



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

EXTENSIÓN LATACUNGA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
POSICIONAMIENTO TERRESTRE UTILIZANDO PROTOCOLO
ZIGBEE PARA PROVEER DE UN SISTEMA DE SEGURIDAD
EN EL BARRIO SANTA LUCIA CENTRO DEL CANTÓN
TISALEO”.**

VÍCTOR RAFAEL PÉREZ MIRANDA

**Tesis presentada como requisito previo a la obtención del
grado de:**

INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN

2013

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

EXTENSIÓN LATACUNGA

CARRERA DE INGENIERÍA EN

ELECTRÓNICA E INSTRUMENTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

PÉREZ MIRANDA VÍCTOR RAFAEL

DECLARO QUE:

El proyecto de grado denominado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE POSICIONAMIENTO TERRESTRE UTILIZANDO PROTOCOLO ZIGBEE PARA PROVEER DE UN SISTEMA DE SEGURIDAD EN EL BARRIO SANTA LUCIA CENTRO DEL CANTÓN TISALEO”, ha sido desarrollado con base a una investigación exhaustiva conforme las citas que constan el pie de las páginas correspondiente, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Diciembre de 2013.

Pérez Miranda Víctor Rafael

180418270-5

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
EXTENSIÓN LATACUNGA

CARRERA DE INGENIERÍA EN
ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

ING. JOSÉ BUCHELI

ING. DAVID RIVAS

CERTIFICAN:

Que el trabajo titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE POSICIONAMIENTO TERRESTRE UTILIZANDO PROTOCOLO ZIGBEE PARA PROVEER DE UN SISTEMA DE SEGURIDAD EN EL BARRIO SANTA LUCIA CENTRO DEL CANTÓN TISALEO” realizado por el señor: VÍCTOR RAFAEL PÉREZ MIRANDA, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas-ESPE.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, **SI** recomiendan su publicación. El mencionado trabajo consta de 2 documentos empastados y 2 discos compactos el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a: VÍCTOR RAFAEL PÉREZ MIRANDA, que lo entregue al Ing. José Bucheli Andrade, en su calidad de Director de la Carrera.

Latacunga, Diciembre de 2013.

Ing. José Bucheli

Ing. David Rivas

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
EXTENSIÓN LATACUNGA

CARRERA DE INGENIERÍA EN
ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Yo, VÍCTOR RAFAEL PÉREZ MIRANDA

Autorizan a la Universidad de las Fuerzas Armadas-ESPE la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE POSICIONAMIENTO TERRESTRE UTILIZANDO PROTOCOLO ZIGBEE PARA PROVEER DE UN SISTEMA DE SEGURIDAD EN EL BARRIO SANTA LUCIA CENTRO DEL CANTÓN TISALEO”, cuyo contenido, ideas y criterios son de MI exclusiva responsabilidad y autoría.

Latacunga, Diciembre del 2013.

Pérez Miranda Víctor Rafael

180418270-5

DEDICATORIA

A mi madre Rosario, gestora de mis más grandes logros, incondicional sobre
todas las cosas.

Una de las muchas metas que serán dedicadas para ti.

Los hombres son los que sus madres han hecho de ellos.

Ralph Waldo Emerson

AGRADECIMIENTO

Agradezco a todos los que han hecho posible la realización del presente proyecto.

A todos los habitantes de Santa Lucia Centro que han colaborado en todo momento siendo un gran apoyo para mí.

Al Ing. David Rivas, que ha formado parte de mi instrucción desde el inicio, además brindándome el apoyo para este proyecto.

Al Ing. José Bucheli, a quien admiro por ser un gran profesional, del cual he aprendido y que considero como ejemplo para mí y las generaciones posteriores. Lo considero más que un maestro un gran amigo

ÍNDICE DE CONTENIDO

DECLARACIÓN DE RESPONSABILIDAD	ii
CERTIFICACIÓN	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
RESUMEN	1
ABSTRACT	2
1. FUNDAMENTOS TEÓRICOS	3
1.1. Introducción	3
1.2. Características Red Zigbee	3
1.2.1. Definición Red Zigbee	4
1.3. Arquitectura Red Zigbee	7
1.4. Tipos de Dispositivos	8
1.4.1. Coordinador	8
1.4.2. Routers	9
1.4.3. End Device	9
1.5. Modelo de Red Zigbee	10
1.6. Zigbee en el modelo OSI	11
1.6.1. Capa de Red	11
1.6.2. Capa Física	13
1.6.3. Capa de Enlace	14
1.7. Módulos Xbee DIGI	15
1.7.1. Distribución física de los terminales	17
1.8. Tarjeta Arduino UNO	18
1.9. Tarjeta Arduino Mega	19
1.10. Arduino IDE	20
1.11. Xbee Shield	23
1.12. Google Earth	24

1.12.1. Uso de Google Earth	24
1.13. Nokia Maps.....	26
1.13.1. Características Mapas	27
2. DISEÑO DEL SISTEMA	28
2.1. Distribución física de los transmisores.....	28
2.2. Diseño de la Red Zigbee	29
2.2.1. Modo API.....	29
2.2.2. Longitud	29
2.2.3. Campo de Datos.....	30
2.2.4. CheckSum	31
2.2.5. API frames de mayor uso	31
2.2.6. AT Command-pregunta por el valor del parámetro.....	32
2.2.7. Zigbee Transmit Request.....	32
2.2.8. Explicit Addressing Zigbee Command frame	34
2.2.9. Remote AT command request	35
2.2.10. Create source route	36
2.2.11. AT command response.....	37
2.2.12. Modem Status.....	37
2.2.13. Zigbee Transmit status	38
2.2.14. Zigbee Recieve packet	40
2.2.15. Zigbee explicit RX indicator.....	41
2.2.16. Zigbee IO data simple RX indicator	42
2.2.17. Xbee sensor read indicator	43
2.2.18. Node Identification Indicator.....	44
2.2.19. Remote Command response	45
2.3. Técnicas de Localización.....	46

2.3.1. Determinación de la distancia por índice de potencia de señal recibida (RSSI)	47
2.3.2. Trilateración	49
2.4. Configuración de los módulos Xbee	56
2.5. Análisis con Radio Mobile	61
2.6. Programación Arduino Mega	65
2.7. Diseño terminal remoto	67
2.8. Recepción de datos desde un terminal remoto	68
2.9. Librería Arduino para LCD	70
2.10. Programación Arduino UNO	71
2.11. Interface de Usuario	73
2.11.1. Labview VISA	74
3. IMPLEMENTACIÓN DEL SISTEMA	76
3.1. Estación Base	76
3.2. Nodo Móvil	77
3.3. Panel frontal para Usuario	77
3.4. Datos de Usuarios	78
3.4.1. Referencia de texto	79
3.4.2. Nombre usuario	80
3.4.3. Coordenadas espaciales de los nodos	81
3.5. Localización del Nodo Móvil en Google Earth	82
3.5.1. Archivo kml	83
3.5.1.1. Marcas de Posición	83
3.6. Interface para visualizar el nodo móvil	84
3.7. Visualización mediante Google Earth	85
3.8. Pruebas de Nodo móvil	86
3.9. Alcances y Limitaciones	87
3.9.1. Alcances	87
3.9.2. Limitaciones	88

4. CONCLUSIONES Y RECOMENDACIONES	89
4.1. Conclusiones	89
4.2. Recomendaciones	90
BIBLIOGRAFIA	91
ANEXOS	93
1. Estación Base	94
2. Nodo Móvil	95
3. Instructivo para el uso del transmisor XBee	96
4. Manual del Operador	98
5. Diagrama esquemático Arduino Uno	110
6. Diagrama esquemático Arduino Mega	111
7. Diagrama esquemático Xbee Shield	112
8. Diagramas Electricos Nodo Fijo	113
9. Diagrama esquemático elementos adicionales Estación Base	114
10. Zonas de cobertura de la red	115

ÍNDICE DE FIGURAS

1. FUNDAMENTOS TEÓRICOS	3
1.1. Paquetes incluidos en la capa MAC	8
1.2. Redes Zigbee	10
1.3. Diagrama de bloques de una red Zigbee	11
1.4. Ejemplo de Red Zigbee	12
1.5. ZigBee en el modelo OSI	14
1.6. Módulo xbee de Digi	15
1.7. Características físicas del chip xbee	17
1.8. Arduino UNO	18
1.9. Arduino Mega	19
1.10. Interface de programación Arduino	21
1.11. Xbee Shield	23
1.12. Funciones Google Earth	24
1.13. Íconos de la barra de herramientas	26
1.14. Nokia 500.....	26
1.15. Nokia Maps	27
1.16. Ejemplo Panel Frontal	28
1.17. Ejemplo Diagrama de Bloques	29
1.18. Paleta de Herramientas	30
1.19. Paleta de Controles	32
1.20. Paleta de Funciones	32
2. DISEÑO DEL SISTEMA	28
2.1. Mapa Satelital Google Maps con la imagen correspondiente al caserío Santa Lucía	28

2.2.	Frame en modo API	29
2.3.	Campo de datos en el API frame	30
2.4.	Potencia versus distancia	48
2.5.	Potencia recibida como RSSI	48
2.6.	Intersección de 3 esferas	50
2.7.	Interface inicial X-CTU	53
2.8.	Conexión de X-CTU con el módulo xbee	54
2.9.	Parámetros a escribir en el firmware del chip xbee	55
2.10.	Parámetro D0	56
2.11.	Parámetro IR	57
2.12.	Altitudes en el barrio Santa Lucía Centro	62
2.13.	Transmisores ubicados en el mapa	63
2.14.	Radiación transmisores	64
2.15.	Estación Base con transmisores restantes	64
2.16.	Circuito regulador de voltaje con zener	67
2.17.	Diagrama de flujo para calcular el Checksum	72
2.18.	Diagrama de bloques VISA read	73
2.19.	Panel frontal VISA read	74
2.20.	Diagrama de bloques para la escritura	74
3. IMPLEMENTACIÓN DEL SISTEMA		
3.1.	LCD 2x16 distribución de pines	69
3.2.	Panel frontal de Usuario	71
3.3.	Interface para visualización del Mapa	77
3.4.	Localización de los nodos fijos	78
3.5.	Marca de prueba	78
3.6.	Datos de localización del nodo móvil	78

ÍNDICE DE TABLAS

1. FUNDAMENTOS TEÓRICOS	3
1.1. Comparativa entre tecnologías wireless	6
1.2. Parámetros de modulación	13
1.3. Características módulos xbee	16
1.4. Distribución de pines del chip xbee	17
1.5. Características Arduino Uno	22
1.6. Características Arduino Mega	22
1.7. Iconos del Arduino IDE	22
2. DISEÑO DEL SISTEMA	28
2.1. Identificadores de API frame	30
2.2. AT Command frame	31
2.3. AT Command – Pregunta por el valor del parámetro frame	32
2.4. Transmit Request API frame	33
2.5. Explicit Addressing ZigBee Command Frame	35
2.6. Remote AT Command Request frame	35
2.7. Create Source Route frame	36
2.8. AT Command Response frame	37
2.9. Modem status frame	38
2.10. ZigBee Transmit Status frame	39
2.11. ZigBee Receive Packet frame	40
2.12. ZigBee Explicit Rx Indicator frame	41
2.13. ZigBee IO Data Sample Rx Indicator frame	42
2.14. XBee Sensor Read Indicator frame	43
2.15. Node Identification Indicator frame	44
2.16. Muestra un Remote Command Response frame	45

2.17. Parámetros coordinadores y routers	56
2.18. Frame recibido por el coordinador	61

3. IMPLEMENTACIÓN DEL SISTEMA

3.1. Direcciones lógicas y físicas	72
3.2. Direcciones lógicas y texto	73
3.3. Direcciones y Nombres de Usuario	74
3.4. Direcciones y coordenadas espaciales de los nodos	75
3.5. Coordenadas resultantes.....	78

RESUMEN

Con la aparición de varias tecnologías inalámbricas, contrastado con la gran inversión que representa un sistema de seguridad, se presenta mediante el siguiente proyecto una alternativa en cuanto a seguridad se refiere con tecnología inalámbrica usando protocolo Zigbee.

Además se ve complementado con el establecimiento de una red tipo malla que permitirá la localización de un nodo móvil mediante trilateración usando algoritmos que permitirán establecer las coordenadas de dicho nodo, prestando así mayores prestaciones en caso de una señal de alarma generada desde un lugar dentro del barrio Santa Lucia Centro.

Como complemento a esto, se presentan los dispositivos que conforman el mencionado proyecto como son: una estación base, un nodo móvil, y nodos fijos en los domicilios. Se realiza la presentación de una interface gráfica de usuario (HMI), que permitirá visualizar de manera óptima los datos manejados en la red presentada.

Como parte final podemos resaltar que el presente proyecto tiene como objetivo principal beneficiar a las familias del barrio Santa Lucia centro, de tal suerte que se establezca una forma de alertar posibles amenazas delictivas.

Se pretende además proporcionar información que levante el interés de los moradores del ya mencionado barrio para de esta forma incrementar el número de usuarios de la red.

ABSTRACT

With the emergence of various wireless technologies, contrasted with the huge investment that represents a security system, the following is presented as an alternative project for security by using the wireless technology known as Zigbee protocol.

In addition it's complemented by the establishment of a mesh network that will allow the location of a mobile node using trilateration applying algorithms that will provide the coordinates of that node, thus providing better performance in case of an alarm signal generated from a location within Santa Lucia Centro.

As a complement to this, the devices that make up the project as are mentioned: a base station, a mobile node and fixed nodes in homes. It makes filling a graphical user interface (HMI), which allow optimal way to visualize the data handled in the presented Network

As the final part we stress that this project's main objective benefit families of Santa Lucia Centro, in such a way to establish options to alert possible criminal threats.

It aims to provide information to raise the interest of the inhabitants of the mentioned neighborhood to thereby increase the number of network users.

CAPÍTULO 1

Fundamentos Teóricos

1.1. Introducción

El creciente desarrollo de tecnologías inalámbricas como solución para distintas aplicaciones tanto domésticas como industriales generando una disminución en el costo de producción, por tanto de venta de micro controladores que manejan protocolos de transmisión inalámbricos, es por esto que la creación de sistemas mediante el uso de dichos dispositivos es cada vez mayor.

Si bien la electrónica necesaria está disponible la oferta de soluciones inalámbricas es aún baja, por esta razón y la falta de servicios de localización asociados a estas redes se da la necesidad de crear un posicionamiento similar al GPS basándose en esta tecnología ya disponible en el mercado. Existe un gran potencial tanto social como económico en los servicios basados en la localización, y, por distintas circunstancias, aún no han sido muy explotados.

1.2. Características Red Zigbee¹

Algunas de las características de ZigBee son:

- ZigBee opera en las bandas libres ISM (Industrial, Scientific & Medical) de 2.4 GHz, 868 MHz (Europa) y 915 MHz (Estados Unidos).
- Tiene una velocidad de transmisión de 250 Kbps y un rango de cobertura de 10 a 75 metros.

¹ <http://www.seccperu.org/files/ZigBee.pdf>

- A pesar de coexistir en la misma frecuencia con otro tipo de redes como WiFi o Bluetooth su desempeño no se ve afectado, esto debido a su baja tasa de transmisión y, a características propias del estándar IEEE 802.15.4.
- Capacidad de operar en redes de gran densidad, esta característica ayuda a aumentar la confiabilidad de la comunicación, ya que entre más nodos existan dentro de una red, entonces, mayor número de rutas alternas existirán para garantizar que un paquete llegue a su destino.
- Cada red ZigBee tiene un identificador de red único, lo que permita que coexistan varias redes en un mismo canal de comunicación sin ningún problema.
- Teóricamente pueden existir hasta 16 000 redes diferentes en un mismo canal y cada red puede estar constituida por hasta 65 000 nodos, obviamente estos límites se ven truncados por algunas restricciones físicas (memoria disponible, ancho de banda, etc.).
- Es un protocolo de comunicación multi-salto, es decir, que se puede establecer comunicación entre dos nodos aun cuando estos se encuentren fuera del rango de transmisión, siempre y cuando existan otros nodos intermedios que los interconecten, de esta manera, se incrementa el área de cobertura de la red.
- Su topología de malla (MESH) permite a la red auto recuperarse de problemas en la comunicación aumentando su confiabilidad.

1.2.1. Definición de Zigbee

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE-802.15.4. Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más

de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philips, Motorola, Invensys,...), muchas de ellas fabricantes de semiconductores.²

Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticos.

Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth no utiliza FHSS (Frequency hopping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de emisión del dispositivo así como el tipo de antenas utilizadas (cerámicas, dipolos,...) El alcance normal con antena dipolo en visión directa suele ser aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m. La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Por último decir que una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos. La realidad es menor, siendo, de todas formas, de miles de equipos.³

La tabla 1.2 presenta una comparación entre las tecnologías wireless como son Bluetooth, Wi-Fi y Zigbee.

² <http://bibdigital.epn.edu.ec/bitstream/15000/1363/1/CD-2064.pdf>

³

<http://www2.uacj.mx/IT/IEC/Digitales/PROYECTOS/Proyectos%20Titulacion%20Enero%20Junio%202011/Monitoreo%20de%20sensado%20de%20temperatura%20y%20humedad%20en%20cuartos%20de%20comunicaciones%20con%20tecnologia%20de%20bajo%20costo.pdf>

Tabla 1.1. Comparativa entre tecnologías wireless.⁴

Estándar	Ancho de Banda	Consumo de potencia	Ventajas	Aplicaciones
Wi-Fi	Hasta 54Mbps	400mA transmitiendo, 20mA en reposo	Gran ancho de banda	Navegar por Internet, redes de ordenadores, transferencia de ficheros
Bluetooth	1 Mbps	40mA transmitiendo, 0.2mA en reposo	Interoperatividad, sustituto del cable	Wireless USB, móviles, informática casera
ZigBee	250 kbps	30mA transmitiendo, 3mA en reposo	Batería de larga duración, bajo coste	Control remoto, productos dependientes de la batería, sensores , juguetería

⁴ <http://www.domodesk.com/a-fondo-zigbee>

1.3. Arquitectura Red Zigbee ⁵

Siguiendo el estándar del modelo de referencia OSI (Open Systems Interconnection), en el gráfico 1, aparece la estructura de la arquitectura en capas. Las primeras dos capas, la física (PHY) y la de acceso al medio (MAC), son definidas por el estándar IEEE 802.15.4. Las capas superiores son definidas por la Alianza ZigBee. El grupo de trabajo de IEEE pasó el primer borrador de la capa física y la de acceso al medio en 2003. Una versión final de la capa de red (NWK) se acabó el año pasado, y en Junio del 2005 tenemos ya un Zigbee 1.0 publico.

Los productos ZigBee trabajan en una banda de frecuencias que incluye la 2.4 Ghz (mundial), de 902 a 928 Mhz (en Estados Unidos) y 866Mhz (en Europa). La transferencia de datos de hasta 250 Kbs puede ser transmitido en la banda de 2.4Ghz (16 canales), hasta 40kps en 915Mhz (10 canales) y a 20kps en la de 868Mhz (un solo canal). La distancia de transmisión puede variar desde los 10 metros hasta los 75, dependiendo de la potencia de transmisión y del entorno. Al igual que WiFi, ZigBee usa la DSSS (secuencia directa de espectro ensanchado) en la banda 2.4 Ghz. En las bandas de 868 y 900Mhz también se utiliza la secuencia directa de espectro ensanchado pero con modulación de fase binaria.

La fig. 1.1 nos muestra los campos de los cuatro tipos de paquetes básicos: datos, ACK, MAC y baliza.

⁵ <http://www.seccperu.org/files/ZigBee.pdf>

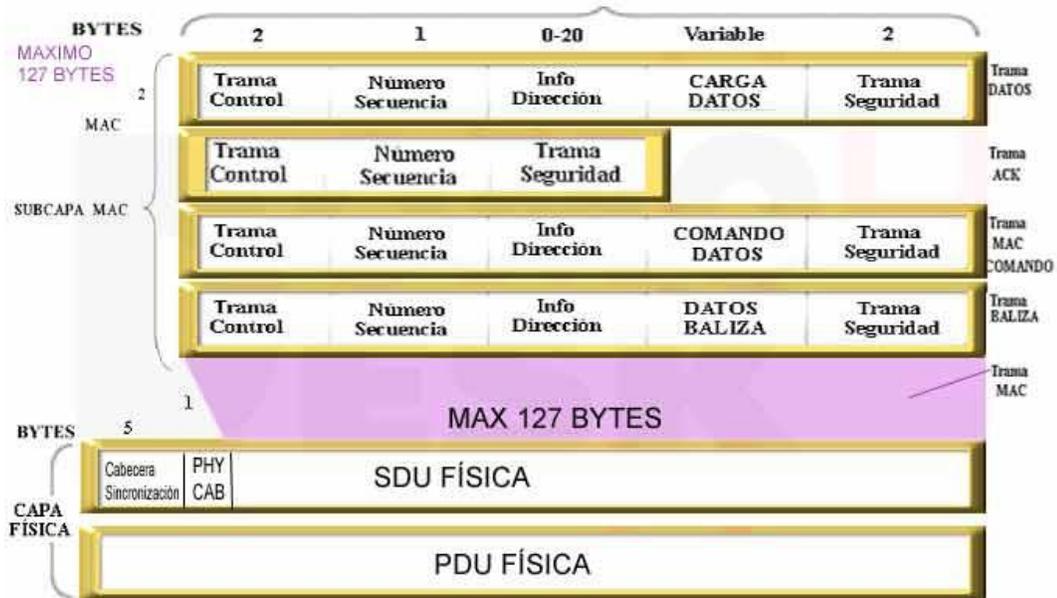


Fig. 1.1. Paquetes incluidos en la capa MAC.⁶

1.4. Tipos de Dispositivos⁷

Una red Zigbee la forman básicamente 3 tipos de elementos. Un único dispositivo Coordinador, dispositivos Routers y dispositivos finales (end points).

1.4.1. Coordinador

Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones (como hablábamos antes) y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points. Una vez formada

⁶ <http://sx-de-tx.wikispaces.com/ZIGBEE>

⁷ <http://www.seccperu.org/files/ZigBee.pdf>

la red, el Coordinador hace las funciones de Router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

1.4.2. Routers

Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para enrutar un paquete de información. Lógicamente un router debe unirse a una red Zigbee antes de poder actuar como Router retransmitiendo paquetes de otros routers o de End points.

1.4.3. End Device

Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Router, es decir, no puede enviar información directamente a otro end device. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

ZigBee es un protocolo pensado para comunicaciones a baja velocidad entre dos o varios dispositivos, se pueden formar redes con miles de dispositivos comunicándose entre sí, por lo que es ideal para muchas aplicaciones.

ZigBee estuvo desde un inicio pensado para ser de implementación y donde los dispositivos que forman parte de una red pueden requerir un bajo consumo de energía, llegando a estar funcionando durante años con un par de pilas.

En la fig. 1.2 se establece la función de cada uno de los dispositivos asociados a los tipos de red que permite el protocolo.

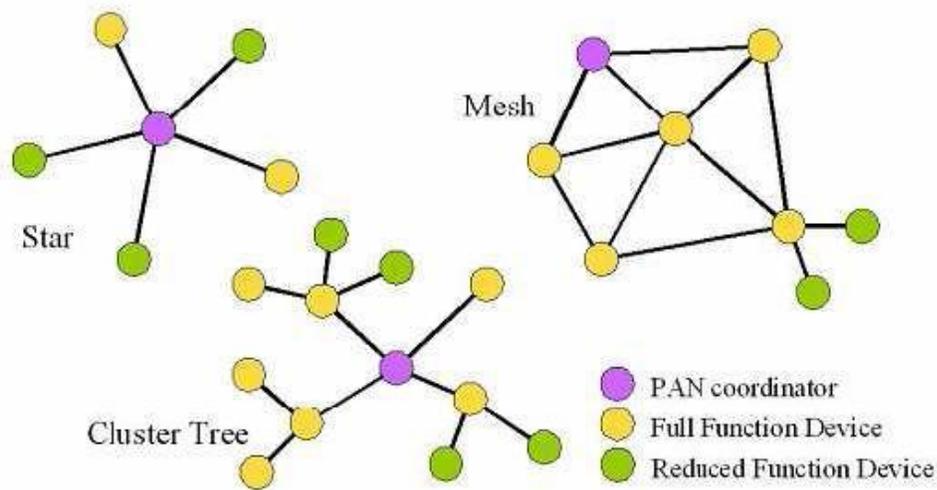


Fig. 1.2. Redes Zigbee.

1.5. Modelo de Red Zigbee⁸

En la configuración en estrella, uno de los dispositivos tipo FFD asume el rol de coordinador de red y es responsable de inicializar y mantener los dispositivos en la red. Todos los demás dispositivos Zigbee, conocidos con el nombre de dispositivos finales, ‘hablan’ directamente con el coordinador. En la configuración de malla, el coordinador ZigBee es responsable de inicializar la red y de elegir los parámetros de la red, pero la red puede ser ampliada a través del uso de routers ZigBee. El algoritmo de encaminamiento utiliza un protocolo de pregunta-respuesta (request-response) para eliminar las rutas que no sean óptimas, La red final puede tener hasta 254 nodos. Utilizando el direccionamiento local, se puede configurar una red de más de 65000 nodos.

La trama general de operaciones (GOF) es una capa que existe entre la de aplicaciones y el resto de capas. La GOF suele cubrir varios elementos que

⁸ <http://www.domodesk.com/a-fondo-zigbee>

son comunes a todos los dispositivos, como el subdireccionamiento y los modos de direccionamientos y la descripción de dispositivos, como el tipo de dispositivo, potencia, modos de ‘dormir’ y coordinadores de cada uno. Utilizando un modelo, la GOF especifica métodos, eventos, y formatos de datos que son utilizados para constituir comandos y las respuestas a los mismos.

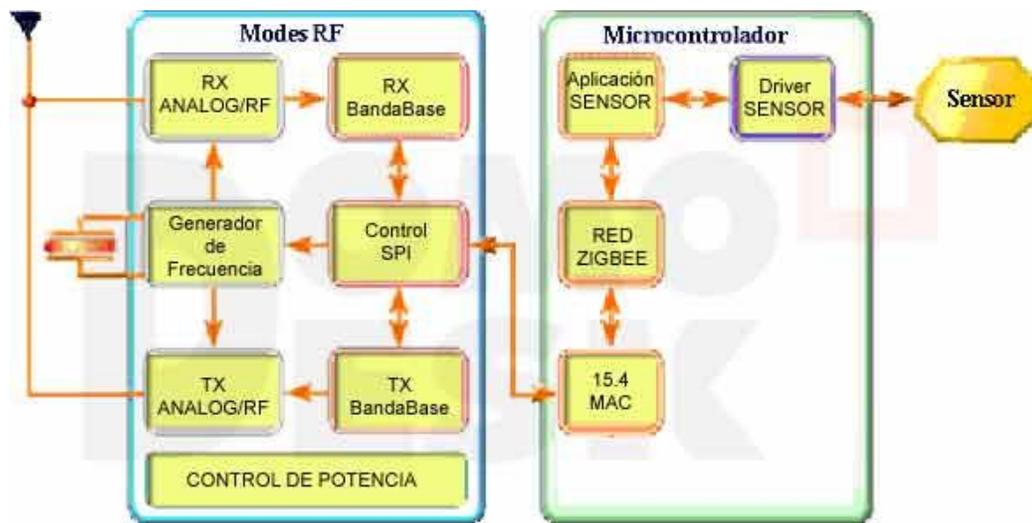


Fig. 1.3. Diagrama de bloques de un dispositivo Zigbee.⁹

1.6. ZigBee en el modelo OSI

1.6.1. Capa de red ¹⁰

El estándar IEEE 802.15.4 soporta múltiples topologías para su conexión en red, entre ellas la topología tipo “Estrella”, la topología “Punto a Punto” y “Punto a Multipunto”. La topología a escoger es una elección de diseño y va a estar dada por la aplicación a la que se desee orientar, algunas aplicaciones como periféricos e interfaces de PC, requieren de conexiones de baja potencia de tipo estrella, mientras que otras como los perímetros de seguridad

⁹ <http://www.domodesk.com/a-fondo-zigbee>

¹⁰ <http://dspace.ups.edu.ec/bitstream/123456789/206/4/Capitulo%203.pdf>

requieren de una mayor área de cobertura por lo que es necesario implementar una red Punto a Punto.

La red puede tener un Coordinador de red (PAN) y múltiples RFDs/FFDs, en la *fig. 1.1* podemos observar un ejemplo de red ZigBee: ¹¹

- FFD (Full Function Device): Es un dispositivo de función completa que se puede usar en cualquier topología. Tiene la capacidad de poder ser Coordinador de red y puede hablar con cualquier dispositivo.
- RFD (Reduced Function Device): Es un dispositivo de función reducida. Limitado a una topología en estrella. No puede ser Coordinador de red, solo puede hablar con un Coordinador y es muy fácil de implementar.

