

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE UN ROBOT MÓVIL PARA MEDIR
PARÁMETROS DE CALIDAD DEL AIRE.**

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

TILINCHANA SIMBAÑA DANNY RAFAEL

danny.tilinchana@epn.edu.ec

DIRECTOR: ING. MONICA VINUEZA RHOR. MSc.

monica.vinueza@epn.edu.ec

Quito, marzo 2019

DECLARACIÓN

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

Danny Rafael Tilinchana Simbaña

C.I.: 1713553087

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por DANNY RAFAEL TILINCHANA SIMBAÑA, bajo nuestra supervisión.

Ing. Mónica Vinueza Rhor, MSc.
DIRECTORA DE PROYECTO

DEDICATORIA

Este trabajo de graduación producto de mi esfuerzo y entrega en cada una de sus etapas, les dedico a mi esposa, hijos, padres, y mis hermanas quienes a lo largo de toda mi vida me han apoyado y motivado en mi formación académica, creyeron en mí en todo momento y con sus consejos me han ayudado en muchas ocasiones a sobrellevar mi camino de superación y triunfo.

Danny Tilinchana S.

AGRADECIMIENTO

Agradecer a mis padres por todo, mi esposa, hijos y mis hermanas por su cariño, amor y comprensión, por su apoyo económico y moral para culminar mi proyecto, por toda la confianza puesta en mí en todo momento en el que lo necesite.

De manera especial y sincera expreso mi agradecimiento a la Ing. Mónica Vinueza Rhor mi director de tesis por su valiosa orientación y apoyo; A mi amigo y compañero Ing. Santiago Solórzano por el tiempo y apoyo para la culminación del mismo.

Danny Tilinchana S.

ÍNDICE

DECLARACIÓN.....	II
CERTIFICACIÓN	III
DEDICATORIA	IV
AGRADECIMIENTO	V
ÍNDICE	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	X
RESUMEN.....	XI
ABSTRACT	XII
1. INTRODUCCIÓN	2
1.1. Marco teórico	3
HARDWARE.....	3
Robot móvil.....	4
TReXDMC01	4
Control Remoto FlySky FS-i6X.....	5
SENSORES.....	6
Sensor de Humedad y Temperatura DTH11	6
Sensor Dióxido de Carbono CO2 (MG-811)	7
Sensor de Ozono MQ131	8
Sensor de Metano MQ-4	9
COMUNICACIÓN Y TRANSMISIÓN DE DATOS	9
Módulo XBee Pro S2B	9
SISTEMA DE ALIMENTACIÓN	10
Litio y Polímero (Lipo)	10
Iones de Litio (Li-ion)	11
SOFTWARE	12
Interfaz gráfica.....	12
Visual Studio	12
Arduino	12
Next Generation Configuration Platform for Xbee/RF Solutions (XCTU).....	13
2. METODOLOGÍA	13
3. RESULTADOS Y DISCUSIÓN	16
3.1. Diseño del prototipo	16
3.2. Diseño de la placa electrónica	21
3.3. Diseño de conexiones ARES	22
3.4. Configuración de la red de sensores inalámbricos	27
3.5. Ensamblaje del <i>Chasis</i> POLOLU 4WD	33

3.6. Emparejamiento del transmisor y receptor <i>FlySky</i>	38
3.7. Programación de Interfaz gráfica	40
3.8. Pruebas de Funcionamiento	45
3.9. Pruebas de comunicación.....	47
3.10. Pruebas de Sensores	48
3.11. Resultados.....	51
3.12. Costos de Implementación.....	55
4. CONCLUSIONES Y RECOMENDACIONES	56
4.1. Conclusiones	56
4.2. Recomendaciones	58
5. REFERENCIAS BIBLIOGRÁFICAS.	59
Anexo A1: Código implementado para el funcionamiento en el proyecto en Arduino.....	61
Anexo A2: Código implementado para el funcionamiento en el proyecto en Visual Studio.	61
Anexo A3: Módulos Xbee Pro.	61
Anexo A4: Arquitectura de módulos sensores.	61
Anexo A5: Manual de Usuario y mantenimiento.....	61

ÍNDICE DE FIGURAS

Figura 1.1 Chasis 4WD	3
Figura 1.2 TreXDMC01	4
Figura 1.3 Control remoto <i>Flysky FS-i6X 10CH 2.4GHz AFHDS RC</i>	5
Figura 1.4 Sensor de Humedad y Temperatura DTH11	6
Figura 1.5 Sensor Dióxido de Carbono CO2	7
Figura 1.6 Sensor Ozono MQ131	8
Figura 1.7 Sensor de Metano MQ4	9
Figura 1.8 Módulo <i>XBee Pro S2B</i>	10
Figura 1.9 Batería <i>Lipo</i>	11
Figura 1.10 Batería Lio-ion	11
Figura 1.11 Logo Visual Studio	12
Figura 2.1 Diagrama de proceso para la Implementación	15
Figura 3.1 Esquema general del prototipo	18
Figura 3.2 Circuito de elementos electrónicos y distribución de pines	19
Figura 3.3 Conexiones de alimentación	21
Figura 3.4 Conexiones Micro-controlador	21
Figura 3.5 Conexiones de sensores	22
Figura 3.6 Circuito en Ares	22
Figura 3.7 Visualización 3D	23
Figura 3.8 Diseño de placa principal	23
Figura 3.9 Elaboración de la placa principal	24
Figura 3.10 Placa principal con elementos soldados	24
Figura 3.11 Caja negra con placa principal	25
Figura 3.12 Algoritmo en <i>Arduino</i>	26
Figura 3.13 Conexiones <i>XBee Pro</i> con PC mediante <i>USB</i>	27
Figura 3.14 <i>XCTU</i> Puerto <i>Com</i> del Módulo <i>XBee Pro</i>	28
Figura 3.15 Agregar nuevo dispositivo	28
Figura 3.16 Parámetros en el <i>software</i> <i>XCTU</i>	29
Figura 3.17 Datos cargados con <i>XCTU</i>	30
Figura 3.18 Carga de parámetros en módulo	31
Figura 3.19 Carga de firmware en nodo Coordinador	32
Figura 3.20 Carga de <i>firmware</i> en Nodo	33
Figura 3.21 Cables soldados a motores	34
Figura 3.22 Ensamblaje de motores a chasis de aluminio	34
Figura 3.23 Conexión en serie de motores	35
Figura 3.24 Conexión sentido de giro	35
Figura 3.25 Conexión de motores al controlador	36
Figura 3.26 Alimentación con batería de <i>LiPo</i>	36
Figura 3.27 Conexiones controlador	37
Figura 3.28 Componentes ensamblados en robot móvil	37
Figura 3.29 <i>Binding</i> FS-IA6B	38
Figura 3.30 Botón <i>binding</i>	38
Figura 3.31 Mensaje <i>binding</i> en <i>display</i> del control	39
Figura 3.32 Control TX y receptor RX enlazados	39
Figura 3.33 Robot móvil con control remoto	40
Figura 3.34 Pasos para la crear un proyecto.	41
Figura 3.35 Pasos para crear <i>Windows Form</i>	41

Figura 3.36 <i>Windows Form</i>	42
Figura 3.37 Interfaz gráfica de parámetros de calidad del aire.....	42
Figura 3.38 Algoritmo de Interfaz gráfica.....	43
Figura 3.39 Medición de parámetros de calidad del aire.....	44
Figura 3.40 Prueba en césped	45
Figura 3.41 Pruebas en escaleras	46
Figura 3.42 Probador de batería <i>LiPo</i>	46
Figura 3.43 <i>Software Realterm</i>	47
Figura 3.44 Prueba de envío y recepción de datos	47
Figura 3.45 Robot móvil pruebas en campo abierto.....	49
Figura 3.46 <i>Robot</i> móvil expuesto a gases de vehículo.....	49
Figura 3.47 Resultado de muestra de sensores temperatura, humedad.....	50
Figura 3.48 Resultados de muestras obtenidas	51

ÍNDICE DE TABLAS

Tabla 1.1 Características Chasis 4WD	4
Tabla 1.2 Características <i>TReXDMC01</i>	5
Tabla 1.3 Características sensor DTH11	6
Tabla 1.4 Trama de Datos Sensor DTH11	7
Tabla 1.5 Características sensor CO2	8
Tabla 1.6 Características Sensor Ozono MQ131	8
Tabla 1.7 Características sensor MQ4	9
Tabla 1.8 Características técnicas <i>Xbee Pro S2B</i> .	10
Tabla 3.1 Rangos de medición y recomendación	20
Tabla 3.2 Parámetros en <i>XBee Pro</i>	30
Tabla 3.3 Datos nodo	32
Tabla 3.4 Características <i>Robot Móvil</i>	52
Tabla 3.5 Ventajas y desventajas de la Implementación	52
Tabla 3.6 Expresiones matemáticas para el cálculo del IQCA	53
Tabla 3.7 Límites numéricos de cada categoría del IQCA ($\mu\text{g}/\text{m}^3$)	54
Tabla 3.8 Rangos, significados y colores de las categorías del IQCA	54
Tabla 3.9 Identificación de individuos sensibles por tipo de contaminante del aire	54
Tabla 3.11 Costos primarios	55
Tabla 3.12 Costo total	55

RESUMEN

Para el desarrollo de este plan de titulación se contó, con una primera etapa que fue en definir las características del *robot* móvil, para la cual se utilizó un *kit* disponible en el mercado, accesible y cumplió con las necesidades que se planteen; accesible a terrenos no pantanosos ni acuáticos, se ensambló en una carcasa de aluminio, posee una protección para circuitos integrados, resistentes, de tamaño ergonómico para que pueda acceder a lugares reducidos, será controlado por un dispositivo de radio frecuencia y contará con una red punto a punto para transmisión de datos.

La segunda etapa definida como una pequeña red punto a punto que permitió la transmisión de datos inalámbricos con un alcance de 500 metros de longitud utilizando módulos *XBee*; desde el *robot* móvil hasta una computadora, el mismo que se visualizó mediante una interfaz gráfica amigable que fue desarrollada en el *software Visual Studio*, donde se tiene la posibilidad de observar los valores obtenidos por el sistema; parámetros como humedad, temperatura, ozono, CO2 y metano; obtenidos por el *robot* de los diferentes lugares de la unidad educativa y que la información obtenida se pueda observar en tiempo real directamente en la interfaz gráfica del ordenador.

Y una tercera etapa que validó el desempeño del sistema; se realizó mediciones de la contaminación ambiental en la Unidad Educativa San Martín ubicada en el sector de la Argelia en la ciudad de Quito Provincia de Pichincha, y se tomó como universo de estudio las instalaciones de dicha unidad educativa. Los datos obtenidos se grabaron en un archivo plano.

PALABRAS CLAVES: *XBee Pro* , Ozono, Metano, CO2, Radio Frecuencia

ABSTRACT

For the development of this plan, we counted with the following steps: first stage was to define the characteristics of the mobile robot, we used a kit available in stores, it was accessible and fulfilled the needs that were planet. It is not accessible in swamp like or water terrain. It was assembled whit an aluminum body. It has protection for its integrated circuit. It was made whit a size to able to reach small areas. It will be controlled using radio frequency and will have a wifi signal so it will be able to transmit data.

The second stage dealt with its wifi connection that allowed wireless transmission of data up to 500 meters of longitude using Xbee Pro module from the mobile robot to a computer, the computer was set up whit a user friendly graphic interface that was developed using the Visual Studio software, there you have the possibility to observe the values obtained by the system; Values like humidity, temperature, ozone, CO2 and methane; the information obtained by the robot can be viewed in real time directly in the computer using the graphic interface.

The third stage was to execute the function of the system; the studies were done at Unidad Educativa San Martin in the sector of La Argelia in the city of Quito, province of Pichincha by measuring the contamination of the environment in the rooms facilities of said school, the data obtained were recorded in a hard drive.

KEYWORDS: *Xbee Pro, Ozone, Methane, CO2, Radio Frequency*

1. INTRODUCCIÓN

La contaminación del aire representa un importante riesgo ambiental para la salud, bien sea en los países desarrollados o en los países en vías de desarrollo; según la Organización Mundial de la Salud (OMS) la contaminación ambiental es responsable que cada año millones de personas, mueran como consecuencia de este fenómeno; el aire en los hogares ha provocado un 34% de accidentes cerebrovasculares, 26% cardiopatías isquémicas, 22% neuropatías obstructivas crónicas, 12 % neumonía infantil y 6% por cáncer de pulmón; las consecuencias derivadas de este tipo de contaminación representan un creciente problema para la salud pública.

La calidad del aire está directamente influenciada por la distribución y la intensidad de las emisiones contaminantes vehicular y atmosférica principalmente. Sin embargo, juegan un papel fundamental la topografía y las condiciones meteorológicas.

Las emisiones vehiculares desempeñan un papel importante en los niveles de contaminación del aire en la totalidad del Distrito Metropolitano de Quito, mientras que las emisiones industriales afectan significativamente la calidad del aire en regiones específicas. Los resultados del monitoreo de la calidad del aire en el Distrito Metropolitano de Quito se presentan por contaminantes y su evaluación se la realiza comparando con la Norma Ecuatoriana de Calidad del Aire.

Los efectos negativos incluyen molestias en la salud con presencia de dolores de cabeza, estrés, migraña, insomnio, problemas del sistema nervioso, menos productividad y afectación a la salud mental, entre otros.

Debido al daño que causa la contaminación ambiental en las personas, es de vital importancia llevar un control de los cambios en los gases contaminantes y monitorear la afectación que provoca en el cuerpo humano.

Por tal motivo se necesario implementar un sistema de medición de contaminación ambiental utilizando un robot móvil, equipado con dispositivos o sensores que sirven para monitorear parámetros de calidad del aire, complementado a una red inalámbrica punto a

punto, que ayudará a transmitir al ordenador y visualizar los valores obtenidos para poder saber el grado de contaminación ambiental y prevenir el estado de la salud de las personas.

Para la implementación de este *robot* se determinó primero los requerimientos necesarios para el *hardware* como sensores, baterías, control remoto, *chasis* y componentes de transmisión de datos.

Luego se realizó la correspondiente implementación, se diseñó el sistema del *hardware* para la conexión respectiva de los elementos, con la ayuda de un diseño de una red inalámbrica punto a punto que fue necesaria para poder transmitir los datos.

Es importante que este sistema cuente con una interfaz gráfica para que la comunicación sea sencilla para el usuario y finalmente se realizó pruebas de funcionamiento.

1.1. Marco teórico

HARDWARE



Figura 1.1 *Chasis* 4WD ^[1]

Robot móvil

Este robusto chasis de 4 ruedas de *Dagu Electronics* tiene una suspensión independiente para cada una de sus ruedas de 120 mm de diámetro, lo que permite una tracción excepcional sobre terrenos irregulares y su cuerpo está hecho de aluminio anodizado de 2 mm de espesor con una rejilla de 10 mm de paso agujeros de 4 mm, lo que facilita el montaje de componentes electrónicos y accesorios en el chasis.

Tabla 1.1 Características Chasis 4WD^[1]

Modelo	Pololu 4WD
Voltaje de Alimentación	3.7v a 7.2v
Reductor	75:1
Medidas interiores	280x300x130
Corriente Máxima	6.6 A. por motor
Cantidad de Motores	4
Color	Plata

TReXDMC01

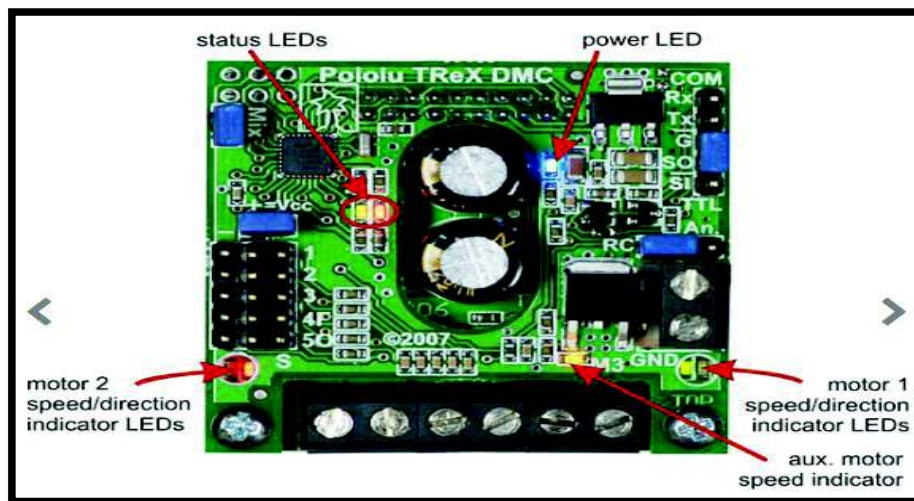


Figura 1.2 TreXDMC01^[1]

El *TReXDMC01* combina entradas de radio control (RC), analógicas y en serie para solución de control a los motor y que son capaz de entregar 13 A continuos por motor, diseñado para combinar a la perfección el control autónomo y humano de robots pequeños y medianos; puede controlar motores a través de interfaces de control independientes.

Tabla 1.2 Características *TReXDMC01*^[1]

Modelo	<i>TReXDMC01</i>
Voltaje de Alimentación	6v a 16v
Corriente Máxima por canal	13 A.
Corriente Máxima paralelo	25 A.
Control interface	<i>non-inverted TTL</i> serial (2-way); RS-232 serial (1-way), RC servo pulses, analog voltage

Control Remoto FlySky FS-i6X

Este robusto control remoto de 10 canales de marca *Flysky* FS-i6X ofrece una protección contra interferencias, mantiene un consumo de energía y una sensibilidad del receptor, la comunicación bidireccional capaz de enviar y recibir datos, tiene su propia identificación única y una vez que el transmisor y el receptor se hayan emparejado, solo se comunicarán entre sí, evitando que otros sistemas se conecten accidentalmente o interfieran con el funcionamiento del sistema la antena omnidireccional reduce la interferencia manteniendo una sensibilidad del receptor, extendiendo la vida útil de la batería.



Figura 1.3 Control remoto *Flysky* FS-i6X 10CH 2.4GHz AFHDS RC^[2]

SENSORES

Los sensores son los que obtienen datos del medio ambiente y se transforman a medios de señales eléctricas para el prototipo que se va a implementar se mostrarán las opciones de los sensores con los que se va a trabajar.

Sensor de Humedad y Temperatura DTH11

El sensor de temperatura DTH11 permite tener una lectura de Temperatura y humedad ambiental, con el uso de cualquier micro controlador facilita la implementación, en este caso se realizó el módulo con el controlador *Arduino Nano*.

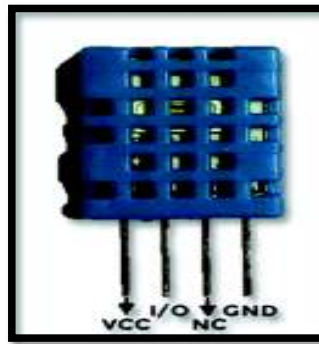


Figura 1.4 Sensor de Humedad y Temperatura DTH11 ^[3]

El sensor DTH11 posee la ventaja de ser digital por lo tanto tiene mayor tolerancia a interferencias de ruido.

Tabla 1.3 Características sensor DTH11 ^[3]

Modelo	DTH11
Voltaje de Alimentación	3,5v a 5v
Consumo	2,5mA
Señal de salida	Digital
Rango Temperatura	0oC a 50oC
Resolución	1oC (8 bits)
Rango Humedad	20% HR a 90 HR
Resolución	1%HR

Los datos de temperatura y humedad son analógicos pero la ventaja que tiene el DTH11 es que el propio dispositivo hace la conversión A/D. La trama de datos que realiza es de 40 *bits* correspondiente a la humedad y temperatura.

Tabla 1.4 Trama de Datos Sensor DTH11 ^[3]

0011 0101	8 bits de Humedad	Parte entera de la Humedad
0000 0000	8 bits de Humedad	Parte decimal de la Humedad
0001 1000	8 bits de Temperatura	Parte entera de la Temperatura
0000 0000	8 bits de Temperatura	Parte decimal de la Temperatura
0100 1001	8 Bits de paridad	Para confirmar que no hay datos falsos

Los *bits* de paridad hacen que la información sea la correcta. La suma binaria de los cuatro primeros grupos de 8 *bits* debe dar como resultándolos bits de paridad. Para la programación se utiliza la librería DTH11 con esta se dispone de los siguientes comandos.

Sensor Dióxido de Carbono CO2 (MG-811)

El sensor de CO2 de marca *DfRobot* dispone de una salida analógica la cual mientras mayor es la concentración de CO2 el voltaje disminuye. Puede ser configurado como alerta a la presencia de CO2 ajustando un potenciómetro el que establecerá un umbral de voltaje, si baja del umbral establecido una señal digital de salida será activada.



Figura 1.5 Sensor Dióxido de Carbono CO2 ^[4]

Entre las aplicaciones está el control de calidad del aire, con la detección de concentraciones de CO2 a temperatura ambiente.

Tabla 1.5 Características sensor CO2 ^[4]

Modelo	DFrobot v1.2
Voltaje de Alimentación	3,7v a 5v
Señal de salida	Analógico
Unidad	ppm (partes por millón)
Rango de CO2	>350ppm - <10000ppm
Tamaño	32x42mm

Sensor de Ozono MQ131

El sensor de ozono O3 detecta concentraciones de ozono en el medio y emite su lectura como una tensión analógica, lo gases detectados por este tipo de sensores puede ser mortal en altas concentraciones. ^[5]



Figura 1.6 Sensor Ozono MQ131 ^[6]

Tabla 1.6 Características Sensor Ozono MQ131 ^[6]

Modelo	MQ131
Voltaje de Alimentación	5v
Señal de salida	Analógica digital TTL
Unidad	ppm (partes por millón)
Rango de CO	<2ppm
Dimensiones	3.2cmx2.2cmx2.7cm

Sensor de Metano MQ-4

Es un sensor para detectar Gas metano en el aire, el MQ4 puede detectar concentraciones entes 300ppm a 10000ppm, disponen de una sensibilidad y su salida es analógica.



Figura 1.7 Sensor de Metano MQ4 ^[7]

Las características se describen a continuación.

Tabla 1.7 Características sensor MQ4 ^[7]

Modelo	MQ4
Voltaje de Alimentación	5v
Señal de salida	Analógico
Unidad	Ppm (partes por millón)
Rango de CO	>300ppm - <10000ppm
Consumo	900mW

COMUNICACIÓN Y TRANSMISIÓN DE DATOS

Módulo XBee Pro S2B

El módulo *XBee Pro S2B* proporciona conectividad inalámbrica con bajo consumo de corriente, autonomía extendida a 63mW y velocidad de 1Mbps en la banda 2,4GHz, son muy utilizados en redes de sensores inalámbricos, por el tamaño reducido en este proyecto se dispone del *XBee Pro S2B* ver imagen 1.8 y las características se describen a continuación.

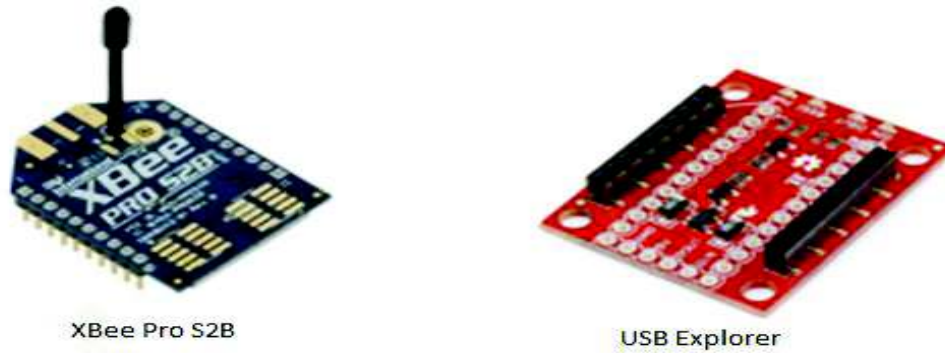


Figura 1.8 Módulo *XBee Pro S2B* [8]

Características de módulos de comunicación *XBee Pro S2B*.

Tabla 1.8 Características técnicas *XBee Pro S2B*. [8]

Parámetros	Descripción
Alcance Línea de vista	1000m
Alcance Interiores	100m
Frecuencia	2,4GHz
Velocidad de Datos	1Mbps
Voltaje de Operación	5V

Para complementar el módulo *XBee Pro* se dispone de la tarjeta *XBee Explorer* ver imagen 1.8 con un regulador de voltaje a 5V y cuatro pines en la parte posterior *Din, Dout, 5V, GND* para fácil acople a la placa principal.

