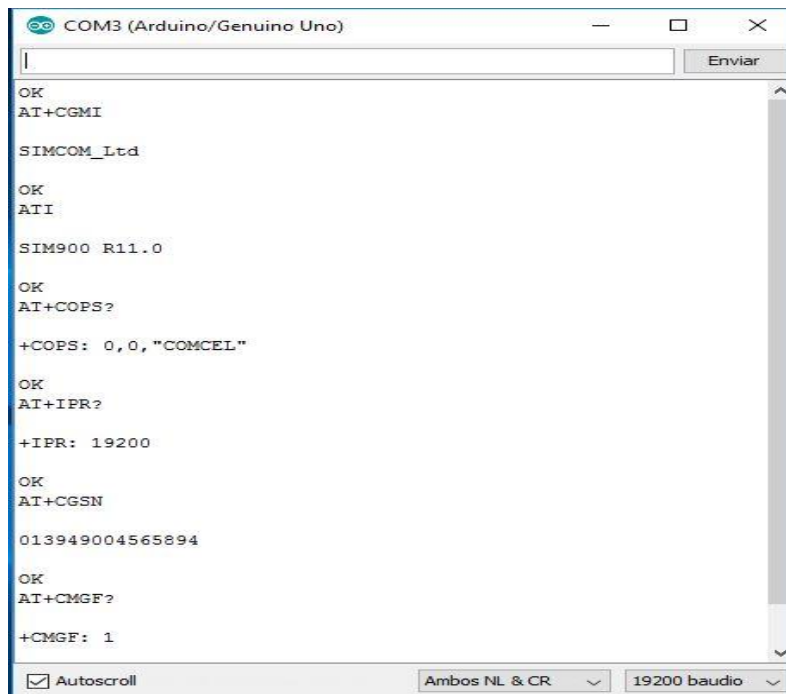


Fig. 26 Monitor serial, comandos AT del módulo SIM900. [16]

Con estos comandos podemos verificar el correcto funcionamiento de nuestro módulo SIM900, estos comandos muestra información importante como: el nombre del fabricante, nombre del producto, nombre de la compañía telefónica, la velocidad de transmisión (Baute Rate), el IMEI del chip utilizado y el tipo de formato del mensaje. Además podemos comprobar todos los demás comandos AT.

En la siguiente tabla 12 resume una lista de los comandos principales y su respuesta esperada cuando se ejecutan.

Tabla 12 Comandos AT y su respuesta. [15]

SYNTAX	EXPECT RESULT
AT	OK
ATI	SIM900 R11.0
AT+CFUN?	OK +CFUN:1
AT+IPR?	OK +IPR:19200
AT+CMGI	OK SIMCOM_Ltd
AT+COPS?	OK +COPS: 0,0 "COMCEL"
AT+CMGF	OK +CMGF: 1
AT+CGSN	013949004565894 OK

4.5 EJEMPLOS CON EL MODULO SIM900 Y ARDUINO UNO

4.5.1 ENCENDER UN LED CON UNA LLAMADA TELEFONICA. [14]

En el siguiente ejemplo vemos cómo podemos encender y apagar un LED mediante una llamada telefónica, utilizando el módulo SIM900 y arduino uno.

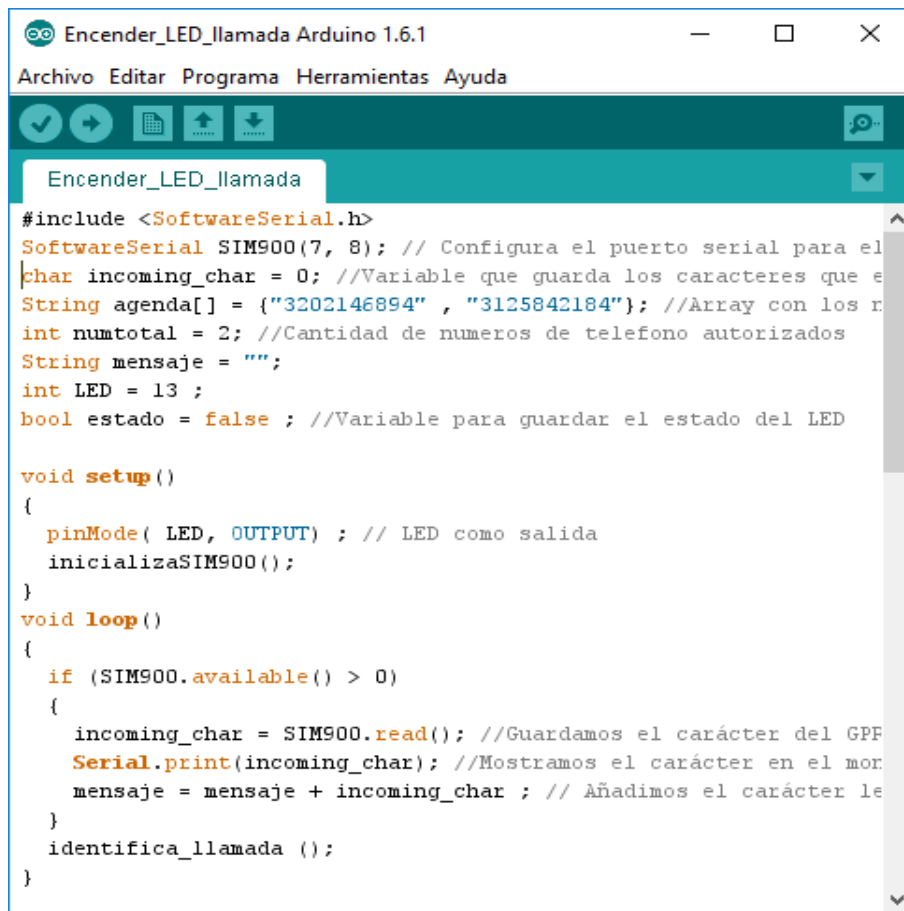
Materiales necesarios:

- Arduino uno
- Módulo SIM900
- Cable de datos
- Fuente de alimentación externa (5V)
- Conectores para arduino

El objetivo principal del programa es: recibir una llamada telefónica, leer el número y compararlo con unos números previamente guardados, si el número está autorizado cambia el estado de un *LED* cada vez que recibe una llamada.

Primero debemos incluir la librería `<SoftwareSerial.h>` y configurar el puerto serial para la *SIM900*. Como tenemos los jumpers en los pines *D7* y *D8*, asignamos esos pines y creamos el objeto para la comunicación serial. Creamos una variable agenda donde se guardaran los números de teléfono autorizados y otra variable mensaje donde almacenen los datos que llegan del *GPRS*.

El código principal para realizar este proyecto se puede apreciar en la siguiente figura 25.

The image shows a screenshot of the Arduino IDE interface. The window title is 'Encender_LED_llamada Arduino 1.6.1'. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. Below the menu bar is a toolbar with icons for file operations and a search icon. The main editor area shows the code for the 'Encender_LED_llamada' sketch. The code includes the 'SoftwareSerial' library, configures the SIM900 module, and defines variables for incoming characters, authorized numbers, total number of authorized numbers, LED pin, and LED state. The 'setup' function initializes the LED and the SIM900 module. The 'loop' function checks for available data from the SIM900, reads the character, prints it, and appends it to the message string. A function 'identifica_llamada()' is called at the end of the loop.

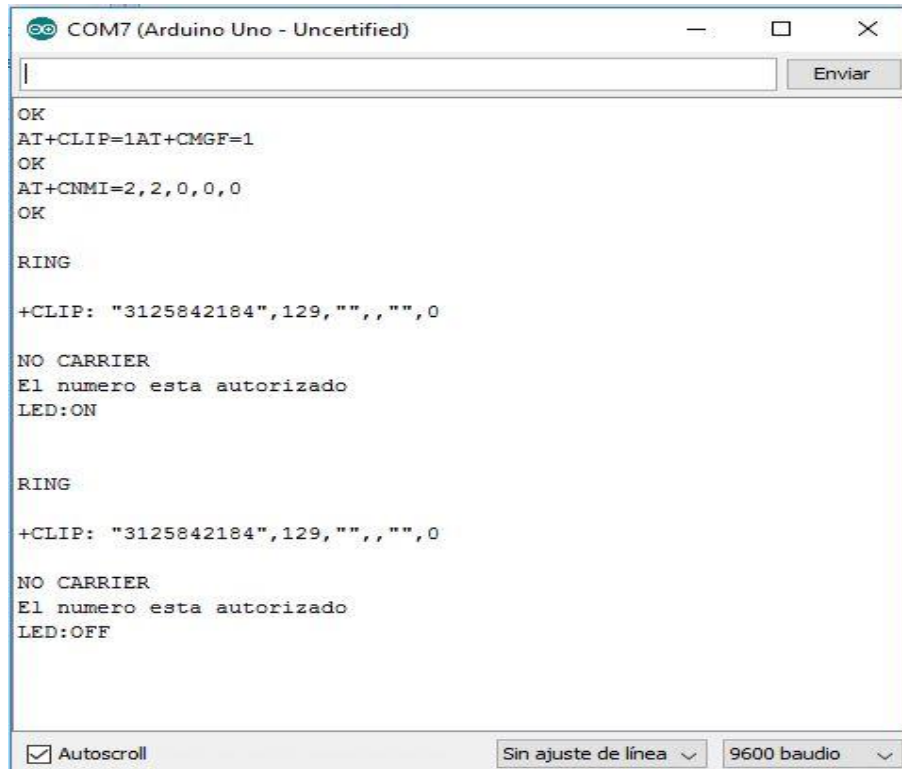
```
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8); // Configura el puerto serial para el
char incoming_char = 0; //Variable que guarda los caracteres que e
String agenda[] = {"3202146894" , "3125842184"}; //Array con los r
int numtotal = 2; //Cantidad de numeros de telefono autorizados
String mensaje = "";
int LED = 13 ;
bool estado = false ; //Variable para guardar el estado del LED

void setup()
{
  pinMode( LED, OUTPUT) ; // LED como salida
  inicializaSIM900();
}
void loop()
{
  if (SIM900.available() > 0)
  {
    incoming_char = SIM900.read(); //Guardamos el carácter del GPF
    Serial.print(incoming_char); //Mostramos el carácter en el mor
    mensaje = mensaje + incoming_char ; // Añadimos el carácter le
  }
  identifica_llamada ();
}
```