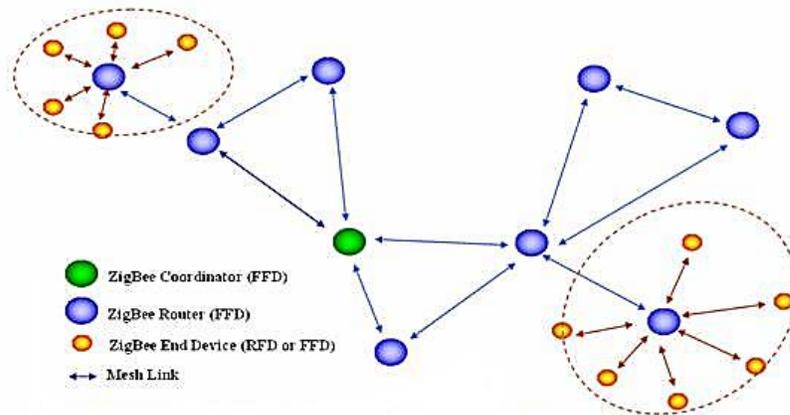


Fig. 1.4. Ejemplo de Red Zigbee¹²

¹¹ <http://www.javierlongares.com/arte-en-8-bits/introduccion-a-zigbee-y-las-redes-de-sensores-inalambricas/>

¹² <http://es.convdocs.org/docs/index-57159.html?page=3>

1.6.2. Capa física ¹³

La capa física a 2.4 GHz emplea una técnica de modulación semiortogonal basada en métodos DSSS.

Los datos binarios están agrupados en símbolos de 4 bits, y cada símbolo especifica una de las 16 secuencias de transmisión semi-ortogonales de código de pseudo-ruido (PN). Las secuencias PN son concadenadas para que sean datos de símbolos exitosos, y la secuencia agregada al chip es modulada en la portadora utilizando MSK (Minimum Shift Keying). El uso de símbolos “casi ortogonales” simplifica la implementación a cambio de un desempeño ligeramente menor (< 0.5 dB).

En términos de eficiencia (energía requerida por bit), la señalización ortogonal mejora su funcionamiento en 2 dB respecto a la BPSK diferencial. Sin embargo, en términos de sensibilidad de recepción, la capa física 868/915 MHz tiene una ventaja de 6-8 dB debido a que tiene velocidades de transmisión más bajas.

Por supuesto, que en ambos casos las pérdidas de implementación debido a la sincronización, forma del pulso, simplificaciones en el detector y demás cosas, resultan en desviaciones en sus curvas óptimas de detección. Los parámetros de modulación se resumen en la tabla 1.1.

¹³

http://www.dea.icaei.upco.es/sadot/Comunicaciones/avanzadas/Alberto_Gasc%C3%B3n_Zigbee%20y%20el%20Est%C3%A1ndar%20IEEE%20802.15.4.pdf

Tabla 1.2. Parámetros de modulación¹⁴

Capa Física	Banda	Parámetros de los Datos			Parámetros de chip	
		Velocidad bits(kbps)	Velocidad símbolos (kbaud)	Modulación	Velocidad chip (kchip/s)	Modulación
868/915 MHz	868-868.6 MHz	20	20	BPSK	300	BPSK
	902-928 MHz	40	40	BPSK	600	BPSK
2.4 GHz	2.4-2.4835 GHz	250	62.5	16-ary ortogonal	2000	O-QPSK

1.6.3. Capa de enlace

El estándar IEEE 802 divide la capa de enlace de datos (DLL) en dos sub-capas, la sub-capa de enlace al Control de Acceso al Medio (Medium Access Control, MAC) y la sub-capa de Control de enlaces lógicos (Logical Link Control, LLC). El LLC es común a todos los estándares 802, mientras que la sub-capa MAC depende del hardware y varía respecto a la implementación física de esta capa. La *figura 1.2* muestra la forma en que el estándar IEEE 802.15.4 se basa en la Organización Internacional para la Estandarización (ISO) del modelo de referencia para la Interconexión de Sistemas Abiertos (OSI).

¹⁴

http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Redes_y_Seguridad/Trabajos_Finales/Dig nanni_Jorge_Pablo.pdf

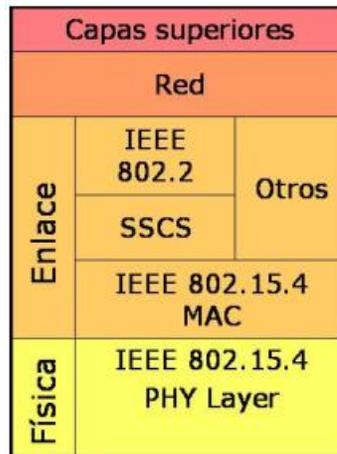


Fig. 1.5. ZigBee en el modelo OSI

1.7. Módulos XBEE DIGI¹⁵

El chip Xbee fabricado por Digi Internacional, si bien no es el único fabricante, su construcción tiene estándares que termina por hacer indiferente que dispositivo sea usado, en este caso el mostrado en la figura 1.6.



Fig.1.6. Módulo xbee de Digi.

¹⁵ http://ftp1.digi.com/support/documentation/90000976_P.pdf

El fabricante Digi, establece una tabla comparativa de sus productos actualizados de tal suerte que, se puede considerar información tal como potencia, alcance, velocidad de datos, presentados en la tabla 1.3.

Tabla 1.3. Características módulos xbee.¹⁶

ESPECIFICACIONES TÉCNICAS DEL XBEE PRO S2B		
Rendimiento	Cobertura en interiores (sin línea de vista)	Hasta 300 pies (90 m)
	Cobertura con línea de vista	Hasta 2 millas (3200 m)
	Potencia de salida	63 mW (+18 dBm)
	RF, Velocidad de datos	250 Kbps
	Interfaz de proporción de datos	1200 bps hasta 1 Mbps
	Sensibilidad del receptor	-102 dBm
Requerimientos de potencia	Suministro de voltaje	2.7 V – 3.6 V DC
	Transmisión de corriente	205 mA @ 3.3 V
	Recepción de corriente	47 mA @ 3.3 V
	Intensidad de reposo	3.5 µA @ 25 °C
Información general	Banda de frecuencia	2.4 GHz ISM
	Métodos de configuración	AT y API, local o por aire
	Dimensiones	2.438cm x 3.294cm
	Peso	0.10 oz (3g)
	Temperatura de operación	-40 a 85 °C (industrial)
Trabajo en red y seguridad	Tipo de antena	Conector RP-SMA
	Topologías de red	Punto a punto, punto a multipunto, igual a igual y mesh
	Capacidad de canales	15 Canales de secuencia directa (11 a 25)
	Cifrado	128 bits
Certificaciones	Direccionamiento	PAN ID, 64-bit IEEE MAC
	FCC (EE.UU.), IC (Canadá), ETSI (Europa) y C-TICK (Australia)	

¹⁶ http://ftp1.digi.com/support/documentation/90000976_P.pdf

1.7.1 Distribución física de los terminales.¹⁷

La figura 1.7 muestra las características físicas del chip xbee, y la correspondiente distribución de pines en la tabla 1.3.

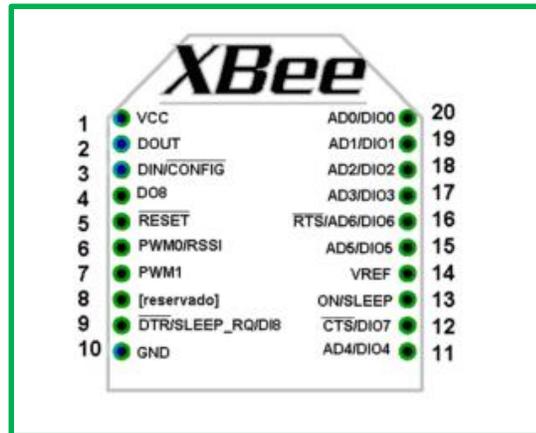


Fig. 1.7. Características físicas del chip xbee.

Tabla 1.4. Distribución de pines del chip xbee.

Pin	Nombre	Dirección	Descripción
1	VCC	-	Alimentación
2	DOUT	Salida	Salida UART
3	DIN	Entrada	Entrada UART
4	DIO12	E/S	Digital E/S 12
5	RST	Entrada	Reinicio Módulo
6	PWM0/RSSI/DIO10	E/S	Salida PWM /Intensidad de señal recibida/salida digital 10
7	DIO11	E/S	E/S Digital 11
8	Reservado	-	No conectar
9	SLEEP-RQ/DIO8	E/S	Sleep/ E/S Digital 8
10	GND	-	Tierra
11	DIO4	E/S	E/S Digital 4
12	DIO7	E/S	Control de flujo,E/S digital 7,CTS
13	ON	Salida	Indicador de estado, E/S digital 9
14	VREF	Entrada	
15	Asociado/dio5	E/S	Indicador Asociado, DE/S digital 5
16	DIO6	E/S	Control de flujo, RTS, E/S digital 6
17	AD3/DIO3	E/S	Entrada analógica o E/S digital 3
18	AD2/DIO2	E/S	Entrada analógica o E/S digital 2
19	AD1/DIO1	E/S	Entrada analógica o E/S digital 1
20	AD0/DIO0/Command Button	E/S	Entrada analógica o E/S digital 0, command button

¹⁷ http://ftp1.digi.com/support/documentation/90000976_P.pdf

1.8. Tarjeta Arduino Uno¹⁸

La Arduino Uno es una placa microcontrolador basada en el ATmega328. Tiene 14 entradas/salidas digitales (de las cuales 6 proporcionan salida PWM), 6 entradas analógicas, un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset.

La figura 1.8 muestra una Arduino Uno en su vista frontal.



Fig. 1.8. Arduino UNO

Tabla 1.5. Características Arduino Uno

<i>Microcontrolador</i>	ATmega328
<i>Voltaje de funcionamiento</i>	5V
<i>Voltaje de entrada (recomendado)</i>	7-12V
<i>Voltaje de entrada (limite)</i>	6-20V
<i>Pines E/S digitales</i>	14 (de los cuales 6 proveen salida PWM)

¹⁸ <http://arduino.cc/en/Main/arduinoBoardUno>

<i>Pines de entrada analógica</i>	6
<i>Intensidad por pin</i>	40 mA
<i>Intensidad en pin 3.3V</i>	50 mA
<i>Memoria Flash</i>	32 KB (ATmega328) de la cual 0.5 KB es usado como bootloader
<i>SRAM</i>	2 KB (ATmega328)
<i>EEPROM</i>	1 KB (ATmega328)
<i>Velocidad de reloj</i>	16 MHz

1.9. Arduino Mega¹⁹

La Arduino Mega es una placa microcontrolador basada ATmega1280. Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset.

La figura 1.9 muestra una Arduino Mega en su vista frontal.

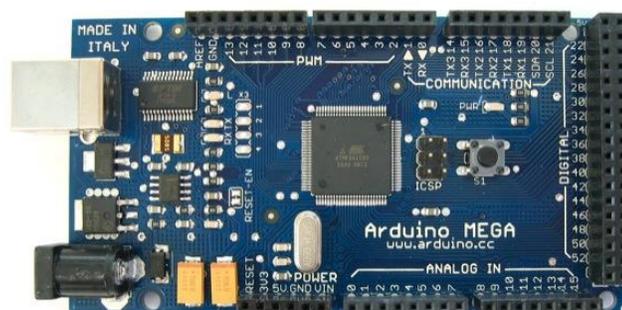


Fig. 1.9. Arduino Mega

¹⁹ <http://arduino.cc/en/Main/arduinoBoardMega>

Tabla 1.6. Características Arduino Mega.

<i>Microcontrolador</i>	ATmega1280
<i>Voltaje de funcionamiento</i>	5V
<i>Voltaje de entrada (recomendado)</i>	7-12V
<i>Voltaje de entrada (limite)</i>	6-20V
<i>Pines E/S digitales</i>	54 (14 proporcionan salida PWM)
<i>Pines de entrada analógica</i>	16
<i>Intensidad por pin</i>	40 mA
<i>Intensidad en pin 3.3V</i>	50 mA
<i>Memoria Flash</i>	128 KB de las cuales 4 KB las usa el gestor de arranque(bootloader)
<i>SRAM</i>	8 KB
<i>EEPROM</i>	4 KB
<i>Velocidad de reloj</i>	16 MHz

1.10. Arduino IDE²⁰

El entorno de Desarrollo Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús. Permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.

²⁰ <http://arduino.cc/en/Main/Software>

Arduino utiliza para escribir el software lo que denomina "sketch" (programa). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/reemplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie, la figura 1.10 muestra la interface de usuario del entorno de programación de Arduino.

Arduino siempre está desarrollando hardware y software continuamente, por tanto existirán nuevas versiones de Arduino IDE, cuando existe el lanzamiento de nuevas tarjetas, necesitan tener el soporte con controladores, al momento de realizada la programación de tarjetas para el presente proyecto la versión más actualizada, con soporte para las tarjetas usadas en el desarrollo es 1.0.5.

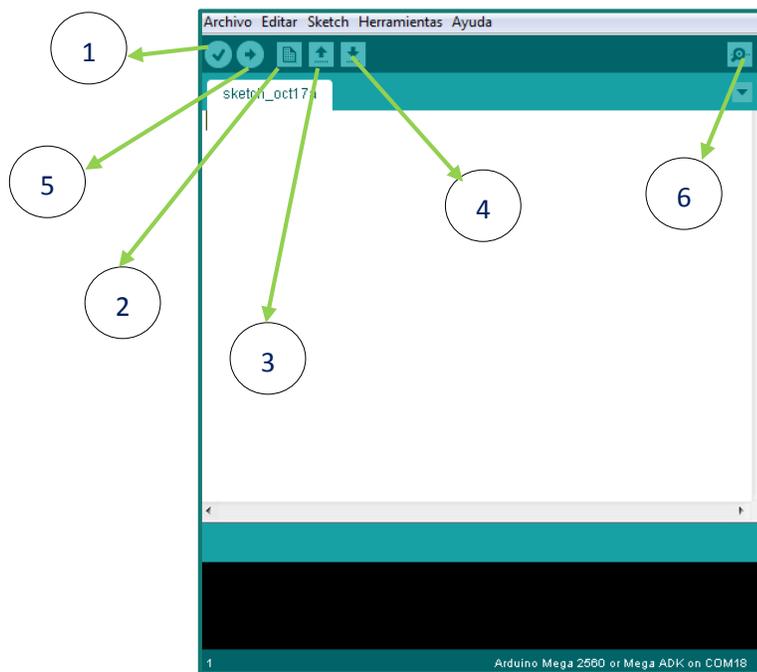


Fig. 1.10. Interface de programación Arduino.

La tabla 1.4 muestra los íconos que están en el IDE.

Tabla 1.7. Iconos del Arduino IDE.²¹

	<p>1. <i>Verify/Compile</i> Chequea el código en busca de errores.</p>
	<p>2. <i>New</i> Crea un nuevo <i>sketch</i>.</p>
	<p>3. <i>Open</i> Presenta un menú de todos los programas <i>sketch</i> de su "sketchbook", (<i>librería de sketch</i>).</p>
	<p>4. <i>Save</i> Salva el programa <i>sketch</i>.</p>
	<p>5. <i>Upload to I/O Board</i> Compila el código y lo vuelca en la placa E/S de Arduino.</p>
	<p>6. <i>Serial Monitor</i> Inicia la monitorización serie.</p>

²¹ <http://arduino.cc/en/Main/Software>

1.11. Xbee Shield²²

La Xbee shield permite a una placa Arduino comunicarse de forma inalámbrica usando Zigbee. Está basada en el módulo Xbee de MaxStream. La shield tiene pistas desde cada pin del Xbee hasta un orificio de soldar. También provee conectores hembra para usar los pines digitales desde 2 hasta 7 y las entradas analógicas, las cuales están cubiertas por la shield (los pines digitales de 8 a 13 no están cubiertos por la placa).

La figura 1.11 muestra un xbee shield con un módulo en su lugar correspondiente.



Fig. 1.11. Xbee Shield.

²² <http://arduino.cc/es/Main/ArduinoXbeeShield>

1.12. Google Earth²³

Google Earth permite viajar a cualquier parte del mundo a través de un globo terráqueo virtual y ver imágenes de satélite, mapas, relieve y edificios 3D, entre otras cosas. Gracias al detallado contenido geográfico de Google Earth, se puede experimentar una visión más realista del mundo, convirtiéndose en una herramienta más en el desarrollo de tecnología, tiene versiones empresariales además de una licencia de uso libre.

1.12.1. Uso de Google Earth

La figura 1.12 describe algunas de las funciones disponibles en la ventana principal de Google Earth.

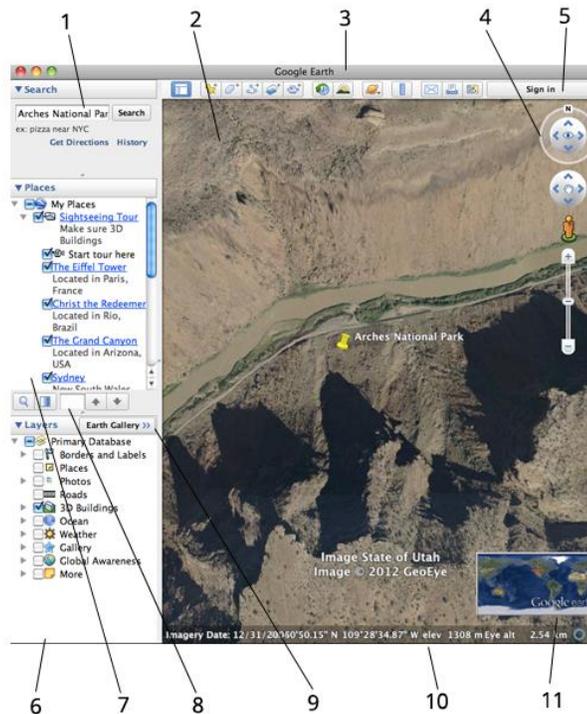


Fig. 1.12. Funciones Google Earth.

²³ http://www.google.es/intl/es_es/earth/learn/

1. **Panel de búsqueda:** permite buscar sitios e indicaciones, así como administrar los resultados de búsqueda.
2. **Visor 3D:** esta ventana permite ver el planeta y su relieve.
3. **Botones de la barra de herramientas:** Descritos en la figura 1.13.
4. **Controles de navegación:** permiten acercar o alejar la imagen, así como observarla y desplazarse por ella.
5. **Opción para iniciar sesión con Google.**
6. **Panel "Capas":** este panel permite mostrar puntos de interés.
7. **Panel "Lugares":** este panel permite localizar, guardar, organizar y volver a visitar marcas de posición.
8. **Cuadro de búsqueda del panel "Lugares".**
9. **Galería de Google Earth.**
10. **Barra de estado:** en ella se ven las coordenadas, la elevación, la fecha de las imágenes y su estado de streaming.
11. **Vista general del mapa:** este mapa permite obtener otra perspectiva de la Tierra.

A continuación la figura 1.13 muestra los íconos incluidos en la barra de herramientas.

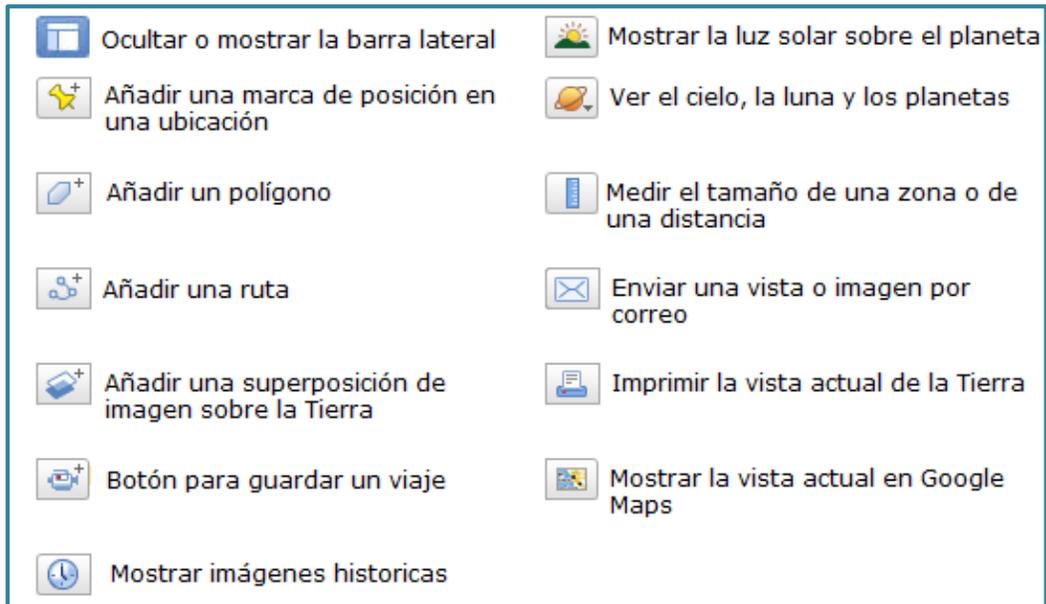


Fig. 1.13. Íconos de la barra de herramientas.

1.13. Nokia Maps²⁴

Con el uso del Software Nokia Maps, en un dispositivo Nokia modelo 500 mostrado en la figura 1.14, con características detalladas a continuación se realizara el posicionamiento de los nodos fijos.



Fig. 1.14. Nokia 500.

²⁴ http://nds1.nokia.com/phones/files/guides/Nokia_500_Nokia_Belle_UG_es.pdf

1.13.1. Características mapas

La figura 1.15, muestra la aplicación Nokia Maps y con sus elementos más relevantes.

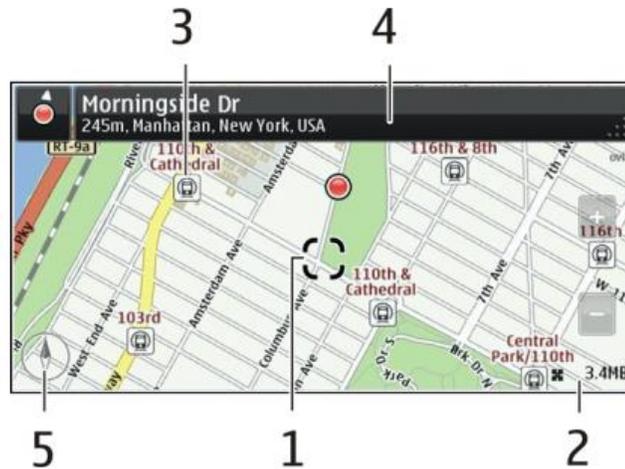


Fig.1.15. Nokia Maps

1. Ubicación seleccionada
2. Área de indicador
3. Punto de interés (por ejemplo, una estación ferroviaria o un museo)
4. Área de información
5. Brújula

1.14. LabView²⁵

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. Permite diseñar interfaces de usuario mediante una consola interactiva basada en software, se puede diseñar

²⁵ <http://repositorio.espe.edu.ec/handle/21000/6443>

especificando su sistema funcional, su diagrama de bloques o una notación de diseño de ingeniería.

Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software. Da la posibilidad a los usuarios de crear soluciones completas y complejas.

Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos. El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible. Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

1.14.1. Funcionamiento

Los programas desarrollados mediante LabVIEW se denominan Instrumentos Virtuales (VIs), porque su apariencia y funcionamiento imitan los de un instrumento real. Sin embargo son análogos a las funciones creadas con los lenguajes de programación convencionales. Los VIs tienen una parte interactiva con el usuario y otra parte de código fuente, además aceptan parámetros procedentes de otros VIs.

1.14.2. Entorno

a. Panel Frontal

Se trata de la interfaz gráfica del VI con el usuario, esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc., como se indica en la figura 1.16.

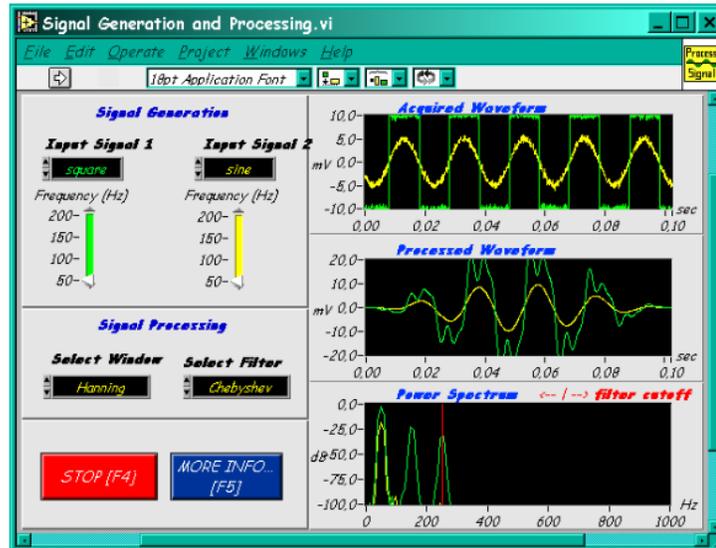


Fig. 1.16. Ejemplo Panel Frontal.²⁶

b. Diagrama de Bloques

El diagrama de bloques constituye el código fuente del VI. En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizar cualquier procesado de las entradas y salidas que se crearon en el panel frontal.

El diagrama de bloques incluye funciones y estructuras integradas en las librerías que incorpora LabVIEW. En el lenguaje G las funciones y las estructuras son nodos elementales. Son análogas a los operadores o librerías de funciones de los lenguajes convencionales.

Los controles e indicadores que se colocaron previamente en el Panel Frontal, se materializan en el diagrama de bloques mediante los terminales. La fig. 1.17, presenta un ejemplo de lo recién citado:

²⁶ http://www.gte.us.es/ASIGN/IE_4T/Tutorial%20de%20Labview.pdf

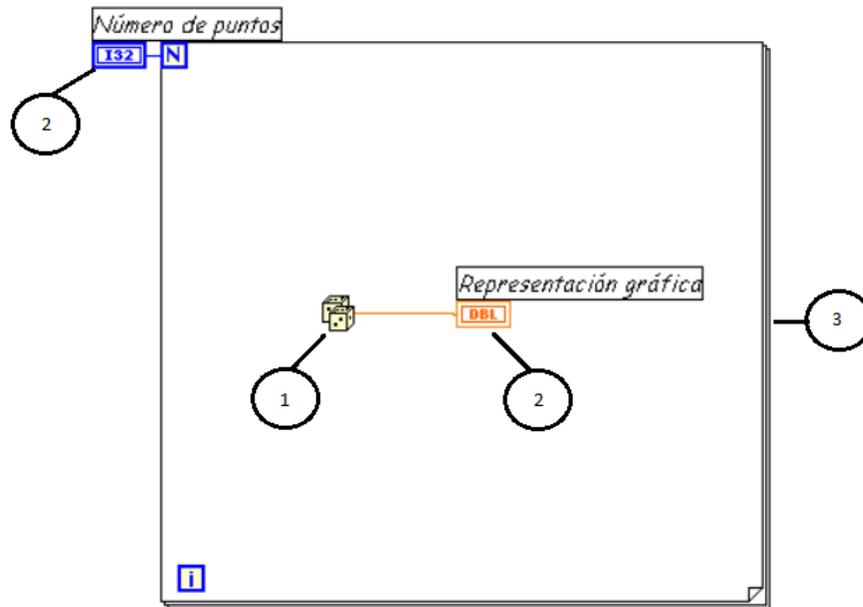


Fig. 1.17. Ejemplo Diagrama de Bloques.²⁷

1. Función.
2. Terminales (control e indicador).
3. Estructura.

El diagrama de bloques se construye conectando los distintos objetos entre sí, como si de un circuito se tratara. Los cables unen terminales de entrada y salida con los objetos correspondientes, y por ellos fluyen los datos.

LabVIEW posee una extensa biblioteca de funciones, entre ellas, aritméticas, comparaciones, conversiones, funciones de entrada/salida, de análisis, etc.

Las estructuras, similares a las declaraciones causales y a los bucles en lenguajes convencionales, ejecutan el código que contienen de forma condicional o repetitiva (bucle for, while, case,...).

²⁷ http://www.gte.us.es/ASIGN/IE_4T/Tutorial%20de%20Labview.pdf

Los cables son las trayectorias que siguen los datos desde su origen hasta su destino, ya sea una función, una estructura, un terminal, etc. Cada cable tiene un color o un estilo diferente, lo que diferencia unos tipos de datos de otros.

c. Paletas

Las paletas de LabVIEW proporcionan las herramientas que se requieren para crear y modificar tanto el panel frontal como el diagrama de bloques. Existen las siguientes paletas:

c.1. Paleta de herramientas (Tools palette)

Se emplea tanto en el panel frontal como en el diagrama de bloques. Contiene las herramientas necesarias para editar y depurar los objetos tanto del panel frontal como del diagrama de bloques.



Fig. 1.18. Paleta Herramientas.

c.2. Paleta de controles (Controls palette)

Se utiliza únicamente en el panel frontal. Contiene todos los controles e indicadores que se emplearán para crear la interfaz del VI con el usuario.

La fig. 1.19, muestra la paleta de controles.

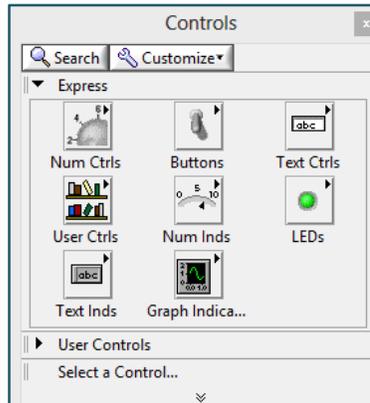


Fig. 1.19. Paleta de controles.

c.3. Paleta de funciones (functions palette)

Se emplea en el diseño del diagrama de bloques. La paleta de funciones contiene todos los objetos que se emplean en la implementación del programa del VI, ya sean funciones aritméticas, de entrada/salida de señales, entrada/salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa. La fig. 1.20, muestra la paleta de funciones.