SISTEMA DE ALIMENTACIÓN

Para el Sistema de alimentación se considera dos tipos de baterías:

Litio y Polímero (Lipo)

Las baterías *Lipo* son recargables se utilizan por lo general en los sistemas de radiocontrol, especialmente en carros a escala; son baterías ligeras de gran capacidad, esto quiere decir que almacenan gran energía en tamaño reducido, tienen una tasa de descarga para alimentar sistemas eléctricos más exigentes. Uno de los problemas que se tiene con estas baterías de *Lipo*, es debido al uso y la presencia de un electrolito volátil puede llegar a dañarse con facilidad y requieren un cuidado a la hora de carga y descarga.



Figura 1.9 Batería Lipo^[4]

En este proyecto se va a utilizar la batería de 3s a 11.1v de 5000mAh para el sistema del robot móvil debido al consumo de los motores ver figura 1.9.

Iones de Litio (Li-ion)

Para el sistema de sensores y placa principal del sistema se utilizó baterías de iones de litio, denominadas baterías de Li-ion, posee características semejantes a las Lipo pero debido al arranque de los motores se descargarían en corto tiempo, es por eso que solo van a ser usadas en el sistema de alimentación de sensores y transmisión de datos ver figura 1.10.



Figura 1.10 Batería Lio-ion^[4]

Se usaron dos tipos de alimentación con el objetivo de no mezclar la parte control con la parte de potencia y poder evitar que si se descargan las baterías de alguno de los sistemas el otro seguirá funcionando por separado.

SOFTWARE

Interfaz gráfica

Para la interfaz gráfica se utilizó el *software Visual Studio* y se programó en lenguaje *c#*, dentro de la programación se realizó una conexión con el puerto serial, para poder recibir los parámetros obtenidos y poder visualizarlos en la interfaz gráfica.

Visual Studio

Es un programa que sirve para el desarrollo de aplicaciones de escritorio o *web*. Ya sea en consola o en interfaz gráfica. Este capítulo se enfocó únicamente en el desarrollo de la interfaz gráfica. *Visual Studio* tiene un entorno de desarrollo muy interactivo ya que el programador puede añadir botones, *etiquetas*, cajas de texto, etc. Se pueden modificar su aspecto visual haciéndolo interesante para el usuario. Posee un depurador que ayuda a detectar errores o advertencias dentro del código fuente.



Figura 1.11 Logo Visual Studio ^[9]

Arduino

La plataforma de *Arduino* se puede utilizar para programar ciertos controladores ayudando a la realización de proyectos de código abierto, está enfocada a un *hardware* y *software* que facilita su uso. El funcionamiento de la placa de *Arduino* permite leer las entradas adquiridas de elementos como luz, temperatura, humedad, gases del ambiente y convertirlo en una señal de salida.

Next Generation Configuration Platform for Xbee/RF Solutions (XCTU)

XCTU es un software multi-plataforma que permite interactuar módulos *Zigbee/RF*. Es una aplicación gratuita compatible con *Windows*, *MacOS* y *Linux*, diseñada para ejecutar aplicaciones con los módulos RF de Digi mediante una interfaz gráfica; y permite la visualización gráfica de la red, su configuración y arquitectura es sencillas en redes inalámbricas del tipo PAN (Personal Area Network).

La configuración es amigable para diversos dispositivos en diferentes condiciones remotas; la actualización es automática en el sistema y mantiene dos consolas API y AT específicas que facilitan la configuración.

2. METODOLOGÍA

En el proyecto se utilizó métodos del tipo analítico, exploratorio, experimental y deductivo; empezando con el análisis de la situación actual del problema en la unidad educativa e información obtenida de la Secretaría de Ambiente del Municipio de Quito.

2.1. Método Analítico

El método analítico permitió conocer y aprender el funcionamiento de todos los elementos y sus partes, así como el *software* que se utilizó, las librerías necesarias que permitieron que el proyecto funcione.

2.2. Método Exploratorio

Este tipo de investigación ofreció un primer acercamiento al problema que se pretende estudiar; en este caso no se dispuso de una plataforma móvil para medir parámetros de calidad del aire. La investigación de tipo exploratoria se realizó para conocer el tema que se abordó, lo que permitió familiarizarse con algo desconocido hasta el momento.

Como resultado se obtuvo un panorama superficial del tema, pero como en toda investigación son los primeros pasos inevitables que se tienen que realizar. El fin de este

tipo de investigación va a ser plantear una hipótesis del tema para que retomen en nuevas investigaciones más exhaustivas o solo se obtendrá la línea de investigación inicial.

2.3. Método Experimental

El método experimental se usó al momento de realizar pruebas para verificar el funcionamiento del sistema de medición de parámetros de calidad del aire, y también el funcionamiento de la red punto a punto que usó el robot móvil y la interfaz que se instaló en el ordenador de la unidad educativa.

2.4. Método Deductivo

Mediante este método se obtuvo conclusiones y puntos importantes a tomar en cuenta como resultado de pruebas realizadas en campo; pudiendo establecer un rendimiento más óptimo y eficiente del sistema.

En el proyecto se utilizó el método inductivo empezando con el análisis de la situación actual de la Unidad Educativa “San Martín” información obtenida de la Secretaría de Ambiente del Municipio de Quito y artículos científicos.

Posteriormente se realizó el diseño e implementación del sistema de robot móvil que contó con sensores para medir parámetros de calidad del aire como monóxido de carbono, metano y ozono; también contó con dispositivos inalámbricos que permitieron la implementación de una red punto a punto para la transmisión de datos.

Esto permitió realizar mediciones de la contaminación generada en el sitio especificado anteriormente. Finalmente para evaluar el desempeño del diseño se realizaron mediciones en la Unidad Educativa “San Martín” y como universo de estudio fueron las aulas y lugar de mayor afluencia estudiantil de la misma institución.

El método experimental se usó al momento de realizar pruebas para verificar el funcionamiento del sistema de medición de parámetros de calidad del aire, y también el

funcionamiento de la red punto a punto que usó el robot móvil y la interfaz que estará instalada el ordenador de la unidad educativa.

La investigación del proyecto se realizó con información de fuentes locales, libros y artículos científicos; tanto su diseño e implementación se desglosa en tres etapas:

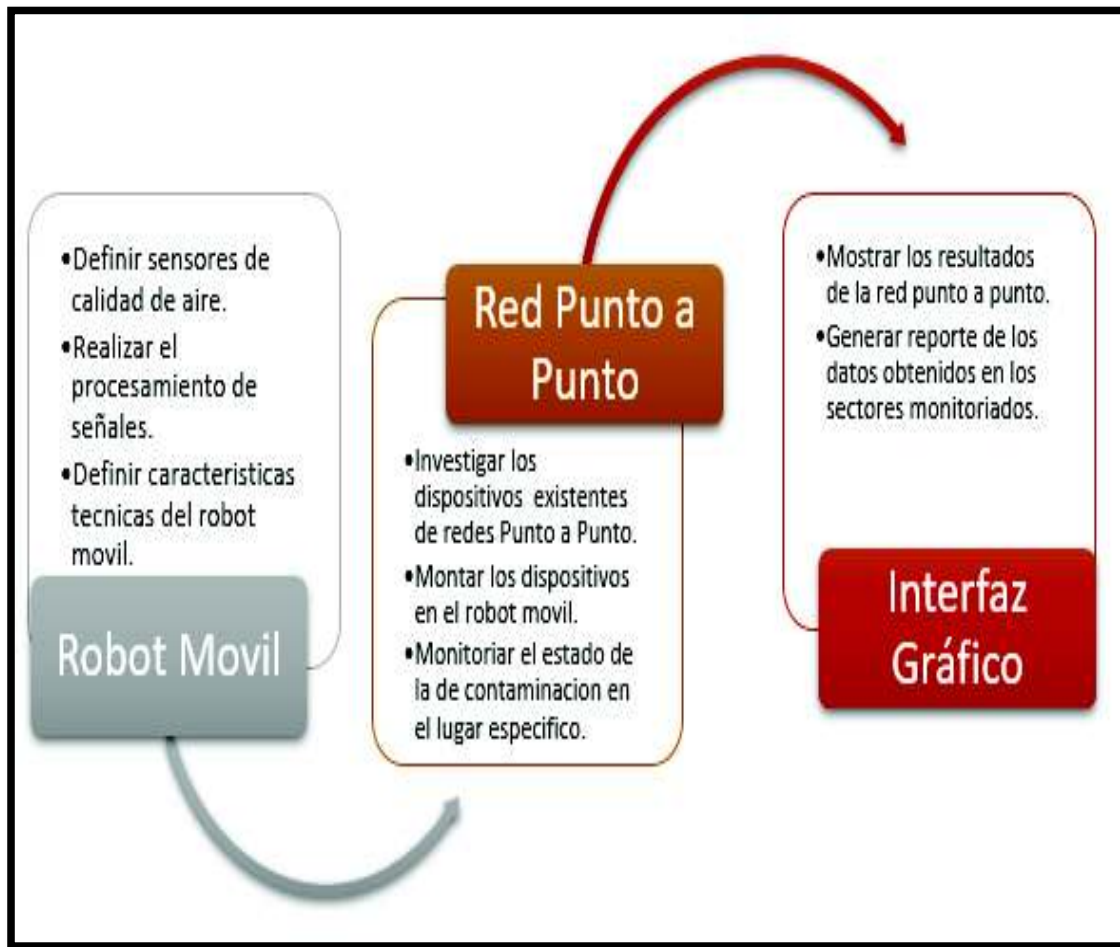


Figura 2.1 Diagrama de proceso para la Implementación

Para realizar este proyecto se inició con un análisis de la situación ambiental y sus efectos a la salud en la comunidad educativa, debido a la ubicación geográfica de la misma ya que se encuentra cercana a una vía de alto tráfico como es la avenida Simón Bolívar y a la cercanía de fábricas como es TANASA y la Fosforera Nacional.

Es necesario implementar mecanismos para poder llevar estadísticas de los niveles normales y críticos de estos gases y poder prevenir sus daños a la salud.

La implementación conto con una primera etapa contó con la construcción del *robot* móvil para lo cual se ensablo y configuro los distintos elementos requeridos (motores, suspensión, llantas, *chasis*, entre otros), los cuales se encontraron disponibles en el mercado. Se logró que el *robot* cumpla con las necesidades planteadas, es decir que pueda movilizarse en terreno no pantanoso ni acuático.

Se ensablo en carcasa de aluminio, con una protección resistente para circuitos integrados, y de ayuda para la reducción de la interferencia electromagnética sus siglas en inglés (EMI), su tamaño pequeño permite que pueda acceder a lugares reducidos, y se equipó con sensores que midan los parámetros de calidad del aire ya establecidos.

La segunda etapa fue definida como la configuración de la PAN que permitió la transmisión de datos desde el *robot* móvil hasta una computadora mediante una interfaz gráfica o *software* HMI (Interfaz Hombre Máquina). La interfaz gráfica fue una aplicación de escritorio para *PC's* amigable con el usuario y se desarrolló en *Visual C share*; por medio de esta se presentarán los parámetros de la calidad del aire obtenidos por el *robot*

En la tercera etapa se validó el desempeño del sistema, se realizó mediciones de la contaminación ambiental en diferentes escenarios como el aula computación, área de recreación, el comedor y las áreas de estudio (aulas) de la Unidad Educativa "San Martin" ubicada en el sector de la Argelia en la ciudad de Quito Provincia de Pichincha.

3. RESULTADOS Y DISCUSIÓN

3.1. Diseño del prototipo

Se presenta paso a paso el desarrollo del sistema que permite medir parámetros de calidad del aire, este sistema está conformado por el vehículo explorador el cual está formado por

un chasis de aluminio con 4 motores y ruedas una suspensión independiente lo cual facilita una tracción en terrenos irregulares, posee una rejilla con huecos lo cual es eficiente para el montaje de carga útil también dispone de una batería de LiPo de 12.4 voltios y 5000 mAh que ayudará con la autonomía del vehículo explorador tendrá una comunicación de radio control para el manejo el prototipo en diferentes escenarios.

Otra parte del sistema es de sensorización la cual consta de 5 sensores (Dióxido de Carbono CO₂, Metano CH₄, Ozono O₃, Temperatura, Humedad) estos datos son adquiridos por un micro-controlador Atmega328 que está contenido en una placa principal *Arduino Nano* estará conectado a una batería de *Lipo de 7.4 voltios y 300mAh*.

Adicional se dispone de la parte de transmisión de datos conformada por un *XBee Pro S2B* que tiene un alcance en línea de vista de 500 metros de distancia a un receptor *XBee USB*, las baterías de Li-ion contenidas en una *power back* de 5 voltios proveerán la autonomía de la comunicación inalámbrica del prototipo.

La interfaz amigable desarrollada en *Visual Studio* le permite ver al operador datos adquiridos de manera fácil y sencilla.

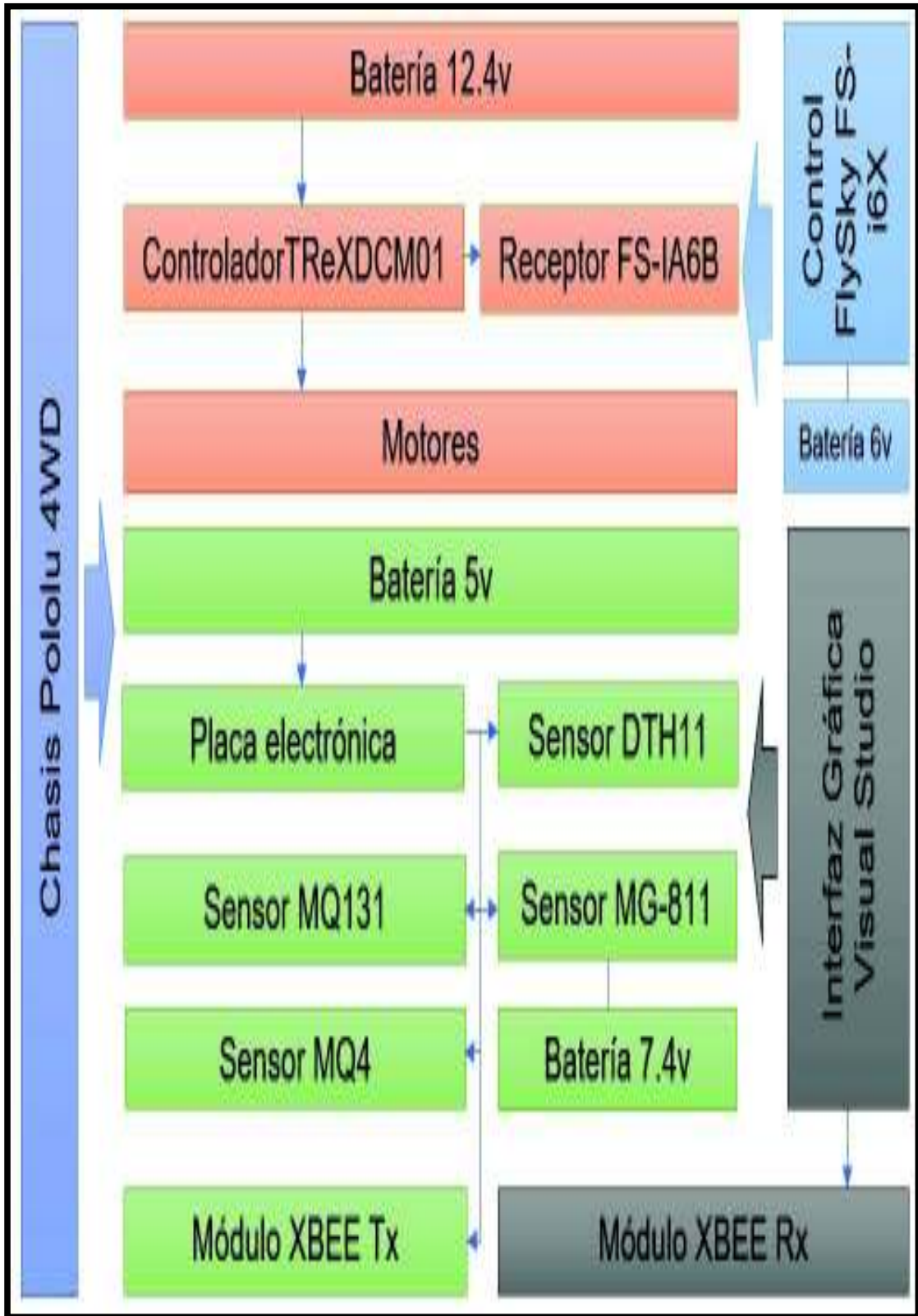


Figura 3.1 Esquema general del prototipo

Circuito de elementos electrónicos y distribución de pines

TRex DCM01

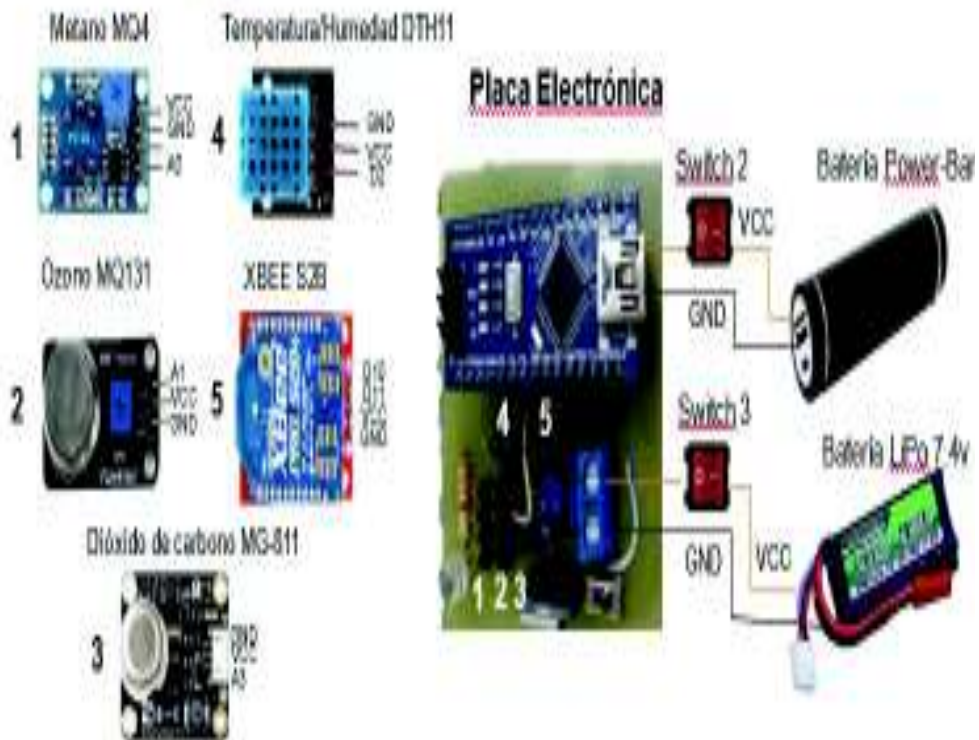
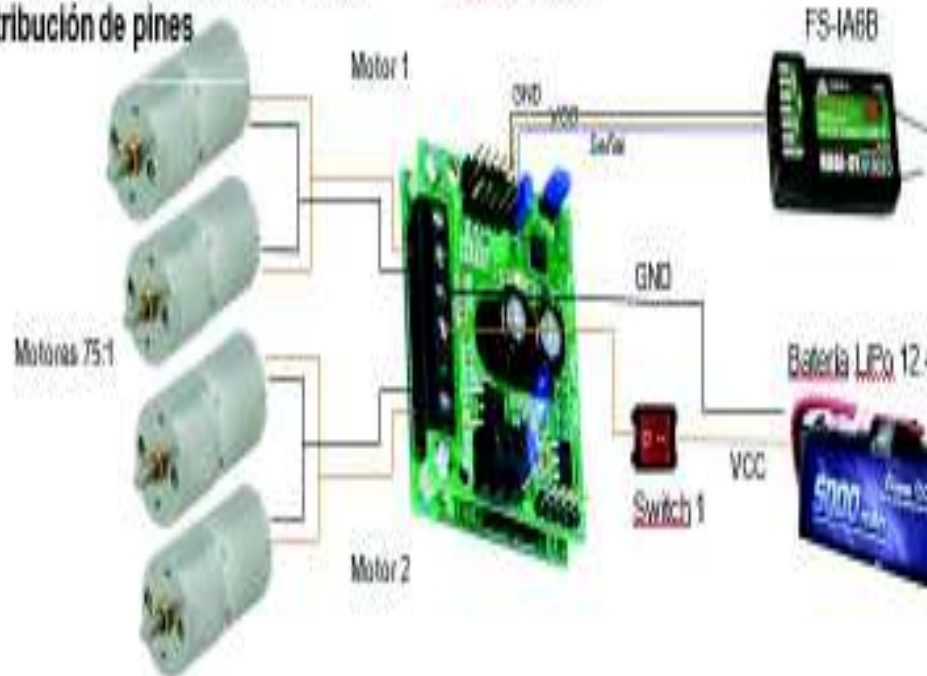


Figura 3.2 Circuito de elementos electrónicos y distribución de pines

La primera parte del circuito de los elementos electrónico corresponde al módulo *TRex DCM01* en el cual se conecta el FS-IA6B que es el receptor del control remoto; este módulo se alimenta con VCC=5v, GND que es tierra y la señal de los pulsos PWM al canal 1 y 2; el módulo actúa a estas señales para que se pueda comandar el robot en las direcciones atrás, adelante, izquierda, derecha.

Este módulo tiene una bornera de 6 conectores, los del centro corresponden a la alimentación que en este caso es una batería de *LiPo* de 12.4v y en los extremos se conecta los motores del chasis pololu 4WD. Por último, dispone de un *switch* de encendido y apagado del sistema.

La segunda parte consta de una placa electrónica la cual está conectada a un *Arduino Nano* alimentado por una *power bank* con un *switch* de encendido y apagado; en éste se conecta el sensor de Metano MQ4 al pin A0, el sensor de Ozono MQ131 al pin A1 y el sensor de dióxido de carbono MG-811 al pin A3 para éstos se utiliza una adquisición análoga a digital, el sensor DHT11 al ser digital se conecta al pin D2 del Arduino por medio del protocolo I²C.

Para la comunicación inalámbrica y transmisión de datos se conecta el módulo *XBee Pro S2B* a los pines D10, D11 que corresponden a transmisión (Tx) y recepción (Rx) de datos y también los pines de VCC a 5v y GND. El sensor MG-811 al consumir 500mA y para que trabaje de una manera óptima dispone de su propia alimentación por medio de una batería de *LiPo* de 7.4v. La placa electrónica también posee dos *leds* el uno indica que él está recibiendo alimentación de voltaje y el otro que se estableció la comunicación *XBee Pro*.

Tabla 3.1 Rangos de medición y recomendación

		Rango de Medición		Recomendable para vivir	
		Valor Min	Valor Max	Valor Min	Valor Max
Sensor DHT11	Temperatura	0°C	50°C	37°C	55°C
	Humedad	20% rH	90% rH	20% rH	80% rH
Sensor CO2		400 ppm	10000 ppm	300 ppm	900 ppm
Sensor CH4		0%ppm	100%ppm	0%ppm	1%ppm

3.2. Diseño de la placa electrónica

Esta etapa abarcó el diseño de la placa que estará comandada por el *Arduino Nano*, para realizar esta placa se usó el *software* ISIS y para realizar el esquema de conexiones el *software* ARES ambos proporcionados por *PROTEUS*. Se procedió a poner los elementos necesarios como conector de batería, *switch* para encendido y apagado de todo el circuito, regulador de voltaje LM7805 el cual sirvió para la alimentación del sensor de CO2 y un led para indicar si la placa está o no encendida, ver figura 3.3

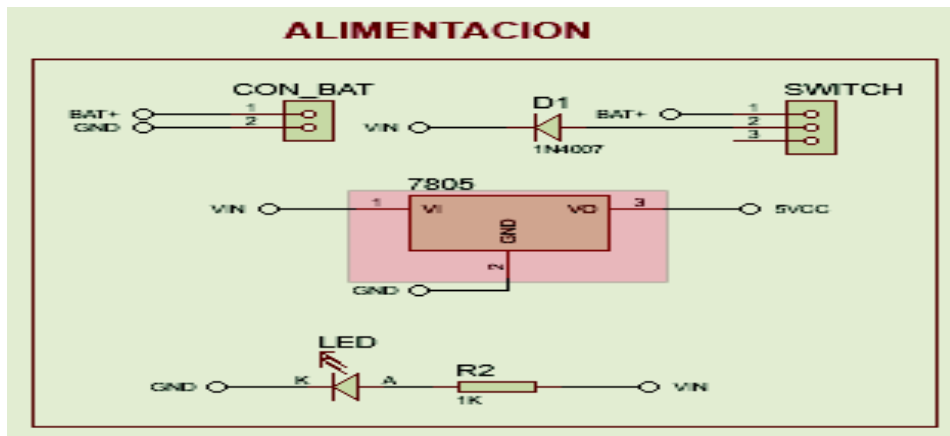


Figura 3.3 Conexiones de alimentación

Las versiones superiores a la 8.0 cuentan con la facilidad de incluir las placas de desarrollo *Arduino* de esa forma hace más fácil el realizar las conexiones del *Arduino Nano* al resto de pines que se van a conectar, ver figura 3.4.

