



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN

CARRERA: INGENIERÍA EN ELECTRÓNICA DIGITAL Y
TELECOMUNICACIONES

Tema: Implementación de un sistema de posicionamiento electrónico para el control de equipos y monitoreo de personas dentro de la empresa LR Telecomunicaciones

Autor: Nancy Vanessa Rodríguez Morán

Tutor: Ing. Tannia Mayorga Mg.

AÑO: 2015

UNIVERSIDAD TECNOLÓGICA ISRAEL

AUTORÍA DE PROYECTO INTEGRADOR DE CARRERA

Yo, Nancy Vanessa Rodríguez Morán, en calidad de estudiante de la Carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este Proyecto Integrador de Carrera, requisito previo a la obtención del Grado de Ingeniería en Electrónica Digital y Telecomunicaciones, son absolutamente originales y de exclusiva responsabilidad legal y académica del autor.

Quito D.M. febrero del 2015

Nancy Vanessa Rodríguez Morán
CC: 1720255866

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi aprobación de Tutor del Proyecto Integrador de Carrera certifico:

Que el proyecto integrador de carrera **“IMPLEMENTACIÓN DE UN SISTEMA DE POSICIONAMIENTO ELECTRÓNICO PARA EL CONTROL DE EQUIPOS Y MONITOREO DE PERSONAS DENTRO DE LA EMPRESA LR TELECOMUNICACIONES”**, presentado por la señorita Nancy Vanessa Rodríguez Morán, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. febrero del 2015

TUTOR

Ing. Tannia Mayorga Mg.

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de grado, aprueban el Proyecto Integrador Carrera de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para Títulos de Pregrado.

Quito D.M. febrero del 2015

Para constancia firma:

TRIBUNAL DE GRADO

PRESIDENTE

MIEMBRO 1

MIEMBRO 2

AGRADECIMIENTO

Este proyecto es el resultado del esfuerzo conjunto de todos los que formamos un gran grupo de trabajo. Por esto agradezco a mis profesores y mis compañeros, quienes han puesto a prueba mis capacidades y conocimientos a lo largo de este tiempo, el cual ha finalizado llenando todas las expectativas.

A mis padres quienes a lo largo de toda mi vida han apoyado y motivado mi formación académica, creyeron en mí en todo momento y no dudaron de mis habilidades.

A mis profesores a quienes les debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza; finalmente un eterno agradecimiento a esta prestigiosa universidad la cual abrió sus puertas a jóvenes como yo, preparándonos para un futuro competitivo, formándonos como personas de bien y como excelentes profesionales.

DEDICATORIA

La concepción de este proyecto está dedicada a mis padres, pilares fundamentales en mi vida. Sin ellos, jamás hubiese podido conseguir lo que hasta ahora he logrado. Su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar, no sólo para mí, sino para mis hermanos y familia en general. También dedico este proyecto a mi hija Nadeline, compañera inseparable de cada día. Ella representa todo mi esfuerzo y tesón en momentos de decline y cansancio. Para ellos este proyecto, que sin ellos, no hubiese podido ser.

AUTORÍA DE TRABAJO DE TITULACIÓN.....	I
APROBACIÓN DEL TUTOR.....	II
APROBACIÓN DEL TRIBUNAL DE GRADO.....	III
AGRADECIMIENTO.....	IV
DEDICATORIA.....	V

ÍNDICE

INTRODUCCIÓN.....	1
Antecedentes.....	1
Objetivos	
Objetivo Principal.....	3
Objetivos Secundarios.....	3
CAPÍTULO 1: Fundamentación teórica.....	4
1.1 Problema Principal.....	4
1.2 Marco Teórico.....	5
1.2.1 Metodología del posicionamiento.....	5
1.2.1.1 Método Huella Multitrayecto.....	5
1.2.1.2 Técnica de intensidad de señal recibida.....	6
1.2.1.3 Técnica tiempo de llegada.....	6
1.2.2 Protocolos y estándares de comunicación.....	7
1.2.2.1 IEEE 802.15.4.....	7
1.2.2.2 MiWi.....	7
1.2.3 Conclusiones de las técnicas, métodos y estándares descritos.....	8
1.2.4 Sistemas de posicionamiento.....	8
1.2.4.1 Radar.....	8
1.2.4.2 Identidad celular global (CGI).....	8
1.2.4.3 Identificación por radiofrecuencia (RFID).....	9
1.2.4.4 Sistema de posicionamiento Global (GPS).....	9
1.2.5 Conclusión de los sistemas existentes.....	10

CAPÍTULO 2: Diagnóstico del Problema.....	11
2.1 Hipótesis.....	11
2.2 Metodología.....	11
2.2.1 Proceso de Investigación.....	11
2.2.1.1 Unidad de análisis.....	11
2.2.1.2 Tipo de investigación.....	11
2.2.1.3 Método.....	12
2.2.1.4 Técnica.....	12
2.2.1.5 Instrumento.....	12
2.3 Resultados que se espera del proyecto.....	12
CAPÍTULO 3: Presentación de resultados.....	14
3.1 Introducción.....	14
3.2 Dispositivo Base.....	15
3.2.1 Funcionamiento del dispositivo Base.....	15
3.2.2 Elementos del dispositivo Base.....	16
3.2.2.1 Módulo de transmisión/recepción MRF24J40MA.....	18
3.2.2.2 Microcontrolador PIC16F887.....	19
3.2.2.3 Regulador de energía.....	21
3.2.2.4 Circuito Integrado MAX232	22
3.3 Dispositivo Móvil.....	23
3.3.1 Funcionamiento del dispositivo móvil.....	24
3.3.2 Elementos del dispositivo móvil.....	24
3.4 Aplicación.....	26
3.5 Validación de la implementación.....	28
3.6 Estudio de Factibilidad.....	29
3.6.1 Factibilidad Técnica.....	29
3.6.2 Factibilidad Operativa.....	29
3.6.3 Factibilidad Económica.....	30
3.7 Evaluación de la implementación por parte de la Empresa.....	31
CONCLUSIONES Y RECOMENDACIONES.....	34
BIBLIOGRAFÍA.....	36

ANEXOS.....	37
Anexo N.1 Fotografías de los dispositivos electrónicos.....	38
Anexo N.2 Código de programación del dispositivo Base.....	42
Anexo N.3 Código de programación del dispositivo móvil.....	57
Anexo N.4 Código de programación de la aplicación.....	71
ÍNDICE DE FIGURAS Y GRÁFICOS.....	IX
ÍNDICE DE TABLAS.....	X
ÍNDICE DE ECUACIONES.....	XI

INDICE DE FIGURAS

Figura N. 1: Ubicación macro del proyecto a realizarse.....	2
Figura N. 2: Ubicación micro de la Empresa LR Telecomunicaciones.....	3
Figura N. 2.1: Diagrama del sistema.....	13
Figura N. 3.1: Diagrama de bloques del sistema de rastreo.....	14
Figura N. 3.2: Gráfico del dispositivo Base.....	15
Figura N.3.3: Diagrama esquemático del dispositivo Base.....	17
Figura N. 3.4: Módulo RadioTransceiver MRF24J40MA.....	18
Figura N. 3.5: Programa para configurar el Módulo MRF24J40MA.....	19
Figura N. 3.6: Pantalla principal del programa Terminal.....	19
Figura N. 3.7: Microcontrolador PIC16F887.....	20
Figura N. 3.8: Diagrama de pines del PIC16F887.....	21
Figura N. 3.9: Diagrama de la parte de suministro de energía.....	21
Figura N. 3.10: Diagrama de pines del MAX232.....	22
Figura N. 3.11: Diagrama de pines de la pantalla LCD.....	23
Figura N. 3.12: Gráfico del dispositivo móvil.....	23
Figura N. 3.13: Diagrama esquemático del dispositivo móvil.....	25
Figura N. 3.14: Interfaz gráfica de la aplicación en Visual Basic.....	26
Figura N. 3.15: Interfaz gráfica de la aplicación al ejecutarla.....	27
Figura N. 3.16: Interfaz gráfica de la aplicación mostrando la zona de cada dispositivo móvil	28

INDICE DE TABLAS

Tabla N. 3.1: Características del PIC18F4620.....	20
Tabla N. 3.2: Consumo aproximado de energía.....	22
Tabla N. 3.3: Precio del dispositivo electrónico móvil.....	30
Tabla N. 3.4: Precio del dispositivo electrónico fijo.....	30
Tabla N. 3.5: Resumen de precios de los componentes del sistema.....	30

INDICE DE ECUACIONES

Ecuación N. 1.1: Ecuación de la Técnica tiempo de llegada (TOA).....	5
Ecuación N. 3.1: Fórmula de la muestra para el resultado de encuestas.....	31

INTRODUCCIÓN

El proyecto se realizará en Ecuador, en la provincia de Pichincha, en la ciudad de Quito, en el sector de Monjas, en el Barrio Alma Lojana donde se encuentra la Empresa LR Telecomunicaciones de propiedad del Sr. Luis Rodríguez.

Dicha empresa se dedica a la venta, instalación, mantenimiento y programación de centrales telefónicas, porteros eléctricos, video portero, cámaras de seguridad; también realizan instalaciones de redes de datos, de cableado estructurado y cableado eléctrico.



Figura N. 1: Ubicación macro del proyecto a realizarse.

(Fuente: Google Maps)

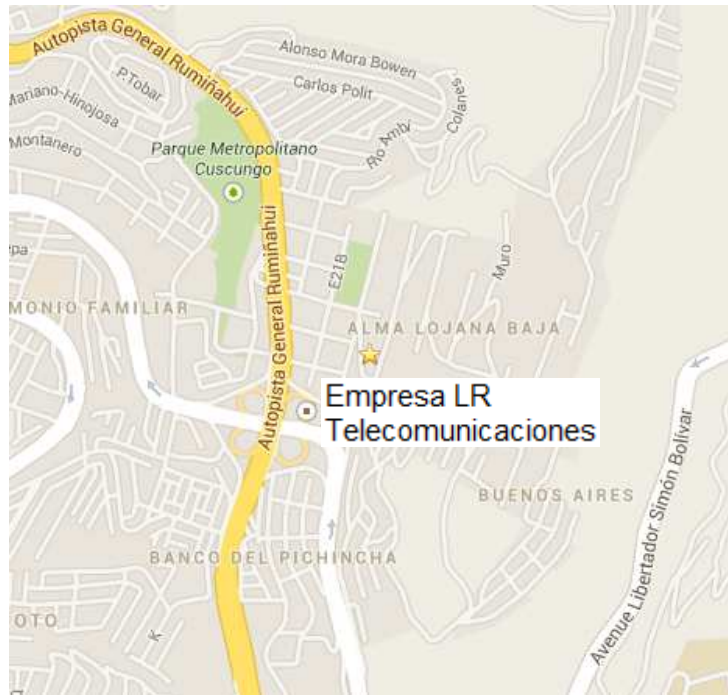


Figura N. 2: Ubicación micro de la Empresa LR Telecomunicaciones
(Fuente: Google Maps)

Al finalizar el proyecto se tendrá un producto el cual será un sistema de rastreo interno que contendrá dispositivos electrónicos inalámbricos que constan de una base (fijo) y varias tarjetas (móvil), los cuales se ubicarán en las instalaciones de la empresa y se podrá rastrear con una aplicación a la persona u objeto que lleve puesto la tarjeta y así poder monitorear los movimientos y el lugar exacto de donde se encuentran.

El impacto tecnológico que tendrá este nuevo sistema será en el ámbito de la seguridad porque ningún sistema que existe actualmente en el mercado logra monitorear a las personas dentro de las oficinas.

El principal problema de la Empresa LR Telecomunicaciones es que necesitan controlar al personal que trabaja dentro de la empresa y a los equipos de cómputo, por razones de seguridad (pérdidas: materiales, objetos de valor e información).

Dentro del país no se tiene información de un sistema de posicionamiento que permita rastrear y determinar las coordenadas exactas de dispositivos en un área cerrada con relación a un punto de referencia.

Fuera del país existe un sistema de registro y control de salida de elementos mediante la tecnología de identificación por radiofrecuencia conocida comúnmente como RFID; también existe un sistema que utiliza tecnología inalámbrica como es el Bluetooth en dispositivos móviles parecido a lo que realiza el sistema de posicionamiento global GPS y también existe un sistema que se lo realizó en Madrid mediante la tecnología Bluetooth por el estudiante Francisco José Sánchez de la Universidad de Madrid en Julio del 2010.

Objetivos

Objetivo general

Implementar un sistema de posicionamiento electrónico para el monitoreo de personas y equipos de cómputo dentro de la Empresa LR Telecomunicaciones, mediante la realización de dispositivos electrónicos inalámbricos.

Objetivos específicos

- Investigar el funcionamiento del módulo de transmisión/recepción que consta de un circuito integrado MRF24J40MA y realizar las pruebas respectivas.
- Implementar el módulo de microcontrolador que consta de un PIC, realizar la programación respectiva y verificar su funcionamiento.
- Implementar el módulo de energía y realizar las pruebas de administración de carga de la batería.
- Conectar todos los módulos, verificar su funcionamiento y realizar las pruebas mediante la pantalla LCD que se encuentra en el dispositivo electrónico fijo.
- Realizar la aplicación, la cual se encuentra en un computador y se encarga de mostrar donde se ubican los dispositivos electrónicos móviles.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.1. Problema Principal

La empresa LR Telecomunicaciones requiere monitorear los movimientos exactos de las personas dentro de sus instalaciones, para tener un mejor control del personal y de los objetos que la empresa posee.

Dentro de la empresa el principal problema es la seguridad como por ejemplo: personal infiltrado y pérdidas de objetos de valor e información.

Al realizar la investigación de los sistemas de seguridad y de rastreo interno dentro del mercado nacional no se encuentran sistemas de seguridad que brinden la ubicación de personas, referente a un punto o a un mapa dentro de una casa o una empresa; mucho menos sea mediante tecnología inalámbrica como es la radiofrecuencia.

Por tal motivo se necesita un sistema de seguridad y rastreo que cumpla con las necesidades que exige la empresa y esto se logrará mediante la implementación de dispositivos inalámbricos que se podrán poner en los computadores o lo llevarán las personas en un gafete para poder ubicarlos mediante una aplicación a la que tendrá acceso el personal de seguridad y así lograr mejorar la seguridad de la empresa.