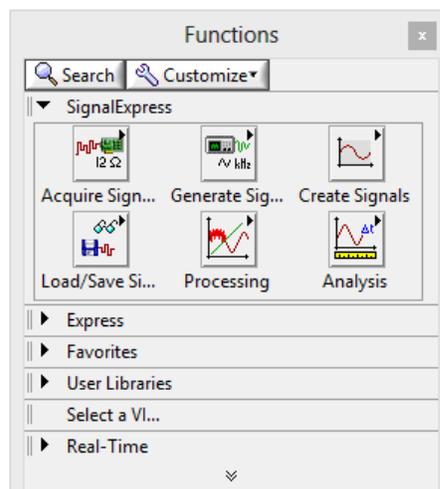


Fig. 1.20. Paleta de Funciones.

CAPITULO 2

DISEÑO DEL SISTEMA

En este capítulo se busca determinar parámetros necesarios para la elaboración de los transmisores y la estación base que serán pilares de la red MESH.

2.1. Distribución física de los transmisores

En la figura 2.1, se presenta la imagen satelital correspondiente al caserío Santa Lucia, barrio Centro perteneciente al cantón Tisaleo en la Provincia de Tungurahua en el correspondiente mapa se presenta el lugar que albergará la red, por tanto bajo estas condiciones se realizará el correspondiente diseño.



Fig. 2.1. Mapa Satelital Google Maps con la imagen correspondiente al caserío Santa Lucía.

2.2. Diseño de la red Zigbee

2.2.1. Modo API²⁸

Como una forma alternativa al modo de operación transparente el modo API (Application Programming Interface), requiere que una comunicación con el módulo sea hecho a través de una interface estructurada (la comunicación se realiza en tramas, “frames” en un orden definido). El modo API especifica como los comandos, respuestas a comandos y mensajes de estado de los módulos son enviados y recibidos desde el módulo usando la UART.

Los datos dependerán de la clase de frame que se reciba, que se determinará de acuerdo al firmware existente para los xbee.

La figura 2.2 muestra la estructura de un frame para modo API, en donde el byte número uno es el identificador 0x7E, que marca el inicio de la transmisión.²⁹

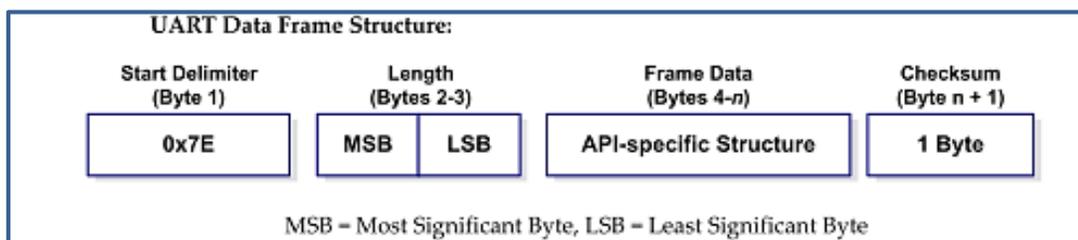


Fig. 2.2. Frame en modo API.

2.2.2. Longitud

El campo de longitud contiene dos bytes que especifican el número de bytes que formarán parte del campo de datos de dicho frame, sin incluir el campo de checksum.

²⁸ http://ftp1.digi.com/support/documentation/90000976_P.pdf

²⁹ http://www.sase.com.ar/2012/files/2012/09/XBeeZB_Workshop.pdf

2.2.3. Campo de datos

El campo de datos como indica la figura 2.3 contiene un identificador API, que se detallaran a continuación.

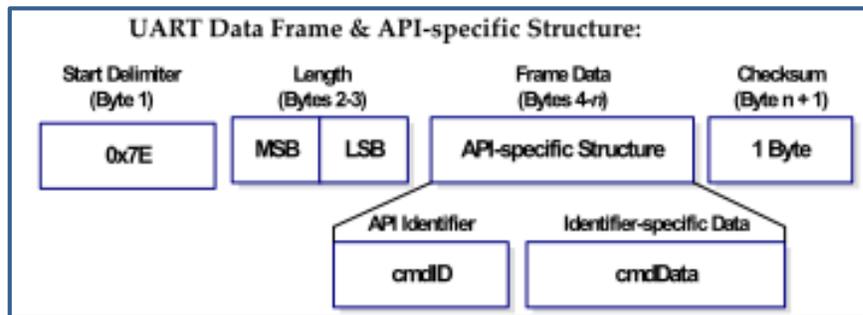


Fig. 2.3. Campo de datos en el API frame.

El campo *API identifier* está especificado por el fabricante de tal suerte que el firmware de cada elemento será escrito con esta información. La tabla 2.1 muestra todos los frames dados por el fabricante.

Tabla 2.1. Identificadores de API frame.

Nombre API frame	API ID
AT Command	0x08
AT Command-Pregunta Valor Parámetro	0x09
Zigbee petición transmisión	0x10
Frame Comando Dirección explicita Zigbee	0x11
Petición de Comando Remoto	0x17
Crear ruta de fuente	0x21
Respuesta a comando remoto	0x88
Estado del modem	0x8A
Estado de transmisión Zigbee	0x8B
Paquete Zigbee recibido (AO=0)	0x90
Indicador Rx explicito Zigbee (AO=1)	0x91
Indicador Rx de muestras de datos IO	0x92
Indicador de lectura de sensor IO(AO=0)	0x94
Indicador de identificación de Nodo (AO=0)	0x95
Respuesta a comando remoto	0x97
Estado de actualización de firmware sobre -el-aire	0xA0
Indicador de grabado de ruta	0xA1
Indicador de petición de ruta varios-a-uno	0xA3

2.2.4. Checksum

Para verificar la integridad de los datos, un checksum es calculado y verificado en los datos no perdidos.

Para calcular: Sin incluir los límites de frame y la longitud, sumar todos los bytes, manteniendo solamente los 8 menos significativos bits del resultado y restar de 0xFF.

Para verificar: Sumar todos los bytes (incluyendo el checksum, pero no el delimitador y longitud). Si el checksum es correcto, la suma será igual a 0xFF.

2.2.5. API Frames de mayor uso³⁰

AT Command

Frame type: 0x08

Usado para preguntar o configurar los parámetros en el módulo local.

La tabla 2.2 muestra el AT command frame.

Tabla 2.2. AT Command frame

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x04	
	Tipo Frame	3	0x08	
	Frame ID	4	0x52 (R)	
	Comando AT	5	0x4E (N)	Nombre Comando- dos caracteres ASCII que identifican el comando AT
		6	0x4A (J)	
	Valor parámetro (opcional)			
Checksum	7	0x0D		0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte

³⁰ http://ftp1.digi.com/support/documentation/90000976_P.pdf

2.2.6. AT Command – Pregunta por el valor del parámetro

Frame type 0x09

Este frame permite que los valores sean preguntados o configurados en el módulo. En contraste con el frame tipo 0x08 los nuevos valores de los parámetros son preguntados, y no configurados ya sea con el “AT Command” (0x08) o el AC (Apply Changes) command es usado. La tabla 2.3 muestra el frame type 0x09.

Tabla 2.3. AT Command – Pregunta por el valor del parámetro frame.

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x05	
	Tipo Frame	3	0x09	
	Frame ID	4	0X01	
	Comando AT	5	0x42 (B)	Nombre Comando- dos caracteres ASCII que identifican el comando AT
		6	0x44 (D)	
	Valor parámetro (ATDB7-115200 baud)	7	0X07	Si está presente indica el valor del parámetro a escribirse en el registro
Checksum	8	0x68	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.7. ZigBee Transmit Request

Frame type 0x10

Un *Transmit Request API frame* causa que el módulo envíe datos en forma de un paquete RF a la dirección especificada. La tabla 2.4 muestra un *Transmit Request API frame*.

Tabla 2.4. Transmit Request API frame.

	Campo	Offset	Ejemplo	Descripcion
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x16	
	Tipo Frame	3	0x10	
	Frame ID	4	0x01	
	Dirección destino 64 bits	5	0x00	
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x0A	
		11	0x01	
	Dirección de red 16 bits	12	0x27	
		13	0xFF	
	Radio Broadcast	14	0xFE	
		15	0x00	Fija el número de saltos que deben existir en una transmisión broadcast, si es cero el valor será el máximo de saltos posibles
Opciones	16	0x00	Las opciones incluyen 0x01- ACK deshabilitado 0x20-habliada encriptación APS 0x40-Usar tiempo de espera extendido	
Datos RF	17	0x54	Datos hacia el dispositivo de destino	
	18	0x78		
	19	0x44		
	20	0x61		
	21	0x74		
	22	0x61		
	23	0x30		
24	0x41			
Checksum	25	0x13	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.8. Explicit Addressing ZigBee Command Frame

Frame Type: 0x11

Tabla 2.5. Explicit Addressing ZigBee Command Frame.

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x1A	
	Tipo Frame	3	0x11	
	Frame ID	4	0X01	
	Dirección destino 64 bits	5	0x00	
		6	0x00	
		7	0x00	
		8	0x00	
		9	0x00	
		10	0x00	
		11	0x00	
	Dirección de red 16 bits	12	0x00	
		13	0xFF	
		14	0xFE	
	Punto final de fuente	15	0xA0	
	Punto final de destino	16	0xA1	
	ID clúster	17	0x15	
		18	0x54	
	ID perfil	19	0xC1	
		20	0x05	
	Radio Broadcast	21	0x00	Fija el número de saltos que deben existir en una transmisión broadcast, si es cero el valor será el máximo de saltos posibles
	Opciones	22	0x00	Las opciones incluyen 0x01- ACK deshabilitado 0x20-habliada encriptacion APS 0x40-Usar tiempo de espera extendido
	Carga de Datos	23	0x54	Datos hacia el dispositivo de destino
		24	0x78	
		25	0x44	
		26	0x61	
		27	0x74	
		28	0x61	
Checksum	29	0x3A	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.9. Remote AT Command Request

Frame Type: 0x17

Usado para preguntar o configurar parámetros en un dispositivo remoto. La tabla 2.6 muestra un frame tipo Remote AT Command Request.

Tabla 2.6. Remote AT Command Request frame.

	Campo	Offset	Ejemplo	Descripcion
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x1A	
	Tipo Frame	3	0x17	
	Frame ID	4	0X01	
	Dirección destino 64 bits	5	0x00	
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x40	
		11	0x11	
	Dirección de red 16 bits	12	0x22	
		13	0xFF	
		14	0xFE	
	Opciones de comando remoto	15	0x02 (aplicar cambios)	Las opciones incluyen 0x01- ACK deshabilitado 0x20-habliada encriptacion APS 0x40-Usar tiempo de espera extendido
	Comando AT	16	0x42(B)	
17		0x48(H)		
Parámetro comando	18	0x01		
Checksum	19	0xF5	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.10. Create Source Route

Frame Type: 0x21

Este frame crea una ruta en el módulo. Una ruta especifica una ruta completa que un paquete debe atravesar para llegar desde la fuente hacia el destino. El ruteo debe ser usado en el ruteo varios-a-uno para mejores resultados. La tabla 2.7 muestra un *Create Source Route* frame.

Tabla 2.7. Create Source Route frame.

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x14	
	Tipo Frame	3	0x21	
	Frame ID	4	0X00	
	Dirección destino 64 bits	5	0x00	
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x40	
		11	0x11	
	Dirección de red 16 bits	12	0x22	
		13	0x33	
		14	0x44	
		15	0x00	
	Opciones de comando ruta	15	0x00	Configurado a cero
	Numero de Direcciones	16	0x03	Numero de direcciones en la ruta(excluyendo fuente y destino)
	Dirección 1	17	0xEE	Vecino o destino
		18	0xFF	
	Dirección 2	19	0xCC	Dirección del salto inmediato
		20	0xDD	
	Dirección 3	21	0xAA	Vecino o fuente
22		0xBB		
Checksum	23	0xF5	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.11. AT Command Response

Frame Type: 0x88

En respuesta a un mensaje AT command, el módulo enviará una mensaje *AT command response*. Algunos comandos enviarán múltiples frames.

Tabla 2.8. AT Command Response frame.

	Campo	Offset	Ejemplo	Descripción	
API Packet	Delimitador de Inicio	0	0x7E		
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum	
		2	0x05		
	Tipo Frame	3	0x88		
	Frame ID	4	0x01		
	Comando AT	5	0x42 (B)	Nombre Comando- dos caracteres ASCII que identifican el comando AT	
		6	0x44 (D)		
	Estado del Comando		7	0x00	0= OK
					1= Error
	Datos del Comando				2= Comando no valido
3= Parámetro no valido					
Checksum		8	0xF0	4= Falla de transmisión	
				0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.12. Modem Status

Frame Type: (0x8A)

Los mensajes de estado son enviados desde el módulo. La tabla 2.9 muestra un *Modem status frame*.

Tabla 2.9. Modem status frame.

	Campo	Offset	Ejemplo	Descripción	
API Packet	Delimitador de Inicio	0	0x7E		
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum	
		2	0x02		
	Tipo Frame	3	0x8A		
	Estado		4	0x06	0= Hardware Reset
					1=Watchdog timer Reset
Checksum		5	0x68	2= Joined Network(Routers and End devices)	
				3= Disassociated	
				6= Coordinador Started	
				7= Seguridad Actualizada	
				0x0D= Superado voltaje de alimentación(PRO S2B solo)	
				0x11= Cambio la configuración del módulo	
				0x80+= Falla de la pila	
				0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.13. ZigBee Transmit Status

Frame Type: 0x8B

Cuando una petición de transmisión es completada, el módulo envía un mensaje de estado de transmisión. Este mensaje indicara si el paquete fue transmitido exitosamente o si existió una falla. La tabla 2.10 muestra un *ZigBee Transmit Status* frame.

Tabla 2.10. ZigBee Transmit Status frame.

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x07	
	Tipo Frame	3	0x8B	
	Frame ID	4	0x01	
	Dirección de destino de 16 bits	5	0x7D	
		6	0x84	
	Conteo de transmisiones	7	0x00	
	Estado de entrega		0x00	0x00= Exitoso
	Estado de Descubrimiento		0x01	
Checksum	8	0x71	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.14. ZigBee Receive Packet

Frame Type: (0x90)

Cuando el módulo recibe un paquete RF, este es enviado a la UART usando este tipo de mensaje. La tabla 2.11 muestra un *ZigBee Receive Packet* frame.

Tabla 2.11. ZigBee Receive Packet frame.

	Campo	Offset	Ejemplo	Descripcion
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x14	
	Tipo Frame	3	0x90	
	Direccion destino 64 bits	4	0x00	
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		11	0xAA	
	Direccion de red 16 bits	12	0x7D	
		13	0x84	
	Opciones de recepcion	14	0x01	0x00-Paquete conocido 0x01-Paquete broadcast 0x20-Paquete encriptado APS 0x40-paquete enviado desde un end device
	15	0x52		
	16	0x78		
	17	0x44		
Datos Recibidos	18	0x61	Datos RF recibidos	
	19	0x74		
	20	0x61		
Checksum	21	0x0D	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.15. ZigBee Explicit Rx Indicator

Frame Type: 0x91

Cuando el modem recibe un paquete RF es enviado de la UART usando este tipo de mensaje (cuando AO=1). La tabla 2.12 muestra un *ZigBee Explicit Rx Indicator* frame.

Tabla 2.12. ZigBee Explicit Rx Indicator frame.

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x18	
	Tipo Frame	3	0x91	
	Dirección destino 64 bits	4	0x00	
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
	Dirección de red 16 bits	11	0xAA	
		12	0x7D	
	Dirección de red 16 bits	13	0x84	
		14	0xE0	
	Punto final de fuente	14	0xE0	
	Punto final de destino	15	0xE0	
	ID clúster	16	0x22	
		17	0x11	
	ID perfil	18	0xC1	
		19	0x05	
	Opciones de Recepción	20	0x02	0x00-Paquete conocido 0x01-Paquete broadcast 0x20-Paquete encriptado APS 0x40-paquete enviado desde un end device
	Datos recibidos	21	0x52	Datos RF recibidos
		22	0x78	
		23	0x44	
		24	0x61	
		25	0x74	
26		0x61		
Checksum	27	0x52	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.16. ZigBee IO Data Sample Rx Indicator

Frame Type: 0x92

Cuando el módulo recibe una muestra IO desde un dispositivo remoto, envía la muestra a la UART usando este tipo de frame (cuando AO=0). La tabla 2.13 muestra un ZigBee IO Data Sample Rx Indicator frame.

Tabla 2.13. ZigBee IO Data Sample Rx Indicator frame.

	Campo	Offset	Ejemplo	Descripcion
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x14	
	Tipo Frame	3	0x92	
	Dirección destino 64 bits	4	0X00	
		5	0x00	
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x52	
		11	0x8B	
	Dirección de red 16 bits	12	0xAA	
		13	0x7D	
	Opciones de Recepción	14	0x84	0x01-Paquete conocido 0x02-Paquete broadcast
		15	0x01	
	Numero de muestras	16	0x01	
	Mascara digital de canal	17	0x00	
		18	0x1C	
	Mascara de canal análogo	19	0x02	
	Muestras digitales	20	0x00	
		21	0x14	
	Muestra análoga	22	0x02	
		23	0x25	
Checksum	24	0xF5	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.17. XBee Sensor Read Indicator

Frame Type: 0x94

Cuando el módulo recibe una muestra desde el sensor(desde un adaptador para sensor Digi 1-cable), es enviado fuera de la UART usando este tipo de mensaje (cuando AO=0). La tabla 2.14 muestra un XBee Sensor Read Indicator frame.

Tabla 2.14. XBee Sensor Read Indicator frame.

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Número de Bytes entre la longitud y el CheckSum
		2	0x17	
	Tipo Frame	3	0x94	
	Dirección destino 64 bits	4	0X00	
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		11	0xAA	
	Dirección de red 16 bits	12	0xDD	
		13	0x6C	
	Opciones de Recepción	14	0x01	0x01-Paquete conocido 0x02-Paquete broadcast
	Sensores 1-Cable	15	0x03	
	Valores A/D	16	0x00	
		17	0x02	
		18	0x09	
		19	0xCE	
		20	0x00	
		21	0xEA	
		22	0x00	
		23	0x52	
	Lectura Temperatura	24	0x01	
		25	0x6A	
Checksum	26	0xF5	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.18. Node Identification Indicator

Frame Type: 0x95

La tabla 2.15 muestra un Node Identification Indicator frame.

Tabla 2.15. Node Identification Indicator frame.

	Campo	Offset	Ejemplo	Descripcion
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x20	
	Tipo Frame	3	0x95	
	Direccion destino 64 bits	4	0X00	
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		11	0xAA	
	Direccion de red 16 bits	12	0x7D	
		13	0x84	
	Opciones de Recepción	14	0x01	0x01-Paquete conocido 0x02-Paquete broadcast
	Direccion de fuente de 16 bits	15	0x03	
		16	0x00	
	Direccion de red de 64 bits	17	0X00	
		18	0x13	
		19	0xA2	
		20	0x00	
		21	0x40	
		22	0x52	
		23	0x2B	
		24	0xAA	
	Texto NI	25	0x20	
		26	0x00	
	Direccion 16 bits del padre	27	0xFF	
		28	0xFE	
	Tipo de Dispositivo	29	0x01	
	Evento fuente	30	0x01	
	ID perfil Digi	31	0xC1	
		32	0x05	
	ID Manufactura	33	0x10	
		34	0x1E	
Checksum	35	0x1B	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.2.19. Remote Command Response

Frame Type: 0x97

Si un módulo recibe una respuesta remota de datos RF a un comando en respuesta a una demanda por comando AT, el módulo enviara un mensaje respuesta a la UART. La tabla 2.16 muestra un Remote Command Response frame.

Tabla 2.16. Muestra un Remote Command Response frame.

	Campo	Offset	Ejemplo	Descripción
API Packet	Delimitador de Inicio	0	0x7E	
	Longitud	1	0x00	Numero de Bytes entre la longitud y el CheckSum
		2	0x14	
	Tipo frame	3	0x97	
	ID frame	4	0X55	
	Direccion destino 64 bits	5	0x00	
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x52	
		11	0x8B	
		12	0xAA	
	Direccion de red 16 bits	13	0x7D	
		14	0x84	
	Comando AT	15	0x53	
		16	0x4C	
	Estado Comando	17	0x00	0= OK 1= Error 2= Comando no valido 3= Parametro no valido 4= Transmision no valida
	Datos Comando	19	0x02	
		20	0x00	
		21	0x14	
		22	0x02	
Checksum	24	0xF5	0xFF-la suma de 8 bits de los bytes offset 3 hasta este byte	

2.3. Técnicas De Localización³¹

Existe una gran cantidad de técnicas de localización que intentan resolver el problema de la localización sin el uso de sensores específicos:

- *Localización con balizas:* esta técnica permite localizar al objeto en un entorno restringido mediante el emplazamiento específico de un determinado número de balizas con posiciones conocidas. Para estimar la posición se puede utilizar la triangulación basándose en el ángulo con que se ven las balizas y la trilateración basándose en la distancia a las balizas.
- *Los filtros de Kalman:* son otra técnica que trata de estimar recursiva y periódicamente la posición de mínima varianza fusionando información parcial e indirecta sobre localización. Su principal limitación es que es una técnica unimodal y exclusivamente gaussiana. Esta técnica no soporta bien el ruido de lecturas sensoriales ni entornos dinámicos y no es capaz de manejar múltiples hipótesis.
- *El scan matching:* es otra técnica que utiliza mapas locales para compararlos con lecturas sensoriales alineando estas lecturas con los diferentes mapas en posiciones cercanas a la que creemos que está el dispositivo, necesitando para ello una estimación de la posición inicial del dispositivo representada como una distribución gaussiana que se va actualizando con las lecturas sensoriales.
- *Localización probabilística:* es adecuada para interiores ya que incorpora incertidumbre de acciones y observaciones que se acoplan a la incertidumbre que muestran los sensores. Este tipo de localización consiste en determinar la probabilidad de que el objeto se encuentre en una determinada posición a través de sus lecturas sensoriales y

³¹ <http://dspace.esPOCH.edu.ec/bitstream/123456789/374/1/38T00183.pdf>

movimientos a lo largo del tiempo. A cada posible posición se le asocia una probabilidad reflejando la verosimilitud de ser la posición actual del objeto. Esta probabilidad se va actualizando con la incorporación de nuevas lecturas y movimientos del objeto. Estas técnicas nos permiten localizar al objeto aun desconociendo su posición inicial permitiendo representar situaciones ambiguas que se irán desambiguando posteriormente.

2.3.1. Determinación de la distancia por Índice de Potencia de Señal Recibida (RSSI).³²

Muchos de los algoritmos de localización requieren una distancia para calcular la posición del dispositivo desconocido. Una posibilidad de adquirir una distancia es midiendo la fuerza de señal recibida de la señal de radio entrante. La idea atrás de la RSS es que variando la potencia de transmisión en el dispositivo transmisor (P_{TX}) directamente afecta la potencia recibida en el dispositivo receptor (P_{RX}). De acuerdo con la ecuación de transmisión en el espacio libre de Friis (Ec. 2.1), la potencia de la señal detectada decrementa cuadráticamente con la distancia para el transmisor:

$$P_{RX} = P_{TX} G_{TX} G_{RX} \left(\frac{\lambda}{4\pi d} \right)^2 \quad \text{Ec. 2.1.}$$

Dónde:

P_{TX} = Potencia de transmisión.

P_{RX} = Potencia recibida en receptor.

G_{TX} = Ganancia de transmisión.

³² <http://www.zigbe.net/archivos/424>

G_{RX} =Potencia de recepción.

λ = Longitud de onda.

d = distancia entre transmisor y receptor.

En dispositivos embebidos, la potencia recibida es convertida a un índice de potencia de señal recibida (RSSI) definida como la proporción del potencia recibida con respecto a la potencia de referencia cuyo valor es $P_{ref} = 1\text{mW}$.

$$RSSI = 10 * \log \frac{P_{RX}}{P_{ref}} [dBm] \quad \text{Ec. 2.2}$$

Dónde:

P_{RX} = Potencia de transmisión.

$RSSI$ =índice de potencia de señal recibida

P_{Ref} =Potencia de referencia -1mW

Un incremento en la potencia recibida incrementa el índice RSSI. La Figura 2.4 y 2.5 ilustra la relación entre la potencia recibida y el RSSI, graficando la relación RSSI y la distancia el gráfico es simétrico con relación a la abscisa; la distancia d es inversamente proporcional al RSSI.

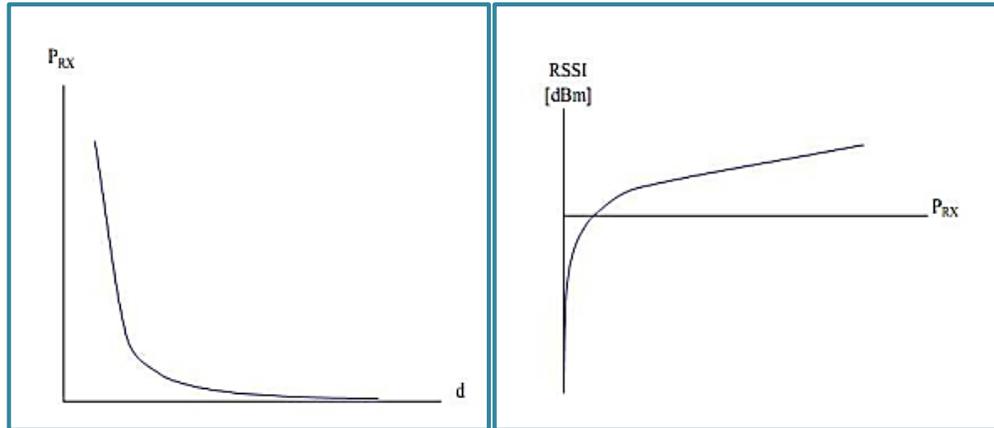


Fig. 2.4. Potencia versus distancia. **Fig. 2.5.** Potencia recibida como RSSI

En escenarios prácticos la distribución ideal no es aplicable, porque existen varios tipos de interferencia como por ejemplo:

- Reflexión de objetos metálicos.
- Superposición de campos electro-magnéticos.
- Límites de difracción.
- Refracción por medios con diferente velocidad de propagación.
- Polarización de campos electro-magnéticos.

Estos efectos degradan significativamente la calidad del índice RSSI, por lo tanto en muchas aplicaciones, el índice RSSI tiene una varianza elevada y una entropía baja.

2.3.2. Trilateración.

La trilateración es un método para determinar la posición relativa de objetos usando la geometría de triángulos en una forma parecida a la triangulación.

A diferencia de la triangulación, la cual usa medición de ángulos (junto a por lo menos una distancia conocida) para calcular la localización del sujeto, la trilateración usa las localizaciones conocidas de dos o más puntos de

referencia, y la distancia medida entre el sujeto y cada punto de referencia. Para determinar de forma única y exacta la posición relativa de un punto en un plano 2D usando la trilateración solamente, generalmente son necesarios por lo menos 3 puntos de referencia (por lo menos 4 puntos son necesarios en el plano 3D).

Considerando la fórmula básica para una esfera como se muestra en la Ec. 2.3.

$$d^2 = x^2 + y^2 + z^2 \quad \text{Ec. 2.3.}$$

Para una esfera centrada en el punto (x_a, y_a, z_a) la ecuación se simplifica tal como se muestra en la Ec. 2.4.

$$d^2 = (x - x_a)^2 + (y - y_a)^2 + (z - z_a)^2 \quad \text{Ec. 2.4.}$$

Si asumimos a todos los nodos dentro del mismo plano, consideramos tres nodos de referencia (a, b, y c) que tienen distancias (d_a, d_b, d_c) hacia el nodo buscado, entonces.

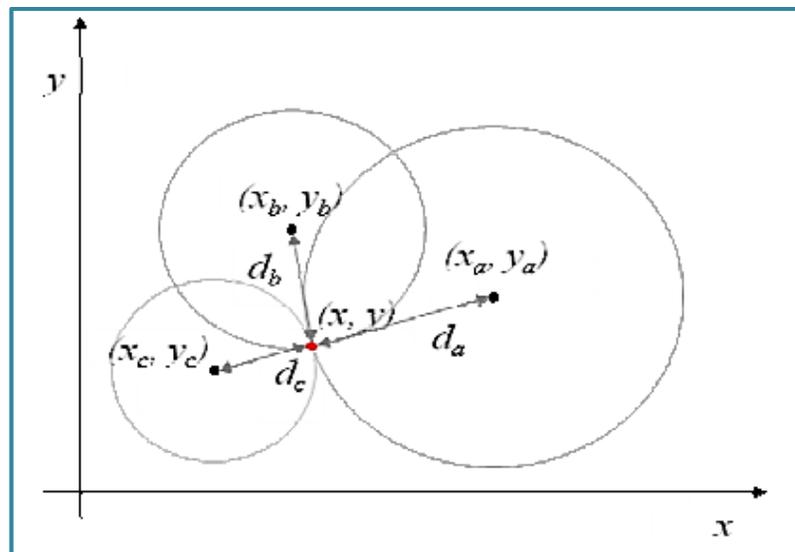


Figura 2.6. Intersección de 3 esferas.