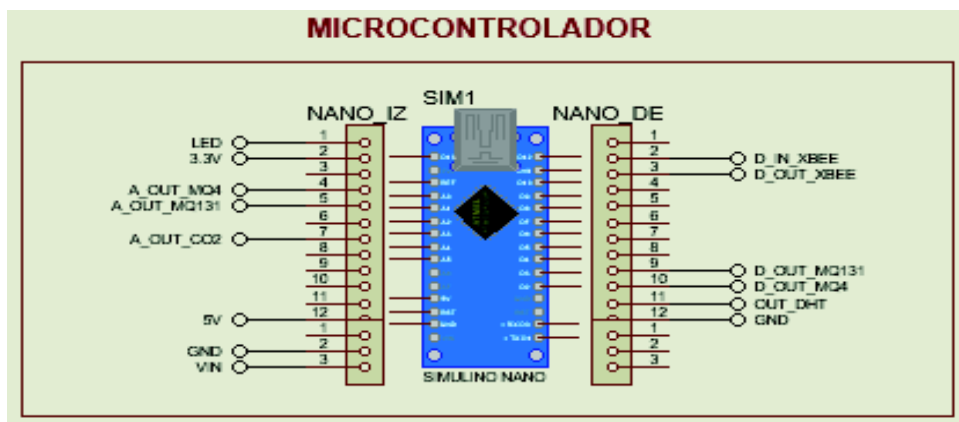


Figura 3.4 Conexiones Micro-controlador

Posteriormente y siguiendo con el diseño de la placa en la parte de sensores se colocará pines de conexión para cada sensor como DHT11, MQ4, MQ131 y CO2 de igual forma pines para un dispositivo de comunicación *XBee Pro* y un *led* de visualización, ver figura 3.5.

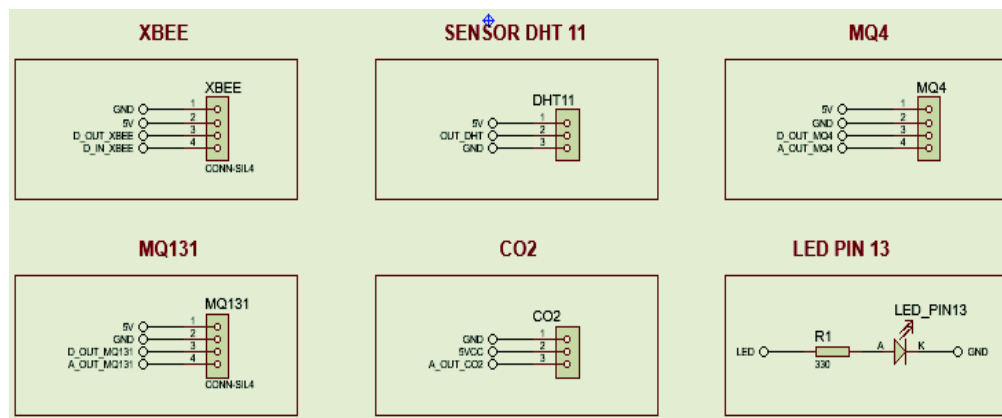


Figura 3.5 Conexiones de sensores

3.3. Diseño de conexiones ARES

Para este diseño, *Proteus* cuenta con la herramienta *Ares* la cual permite mediante el circuito diseñado anteriormente crear una placa del circuito. Para esto solo se debe colocar los elementos de manera adecuada y el *software* se encargará de realizar las pistas por el lugar adecuado y de un tamaño previamente establecido ver figura 3.6.

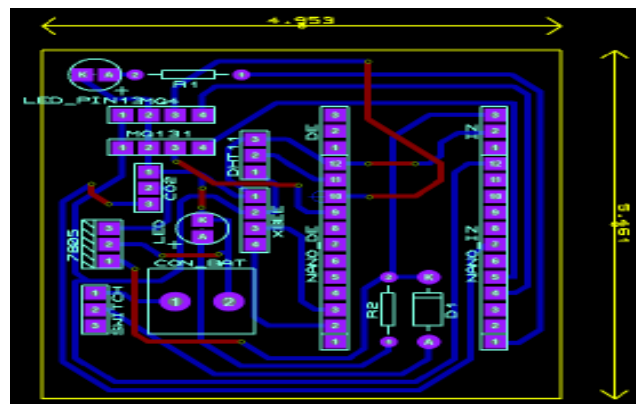


Figura 3.6 Circuito en Ares

Antes de realizar la placa se puede hacer una visualización en 3D de cómo están ubicados los elementos de la placa principal ver figura 3.7.

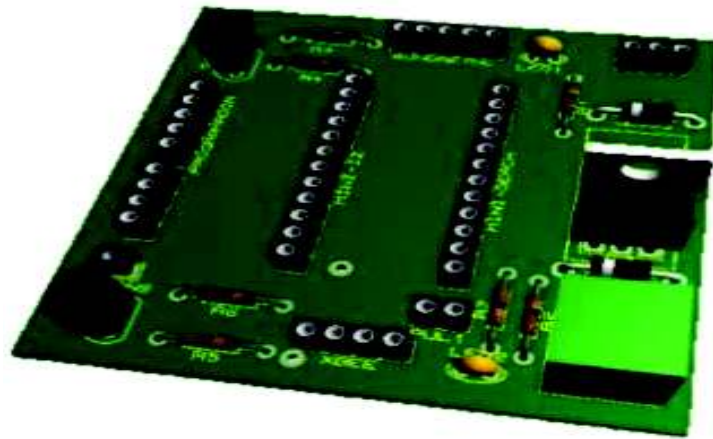


Figura 3.7 Visualización 3D

El software *Proteus* permite generar un archivo de impresión donde se puede obtener el PCB en el cual se va a fabricar las pistas, que son las únicas necesarias para la realización de la placa en este caso las pistas del circuito se observan en la figura 3.8.

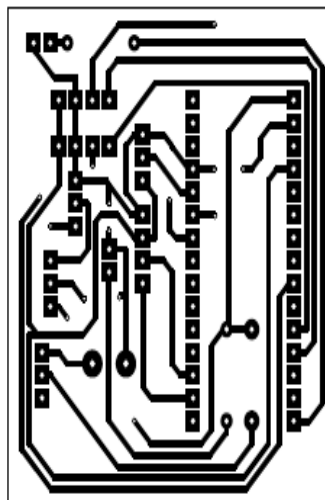


Figura 3.8 Diseño de placa principal

Una vez obtenidas las pistas se procedió a imprimir en papel transfer de preferencia azul, se imprimió en la parte no brillante y la impresión fue de tipo láser. Hecho esto, se colocó el papel sobre la baquelita con la impresión sobre la parte de cobre, se sujetó de forma

adecuada para evitar que se mueva y se procedió a planchar para que la impresión se quede en el cobre, esto pasó aproximadamente 10 minutos después de haber planchando, ver figura 3.9



Figura 3.9 Elaboración de la placa principal

Una vez que se terminó de planchar, se retiró el papel, para tener únicamente las pistas de cobre, se sumergió la placa en ácido férrico para que este desintegre el cobre sobrante y solo deje las pistas. A continuación se procedió a limpiar bien la placa lavándola y quitando la parte de tinta que es la única que quedó impresa. La placa quedó lista y se procedió a realizar los huecos con la ayuda de un taladro y una broca de 1mm de grosor para soldar los pines. Para finalizar, se procedió a soldar todos los elementos, y se probó su funcionamiento en la placa como se puede ver en la figura 3.10.



Figura 3.10 Placa principal con elementos soldados

La placa principal y los sensores se colocaron en una caja aislante o de protección al agua, para su seguridad y portabilidad cuando sea instalada en el *robot* móvil ver figura 3.11.



Figura 3.11 Caja negra con placa principal

La etapa de control está comandada por el *Arduino Nano* que tiene un micro-controlador ATmega 328, al cual están conectadas las salidas para el módulo *XBee Pro*, los conectores para los sensores y un led que se enciende cuando la placa está alimentada con voltaje. En esta parte hay que tomar en cuenta los pines a los cuales fueron conectados los sensores, debido a que algunos son digitales y otros analógicos, una vez conocido los pines se implementó los algoritmos para adquisición y envío de datos.

En el diagrama de flujo de la figura 3.12 se observó cómo está la lógica del código *Arduino*, primero, se inició el programa; segundo, se declaró librerías para el uso de los sensores como el de temperatura, ozono, humedad; tercero, se abrió comunicación serial a través de los *XBee Pro* y se realizó la adquisición analógica de los sensores; cuarto, se realizó las operaciones para obtener los datos adquiridos; quinto, se realizó el formato de la trama para visualizar y sexto final del programa.

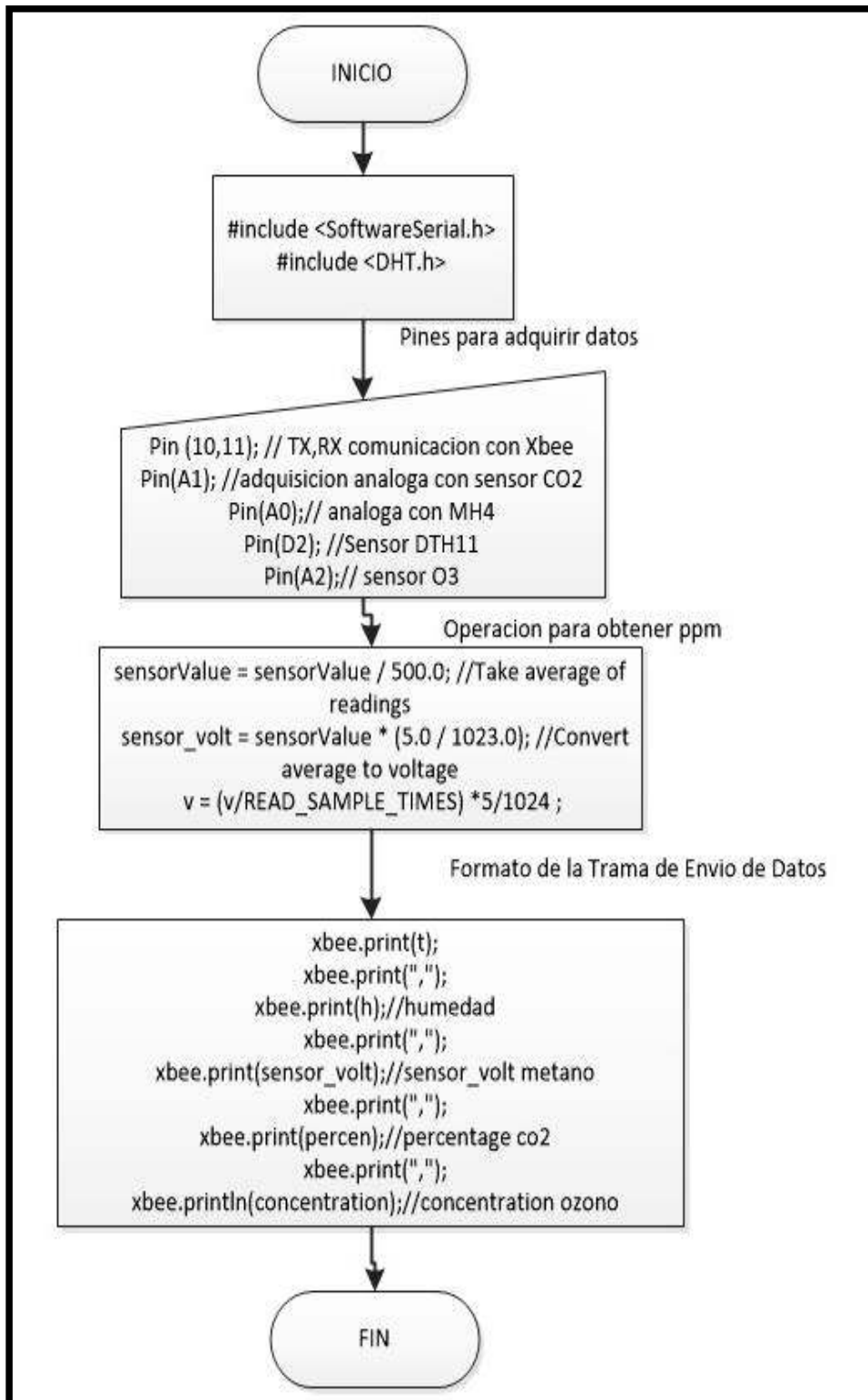


Figura 3.12 Algoritmo en *Arduino*

3.4. Configuración de la red de sensores inalámbricos



Figura 3.13 Conexiones *XBee Pro* con PC mediante *USB*

Para la configuración de la WSN se utilizó el *software* de la empresa *digi XCT.exe* se utilizó una topología punto a punto. Para configurar los parámetros se abrió el *software XCTU.exe* se dio un doble *click* y se realizó los siguientes pasos:

- a) Se conectó el *XBee Pro* al computador por medio de un cable *USB*, y un *USB Explorer*, y previamente se instaló el *software* en el computador ver figura 3.13.
- b) Al abrir el programa se mostrará en la pantalla el puerto *Com* al que está asociado el módulo *XBee Pro*, el cual se selecciona y por último un *click* en el botón *finish* ver figura 3.14.

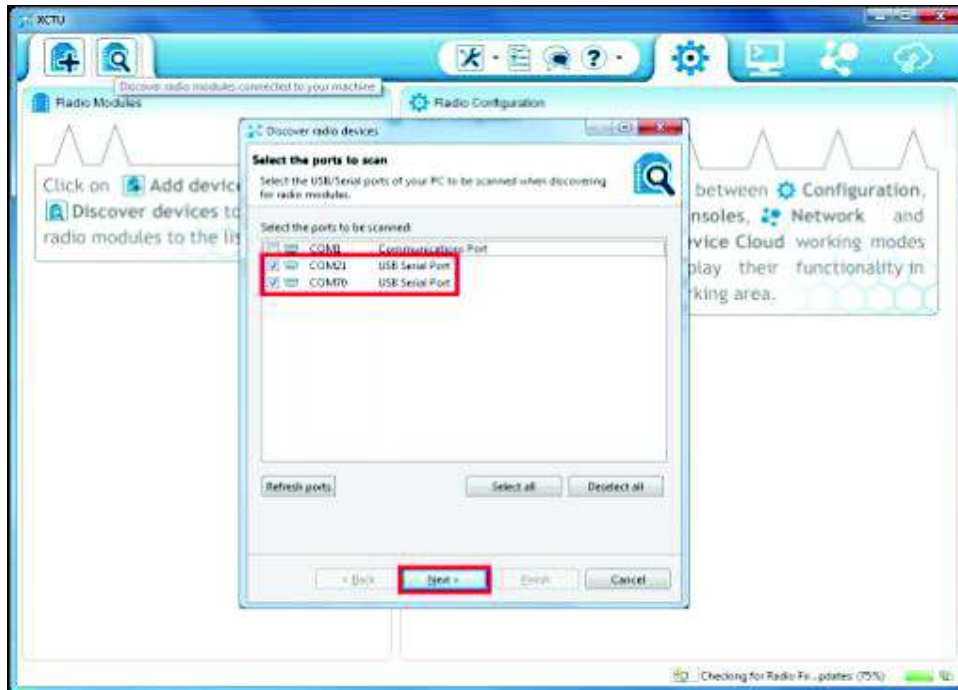


Figura 3.14 XCTU Puerto Com del Módulo XBee Pro

- c) Se procede a agregar un nuevo dispositivo en el *software* XCTU, para ello se da clic en el símbolo marcado con el símbolo más, en la parte superior izquierda, ver figura 3.15.

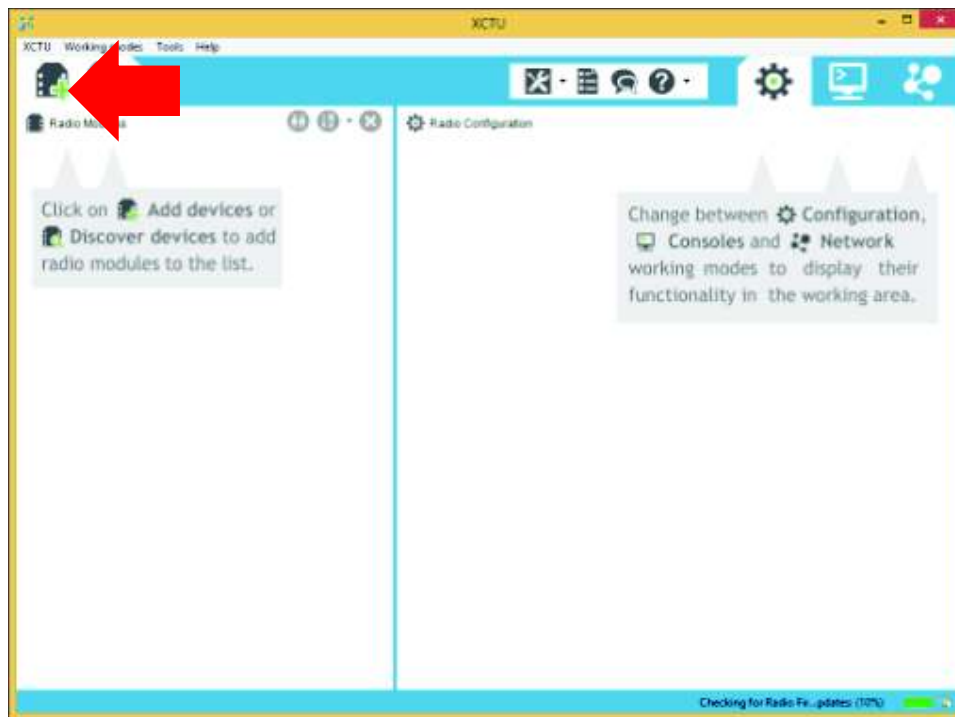


Figura 3.15 Agregar nuevo dispositivo

- d) Al realizar doble clic en el dispositivo encontrado en el lado izquierdo de la pantalla aparecerán los datos de dispositivo, como nombre ID, etc. En el *software XCTU* se puede apreciar en la figura 3.16.

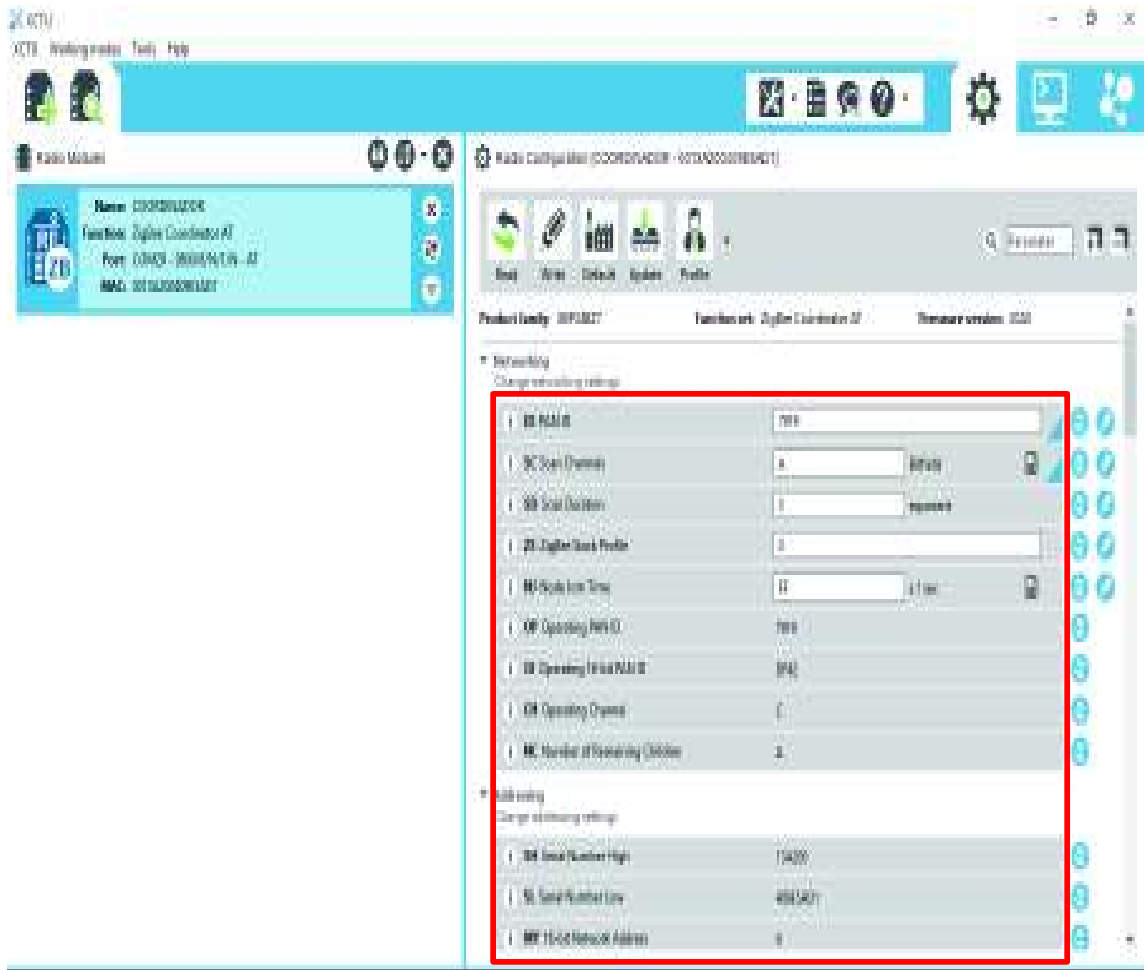


Figura 3.16 Parámetros en el *software* XCTU

En la parte de configuraciones de la figura 3.16, se identifica *PAN ID*, el cual debe ser configurado el mismo en todos los dispositivos, es una dirección a través de la cual van a enlazarse, formando una red *PAN (Personal Area Network)*, en el *SC Scan Channel* el número por defecto que se usa es 7FFF, en *ZS Zigbee Stack Profile* se usa la opción 2 ya que corresponde al tipo de *zigbee PRO* usado, la siguiente configuración es el *NI (Node Identifier)*, el cual se denomina *COORDINADOR* y por pruebas se denominarán *NODO – S2B1* y *NODO – S2B12*; También se colocó las direcciones *SH* y *SL*, cada *XBee Pro* tiene un *SH* o *Serial High* y *SL* o *Serial Low*. Estos son sus números de serie que vienen definidos de fábrica. *DH* y *DL* son los números de serie del módulo al que se envía el dato.

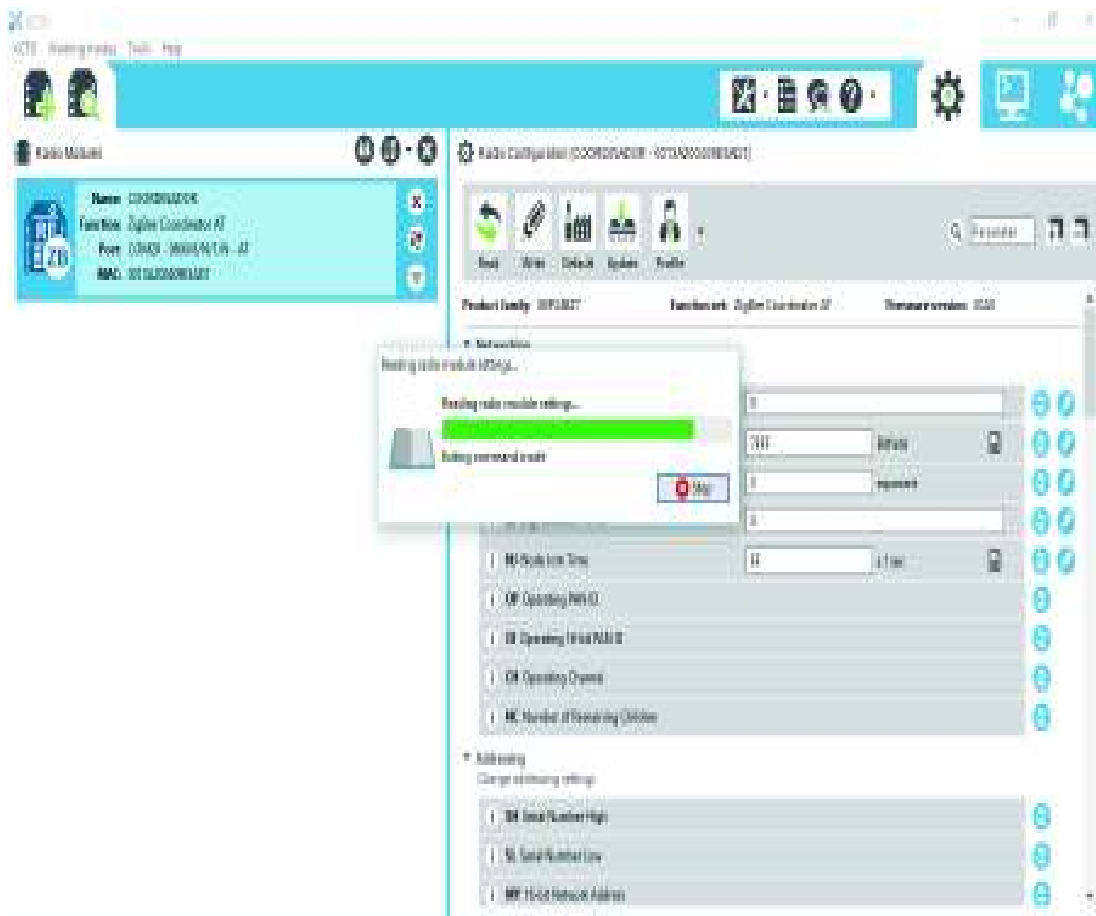


Figura 3.17 Datos cargados con XCTU.