Fig. 27 Encender LED mediante llamada telefónica. [16]

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

En la figura 26 vemos el monitor serial de arduino IDE (Entornos de Desarrollo Integrado), podemos observar cuando se activa y desactiva el LED mediante una llamada telefónica.



```
COM7 (Arduino Uno - Uncertified)
Enviar
OK
AT+CLIP=1AT+CMGF=1
OK
AT+CNMI=2,2,0,0,0
OK
RING
+CLIP: "3125842184",129,"",,,",0
NO CARRIER
El numero esta autorizado
LED:ON
RING
+CLIP: "3125842184",129,"",,,",0
NO CARRIER
El numero esta autorizado
LED:OFF
Autoscroll Sin ajuste de línea 9600 baudio
```

Fig. 28 Monitor serial encender LED mediante llamada.

La función *identifica_llamada()* realiza lo siguiente:

- Cuando realizamos una llamada finaliza con "NO CARRIER", lo primero que haremos es buscar si existe, para esto vamos a utilizar la función *cadena.indexOf()*. Esta función nos devuelve la posición, mediante un número entero, de una cadena o carácter dentro de una cadena de texto dada. En el caso que no exista devuelve un "-1". La forma de usarla es la siguiente:

```
int pos = "texto".indexOf("caracteres a buscar");
if (pos >= 0)
{
    Acciones que queremos ejecutar;
}
```

Fig. 29 Función *.indexOf*

- La función *identifica_llamada()* primero busca el fin de la llamada y después el número de teléfono si está autorizado. Si está autorizado cambiamos el estado del *LED*, y sino no hacemos nada. Una vez hecho todo esto, borramos el contenido de la variable mensaje para dejarla lista para el siguiente llamada. La función completa la podemos ver en la siguiente figura.

```
void identifica_llamada()
{
  int finllamada = mensaje.indexOf("NO CARRIER");
  if (finllamada >= 0)
  {
    for ( int i = 0; i < numtotal; i++)
    {
      int pos = mensaje.indexOf(agenda[i]);
      if (pos >= 0)
      {
        estado = ! estado ; // cambiamos el estado del LED
        digitalWrite(LED, estado) ;
        Serial.println("\nEl numero esta autorizado");
        if (estado)
          Serial.println("LED:ON");
        else
          Serial.println("LED:OFF");
      }
    }
    mensaje = "" ; //Bórralo para la próxima vez
  }
}
```

Fig. 30 Función *identifica_llamada ()*.

4.6 CALIBRACION DE LOS SENSORES MQ

4.6.1 SENSOR DE GAS MQ

Estos sensores son electroquímicos y varían su resistencia cuando se exponen a determinados gases, internamente posee un calentador encargado de aumentar la temperatura interna y con esto el sensor pueda reaccionar con los gases provocando un cambio en el valor de la resistencia. El calentador dependiendo del modelo puede necesitar un voltaje entre 2 y 5 voltios, el sensor se comporta como una resistencia y necesita una resistencia de carga (RL) para cerrar el circuito y con este hacer un divisor de tensión y poder leerlo desde un microcontrolador.

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL



Fig. 31 Sensores de gas MQ. [17]

Tabla 13 MQ series Semiconductor Gas Sensor. [17]

Semiconductor sensor for Flammable gas, Plastic or Metal cover	
MODEL	TARGET GAS
MQ-2	General combustible gas
MQ-3	Alcohol
MQ-4	Natural gas, Metano
MQ-5	Natural gas, Coal gas
MQ-6	LPG, Propano
MQ-7	Carbón Monóxido
MQ-8	Hidrogeno
MQ-9	CO and Combustible gas
MQ306	LPG, Propano
MQ307	Carbón Monóxido
MQ303	Alcohol
MQ131	Ozone
MQ135	Air Quality Control
Semiconductor sensor for Toxic gas	
MQ136	Sulfureted Hydrogen (H ₂ S)
MQ137	Ammonia (NH ₃)
MQ138	VOC (Mellow, Benzene, Aldehyde, Ketone, Ester)

4.6.2 ESCALANDO LOS SENSORES MQ

En nuestra aplicación que estamos implementando necesitamos los valores en unidades correspondientes a la medición del gas, necesitamos escalar el valor leído, el inconveniente de esto es que la relación entre la lectura analógica y el valor real no es lineal. Por lo que necesitamos estimar la curva que nos da el datasheet del sensor MQ.

Los siguientes pasos los trabajamos para el MQ-4, pero se aplica análogamente para los otros MQ. Por Ejemplo para el sensor MQ-4 según el datasheet la curva es la siguiente:

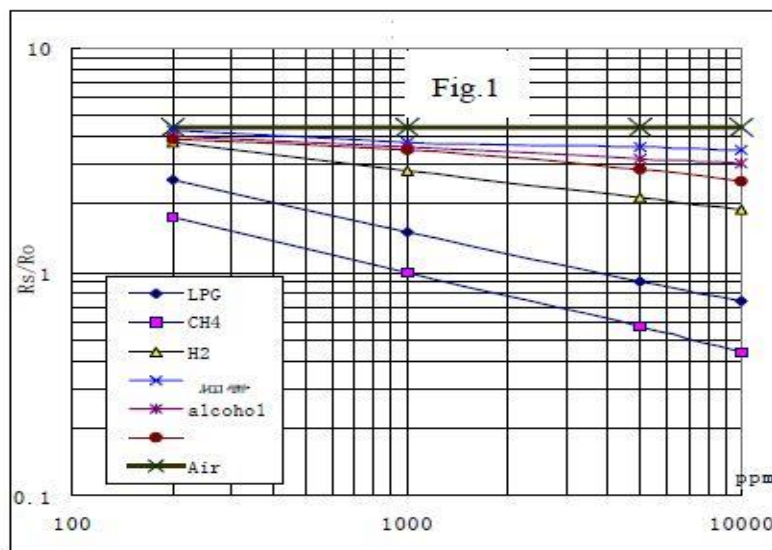


Fig. 32 Sensitivity characteristics of the MQ-4. [17]

En la figura 30 se observar las características típicas del sensor MQ-4. En el eje de las ordenadas muestra la relación de la resistencia promedio del sensor (R_s/R_o), en el eje de las abscisas la concentración del gas, R_o es la resistencia promedio del sensor a 1000 ppm de metano.

Debido a que nos dan la curva y no la ecuación es necesario estimar y por regresión hallar la ecuación, en nuestro caso usaremos Excel, para eso ingresamos datos de la curva de MQ-4, la mayor cantidad de puntos que podamos, y graficamos en Excel. Agregamos línea de tendencia y escogemos ecuación potencial.

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

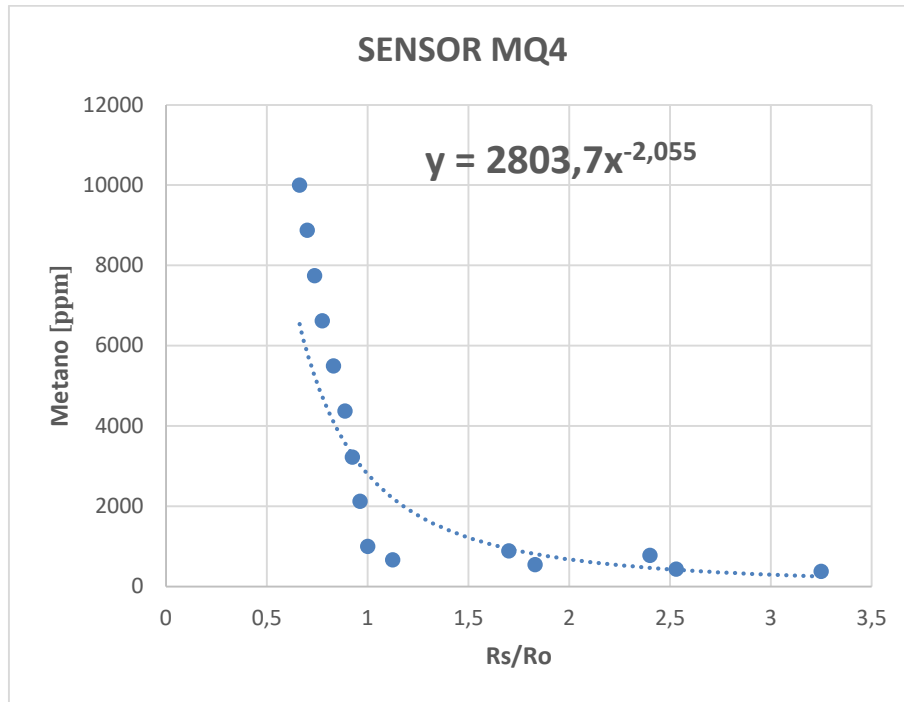


Fig. 33 Regresión y ecuación del sensor MQ-4

La ecuación que obtenemos es la siguiente:

$$\text{Metano} = 327,58 * \left(\frac{R_s}{R_o}\right)^{-1,655} \quad (\text{Ec. 1})$$

Donde:

Metano = Es el valor de la concentración del Gas metano

Ro= Constante que equivale al valor de la resistencia del sensor cuando se le expone a una concentración de 1000 ppm

Rs= Es la resistencia del sensor, la cual se lee desde el arduino

Para calcular el valor de R_s despejamos la ecuación del divisor de voltaje que forma el sensor con la resistencia de carga R_L que en este caso es de un 1 K Ω .

$$V_L = 5 * \frac{1000}{R_s + 1000} \quad (\text{Ec. 2})$$

$$R_s = 1000 * \frac{5 - V_L}{V_L} \quad (\text{Ec. 3})$$

Un método para conseguir de forma aproximada el valor de R_o y así tener la ecuación resuelta es el siguiente.