La fórmula para todas las esferas en el plano (círculos) es como se muestra en la Ec. 2.5. y Ec. 2.6. y la Ec. 2.7.

$$\text{Esfera A; } d_a^2 = (x - x_a)^2 + (y - y_a)^2 \quad \text{Ec. 2.5}$$

$$\text{Esfera B; } d_b^2 = (x - x_b)^2 + (y - y_b)^2 \quad \text{Ec. 2.6}$$

$$\text{Esfera C; } d_c^2 = (x - x_c)^2 + (y - y_c)^2 \quad \text{Ec. 2.7}$$

La ecuación 2.5., ecuación 2.6. y ecuación 2.7 son expandidas para convertirse en la Ec. 2.8., Ec 2.9. y Ec 2.10.;

$$d_a^2 = x^2 - 2x \cdot x_a + x_a^2 + y^2 - 2y \cdot y_a + y_a^2 \quad \text{Ec. 2.8}$$

$$d_b^2 = x^2 - 2x \cdot x_b + x_b^2 + y^2 - 2y \cdot y_b + y_b^2 \quad \text{Ec. 2.9}$$

$$d_c^2 = x^2 - 2x \cdot x_c + x_c^2 + y^2 - 2y \cdot y_c + y_c^2 \quad \text{Ec. 2.10}$$

Las tres ecuaciones 2.8., 2.9 y 2.10 son ecuaciones no lineales simultáneas independientes las cuales no pueden ser resueltas matemáticamente; sin embargo, usando el método propuesto por Dixon para obtener la intersección en el plano de dos esferas, restando 2 esferas; la ecuación 2.10 de la ecuación 2.9 obtenemos la siguiente ecuación lineal 2.11.

$$d_b^2 - d_c^2 = 2x(x_c - x_b) + x_b^2 - x_c^2 + 2y(y_c - y_b) + y_b^2 - y_c^2 \quad \text{Ec. 2.11}$$

Y restando la ecuación 2.8 de 2.9, obtenemos la siguiente ecuación lineal 2.12;

$$d_b^2 - d_a^2 = 2x(x_a - x_b) + x_b^2 - x_a^2 + 2y(y_a - y_b) + y_b^2 - y_a^2 \quad \text{Ec. 2.12}$$

Reordenando la ecuación 2.11, para producir una variable llamada V_a , tal como en la ecuación 2.13, tal que,

$$x(x_b - x_c) - y(y_b - y_c) = \frac{(x_c^2 - x_b^2) + (y_c^2 - y_b^2) + (d_b^2 - d_c^2)}{2} = V_a \quad \text{Ec. 2.13}$$

Reordenando la ecuación 2.12, para producir una variable llamada V_b , tal como en 2.14.

$$x(x_b - x_a) - y(y_b - y_a) = \frac{(x_a^2 - x_b^2) + (y_a^2 - y_b^2) + (d_b^2 - d_a^2)}{2} = V_b \quad \text{Ec. 2.14}$$

Resolviendo la ecuación 2.13 y 2.14 para obtener los puntos de intersección 'x' e 'y' de estas dos ecuaciones se obtiene 2.15 para el valor de 'y' y 2.16 para el valor de 'x';

$$y = \frac{V_b(x_b - x_c) - V_a(x_b - x_a)}{(y_a - y_b)(x_b - x_c) - (y_c - y_b)(x_b - x_c)} \quad \text{Ec. 2.15}$$

$$x = \frac{y(y_a - y_b) - V_b}{(x_b - x_c)} \quad \text{Ec. 2.16}$$

Si las ecuaciones no producen ecuaciones reales, significa que las dos esferas no tienen punto de intersección (posiblemente las esferas no se encuentran).

2.4. Configuración de los módulos xbee³³

Para poder utilizar los módulos lo primero es configurarlos, para ello podemos usar el software X-CTU de Digi, con el podemos cambiar el firmware de los

³³ <http://www.tutoelectro.com/tutoriales/otros/x-ctu/>

módulos xbee, también lo podemos usar como terminal serie para enviar y recibir datos por el módulo desde el PC. En la figura 2.1 se muestra la interface del software X-CTU de digi. Para esto, la fig. 2.7 muestra la interface del software;

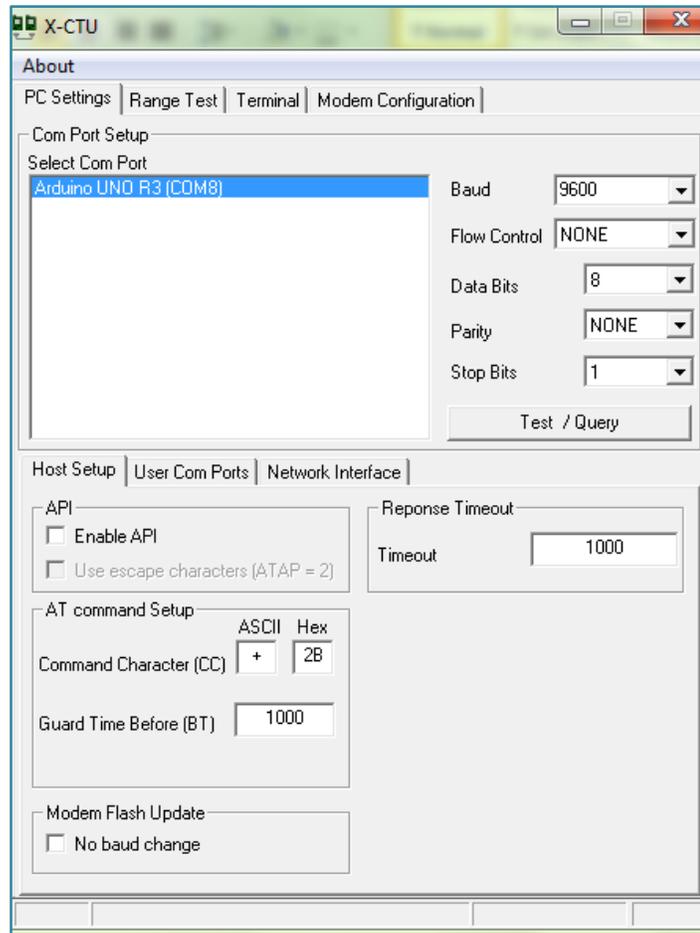


Fig. 2.7. Interface inicial X-CTU.

Ya que se usa como programador la tarjeta Arduino Uno, como única condición para este uso es necesario retirar el micro controlador, de tal suerte que la nomenclatura del puerto serie que vamos a usar viene descrito por el número de puerto, además del nombre de la tarjeta ya expuesta.

Una vez seleccionado dicho puerto, se procede a presionar en el botón *Test / Query* y se muestra en la figura 2.8 la respuesta.

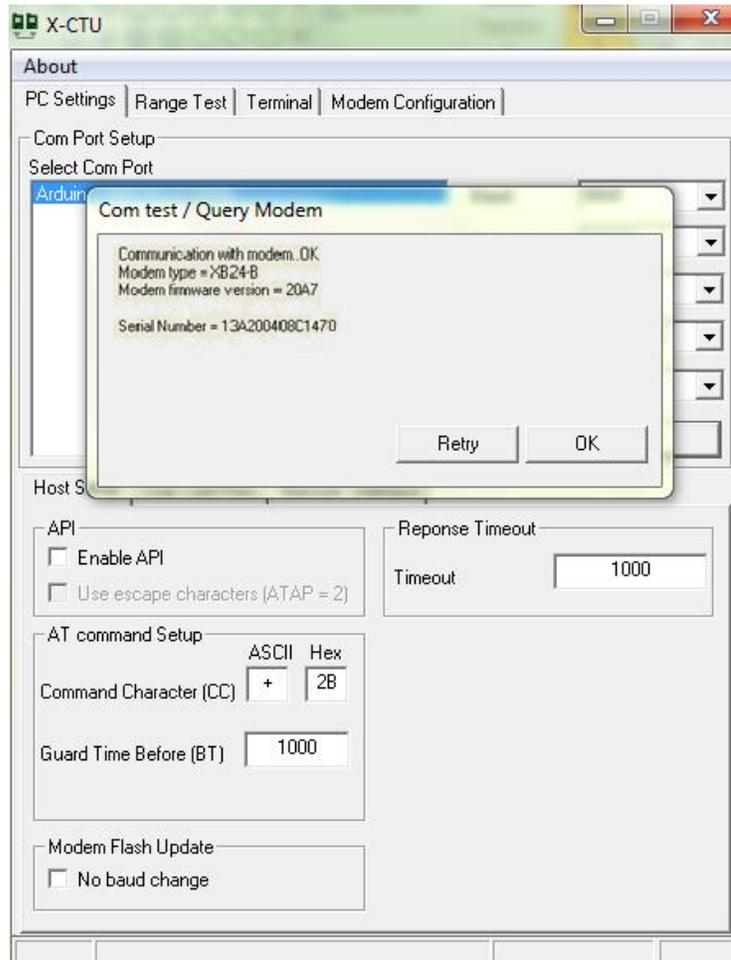


Fig. 2.8. Conexión de X-CTU³⁴ con el módulo xbee.

Después de una conexión exitosa se procederá a programar el firmware en el chip xbee. La figura 2.9 muestra los campos mínimos a modificar en el coordinador para establecer una conexión.

³⁴ <http://www.digi.com/support/productdetail?pid=3352>

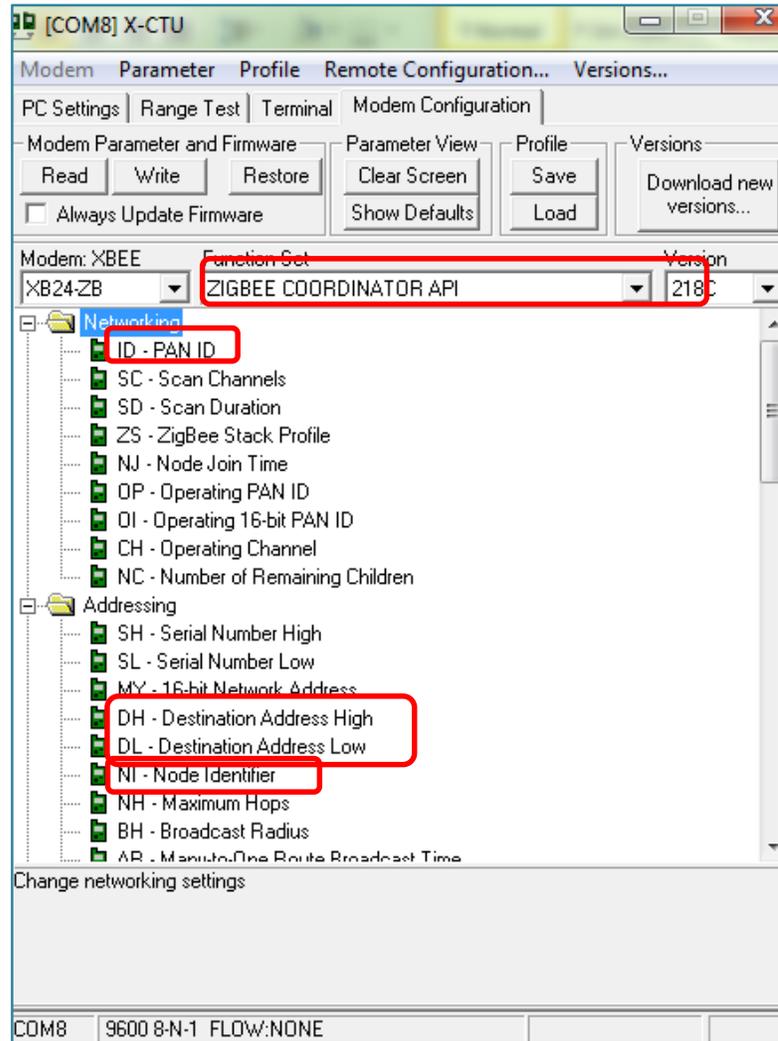


Fig. 2.9. Parámetros a escribir en el firmware del chip xbee.

Una vez que se tiene el coordinador programado se procede con los dispositivos routers, la tabla 2.17, muestra los parámetros comunes a modificarse mientras tanto las figuras 2.10 y 2.11 muestran los parámetros adicionales a modificar solamente en routers.

Tabla 2.17. Parámetros coordinadores y routers.

Tipo	Parámetro	ID	DH	DL	NI
Coordinador		100	0000	0000	“COORDINADOR”
Router		100	SH-destino	SL-destino	“ROUTER”

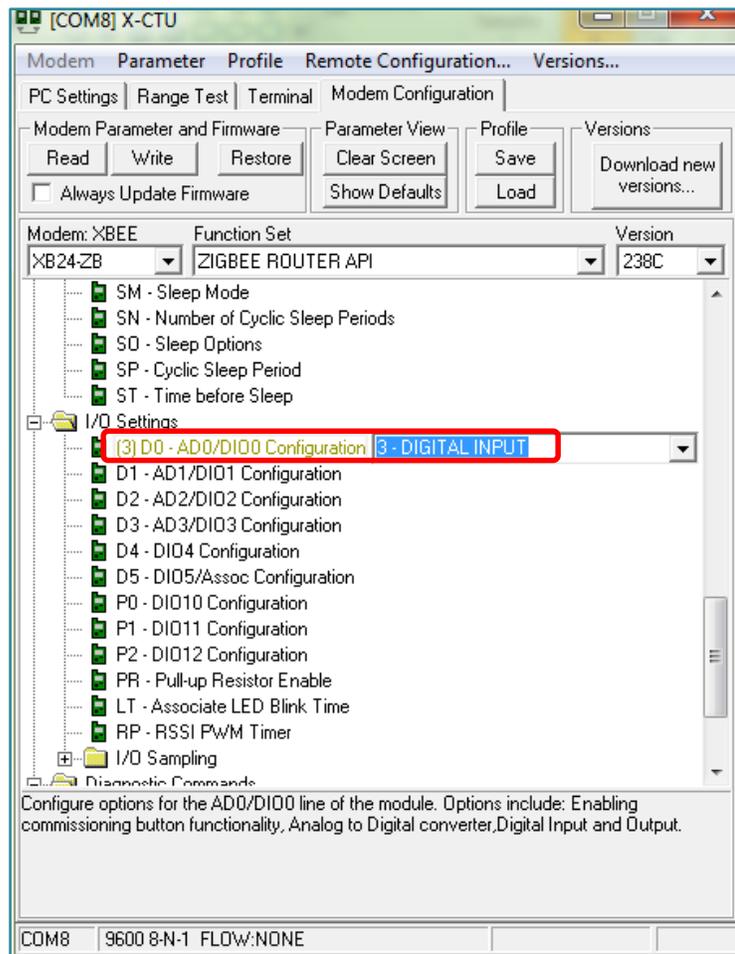


Fig. 2.10. Parámetro D0.

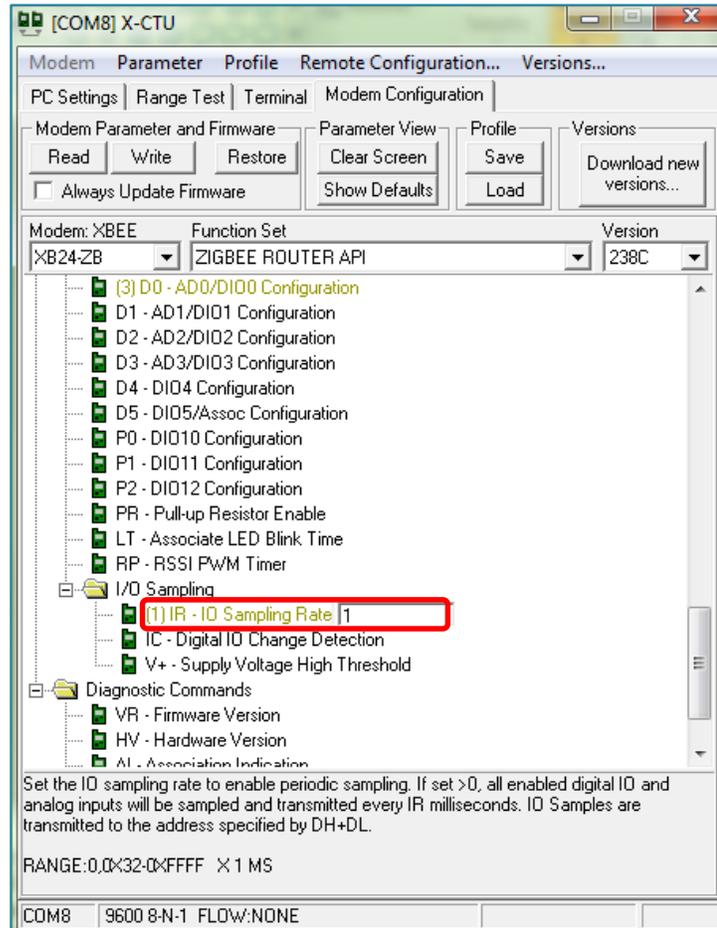


Fig. 2.11. Parámetro IR.

2.5. Análisis con Radio Mobile

Se procede a un análisis y posterior simulación del comportamiento de los dispositivos sobre la superficie del barrio Santa Lucía Centro, para esto con el uso de software Radio Mobile se obtiene los perfiles Topográficos de la zona a estudio.

La fig. 2.12, muestra las alturas comprendidas en la zona a ser analizada.

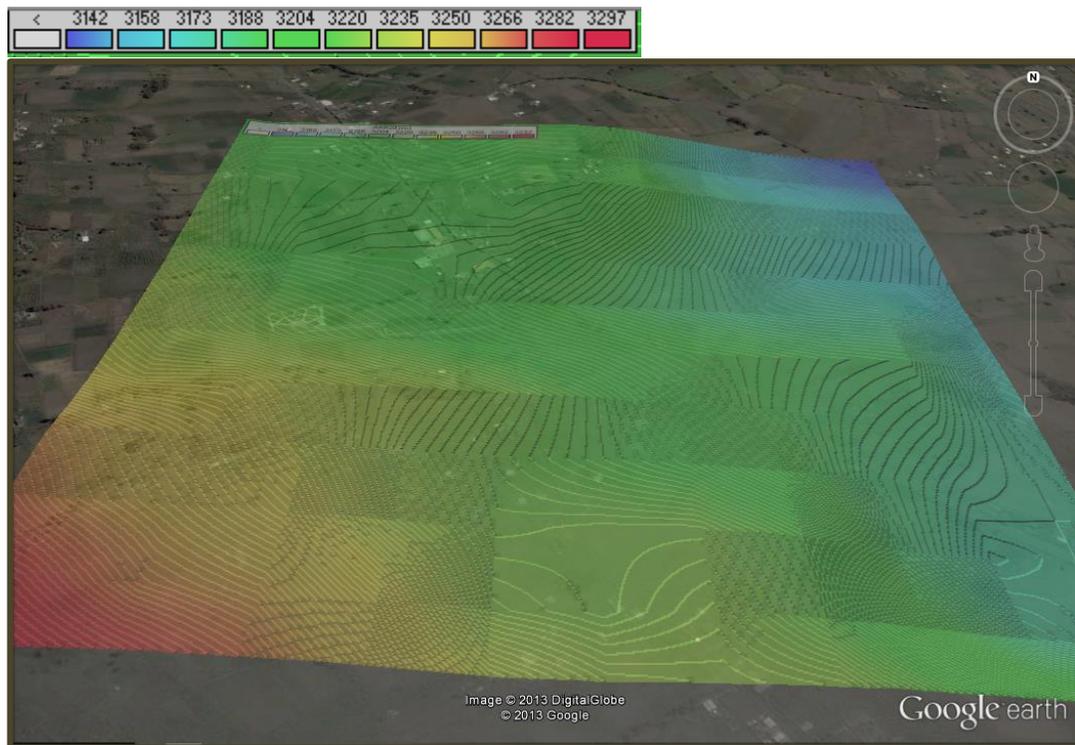


Fig. 2.12. Altitudes sobre el barrio Santa Lucía.

Luego de esto se sitúa en el perfil, los transmisores de acuerdo a las coordenadas previamente obtenidas. Los transmisores se encuentran ubicados con símbolos en los lugares correspondientes a los domicilios.

La fig. 2.13, muestra los transmisores ubicados.

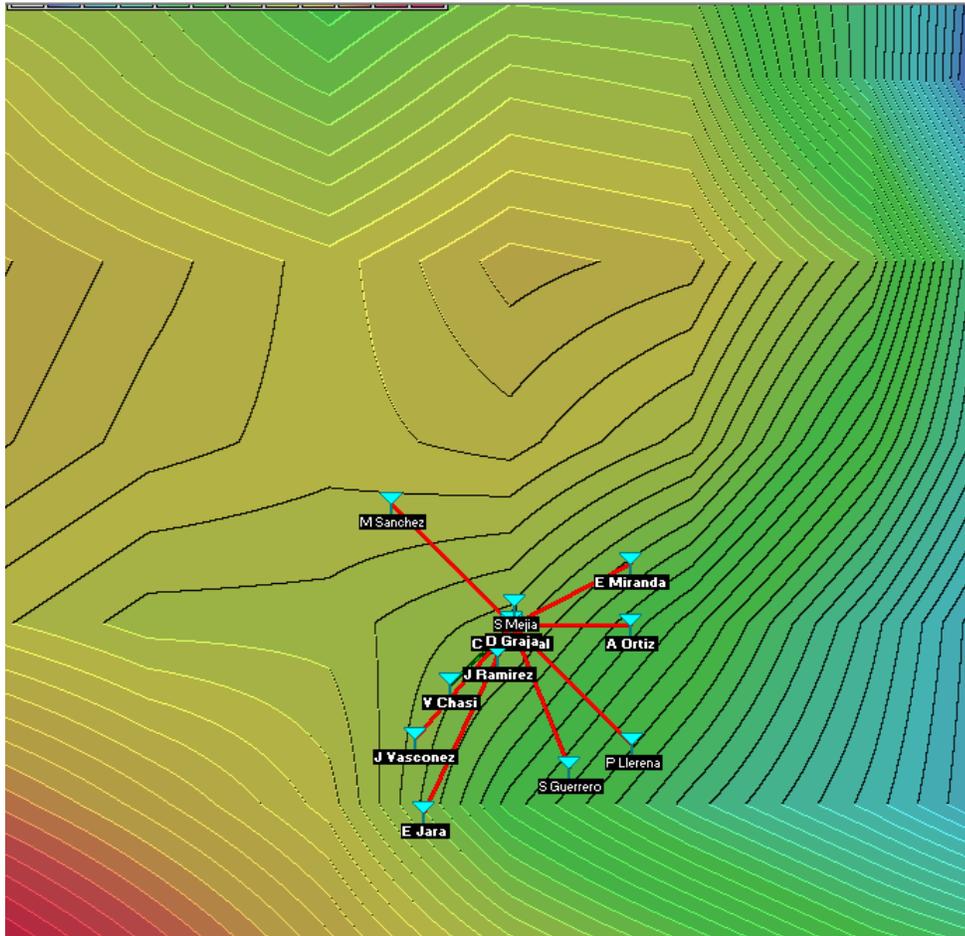


Fig. 2.13. Transmisores ubicados en el mapa.

Luego, se realiza el trazado de una circunferencia por cada transmisor con 100 metros de radio cada una, por tanto, la fig. 2.14, muestra la radiación de los transmisores.

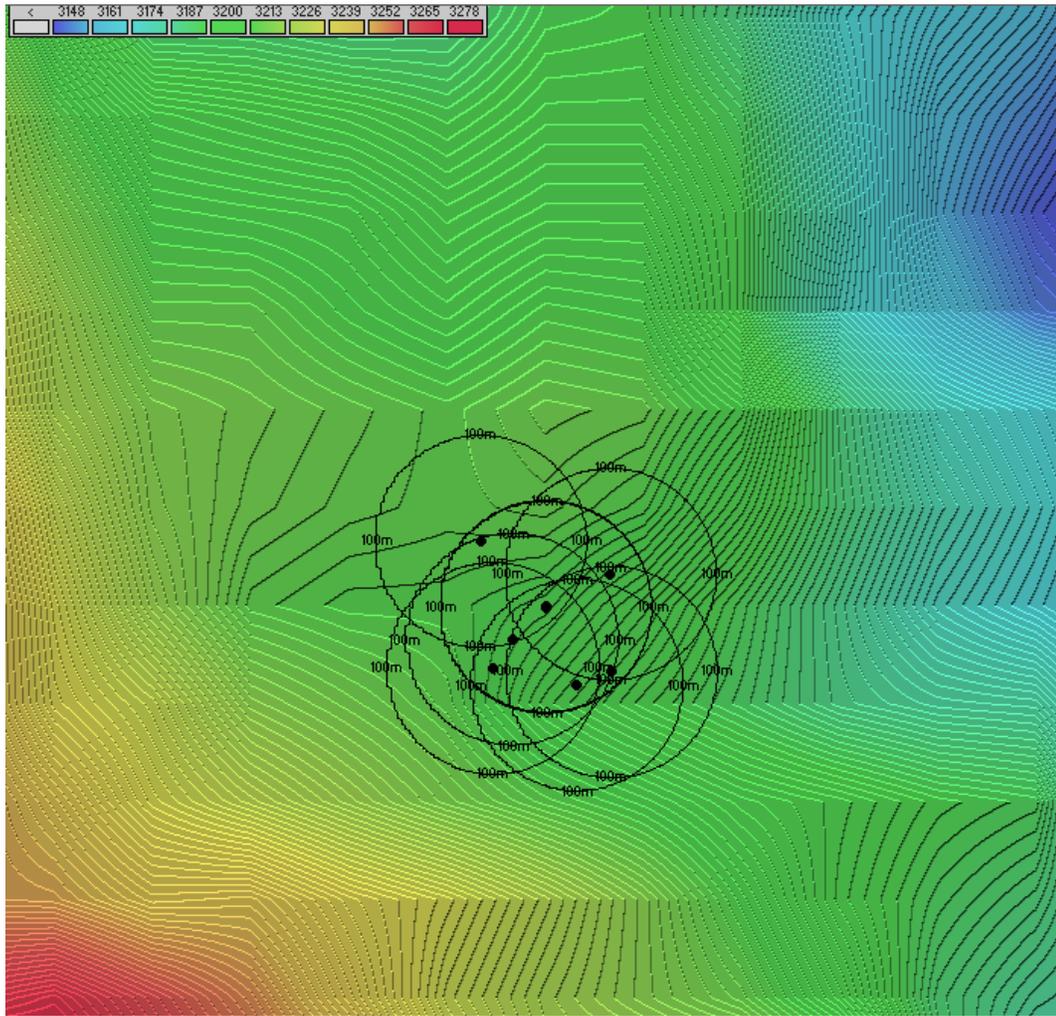


Fig. 2.14. Radiación transmisores.

Como parte del análisis se simula las zonas de cobertura que tendrán los dispositivos, para esto, la fig. 2.15, muestra la zona de cobertura total de la red, además el **ANEXO 10**, detalla el proceso de simulación respecto de todos los transmisores que forman parte de la red.

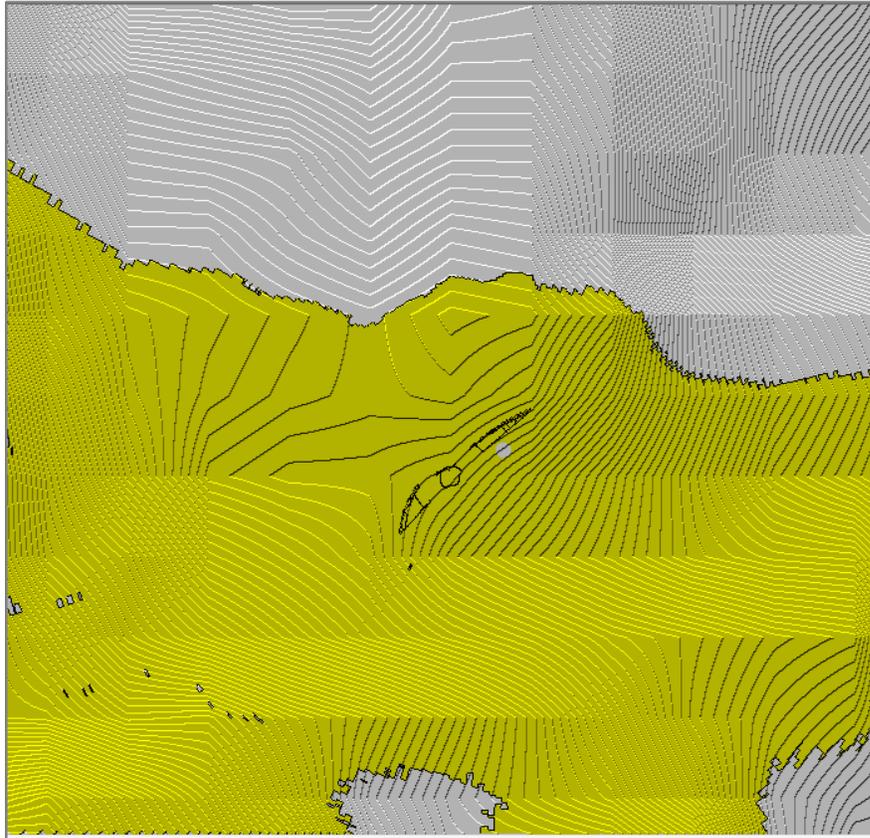


Fig. 2.15. Estación base con transmisores restantes.

2.6. Programación Arduino Mega

La tarjeta Arduino Mega será usada como el elemento principal de control y además será el encargado del flujo de datos hacia y desde la PC, permitiendo de esta forma también tener una interface con mayor información hacia el operador.

Como primer elemento a tener en cuenta es el hecho de que el Xbee Shield está por diseño conectado con el puerto serial 0, siendo compatible ya sea con el modelo Arduino uno, como también con el modelo mega, esto representa posteriormente un problema si necesitamos tener una comunicación tanto con el módulo Xbee, al mismo tiempo que con la PC. Esto se debe a que el puerto serie 0 es el único que está conectado con el chip FTDI conversor a USB, embebido en la tarjeta, que es el encargado de posibilitar el flujo de datos hacia la PC. Por esta razón es necesario modificar físicamente la conexión de la placa Xbee Shield hacia la Arduino, además de realizar por software estas modificaciones.