- e) Para la configuración del Dispositivo como coordinador se modifican los parámetros de la tabla 3.2.

Tabla 3.2 Parámetros en XBee Pro

ID PAN ID	7919
SC Scan Channels	A
DH	0
DL	FFFF
NI Nodo Identifier	COORDINADOR
BD Baud Rate	9600

- f) Una vez modificado estos parámetros se realizó *clic* en *WRITE* para grabar los valores en el dispositivo XBee Pro ver figura 3.18.

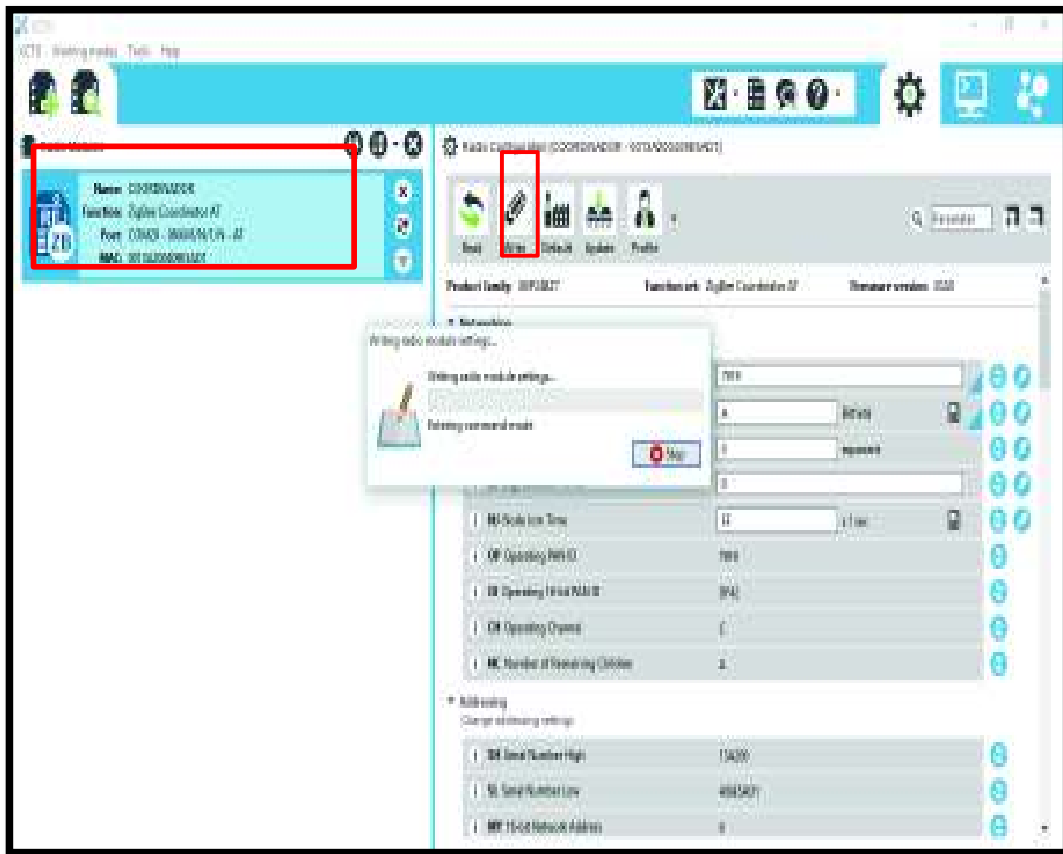


Figura 3.18 Carga de parámetros en módulo

Como se observa en la figura 3.18 se han colocado los campos que se necesitó cambiar para que el coordinador pueda recibir la información del nodo final ubicado en el robot móvil, se observó en la parte superior izquierda que el nodo configurado tiene ya sus respectivos parámetros como *Name*, *Function*, *Port*, *Mac* ver el figura 3.18.

- g) Adicionalmente como este dispositivo es el coordinador de la red, se carga el *firmware* correspondiente, para esto se realiza *click* en *Update* donde se despliega la ventana de la figura 3.19 se selecciona (*Zigbee Coordinador AT*) y después *Update*.

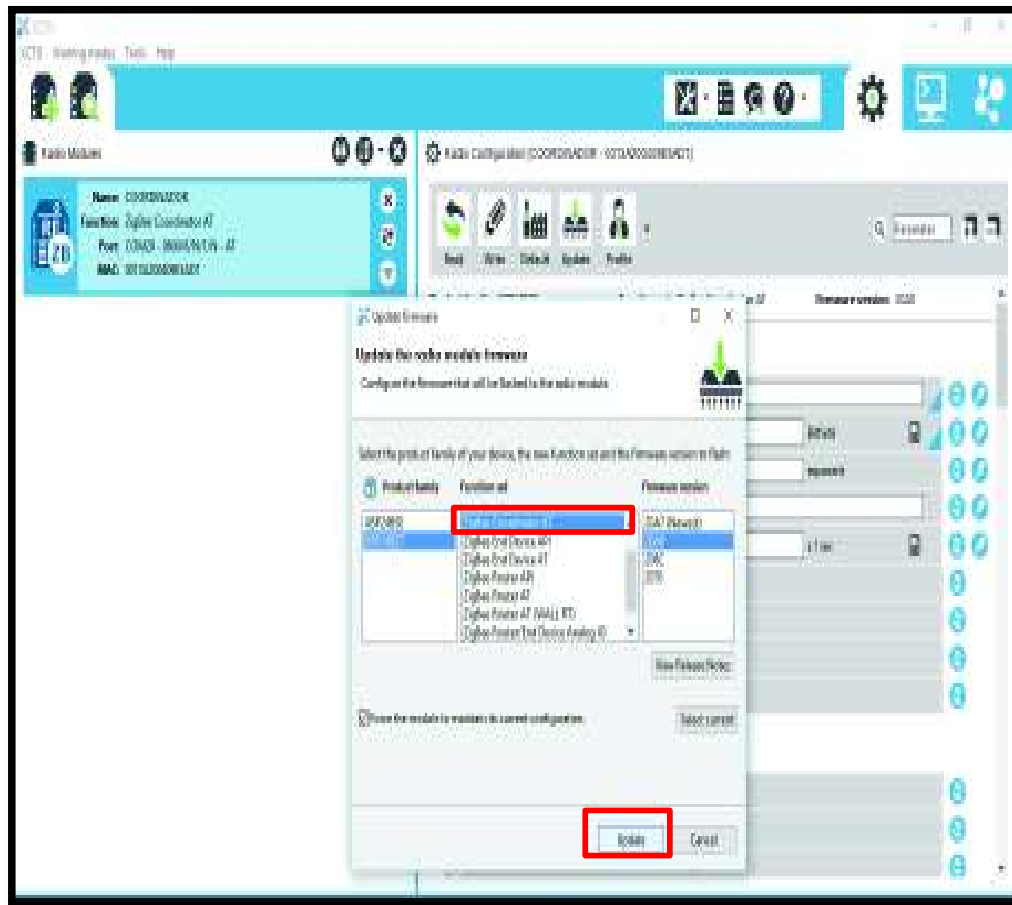


Figura 3.19 Carga de firmware en nodo Coordinador

- h) Para el nodo final se realizó los pasos a, b, c y se cargan los datos de la tabla 3.3, y se realiza el paso f.

Tabla 3.3 Datos nodo

ID PAN ID	7919
SC Scan Channels	A
DH	13A 200
DL	409E5AD1
NI Nodo Identifier	N1
BD Baud Rate	9600

- i) Como es un dispositivo final se cargó el *firmware* de *End Device* ver figura 3.20.

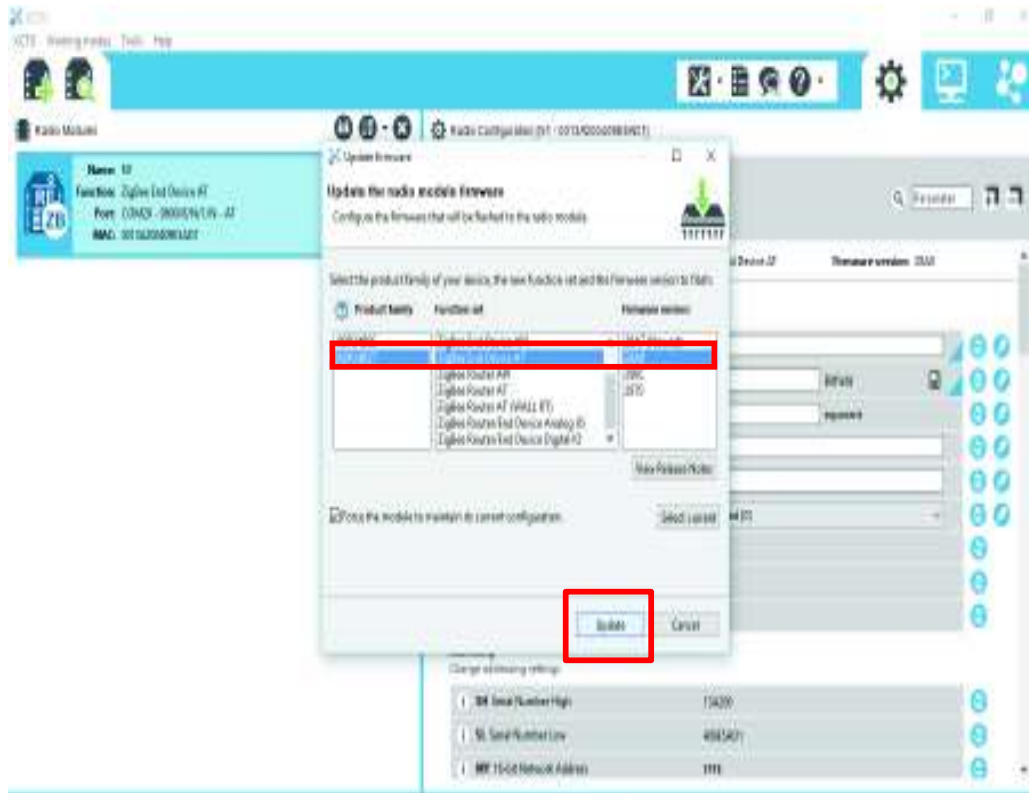


Figura 3.20 Carga de *firmware* en Nodo

Como se observa en la figura 3.20 se han colocado los campos que se necesitó cambiar para que el nodo final pueda enviar la información al nodo coordinador, se puede observar en la parte superior izquierda el nodo configurado tiene ya sus respectivos parámetros como *Name*, *Function*, *Port*, *Mac*.

De esta forma quedan configurados los *XBee Pro* con una topología punto a punto, para probar que esté funcionando la red se puede hacer una transmisión de datos bidireccional mediante alguna consola serial como *RealTerm*.

3.5. Ensamblaje del *Chasis* POLOLU 4WD

Para el ensamblaje del *chasis* móvil se detallan los siguientes pasos.

- a) Soldar cables a los motores para mantener una norma en el sentido de giro el color rojo representara positivo y negro negativo. Realizar este procedimiento con los cuatro motores disponibles ver figura 3.21.



Figura 3.21 Cables soldados a motores

- b) Montar los motores en los sujetadores de plástico y ajustarlos al chasis de aluminio ver figura 3.22.

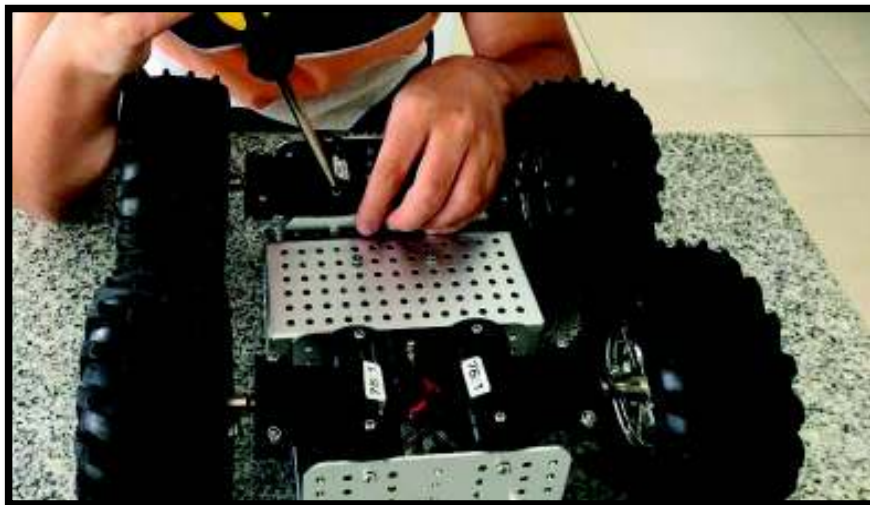


Figura 3.22 Ensamblaje de motores a chasis de aluminio

- c) Realizar una conexión en serie de los motores de la izquierda y otra conexión en serie de los motores de la derecha con borneras ver figura 3.23.

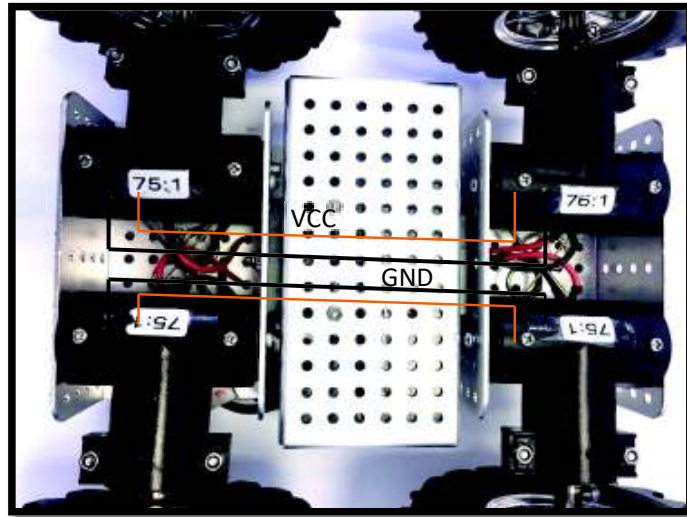


Figura 3.23 Conexión en serie de motores

- d) Una vez realizado la conexión en serie de motores de la derecha y motores de la izquierda, se dispone de dos conexiones, dos cables que representarán giro izquierdo y dos cables giro derecho, ver figura 3.24.

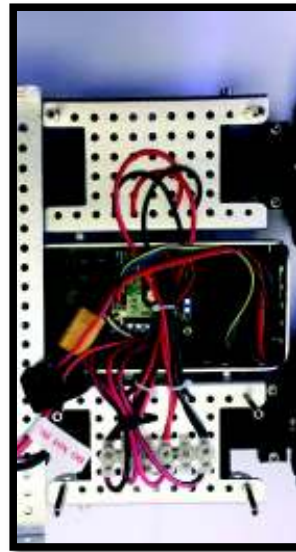


Figura 3.24 Conexión sentido de giro

- e) Realizar la conexión de los motores al *TReX* siguiendo el diagrama de la figura 3.25.

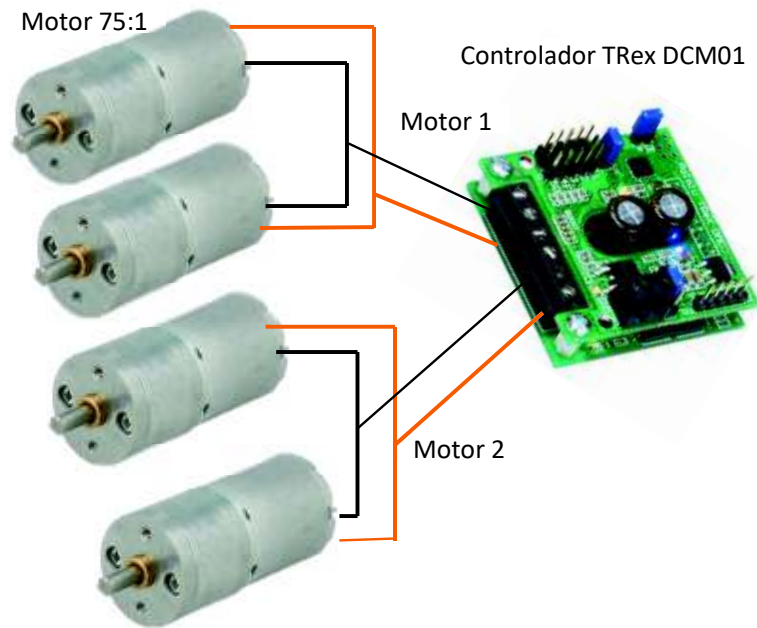


Figura 3.25 Conexión de motores al controlador

- f) Ajustar los cables a las borneras de salida Motor 1, Motor 2; añadir la batería a la bornera de la mitad teniendo en cuenta la polarización GND y VCC ver figura 3.26.

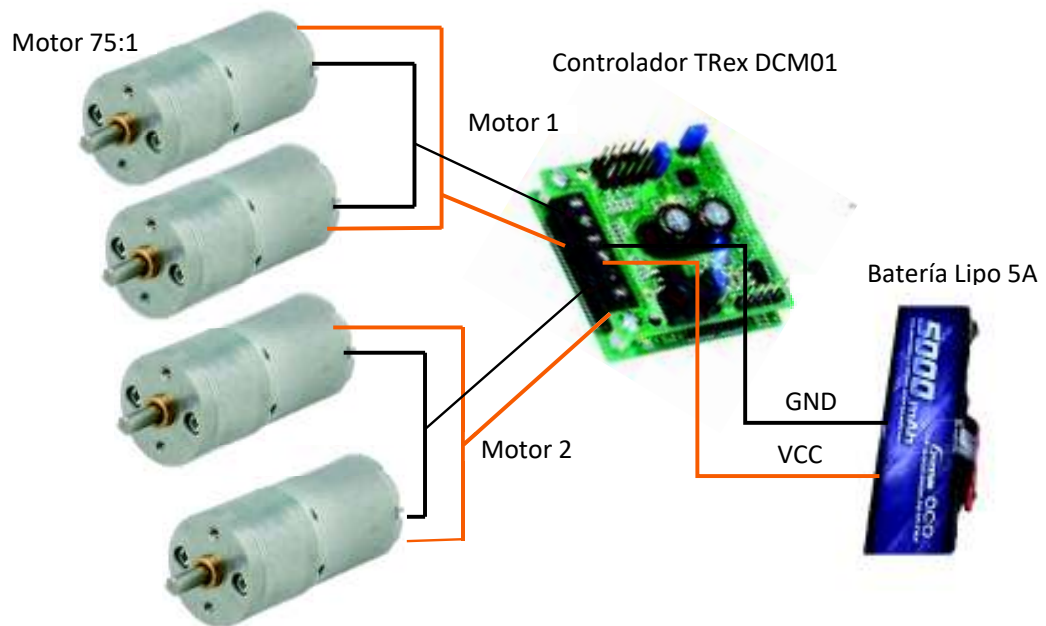


Figura 3.26 Alimentación con batería de *LiPo*

- g) Se añadió al controlador el receptor FS-IA6B para eso se toma en cuenta los tres cables que dispone el receptor (GND, VCC, Señal) ver figura 3.27.

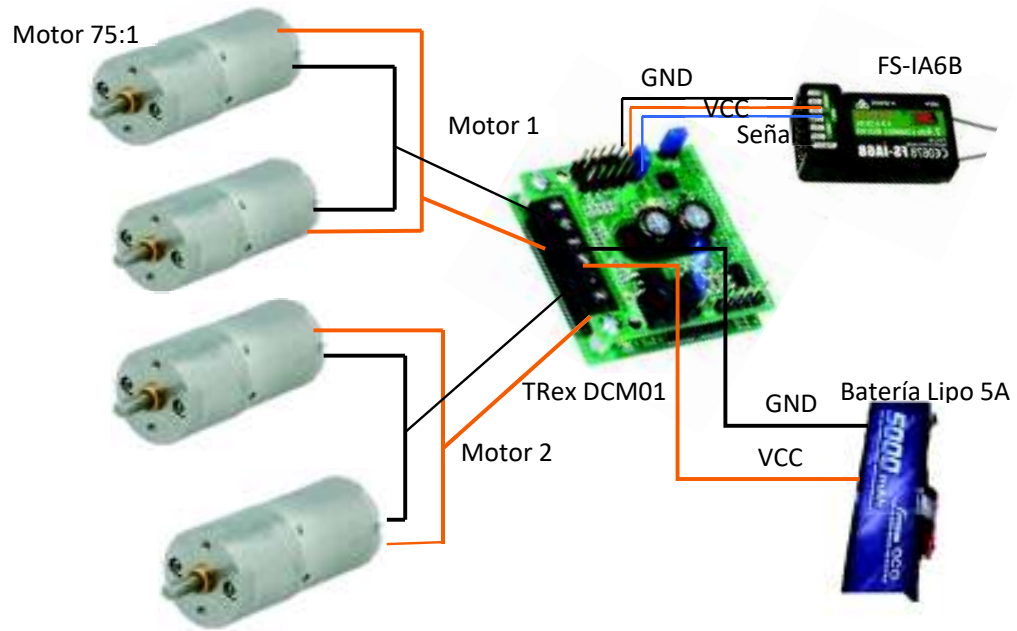


Figura 3.27 Conexiones controlador

- h) Finalmente se ensambló cuidadosamente las partes y placa controladora en la estructura de aluminio y se colocó las ruedas del *robot* móvil, como se observa en la figura 3.28.

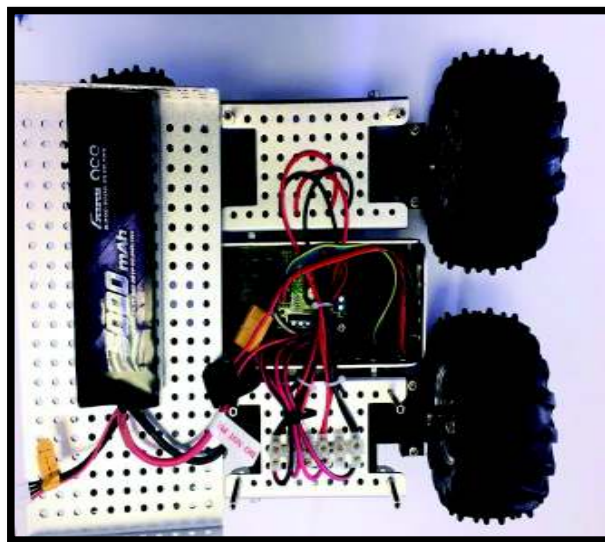


Figura 3.28 Componentes ensamblados en robot móvil

3.6. Emparejamiento del transmisor y receptor FlySky.

El termino *Binding*, hace referencia a emparejar el Transmisor FS-i6 (TX) con el receptor FS-IA6B (RX) con el fin de establecer una comunicación fija, es decir, que así los módulos de TX y RX pierdan energía, la comunicación siempre existirá para enviar las órdenes de control del *robot* móvil, para ello se sigue los siguientes pasos:

- a) Alimentar el receptor FS-IA6B, y colocar un puente en el primer grupo de pines como se puede observar en la figura 3.29.



Figura 3.29 *Binding* FS-IA6B

- b) Una vez conectada las pilas de alimentación del control FS-i6 se enciende teniendo aplastado el botón de *bind* como se puede ver la figura 3.30.