Con este nuevo sistema de rastreo interno y seguridad se podrá controlar al personal propio y ajeno que ingresen a las instalaciones de la empresa LR Telecomunicaciones, garantizando así que no existan pérdidas tanto materiales como de la información que se posee en la empresa; también se mejorará el control del personal pudiendo saber el lugar donde se encuentre cada persona.

1.2. Marco Teórico

1.2.1. Metodología del posicionamiento

A los sistemas de posicionamiento se los separa en dos terminologías como son: métodos y técnicas.

- **Métodos:** establece la forma matemática en que un sistema de posicionamiento determina la ubicación de un objeto en el espacio, usando a otros como referencia conociendo y determinando ángulos y distancias.
- **Técnicas:** definen el proceso real para determinar ángulos y distancias entre dispositivos por radiofrecuencia.

1.2.1.1. Método huella multitrayecto (MF)

Este método aprovecha una de las perturbaciones que son molestas cuando se quiere localizar un transmisor, las señales multitrayecto.

La huella multitrayecto es un método que caracteriza las señales que llegan desde diferentes localizaciones. Esto se logra enviando unidades de prueba a distintos lugares con el fin de que las estaciones base graben las huellas multitrayecto y creen una base de datos para efectuar comparaciones.

Mediante este método se puede aplicar cualquier técnica donde se pueda censar algún parámetro del transmisor que se quiera ubicar, y almacenar los resultados obtenidos asociados a las posiciones reales.

La precisión de este método depende de la cantidad de dispositivos fijos que se coloquen en el área a controlar.

1.2.1.2. Técnica de intensidad de señal recibida (RSS)

La técnica de intensidad de señal recibida es llamada también "Received Strength Signal", mediante esta técnica se puede medir la intensidad de la señal entrante y así poder estimar la distancia a la que se encuentra, pero debido a que la intensidad puede afectarse fácilmente por obstáculos o interferencias, se usa para corroborar la información obtenida por otra técnica.

La técnica de intensidad de señal recibida también permite detectar problemas de pérdida de señal o interferencia.

La implementación de esta técnica se la debe realizar conjuntamente con el método de huella multitrayectoria, debido a que la relación entre la pérdida de una señal electromagnética en el espacio libre respecto a la distancia que recorre no es lineal, si no logarítmica.

1.2.1.3. Técnica tiempo de llegada (TOA)

La técnica Tiempo de Llegada (TOA) consiste en medir el tiempo que demora una señal electromagnética en viajar al objetivo y regresar, lo que nos permitirá calcular la distancia al mismo.

Esta técnica es la más eficiente, ya que los obstáculos e interferencia afectan muy poco la velocidad de la señal. Su fórmula se detalla a continuación.

$$D = C * T_r \quad \text{Ecuación (1.1)}$$

D Distancia [m]

C Velocidad de la luz= 3×10^8 [m/s]

T_r Tiempo de retardo [s]

En la fórmula 1.1, el tiempo de vuelo de una onda electromagnética en un metro es de 3.33 ns por lo que el retardo a medir es en el orden de los nanosegundos [ns], si se requiere precisión se lo realiza en centímetros y así se delimita que la frecuencia este en los GHz para la señalización del sistema.

1.2.2. Protocolos y estándares de comunicación

1.2.2.1. Protocolo para redes Wireless IEEE 802.15.4

El IEEE 802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos. **(Moretto, 2008, pág. 8)**

Define los niveles de red básicos para dar servicio a un tipo específico de red inalámbrica de área personal (WPAN) la cual se centra en la habilitación de comunicación entre dispositivos con bajo costo y velocidad.

Básicamente se concibe un área de comunicación de 10 metros con una tasa de transferencia de 250 Kbps.

Los aspectos más importantes son: la adecuación de su uso para tiempo real por medio de slots de tiempo garantizados, evasión de colisiones por CSMA/CA (acceso múltiple por detección de portadora con evasión de colisiones) y soporte integrado a las comunicaciones seguras. También incluye funciones de control del consumo de energía como calidad del enlace y detección de energía.

1.2.2.2. Protocolo MiWi

El protocolo MiWi es un protocolo inalámbrico, para redes de área personal basado en el estándar IEEE 802.15.4 que engloba las redes de área personal con tasas bajas de transmisión de datos. **(Moretto, 2008, pág. 17)**

Dispone de dispositivos de transferencia pequeños y de poco consumo de energía, opera en la frecuencia de los 2.4 GHz lo que permite la propagación de altas frecuencias en áreas cerradas.

Utiliza el microcontrolador MRF24J40 que tienen una alta transferencia de datos (250Kbit/s), a corta distancia (50 metros sin obstáculos).

1.2.3. Conclusión de las técnicas, métodos y estándares descritos.

De todos los métodos y técnicas la más factible es la técnica de intensidad de señal recibida (RSS) conjuntamente con el método de huella de multitrayecto, ya que permite utilizar circuitos transmisores/receptores existentes en el mercado.

La alternativa factible y económica es mediante el estándar MiWi definido por la IEEE 802.15.4 para redes inalámbricas de baja potencia y transferencia de datos, lo que dispone de dispositivos de transferencia pequeños y de poco consumo.

1.2.4. Sistemas de posicionamiento

Como parte de la investigación se explicará los sistemas de posicionamiento y rastreo existentes en el mercado nacional e internacional. **(Río, 2008, pág. 46)**

1.2.4.1. Radar

Este sistema ocupa la técnica de tiempo de llegada donde el transmisor emite ondas electromagnéticas a través de una antena que concentran las ondas en un haz coherente apuntando en la dirección deseada. **(Paramvir & Venkata, 2000)**

Cuando las ondas chocan con un objeto que se halla en la trayectoria del haz, algunas se reflejan y forman una señal de eco. La antena capta la energía contenida en dicha señal y la envía al receptor. Mediante un proceso de amplificación y tratamiento informático el receptor del radar genera una señal en el dispositivo de visualización.

Por lo tanto este tipo de posicionamiento no es factible para el proyecto que se está realizando porque su rastreo es en exteriores.

1.2.4.2. Identidad celular global (CGI)

La identidad celular global está disponible sin realizar ninguna conversión ni modificación en red o terminales, pues la posición se la obtiene identificando la celda en donde esté el terminal móvil.

Este sistema ubica cualquier tipo de dispositivo móvil en las redes: GSM, GPRS, UMTS y CDMA.

Esta técnica identifica la célula en la que está el terminal móvil; es una estimación de la distancia desde el terminal móvil a la estación base.

La exactitud de este método depende del radio de la celda, que puede variar de 50 m en áreas urbanas a 3-4 km en áreas rurales. Este sistema lo utilizan los operarios por ofrecer servicios en entornos urbanos.

1.2.4.3. Identificación por radiofrecuencia (RFID)

La identificación por radiofrecuencia es una denominación genérica para las personas que utilizan ondas de radio para identificar automáticamente personas u objetos.

Se puede implementar mediante el conocimiento de la ubicación del dispositivo en un área determinada, pero no se puede definir la posición exacta con coordenadas, ni corroborar que realmente el RFID está en esa área.

En conclusión este sistema no ubica a los dispositivos móviles con coordenadas exactas, si no que estima su posición por los lectores en los que se haya registrado.

1.2.4.4. Sistema de posicionamiento Global (GPS)

El GPS es un sistema de radionavegación mundial que emplea al menos cuatro de los satélites que rodean la Tierra en órbitas conocidas para fijar la posición de una unidad móvil. **(Rabbany, 2002)**

El sistema GPS está formado por tres segmentos:

- El segmento espacial: satélites.
- El segmento de control: infraestructuras terrestres.
- El segmento de usuario: equipos de recepción y el software de procesos de señales.

La localización de GPS se lo realiza mediante el cálculo de un punto fijo terrestre con la triangulación de tres satélites con posición perfectamente conocida.

En conclusión el GPS no se lo puede integrar el sistema en posicionamiento en interiores con la localización satelital. Y su dificultad es la extrema precisión de sus dispositivos.

1.2.5. Conclusión de los sistemas existentes

Se puede concluir entonces que los sistemas de posicionamiento existentes en el mercado, ninguno se adapta a la aplicación que se desea realizar por varias razones como por ejemplo, los sistemas comunes no ubican a los dispositivos móviles con coordenadas exactas, sino que sólo estiman su posición por los lectores en los que se haya registrado y porque los sistemas existentes tienen la dificultad de rastrear con precisión a dispositivos móviles en interiores de los edificios.

CAPÍTULO II

DIAGNÓSTICO DEL PROBLEMA

2.1. Hipótesis

¿El sistema de posicionamiento electrónico mejorará el control de equipos y monitoreo de personas dentro de la Empresa LR Telecomunicaciones?

¿El sistema de posicionamiento electrónico no mejorará el control de equipos y monitoreo de personas dentro de la Empresa LR Telecomunicaciones?

2.2. Metodología

2.2.1. Proceso de Investigación

2.2.1.1. Unidad de Análisis

La implementación del sistema de rastreo interno se lo realizará en la Empresa LR Telecomunicaciones, ubicada en la Provincia de Pichincha, en la ciudad de Quito, en el sector de Monjas, en el Barrio Alma Lojana calle E S3-126, número de teléfono 022605146.

Empresa dedicada a la venta, instalación, mantenimiento y programación de centrales telefónicas marca Panasonic, porteros eléctricos, video portero, cámaras de seguridad; también realizan instalaciones de redes de datos, de cableado estructurado y cableado eléctrico, que necesitan controlar al personal mediante un sistema que identifique con exactitud la posición del personal dentro de la empresa.

2.2.1.2. Tipo de Investigación

El tipo de investigación que se utilizó es la Aplicada, porque es aquella investigación relacionada con la generación de conocimientos en forma de teoría o métodos que en un futuro podrían desembocar en aplicaciones al sector productivo.

Dicha investigación ocupa los conocimientos adquiridos en los años de estudio como son la Electrónica, las Redes de Datos, el Diseño de Circuitos Integrados, etc., para aplicarlos en provecho de la sociedad como es el caso de la Empresa LR Telecomunicaciones.

2.2.1.3. Método

El método a utilizar es el Deductivo porque parte de verdades previamente establecidas como principios generales, para luego aplicarlo a casos individuales y comprobar así su validez; es decir el proyecto parte de una investigación general de varios sistemas de posicionamiento existentes para llegar a la conclusión de que se necesita realizar un nuevo sistema de posicionamiento que rastree a las personas u objetos.

2.2.1.4. Técnica

El procedimiento para obtener datos o información se lo realizó mediante la observación directa, para esto, se utilizó: textos, ensayos, tesis y fuentes bibliográficas.

2.2.1.5. Instrumento

Se aplicará la técnica de recolección de datos mediante los siguientes instrumentos:

- Libreta de borrador
- Datasheet de los elementos
- Tesis relacionada con el tema
- Libros referenciados con el tema

2.3. Resultados que se espera del proyecto

Con el dispositivo diseñado la empresa podrá controlar al personal y a las personas que ingresen a sus instalaciones mediante dispositivos electrónicos, los cuales permitirán controlar y evitar el paso a lugares restringidos; también se reforzará el sistema de seguridad de la empresa para solucionar las pérdidas de: materiales, objetos e información.

Así se podrá tener en la empresa un sistema que brinde la ubicación de las personas dentro de una instalación mediante la tecnología inalámbrica.

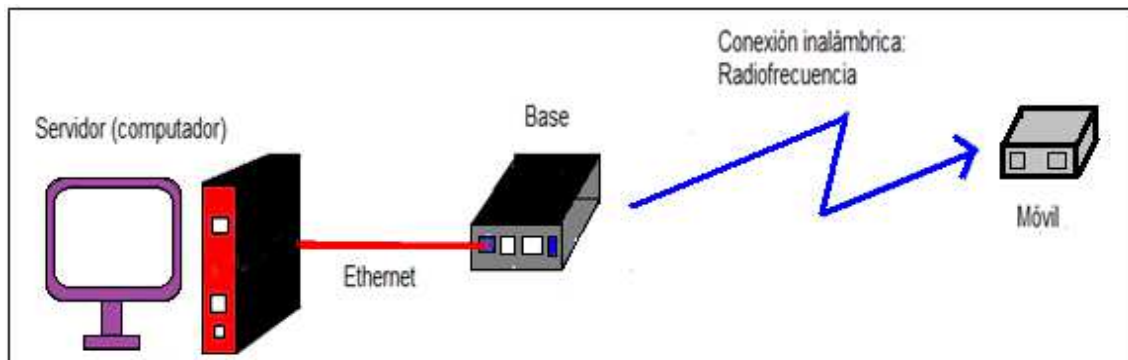


Figura N. 2.1: Diagrama del sistema
(Fuente: Investigador)

CAPÍTULO III PRESENTACIÓN DE LOS RESULTADOS

3.1. Introducción

El sistema de rastreo se lo realizó mediante dos dispositivos electrónicos a los que se les llamará Base y Móvil, por sus nombres se puede deducir que un dispositivo va a estar fijo y los otros dispositivos van a ser movibles.

El sistema de rastreo utiliza técnicas y métodos antes mencionados para poder ubicar los dispositivos móviles en un lugar específico como es una oficina o un pasillo de una empresa.

La ubicación de los dispositivos móviles se hace mediante la triangulación de tres puntos o coordenadas que se las establecerá en la aplicación respecto al plano de las oficinas y la ubicación del dispositivo base, estas coordenadas se archivan en una base de datos que cuando se requiera utilizarlas simplemente se compara los nuevos datos censados con los datos guardados y así se obtiene la triangulación de los puntos y la ubicación de los dispositivos.