Para este proceso se puede seleccionar sin distinción de los tres puertos restantes, ya que la Arduino mega nos ofrece en total cuatro. En este caso se optó por usar el puerto *serial1*.

Entonces como primer paso se muestra el proceso para enviar datos desde el puerto 0 hacia el 1, causando esto un flujo de datos entre el coordinador xbee y la PC;

```
if (Serial1.available ()>0) {  
  while (Serial1.available ()>0) {  
    int sensorValue = Serial1.read (); //Se datos desde el puerto  
    int ledPins[100];  
    ledPins[i]=sensorValue; //Se asigna a un vector el frame  
    Serial.write (sensorValue); //conectado al shield y se envía hacia  
  } // el puerto conectado al chip FTDI  
}
```

Mediante las instrucciones indicadas enviamos información entre los puertos físicos de la Arduino mega resolviendo el problema de comunicación simultánea entre la red xbee y la PC.

2.7. Diseño del terminal remoto

Los transmisores xbee necesitan de una alimentación de 3.3 V, que son suministrados por la tarjeta Arduino si es que está conectado mediante el xbee shield, puesto que se tiene como objetivo la disminución del costo, no es imperativo constar de una Arduino con la inclusión de un shield ya que sería un desperdicio de recursos. Se optara en lugar de esto por el diseño de una fuente, incluyendo en dicha placa el botón de pánico.

La solución más sencilla y económica es utilizar un diodo zener como regulador específicamente el 1N746A de ¼ de watio. Este diodo tiene un voltaje zener de 3.3 voltios ideales para las demandas de energía del chip xbee, Ira acompañado de una resistencia que absorberá el voltaje restante. Además solo es necesario un regulador puesto que al ser transmisores que no se modifica su ubicación dentro del domicilio se utiliza como fuente AC-DC un cargador de celular, que nos provee un voltaje de aproximadamente 9V, y la suficiente corriente para el transmisor.

Cálculo del valor para la resistencia:

$$V_z = 3.3\text{v (voltaje Zener)}$$

$$V_{\text{car}} = 9\text{v (voltaje cargador celular)}$$

$$I_z = 20\text{mA (corriente polarización zener)}$$

$$V_r = V_{\text{bat}} - V_z \text{ (voltaje sobre la resistencia)}$$

$$V_r = 9 - 3.3 \text{ v}$$

$$V_r = 5.7 \text{ V}$$

$$R = V_r / I_z \text{ (Ley de Ohm)}$$

$$R = 5.7 / 0.02$$

$$R = 285 \text{ ohmios.}$$

La figura 2.16, muestra el circuito correspondiente al regulador de voltaje con diodo zener.

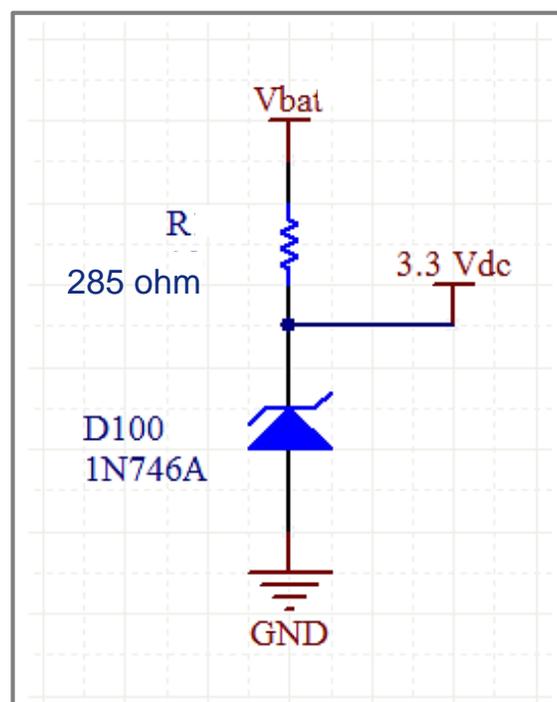


Fig. 2.16. Circuito regulador de voltaje con zener

2.8. Recepción de datos desde un terminal remoto

Una vez que se tiene el transmisor con su fuente y botón de pánico, en el coordinador se obtiene un frame tipo 0x95 como se indica en la tabla 2.18.

Tabla 2.18. Frame recibido por el coordinador

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Frame	7E	00	20	95	00	13	A2	00	40	52	2B	AA	7D	84	2	7D	84	00	13	A2	00	40	52	2B	AA	20	00	FF	FE	1	1	C1	5	10	1E	1B
Descripción	Encabezado		Número de bytes entre la longitud y el checksum		Frame type		Dirección de 64 bits del emisor						Dirección de 16 bits del emisor		0x02-Paquete Broadcast		Dirección emisor de 16 bits		Dirección de 64 bits del módulo remoto emisor						Texto que identifica al módulo remoto		Dirección del transmisor pariente		0=Coordinador 1=Router 2=End Device 1= Pushbutton Event		Fabricante		Checksum			

Debemos tomar en cuenta dos campos que son importantes para nuestra competencia, estos son el frame type, y la dirección de 16 bits. Siempre que el coordinador recibe un frame tipo 0x95, significa que se ha presionado el botón de pánico, y se puede obtener la dirección del dispositivo, y por consecuencia el domicilio del cual proviene la alerta.

A continuación se muestra el programa que determinará el comportamiento de la Arduino Mega si recibe un frame de este tipo.

```

if (ledPins[3] ==149) {           //preguntamos si el frame es tipo 0x95 si es
    digitalWrite (led, HIGH);    // el caso se activa la alarma que indica que
    digitalWrite(rele, HIGH);    // hay una alarma, activando un indicador
}                                 // LED y un relé

```

Puesto que se hace necesario conocer información que nos lleve a determinar desde que transmisor se generó la alerta se escribirán a un LCD el texto que identifica a dicho transmisor que está contenido dentro del frame.

2.9. Librería Arduino para LCD³⁵

Una de las librerías que forman parte del entorno Arduino es precisamente la que nos permite el uso de un LCD, para nuestro caso uno de tipo 2x16, pero de igual manera tiene la capacidad de ser compatible con diferentes tipos de LCD.

Esta biblioteca permite a la placa Arduino controlar displays LCD basados en el chipset Hitachi HD44780 (o compatibles), que se encuentra en la mayoría de LCD's de texto. La biblioteca trabaja en modo 4-bit o en 8-bit (es decir, por medio de 4 u 8 líneas de datos, además de RS, ENABLE, y, opcionalmente, las líneas de control RW).

Como referencia al uso de los comandos que se necesitan para la aplicación podemos apuntar;

- `LiquidCrystal(rs, enable, d4, d5, d6, d7)`
- `lcd.begin(columnas, filas);` *lcd* corresponde a una variable tipo `LiquidCrystal`
- `lcd.print (data).`

La aplicación y uso de esta librería la podemos ver de esta forma;

```
#include <LiquidCrystal.h>           // Se incluye el uso de la librería LCD
LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // Se inicializan los pines a ser usados
void setup() {
  lcd.begin(16, 2);                   // Se identifica al LCD como
  lcd.print("hola mundo!");          // uno de 2x16 y se escribe hola mundo!
}
```

³⁵ <http://arduino.cc/es/Reference/LiquidCrystal>

Mediante el LCD podremos desplegar la información contenida en el frame tipo 0x95.

2.10. Programación Arduino Uno

Ya que un terminal remoto por sí solo no se puede programarlo con instrucciones de alto nivel, se lo realiza con la inclusión de una Arduino, en este caso del modelo UNO, a continuación tenemos el código que permite enviar la información correspondiente a los RSSI de los nodos vecinos de tal suerte que serán utilizados para la estimación de la posición del nodo móvil.

Por tanto si existe una activación del botón en el transmisor móvil, es decir, el nodo que deseamos localizar, éste debe enviar hacia el coordinador la información de RSSI, de los nodos más cercanos a él.

De esta forma al existir presión del botón el transmisor debe “preguntar” por el valor RSSI, de los nodos cercanos, y posterior a eso enviarlos hacia el coordinador que procesará dicha información, para esto usamos dos tipos de frames, tipo 0x17 para solicitar valores de parámetros de forma remota y 0x10 para enviar dichos datos encapsulados en un paquete RF.

Bajo la referencia respecto de los frames de envío recepción, la única distinción se apunta a que están siendo usados los correspondientes valores decimales de la trama ya descrita, además apuntaremos que para todos los casos se realiza la conversión decimal-carácter para poder enviar datos hacia la UART.

```
Serial.print(char(126));  
Serial.print(char(00)); // Enviamos mediante la UART una trama  
Serial.print(char(15)); // solicitando valores de RSSI  
Serial.print(char(23));  
Serial.print(char(01));  
Serial.print(char(00));  
Serial.print(char(00));  
Serial.print(char(00));  
Serial.print(char(00));  
Serial.print(char(00));  
Serial.print(char(00));  
Serial.print(char(00));  
Serial.print(char(255));  
Serial.print(char(255));  
Serial.print(char(255));  
Serial.print(char(254));  
Serial.print(char(02));  
Serial.print(char(68));  
Serial.print(char(66));  
Serial.print(char(100)); //Checksum
```

Una vez enviado este frame, la respuesta serán tramas consecutivas de datos de RSSI de nodos cercanos, dichos datos los almacenamos y posteriormente los enviamos hacia el coordinador, encapsulado en un RF, en este caso el valor de checksum necesita ser calculado, puesto que los valores esperados siempre cambiarán su valor, tal como sigue su diagrama de flujo será;

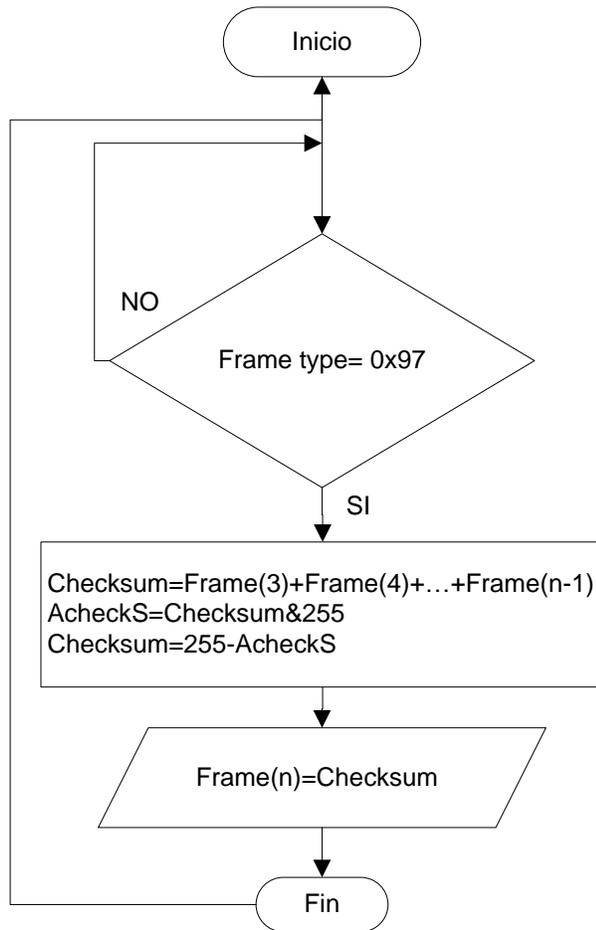


Fig. 2.17. Diagrama de flujo para calcular el Checksum.

De una manera similar a la anterior se procede a enviar en un frame tipo 0x10 los valores de RSSI encapsulados como el cuerpo del mensaje RF hacia el coordinador.

2.11. Interface de Usuario

Cada vez que existe una señal de alarma, es decir, se ha presionado el botón de pánico, el xbee en funcionamiento como coordinador, que está en conexión con la UART de la Arduino Mega, recibirá el respectivo frame tipo 0x95, y mediante el uso de un LCD, permitirá visualizar el identificador de usuario.

Además, este frame es enviado hacia la PC, para ser procesada y permitir una mejor visualización de los datos desde donde se generó la alerta.

2.11.1. LabView Visa³⁶

Ya se analizó previamente que tanto la Arduino Uno, como la Arduino Mega, se conectan al PC mediante el uso de un chip FTDI que realiza una conversión creando así un puerto serie virtual, de esta forma el manejo de datos en LabView se realiza mediante el uso de VISA serial.

De esta forma la figura 2.18 muestra el diagrama de bloques que permite la recepción de datos en forma serie.

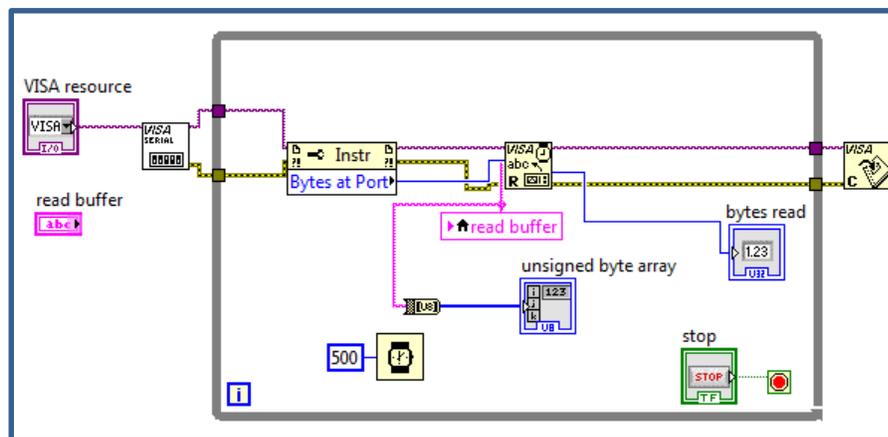


Fig. 2.18. Diagrama de bloques VISA read.

Es necesario determinar el número de puerto (COM) que la PC asigna a nuestra tarjeta para que este sea el puerto especificado en VISA resource. Además se realiza una conversión String to Array para visualizar cada uno de los elementos de la trama, por tanto cada byte estará separado en cada índice de dicho vector.

³⁶ <http://www.ni.com/white-paper/7907/es/>

La figura 2.19 muestra el Panel frontal para esta lectura.

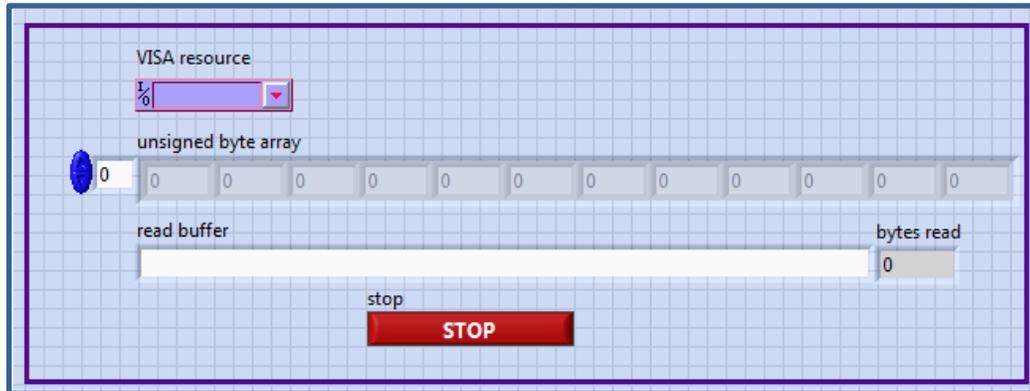


Fig. 2.19. Panel frontal VISA read.

De igual forma, se tiene los VI's correspondientes a la escritura en el puerto serie en la figura 2.20.

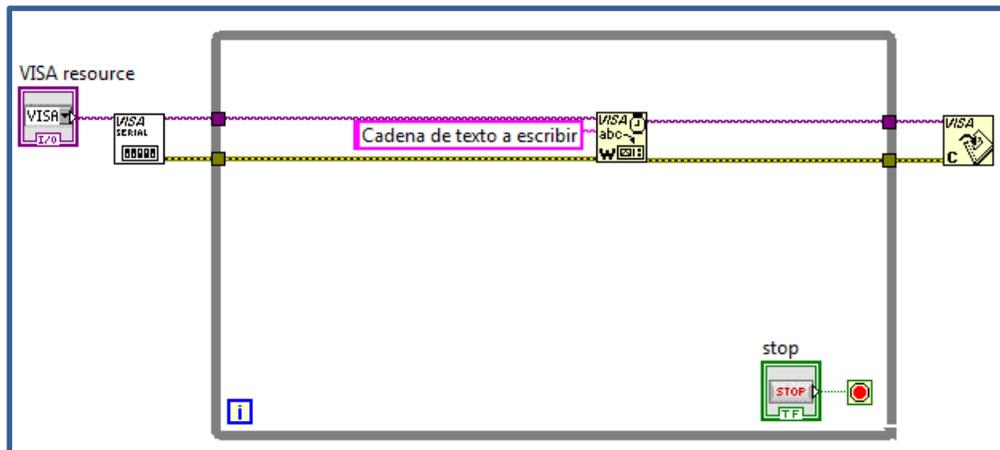


Fig. 2.20. Diagrama de bloques para la escritura.

Estos VI's correspondientes a VISA toolkit, nos permitirá realizar el intercambio de datos desde la Arduino Mega y la PC.

Una vez que esto es posible a cada evento de alerta, la interface necesitará responder con información visible del lugar donde se generó la alarma.

CAPITULO 3

IMPLEMENTACIÓN DEL SISTEMA

Como parte del desarrollo del presente proyecto restará poner a punto el sistema, tal que será provisto de los datos necesarios para cada uno de los usuarios así como de una interface intuitiva para usuario.

3.1. Estación base

Una vez analizados los elementos necesarios, además de los módulos que debe tener nuestro software, tanto en Arduino como en LabView, podemos establecer nuestra estación base, compuesta de una Arduino Mega, Su respectivo shield que contiene al xbee coordinador. Constará también una forma de visualizar la información mínima de tal forma que nos permita saber el campo que identifica cada transmisor, en este caso ese elemento será un LCD.

La figura 3.1 muestra la distribución de pines de un LCD³⁷.

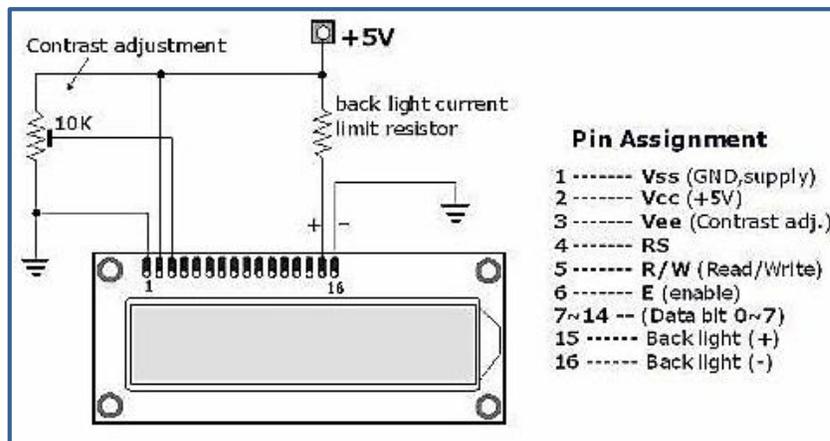


Fig. 3.1. LCD 2x16 distribución de pines.

³⁷ <http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>

En cuanto a software se refiere, la Arduino Mega estará compuesta con un módulo que recibirá desde la red xbee, y se encargara de enviarla hacia la PC, además deberá mostrar hacia el LCD el identificador del nodo que generó la alarma.

El resultado final se muestra en el **ANEXO 1**.

3.2. Nodo Móvil

De igual manera el nodo móvil estará compuesto de una Arduino UNO, con su respectivo shield para un xbee router. Constará demás con un pulsador que permite enviar la señal de que necesita ser localizado. Previamente se detalló las particularidades en cuanto a programación se refiere, además podemos observar el resultado final del nodo móvil en el **ANEXO 2**.

3.3. Panel frontal para Usuario

Para la visualización se ha determinado los campos para configurar el puerto serie (COM) además consta de un indicador del número de bytes que han sido leídos, además tenemos el campo donde esta expresado la referencia del nodo que será detallada posteriormente. Cuando exista una alarma, se tendrá el nombre del usuario en el domicilio, en el campo de referencia, también se visualizarán las coordenadas tanto de longitud como de latitud. Consta de un botón reset, que tiene como función la de reiniciar el sistema a condiciones iniciales.

La figura 3.2 muestra el panel frontal.

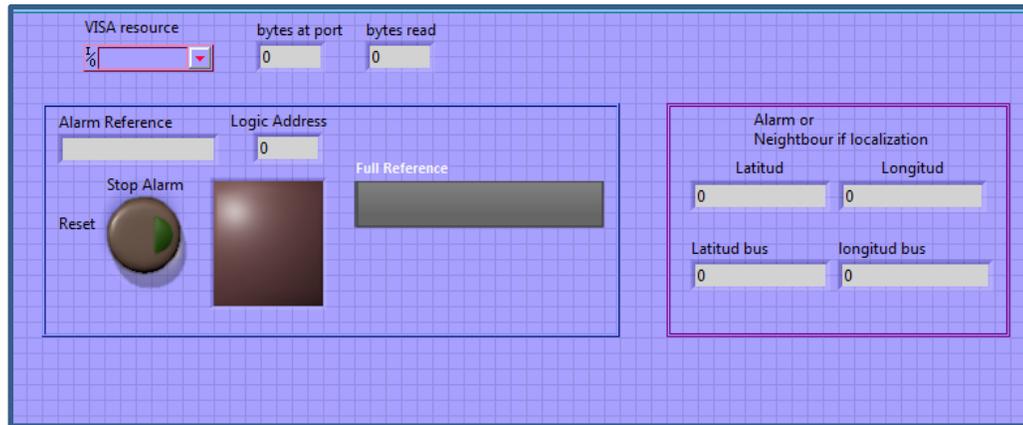


Fig. 3.2. Panel frontal de Usuario.

Como es evidente el programa debe tener la capacidad de separar cada caso se presenta, si es una llamada de alarma, o un requerimiento de localización de nodo móvil, toda la información necesaria para esto, está contenida en cada uno de los frame que son enviados hacia la PC por intermedio de la UART de la Arduino Mega.

3.4. Datos de Usuarios

Una vez que se ha identificado que tipo de frame es, se necesita tomar la información considerada mínima y devolver una más detallada para el usuario. Dicha información contenida en el diagrama de bloques servirá de base de datos de nombre, coordenadas, referencias y direcciones lógicas. La tabla 3.1 muestra las direcciones físicas y las direcciones lógicas que adoptaron una vez escrito el firmware.

Tabla 3.1. Direcciones lógicas y físicas

Dirección lógica (16 bits)	Dirección física(64 bits)
D441	0013A200409C12A2
7131	0013A20040992CCE
2020	0013A200408B2EE8
BBD9	0013A200408BEDC
E9B5	0013A20040A09C33
892A	0013A200408B2EBF
C2B8	0013A200408B2EF3
8024	0013A200408B2ED3
B483	0013A200409C1291
9BF8	0013A200408B2EDF
5E 49	0013A2004089B6B9
3040	0013A2004089B6BB
78B3	0013A20040992D8C

3.4.1. Referencia de texto

Una cadena de texto también forma parte del identificador de cada uno de los transmisores, además de la ya mostrada dirección, ya que es suficiente para identificarlo solo una dirección, la cadena de texto será usada solamente en el LCD. La tabla 3.2 muestra las direcciones y sus cadenas de texto identificativas.

Tabla 3.2. Direcciones lógicas y texto.

Dirección lógica (16 bits)	Texto Identificativo
D441	ROUTERMED
7131	ROUTERAUX
2020	AORTIZ
BBD9	PLLERENA
E9B5	VICHASI
892A	DEGRAN
C2B8	ROBELT
8024	MSANCHEZ
B483	SGUERRERO
9BF8	JVASCONEZ
5E 49	EJARA
3040	SMEJIA
78B3	EMIRANDA

3.4.2. Nombre Usuario

Si bien es suficiente una dirección, y además de eso una referencia de texto intuitiva, la interface de usuario dará información completa del nodo que generó la alarma. La tabla 3.3 muestra las direcciones y sus correspondientes usuarios.

Tabla 3.3. Direcciones y Nombres de Usuario

Dirección lógica (16 bits)	Nombre Usuario
D441	Casa Comunal
7131	José Ramírez
2020	Alberto Ortiz
BBD9	Paulo Llerena
E9B5	Violeta Chasi
892A	Delia Granja
C2B8	Robert Beltrán
8024	Marcelo Sánchez
B483	Segundo Guerrero
9BF8	Juan Vásconez
5E 49	Elena Jara
3040	Salomón Mejía
78B3	Elvira Miranda

3.4.3. Coordenadas Espaciales de los Nodos

Una vez que se tiene los datos del usuario, restará por obtener los datos espaciales de cada uno de los nodos fijos, ya que estas coordenadas sirven como base para determinar las posibles coordenadas del nodo móvil.

La tabla 3.4 muestra las direcciones lógicas y sus coordenadas decimales de longitud y latitud.

Tabla 3.4. Direcciones y coordenadas espaciales de los nodos.

Dirección lógica (16 bits)	Latitud	Longitud
D441	-1,36167	-78,66083
7131	-1.36181	-78.66089
2020	-1,36167	-78,66028
BBD9	-1,36222	-78,66027
E9B5	-1,36194	-78,66111
892A	-1,36166	-78,66083
C2B8	-1.35944	-78.65972
8024	-1.36111	-78.66138
B483	-1.36233	-78.66056
9BF8	-1.36219	-78.66127
5E49	-1.36253	-78.66123
3040	-1.36158	-78.66081
78B3	-1,36139	-78,66028

3.5. Localización del nodo móvil en Google Earth

Si bien se puede obtener la referencia del nodo cercano al que esperamos localizar, además de sus coordenadas, se determina como complementario el uso de un elemento adicional para visualizarlo, de esta manera se lo hará en el mapa provisto por Google. Para esto la interface creada en Labview maneja el control de un archivo tipo “.kml” que nos permitirá hacer lo ya expuesto.

3.5.1. Archivo “.kml”³⁸

KML es un formato de archivo que se utiliza para mostrar datos geográficos en un navegador terrestre, como Google Earth, Google Maps y Google Maps para móviles. KML utiliza una estructura basada en etiquetas con atributos y elementos anidados y está basado en el estándar XML. Todas las etiquetas distinguen entre mayúsculas y minúsculas y deben aparecer exactamente como aparecen en la Referencia de KML.