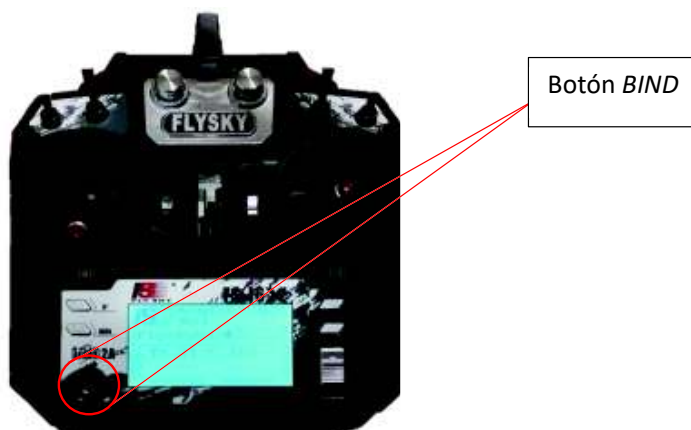


Figura 3.30 Botón *binding*

- c) Realizado los pasos 1 y 2 de manera correcta en el *display* del control se podrá observar el mensaje mostrado en la figura 3.31.

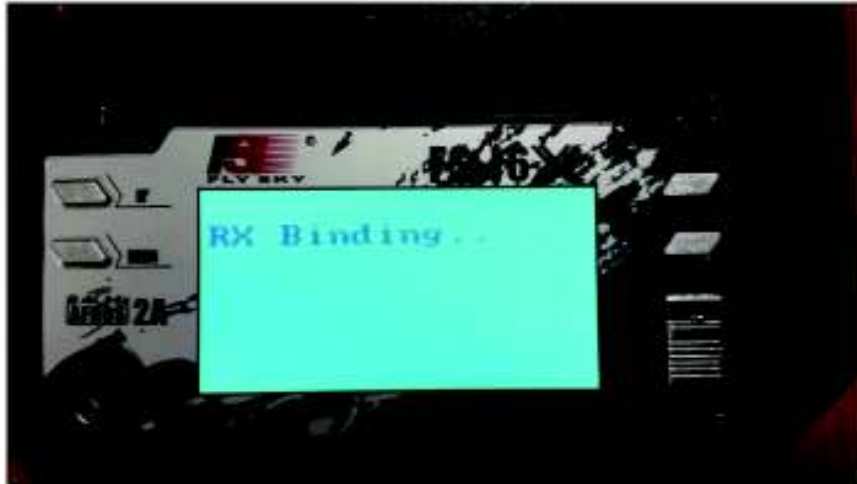


Figura 3.31 Mensaje *binding* en *display* del control

- d) A la espera de unos segundos hasta que se establezca la comunicación se podrá observar el mensaje de la figura 3.32.



Figura 3.32 Control TX y receptor RX enlazados

De esta forma, realizado los procedimientos anteriores se pueden controlar al robot móvil con el control FS-i6 la distancia que puede alcanzar es de aproximadamente 500 metros en línea de vista ver figura 3.33.

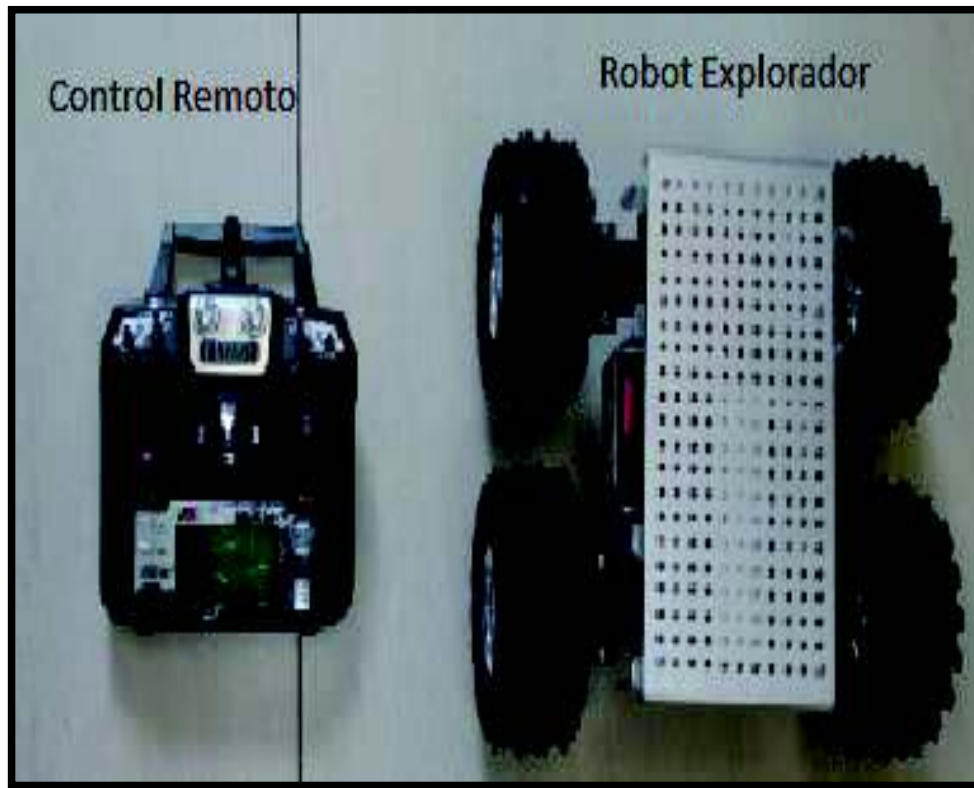


Figura 3.33 Robot móvil con control remoto

3.7. Programación de Interfaz gráfica

Antes de comenzar es importante aclarar que se denomina “proyecto” a la –aplicación o programa que se diseñó en *Visual Studio*. Dicho esto, se decir que antes de comenzar a trabajar es imprescindible crear un nuevo proyecto. Para hacerlo se procede a seguir los siguientes pasos:

- a) Dentro de la interfaz de Visual Studio, se selecciona en la barra de tareas ubicada en la parte superior “ARCHIVO” ver figura 3.34 número 1.
- b) Se debe seleccionar la opción “Nuevo” ver figura 3.34 número 2.
- c) Se debe seleccionar la opción “Proyecto” ver figura 3.34 número 3.

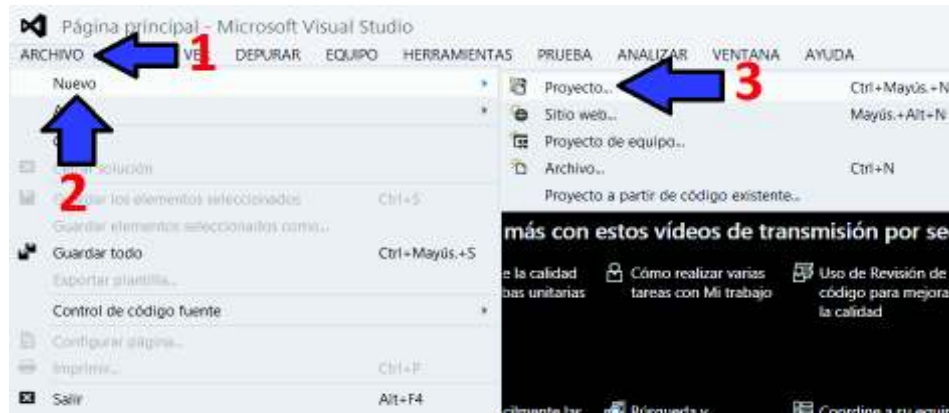


Figura 3.34 Pasos para la crear un proyecto.

- d) En la ventana emergente se procede a seleccionar en las opciones de Visual C# la opción “Windows” ver figura 3.35 número 4.
- e) Se selecciona la opción “Aplicación de Windows Forms” ver figura 3.35 número 5.
- f) Se escribió el nombre que se desea que el proyecto lleve (En este caso fue “Saludo”) ver figura 3.35 número 6.
- g) Se selecciona la ruta donde se desea que el proyecto se guarde. Este paso se puede omitir sin que afecte el comportamiento de este. Así mismo, existe la opción para que el proyecto se guarde automáticamente al momento que se cree el proyecto.

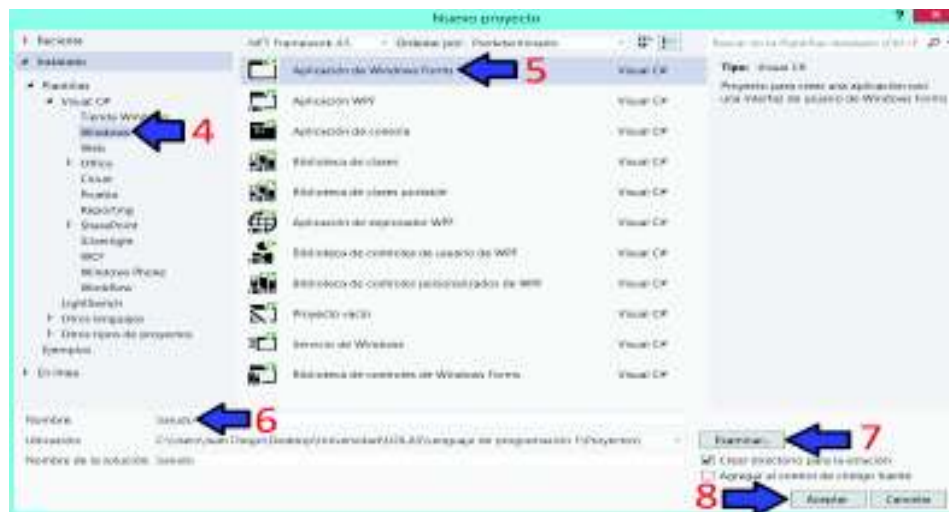


Figura 3.35 Pasos para crear Windows Form

h) Se da clic en el botón aceptar y se obtiene la figura 3.36.

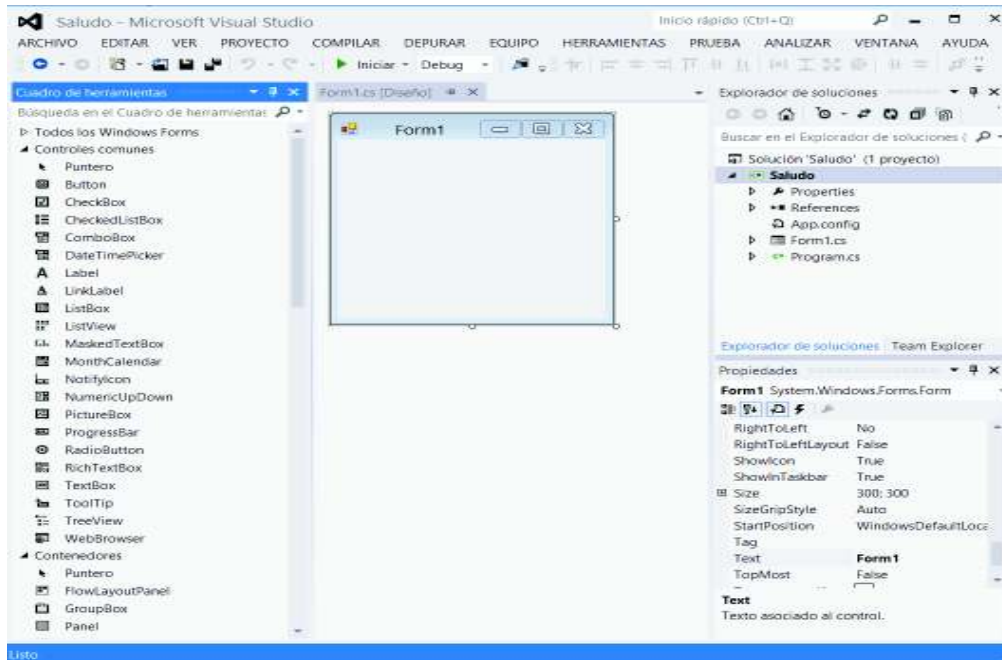


Figura 3.36 Windows Form

Una primera vista de la interfaz se observó en la siguiente figura, donde se tuvo los botones *Start*, *Stop*, *Almacenar*, *Refresh*; posee un cuadro de texto para visualizar la trama recibida con los caracteres enviados desde la placa principal mediante los *XBee Pro al ordenador*, y gráficos para interpretar los datos, se puede ver figura 3.37.

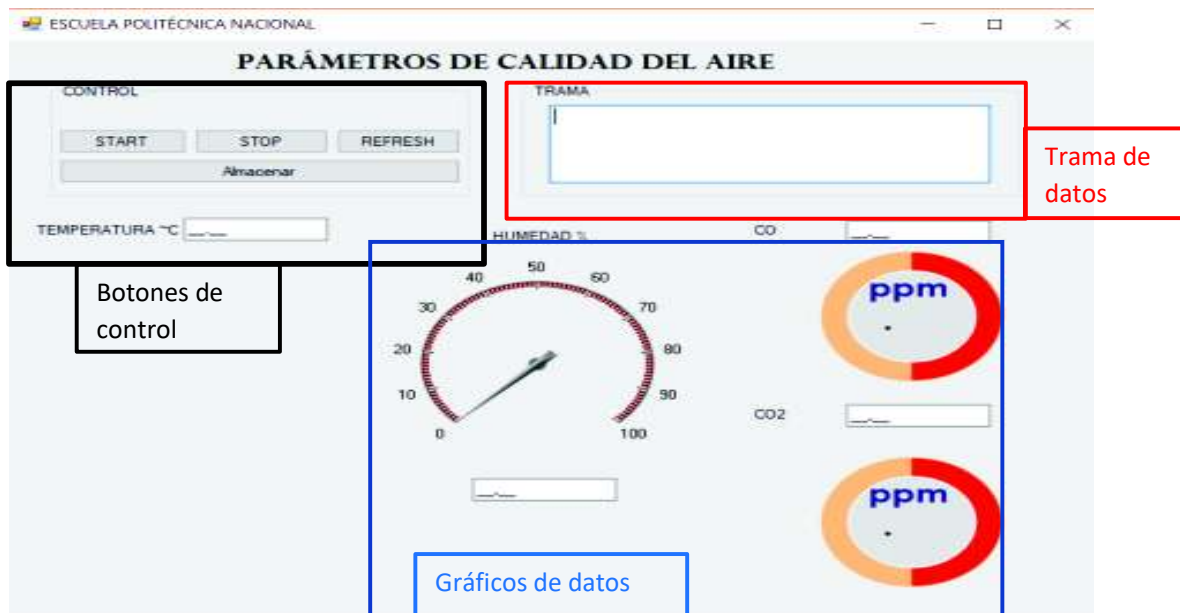


Figura 3.37 Interfaz gráfica de parámetros de calidad del aire.

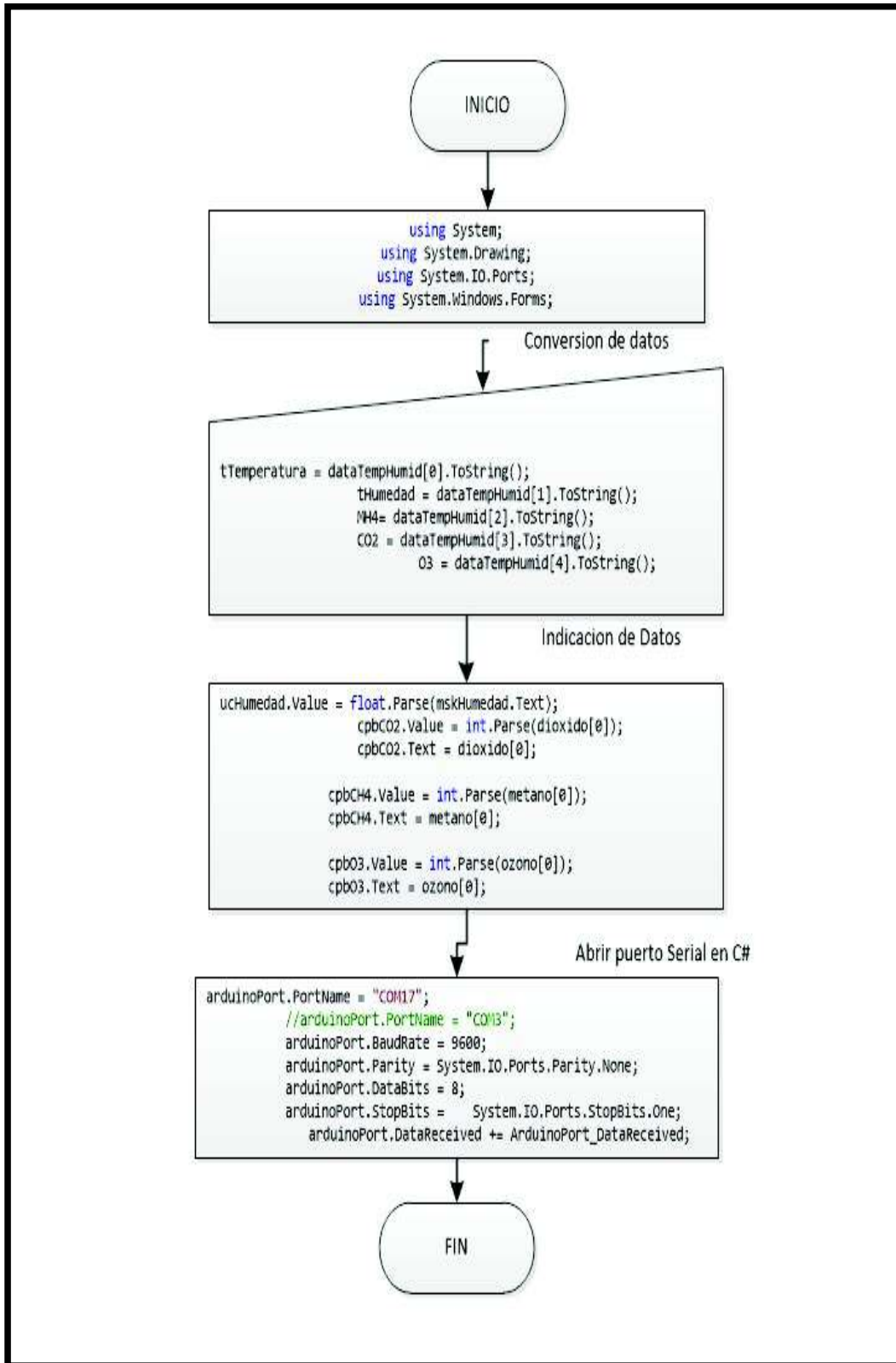


Figura 3.38 Algoritmo de Interfaz gráfica

Para la programación de esta interfaz se implementó el algoritmo de la figura 3.38, para la parte de control se realizó una interfaz serial, se abrió un puerto serial para captar los datos, esta interfaz es la que tiene conectado el coordinador al ordenador. Todos los datos son recibidos y separados por comas y se realizó una transformación de datos para mostrar en los indicadores gráficos.

Finalmente, el *robot* móvil equipado con la placa principal que contiene a los sensores, se comunica con un ordenador mediante módulos de radiofrecuencia *XBee Pro*, para el envío de datos hasta 500 metros de distancia, esta aplicación captura los datos de una trama y los separa en gráficos amigables para el usuario, por otra parte, el operador también tiene el control remoto FS-i6 con el cual hace las maniobras necesarias para mover al *robot* en distintos puntos y realizar las medidas necesarias en alguna zona de interés, estos datos son almacenados en un bloc de notas como trama de datos, todo esto en tiempo real, se puede ver en la figura 3.39.



Figura 3.39 Medición de parámetros de calidad del aire

3.8. Pruebas de Funcionamiento

En esta fase de pruebas y resultados, se describe los pasos para llevar el prototipo a su funcionamiento en varias etapas tanto de *software* como *hardware*. La primera etapa de pruebas consiste en llevar el robot móvil a diferentes terrenos irregulares para probar la robustez. El primer terreno en el que se probó, fue en el interior de la unidad educativa.

En suelo liso el *robot* móvil presenta adherencia a la superficie, sin embargo, por sus cuatro ruedas disponibles y energía en cada una de estas le permite avanzar sin derrapar manteniendo su velocidad constante.

El *robot* móvil también está en la capacidad de subir rampas de 30 grados de inclinación; se realizó pruebas en este estilo de pendientes y no presento dificultad en superarlos.

Después, se le lleva el robot a un terreno de césped, en este punto se pueden observar una mejor adherencia a la superficie, presenta menos velocidad por lo tanto un poco más lento, pero no presentó problemas en pasarlo ver figura 3.40.



Figura 3.40 Prueba en césped

Otra prueba que se realizó, es en terreno de cemento con la presencia de escaleras, la altura máxima que pasa el prototipo es de 8cm, se realizó pruebas con escaleras de más de esa altura y por no disponer de un buen centro de gravedad terminaba volteado, ver figura 3.41.

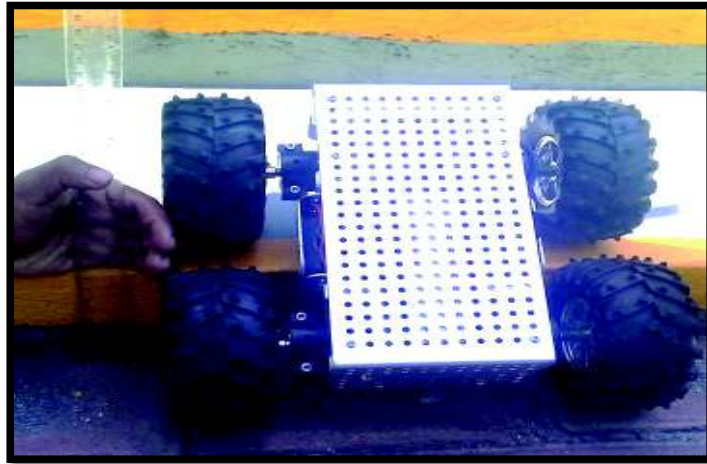


Figura 3.41 Pruebas en escaleras

En cuanto a la duración de energía se realizó pruebas con el *robot* conectado hasta que se descargue la batería llegando a una autonomía aproximada de 2 horas, para verificar el estado de la batería se coloca un probador de baterías figura 3.42 el cual indica el porcentaje de carga.

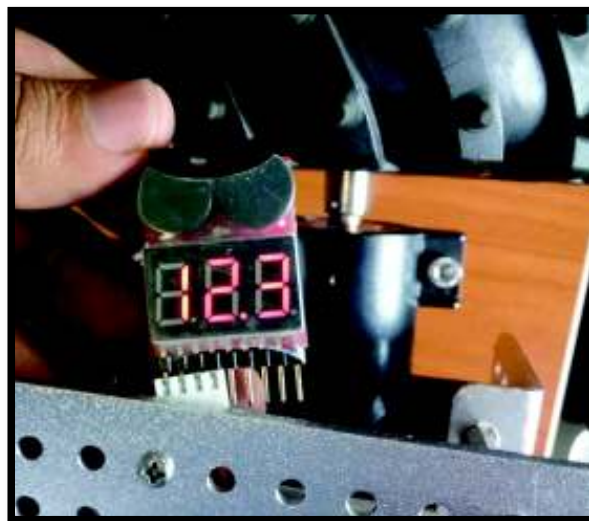


Figura 3.42 Probador de batería *LiPo*

3.9. Pruebas de comunicación

Para realizar las pruebas de *software* primero se estableció unas pruebas sencillas de comunicación entre los módulos *XBee Pro*, para esto se usó del *software Realterm* que es un visualizador serial como se puede observar en la figura 3.43.

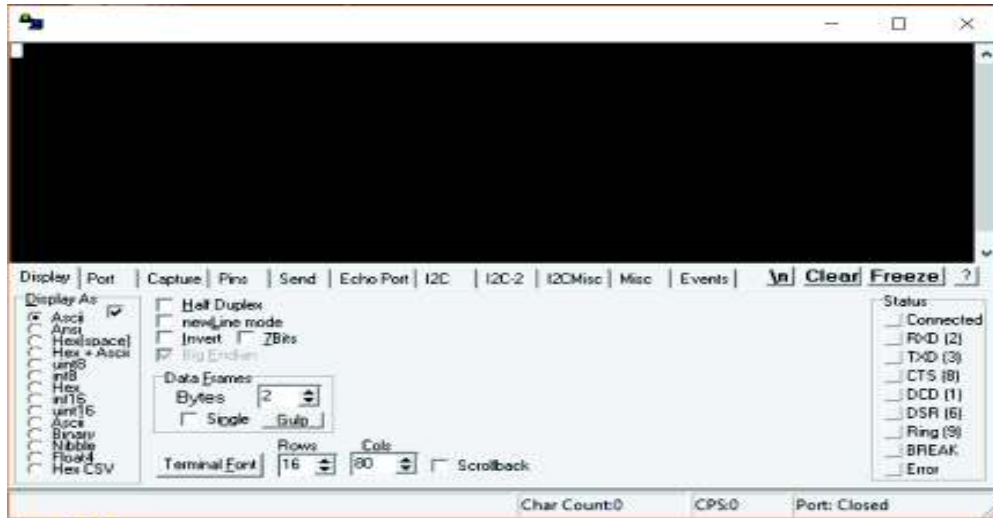


Figura 3.43 Software Realterm

En este *software* ver figura 3.44, se dispone de una pantalla para visualizar los datos transmitidos o recibidos, antes de iniciar la comunicación se debió seleccionar el puerto *com* al que estén conectado el *XBee Pro*, y se configura la velocidad de transmisión de datos.