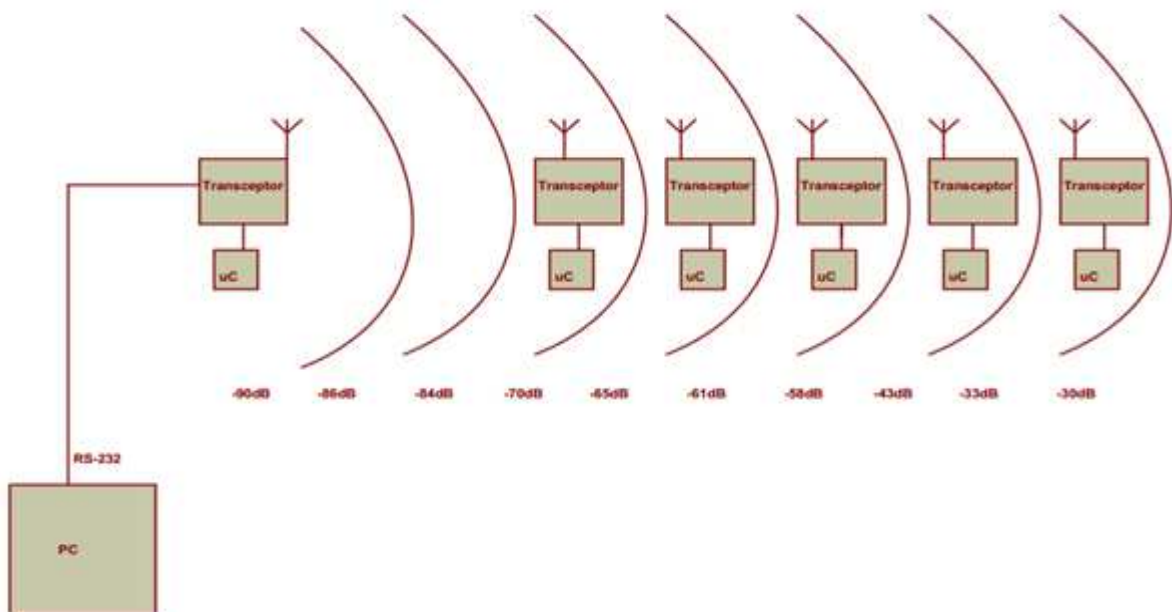


Figura N. 3.1: Diagrama de bloques del sistema de rastreo.
(Fuente: Investigador)

3.2. Dispositivo Base

La Base es un dispositivo fijo que estará conectado a un computador y se lo va a ubicar en la estación de trabajo del personal de seguridad de la empresa LR Telecomunicaciones, para que rastree a los dispositivos móviles mediante una aplicación.

El dispositivo Base se lo realizará por partes para evitar errores y mejorar el funcionamiento total.

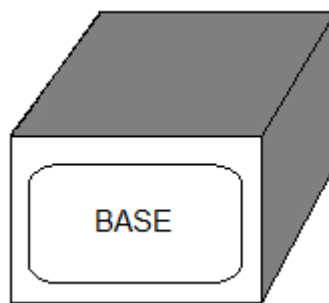


Figura N. 3.2: Gráfico del dispositivo Base
(Fuente: Investigador)

3.2.1. Funcionamiento del dispositivo Base

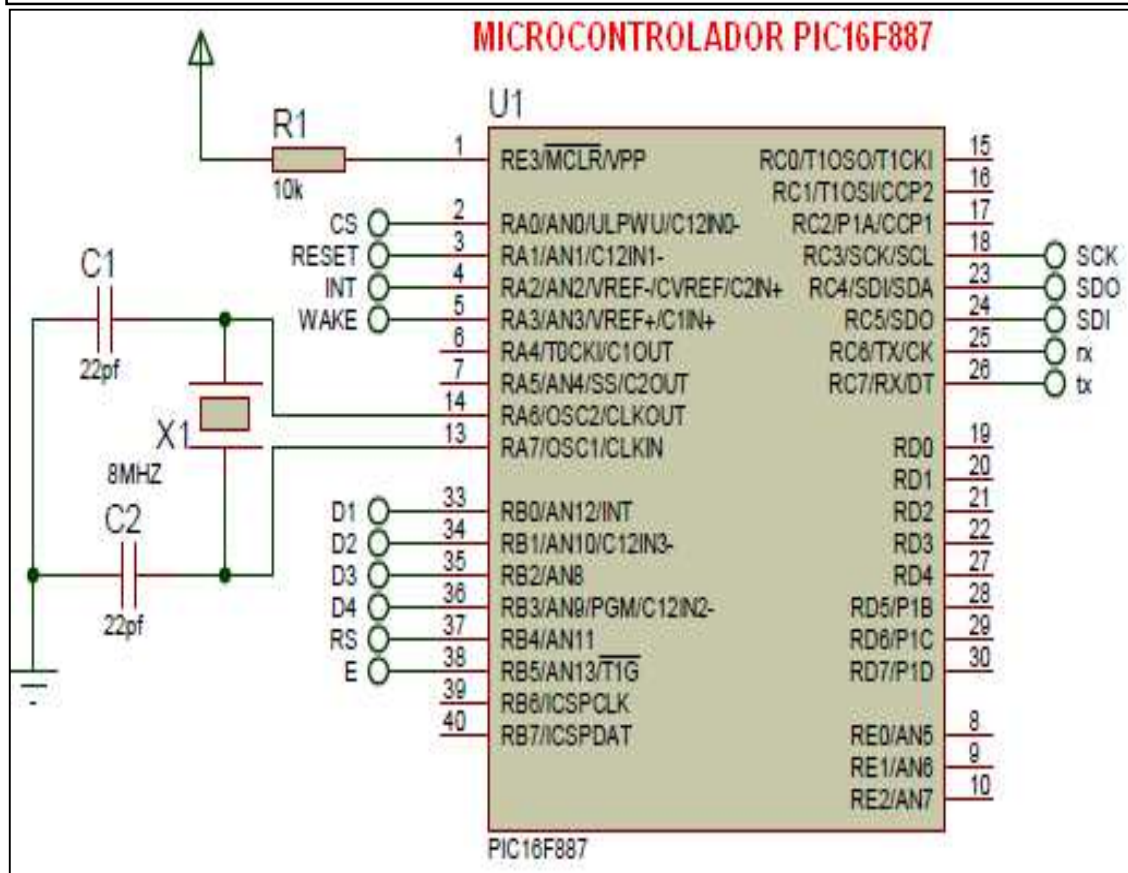
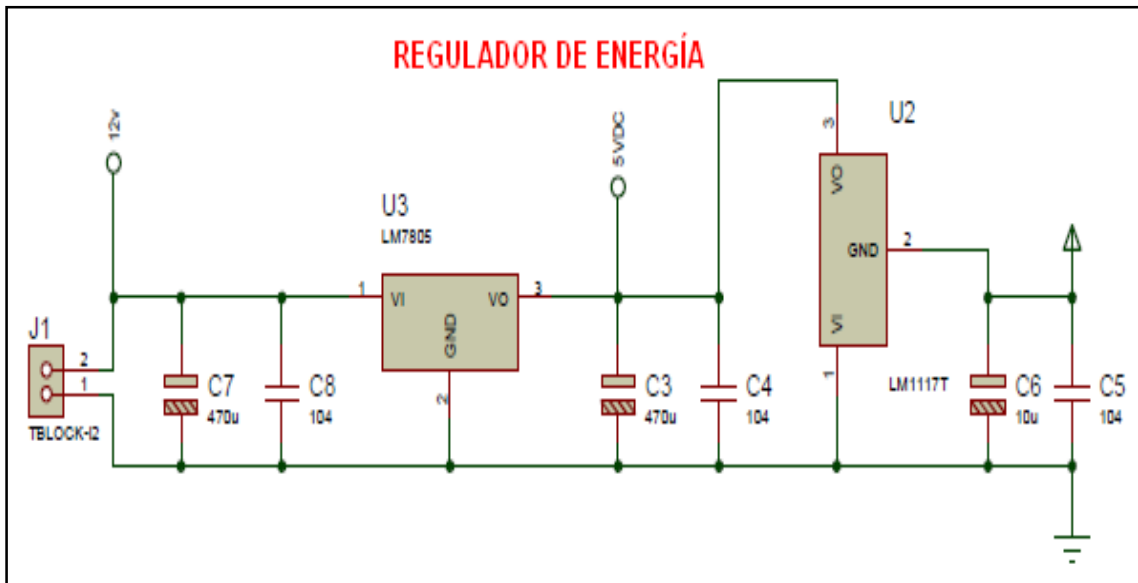
El funcionamiento del dispositivo Base se lo realiza mediante el envío de información continua de manera inalámbrica, en la que se especifica un nombre o ID para que los dispositivos móviles puedan encontrar a la Base.

Se debe esperar a que los dispositivos móviles manden una respuesta a la información enviada, la Base extrae los datos de la respuesta y los guarda en una base de datos en el computador al que está conectada la Base.

Con los datos almacenados, la Base ya puede rastrear a todos los dispositivos móviles que se encuentren distribuidos en las instalaciones de la empresa y así poder ubicarlos cuando sea necesario.

La Base configurará los parámetros necesarios, como son las direcciones IP, potencia de salida de la señal inalámbrica y nombres de los dispositivos fijos.

3.2.2. Elementos del dispositivo Base



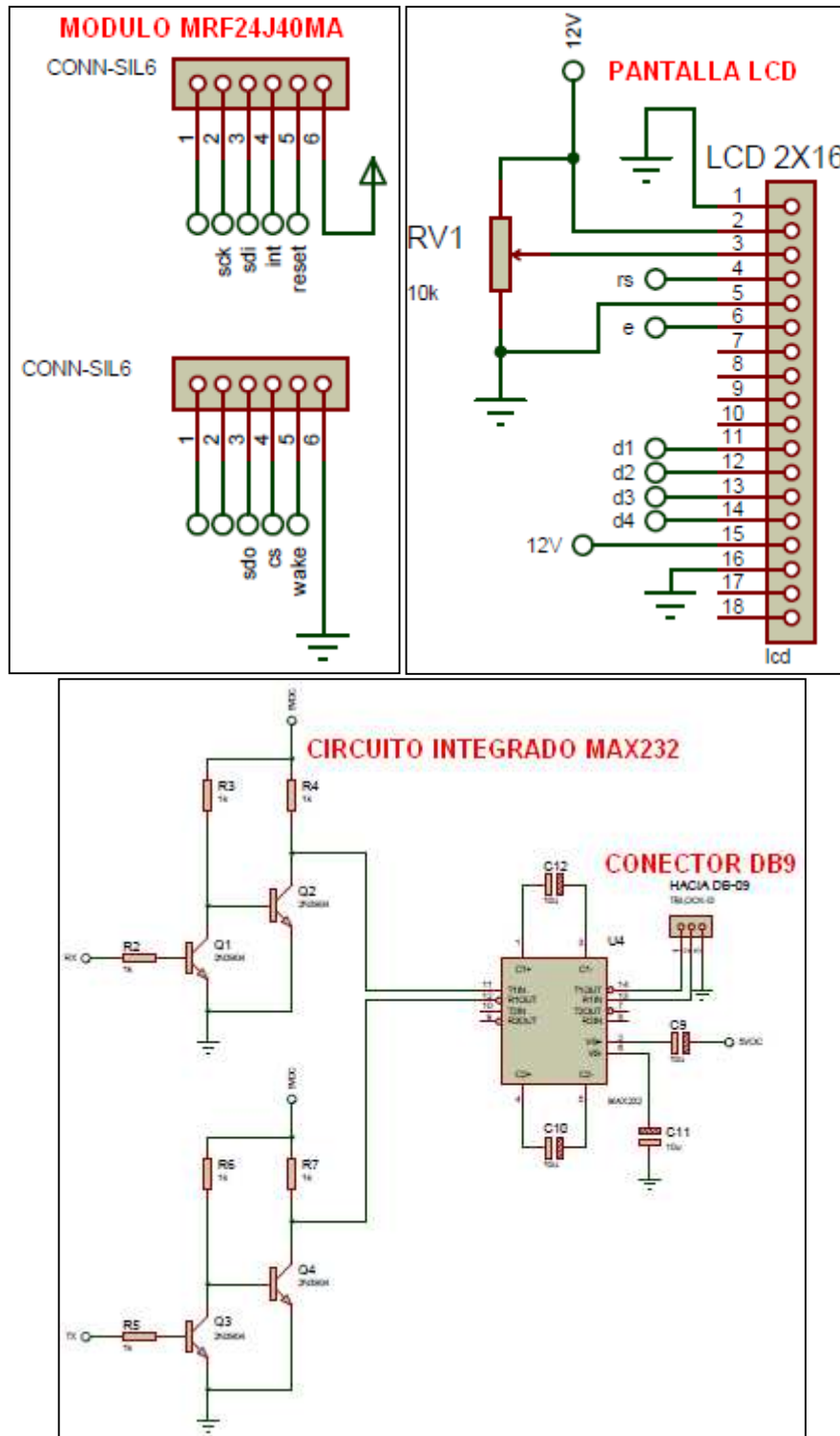


Figura N.3.3: Diagrama esquemático del dispositivo Base.

(Fuente: Investigador)

Las partes principales del dispositivo Base son las siguientes: el módulo transmisor-receptor inalámbrico MRF24J40MA, el microcontrolador PIC16F887, el regulador de energía y las partes secundarias son la adaptación para la pantalla LCD y la conexión para el computador mediante el circuito integrado MAX232 con el adaptador DB9.

3.2.2.1. Módulo de transmisión/recepción MRF24J40MA

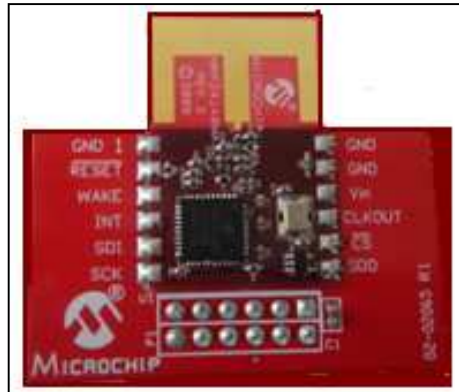


Figura N. 3.4: Módulo RadioTransceiver MRF24J40MA.
(Fuente: Datasheet módulo MRF24J40MA)

El módulo transmisor-receptor MRF24J40MA trabaja en la frecuencia de 2.4 GHz certificado por el estándar IEEE 802.15.4 mencionado en el Capítulo I, caracterizado por el bajo consumo en las comunicaciones inalámbricas.

Tiene incorporado una antena a la tarjeta de circuito impreso (PCB) y circuitos de acoplamiento que permiten soportar protocolos de comunicación como el ZigBee, MiWi y MiWi P2P (punto a punto).

El módulo se puede montar en una superficie y utilizar con controladores de 8, 16 o 32 bits mediante una interfaz del tipo SPI (interfaz periférico serial) de 4 hilos.

La pila de protocolo P2P soporta topologías de red inalámbrica en estrella y punto a punto que es lo que se requiere para el rastreo del sistema.

Ofrece diversos modos de funcionamiento como son: dormido, barrido activo y detección de energía y se caracteriza por un funcionamiento robusto y con un bajo consumo de energía por su tamaño.

Al módulo MRF24J40MA transmisor/receptor se lo puede programar como transmisor o receptor utilizando el programa Terminal.exe como se indica en la figura 3.5 y 3.6.