En el gráfico de la figura 33 observamos la ecuación potencial, el valor de R_s/R_o se acerca a 0.66 para valores superiores a la máxima concentración de Metano que el

sensor puede pensar, en los puntos cercanos a este punto, la variación de R_s/R_o es mínima por lo que los errores que podamos tener acá para calcular R_o son pequeños.

Entonces asumiendo que el sensor se satura con 1000 ppm al cual según la gráfica le corresponde un R_s/R_o de 0.66, los puntos cercanos por mayor variación de metano que exista el R_s/R_o tendrán variaciones mínimas. Para estar en este punto generamos un ambiente con bastante metano y si medimos con nuestro sensor obtendremos un voltaje de 4.03, que equivale a un $R_s=240,6$ entonces:

$$(R_s/R_o)=0.7 \text{ si } R_s = 655; \text{ entonces } R_o = 343,84$$

Con R_o calculado, ya tenemos una solución de la ecuación y con esto podemos obtener los valores aproximados de concentración de Metano.

Tabla 14 Datos de Calibración del sensor MQ-4

	VRL	VC	RL	RS	RS/RO	RO
MQ4	4,03 V	5 V	1000 Ω	240,69 Ω	0,7 Ω	343,84 Ω

Para probar la calibración y funcionamiento de los sensores se utilizó la tarjeta de arduino y la Shield de sensores de MQ-4 previamente diseñada. En la figura 34 se muestra el esquemático diseñado con la herramienta proteus.

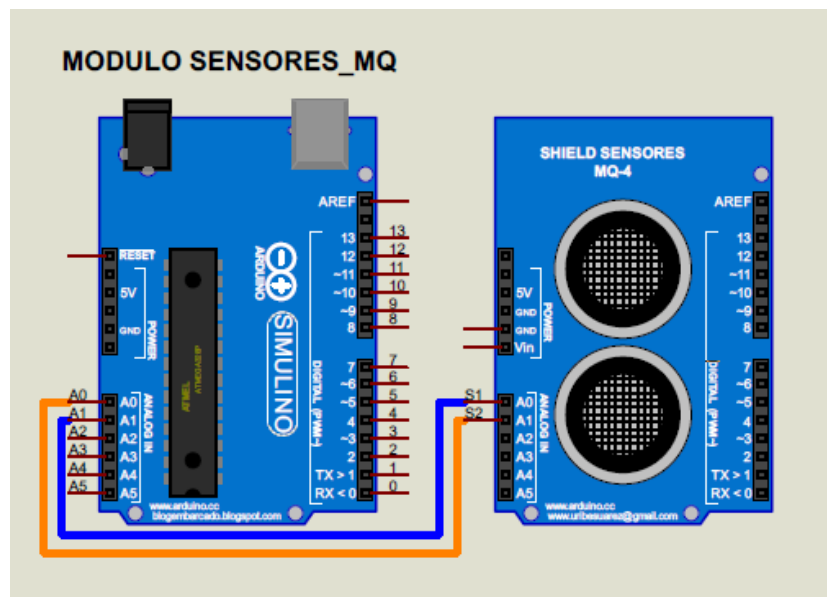


Fig. 34 Diagrama esquemático para leer el sensor MQ-4. [16]

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

Luego realizamos el siguiente código en el IDE de arduino para comprobar el funcionamiento y respuesta del sensor, el código de programación es el siguiente que podemos observar en la figura 35.

```
Read_sensor_MQ4 $
1 int MQ4=0;
2 float VRL, Rs =0;
3 double metano=0;
4 void setup() {
5   // put your setup code here, to run once:
6   Serial.begin(9600);
7 }
8
9 void loop() {
10  // put your main code here, to run repeatedly:
11
12  while (1) {
13    //for (int i=0;i<5;i++){
14    MQ4 = analogRead(A0); // METANO
15
16    VRL = MQ4 * (5.0 / 1023.0); //conversion en valor de voltaje
17    Rs = 1000 * ((5 - VRL) / VRL); // Calculamos Rs con un RL de 1k
18    metano = 2803.7 * pow(Rs / 343, -2.055); // calculamos la concentracion ppm
19
20    ///----- Enviamos los valores por el monitor serial -----///
21    imprimir();
22    delay(1000);
23  }
24 }
```

Fig. 35 Programa para leer el sensor MQ-4. [16]

En el monitor serial del programa *IDE* de arduino figura 36 podemos observar los datos capturados con el sensor *MQ-4* y ver su comportamiento de la ecuación obtenida para calcular la concentración del gas metano en unidades *ppm*, el valor de la entrada analógica *A0*, el valor de *Rs* y el voltaje *VRL* en la resistencia *RL*.

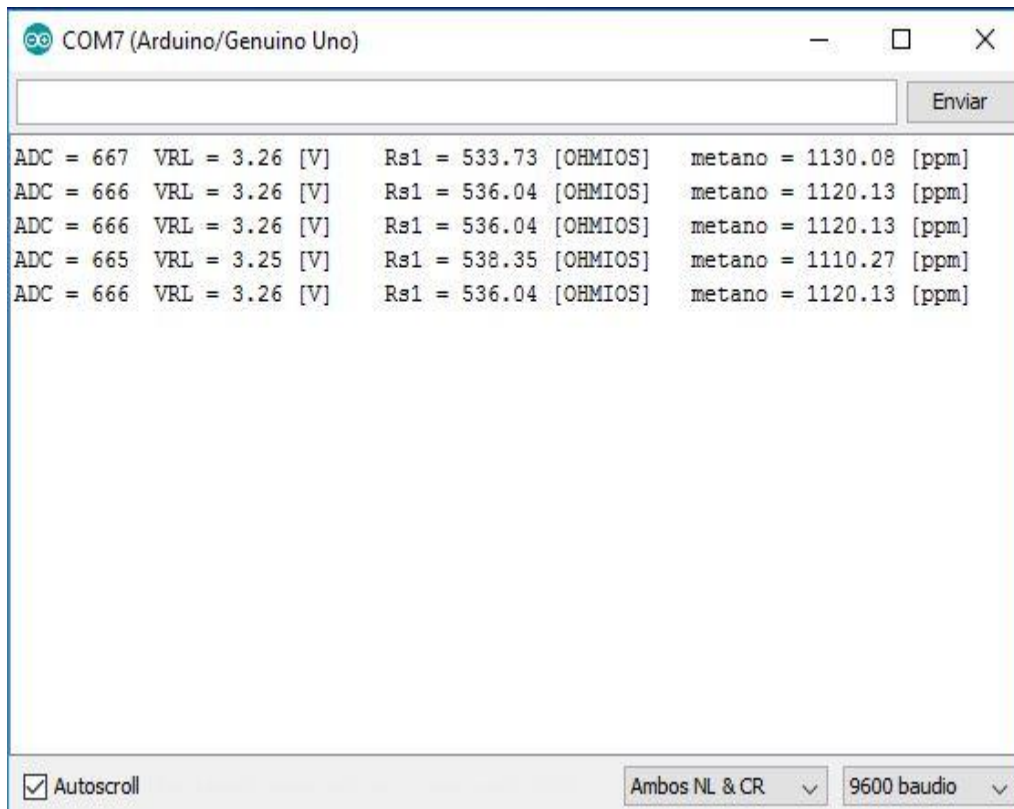


Fig. 36 Monitor serial valores obtenidos del sensor MQ-4. [16]

4.7 IMPLEMENTACION DEL SISTEMA DE MEDICION DE GAS METANO

En la figura 37 se muestra un diagrama de bloques, el cual explica la configuración del sistema de medición de gas metano y transmisión inalámbrica de los datos a un punto de control, donde se almacena los datos capturados y su procesamiento respectivo.

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

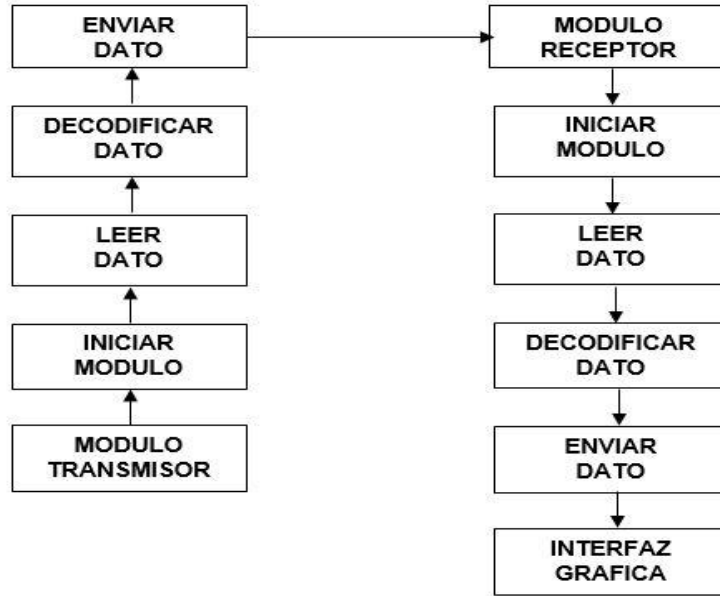


Fig. 37 Diagrama de bloques del sistema de comunicación

El sistema de comunicación implementado lo podemos apreciar en la siguiente figura 38.