3.5.1.1. Marcas de Posición

Las marcas de posición (Placemark) son uno de los recursos más utilizados en Google Earth. Permiten marcar una posición en la superficie de la Tierra con un icono de pin amarillo. La marca de posición (Placemark) más sencilla incluye solo un elemento de punto (<Point>), que especifica la ubicación de la marca de posición. El código KML para la marca de posición simple es el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">

  <Placemark>
    <name>Simple placemark</name>
    <description>Attached to the ground. Intelligently places itself
      at the height of the underlying terrain.</description>
    <Point>
      <coordinates>-
122.0822035425683,37.42228990140251,0</coordinates>
    </Point>
  </Placemark>

</kml>
```

³⁸ <https://developers.google.com/kml/documentation/kmlreference?hl=es>

3.6. Interface para visualizar el Nodo Móvil.

Puesto que se tiene una interface que realiza el manejo de los datos recibidos por el puerto serie, además de entregarnos la información detallada de los usuarios, se ha desarrollado una nueva ventana que nos permitirá visualizar la lista de usuarios dentro del sistema así como la localización del nodo móvil en el mapa. La figura 3.3, muestra la interface propuesta.



Fig. 3.3. Interface para visualización del Mapa.

El presente Panel Frontal consta de un control numérico que indica el número de Usuario y de indicadores que permiten ver el nombre completo, la dirección lógica y sus respectivas coordenadas, además de un botón que realiza el enlace con la aplicación Google earth para desplegar las marcas de los domicilios en el mapa.

3.7. Visualización mediante Google Earth

Con el uso del mapa satelital del que consta Google, situamos con marcas los lugares donde se encuentran los nodos fijos esto estará previamente determinado con el uso del archivo de extensión “.kml”, de esta forma no es necesario recurrir a elementos externos para determinar dichos lugares, y no demanda un estricto conocimiento del lugar que se está monitoreando. La figura 3.4, muestra la vista de las marcas que indican los domicilios con los nodos fijos instalados.



Fig. 3.4. Localización de los nodos fijos.

3.8. Pruebas de nodo móvil

Se pretende establecer la confiabilidad del dispositivo y del cálculo de la posición del dispositivo móvil, por tanto se ubica el dispositivo en la posición indicada en la siguiente figura 3.5.

Una vez ubicado el nodo móvil en el punto especificado se procede a realizar pruebas sucesivas, bajo las mismas condiciones, de tal suerte que se obtendrá la posición dada por el algoritmo encargado de dicha acción, y así se estimara la veracidad del cálculo.



Fig. 3.5. Marca de prueba.

Con el uso de la interface previamente detallada, se realiza la localización del nodo móvil, con cuatro pruebas consecutivas, la dispersión de los datos se muestra en la siguiente figura 3.6, mientras tanto, la tabla 3.5, muestra las coordenadas resultantes.

Tabla 3.5. Coordenadas resultantes.

Prueba	Longitud	Latitud
1	-78.66062	-1.36119
2	-78.66008	-1.36101
3	-78.66083	-1.36133
4	-78.66037	-1.36149



Fig. 3.6. Datos de la localización del nodo móvil.

3.9. Alcances y Limitaciones

3.9.1. Alcances

Al tener una red establecida, sirve como una base de una red que puede ser expandida si existen requerimientos de nuevos usuarios ya que una de las propiedades de zigbee es ser escalable, de esta forma se cumple con el

objetivos de que sea una alarma comunitaria, ya que los elementos que sirvieron en un inicio para un número mínimo de usuarios, son igual de útiles si se incrementa la misma, haciéndola una red con optimización de recursos.

Como contamos con elementos que tienen aún mayores prestaciones, los servicios que hasta el momento prestan pueden ser mejorados de tal suerte que el usuario puede adaptarlo y personalizarlo a sus necesidades.

El HMI, es amigable al usuario, permitiendo así que pueda ser utilizado en la manera de lo posible por todo tipo de persona que se considere en capacidad de usar un PC en su operación más básica, haciendo así innecesario un conocimiento adicional para operar el sistema.

3.9.2. Limitaciones

Al ser un sistema que en su mayor parte está conectado a la red eléctrica convencional, bajo fallas de energía el sistema quedará obsoleto o sin funcionar.

Al constar de elementos que si bien presentan beneficios en costo y funcionalidad, como son elementos de baja potencia siempre estarán sujetos a interferencias que los afectaran tales como clima, obstáculos físicos, etc. Serán causas principales de fallo en la estimación de posiciones en el nodo móvil.

CAPITULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Al concluir el presente proyecto se ha cumplido con el objetivo planteado que fue el de poner en funcionamiento una red que permita realizar posicionamiento terrestre, además de un sistema de alarma para el barrio Santa Lucia centro del cantón Tisaleo.
- Se desarrolló un algoritmo que permite la localización del nodo móvil, de tal suerte que en cada caso se vean reflejadas las coordenadas que indique la situación geográfica de dicho elemento lo más cercano a su realidad.
- Se diseñó una interface amigable con el operador que no demande conocimientos avanzados de uso de una PC, para la simplificación de su uso.
- Se realizó la construcción de todo el equipo necesario para la operación del sistema ya sea con atención del operador como con ausencia del mismo.
- Se elaboró el correspondiente manual de operador que estará al alcance de la persona encargada de esta función.
- Se entregó a cada uno de los usuarios un instructivo que permita la comprensión del funcionamiento y cuidados necesarios del dispositivo.

4.2. Recomendaciones

- Se recomienda que los dispositivos estén siempre conectados a la red eléctrica puesto que su alimentación depende de la misma, y no deben dejar de funcionar para que exista la red tipo malla.
- Realizar la instalación de los dispositivos considerando la mejor ubicación que minimice los efectos ambientales sobre las comunicaciones inalámbricas.
- Procurar realizar una verificación coordinada de los dispositivos para que su funcionamiento sea como el esperado y no exista problemas bajo una alarma real.
- Es necesario que si existe la necesidad de reubicar un dispositivo en el domicilio, se notifique previamente ya que dichos elementos se encuentran ubicados de tal forma que permitan la comunicación con el resto de la red, y cambios sin consideración de esta recomendación, causarían el malfuncionamiento de la malla.
- Se considera como recomendación a tomar muy en cuenta la responsabilidad que demanda la activación de una señal de alarma, ya que si es falsa, creará pánico en los habitantes del barrio omitiendo la importancia de una posible alerta real.