Figura N. 3.5: Programa para configurar el Módulo MRF24J40MA.
(Fuente: Investigador)

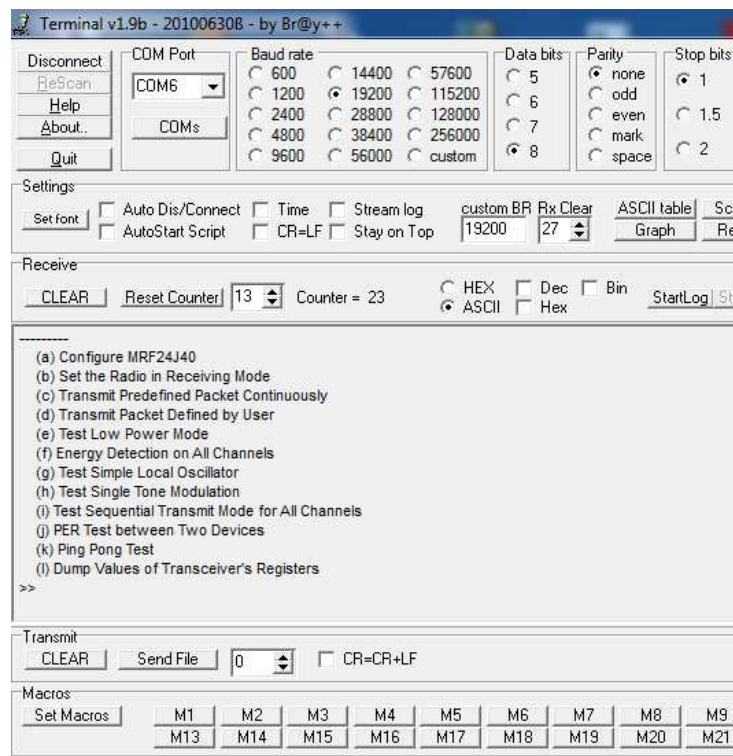


Figura N. 3.6: Pantalla principal del programa Terminal.
(Fuente: Programa Terminal.exe)

3.2.2.2. Microcontrolador PIC16F887

Para el proyecto se eligió el microcontrolador PIC887 de la familia 16F, porque este posee una memoria programable de 128 Kb y memoria de datos de 4 Kb, siendo suficiente para la programación de todas las instrucciones.

Este PIC tiene un nivel máximo de rendimiento con una velocidad de hasta 16 millones de instrucciones por segundo (MIPS) de potencia de procesamiento.

El PIC16F887 cuenta con periféricos avanzados, como la CAN (red de área de control), USB, Ethernet, LCD.

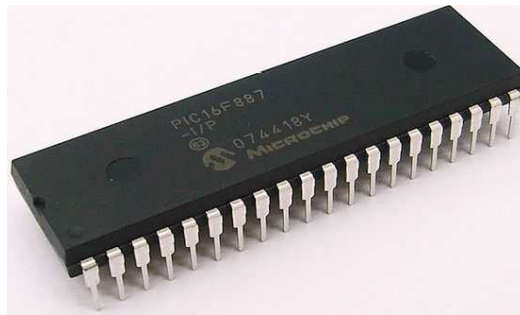


Figura N. 3.7: Microcontrolador PIC16F887.
(Fuente: Datasheet PIC16F887)

Características del PIC16F887:

Número de pines	16-100
Interrupciones	Capacidad de interrupción múltiple con contexto hardware ahorro
Rendimiento	Hasta 16 MIPS
Instrucciones	83, 16 bits
Memoria de programa	Hasta 128 KB
Memoria de datos	Hasta 4 KB
Pila Hardware	32 de nivel
Características	Además de la mejorada de medio rango: Multiplicador 8x8 Hardware CAN (red de área de control) USB Ethernet
Reflejos	Alto rendimiento, optimizado para la programación en C, periféricos avanzados
Número total de dispositivos	193
Familias	PIC16

Tabla N. 3.1: Características del PIC16F887

(Fuente: Datasheet PIC16F887)

Para conectar la Base al computador se ocupará el módulo MAX232 que es el que se encarga de la comunicación entre el microcontrolador con el puerto DB9. Esta conexión se la puede apreciar en la Figura N. 3.8 en los pines 25 y 26.

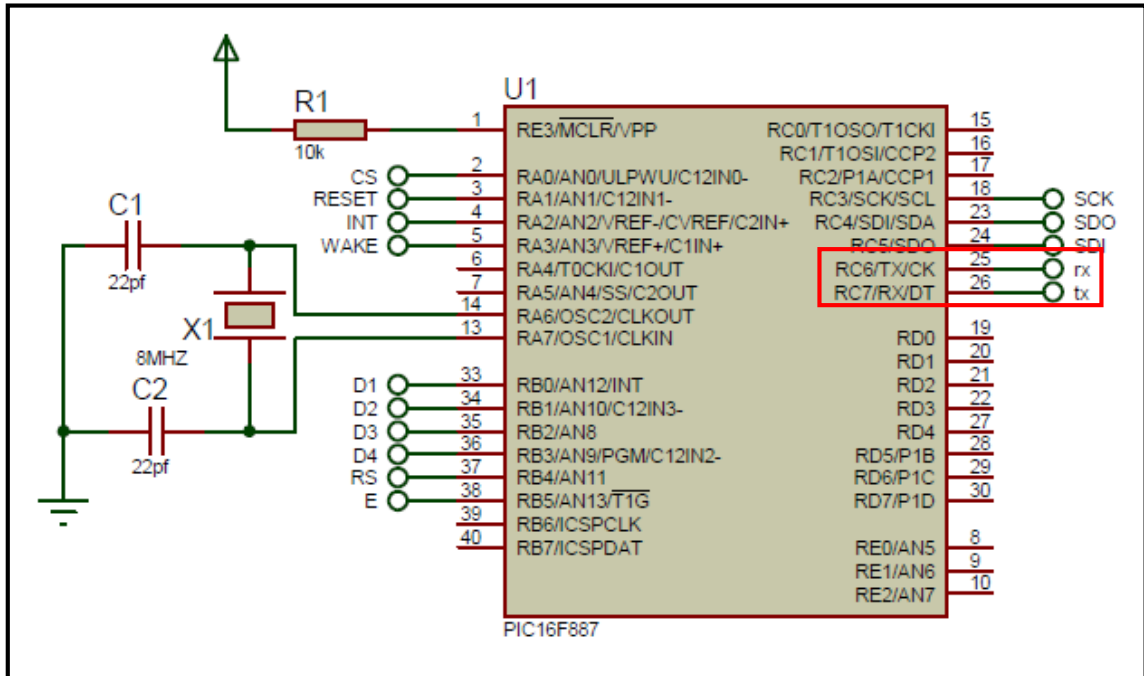


Figura N. 3.8: Diagrama de pines del PIC16F887
(Fuente: Investigador)

3.2.2.3. Regulador de energía

En esta parte se administrará la energía que consumirá el dispositivo mediante los circuitos integrados reguladores de tensión LM7805 y LM1117T.

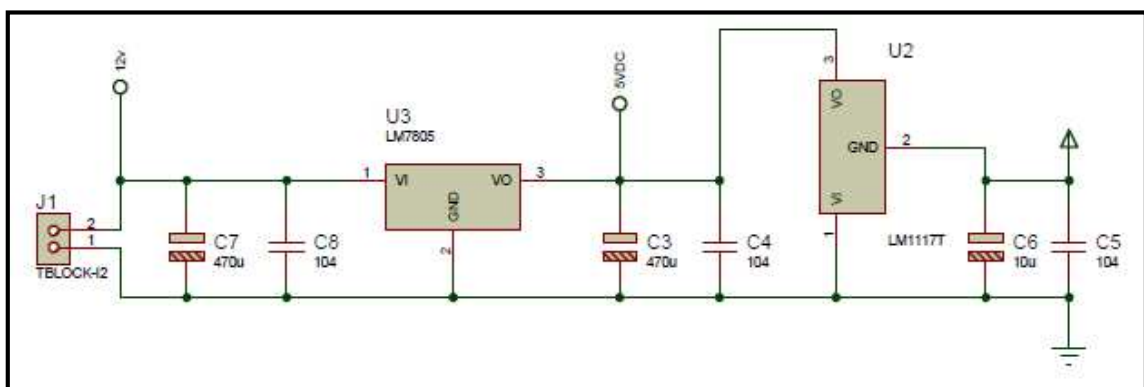


Figura N. 3.9: Diagrama de la parte de suministro de energía
(Fuente: Investigador)

Se realizó un análisis del consumo de energía mediante la siguiente tabla:

Componente electrónico	Consumo energía
Módulo MRF24J40MA	23 mA
PIC16F887	25 mA
LM7805	1 mA
LM1117T	1 mA
TOTAL	50 mA

Tabla N. 3.2: Consumo aproximado de energía

(Fuente: Investigador)

La Base va a estar funcionando todos los días, ocho horas diarias, por lo tanto necesita que la energía sea administrada de manera controlada para que no haya problemas de mal funcionamiento.

3.2.2.4. Circuito Integrado MAX232

Esta parte es la que permite que el dispositivo Base envíe y reciba la información de los dispositivos móviles con el computador, esto se lo realiza mediante el circuito integrado MAX232 que se conecta al microcontrolador PIC con el puerto DB9 del computador. Es una de la partes más pequeñas pero la más importante para la comunicación.

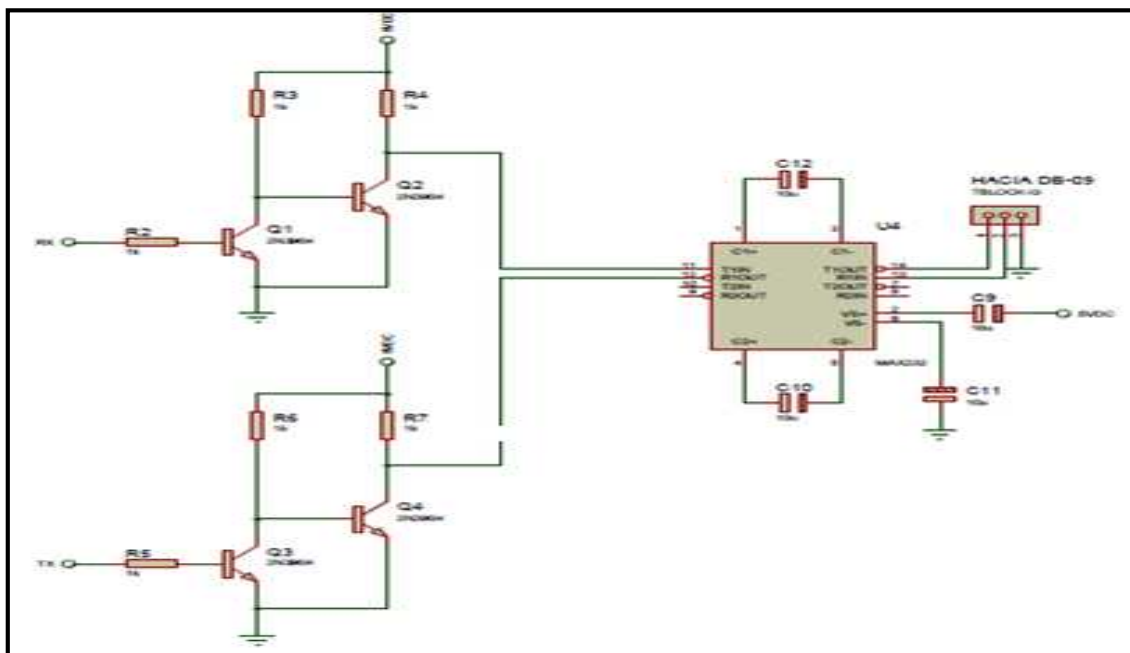


Figura N. 3.10: Diagrama de pines del MAX232.

(Fuente: Investigador)

Se colocó una pantalla LCD para realizar las pruebas de la comunicación entre los dispositivos móviles y la base, con esto se comprobó el lugar donde se encontraba cada uno de los dispositivos dentro de la empresa.

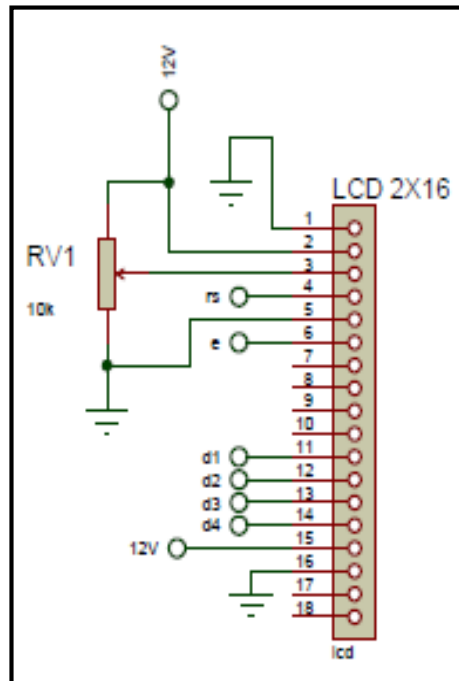


Figura N. 3.11: Diagrama de pines de la pantalla LCD.
(Fuente: Investigador)

La implementación del dispositivo base se lo muestra en las fotografías, las cuales se encuentran en el Anexo N. 1.

El código de programación del dispositivo Base se encuentra en el Anexo N.2.

3.3. Dispositivo Móvil

Los dispositivos móviles van a ser cajas pequeñas que se entregará a cada persona que ingrese a la Empresa LR Telecomunicaciones, para rastrear su ubicación dentro de las oficinas y saber el lugar en el que se encuentran.

El rastreo también se lo podrá realizar con los computadores de la empresa.