Fig. 38 Sistema de comunicación inalámbrico

4.7.1 MODULO TRANSMISOR (TX)

En este módulo se requirió de los siguientes materiales para la elaboración de esta etapa de comunicación.

- Arduino uno
- Shield GPRS SIM900
- Shield sensores MQ
- Fuente de alimentación externa
- Conectores

4.7.1.1 DESCRIPCION DEL MODULO TRANSMISOR (TX)

El módulo de transmisión (Tx) es el encargado de transmitir los datos capturados por los sensores, la cual su procedimiento se muestra en el siguiente diagrama de bloques.

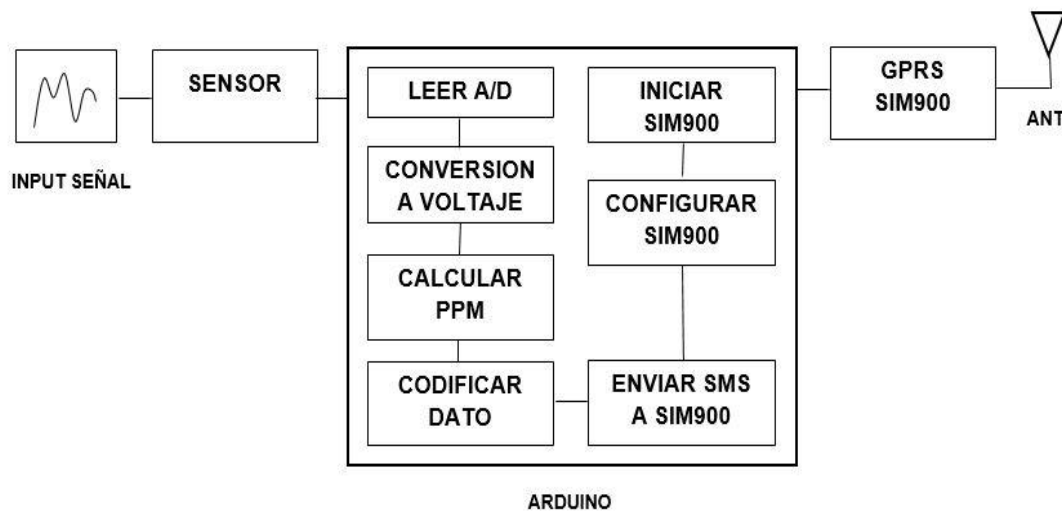


Fig. 39 Diagrama de bloques del módulo transmisor (Tx)

4.7.1.2 DIAGRAMA DE CONEXIÓN DEL MODULO (TX)

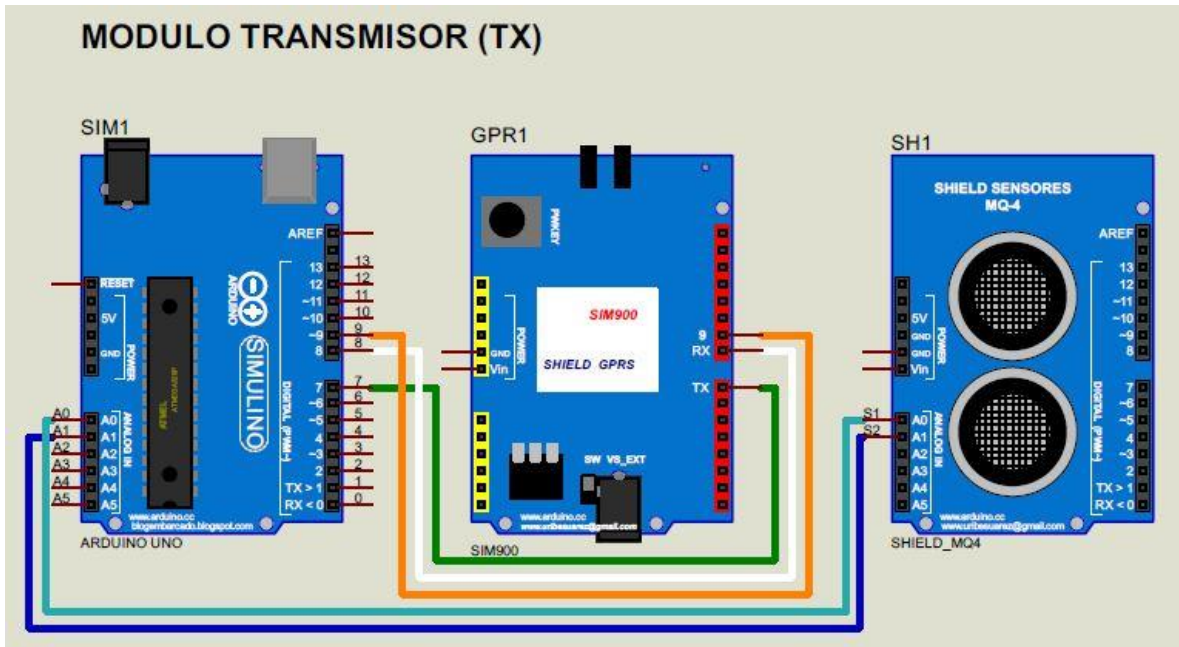


Fig. 40 Diagrama de conexión del módulo transmisor (Tx). [8]

4.7.1.3 PROGRAMACIÓN DEL MODULO TRANSMISOR (TX)

En esta etapa de comunicación, las funciones principales del transmisor es leer el dato que se captura con los sensores, realizamos su procesamiento necesario y después lo enviamos al módulo receptor (Rx).

En la siguiente tabla 15 se explica muy detalladamente la descripción de cada función utilizada en este programa.

Tabla 15 Descripción de las funciones utilizadas en el programa

FUNCION	DESCRIPCIÓN
Configuraciones iniciales	Se incluye la librería >SoftwareSerial.h> la cual configura el puerto serial. Se debe tener en cuenta los pines que vamos a utilizar para la comunicación serial de la SIM900.

	<p>El arduino uno emplea por defecto los pines digitales (0,1) para la comunicación serial con el PC.</p> <p>Se declaran las variables globales que se requieren para realizar los procesos necesarios en el programa.</p>
void setup ()	<p>Se configuran la velocidad del puerto serie para la SIM900 y el puerto serie del arduino.</p> <p>Se declara el <i>pin_12</i> digital del arduino como salida (<i>OUTPUT</i>), la cual va ser un indicador para él envío de los mensajes de texto.</p> <p>Se llaman las funciones <i>power_on()</i>, <i>iniciar()</i> y <i>empezar()</i>, las cuales son las encargadas de realizar todo los procedimientos del programa.</p> <p>Comprueba que el módulo SIM900 está disponible</p>
void power_on()	<p>Verifica la conexión a la red y enciende el GPRS enviando un pulso de un 1 segundo a través del PIN_9.</p>
void power_off()	<p>Apaga el módulo GPRS enviándole un pulso de un 1 segundo a través del pin_9</p>
void reiniciar()	<p>Cuando la señal del GPRS se cae, esta función apaga el modulo y lo enciende de nuevo, utilizando las funciones <i>power_off()</i> y <i>power_on()</i>.</p> <p>Esta función primero comprueba que la SIM900 esté conectada a la red móvil, enviando un comando AT en un ciclo while cada segundo.</p>
void iniciar()	<p>Si se logra conectar a la red, después envía los comandos AT que configura la SIM900 para identificar llamada, modo texto y nos muestre los SMS recibidos.</p>
Int enviarAT (ATcomand,resp, tiempo) char	<p>Esta función se encarga de recibir los argumentos de comando AT, respuesta esperada, tiempo y retornar un valor booleano. Si el comando AT es enviado correctamente a la SIM900 retorna un valor verdadero, de lo contrario retorna falso.</p>

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

void send_sms()	Configura la SIM900 para enviar SMS, el número móvil al cual se va enviar el SMS, llama función <i>Read_sensor()</i> y codifica el mensaje para luego ser enviado.
void Read_sensor()	Lee el sensor, realiza la conversión del A/D a voltaje, calcula la concentración del gas en unidades ppm y envía el dato a la SIM900.
void empezar()	Es la función principal del programa, es repetitivo los procesos que se ejecutan en ella. Envía el dato a través de la función <i>send_sms()</i> , luego se le agrega un retardo para que tenga tiempo de enviar el mensaje, comprueba que el mensaje se haya enviado correctamente. Si existe un error en el envío del SMS invoca a las funciones <i>reiniciar()</i> e <i>iniciar()</i> para reestablecer el sistema.

En la siguiente tabla 16 podemos ver el código de cada una de las funciones empleadas en este programa.

Tabla 16 Códigos de las funciones utilizadas en el programa.