BIBLIOGRAFÍA

Textos

- Miguel Rodríguez et al., “Blueps: Sistema de localización en interiores utilizando Bluetooth”, URSI 2005.
- P. Corral, R. Barbéra et al., “Mejoras en Sistema de Localización mediante ZigBee” XXIV Simposium Nacional de la Unión Científica Internacional de Radio URSI 2009.

Links de internet

- Arango Z, Eliana I. *Plataformas Zigbee*. Desde <http://plataformaszigbee.blogspot.com/>
- *Xbee*. Desde <http://xbee.wikispaces.com/Mesh+with+Xbee>
- *Arduino Uno*. Desde <http://arduino.cc/en/Main/ArduinoBoardUno>
- Mazurov, Oleg. *Playing Xbee. Part 1 - First impression*. Desde <http://www.circuitsathome.com/mcu/playing-xbee>
- *Calculating the Checksum of an API Packet*. Desde <http://www.digi.com/support/kbase/kbaseresultdetl?id=2206>
- *API Mode*. Desde https://sites.google.com/site/xbeetutorial/xbee-introduction/api_mode
- Engineers Garage. *LCD*. Desde <http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>
- Internet de las cosas. *Primeros pasos en XBee, usando arduino*. Desde <http://www.internetdelascosas.cl/2012/08/26/primeros-pasos-en-xbee-usando-arduino/>

- Seco G, Fernando. *Sistemas de localización en interiores basados en radiofrecuencia*. Desde <http://www.car.upm-csic.es/lopsi/static/publicaciones/docencia/Apuntes%20RF-LPS.pdf>
- Zigbee Labs. *Métodos de localización en interiores con redes inalámbricas*. Desde <http://www.zigbe.net/archivos/424>

ANEXOS

ANEXO 1

Estación Base

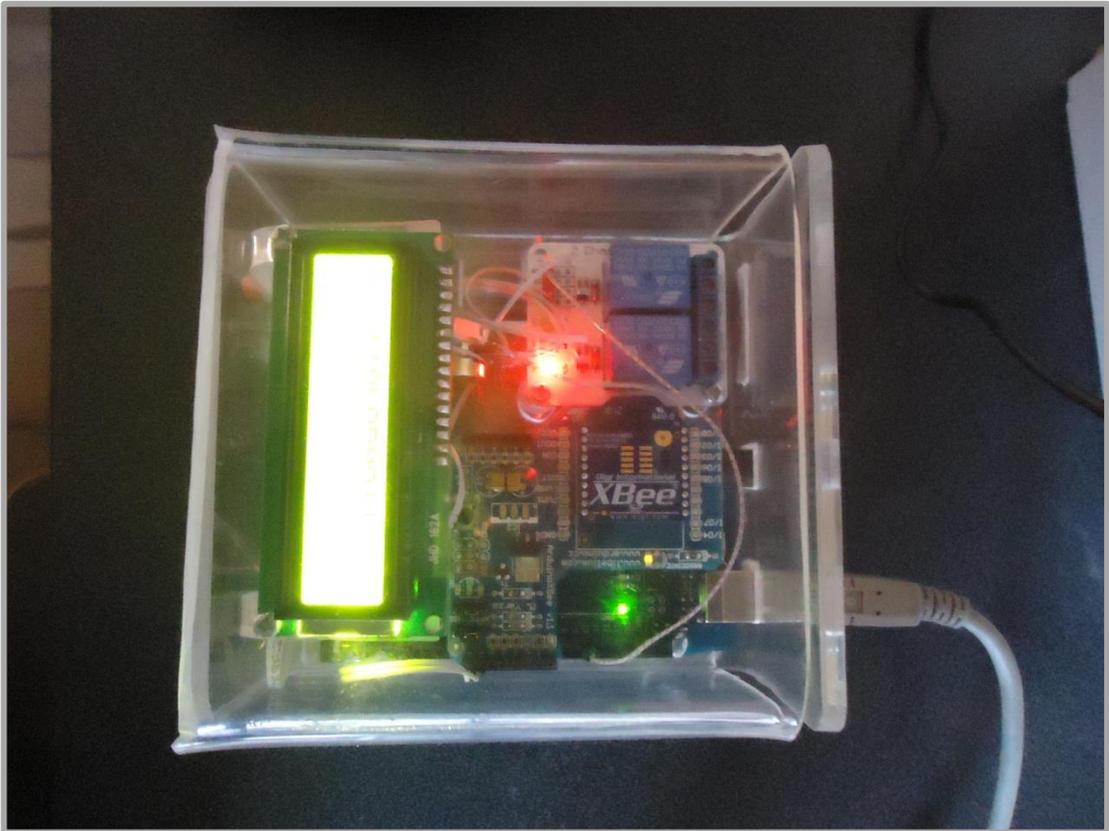


Fig A1.1. Estacion Base.

ANEXO 2

Nodo Móvil

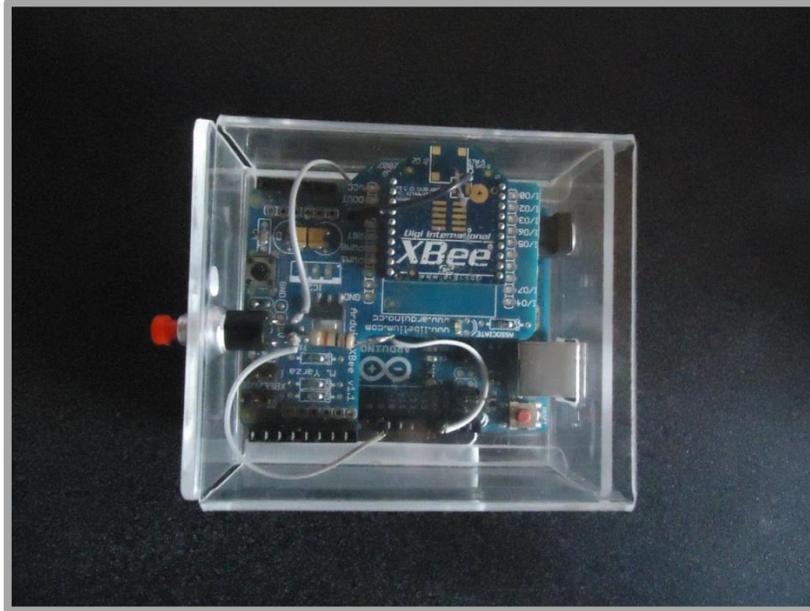


Figura A2.1. Nodo móvil, descrito en el manual de operador.

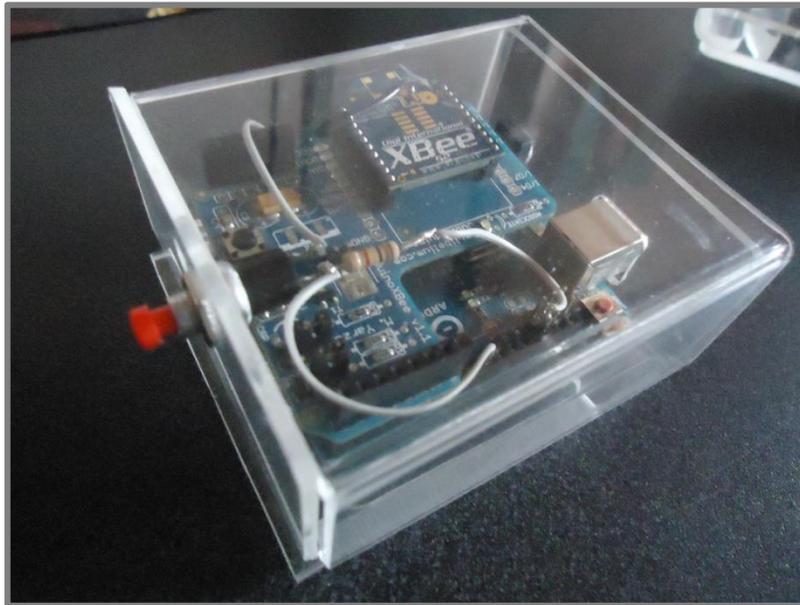


Figura A2.2. Pulsador que permite la petición de localización.

ANEXO 3

Instructivo para el uso del transmisor XBEE

En el presente instructivo se presenta una breve guía de los elementos constitutivos y la función de cada uno de los mismos, de tal suerte que sea transición al uso del presente equipo propendiendo al uso responsable, al igual que sea beneficio para cada uno de los usuarios y de la comunidad por extendido tiempo.

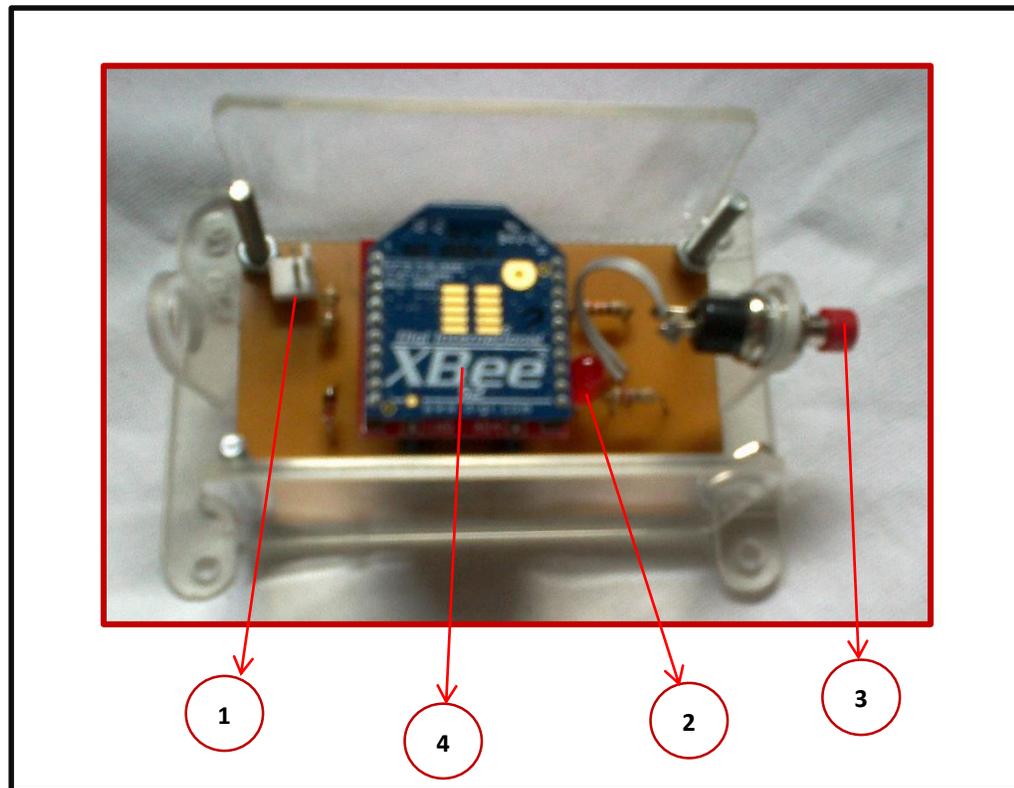


Figura. A3.1. Nodo fijo.

1. Conector de carga
2. Led indicador de estado activo.
3. Pulsador de alerta (Panic Button).
4. Elemento transmisor y controlador del circuito.

- El led indicador (2) se encontrara en estado activo y titilando cuando se encuentre en estado normal.
- El pulsador (3) es el elemento por el cual se dará una señal de alerta, considerando la responsabilidad implicada en generar una llamada de atención hacia la comunidad, se realizara presionando dicho elemento por aproximadamente por 2 segundos, así dando la señal de alerta.
- El elemento (4) no debe ser removido bajo ninguna condición.
- En caso de daño comunicar al titular del presente instructivo.

ANEXO 4

Manual Operador

Con la intención de proporcionar información necesaria para el óptimo uso del equipo que forma parte de la red, se presenta el siguiente manual de operación.

Como primer punto se debe tomar en cuenta que no es necesario un conocimiento avanzado de manejo de un PC de hogar, puesto que se ha previsto de una interface amigable con el usuario.

Como primer punto debemos establecer que los usuarios están identificados por varios campos, además de sus coordenadas tanto de longitud como de latitud, que son mostrados en la tabla A1.1.

Tabla A1.1. Usuarios e Identificadores

Dirección lógica (16 bits)	Dirección física (64 bits)	Texto Identificativo	Nombre Usuario	Coordenadas	
				Latitud	Longitud
D441	0013A200409C12A2	ROUTERMED	Casa Comunal	-1,361667	-78,66083
7131	0013A20040992CCE	ROUTERAUX	José Ramírez	-1.36181	-78.66089
2020	0013A200408B2EE8	AORTIZ	Alberto Ortiz	-1,361667	-78,66028
BBD9	0013A200408BEDC	PLLERENA	Paulo Llerena	-1,36222	-78,66027
E9B5	0013A20040A09C33	VICHASI	Violeta Chasi	-1,36194	-78,66111
892 ^a	0013A200408B2EBF	DEGRAN	Delia Granja	-1,36166	-78,66083
C2B8	0013A200408B2EF3	ROBELT	Robert Beltrán	-1.35944	-78.65972
8024	0013A200408B2ED3	MSANCHEZ	Marcelo Sánchez	-1.36111	-78.66138
B483	0013A200409C1291	SGUERRERO	Segundo Guerrero	-1.36233	-78.66056

9BF8	0013A200408B2EDF	JVASCONEZ	Juan Vásconez	-1.36219	-78.66127
5E49	0013A2004089B6B9	EJARA	Elena Jara	-1.36253	-78.66123
3040	0013A2004089B6BB	SMEJIA	Salomón Mejía	-1.36158	-78.66081
78B3	0013A20040992D8C	EMIRANDA	Elvira Miranda	1,36139	78,66028

A1.1. Primeros Pasos

Si bien no es necesario conocimientos avanzados de computación para la operación de los dispositivos, como apuntes a tomar en consideración detallaremos la conexión de la estación base mediante USB hacia la PC.

Todo esto debido a que es necesario establecer como paso inicial el número que ha sido asignado al puerto serie denominado virtual, que si bien no existe físicamente, constara como uno más de los periféricos de entrada/salida de la PC, por tanto se detalla el procedimiento para la obtención de dicho número de puerto.

Una vez que se ha realizado la conexión con el cable que está incluido con el equipo, la figura a, y b, muestran los pasos a seguir.

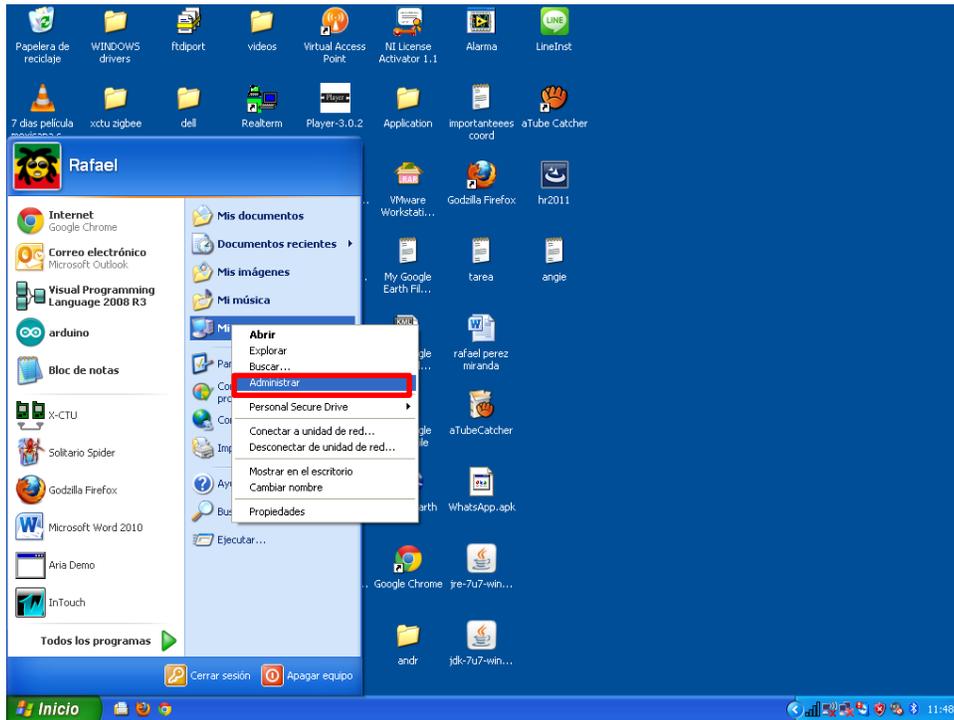


Fig. a. Primeros pasos a seguir



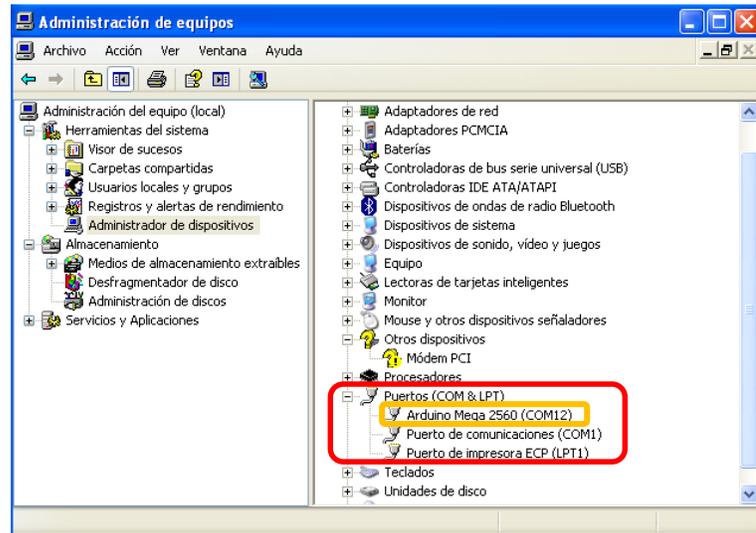


Fig. b. Proceso previo al funcionamiento de la interface gráfica.

En este proceso se muestra la forma de obtener el número que el sistema asigna a la conexión de nuestra estación base, se debe considerar que escogemos la que está acompañada del nombre Arduino Mega 2560, que en este caso es COM12.

Este número será detallado en la descripción extendida de la interface gráfica.

Una vez que se tiene a consideración la presentada información se ejecuta la Aplicación como indica la figura c, y de esta forma finalmente usaremos el software como detalla la figura A1.1.

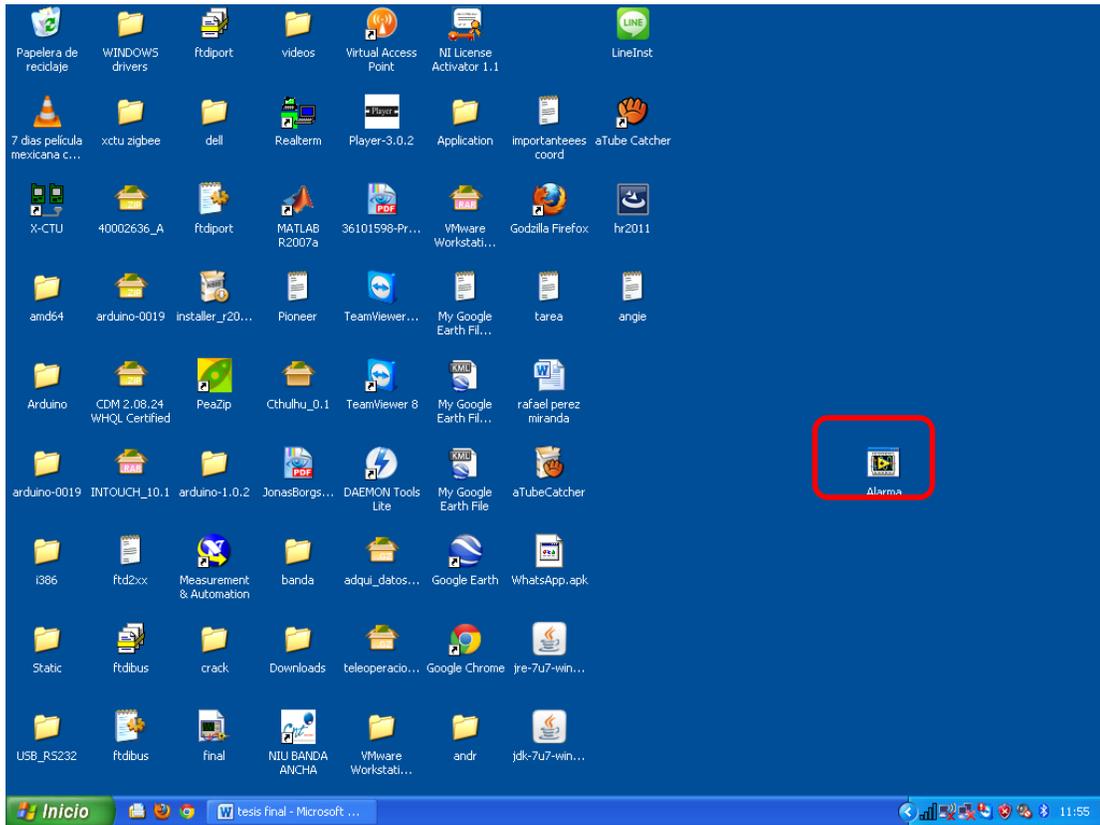


Fig. c. Ejecución de la aplicación Alarma.

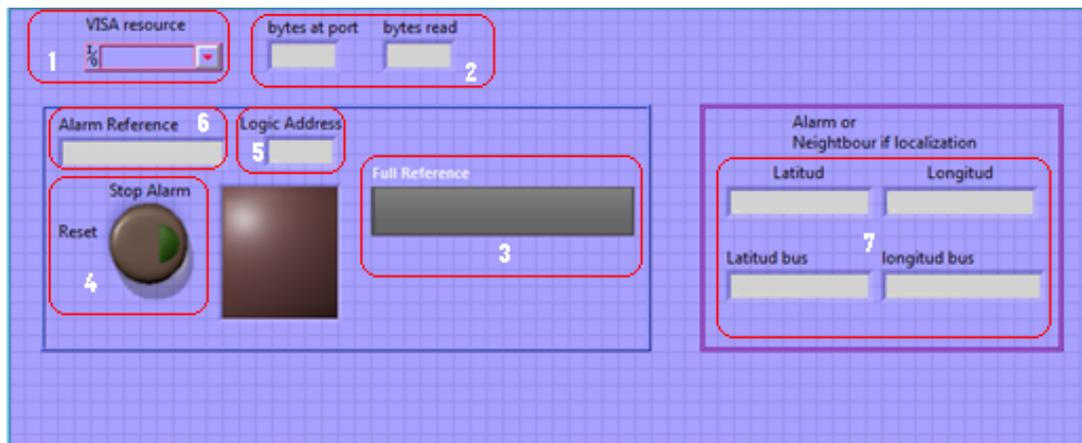


Fig. A1.1. Interface del Operador.

1. **Selección Puerto Serie:** Con este selector podemos escoger el canal por el cual se envía y recibe la información hacia y desde la estación base debemos tomar en cuenta que la denominación de dicho canal serie es “COM X”, siendo x un numero entero.
2. **Numero de bits:** En este campo se muestra de un lado los bits que están en el puerto, y en contraparte el número de bits que fueron leídos hacia el programa.
3. **Nombre de Usuario:** En este apartado se mostrara la información extendida del usuario que ha hecho una llamada de alarma, es decir, ha presionado su botón de pánico, por ende esta información dependerá del nombre que se ha almacenado previamente en la base de datos del programa.