Figura 3.44 Prueba de envío y recepción de datos

La trama de datos está conformada por varios valores lo cuales contiene información de los sensores este envío es realizado por los *XBee Pro* la trama se configuró en la programación del micro-controlador.

TRAMA DE DATOS: 20.00,16.00,0.21,2216,3300.29

La serie de números separada por comas representa:

1. Temperatura
2. Humedad
3. Metano
4. Dióxido de Carbono
5. Ozono

Por lo tanto, en la trama de datos se indica los datos y representara lo siguiente:

1. Temperatura: 20 grados centígrados
2. Humedad: 16% en el ambiente
3. Metano: 0.21 ppm
4. Dióxido de Carbono: 2216 ppm
5. Ozono: 3300.29 ppm

En esta captura de información los valores son altos debido a que no se dispuso del tiempo necesario para una calibración de los sensores. La última prueba realizada fue en distancia, el módulo alcanzó una transmisión de datos de 200 metros en interiores y 500 metros en línea de vista.

3.10. Pruebas de Sensores

Las pruebas se realizaron basándose en información de *datasheet* de los sensores utilizados que indican el tiempo de calentamiento necesario para que den una lectura, los mismos fueron para: metano se necesitó un tiempo de calentamiento de 24 horas, ozono un tiempo de 48 horas, dióxido de carbono, temperatura y humedad no necesitó tiempo de calentamiento también se obtuvo información medios locales como fueron secretaria de ambiente del distrito metropolitano de Quito, se expuso a la caja negra contenedora de la placa principal con los sensores de parámetros de calidad del aire a diferentes tipos de escenarios. El primer escenario fue una exposición directa al ambiental, el segundo escenario a humo producido por fuego de hojas y finalmente a contaminantes producido por automóviles.

- a) Para la primera prueba se colocó el robot móvil con la caja contenedora de la placa principal y sensores sujeta en la parte superior del robot, y se realizó mediciones en campo abierto ver figura 3.45.



Figura 3.45 Robot móvil pruebas en campo abierto

- b) En el segundo escenario se expuso al *robot* móvil a la presencia de humo producido por la quema de hojas de árboles y papel, para verificar en el *software* el incremento de los parámetros de calidad del aire
- c) Finalmente el tercer escenario planteado fue la exposición del *robot* móvil en la parte trasera de un vehículo para medir la contaminación cuando se produzcan gases en el escape, ver figura 3.46.



Figura 3.46 *Robot* móvil expuesto a gases de vehículo

En la figura 3.47 se muestra un historial de las mediciones realizadas en las pruebas, en el eje vertical se tiene el valor de concentración de gas, en porcentaje y la temperatura todos los parámetros versus el número de muestras en este caso 200.

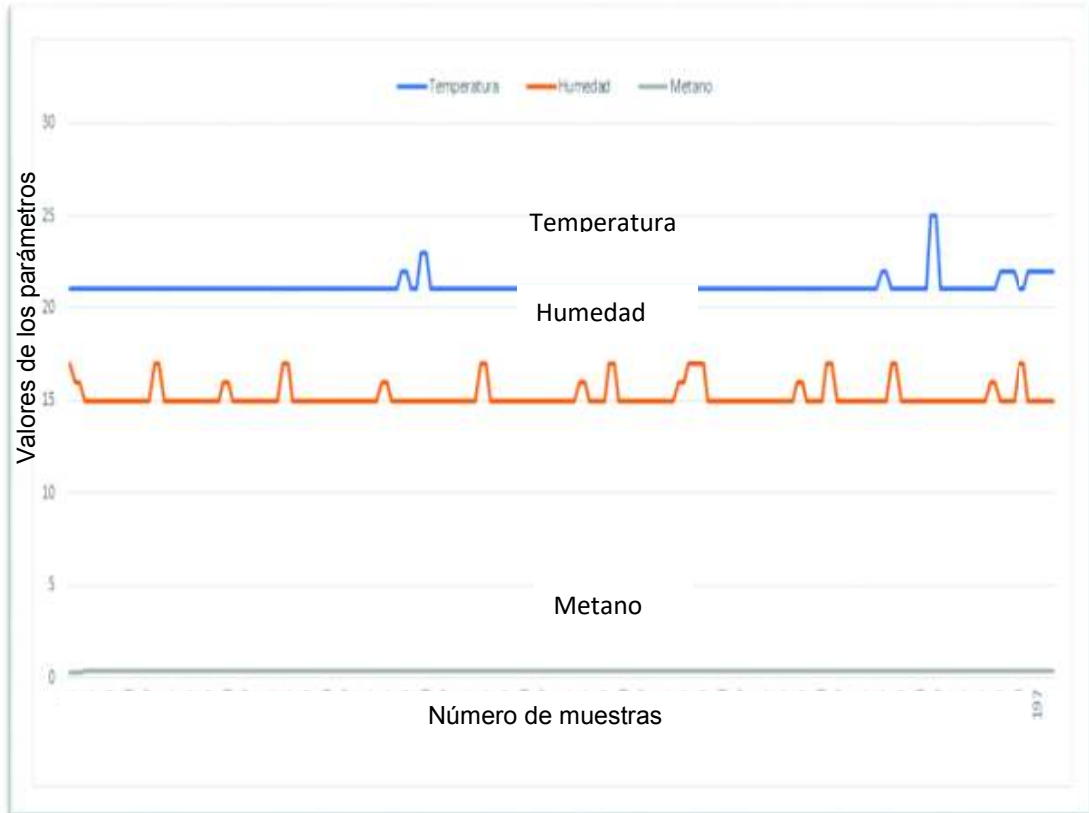


Figura 3.47 Resultado de muestra de sensores temperatura, humedad

En la figura 3.47 se tomó una muestra de 200 mediciones aproximadamente, la temperatura estuvo en grados centígrados con un promedio de 22.5°C, la humedad en porcentaje con una media de 16% en el ambiente y el Metano con un promedio de 0.17% de ppm (partes por millón).

La figura 3.48 indicó las muestras obtenidas en ppm de CO₂ comenzando con 12000 ppm este valor no se toma en cuenta porque es mientras se inicializa y se calibra el sensor, posterior se estabilizó en 400ppm que es el valor mínimo que puede leer del ambiente. En las pruebas del O₃ se mantuvo una media de 400 ppm que es un valor estable, ya que no se pudo generar O₃ para poder comparar.



Figura 3.48 Resultados de muestras obtenidas

3.11. Resultados

El *robot* móvil resultó ser de gran utilidad para todos los terrenos presentados, se puede evidenciar que atraviesa terrenos irregulares de césped, tierra, cemento. La suspensión de cada llantas son de gran ayuda al momento de pasar sobre objetos pequeños, con las demás pruebas se puede evidenciar que puede subir escaleras hasta de 8cm de altura, la red es robusta y de gran alcance y no se presentó pérdida de información.

La interfaz gráfica debido a los gráficos y escalas de los valores presentó latencia, para recalcar todos estos beneficios que se encontraron con la implementación, se establece en la tabla 3.4 la cual presenta un resumen de las características técnicas del robot implementado.

También, se realiza la tabla 3.5 en esta se recalcan las ventajas y desventajas para que futuras implementaciones tengan en cuenta al momento de usar estos elementos utilizados.

Características técnicas del robot móvil para medir parámetros de calidad del aire:

Tabla 3.4 Características Robot Móvil

Sensores	O3, CO2, CH4, Temperatura, Humedad
Alimentación	Batería <i>LiPo</i> : 5ª/12.4v <i>Power Bank</i> : 5ª/5v
Autonomía	2 horas control remoto/ 1 hora transmisión de datos
Software	Interfaz gráfica <i>Visual Studio</i> lenguaje C#
Comunicación Datos	Módulos <i>XBee</i> pro S2B 2.4GHz alcance interiores 200 metros, exteriores 500 metros
Comunicación Movimientos	Control FS-i6 con receptor FS-IA6B 2.4GHz alcance exteriores 500 metros
Carga Útil	3kg.
Velocidad	1m/s
Terrenos	Césped, cemento, tierra, baldosa.

Con los resultados de las pruebas se puede presentar las ventajas y desventajas que tiene el robot móvil para medir parámetros de calidad del aire.

Tabla 3.5 Ventajas y desventajas de la Implementación

	Ventajas	Desventajas
Sensor DTH11	Bajo costo, pequeño salida digital, el módulo incluye un sensor de humedad.	Para su implementación necesita un protocolo adicional.
Sensor CO2	Salida analógica en ppm, tamaño pequeño, de fácil implementación, comunicación con cualquier micro-controlador.	No puede realizar mediciones menores a 400ppm, alto consumo de energía 500mA.
Baterías <i>LiPo</i>	Adecuadas para el uso donde existen picos de corriente como los motores.	Constante monitoreo para verificar que no se descarguen, por debajo de su límite.
Robot Móvil	Adecuado para cargas útiles pequeñas como las de este proyecto.	Poca estabilidad, pierde centro de gravedad y se voltea.
Interfaz Grafica	Programación sencilla, facilidad de implementar modo gráfico.	Costo de suscripción, se trabajó con la versión estudiantil
<i>XBee Pro</i>	Pequeños, bajo consumo de energía.	Ninguna
Sensores CH4, O3	Bajo costo, pequeños, consume de energía bajo.	No se dispone de los gases para hacer una comparativa de valores adquiridos.

Los límites que se tomaron en cuenta para evaluar la calibración de los sensores estuvieron basados en los contaminantes presentes en el aire y fueron realizados por analizadores automáticos de las estaciones remotas de la red de monitoreo y se convirtieron a los

valores del monitoreo IQCA (Índice Quiteño de la Calidad del Aire) esto se observa en la tabla 3.6.

El manejo Ambiental está asignado al Ministerio del Ambiente en Coordinación con la Alcaldía de cada ciudad y emiten informes anuales sobre el índice de la calidad del aire, en este caso Quito, en el cual se enfocamos para obtener los resultados.

En las tablas 3.7 y 3.8 se presentan las categorías del IQCA y sus valores límites, para cada contaminante junto con el código de colores a ser utilizado. La misma incluye el significado para cada categoría y facilita una rápida asimilación del mensaje que se pretende comunicar.

En la tabla 3.9 el término individuos sensibles que se utilizó para detallar a las personas y la afectación debido a los contaminantes.

Tabla 3.6 Expresiones matemáticas para el cálculo del IQCA ^[5]

Contaminante	Expresiones matemáticas para cada rango de concentración			
CO, concentración máxima de promedio de 8 horas, mg/m ³	0 < Ci ≤ 10	10 < Ci ≤ 15	15 < Ci ≤ 30	30 < Ci
	IQCA = 10Ci	IQCA=20Ci- 100.00	IQCA=6.67Ci+100 .00	600=10Ci
O₃, concentración máxima de promedios de 8 horas, µg/m ³	0 < Ci ≤ 100	100 < Ci ≤ 200	200 < Ci ≤ 600	600 < Ci
	IQCA=Ci	IQCA=Ci	IQCA=0.5Ci+100. 00	IQCA=0.5Ci+100. 00
NO₂, Concentración máxima en 1 hora, µg/m ³	0 < Ci ≤ 200	200 < Ci ≤ 1000	1000 < Ci ≤ 3000	3000 < Ci
	IQCA=0.50 Ci	IQCA=0.125 Ci+75.00	IQCA=0.1Ci+100	IQCA=0.1Ci+100
SO₂, promedio en 24 horas, µg/m ³	0 < Ci ≤ 62.5	62.5 < Ci ≤ 125	125 < Ci ≤ 200	200 < Ci
	IQCA=0.8C i	IQCA=1.333 Ci-66.667	IQCA=0.125Ci+17 5.00	IQCA=0.125Ci+17 5.00
PM_{2.5}, promedio en 24 horas, µg/m ³	0 < Ci ≤ 50	50 < Ci ≤ 250	250 < Ci	
	IQCA=2.00 Ci	IQCA=Ci+50	IQCA=Ci+50.00	
PM₁₀, promedio en 24 horas, µg/m ³	0 < Ci ≤ 100	100 < Ci ≤ 250	250 < Ci ≤ 400	400 < Ci
	IQCA=Ci	IQCA=0.6667 Ci+33.333	IQCA=0.6667Ci+3 3.33	IQCA=Ci-100

Ci: Concentración de un determinado contaminante.

Tabla 3.7 Límites numéricos de cada categoría del IQCA ($\mu\text{g}/\text{m}^3$)^[5]

Rango	categoría	CO ³	O ₃ ^b	NO ₂ ^c	SO ₂ ^d	PM _{2.5} ^e	PM10 ^f
0-50	Nivel deseable u óptimo	0-5000	0-50	0-100	0-62.5	0-25	0-50
51-100	Nivel aceptable o bueno	5001-10000	51-100	101-200	63.5-125	26-50	51-100
101-200	Nivel de precaución	10001-15000	101-200	201-1000	126-200	51-150	101-250
201-300	Nivel de alerta	15001-30000	201-400	1001-2000	201-1000	151-250	251-400
301-400	Nivel de alarma	30001-40000	401-600	2001-3000	1001-1800	251-350	401-500
401-500	Nivel de emergencia	>40000	>600	>3000	>1800	>350	>500

Tabla 3.8 Rangos, significados y colores de las categorías del IQCA^[5]

Rangos	Condición desde el punto de vista de la salud	Color de identificación
0-50	Óptima.	Blanco
50-100	Buena.	Verde
100-200	No saludable para individuos extremadamente sensibles (enfermos crónicos y convalecientes).	Gris
200-300	No saludable para individuos sensibles (enfermos).	Amarillo
300-400	No saludable para la mayoría de la población y peligrosa para individuos sensibles.	Naranja
400-500	Peligrosa para toda la población.	Rojo

Tabla 3.9 Identificación de individuos sensibles por tipo de contaminante del aire^[5]

Contaminante	Individuos sensibles
Ozono	Niños que pasan tiempo en exteriores, adultos que realizan actividad física significativa en exteriores e individuos con enfermedades respiratorias como el asma.
Material particulado	Personas que presentan enfermedades de los pulmones o el corazón, tales como asma, obstrucción pulmonar crónica, congestiones cardíacas o similares.
Monóxido de carbono	Personas con enfermedades cardiovasculares, tales como angina o aquellas con afectaciones que comprometen al sistema cardiovascular y respiratorio, mujeres embarazadas, los bebés en gestación y recién nacidos.
Dióxido de azufre	Niños, adultos con asma u otras enfermedades respiratorias crónicas y personas que realizan actividades físicas en exteriores.
Dióxido de nitrógeno	Niños y adultos con enfermedades respiratorias como el asma.

3.12. Costos de Implementación

Para la implementación de este proyecto se determina los siguientes rubros, en los gastos primarios se detallan todos los dispositivos electrónicos y materiales usados para la realización de este proyecto.

Tabla 3.10 Costos primarios

Elementos	Cantidad	C. Unidad	C. Total
<i>Robot Explorador Móvil</i>	1	\$174,50	\$174,50
<i>TRex</i>	1	\$99,50	\$99,50
Control Remoto	1	\$59,79	\$59,79
Arduino MiniPro	3	\$18,89	\$18,89
<i>XBee Pro S2B</i>	1	\$77,81	\$77,81
<i>XBee Explorer</i>	1	\$25,84	\$25,84
Sensor de Ozono MQ131	1	\$27,00	\$27,00
Sensor de Metano MQ4	1	\$20,00	\$20,00
Sensor CO2	1	\$50,00	\$50,00
Sensor de Humedad y Temperatura DTH11	1	\$10,00	\$10,00
Baterías Lipo	1	\$35,00	\$35,00
Materiales Electrónicos	Varios	\$50,00	\$50,00
TOTAL			\$ 648,33

Adicional a los costos primarios se añade el costo de la programación del sistema que fue una semana, con un total de 40 horas de programación a un costo de 8 dólares la hora.

Costo total del Sistema de Parámetros de Contaminación Ambiental:

Tabla 3.11 Costo total

Costos Primarios	\$ 648,33
Costo Programación	\$ 320,00
TOTAL	\$ 968,33

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Se concluye con este prototipo, inalámbrico que permite la transmisión de datos a 500 metros de distancia; se incorporó sensores con parámetros de contaminación ambiental como ozono, gas metano, dióxido de carbono, temperatura y humedad amigable con el ambiente ya que funciona con un sistema de batería recargable, este tipo de sistema actualmente no se encuentra en el mercado.
- En la interfaz, la información del nodo fue recibida en tiempo real y sin pérdidas de datos, se observó mediante números y barras, la afectación en diferentes zonas de la unidad educativa.
- Se dispone de un respuesta para monitorear la presencia de gases incoloros, humedad y temperatura; los datos son obtenidos en la PC de monitoreo de manera inmediata; por su arquitectura modular y tamaño ergonómico facilita el uso portable para monitorear estos parámetros en cualquier ambiente de la unidad educativa.
- En la etapa de pruebas con la ayuda del personal docente, administrativo y estudiantes voluntarios, se pudo concluir que después de estar expuestos en medios cerrados como aulas y laboratorios con poca ventilación, los niveles de concentración y rendimiento estudiantil bajaron, debido a la presencia de gases incoloros y a un incremento de temperatura.
- El sensor DTH11 de humedad y temperatura tiene una resolución de +/- 1 grado centígrado, dispone de 3 pines, voltaje, tierra y una salida digital de 12 bits, se puede interpretar los datos correspondientes a la humedad relativa y la temperatura ambiente.
- El sensor CO2 no permite una medición más abajo de 400ppm por lo tanto para aplicaciones en ambientes abiertos no es recomendable utilizarlo, debido a que la concentración no supera el valor mencionado, su consumo energético es de aproximadamente 500mA, pero su conexión es mediante 3 pines que son voltaje, tierra y una salida analógica.

- Los sensores de Ozono y Metano disponen además de una salida analógica para adquirir los datos de los gases en el ambiente, sus otros pines son digitales los cuales alertan de presencia del gas; para configurar esa alerta basta con modificar el potenciómetro que disponen en las placas.
- EL módulo *Xbee Pro S2B* permite un protocolo de comunicación serial y que lo transforma en un protocolo inalámbrico y por el mismo se puede transmitir datos, la línea de vista según el manual son de 500m pero en las pruebas realizadas se pudo concluir que a una distancia 200m tenemos una viabilidad del 100% y después de esa distancia comienza a disminuir debido a que la señal tiene que atravesar distintos medios como por ejemplo paredes de concreto y árboles.
- Se concluye que el chasis pololu 4WD dispone de una estructura ligera y resistente que puedo acceder fácilmente a terrenos irregulares, probado en césped, asfalto, terrenos lisos y gradas de hasta 8cm de altura, pudo soportar hasta 3kg de peso y si se requiere mejorar el modelo se debe tomar en cuenta este parámetro debido a que en las pruebas realizadas no supero las pendientes de 30° de inclinación con un peso mayor al mencionado.
- Los datos obtenidos de las mediciones se almaceno en formato de texto plano, para que los operadores tengan estadísticas disponibles de esa información y poder tomar los correctivos necesarios si estuvieran fuera de los rangos normales
- Con el presente trabajo de titulación se concluyó que después de haber tomado varias muestras de los parámetros de contaminación ambiental en la Unidad Educativa “San Martin”, en diferentes días y horarios, este se encuentra dentro de los parámetros normales en los que un ser humano puede desempeñar sin sufrir ningún daño al organismo.

4.2. Recomendaciones

- Cuando se trabaja con este tipo de investigación se recomienda trabajar modularmente ya que se puede expandir la aplicación añadiendo más sensores al prototipo y haciendo un sistema más grande obteniendo mejores resultados.
- El sensor MQ4 tienen un tiempo de calentamiento prolongado para su uso, se recomienda tenerlo encendido 24 horas antes para que las lecturas sean exactas y confiables.
- El sensor MQ131 tienen un tiempo de calentamiento prolongado, se recomienda tenerlo encendido 48 horas antes para su uso.
- Para mejorar el modelo se recomienda agregar sensores que permitan tener datos como nivel de estrés, sudoración. Y que permitan saber la afectación a la salud de las personas.
- Se recomienda al momento de ejecutar la aplicación verificar que el puerto *COM* del coordinador sea el mismo en el código fuente, porque si no son los mismos la aplicación no funcionara.
- Para mejorar el modelo se recomienda utilizar sensores como radiación ultravioleta, monóxido de carbono y velocidad del viento, parámetros que permitirán complementar la información ambiental donde el robot realice el monitoreo.
- La interfaz proporciona la opción de guardar los datos en texto plano y opera de manera local en la PC, sin embargo, se recomienda migrar a una interfaz web con una base de datos para que la información esté disponible para todo público de las diferentes aéreas en donde se realice las mediciones.
- La organización mundial de la salud y secretaria de ambiente de la ciudad de Quito, recomiendan tener presente los rangos permitidos, porque la presencia de estos contaminantes en corto o largo plazo pueden terminar ocasionando daños a la salud, tales como migrañas, dolores de cabeza, deterioro de la función pulmonar entre otros, siendo más vulnerables los niños.

5. REFERENCIAS BIBLIOGRÁFICAS.

- [1] Polulu, «Polulu Robotics & Electronics,» [En línea]. Available: <https://www.polulu.com/category/2/robot-kits>. [Último acceso: 1 09 2017].
- [2] F. F.-i. 1. 2. A. R. T. w. F.-i. Receiver, «Amazon,» [En línea]. Available: https://www.amazon.com/Flysky-FS-i6X-Transmitter-FS-iA6B-Receiver/dp/B0744DPPL8/ref=sr_1_19_sspa?ie=UTF8&qid=1525704551&sr=8-19-spons&keywords=rc%20tx%20radio%20and%20rx&psc=1. [Último acceso: 15 09 2017].
- [3] AOSONG, «Temperature and humidity module,» [En línea]. Available: <http://akizukidenshi.com/download/ds/aosong/DHT11.pdf>. [Último acceso: 25 10 2017].
- [4] Dfrobot, «Dfrobot Drive the Future,» [En línea]. Available: http://www.dfrobot.com/index.php?route=product/category&path=156_211. [Último acceso: 20 10 2017].
- [5] C. M. d. Q.-. Ecuador, Código Municipal para el Distrito Metropolitano de Quito. Odenanza Metropolitana N° 404 del 23 de abril de 2013.
- [6] Ozone, «MQ-131 Semi Conductor Sensor,» [En línea]. Available: <http://www.sensorsportal.com/DOWNLOADS/MQ131.pdf>. [Último acceso: 20 10 2017].
- [7] H. Electronics, «MQ-4 data Sheet,» [En línea]. Available: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-4.pdf>. [Último acceso: 20 10 2017].
- [8] Zigbee, «XBee & XBee - Pro,» 2017. [En línea]. Available: https://cdn.sparkfun.com/datasheets/Wireless/Zigbee/ds_xbeezbmodules.pdf. [Último acceso: diciembre 2017].
- [9] F. J. Ceballos, Enciclopedia de Microsoft Visual C#, Madrid: RA-MA, 2013.
- [10] Asamblea Nacional, «Sección segunda del medio ambiente de la Carta Magna del Ecuador,» [En línea]. Available: <http://pdba.georgetown.edu/Parties/Ecuador/Leyes/constitucion.pdf>. [Último acceso: 2017].
- [11] J. Aguirre, «Redes inalámbricas,» 2013. [En línea]. Available: <http://www.monografias.com/trabajos/redesinalam/redesinalam.shtml>. [Último acceso: 21 3 2018].
- [12] A. C. o. O. a. E. m. 2003, «Noise-Induced Hearing Loss. Journal of Occupation and Enviromental Medicine,» 2003. [En línea]. Available: <http://www.acoem.org/>. [Último acceso: 13 Septiembre 2017].

- [13] E. Callaway, «Wireless Sensor Networks: Architecture and Protocols,» 2004. [En línea]. Available: http://phd.cs.pub.ro/sites/default/files/TOC_WSN.pdf. [Último acceso: 19 2 2018].
- [14] Digi, «X-Ctu,» [En línea]. Available: <http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>. [Último acceso: 20 agosto 2018].
- [15] V. Escutar, Diseño de un prototipo de red de sensores inalámbricos para monitorear la contaminación ambiental acústica/ sonora en la ciudad de quito sector la Coruña y san Ignacio. Quito: Universidad Metropolitana AEDE QUITO., 2016.
- [16] J. Guaña y V. Escutar , Aplicación de las Redes de Sensores Inalámbricos. Quito: Universidad Metropolitana., 2016.
- [17] Microsoft, «Programacion po Capas,» [En línea]. Available: <https://msdn.microsoft.com/es-es/library/ms978340.aspx>. [Último acceso: 25 enero 2018].
- [18] L. Navas, «Analizando el ruido ambiental en el entorno de la escuela,» 2012. [En línea]. Available: <http://www.eoi.es/blogs/lauranavas/2012/05/10/analizando-el-ruido-ambiental-en-el-entorno-de-la-escuela/>. [Último acceso: 13 junio 2018].
- [19] O. M. d. I. Salud, «"Escuchar sin Riesgos",» [En línea]. Available: <http://www.who.int/topics/deafness/safe-listening/es/>. [Último acceso: 22 noviembre 2017].
- [20] W. S. Networks, «Introduccion a las redes de sensores inalámbricas,» 2004. [En línea]. Available: <http://www.mfbarcell.es/es/conferencias/wsn.pdf>. [Último acceso: 20 agosto 2017].
- [21] Ublock, «Modules Data Sheet,» 2015. [En línea]. Available: http://botscience.net/store/index.php?route=product/product&product_id= 73. [Último acceso: 20 noviembre 2017].
- [22] A. García, «¿Qué es arduino y para que se utiliza?,» 2015. [En línea]. Available: <http://panamahitek.com/que-es-arduino-y-para-que-se-utiliza/>.