FUNCION	CODIGO
Configuraciones iniciales	<pre>// Configura el puerto serial para el SIM900. SoftwareSerial SIM900(7, 8); int respuesta; char aux_str[50]; int cont=0; char dato[100];</pre>
void setup ()	<pre>void setup() { pinMode(12, OUTPUT); //Configura velocidad del puerto serie para el SIM900 SIM900.begin(19200); //Configura velocidad del puerto serie del Arduino Serial.begin(19200); delay(1000); Serial.println("Iniciando..."); power_on(); iniciar(); delay(1000); empezar(); }</pre>

```

void power_on() {
  int respuesta = 0;
  //comprueba la conexion a la red
  if (enviarAT("AT", "OK", 2000) == 0)
  {
    Serial.println("Encendiendo el GPRS...");
    pinMode(9, OUTPUT);
    digitalWrite(9, HIGH);
    delay(1000);
    digitalWrite(9, LOW);
    delay(1000);
    // Espera la respuesta del modulo SIM900
    while (respuesta == 0) {
      //Envia un comando AT cada 2 segundos y espera la respuesta
      respuesta = enviarAT("AT", "OK", 2000);
      SIM900.println(respuesta);
    }
  }
}

```

void power_on()

```

void power_off()
{
  digitalWrite(9, HIGH);
  delay(1000);
  digitalWrite(9, LOW);
  delay(1000);
}

```

void power_off()

```

void reiniciar()
{
  Serial.println("Reiniciando...");
  power_off();
  delay (5000);
  power_on();
}

```

void reiniciar()

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

Void iniciar()

```
void iniciar() {
  //enviarAT("AT+CPIN=\n1867\n", "OK", 1000);
  Serial.println("Conectando a la red...");
  delay (5000);
  //espera hasta estar conectado a la red movil
  while ( enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 0 ) {}
  Serial.println("Conectado a la red.");
  //Activa identificacion de llamadas
  enviarAT("AT+CLIP=1\n", "OK", 1000);
  //Configura el modo texto para enviar o recibir mensajes
  enviarAT("AT+CMGF=1\n", "OK", 1000);
  //Configuramos el modulo para que nos muestre los SMS recibidos
  enviarAT("AT+CNMI=2,2,0,0,0\n", "OK", 1000);
  Serial.println("Preparado.");
}
```

Int enviarAT
(ATcomand,resp,
tiempo)
char

```
int enviarAT(char* ATcommand, char* resp_correcta, unsigned int tiempo){
  int x = 0;
  bool correcto = 0;
  char respuesta[100];
  unsigned long anterior;
  memset(respuesta, '\0', 100); // Inicializa el string
  delay(100);
  //Limpia el buffer de entrada
  while ( SIM900.available() > 0) SIM900.read();
  SIM900.println(ATcommand); // Envía el comando AT
  x = 0;
  anterior = millis(); // Espera una respuesta
  do {
    // si hay datos el buffer de entrada, lee y comprueba la respuesta
    if (SIM900.available() != 0){
      //Comprueba que no haya desbordamiento en la capacidad del buffer
      if (x < 99) { respuesta[x] = SIM900.read();
        x++;}
      else Serial.println("Desbordamiento!");
      // Comprueba si la respuesta del modulo es la 1
      if (strstr(respuesta, resp_correcta) != NULL){
        correcto = 1; }}
    // Espera hasta tener una respuesta
  while ((correcto == 0) && ((millis() - anterior) < tiempo));
  return correcto;}
}
```

void send_sms()

```
void send_sms(){
  //Comando AT para mandar un SMS
  enviarAT("AT+CMGF=1\n", "OK", 1000);
  sprintf(aux_str, "AT + CMGS = \n3202146894\n", strlen(sms));
  enviarAT(aux_str, ">", 5000);
  SIM900.print('#');
  Read_sensor();
  SIM900.print('*');
  delay(100);
  //End AT command with a ^Z, ASCII code 26
  SIM900.print((char)26);
  delay(1000);
}
```



```

void
Read_sensor()
void Read_sensor(){
  int MQ4= analogRead(A0);
  delay(100);
  //Conversion en valor de voltaje
  float v1 = MQ4* (5.0 / 1023.0);
  //Calculamos Rs con un RL de 1k
  float Rs1= 1000*((5-v1)/v1);
  // calculamos la concentracion gas ppm
  double metano= 2803.7*pow(Rs1/343, -2.055);
  delay(100);
  SIM900.print(metano);
}

```

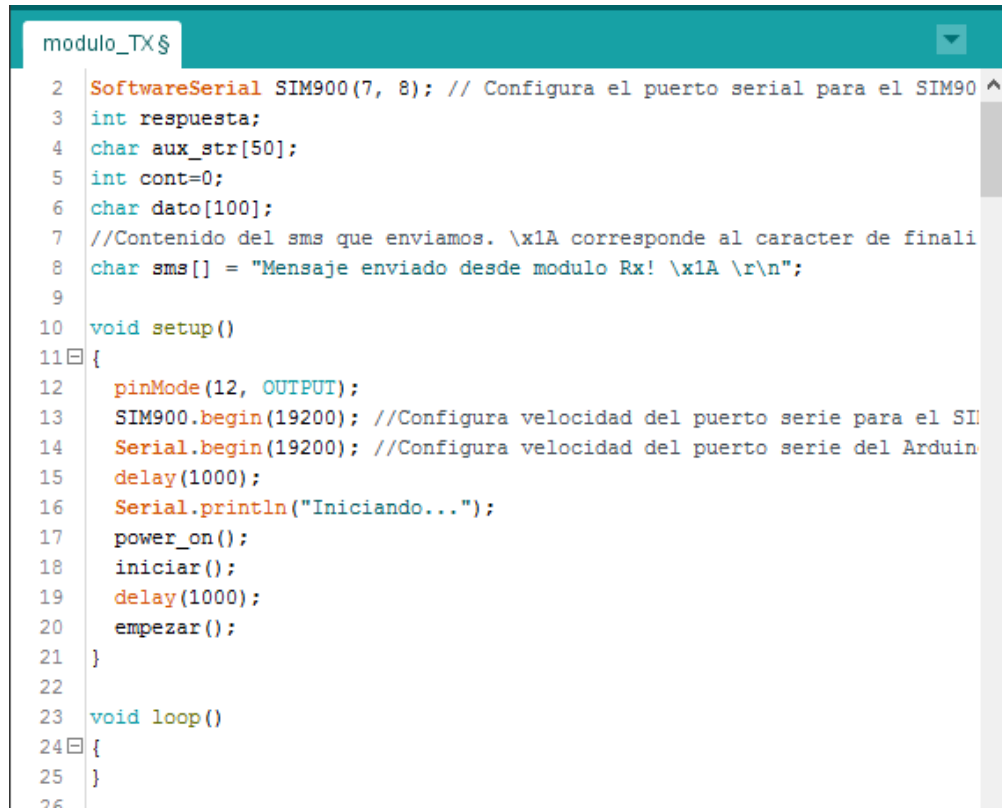
```

void
empezar()
void empezar(){
  inicio:
  int i=0;
  send_sms();
  delay(4000);// retardo para que se envíe el sms
  while(SIM900.available()){
    dato[i] = SIM900.read();
    i++;}
  if (strstr(dato,"OK") != NULL){
    Serial.println("SMS enviado");
    digitalWrite(12, HIGH);
    delay(500);
    digitalWrite(12 , LOW);
    goto inicio;}
  else {
    reiniciar();
    iniciar();}}

```

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

El código principal del módulo transmisor se puede apreciar en la siguiente figura del IDE de arduino.



```
modulo_TX$
2 SoftwareSerial SIM900(7, 8); // Configura el puerto serial para el SIM900
3 int respuesta;
4 char aux_str[50];
5 int cont=0;
6 char dato[100];
7 //Contenido del sms que enviamos. \x1A corresponde al caracter de finali
8 char sms[] = "Mensaje enviado desde modulo Rx! \x1A \r\n";
9
10 void setup()
11 {
12   pinMode(12, OUTPUT);
13   SIM900.begin(19200); //Configura velocidad del puerto serie para el SI
14   Serial.begin(19200); //Configura velocidad del puerto serie del Arduin
15   delay(1000);
16   Serial.println("Iniciando...");
17   power_on();
18   iniciar();
19   delay(1000);
20   empezar();
21 }
22
23 void loop()
24 {
25 }
26
```

Fig. 41 Programa principal del módulo transmisor (Tx)

4.7.2. MODULO RECEPTOR (RX)

En esta etapa de comunicación se utilizaron los siguientes materiales para su construcción:

- Arduino uno
- Shield GPRS SIM900
- Conectores

4.7.2.1 DESCRIPCION DEL MODULO RECEPTOR (RX)

El modulo receptor es el encargado de recibir los datos enviados por el modulo transmisor, y luego enviarlos mediante el puerto serial a la interfaz gráfica

desarrollada en matlab. La cual su función es visualizar los datos y realizar su procesamiento programado.

El siguiente diagrama de bloques explica el funcionamiento de esta etapa de comunicación.