Figura N. 3.12: Gráfico del dispositivo móvil.
(Fuente: Investigador)

3.3.1. Funcionamiento del dispositivo móvil

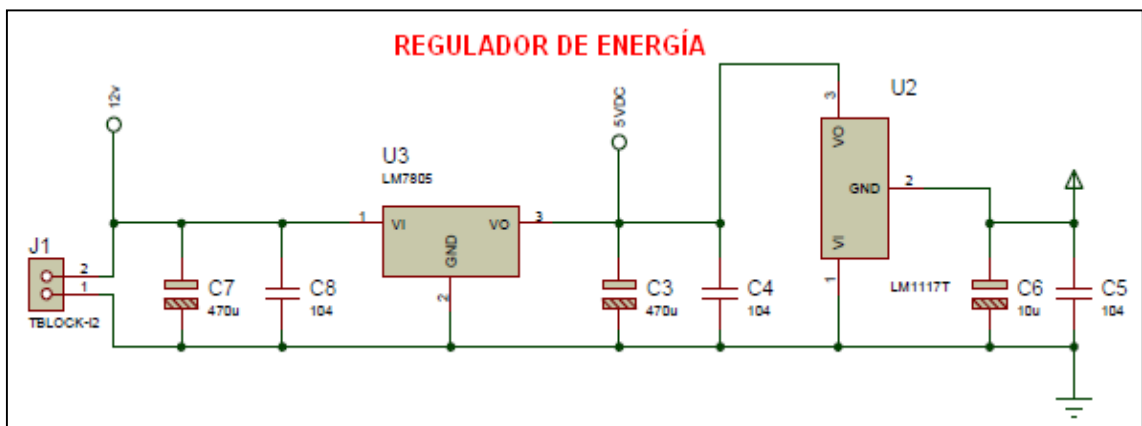
El funcionamiento del dispositivo móvil se lo realiza mediante el envío de información continua de manera inalámbrica, para así entregar a la Base sus datos de ubicación como por ejemplo las coordenadas en las que se encuentra.

Esos datos que se envía a la Base sirven para ubicar el lugar exacto de cada uno de los dispositivos móviles en las oficinas de la empresa.

Los dispositivos móviles ocupan un espacio más reducido porque no poseen la parte Ethernet como la Base, pero se le mejoró la parte de la energía porque debe funcionar más de ocho horas al día, los cinco días de la semana.

Una vez realizada la placa del dispositivo móvil se puede observar que es más grande de lo que se esperaba por motivo de los elementos principales como es el módulo MRF24J40MA y el microcontrolador PIC, como se puede ver en las fotografías del Anexos N. 1.

3.3.2 Elementos del dispositivo móvil



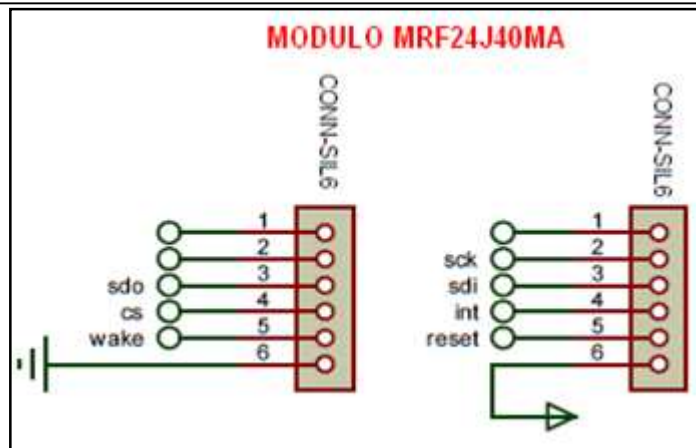
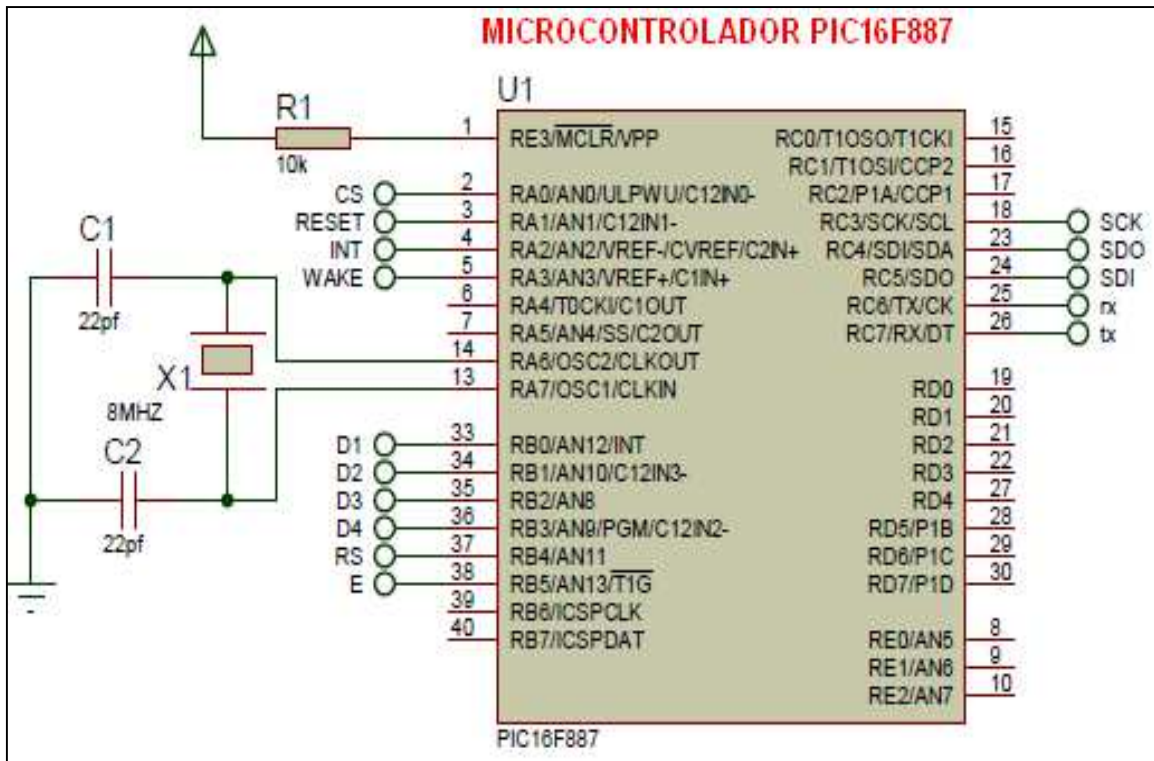


Figura N. 3.13: Diagrama esquemático del dispositivo móvil
(Fuente: Investigador)

Los elementos del dispositivo móvil son los mismos que el dispositivo fijo (Base) a excepción de la parte del circuito integrado MAX232, ya que los móviles envían la información inalámbricamente y no necesitan conectarse con el computador.

Por lo tanto sólo se nombrará los elementos del dispositivo móvil porque su explicación ya se encuentra detallada en el Capítulo 3.1.2.

Los elementos del dispositivo móvil son:

- Módulo de transmisión/recepción MRF24J40MA.- para el dispositivo móvil el módulo se lo configura como transmisor mediante el programa Terminal.exe.
- Microcontrolador PIC16F887
- Regulador de energía

El código de programación del dispositivo móvil se encuentra en el Anexo N.3.

3.4. Aplicación

Esta parte está conformada por el computador que contiene la aplicación que se realizó en Visual Basic 6.0, en la cual se podrá observar los dispositivos electrónicos móviles y así poder saber con exactitud donde se encuentra una persona o un objeto como es una laptop.

La siguiente imagen es de la elaboración de la interfaz gráfica en la plataforma de Visual Basic.

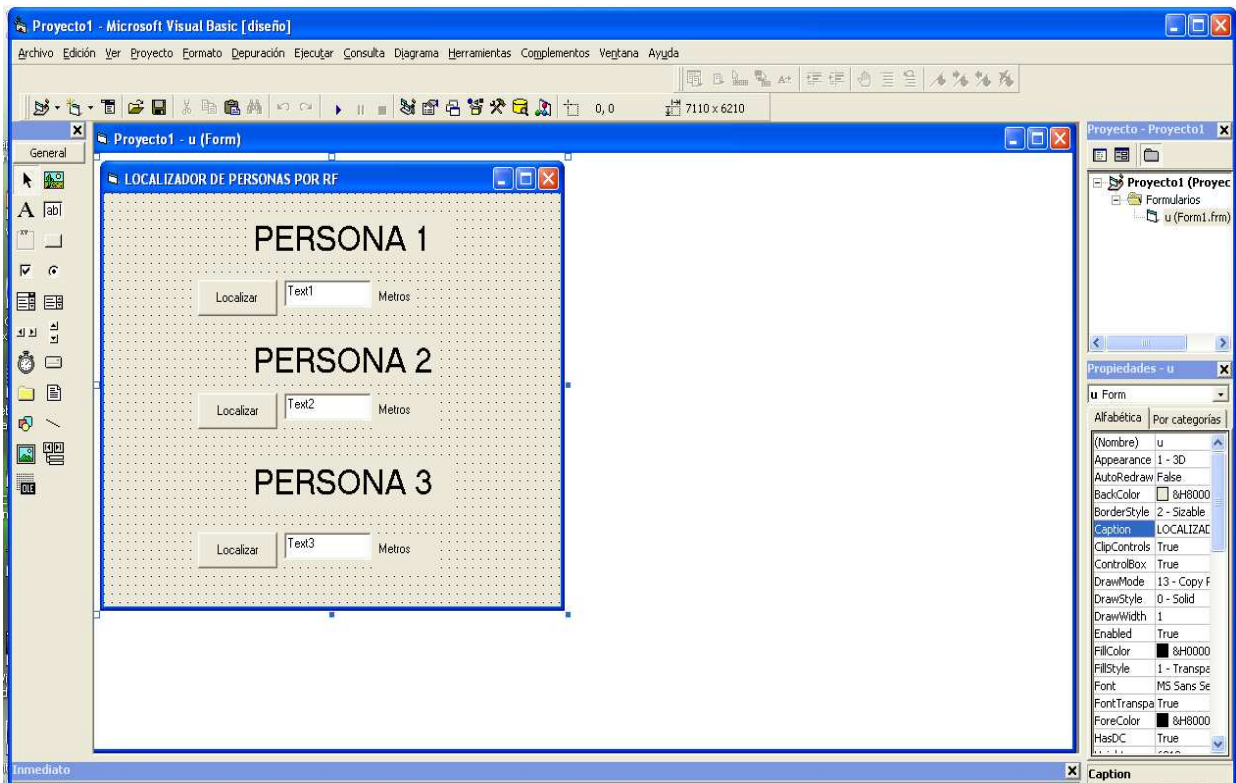


Figura N. 3.14: Interfaz gráfica de la aplicación en Visual Basic.

(Fuente: Visual Basic 6.0)

Para poder saber en qué lugar de la empresa LR Telecomunicaciones se encuentra cada dispositivo móvil se realizó un mapa de las instalaciones, las cuales se las delimitó por zonas, que van desde el garaje hasta la última oficina. En qué zona se encuentra cada dispositivo se podrá observar a un lado del mapa.

Una vez concluida la elaboración de la aplicación se muestra la presentación final de la aplicación que va a ir instalado en el computador, se abrirá una pantalla al presionar doble clic sobre el ícono del programa ejecutable, como se muestra en la Figura N. 3.15.

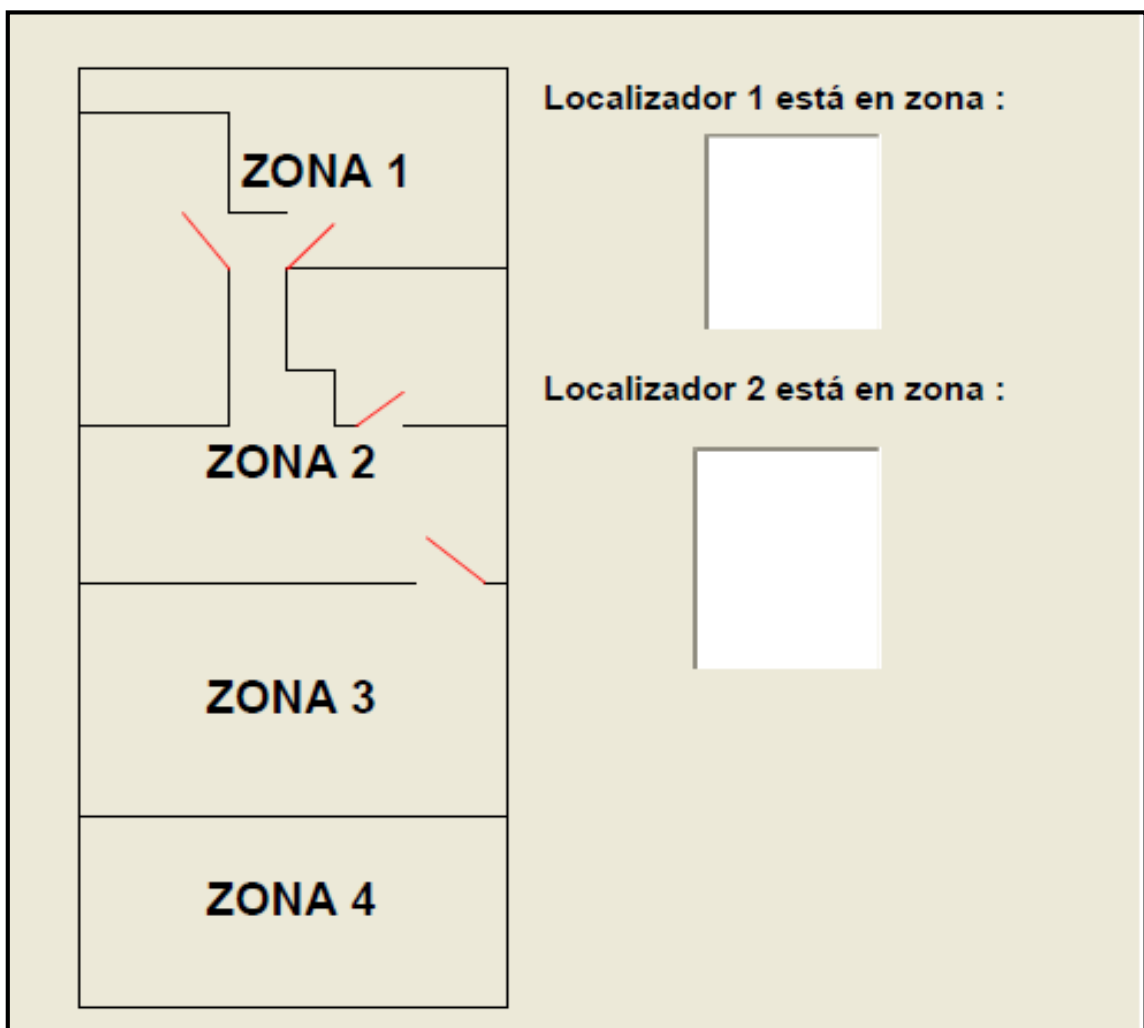


Figura N. 3.15: Interfaz gráfica de la aplicación al ejecutarla.