ANEXOS

Anexo A1: Código implementado para el funcionamiento en el proyecto en *Arduino*.

Anexo A2: Código implementado para el funcionamiento en el proyecto en *Visual Studio*.

Anexo A3: Módulos *Xbee Pro*.

Anexo A4: Arquitectura de módulos sensores.

Anexo A5: Manual de Usuario y mantenimiento.

Anexo A1: Código implementado para el funcionamiento en el proyecto en Arduino

```
//Añadimos la librería necesaria:
#include <DHT.h>

// Pin digital donde se conectamos el sensor.
#define DHTPIN 2
#define DHTTYPE DHT11

//Iniciamos el sensor DHT11.
DHT dht(DHTPIN, DHTTYPE);

//Configuración del setup.
void setup()
{
  Serial.begin(9600); // Inicializamos la comunicación serie.
  dht.begin();      // Inicializamos la comunicación del sensor.
  delay(1500);
  Serial.println("****Sensor DHT11****");
  Serial.println("Lectura de Temperatura, Humedad e Índice de calor ");
}

//Programa principal
void loop()
{
  //Pausa de 3 segundos entre lecturas.
  delay(3000);
  float h = dht.readHumidity(); // Lee la humedad relativa.
  float t = dht.readTemperature(); // Lee la temperatura en °C (por defecto).
  Serial.println();
  Serial.print("Temperatura: ");
  Serial.print(t);
  Serial.print(" °C / ");
  Serial.print("Humedad: ");
  Serial.print(h);
  Serial.println(" %");
}
```

Anexo A2: Código implementado para el funcionamiento en el proyecto en Visual

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
//using CapaEntidades;
//using CapaNegocio;

using System.Net;
using System.Net.Mail;
using System.Net.Mime;

namespace Sensores
{
    public partial class Form1 : Form
    {
        decimal sumaHumedad, sumaTemperatura;
        int barTop = 0;
        int i = 0;
        int bandera = 0;

        string tTemperatura = "1", tHumedad = "1", CO2="1",O3="1",MH4="1";
        string dataFromArduino = "";
        string[] dataTempHumid;
        decimal[] promedioTemp = new decimal[10];
        string hora;
        Bitmap img;
        Bitmap bar;
        DateTime fecha;

        public Form1()
        {
            InitializeComponent();
        }
        private void frmSensorTemperaturaHumedad_Load(object sender, EventArgs e)
        {
            // btnEnviarCorreo.Enabled = false;
            img = Properties.Resources.termometro;
            bar = Properties.Resources.bar;
            bar.MakeTransparent(Color.White);
        }
        private void btnStart_Click(object sender, EventArgs e)
        {
            setArduinoPort();
            arduinoPort.Write("B");
            btnStart.Enabled = false;
            btnStop.Enabled = true;
        }

        private void btnAlmacenar_Click(object sender, EventArgs e)
        {
            string filepath = @"C:\Users\tiago\Desktop\DT\TESIS RF\Sensores\";
            string filename = "DATOSAIRE.txt";
            System.IO.File.WriteAllText(filepath + filename, txtData.Text);
        }
    }
}
```

```

        MessageBox.Show("Datos almacenados Exitosamente", "Aviso",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    private void btnStop_Click(object sender, EventArgs e)
    {
        try
        {
            arduinoPort.Close();
            btnStart.Enabled = true;
            btnStop.Enabled = false;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    private void btnDiscart_Click(object sender, EventArgs e)
    {
        arduinoPort.Write("B");
    }

    private void ArduinoPort_DataReceived(object sender,
    SerialDataReceivedEventArgs e)
    {
        try
        {
            dataFromArduino = arduinoPort.ReadLine().ToString();

            if (dataFromArduino.Contains(","))
            {
                dataTempHumid = dataFromArduino.Split(',');

                tTemperatura = dataTempHumid[0].ToString();
                tHumedad = dataTempHumid[1].ToString();
                MH4= dataTempHumid[2].ToString();
                CO2 = dataTempHumid[3].ToString();
                O3 = dataTempHumid[4].ToString();

                this.Invoke(new EventHandler(showData_event));
            }
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    private void showData_event(object sender, EventArgs e)
    {
        fecha = DateTime.Now;
        hora = fecha.Hour + ":" + fecha.Minute + ":" + fecha.Second;
        txtData.AppendText(hora + "\t\t" + dataFromArduino + "\n");
        //distancia = int.Parse(data);

        bandera = 1;
    }
}

```

```

}
private void timer1_Tick(object sender, EventArgs e)
{
    try
    {
        // btnCapturar.Enabled = true;

        // txtHora.Text = hora;
        mskTemperatura.Text = tTemperatura;
        mskHumedad.Text = tHumedad;
        mskCH4.Text = MH4;
        mskCO2.Text = CO2;
        mskO3.Text = O3;

        if (bandera == 1)
        {
            sumaTemperatura = sumaTemperatura + decimal.Parse(tTemperatura);
            sumaHumedad = sumaHumedad + decimal.Parse(tHumedad);

            i++;
        }
        string[] dioxido = CO2.Split('.');
        string[] metano = MH4.Split('.');
        string[] ozono = O3.Split('.');
        dibujarTemperatura();

        ucHumedad.Value = float.Parse(mskHumedad.Text);
        cpbCO2.Value = int.Parse(dioxido[0]);
        cpbCO2.Text = dioxido[0];

        cpbCH4.Value = int.Parse(metano[0]);
        cpbCH4.Text = metano[0];

        cpbO3.Value = int.Parse(ozono[0]);
        cpbO3.Text = ozono[0];

    }
    catch (Exception)
    {
        MessageBox.Show("Obteniendo Parametros", "Aviso",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void dibujarTemperatura()
{
    if (mskTemperatura.Text == null || img == null | bar == null)
    {
        return;
    }

    try
    {
        Bitmap newImg = new Bitmap(img);
        Graphics gr = Graphics.FromImage(newImg);
        //Calcula el tope de la barra de temperatura
    }
}

```