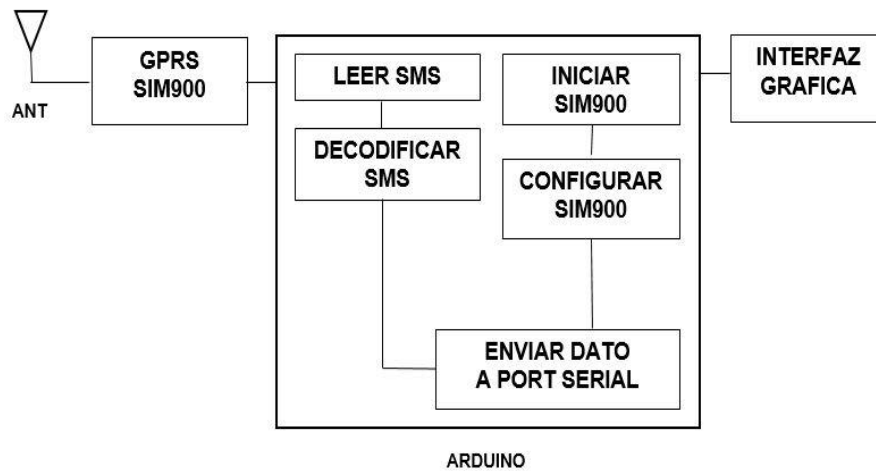


Fig. 42 Diagrama de bloques del módulo receptor (Rx)

4.7.2.2 DIAGRAMA DE CONEXIÓN DEL MODULO (RX)

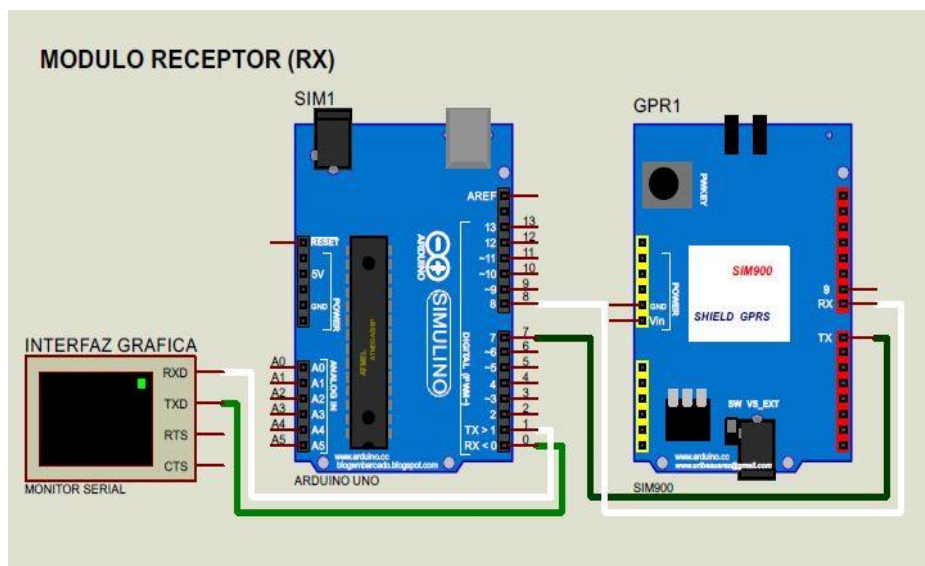


Fig. 43 . Diagrama de conexión del módulo receptor (Rx). [8]

4.7.2.3 PROGRAMACION DEL MODULO RECEPTOR (RX)

En esta etapa de comunicación del módulo receptor (Rx) sus funciones utilizadas en este programa se explican en la siguiente tabla 17.

Tabla 17 Descripción de las funciones utilizadas en el programa.

FUNCION	DESCRIPCIÓN
Configuraciones iniciales	<p>Se incluye la librería >SoftwareSerial.h> la cual configura el puerto serial. Se debe tener en cuenta los pines que vamos a utilizar para la comunicación serial de la SIM900.</p> <p>Se declaran las variables globales que se requieren para realizar los procesos necesarios en el programa.</p>
void setup ()	<p>Se configuran la velocidad del puerto serie para la SIM900 y el puerto serie del arduino.</p> <p>Se declara el <i>pin_12</i> digital del arduino como salida (<i>OUTPUT</i>), la cual va ser un indicador para él envió de los mensajes de texto.</p> <p>Se llaman las funciones <i>power_on()</i>, <i>iniciar()</i> y <i>empezar()</i>, las cuales son las encargadas de realizar todo los procedimientos del programa.</p> <p>Esta función primero comprueba que la SIM900 esté conectada a la red móvil, enviando un comando AT en un ciclo while cada segundo.</p>
void iniciar()	<p>Si se logra conectar a la red, después envía los comandos AT que configura la SIM900 para identificar llamada, modo texto y nos muestre los SMS recibidos.</p>
Int enviarAT (ATcomand,resp, tiempo) char	<p>Esta función se encarga de recibir los argumentos de comando AT, respuesta esperada, tiempo y retornar un valor booleano. Si el comando AT es enviado correctamente a la SIM900 retorna un valor verdadero, de lo contrario retorna falso.</p>
void identifica_SMS()	<p>Verifica si el dato enviado es correcto, de acuerdo al protocolo configurado de codificación</p>

Si el dato es válido lo sustrae de la cadena de texto recibida y luego lo convierte a un dato tipo entero.

Finalmente lo envía por el puerto serial hacia la interfaz gráfica desarrollada.

void loop()	<p>En esta función se repiten las instrucciones programadas continuamente.</p> <p>Lee los datos disponibles en el puerto serial de la SIM900 y si es válido lo detecta y luego lo envía por el puerto serial del arduino hacia la interfaz gráfica.</p>
--------------------	---

En la siguiente tabla 18 se muestran los códigos utilizados en las funciones mencionadas en la tabla anteriormente.

Tabla 18 Códigos de las funciones utilizadas en el programa.

FUNCIÓN	CODIGO
Configuraciones iniciales	<pre>#include <SoftwareSerial.h> // Configura el puerto serial para el SIM900 SoftwareSerial SIM900(7, 8); char incoming_char = 0; String mensaje = ""; String sms = "";</pre>
void setup ()	<pre>void setup() { //Configura velocidad del puerto serie para el SIM900 SIM900.begin(19200); //Configura velocidad del puerto serie del Arduino Serial.begin(19200); delay(1000); //Tiempo para que encuentre una RED Serial.println("Iniciando..."); delay(5000); iniciar(); }</pre>

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

```
void iniciar()
{
  enviarAT("AT", "OK", 1000);
  Serial.println("Conectando a la red...");
  delay (5000);
  //espera hasta estar conectado a la red movil
  while ( enviarAT("AT+CREG?", "+CREG: 0,1", 10000) == 0 ){
  Serial.println("Conectado a la red.");
  enviarAT("AT+CLIP=1\r", "OK", 1000);
  enviarAT("AT+CMGF=1\r", "OK", 1000);
  enviarAT("AT+CNMI=2,2,0,0,0\r", "OK", 1000);
  Serial.println("Preparado.");
}

void identifica_SMS(){
  int lg=mensaje.length();
  int nc= mensaje.indexOf('#'); // detecto el dato
  sms= mensaje.substring(nc+1,lg-1); //sustraer el dato
  int dato = sms.toInt();
  delay(10);
  if( nc>=0){
    Serial.println(dato);
    delay(10);
    dato=0;
    sms="";
    mensaje =""; // borrarlo para la proxima vez
  }}

void loop()
{
  int ban=0;
  if (SIM900.available() > 0){
    incoming_char= SIM900.read();
    mensaje += incoming_char;
    if (incoming_char == '*') ban =1;
  }
  delay(10);
  if(ban==1) identifica_SMS();
}
```

El programa del módulo receptor se muestra en la siguiente figura 42 del IDE de arduino.

```
modulo_RX$
1 // MODULO RX TEL: 320 214 6894
2 #include <SoftwareSerial.h>
3 SoftwareSerial SIM900(7, 8); // Configura el puerto serial para el SIM900
4 //Variable que guarda los caracteres que envia el SIM900
5 char incoming_char = 0;
6 String mensaje = "";
7 String sms = "";
8
9 void setup() {
10     SIM900.begin(19200); //Configura velocidad del puerto serie para el SIM9
11     Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
12     delay(1000); //Tiempo para que encuentre una RED
13     Serial.println("Iniciando...");
14     delay(5000);
15     iniciar();
16 }
17 void loop() {
18     int ban=0;
19     if (SIM900.available() > 0){
20         incoming_char= SIM900.read();
21         mensaje += incoming_char;
22         if (incoming_char == '*') ban =1;
23     }
24     delay(10);
25     if(ban==1) identifica_SMS();
26 }
..
```

Fig. 44 Programa del módulo receptor (Rx)

4.8 DESARROLLO DE LA INTERFAZ GRAFICA EN MATLAB

4.8.1 PROGRAMACION ARDUINO Y MALATB

La comunicación de la plataforma arduino y el software matlab se realiza a través de la comunicación serial (*UART*). Como toda comunicación serial debemos primero definir el COM que vamos a utilizar y la velocidad de comunicación (BaudRate) para el sistema. Definiendo estos parámetros ya podemos enviar y recibir datos desde arduino a matlab y viceversa. En la siguiente figura se observa la sintaxis de programación.

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALÁMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

```

%Comunicacion serial arduino <> matlab

%inicializo el puerto serial
delete(instrfind({'Port'},{'COM7'}));
a = serial('COM7','BaudRate',19200);

% open puerto serial
fopen(a);
disp('Esperando muestras...')
    
```

Fig. 45 Comunicación arduino & matlab.

La figura 43 vemos el código que se ejecuta en el programa matlab en la ventana (*Command Window*). En la variable *a*, se crea el objeto de puerto serial, la función serial se configuran los parámetros: 'COM', 'BaudRate', y la función *fopen* abre el puerto serial creado para iniciar la comunicación. Luego podemos utilizar todas las herramientas que nos ofrece la plataforma matlab, algunas de ellas se mencionan en la siguiente tabla 19.