4. **Reinicio Alarma:** Si se ha presentado una alarma, el indicador de luz estará encendido y por ende la alarma será desactivada, y los datos que se presentaron necesitaran ser borrados con la presión de dicho botón.
5. **Dirección Lógica:** Esta información representa la identificación que tiene cada uno de los transmisores y que son transmitidos cada una de las veces que sea presionado el botón de pánico y que permitirá al software mostrar la información detallada del usuario.
6. **Referencia:** Esta información corresponde a la misma mostrada en el indicador LCD de la estación base conectada a la PC.
7. **Coordenadas:** A cada uno de los usuarios se ha determinado su localización geográfica, por ende será mostrada cada vez que exista una alarma, además consta de indicadores para si el caso existe en el que el nodo que genera la alarma es el nodo móvil mostrara las coordenadas en donde se encuentre.

Considerando que no es suficiente el solo conocer la posición geográfica mostrada numéricamente, también se pone a consideración la interface de la figura A1.2, con la cual se mostrarán graficadas las posiciones en el mapa de Google.

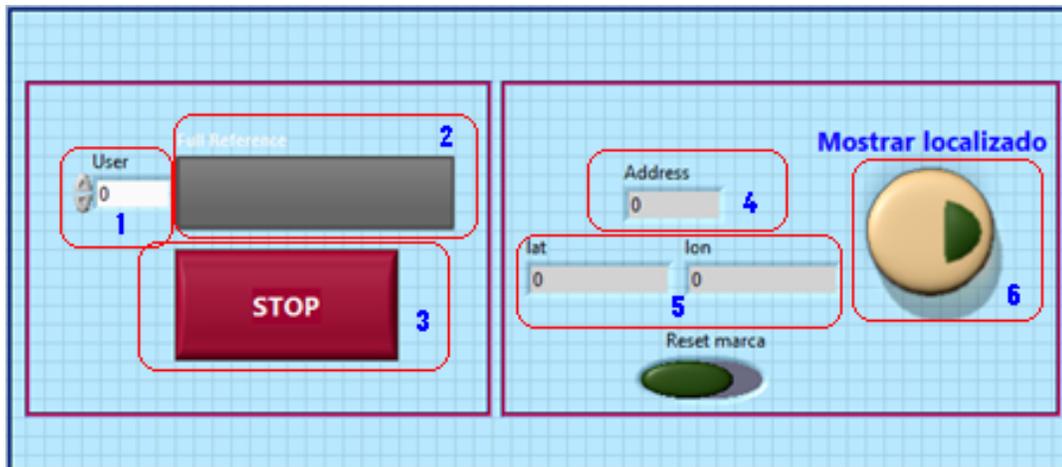


Fig. A1.2. Interface para visualizar marcas.

1. **Indicador de usuario:** Con este selector se mostrarán todos los usuarios que constan en el sistema.
2. **Nombre Usuario:** Detallará el nombre completo del usuario que indica el selector.
3. **Botón de paro:** Detiene la ejecución del programa.
4. **Dirección lógica:** Muestra la dirección lógica de los transmisores.
5. **Coordenadas:** Muestra las coordenadas geográficas almacenadas.
6. **Mostrar Mapa:** Ejecutara Google Earth con las marcas de los domicilios.

A1.2. Equipo Físico

Cada vez que no está disponible el PC, el sistema estará en constante funcionamiento, además se obtendrá la señal de alarma siendo posible

visualizar el lugar desde donde se generó la misma mediante e visualizador LCD, para esto los componentes importantes de esta estación base en la figura A1.3.

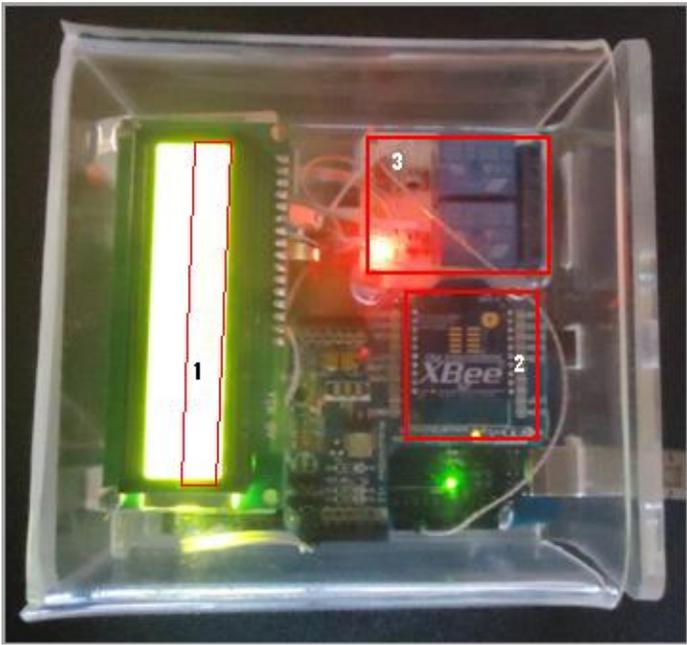


Fig. A1.3. Estación Base.

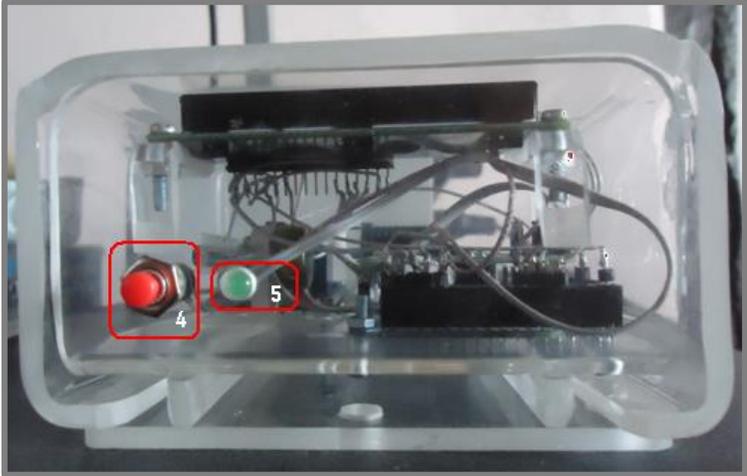


Fig. A1.4. Estación Base vista lateral derecha.

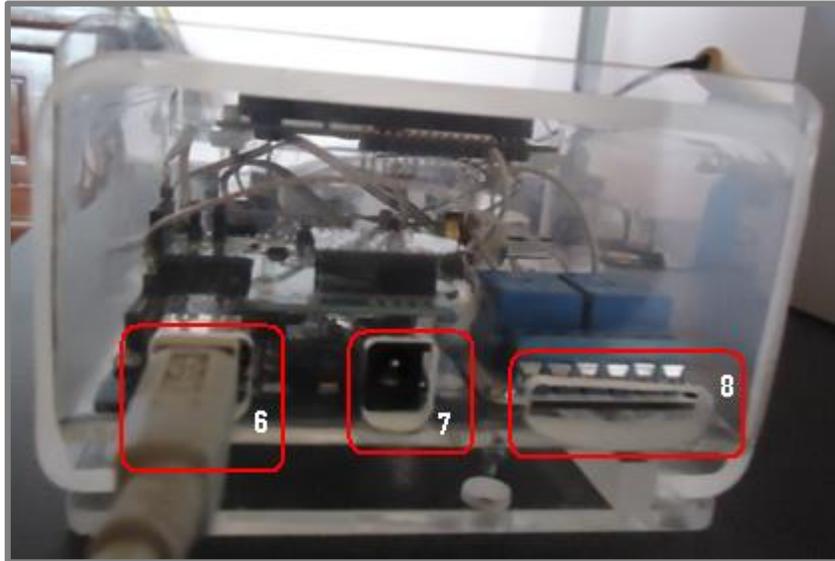


Fig. A1.5. Estación Base vista lateral izquierda

1. **Indicador LCD:** En este campo se podrá visualizar el identificador de usuario previamente establecido en la tabla A1.1, permitiendo así su identificación.
2. **Transmisor Xbee:** Módulo inalámbrico, encargado de la transmisión de datos mediante RF.
3. **Módulo de Potencia:** Relé encargado de activar la alarma sonora que recibe la señal de control desde la tarjeta Arduino Mega para realizar activación de dispositivos de mayor potencia.
4. **Pulsador de Reseteo:** Este pulsador permite al sistema reiniciarse después de una alarma, en caso de no estar conectado a la PC, es equivalente al pulsador de reseteo de la interface gráfica previamente analizada.
5. **Indicador LED:** Este indicador estará encendido siempre que existe una alarma, y por ende se apagará cuando se haya presionado el pulsador detallado en el apartado anterior.

6. **Conector USB:** Nos permite la comunicación con la PC para el flujo de datos.
7. **Conector de alimentación:** Cuando no está presente la conexión hacia la PC, la alimentación de la estación base se realiza por el mencionado conector con el adaptador AC/DC, provisto con el equipo.
8. **Conexiones de Potencia:** Esta sección permite la conexión del elemento sonoro de alarma externo al equipo en descripción.

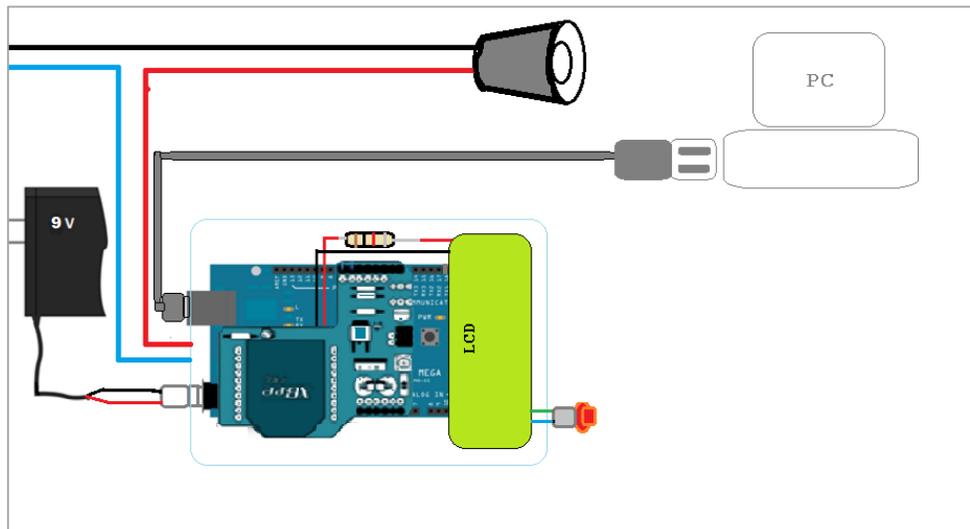


Fig. A1.6. Conexión Estación Base, alimentación y conexión PC.

También se pone a consideración las características principales del nodo móvil en la figura A1.7.

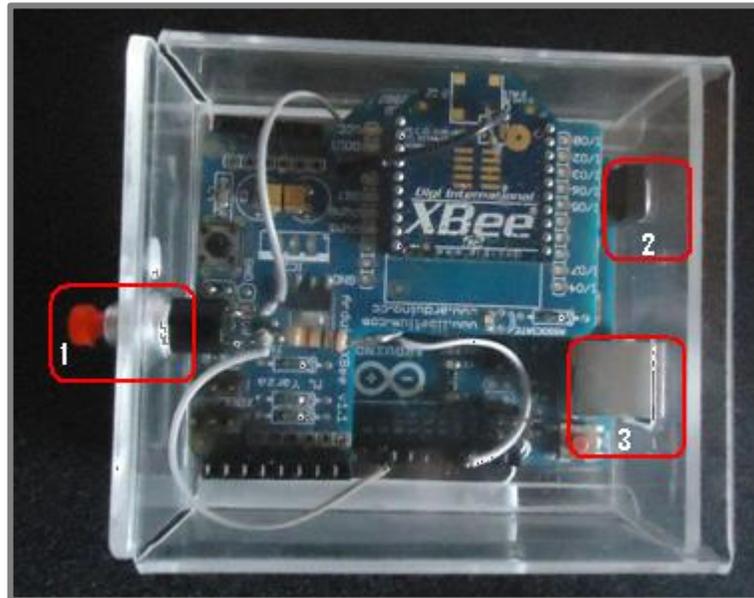


Fig. A1.7. Nodo Móvil.

- 1. Pulsador de petición de localización:** Cuando es presionado el dispositivo envía la información necesaria para ser localizado hacia la estación base.
- 2. Alimentación:** Elemento que permite energizar al dispositivo con baterías ya que no dispone de lugar estático la alimentación será a baterías también provistas con el equipo.
- 3. Conector USB:** Conector que permitirá modificación o reprogramación de los elementos que conforman el dispositivo si éste fuera el caso.

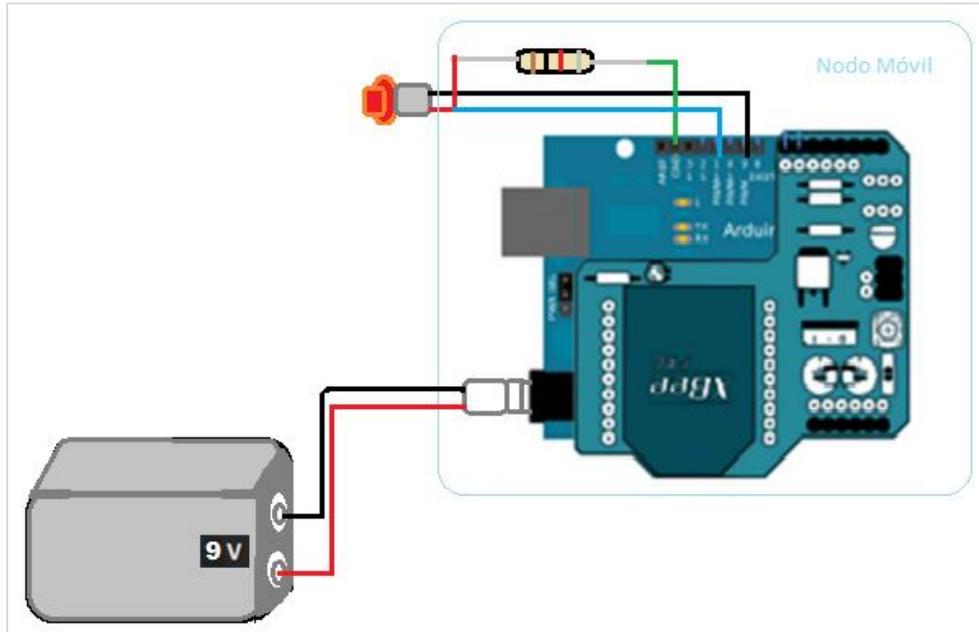


Fig. A1.7. Conexión Nodo móvil con batería.

ANEXO 5

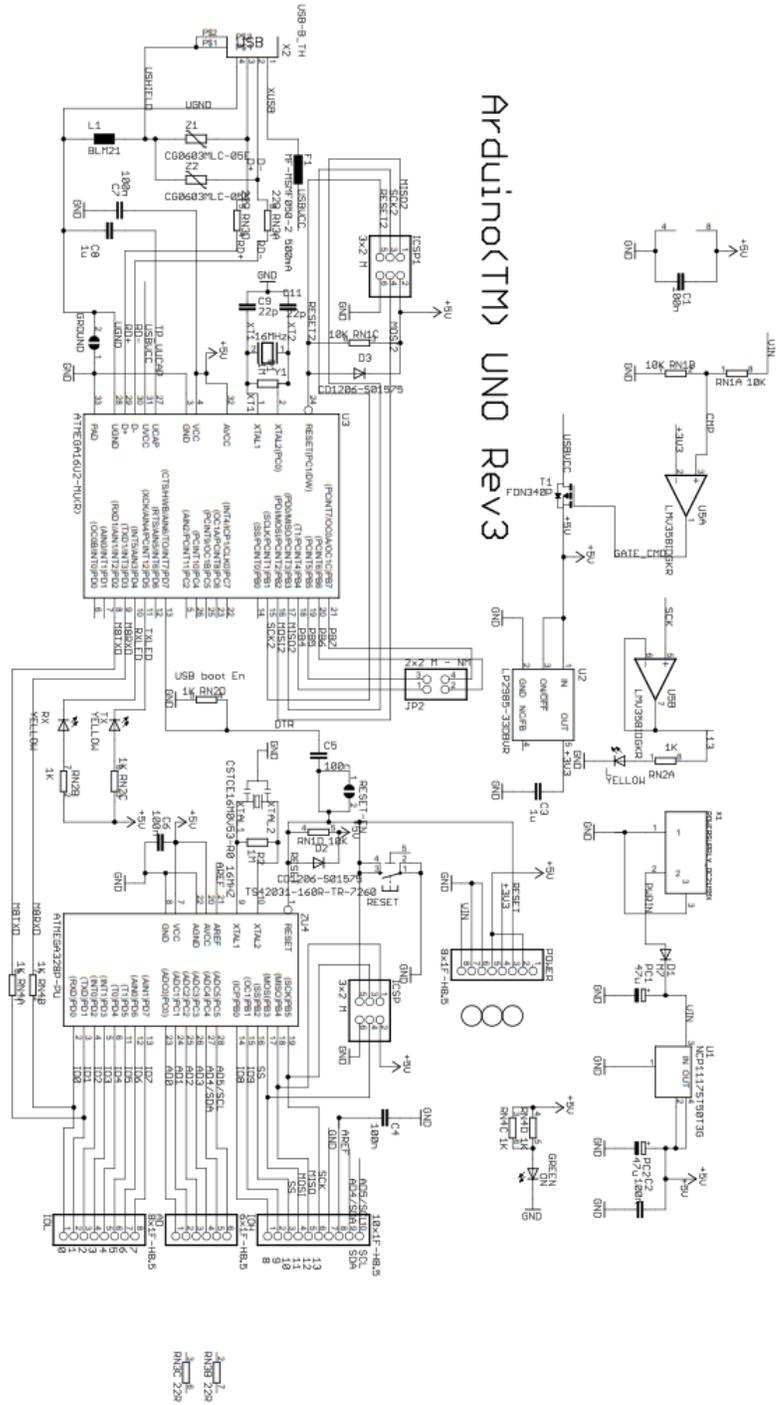


Fig. A5.1. Diagrama esquemático Arduino Uno

ANEXO 6

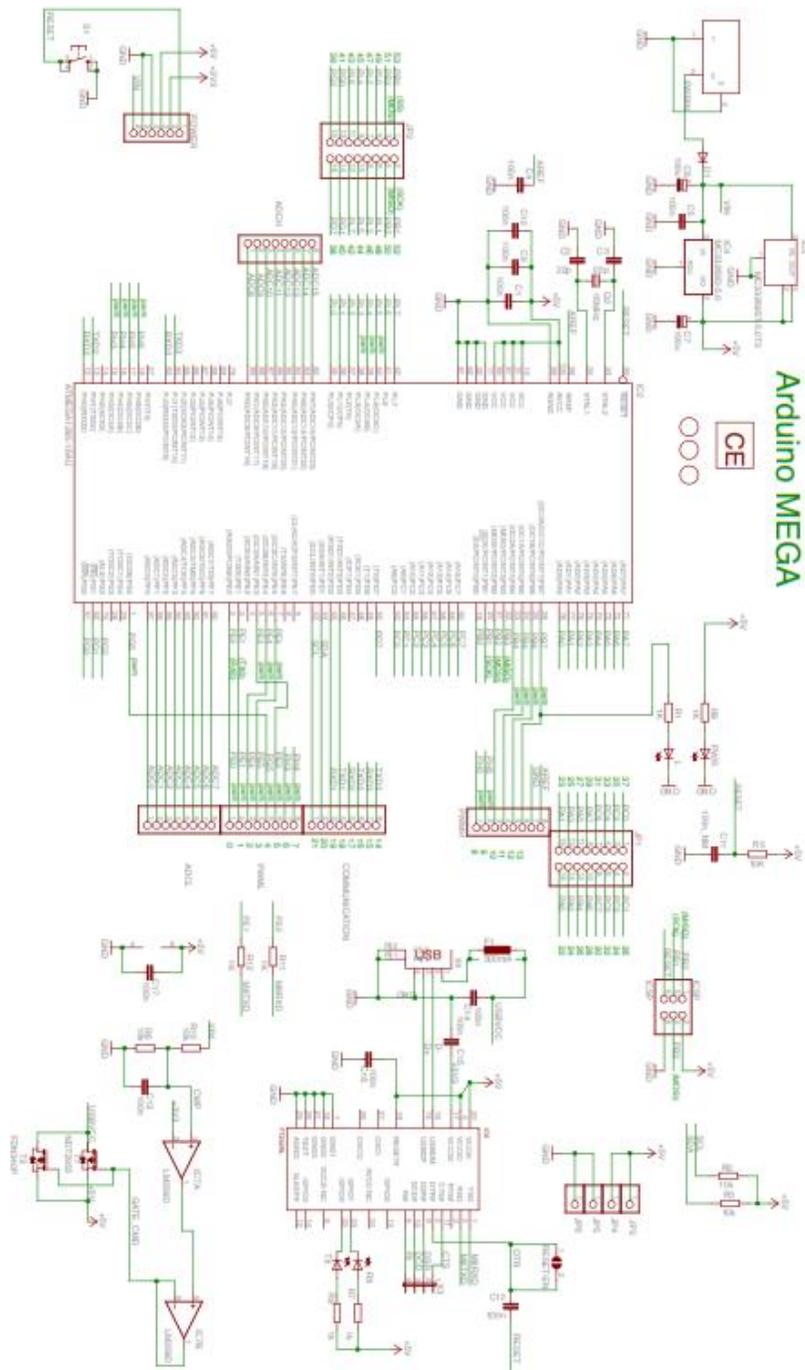


Fig. A6.1. Diagrama esquemático Arduino Mega

ANEXO 7

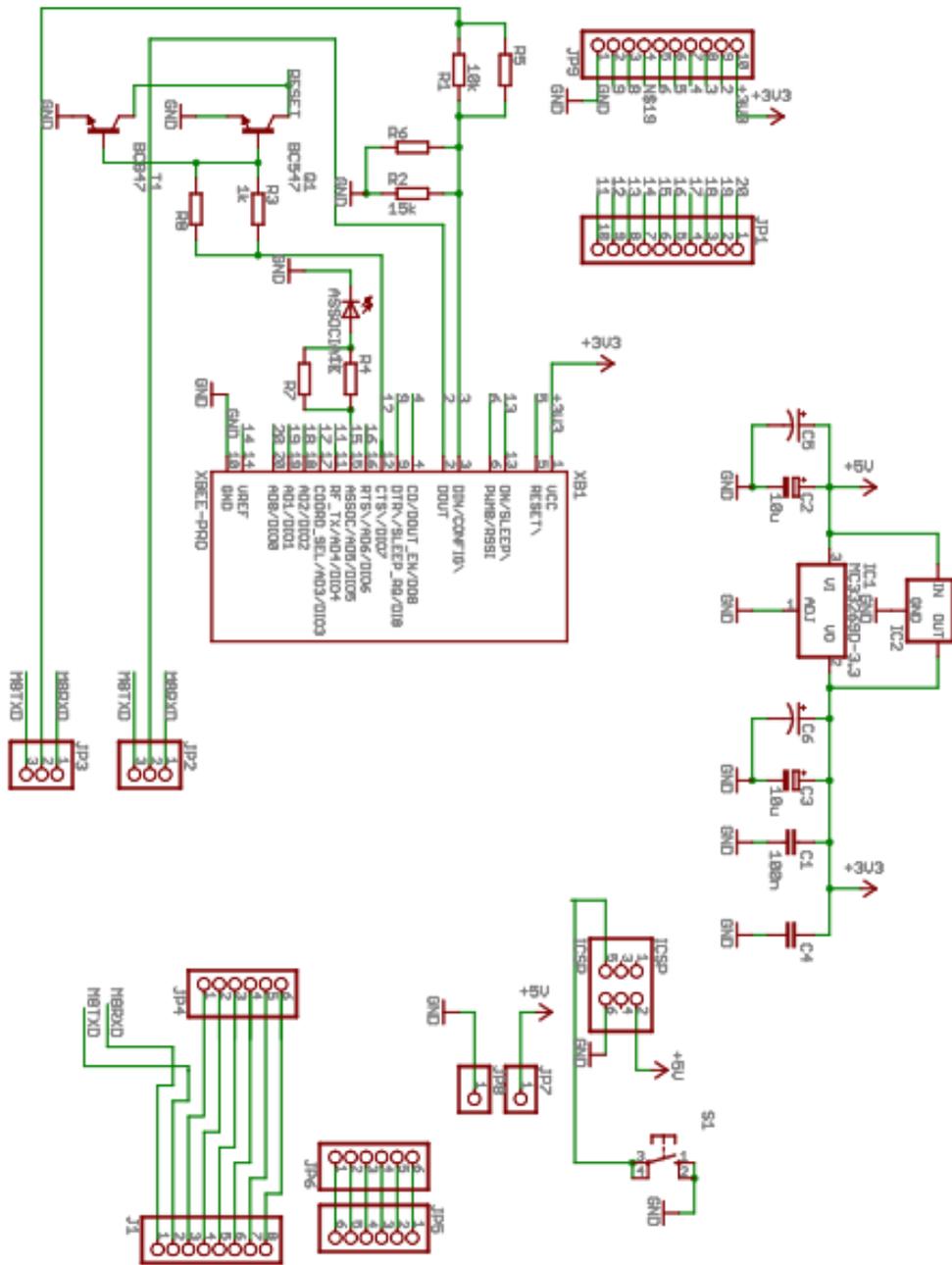


Fig. A7.1. Diagrama esquemático Xbee Shield

ANEXO 8

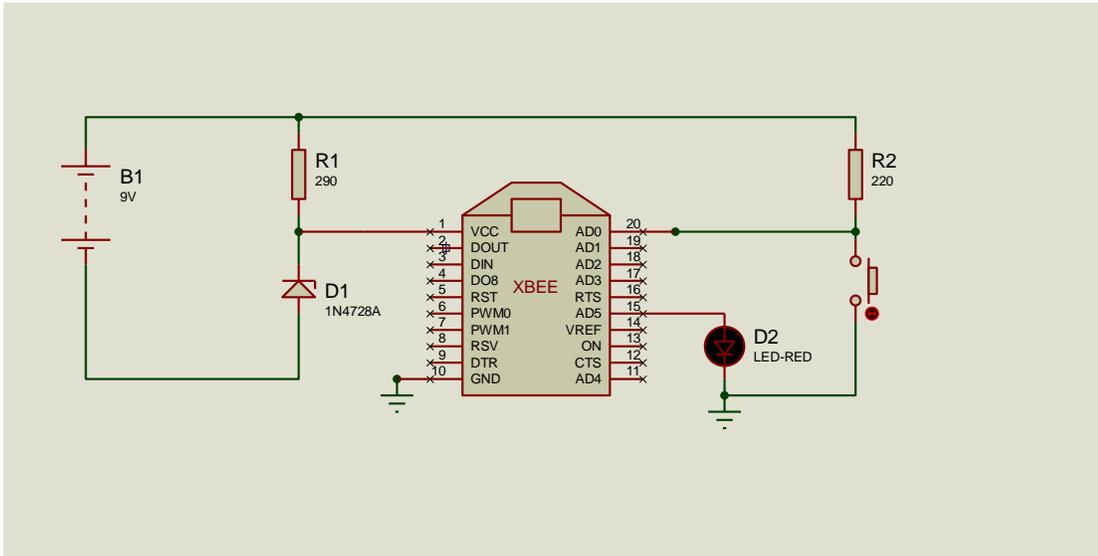


Fig. A8.1. Diagrama esquemático Nodo Fijo

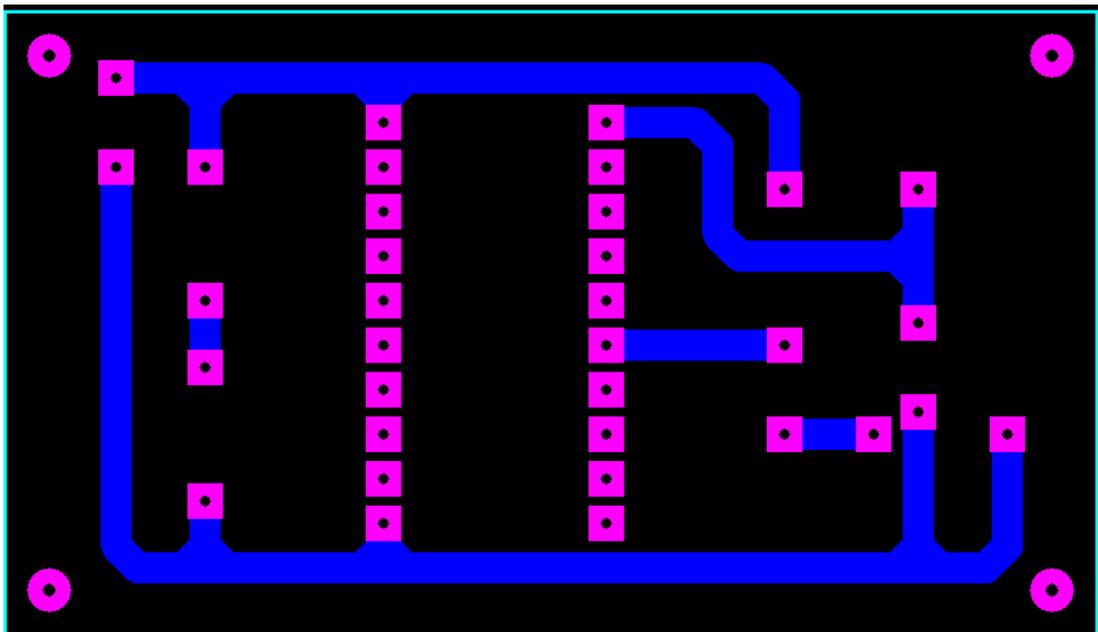


Fig. A8.2. Circuito Impreso Nodo Fijo

ANEXO 9

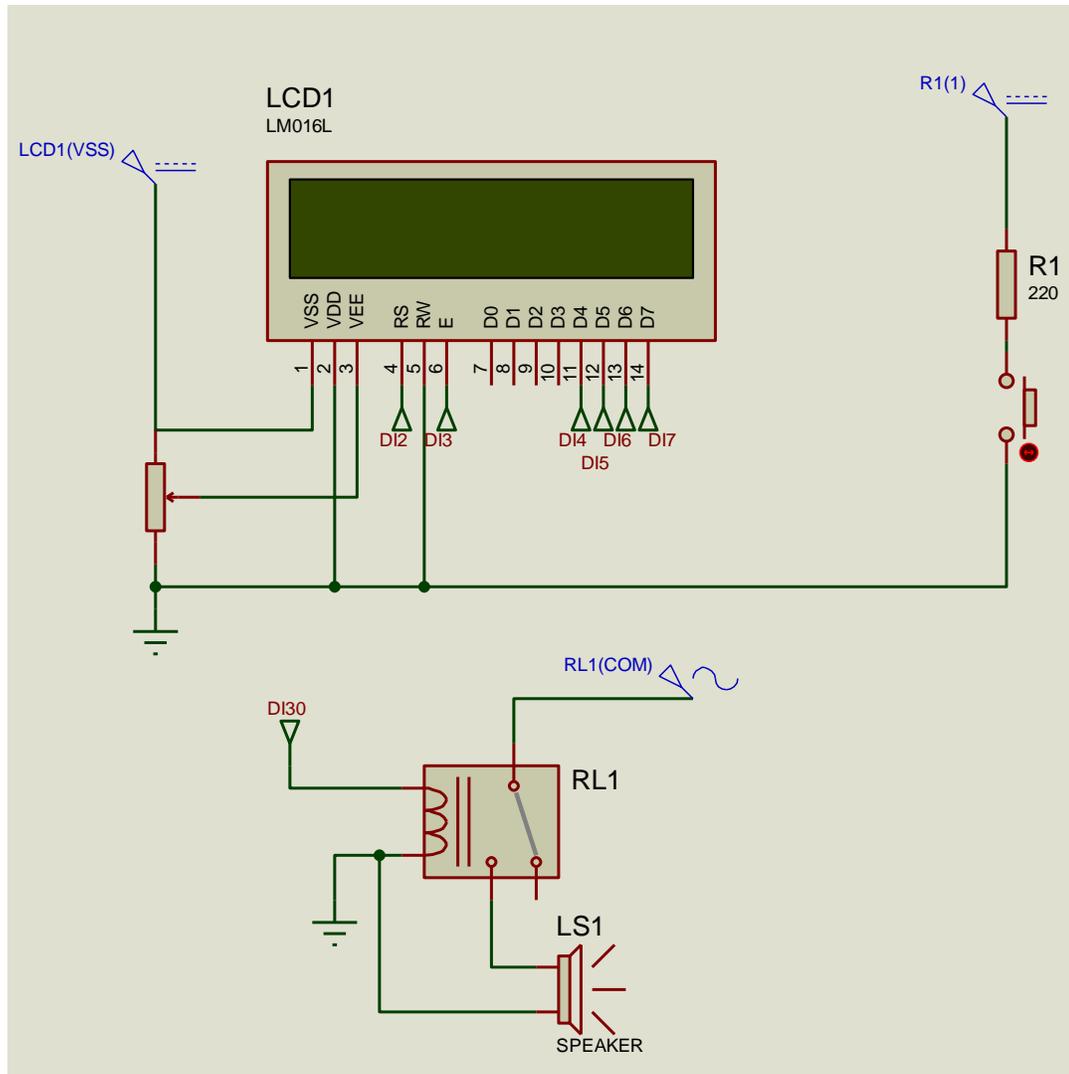


Fig. A9.2. Circuito esquemático elementos adicionales Estación Base

ANEXO 10

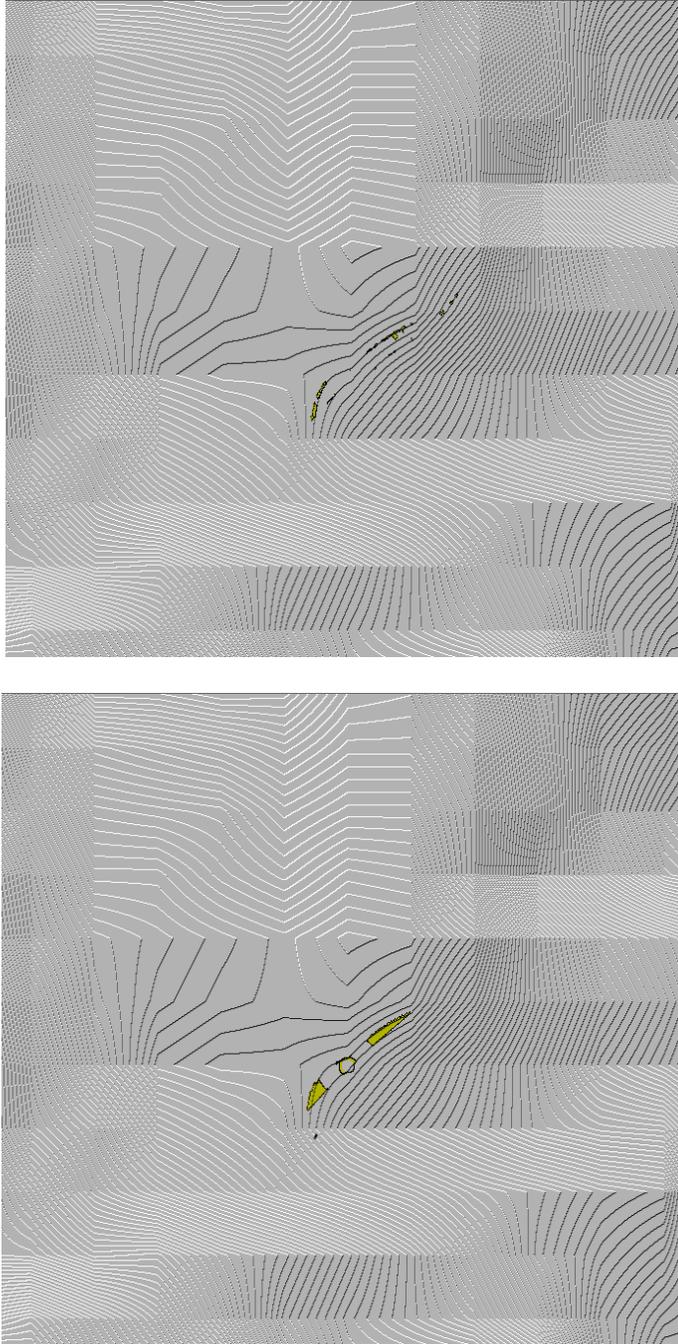


Fig. A10.1. Zonas de cobertura respecto a la estación base

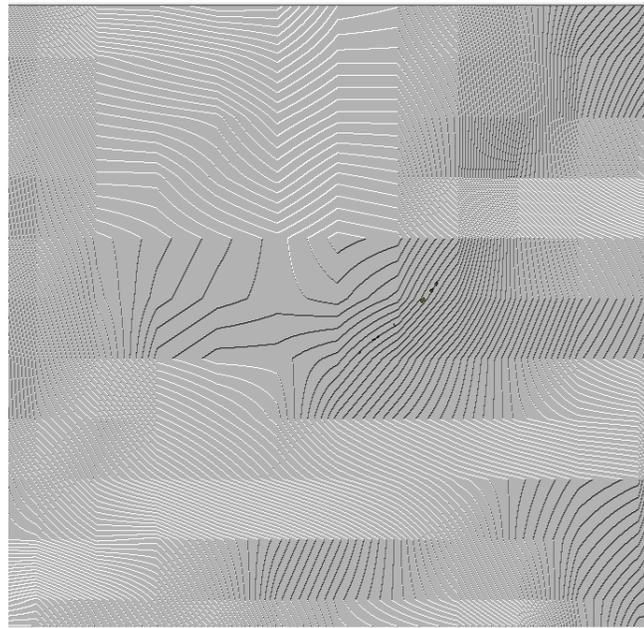
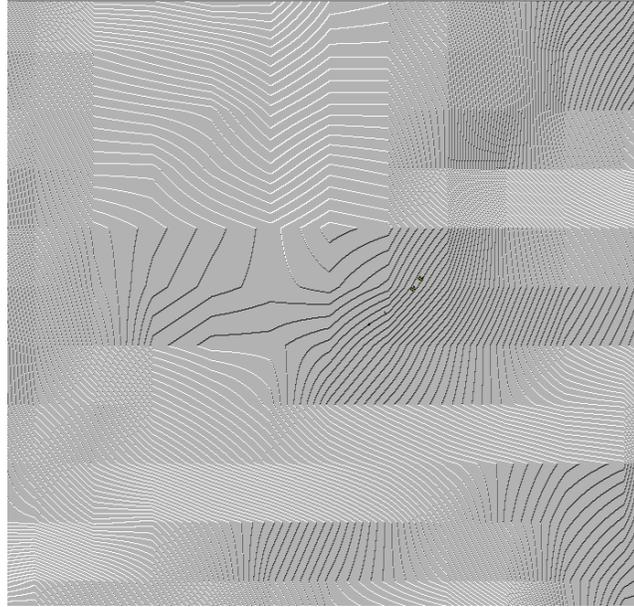


Fig. A10.2. Zonas de cobertura respecto a J. Ramírez.

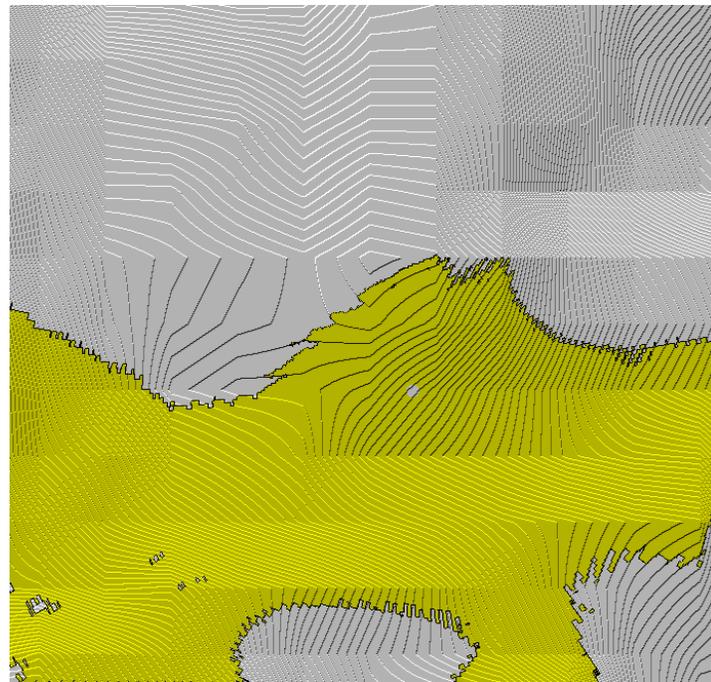
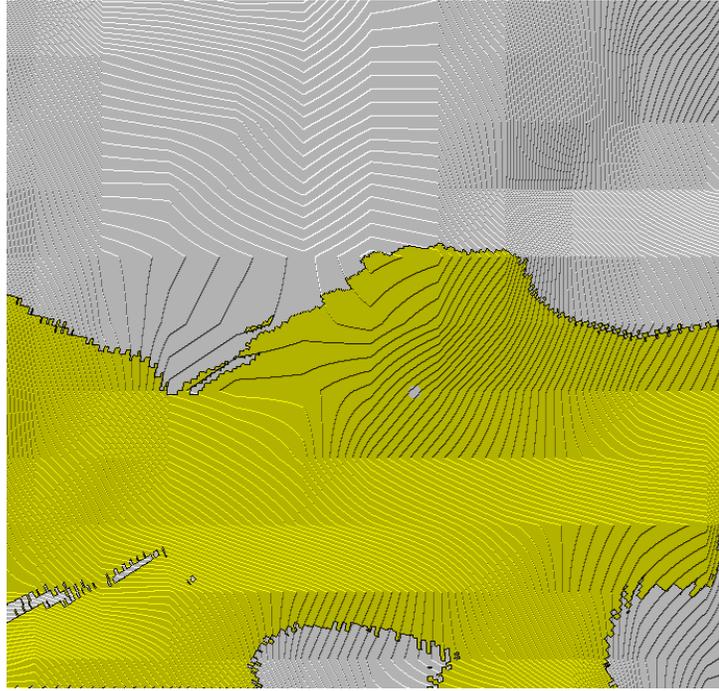


Fig. A10.3. Zonas de cobertura respecto a A. Ortíz.

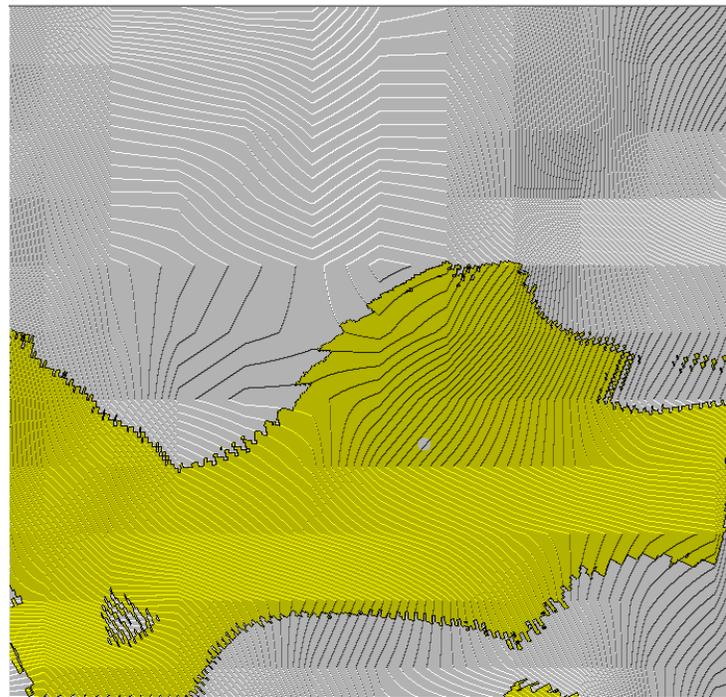
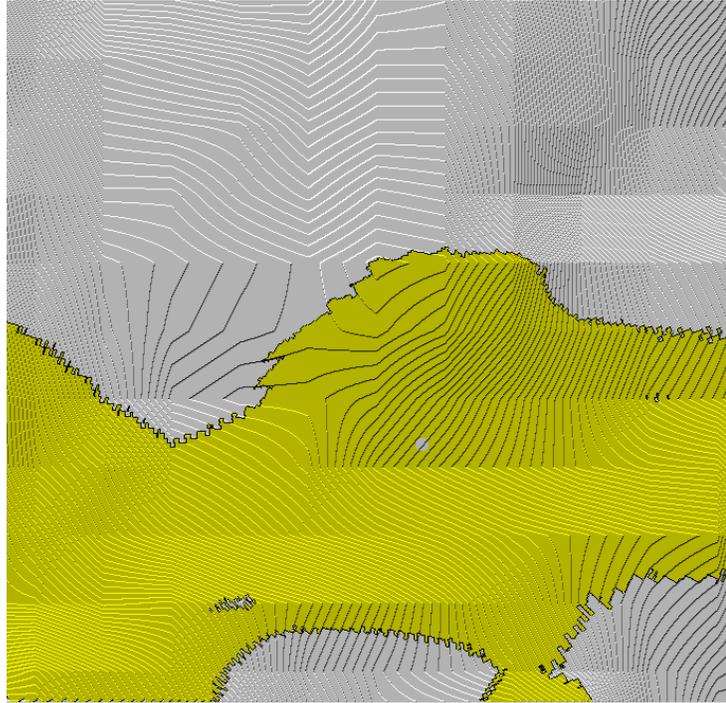


Fig. A10.4. Zonas de cobertura respecto a P. Llerena.



Fig. A10.5. Zonas de cobertura respecto a V. Chasi.



Fig. A10.6. Zonas de cobertura respecto a D. Granja.

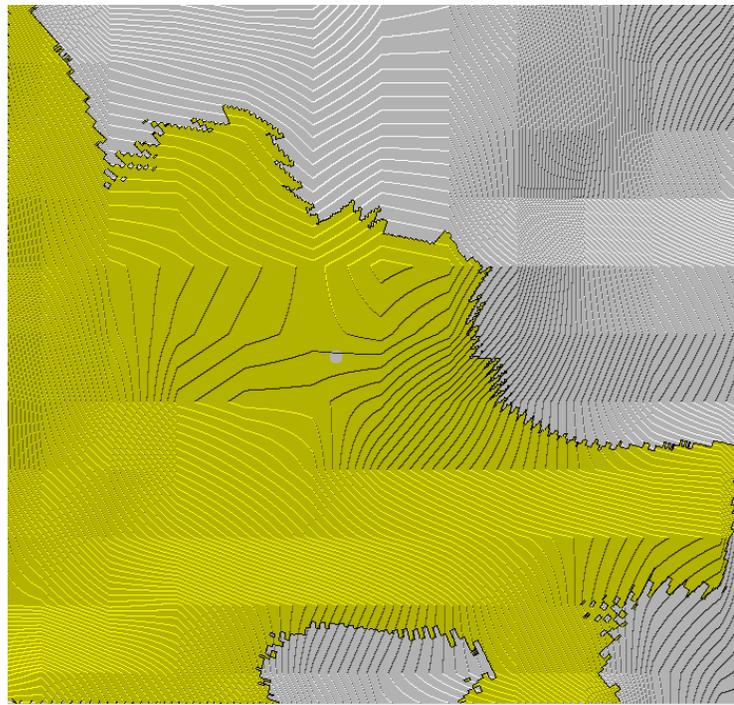


Fig. A10.7. Zonas de cobertura respecto a M. Sánchez.

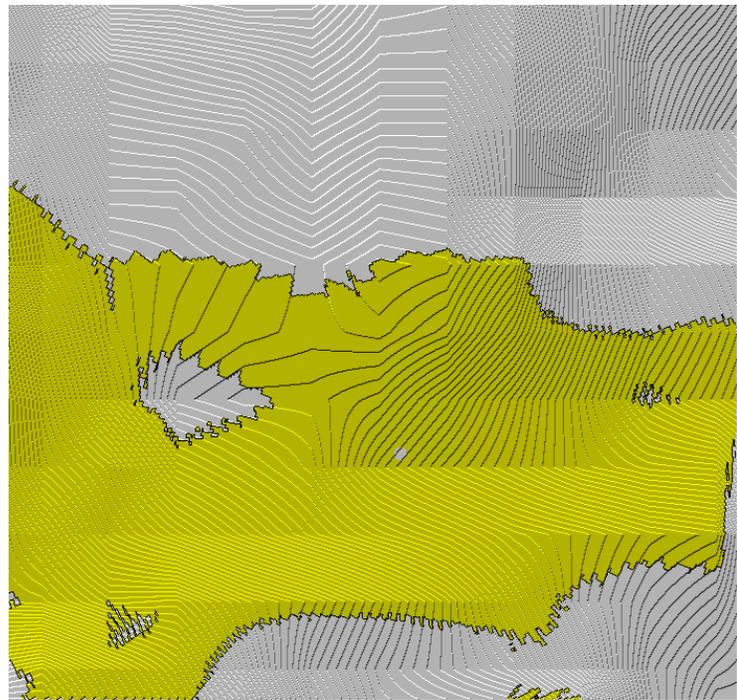
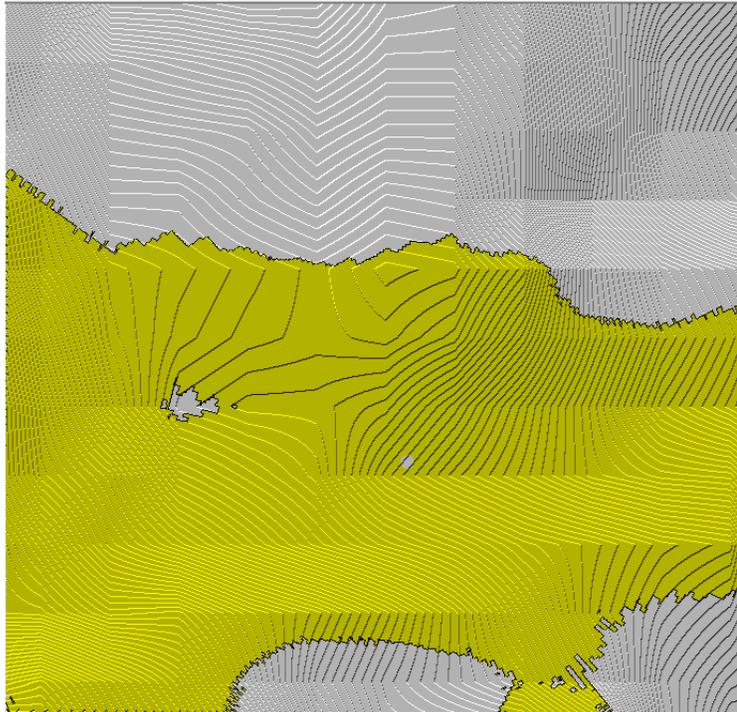


Fig. A10.8. Zonas de cobertura respecto a S. Guerrero.

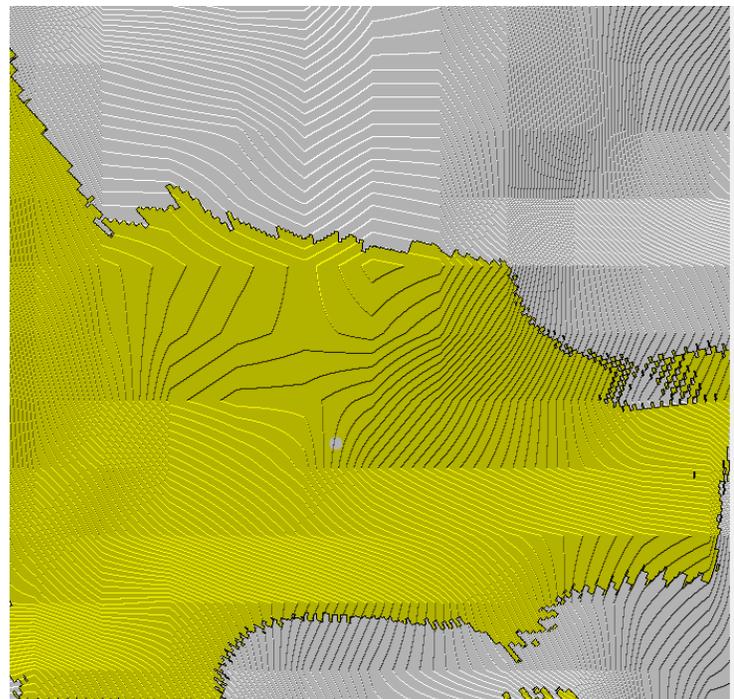


Fig. A10.9. Zonas de cobertura respecto a J. Vásquez.

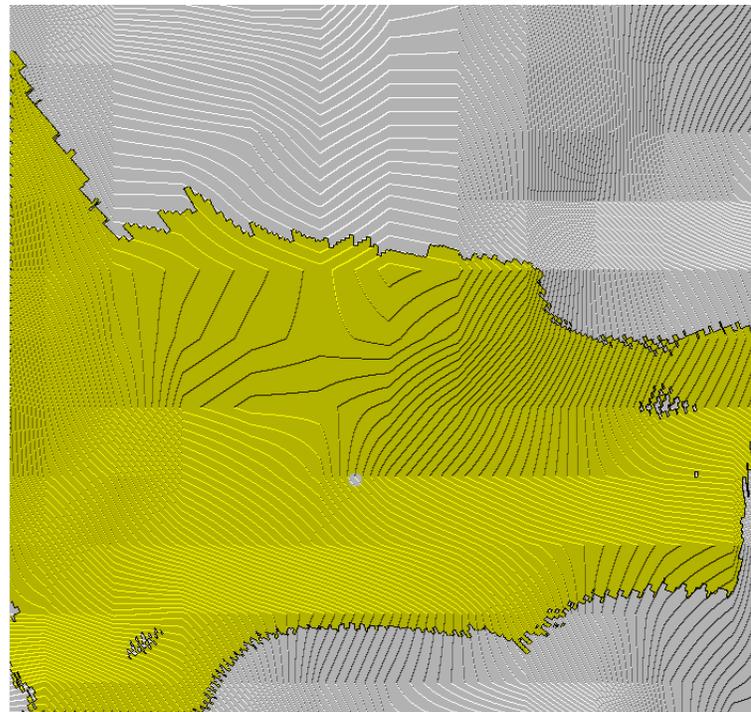
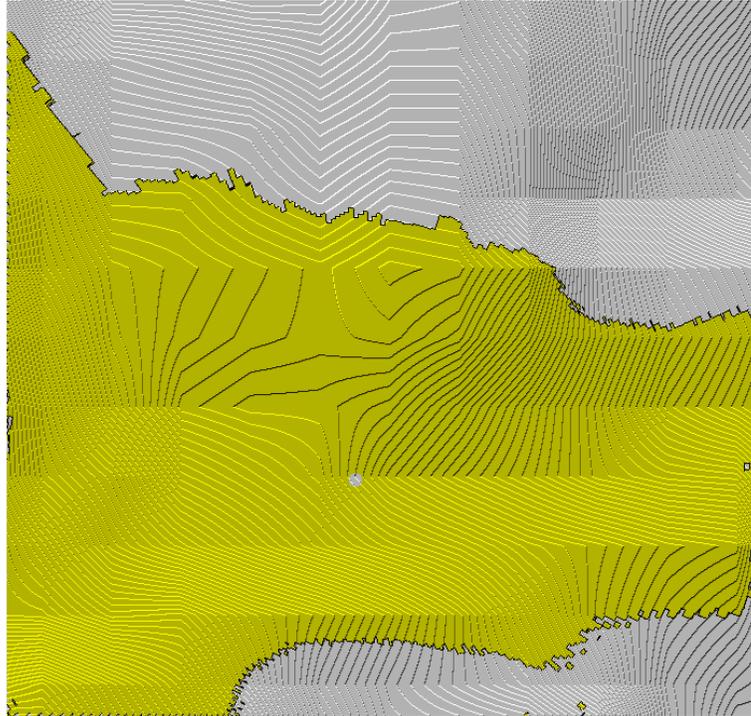


Fig. A10.10. Zonas de cobertura respecto a E. Jara.

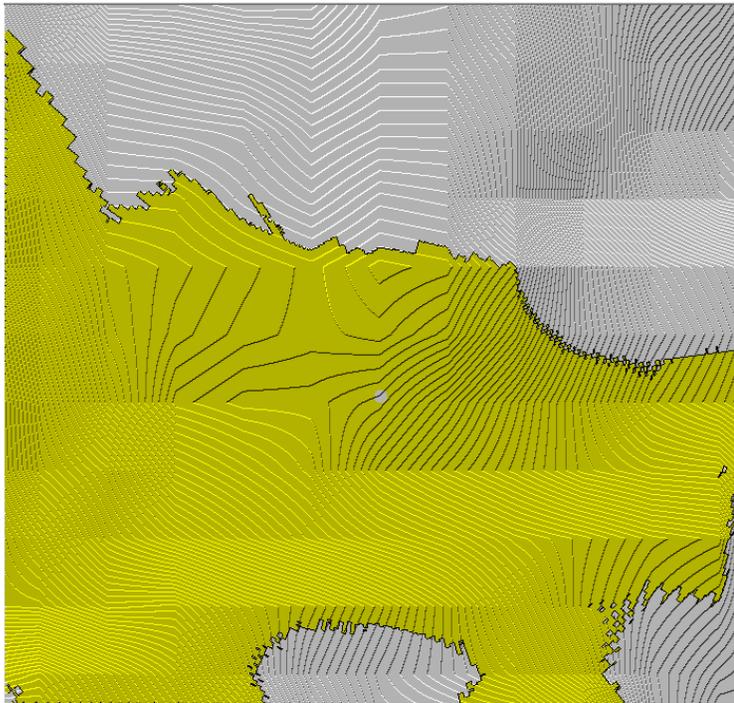


Fig. A10.11. Zonas de cobertura respecto a S. Mejía.

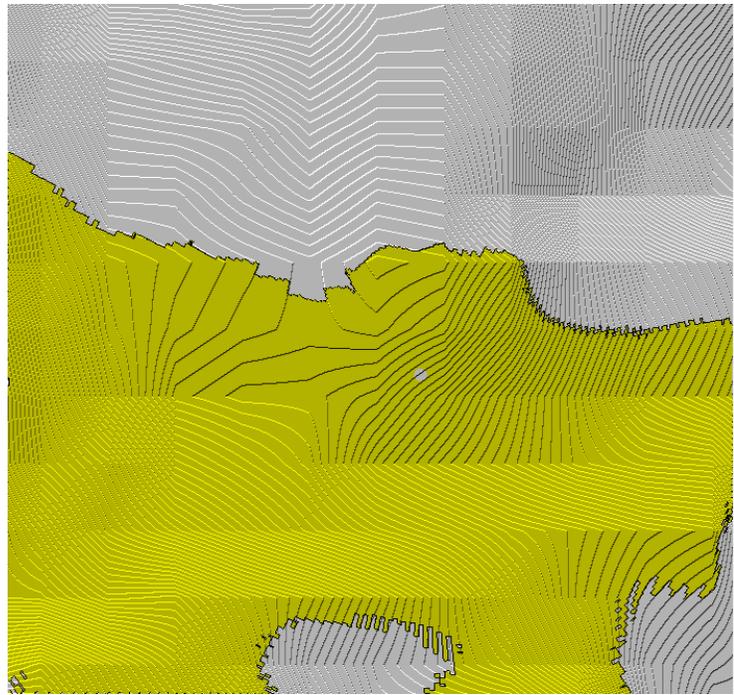
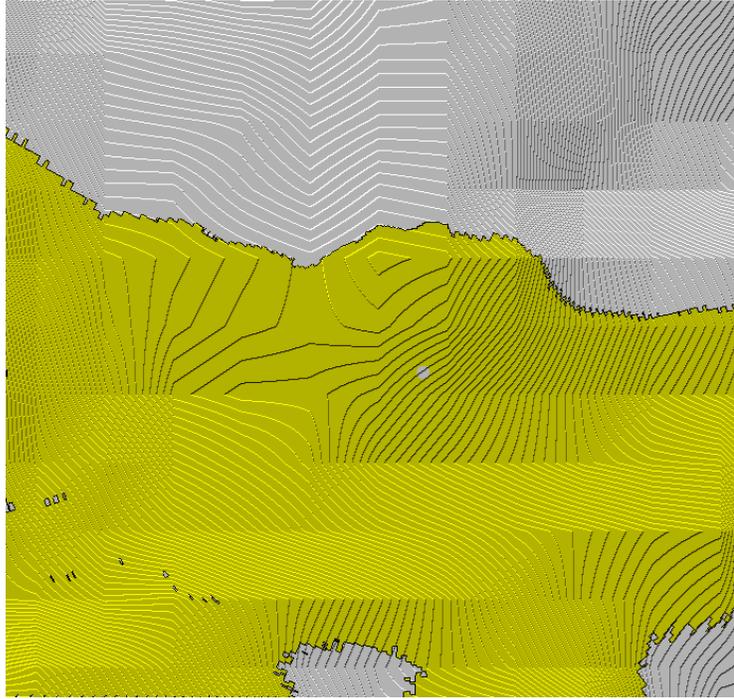


Fig. A10.12. Zonas de cobertura respecto a E. Miranda.

Latacunga, Diciembre del 2013.

AUTORÍA

ELABORADO POR:

Pérez Miranda Víctor Rafael

180418270-5

APROBADO POR:

Ing. José Bucheli
**DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN.**

CERTIFICADO POR:

Dr. Rodrigo Vaca
**SECRETARIO ACADÉMICO
UNIDAD DE ADMISIÓN Y REGISTRO.**