```

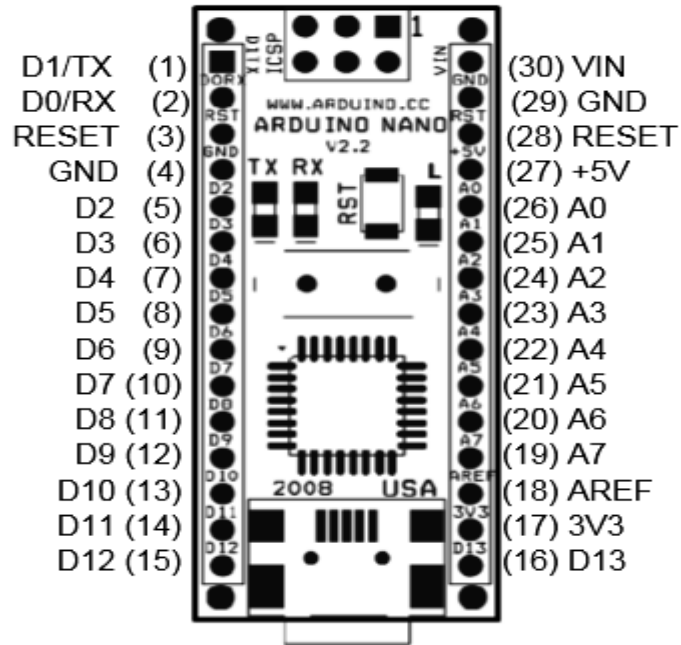
        //Condicion if else comprimida de tal forma que: t ? x : y - si test
        t es verdadero, entnces se evalua y se retorna x; caso contrario se evalua y se
        retorna y
        barTop = (decimal.Parse(mskTemperatura.Text) > 0 ? 9 : 7) +
Convert.ToInt32(((100 - decimal.Parse(mskTemperatura.Text)) / 10) * 20);
        //crea un nuevo bitmap para esta barra
        Bitmap bar2 = new Bitmap(12, 264 - barTop);
        Graphics gr2 = Graphics.FromImage(bar2);
        //dibuja la parte de la barra original en el nuevo bitmap
        gr2.DrawImage(bar, new Rectangle(Point.Empty, bar2.Size), new
Rectangle(0, Properties.Resources.bar.Height - bar2.Height, 12, bar2.Height),
GraphicsUnit.Pixel);
        //dibuja una nueva barra de imagen en la imagen posterior
(termometro)
        gr.DrawImage(bar2, new Point(50, barTop));
        pcbTermometro.Image = newImg;
    }
    catch (Exception)
    {
        MessageBox.Show("Obteniendo Parametros", "Aviso",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
private void setArduinoPort()
{
    arduinoPort.PortName = "COM17";
    //arduinoPort.PortName = "COM3";
    arduinoPort.BaudRate = 9600;
    arduinoPort.Parity = System.IO.Ports.Parity.None;
    arduinoPort.DataBits = 8;
    arduinoPort.StopBits = System.IO.Ports.StopBits.One;
    arduinoPort.DataReceived += ArduinoPort_DataReceived;
    try
    {
        arduinoPort.Open();
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}
}
}

```

Anexo A3: Módulos Xbee Pro

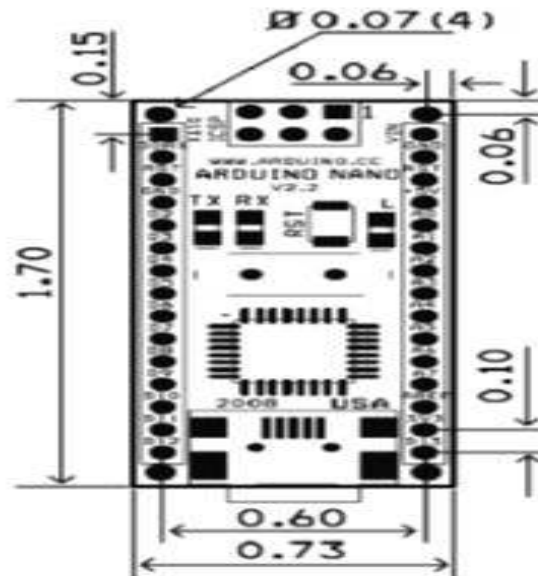
Especificaciones Técnicas

Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Arduino Nano Mechanical Drawing



ALL DIMENSIONS ARE IN INCHES

Anexo A4: Arquitectura de módulos sensores

Especificaciones Técnicas

TECHNICAL DATA

MQ-4 GAS SENSOR

FEATURES

- * High sensitivity to CH₄, Natural gas.
- * Small sensitivity to alcohol, smoke.
- * Fast response . * Stable and long life * Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of CH₄, Natural gas. LNG, avoid the noise of alcohol and cooking fumes and cigarette smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
P _L	Load resistance	20K Ω	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 750mw	

B. Environment condition

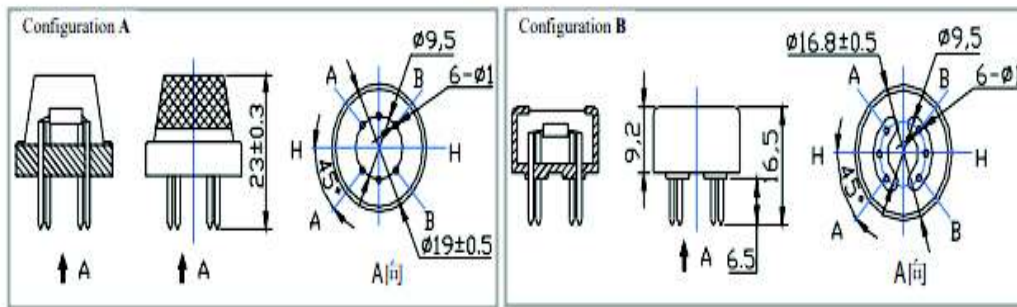
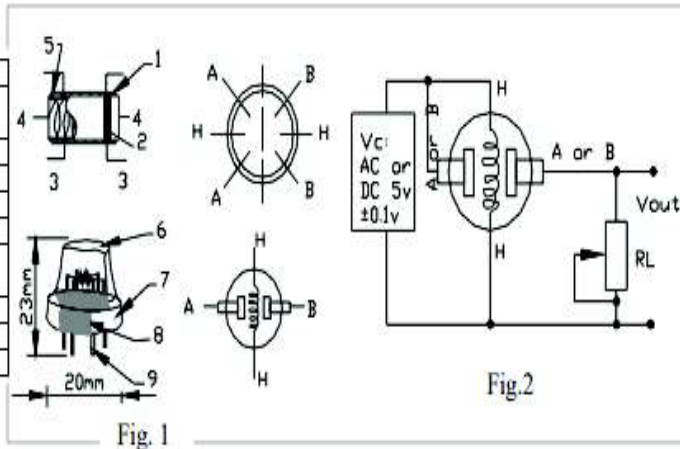
Symbol	Parameter name	Technical condition	Remarks
T _{ao}	Using Tem	-10℃-50℃	
T _{as}	Storage Tem	-20℃-70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R _s	Sensing Resistance	10K Ω - 60K Ω (1000ppm CH ₄)	Detecting concentration scope: 200-10000ppm CH ₄ , natural gas
α (1000ppm/ 5000ppm CH ₄)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20℃ ± 2℃ Humidity: 65%±5%	V _c :5V±0.1 V _H : 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

Parts	Materials
1 Gas sensing layer	SnO ₂
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceramic	Al ₂ O ₃
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni



Structure and configuration of MQ-4 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-4 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig. 2

E. Sensitivity characteristic curve

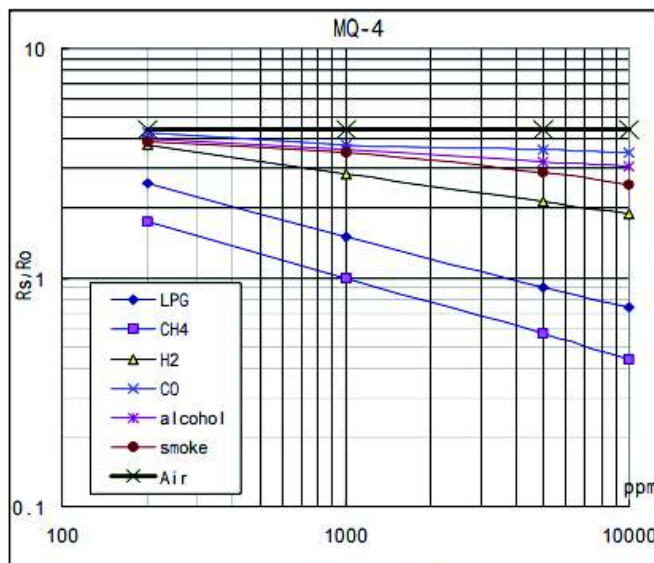


Fig.2 sensitivity characteristics of the MQ-4

Fig.3 is shows the typical sensitivity characteristics of the MQ-4 for several gases.

in their: Temp: 20°C,
Humidity: 65%,
O₂ concentration 21%
RL=20k Ω

Ro: sensor resistance at 1000ppm of CH₄ in the clean air.
Rs: sensor resistance at various concentrations of gases.

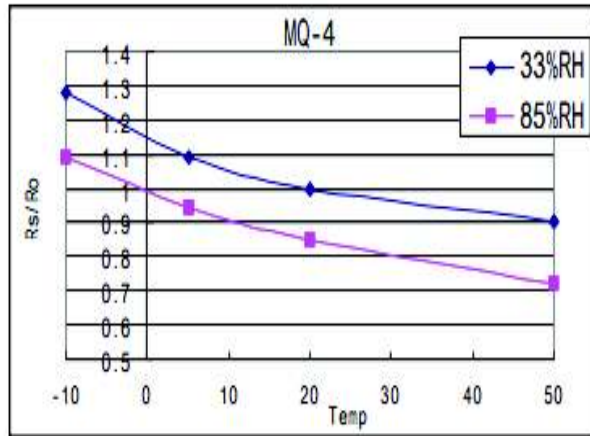


Fig.4 shows the typical dependence of the MQ-4 on temperature and humidity.
 R_o : sensor resistance at 1000ppm of CH_4 in air at 33%RH and 20 degree.
 R_s : sensor resistance at 1000ppm of CH_4 in air at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-4 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 5000ppm of CH_4 concentration in air and use value of Load resistance (R_L) about $20K\Omega$ ($10K\Omega$ to $47K\Omega$).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

MG811

CO2 Sensor

Features

- Good sensitivity and selectivity to CO2
- Low humidity and temperature dependency
- Long stability and reproducibility

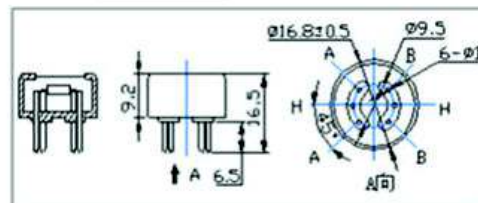
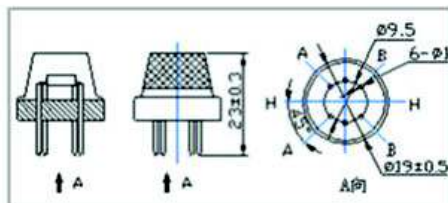
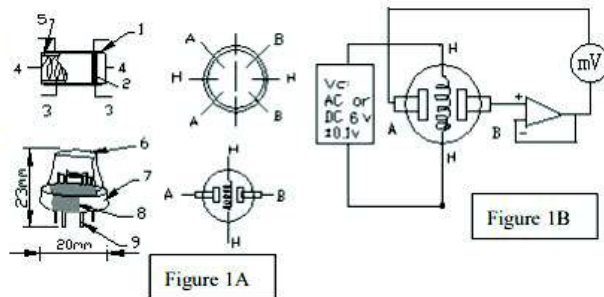
Application

- Air Quality Control
- Ferment Process Control
- Room Temperature CO2 concentration Detection



Structure and Testing Circuit

Sensor Structure and Testing Circuit as Figure. It composed by solid electrolyte layer (1), Gold electrodes (2), Platinum Lead (3), Heater (4), Porcelain Tube (5), 100m double-layer stainless net (6), Nickel and copper plated ring (7), Bakelite (8), Nickel and copper plated pin (9)



Working Principle

Sensor adopt solid electrolyte cell Principle , It is composed by the following solid cells :

Air , Au|NASICON|| carbonate|Au, air , CO₂

When the sensor exposed to CO₂ , the following electrodes reaction occurs :

Cathodic reaction : $2Li + CO_2 + 1/2O_2 + 2e^- = Li_2CO_3$

Anodic reaction : $2Na + 1/2O_2 + 2e^- = Na_2O$

Overall chemical reaction : $Li_2CO_3 + 2Na + Na_2O = 2Li + CO_2$

The Electromotive force (EMF) result from the above electrode reaction, accord with according to Nernst's equation :

$$EMF = E_c - (R \times T) / (2F) \ln (P(CO_2))$$

$P(CO_2)$ —CO₂ partial Pressure E_c —Constant Volume R —Gas Constant volume

T —Absolute Temperature (K) F —Faraday constant

From Figure 1B, Sensor Heating voltage supplied from other circuit , When its surface temperature is high enough , the sensor equals to a cell, its two sides would output voltage signal ,and its result accord with Nernst's equation. In sensor testing, the impedance of amplifier should be within 100—1000GΩ , Its testing current should be control below 1pA.

Specifications :

Symbol	Parameter Name	Technical	Remarks
V _H	Heating Voltage	6.0±0.1 V	AC or DC
R _H	Heating Resistor	30.0±5% Ω	Room Temperature
I _H	Heating Current	@200mA	
P _H	Heating Power	@1200mW	
Tao	Operating Temperature	-20—50	
Tas	Storage Temperature	-20—70	
V _{EMF}	Output	30—50mV	350—10000ppmCO ₂

Sensitivity :

Figure 2 Shows gas sensor sensitivity curve. :

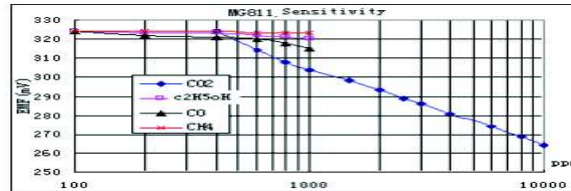
Conditions:

Tem : 28°C,

RH : 65%,

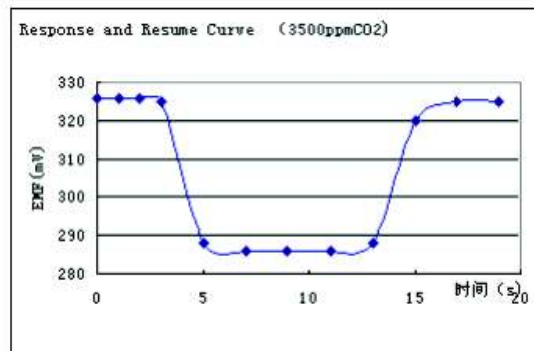
Oxygen : 21%

EMF: sensor EMF under different gas and concentration .

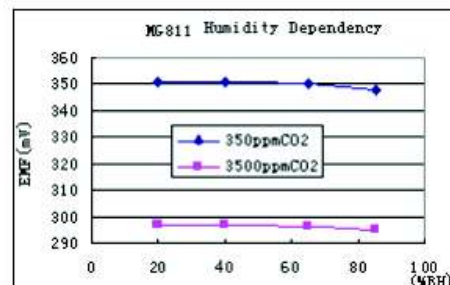
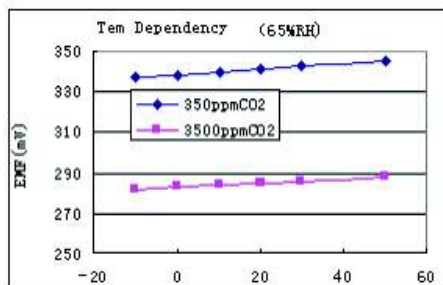


Response and Resume Characteristic :

Figure 3 shows Solid electrolyte sensor response and resume characteristics.



Temperature and Humidity Dependency :



MQ131 Semiconductor Sensor for Ozone

Sensitive material of MQ131 gas sensor is SnO₂ which with lower conductivity in clean air. When Ozone gas exists. The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ131 gas sensor has high sensitivity to Ozone, also sensitive to CL2, NO2, etc.

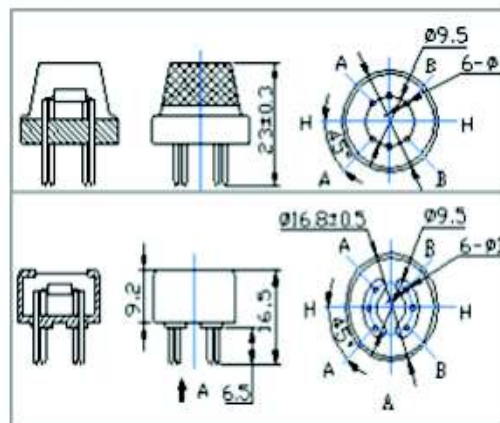
Character

- * Good sensitivity to Ozone in wide range
- * High sensitivity to Ozone
- * Long life and low cost
- * Simple drive circuit

Application

- * Domestic Ozone concentration overload Detector
- * Industrial Ozone concentration overload Detector
- * Portable Ozone concentration overload Detector

Configuration



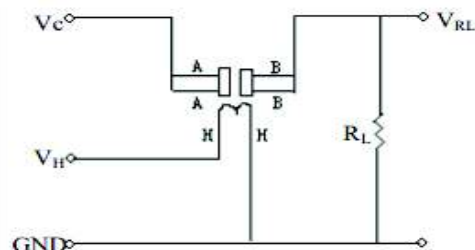
Technical Data

Model No.		MQ131	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite (Black Bakelite)	
Detection Gas		Ozone	
Concentration		10-1000ppm Ozone	
Circuit	Loop Voltage	V _c	≤24V DC
	Heater Voltage	V _H	5.0V±0.2V AC or DC
	Load Resistance	R _L	Adjustable
Character	Heater Resistance	R _H	31Ω±3Ω (Room Tem.)
	Heater consumption	P _H	≤900mW
	Sensing Resistance	R _s	50KΩ-500KΩ(in 50ppm O ₃)
	Sensitivity	S	R _s (in air)/R _s (in 50ppm O ₃)≥3
	Slope	α	(R _{10ppm} /R _{100ppm} O ₃)
Condition	Tem. Humidity	20℃±2℃; 65%±5%RH	
	Standard test circuit	V _c : 5.0V±0.1V; V _H : 5.0V±0.1V	
	Preheat time	Over 48 hours	

Power of Sensitivity body(P_s): $P_s = V_c^2 \times R_s / (R_s + R_L)^2$

Resistance of sensor(R_s): $R_s = (V_c / V_{RL} - 1) \times R_L$

Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage (V_H) and test voltage (V_C). V_H used to supply certified working temperature to the sensor, while V_C used to detect voltage (V_{RL}) on load resistance (R_L) whom is in series with sensor. The sensor has light polarity, V_c need DC power. V_c and V_H could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable R_L value is needed:

Sensitivity Characteristics

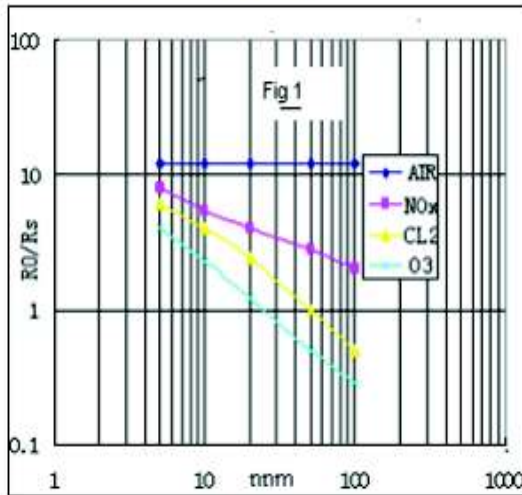


Fig.1 shows the typical sensitivity characteristics of the MQ131, ordinate means resistance ratio of the sensor (R_s/R_s), abscissa is concentration of gases. R_s means resistance in different gases, R_o means resistance of sensor in 50ppm CL_2 . All test are under standard test conditions.

Influence of Temperature/Humidity

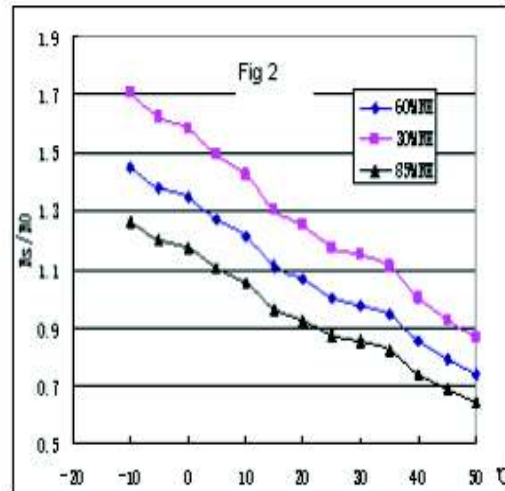
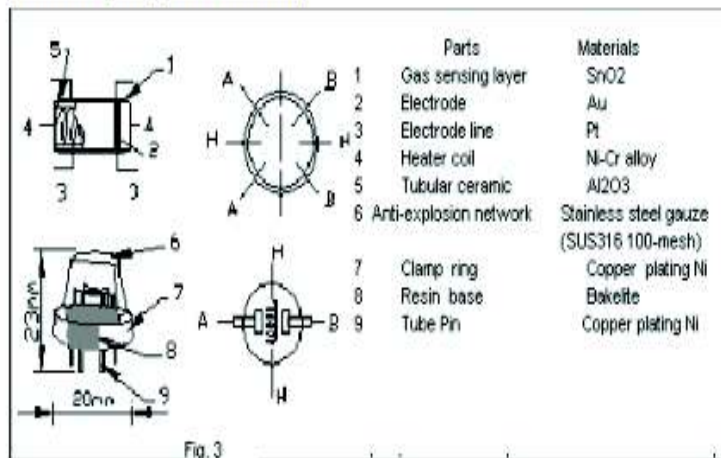


Fig.2 shows the typical temperature and humidity characteristics. Ordinate means resistance ratio of the sensor (R_s/R_o), R_s means resistance of sensor in 50ppm Ozone under different tem. and humidity. R_o means resistance of the sensor in environment of 50ppm Ozone, 20 $^{\circ}C$ /65%RH

Structure and configuration

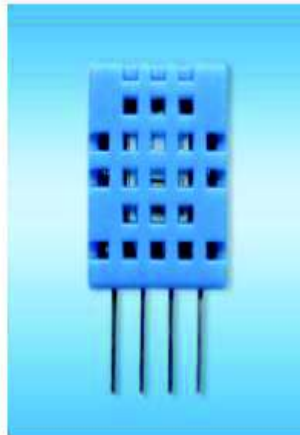


Structure and configuration of MQ131 gas sensor is shown as Fig. 3, sensor composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-4 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Temperature and humidity module DHT11 Product Manual

1、 Product Overview

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement devices, and connected with a high-performance 8-bit microcontroller.



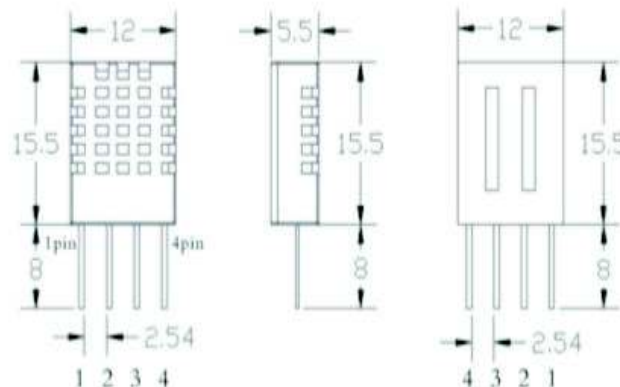
2、 Applications

HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.

3、 Features

Low cost, long-term stability, relative humidity and temperature measurement, excellent quality, fast response, strong anti-interference ability, long distance signal transmission, digital signal output, and precise calibration.

4、 Dimensions (unit: mm)



5、 Product parameters

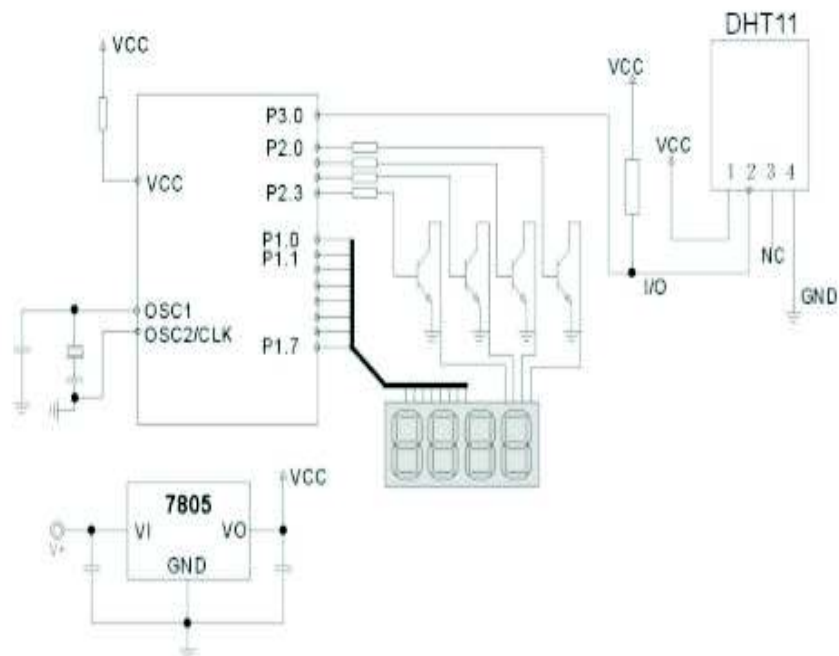
Relative humidity
Resolution: 16Bit
Repeatability: $\pm 1\%$ RH
Accuracy: At 25°C $\pm 5\%$ RH
Interchangeability: fully interchangeable
Response time: 1 / e (63%) of 25°C 6s
1m / s air 6s
Hysteresis: $< \pm 0.3\%$ RH
Long-term stability: $< \pm 0.5\%$ RH / yr in

Temperature
Resolution: 16Bit
Repeatability: $\pm 0.2^\circ\text{C}$
Range: At 25°C $\pm 2^\circ\text{C}$
Response time: 1 / e (63%) 10S

Electrical Characteristics
Power supply: DC 3.5 ~ 5.5V
Supply Current: measurement 0.3mA standby 60 μ A
Sampling period: more than 2 seconds

Pin Description
1, the VDD power supply 3.5 ~ 5.5V DC
2 DATA serial data, a single bus
3, NC, empty pin
4, GND ground, the negative power

6、 Typical circuit

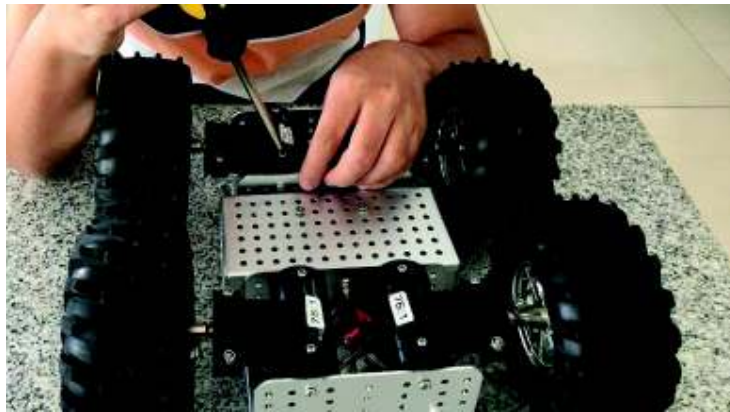


Anexo A5: Manual de Usuario

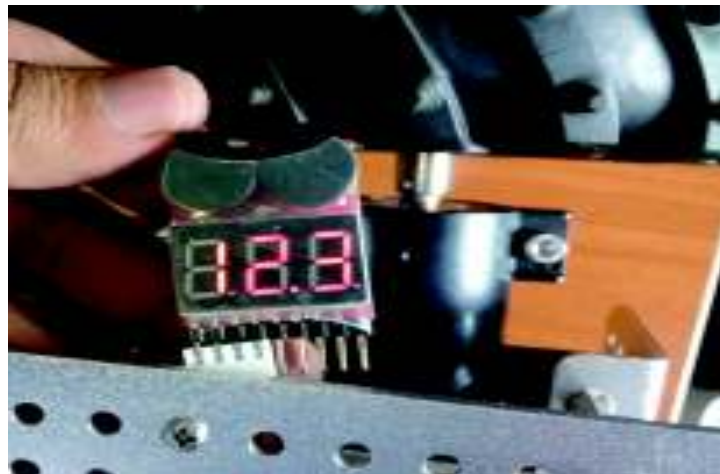
Manual de Usuario

En el presente manual se presenta paso a paso como poner en funcionamiento el robot móvil, para la lectura de parámetros ambientales en cualquier ambiente; se recomienda seguir todos los ítems descritos para no tener dificultad ni error para la puesta en marcha.

- a) Ajustar los tornillos de las ruedas y del chasis pololu 4WD.

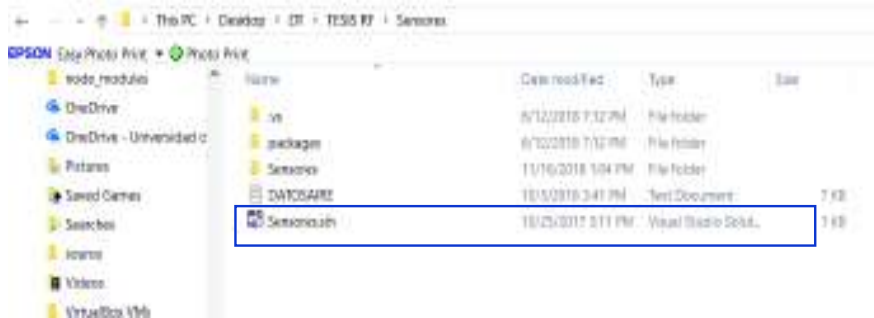


- b) Verificar por medio de un *tester* o voltímetro si las baterías están cargadas, caso contrario no encender el *robot* móvil.

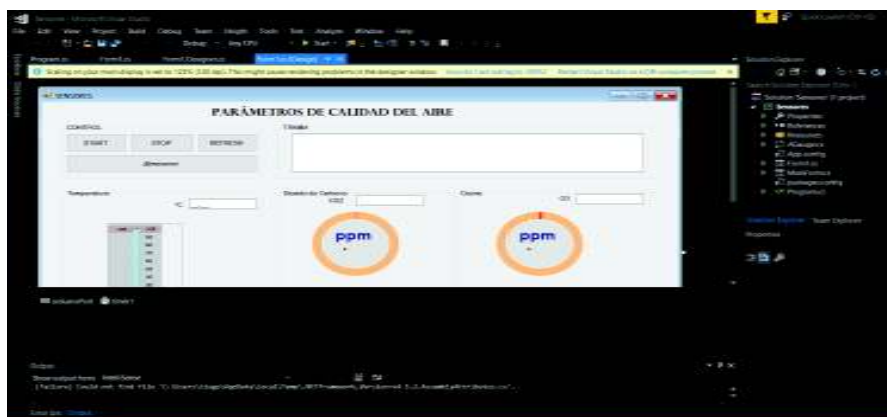


- c) Abrir en la PC la interfaz gráfica del operador.

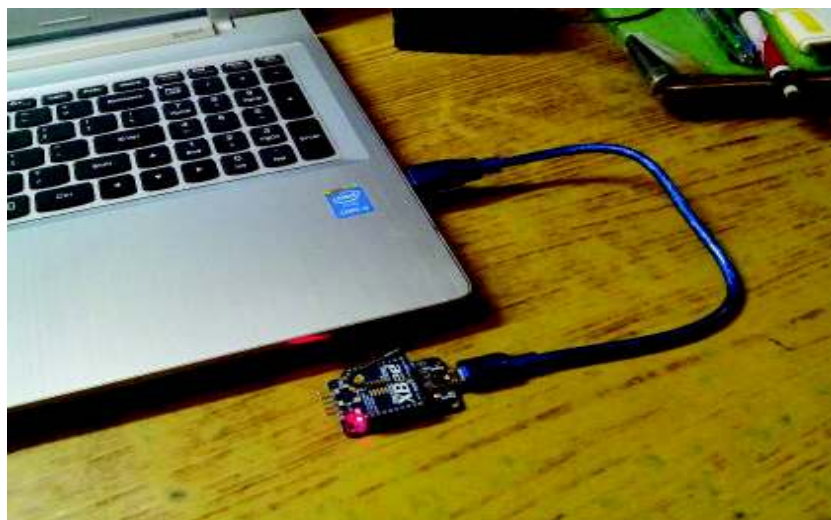
1. La carpeta Sensores tiene el ejecutable de Sensores.sln, para correr el *software* en la PC debe tener instalado previamente *Visual Studio 2017*, al hacer doble clic se abre la interfaz gráfica.



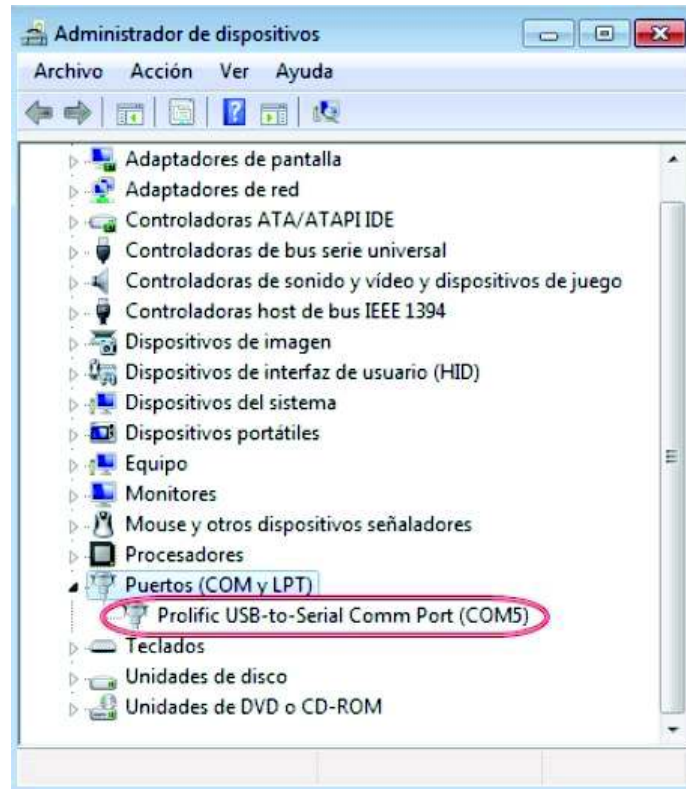
2. Si está bien instalado se dispone de la siguiente figura donde está el diseño de la interfaz gráfica.



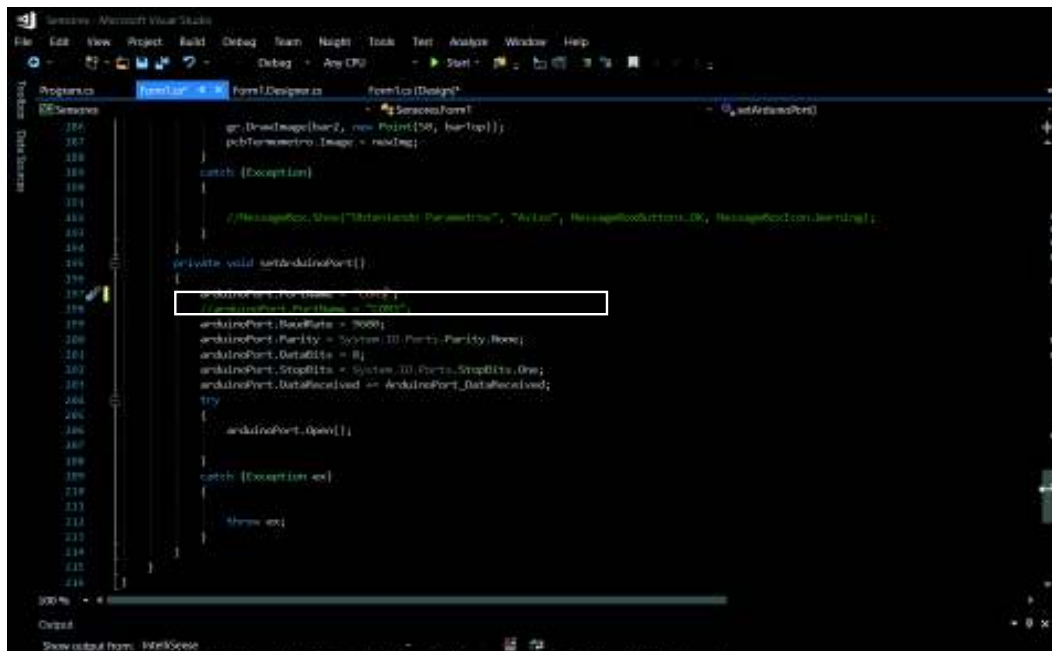
- d) Conectar el módulo *XBee Pro USB* al computador.



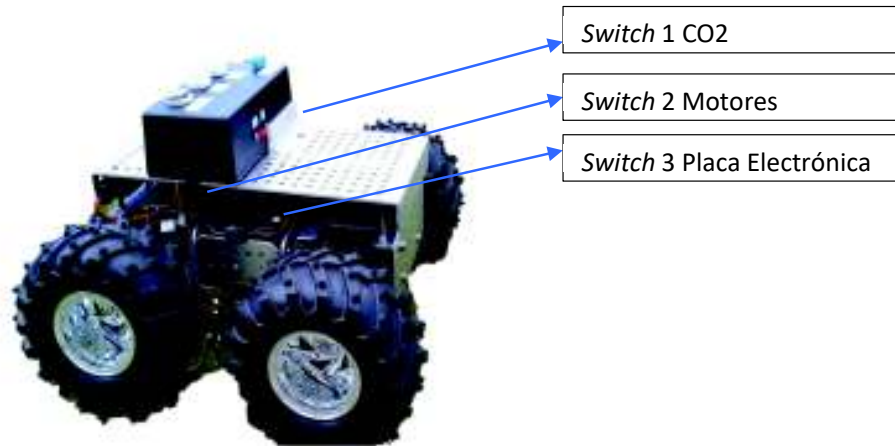
- e) Verificar el puerto COM al cual está conectado el *XBee Pro* en administrador de dispositivos de la PC



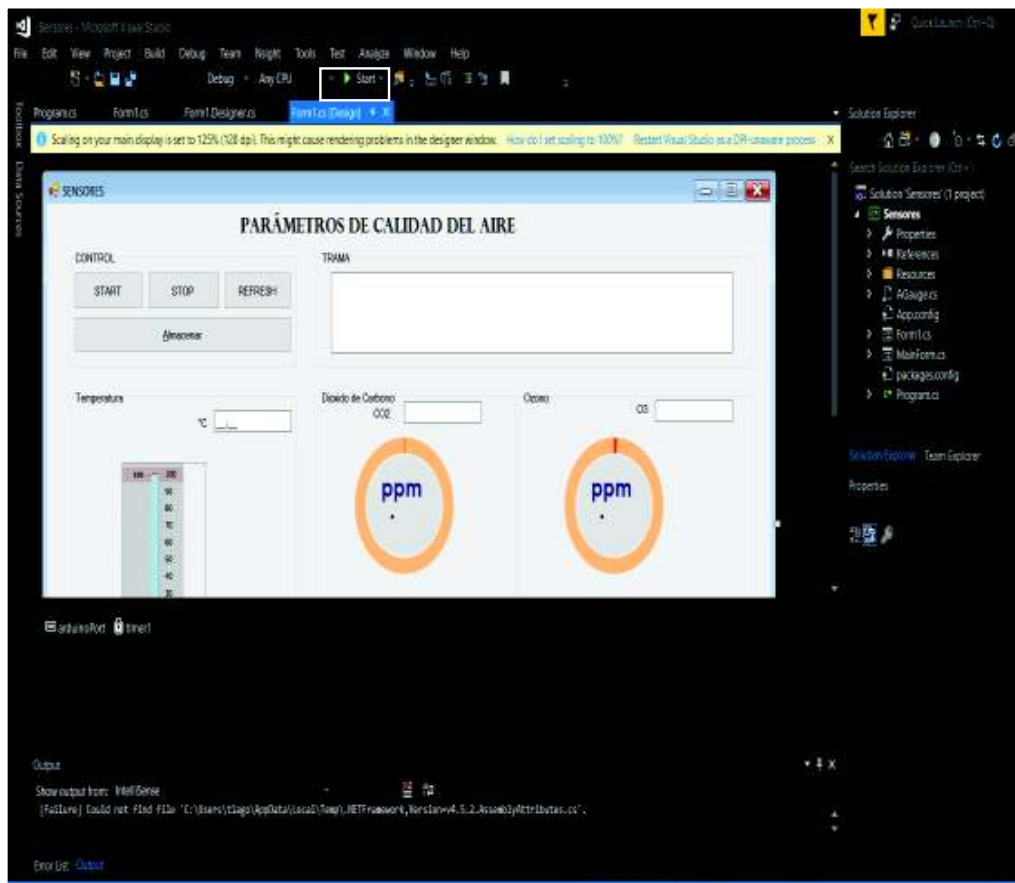
- f) En la interfaz de *Visual Studio* en la pestaña *form1.cs* colocar el número del puerto COM que se dispone en administrador de dispositivos.



- g) Colocar todos los *Switch* del *robot* móvil en encendido y observar si los leds de las diferentes placas se prenden.



- h) En la interfaz de *Visual Studio* dar clic en *Start*.



- i) Se despliega la interfaz principal, para que empiece a adquirir los datos, dar clic en *Start*.



- j) Colocar todos los *Switch* en bajo del control FLY-SKY como se observa en la figura.



k) Encender el control FLY-SKY.



l) Si se realizó bien los pasos el *robot* está listo para su funcionamiento.



Manual de mantenimiento

- a) Usar un cargador balanceador para cargar y descargar las baterías de *LiPo*.



- b) Usar con una frecuencia de 30 min un *tester* de batería para verificar que el valor de la carga no descienda 20% de su capacidad en las baterías de *LiPo*.



- c) Después de utilizar el *robot* en diferentes terrenos limpiar los acoples de las llantas y la estructura con un cepillo para remover impurezas.
- d) Realizar un *chek* de continuidad de cables mediante el uso de un multímetro después de operar el robot.