Tabla 19 Funciones utilizadas en el programa matlab. [18]

FUNCION	DESCRIPCIÓN
a=serial('COM7','BauRate',19200);	Crea el objeto puerto serial
fopen(a)	Open puerto serial
fclose(a)	Close puerto serial
disp('Mensaje...')	Imprime un mensaje de texto
a.BytesAvailable	Pregunta si hay datos en el buffer
fscanf(a,FORMAT)	Leer datos del puerto serial
delete(a);	Elimina el objeto creado
xlswrite('file1.xlsx',dato);	Guarda un archivo formato Excel
xlsread('file1.xlsx');	Carga el archivo guardado

COMANDOS EN GUIDE MATLAB

val= get(handles.slider1,'Value');	Obtener el valor del slider
set(handles.text1,'String',val);	asignar el valor utpl a statictext
Val=(hObject,'String');	Almacena valor ingresado

<code>NewVal= str2double(Val);</code>	transforma a double entero
<code>handles.edit1= NewVal;</code>	almacena en identificador
<code>guidata(hObject,handles);</code>	salva datos aplicación
<code>imagen = imread('imagen.jpg');</code>	Leer una imagen
<code>axes(handles.imagen);</code>	carga la imagen en imagen
<code>imshow(imagen);</code>	Presenta la imagen

4.8.2 DISEÑO DE INTERFAZ GRAFICA EN GUIDE MATLAB

Los datos capturados por el modulo receptor, son visualización y procesados por la interfaz gráfica, la cual es un programa ejecutable desarrollado con el software matlab en la herramienta guide. En las siguientes figuras (42-43) se puede ver esta aplicación.

En la figura 44 se puede ver la ventana inicio de la aplicación, tiene dos botones, uno *star* para empezar el proceso de captura de datos y otro *power* la cual ejecuta una ventana emergente donde podemos seleccionar la opción salir de la aplicación.



Fig. 46 Ventana inicio de la interfaz gráfica

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE GAS METANO Y TRANSMISIÓN INALAMBRICA DE LOS DATOS A UN PUNTO DE CONTROL

En la siguiente figura 45, se observa la ventana capture de la aplicación interfaz gráfica, la cual permite configurar las siguientes opciones:

- Nombre de la muestra (Nombre del archivo donde se guardan los datos capturados en formato .txt y .xlsx)
- El número de muestras a capturar
- Seleccionar el puerto COM a utilizar
- Seleccionar la velocidad del Baud Rate para la comunicación serial
- Botón inicio de proceso
- Display donde visualiza el valor del dato capturado
- Display donde visualiza el número de la muestra
- Display donde visualiza el porcentaje actual del proceso
- Un semáforo donde se visualiza el proceso de adquisición de datos.
- Una plot donde grafica en tiempo real los datos capturados.

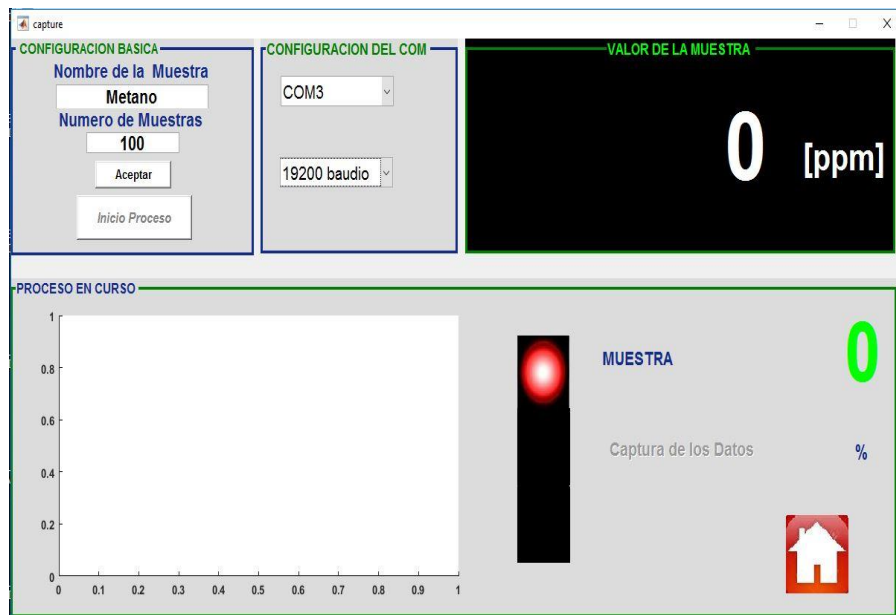


Fig. 47 Ventana capture de la interfaz gráfica

4.9 CREAR EJECUTABLE EN MATLAB

El software matlab tiene la opción de desarrollar nuestras propias aplicaciones (archivo .ex) de los programas realizados. La cual nos permite poder ejecutar los

programas en cualquier pc donde no tenga instalado matlab. En la siguiente figura 44 se muestra el procedimiento para crear los ejecutables (aplicaciones .ex).

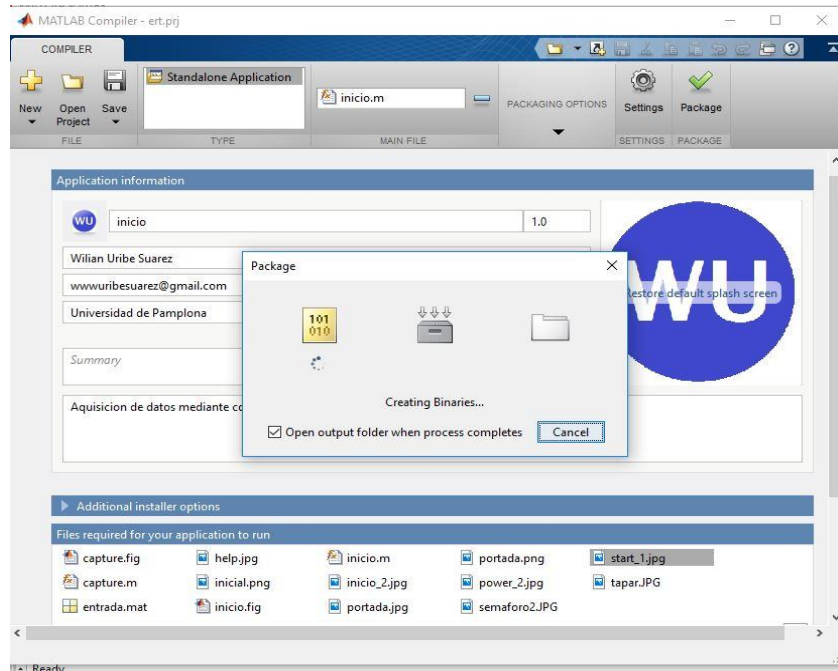


Fig. 48 Crear ejecutable en matlab

En la figura 46 muestra cómo crear nuestra aplicación ejecutable en matlab, el procedimiento es muy sencillo, se da clic en la pestaña Compiler, donde se despliega una ventana, la cual tiene unos campos para completar, en el icono (-) se agrega el archivo principal, y en la parte de abajo icono (+) se añaden todos los archivos utilizados para la ejecución del programa, después se la da clic en el icono Package y empieza a desarrollar nuestro archivo ejecutable (.ex). Una vez terminado el proceso correctamente nos genera todos archivos necesarios para ejecutar nuestra aplicación sin necesidad de tener el software matlab instalado.

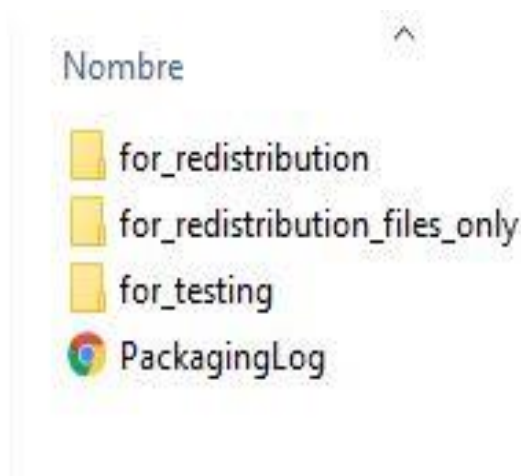


Fig. 49 Archivos generados para el ejecutable de la aplicación creada en matlab

5.

Análisis y Resultados

5. ANALISIS Y RESULTADOS	67
5.1. Prueba nariz E-nose v2.0	67
5.2. Prueba del Sistema diseñado.....	68
5.3. Limitantes.....	69
5.4. Análisis Económico.....	69

5. ANALISIS Y RESULTADOS

5.1 PRUEBA NARIZ E-NOSE V2.0

En el centro de reciclaje (basurero) de Pamplona norte de Santander, se realizaron unas pruebas con la nariz electrónica E-nose V2.0. En las siguientes imágenes se observan dichas pruebas.



Fig. 50 Pruebas E-nose V2.0

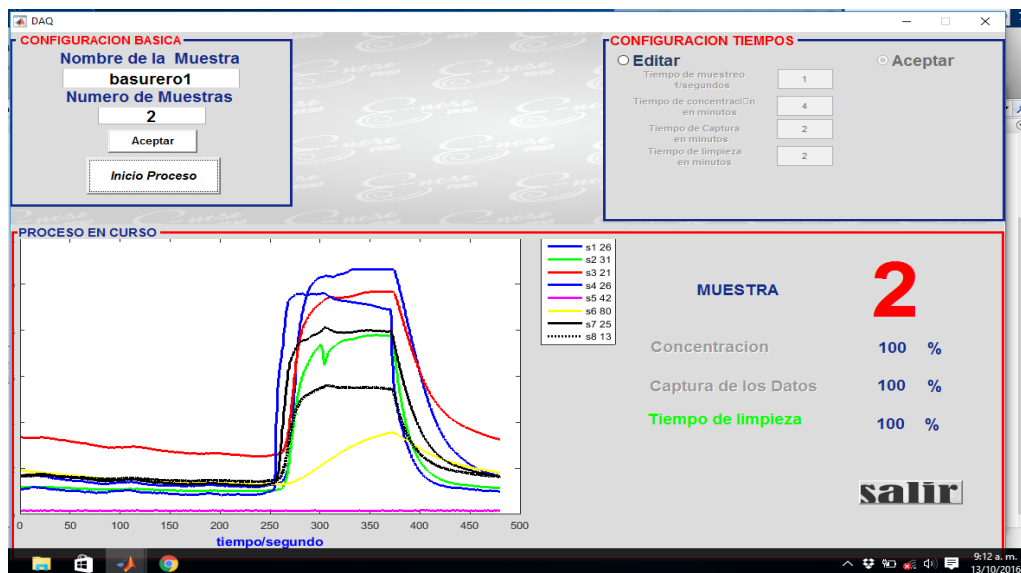


Fig. 51 Resultado E-nose V2.0

En la figura 48, se observa el resultado de la prueba tomada en el basurero de Pamplona, la cual vemos la reacción de los sensores a las emisiones de gases presentes, con ello podemos verificar que en estas calderas hay presencia de emisiones de gas metano. Esta información es importante para el diseño de nuestra nueva nariz electrónica E-nose metano.