(Fuente: Investigador)

3.5. Validación de la implementación

Las pruebas se las realizó con un dispositivo base y dos dispositivos móviles, los cuales se les llevó entre las instalaciones de la Empresa LR Telecomunicaciones y así poder ir viendo en la aplicación a qué zona corresponden cada oficina.

También se pudo observar mediante la pantalla de LCD que viene en el dispositivo base la zona en que se encuentra cada uno de los dispositivos móviles.

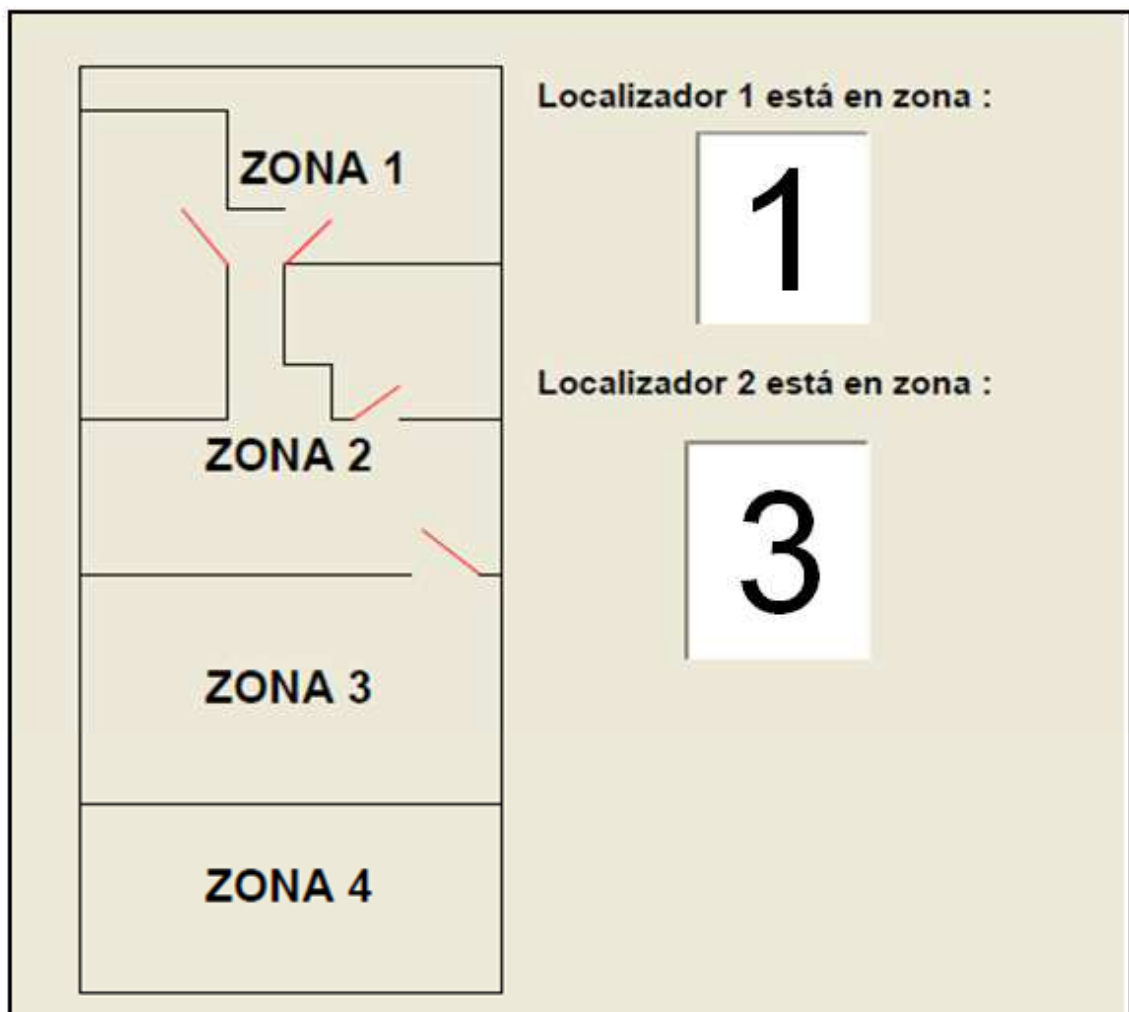


Figura N. 3.16: Interfaz gráfica de la aplicación mostrando la zona de cada dispositivo móvil.

(Fuente: Investigador)

El código de programación de la aplicación se encuentra en el Anexo N. 4.

3.6. Estudios de factibilidad

3.6.1. Factibilidad Técnica

Respecto a la adquisición de los recursos que a continuación se detallan, la mayoría se encuentra en el mercado nacional, pero los módulos de transmisión/recepción se los consiguieron en la empresa Microchip de Estados Unidos y se trajo mediante correo.

Recursos tecnológicos necesarios para la implementación del proyecto:

1. Un servidor con las siguientes características:
 - Procesador: Intel Pentium 5
 - Memoria RAM: 1GB
 - Espacio en el disco duro: 5GB
 - Conectividad: un puerto DB9
2. Módulo de transmisión/recepción MRF24J40MA
3. Microcontrolador PIC16F887
4. Circuito Integrado MAX232
5. Batería de Litio
6. Varios componentes como condensadores, resistencias, transistores, placas, diodos.
7. Sistema Operativo Windows XP.

Técnicamente el sistema es factible, gracias a que los componentes se pueden conseguir de una manera fácil y confiable, para realizar la implementación.

3.6.2. Factibilidad Operativa

El proyecto es importante porque será un nuevo sistema de seguridad y de localización, el cual podrá monitorear los movimientos exactos de las personas dentro de las instalaciones, informará al personal de seguridad en tiempo real y evitará las pérdidas de: materiales, objetos de valor e información.

El personal de la Empresa LR Telecomunicaciones, que usará los dispositivos se adaptará muy fácilmente a su utilización ya que son sencillos, ligeros y recargables, además de la seguridad que ofrece el sistema.

3.6.3. Factibilidad Económica.

El costo aproximado para la implementación del sistema se detalla a continuación en las Tablas N. 3.3, 3.4 y 3.5.

COMPONENTE	PRECIO
MRF24J40MA	\$ 10
PIC16F887	\$ 7
CRISTAL 8 MHz	\$ 1
CI LM7805	\$ 1
CI LM1117	\$ 1
CI MAX232	\$ 1
Batería de Litio	\$ 14
Otros componentes	\$ 10
TOTAL	\$ 45

Tabla N. 3.3: Precio del dispositivo electrónico móvil.

(Fuente: Investigador)

COMPONENTE	PRECIO
MRF24J40MA	\$ 10
PIC16F887	\$ 7
CRISTAL 8 MHz	\$ 1
CI LM7805	\$ 1
CI LM1117	\$ 1
Cargador 110V	\$ 25
Otros componentes	\$ 10
TOTAL	\$ 55

Tabla N. 3.4: Precio del dispositivo electrónico fijo.

(Fuente: Investigador)

COMPONENTE	PRECIO
Dispositivo móvil	\$ 45
Dispositivo fijo	\$ 55
Servidor (computador del guardia de seguridad)	\$ 220

Tabla N. 3.5: Resumen de precios de los componentes del sistema.

(Fuente: Investigador)

Se puede concluir que es un sistema económico y por consiguiente es factible su implementación, porque si bien son costosos los dispositivos electrónicos, el resto de la instalación es bastante económica comparado con otros sistemas de seguridad.

Por ejemplo, en el área de la Empresa LR Telecomunicaciones es de $50m^2$ por lo tanto se requerirá de una base para un perfecto funcionamiento y poseen cinco empleados entonces serán cinco dispositivos móviles, tendría un costo de \$ 500.

3.7 Evaluación de la implementación por parte de la Empresa LR Telecomunicaciones

Se realizó una encuesta al personal que labora en la empresa LR Telecomunicaciones respecto al nuevo sistema de posicionamiento para la seguridad que se implementó.

Se aplicó la fórmula de la muestra que es la siguiente:

$$\frac{k^2 Npq}{e^2(N - 1) + k^2 pq}$$

Ecuación N.3.1: Fórmula de la muestra para el resultado de encuestas.

Datos:

N=número de encuestados

K=constante de confianza en este caso es 1.96

e=error muestral que es 0.05

p=proporción de individuos que poseen en la población la característica de estudio que es 0.5

q=proporción de individuos que no poseen esa característica que es 0.5

Una vez realizada la encuesta del personal de la empresa que en este caso son cinco personas se procede con el resultado.

Resultados de la encuesta:

1.- ¿Estaría dispuesto a probar un nuevo sistema de seguridad?

Si	60%
No	40%
Total	100%

Fuente: Investigador

Conclusión: un porcentaje de 60% afirma estar dispuesto a probar un nuevo sistema de seguridad, lo que es factible para la implementación que se realizó en la empresa, por vendría a ser más de la mitad del personal.

2.- ¿Le importaría usar un dispositivo electrónico como una tarjeta de seguridad?

Si	80%
No	20%
Total	100%

Fuente: Investigador

Conclusión: Según los resultados la mayoría de personas afirma estar dispuesto a usar una tarjeta de seguridad en las horas laborables.

3.- El nuevo sistema de seguridad rastrea sus movimientos en su lugar de trabajo durante las horas laborables. ¿Lo gustaría que le rastreen sus movimientos?

Si	80%
No	20%
Total	100%

Fuente: Investigador

Conclusión: El 80% no le importaría que le rastreen sus movimientos mientras trabaja dentro de la empresa.

4.- ¿Cree usted que el nuevo sistema de posicionamiento, implementado en la empresa ha mejorado la seguridad del personal y de los objetos de valor?

Si	100%
No	00%
Total	100%

Fuente: Investigador

Conclusión: Según los resultados la mayoría de personas afirma estar dispuesto a usar una tarjeta de seguridad en las horas laborables.

5.- Qué problemas ha mejorado el nuevo sistema de posicionamiento dentro de la empresa?

Conclusión: Todo el personal de la empresa concuerda en que ha mejorado la seguridad dentro de la empresa con el nuevo sistema de posicionamiento implementado.

Conclusión General

La encuesta permitió determinar la factibilidad del hecho de implementar un nuevo sistema de posicionamiento en la Empresa LR Telecomunicaciones, debido principalmente a la conclusión que se obtuvo que al personal no le importe que se le monitoree sus movimientos durante las horas laborables y que están dispuestos a probar los dispositivos electrónicos que se ha proporcionado.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Los valores en el registro RSSI del módulo transmisor/receptor MRF24J40MA, albergan un valor del tamaño de un byte, este valor varía dependiendo de la potencia recibida en el dispositivo Base configurado como receptor. Dicho valor no es constante, tiene muchas variaciones, por lo tanto se recurrió a segmentarlo entre ciertos rangos para poder estimar un valor aproximado de la distancia a la que se encuentra el dispositivo móvil (transmisor).
- El microcontrolador PIC16887 se lo trabajó con el programa MikroC, ya que las líneas de programación sobrepasan los 2Kb de memoria de funcionamiento y el módulo MRF24J40MA sólo acepta programas que pertenezcan a la línea de la Empresa Microchip.
- La parte para la energía se tuvo varios inconvenientes como por ejemplo el voltaje para los elementos; por tal motivo se realizaron dos esquemas diferentes: el primero es para el dispositivo fijo con una fuente de alimentación de 12 VDC y se lo reduce a 3.3 VDC por el módulo y el segundo es para el dispositivo móvil que la fuente de alimentación es de 5 VDC y se reduce a 3.3 VDC.
- Cuando ya se obtuvo la escala y los rangos del valor de RSSI del módulo transmisor/receptor MRF24J40MA, se logró enviar hacia el computador una señal serial con un valor estimado en metros de la distancia a la que se encuentra el dispositivo móvil (transmisor) y mediante estos valores ya se puede visualizar en la pantalla LCD.
- La aplicación se la realizó con el mapa de la Empresa LR Telecomunicaciones, mediante líneas básicas y dos cuadros donde se visualiza en la zona está el dispositivo móvil.

Recomendaciones

- El módulo de transmisión/recepción MRF24J40MA es de un tamaño reducido pero necesita un adaptador porque los pines de conexión vienen en sentido horizontal y para la placa deben estar en sentido vertical.
- El microcontrolador PIC16F887 es de gran tamaño y no se lo puede reemplazar por la cantidad de líneas de programación que necesita, por este motivo la placa resultó de un tamaño más grande de lo esperado.
- Para la alimentación de los dispositivos electrónicos móviles deben ser baterías de litio de preferencia recargables por las horas de funcionamiento al día.
- Debido a que los módulos MRF24J40MA trabajan a 3 VDC, el microcontrolador se tienen que alimentar con el mismo potencial, pero si requerimos realizar pruebas de funcionamiento debemos conectar una pantalla LCD 2x16 (provisional) que funciona a 5 VDC, por lo tanto se tiene que construir una placa adaptadora que transforme la salida del microcontrolador 3 VDC a 5 VDC, de esa manera se puede visualizar y tener datos en el LCD.
- En la comunicación hacia el computador, al trabajar el microcontrolador a 3 VDC no se puede conectarlo directo al circuito integrado MAX232, por lo tanto se debe elevar el voltaje del PIC para conectar al MAX232 de 3 a 5 VDC, para ello se utiliza un arreglo de transistores tanto para recibir como para transmitir la información.
- La implementación se la realizó de manera artesanal, es decir que se realizó todo el proceso de la placas y el diseño de las pistas por lo que el dispositivo electrónico móvil resultó más grande de lo que se esperaba, entonces se puede recomendar que el módulo de transmisión/recepción MRF24J40MA sea implantado dentro de la placa y el microcontrolador PIC puede ser de menor tamaño para que la placa tenga una altura no mayor de 2cm y así sea más fácil que sea portable por el usuario.