5.2 PRUEBA DEL SISTEMA DISEÑADO

Para la comprobación del funcionamiento del sistema de comunicación diseñado (*E-nose Metano*), se realizó la siguiente prueba, obteniendo el siguiente resultado.



Fig. 52 Resultado del sistema diseñado.

5.3 LIMITANTES

En el diseño de nuestro sistema de comunicación inalámbrica, la captura de muestras obtenidas por los sensores de gas metano, estos datos se envían con fiabilidad cada 2-3 segundos dependiendo de la disponibilidad de la señal móvil, la cual nos impide aumentar la velocidad de muestreo.

La toma de muestras del módulo transmisor, debe tener un tiempo de reposo entre cada muestra, ya que los sensores deben descontaminarse para no alterar los valores de medición obtenidos. Esto no nos permite captura de datos en tiempo continuo.

5.4 ANÁLISIS ECONÓMICO

El costo económico de este proyecto se muestra a continuación en las siguientes dos figuras (50-51), las cuales indican el presupuesto del módulo transmisor y módulo receptor.

Bill Of Materials for Modulo Transmisor

Design Title Modulo Transmisor
Author Wilian Uribe Suarez
Document Number 2
Revision
Design Created viernes, 21 de abril de 2017
Design Last Modified jueves, 11 de mayo de 2017
Total Parts In Design 4

4 Miscellaneous				
Quantity	References	Value	Stock Code	Unit Cost
1	BOX1	CAJA MODULO TX		\$30.000,00
1	GPR1	SIM900		\$130.000,00
1	SH1	SHIELD_MQ4		\$50.000,00
1	SIM1	ARDUINO UNO		\$50.000,00
Sub-totals:				\$260.000,00
Totals:				\$260.000,00

viernes, 12 de mayo de 2017 7:55:33 a. m.

Fig. 53 Presupuesto modulo Transmisor.

Bill Of Materials for Modulo Receptor

Design Title Modulo Receptor
Author Wilian Uribe Suarez
Document Number 1
Revision
Design Created miércoles, 3 de mayo de 2017
Design Last Modified viernes, 12 de mayo de 2017
Total Parts In Design 4

4 Miscellaneous				
<u>Quantity</u>	<u>References</u>	<u>Value</u>	<u>Stock Code</u>	<u>Unit Cost</u>
1	BOX1	CAJA MODULO RX		\$20.000,00
1	GPR1	SIM900		\$130.000,00
1	INTERFAZ GRAFICA	MONITOR SERIAL		\$20.000,00
1	SIM1	ARDUINO UNO		\$50.000,00
Sub-totals:				\$220.000,00
Totals:				\$220.000,00

viernes, 12 de mayo de 2017 7:57:08 a. m.

Fig. 54 Presupuesto modulo Receptor.

El presupuesto total para la construcción de este sistema de comunicación tiene un costo total de: \$ 480 000,00 (cuatrocientos ochenta mil pesos colombianos) en materiales.

6.

Conclusiones

6. CONCLUSIONES FINALES..... 72

6. CONCLUSIONES FINALES

El módulo de comunicación GPRS SIM900 nos da una gran ventaja en la transmisión de datos a largas distancias, quiere decir que donde exista la cobertura de telefonía móvil de algún operador del servicio, podemos realizar esta transmisión de datos. La desventaja que se nos presenta con este módulo es la frecuencia de muestreo de los datos, la cual en las pruebas que se realizaron de envío de mensajes (SMS), se debe configurar un tiempo de envío entre dato y dato que oscila entre (2-3 segundos) para garantizar la comunicación del sistema. La cual es un tiempo muy alto en el muestreo de datos de una señal.

El procedimiento para garantizar el muestreo de la señal en tiempo real fue el siguiente, primero se realiza el muestreo de la captura de los datos de la señal a medir y después se empiezan a transmitir los datos a la estación de control. De esta manera se garantiza la obtención de la señal de la variable medida.

La creación de aplicaciones ejecutables en el software matlab nos da la facilidad de poder ejecutar nuestros programas diseñados en cualquier ordenador, pero la desventaja de utilizar esta plataforma es que es un servicio privado. La cual nos genera costos en la comercialización de nuestras aplicaciones, se debe contar con las respectivas licencias y permisos.

Una de las ventajas de este sistema de comunicación fue su bajo costo, este diseño presenta un costo en materiales aproximadamente de \$ 500.000 pesos, sin incluir la mano de obra. La cual es bastante competitivo frente a las tarjetas de adquisición de datos y sistemas de comunicaciones actuales en el mercado.

BIBLIOGRAFÍA

- [1] D. Sharvari, B. Rajib , B. Nabarun y P. R.A., «Application of electronic nose for industrial odors and gaseous emissions measurement and monitoring - An overview,» *Talanta*, vol. 144, pp. 329-340, 2015.
- [2] Z. Lei , T. Fengchun y P. Guangshu, «A novel sensor selection using pattern recognition in electronic nose,» *Measurement*, vol. 54, pp. 31-39, 2014.
- [3] S. Andrzej y M. Monika, «"Artificial sniffing" based on induced temporary disturbance of gas sensor response,» *Sensors and Actuators B: Chemical*, vol. 186, pp. 109-116, 2013.
- [4] J. Derek y H. Robert, «Methane emissions measurements of natural gas components using a utility terrain vehicle and portable methane quantification system,» *Atmospheric Environment*, vol. 144, pp. 1-7, 2016.
- [5] I. Moreno, R. Caballero, R. Galán, F. Matia y A. Jiménez, «La Nariz Electrónica: Estado del Arte,» *Revista Iberoamericana de Automática e Informática Industrial*, vol. 6, nº 3, pp. 76-91, 2009.
- [6] O. E. Gualdrón Guerrero y O. Fiallo, «Organochlorine screening using food in fresh an electronic nose,» vol. 10, nº 3, pp. 17-25, 2016.
- [7] J. E. Ortiz Sandoval, Implementacion de un modelo neuronal en un dispositivo Hardware (FPGA) para la clasificacion de compuestos quimicos en un sistema multisensorial (Olfato Electronico), Pamplona, 2012.
- [8] E. Labcenter, «Proteus Design Suite 8.4,» [En línea]. Available: <http://www.labcenter.co.uk>.
- [9] M. google, «Mapas google colombia,» [En línea]. Available: <http://www.mapasgoogle.com/distancia>.
- [10] W. D. Galvis, «Iniciando con Arduino uno r3,» 28 8 2013. [En línea]. Available: <http://www.microkitselectronica.com>. [Último acceso: 12 1 2017].

- [11] «arduino uno,» [En línea]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardUno..>
[Último acceso: 14 11 2016].
- [12] S. Limited, «SIM900_Hardware Design_V2.00,» 15 12 2010. [En línea]. Available:
<http://www.sim.com/wm>. [Último acceso: 13 1 2017].
- [13] I. V1.0, «ITead Studio Make Innovation easter,» 27 05 2011. [En línea]. Available:
<https://iteadstudio.com>.
- [14] Prometec, «Módulo GSM/GPRS,» [En línea]. Available: <http://www.prometec.net>. [Último
acceso: 12 10 2016].
- [15] S. A. c. o. S. Tech, «SIM900 AT Command Manual,» 24 12 2010. [En línea]. Available:
<http://www.sim.com>. [Último acceso: 12 11 2016].
- [16] 1. H. Build, «Arduino,» 21 10 2016. [En línea]. Available: <http://www.arduino.cc>.
- [17] L. Henan Hanwei Electronics Co., «MQ-4 Semiconductor Sensor for Natural Gas,» [En línea].
Available: <http://www.hwsensor.com>.
- [18] D. O. Barragán Guerrero, «Manual de Interfaz gráfica de usuario en Matlab,» 25 05 2008. [En
línea]. Available: <http://www.matpic.com>.
- [19] A. Delgado Castro y O. Rojas Bolaños, «Construcción de un sistema de bajo costo para el uso y
evaluación de sensores semiconductores para gases,» *Educación Química*, vol. 26, pp. 299-
306, 2015.
- [20] F. P. Pérez Fernández, *Diseño e Implementación de un sistema Embebido sobre una
plataforma movil para el analisis de la calida del aire*, Pamplona, 2016.
- [21] S. A. Pinzón, *Sistema de alarmas, transmisión y monitoreo dedatos aplicado a la medición de
variables con redes GSM/GPRS*, Manizales, 2016.
- [22] A. Dominguez Castro, *Sistema de control de temperatura a travez de Arduino y la tecnologia
GPRS/GSM*, 2010.
- [23] C. M. Duran Acevedo y J. C. Rodriguez Gamboa, «Sistema de olfato electrónico para la
detección de compuestos volátiles,» *Revista Colombiana de Tecnologias de Avanzada*, nº 18,
2008.