BIBLIOGRAFÍA

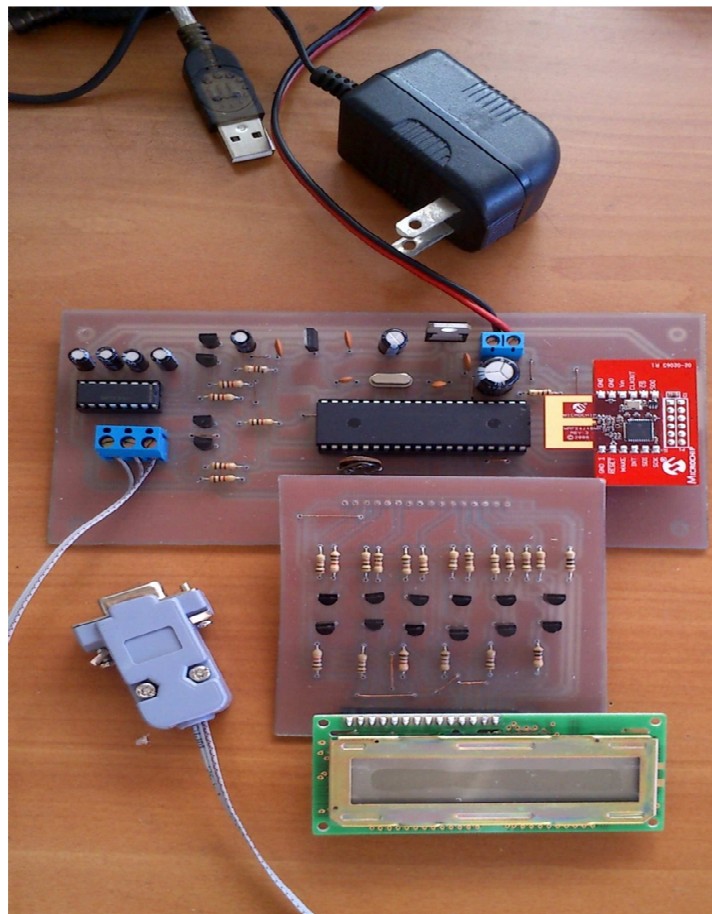
- Bernardos, A. (2003). *Tecnologías de Localización*. España: Universidad Carlos III.
- Moretto, V. (Febrero de 2008). *Tecno Imprese*. Recuperado el 10 de 05 de 2014, Recuperado de http://www.tecnoimprese.it/user/File/Eventi/RF08-14feb_melchioni.pdf
- Paramir, B. & Venkata, N. P. (2000). *Radar: an in-Building RF-based User Location and Tracking System*. IEEE INFOCOM.
- Rabbany, A. e. (2002). *Introduction To GPS*.
- Río, I. A. (2008). *Diseño de un sistema de posicionamiento que permita determinar las coordenadas exactas y/o objetos en un área cerrada preestablecida con relación a un punto de referencia, con un error menor a cincuenta centímetros*. Quito, Pichincha, Ecuador.
- Tomasi, W. (2003). *Sistemas de Comunicaciones Electrónicas*. México: PEARSON EDUCATION.
- Zoe, F. L. (2006). *Sensores de identificación por radio-frecuencia (RFID)*. Universidad Jaime I.
- Cardama, A. & Jofre, L. (2004). *Antenas*. España: Edicions UPC S.L.
- Lee, Y. & Stuntebeck, E. & Miller, O.C. (2006). *MERIT: Mesh of RF sensors for Indoor Tracking*. Recuperado de https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CBwQFjAA&url=http%3A%2F%2Fwww.researchgate.net%2Fprofile%2FERich_Stuntebeck%2Fpublication%2F224760895_MERIT_Mesh_of_RF_sensors_for_Indoor_Tracking%2Flinks%2F0deec5372614f19e7600000.pdf&ei=lt8BVcidEYapNrLVgpgN&usq=AFQjCNEapyPUf3bUM_tnTM5pwwzH04KcRA&bvm=bv.87920726,d.eXY
- González, F. & García-Reinoso, J. (2002). *Bluetooth Location Networks*. Globecom IEEE. Recuperado de <http://163.13.128.75/member/seminar/files/Bluetooth%20Location%20Networks.pdf>
- Lionel, M. & Yunhao, L. & Yiu Cho, L. & Abhishek, P. (2004). *LANDMARC: Indoor Location Sensing Using Active RFID*. Recuperado de <http://www.cs.ust.hk/~liu/Landmarc.pdf>
- Reyes, C. A. (2008). *Microcontroladores PIC – Programación en BASIC*. Ecuador: RISPERGRAF.

ANEXOS

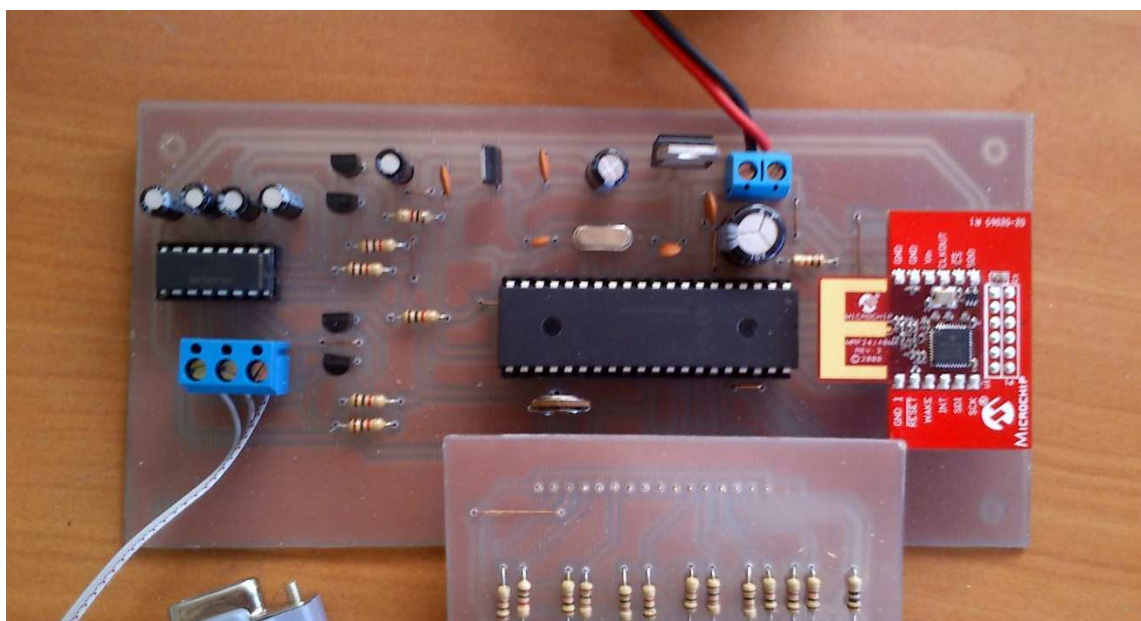
Anexo N.1

Fotografías de los dispositivos electrónicos fijo
y móvil

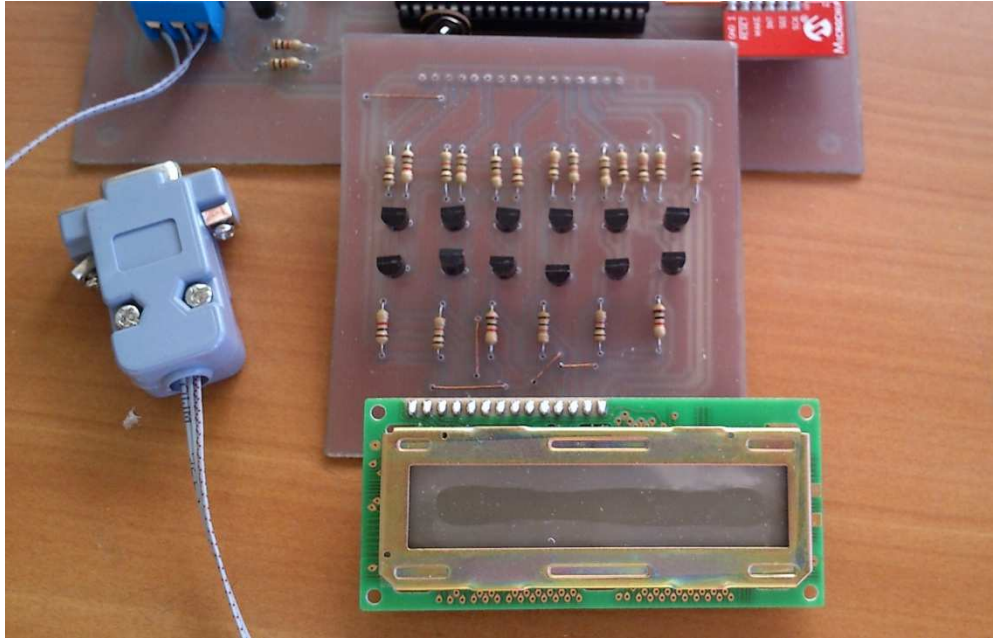
Fotografías del dispositivo electrónico fijo o Base:



Fuente: Investigador

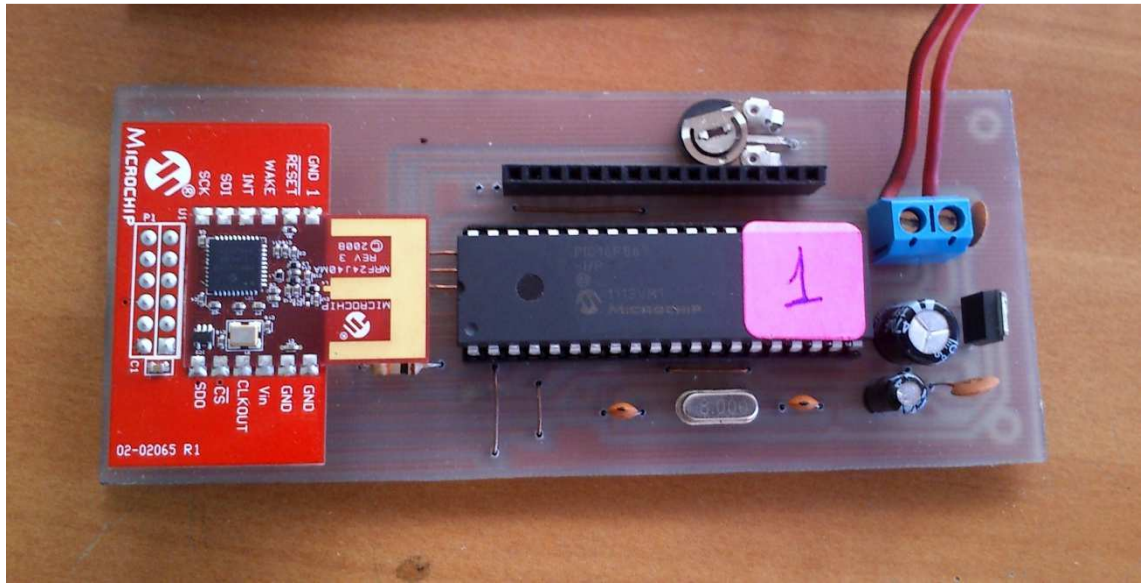


Fuente: Investigador

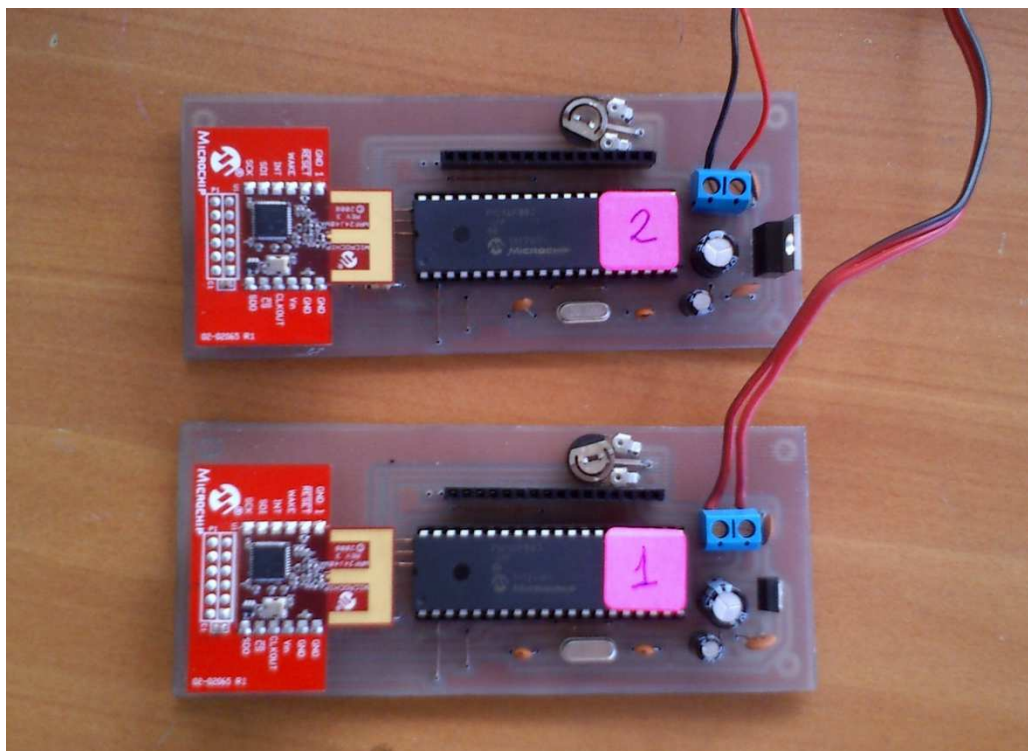


Fuente: Investigador

Fotografías de los dispositivos electrónicos móviles:



Fuente: Investigador



Fuente: Investigador

Anexo N. 2

Código de programación del dispositivo Base

Código de programación del dispositivo Base

```
1:
2:
3: #include "registers.h"
4:
5: // LCD module connections
6: sbit LCD_RS at RB4_bit;
7: sbit LCD_EN at RB5_bit;
8: sbit LCD_D4 at RB0_bit;
9: sbit LCD_D5 at RB1_bit;
10: sbit LCD_D6 at RB2_bit;
11: sbit LCD_D7 at RB3_bit;
12:
13: sbit LCD_RS_Direction at TRISB4_bit;
14: sbit LCD_EN_Direction at TRISB5_bit;
15: sbit LCD_D4_Direction at TRISB0_bit;
16: sbit LCD_D5_Direction at TRISB1_bit;
17: sbit LCD_D6_Direction at TRISB2_bit;
18: sbit LCD_D7_Direction at TRISB3_bit;
19: // End LCD module connections
20:
21: // EasyBee3 connections
22: sbit CS at RA0_bit; //CS pin
23: sbit RST at RA1_bit; //RST pin
24: sbit INT at RA2_bit; //INT pin
25: sbit WAKE at RA3_bit; //WAKE pin
26:
27: sbit CS_Direction at TRISA0_bit; //CS pin direction
28: sbit RST_Direction at TRISA1_bit; //RST pin direction
29: sbit INT_Direction at TRISA2_bit; //INT pin direction
30: sbit WAKE_Direction at TRISA3_bit; //WAKE pin direction
31:
32:
33: char txt[4];
34: char txt2[4];
35:
36: const unsigned short int DATA LENGHT = 3;
37: const unsigned short int HEADER LENGHT = 11;
38:
39: int address_RX_FIFO = 0x300;
40: int address_TX_normal_FIFO = 0x000;
41: short int data_RX_FIFO[1 + HEADER LENGHT + DATA LENGHT + 2 + 1 +
1], lost_data = 0;
42:
43: short int ADDRESS_short_1[] = {1,1}; // Source address
44: short int ADDRESS_long_1[] = {1,1,1,1,1,1,1,1};
45: short int ADDRESS_short_2[] = {2,2}; // Destination address
46: short int ADDRESS_long_2[] = {2,2,2,2,2,2,2,2};
47: short int PAN_ID_1[] = {3,3}; // Source PAN ID
48: short int PAN_ID_2[] = {3,3}; // Destination PAN ID
49:
50: short int DATA_RX[DATA LENGHT], DATA_TX[DATA LENGHT],
data_TX_normal_FIFO[DATA LENGHT + HEADER LENGHT + 2];
51: short int LQI = 0, RSSI2 = 0, SEQ_NUMBER = 0x23;
52:
53: /*
54: * Functions for reading and writing registers in short address
memory space
55: */
56:
57: // write data in short address register
58: void write_ZIGBEE_short(short int address, short int data_r){
59: CS = 0; // Select module
60: address = ((address << 1) & 0b01111111) | 0x01; // Calculating
addressing mode
61: SPI1_Write(address); // Addressing register
62: SPI1_Write(data_r); // Write data in register
63: CS = 1; // Deselect module
```

```

64: }
65:
66: // Read data from short address register
67: short int read_ZIGBEE_short(short int address){
68: short int data_r = 0, dummy_data_r = 0;
69:
70: CS = 0;
71: address = (address << 1) & 0b01111110; // Calculating addressing
mode
72: SPI1_Write(address); // Addressing register
73: data_r = SPI1_Read(dummy_data_r); // Read data from register
74: CS = 1;
75: return data_r;
76: }
77:
78: // Write data in long address register
79: void write_ZIGBEE_long(int address, short int data_r){
80: short int address_high = 0, address_low = 0;
81:
82: CS = 0;
83: address_high = (((short int)(address >> 3)) & 0b01111111) | 0x80;
// calculating addressing mode
84: address_low = (((short int)(address << 5)) & 0b11100000) | 0x10;
// calculating addressing mode
85: SPI1_Write(address_high); // addressing register
86: SPI1_Write(address_low); // addressing register
87: SPI1_Write(data_r); // write data in registerr
88: CS = 1;
89: }
90:
91: // Read data from long address register
92: short int read_ZIGBEE_long(int address){
93: short int data_r = 0, dummy_data_r = 0;
94: short int address_high = 0, address_low = 0;
95:
96: CS = 0;
97: address_high = (((short int)(address >> 3) & 0b01111111) | 0x80;
//calculating addressing mode
98: address_low = (((short int)(address << 5) & 0b11100000);
//calculating addressing mode
99: SPI1_Write(address_high); // addressing register
100: SPI1_Write(address_low); // addressing register
101: data_r = SPI1_Read(dummy_data_r); // read data from register
102: CS = 1;
103: return data_r;
104: }
105:
106: /*
107: * Reset
108: */
109: void pin_reset(void) // Reset from pin
110: {
111: RST = 0; // activate reset
112: Delay_ms(5);
113: RST = 1; // deactivate reset
114: Delay_ms(5);
115: }
116:
117: void PWR_reset(void){
118: write_ZIGBEE_short(SOFTRST, 0x04); //0x04 mask for RSTPWR bit
119: }
120:
121: void BB_reset(void){
122: write_ZIGBEE_short(SOFTRST, 0x02); //0x02 mask for RSTBB bit
123: }
124:
125: void MAC_reset(void){
126: write_ZIGBEE_short(SOFTRST, 0x01); //0x01 mask for RSTMAC bit
127: }

```



```

128:
129: void software_reset(void) { // PWR_reset, BB_reset and MAC_reset at
once
130: write_ZIGBEE_short(SOFTRST, 0x07);
131: }
132:
133: void RF_reset(void) {
134: short int temp = 0;
135: temp = read_ZIGBEE_short(RFCTL);
136: temp = temp | 0x04; //mask for RFRST bit
137: write_ZIGBEE_short(RFCTL, temp);
138: temp = temp & (!0x04); //mask for RFRST bit
139: write_ZIGBEE_short(RFCTL, temp);
140: Delay_ms(1);
141: }
142:
143: /*
144: * Interrupt
145: */
146: void enable_interrupt(void)
147: {
148: write_ZIGBEE_short(INTCON_M, 0x00); //0x00 all interrupts are
enable
149: }
150:
151: /*
152: * Set channel
153: */
154: void set_channel(short int channel_number) { // 11-26 possible
channels
155: if((channel_number > 26) || (channel_number < 11)) channel_number
= 11;
156: switch(channel_number) {
157: case 11:
158: write_ZIGBEE_long(RFCON0, 0x02); //0x02 for 11. channel
159: break;
160: case 12:
161: write_ZIGBEE_long(RFCON0, 0x12); //0x12 for 12. channel
162: break;
163: case 13:
164: write_ZIGBEE_long(RFCON0, 0x22); //0x22 for 13. channel
165: break;
166: case 14:
167: write_ZIGBEE_long(RFCON0, 0x32); //0x32 for 14. channel
168: break;
169: case 15:
170: write_ZIGBEE_long(RFCON0, 0x42); //0x42 for 15. channel
171: break;
172: case 16:
173: write_ZIGBEE_long(RFCON0, 0x52); //0x52 for 16. channel
174: break;
175: case 17:
176: write_ZIGBEE_long(RFCON0, 0x62); //0x62 for 17. channel
177: break;
178: case 18:
179: write_ZIGBEE_long(RFCON0, 0x72); //0x72 for 18. channel
180: break;
181: case 19:
182: write_ZIGBEE_long(RFCON0, 0x82); //0x82 for 19. channel
183: break;
184: case 20:
185: write_ZIGBEE_long(RFCON0, 0x92); //0x92 for 20. channel
186: break;
187: case 21:
188: write_ZIGBEE_long(RFCON0, 0xA2); //0xA2 for 21. channel
189: break;
190: case 22:
191: write_ZIGBEE_long(RFCON0, 0xB2); //0xB2 for 22. channel
192: break;

```

```

193: case 23:
194: write_ZIGBEE_long(RFCON0, 0xC2); //0xC2 for 23. channel
195: break;
196: case 24:
197: write_ZIGBEE_long(RFCON0, 0xD2); //0xD2 for 24. channel
198: break;
199: case 25:
200: write_ZIGBEE_long(RFCON0, 0xE2); //0xE2 for 25. Channel
201: break;
202: case 26:
203: write_ZIGBEE_long(RFCON0, 0xF2); //0xF2 for 26. channel
204: break;
205: }
206: RF_reset();
207: }
208:
209: /*
210: * Set CCA mode
211: */
212: void set_CCA_mode(short int CCA_mode){
213: short int temp = 0;
214: switch(CCA_mode){
215: case 1: //ENERGY ABOVE THRESHOLD
216: {
217: temp = read_ZIGBEE_short(BBREG2);
218: temp = temp | 0x80; //0x80 mask
219: temp = temp & 0xDF; //0xDF mask
220: write_ZIGBEE_short(BBREG2, temp);
221:
222: write_ZIGBEE_short(CCAEDTH, 0x60); //Set CCA ED threshold to -69
dBm
223: }
224: break;
225:
226: case 2: //CARRIER SENSE ONLY
227: {
228: temp = read_ZIGBEE_short(BBREG2);
229: temp = temp | 0x40; // 0x40 mask
230: temp = temp & 0x7F; // 0x7F mask
231: write_ZIGBEE_short(BBREG2, temp);
232:
233: temp = read_ZIGBEE_short(BBREG2); // carrier sense threshold
234: temp = temp | 0x38;
235: temp = temp & 0xFB;
236: write_ZIGBEE_short(BBREG2, temp);
237: }
238: break;
239:
240: case 3: //CARRIER SENSE AND ENERGY ABOVE THRESHOLD
241: {
242: temp = read_ZIGBEE_short(BBREG2);
243: temp = temp | 0xC0; //0xC0 mask
244: write_ZIGBEE_short(BBREG2, temp);
245:
246: temp = read_ZIGBEE_short(BBREG2); // carrier sense threshold
247: temp = temp | 0x38; // 0x38 mask
248: temp = temp & 0xFB; // 0xFB mask
249: write_ZIGBEE_short(BBREG2, temp);
250:
251: write_ZIGBEE_short(CCAEDTH, 0x60); //Set CCA ED threshold to -69
dBm
252: }
253: break;
254: }
255: }
256:
257: /*
258: * Set RSSI mode
259: */

```

```

260: void set_RSSI_mode(short int RSSI_mode){ // 1 for RSSI1, 2 for
RSSI2 mode
261: short int temp = 0;
262:
263: switch(RSSI_mode){
264: case 1:
265: {
266: temp = read_ZIGBEE_short(BBREG6);
267: temp = temp | 0x80; //0x80 mask for RSSI1 mode
268: write_ZIGBEE_short(BBREG6, temp);
269: }
270: break;
271:
272: case 2:
273: write_ZIGBEE_short(BBREG6, 0x40); //0x40 data for RSSI2 mode
274: break;
275: }
276: }
277:
278: /*
279: * Set type of device
280: */
281: void nonbeacon_PAN_coordinator_device(void){
282: short int temp = 0;
283:
284: temp = read_ZIGBEE_short(RXMCR);
285: temp = temp | 0x08; // 0x08 mask for PAN coordinator
286: write_ZIGBEE_short(RXMCR, temp);
287:
288: temp = read_ZIGBEE_short(TXMCR);
289: temp = temp & 0xDF; // 0xDF mask for CSMA-CA mode
290: write_ZIGBEE_short(TXMCR, temp);
291:
292: write_ZIGBEE_short(ORDER, 0xFF); // BO, SO are 15
293: }
294:
295: void nonbeacon_coordinator_device(void){
296: short int temp = 0;
297:
298: temp = read_ZIGBEE_short(RXMCR);
299: temp = temp | 0x04; // 0x04 mask for coordinator
300: write_ZIGBEE_short(RXMCR, temp);
301:
302: temp = read_ZIGBEE_short(TXMCR);
303: temp = temp & 0xDF; // 0xDF mask for CSMA-CA mode
304: write_ZIGBEE_short(TXMCR, temp);
305:
306: write_ZIGBEE_short(ORDER, 0xFF); // BO, SO are 15
307: }
308:
309: void nonbeacon_device(void){
310: short int temp = 0;
311:
312: temp = read_ZIGBEE_short(RXMCR);
313: temp = temp & 0xF3; // 0xF3 mask for PAN coordinator and
coordinator
314: write_ZIGBEE_short(RXMCR, temp);
315:
316: temp = read_ZIGBEE_short(TXMCR);
317: temp = temp & 0xDF; // 0xDF mask for CSMA-CA mode
318: write_ZIGBEE_short(TXMCR, temp);
319: }
320:
321: /*
322: * ACK request
323: */
324: void set_ACK(void){
325: short int temp = 0;
326:

```

```

327: temp = read_ZIGBEE_short(TXNCON);
328: temp = temp | 0x04; //0x04 mask for set ACK
329: write_ZIGBEE_short(TXNCON, temp);
330: }
331:
332: void set_not_ACK(void){
333: short int temp = 0;
334:
335: temp = read_ZIGBEE_short(TXNCON);
336: temp = temp & (!0x04); //0x04 mask for set not ACK
337: write_ZIGBEE_short(TXNCON, temp);
338: }
339:
340: /*
341: * Encrypt
342: */
343: void set_encrypt(void){
344: short int temp = 0;
345:
346: temp = read_ZIGBEE_short(TXNCON);
347: temp = temp | 0x02; // mask for set encrypt
348: write_ZIGBEE_short(TXNCON, temp);
349: }
350:
351: void set_not_encrypt(void){
352: short int temp = 0;
353:
354: temp = read_ZIGBEE_short(TXNCON);
355: temp = temp & (!0x02); // mask for set not encrypt
356: write_ZIGBEE_short(TXNCON, temp);
357: }
358:
359: /*
360: * Transmit packet
361: */
362: void start_transmit(void){
363: short int temp = 0;
364:
365: temp = read_ZIGBEE_short(TXNCON);
366: temp = temp | 0x01; // mask for start transmit
367: write_ZIGBEE_short(TXNCON, temp);
368: }
369:
370: /*
371: * Interframe spacing
372: */
373: void set_IFS_recomended(void){
374: short int temp = 0;
375:
376: write_ZIGBEE_short(RXMCR, 0x93); // Min SIFS Period
377:
378: temp = read_ZIGBEE_short(TXPEND);
379: temp = temp | 0x7C; // MinLIFSPeriod
380: write_ZIGBEE_short(TXPEND, temp);
381:
382: temp = read_ZIGBEE_short(TXSTBL);
383: temp = temp | 0x90; // MinLIFSPeriod
384: write_ZIGBEE_short(TXSTBL, temp);
385:
386: temp = read_ZIGBEE_short(TXTIME);
387: temp = temp | 0x31; // TurnaroundTime
388: write_ZIGBEE_short(TXTIME, temp);
389: }
390:
391: void set_IFS_default(void){
392: short int temp = 0;
393:
394: write_ZIGBEE_short(RXMCR, 0x75); // Min SIFS Period
395:

```

```

396: temp = read_ZIGBEE_short(TXPEND);
397: temp = temp | 0x84; // Min LIFS Period
398: write_ZIGBEE_short(TXPEND, temp);
399:
400: temp = read_ZIGBEE_short(TXSTBL);
401: temp = temp | 0x50; // Min LIFS Period
402: write_ZIGBEE_short(TXSTBL, temp);
403:
404: temp = read_ZIGBEE_short(TXTIME);
405: temp = temp | 0x41; // Turnaround Time
406: write_ZIGBEE_short(TXTIME, temp);
407: }
408:
409: /*
410:  * Reception mode
411:  */
412: void set_reception_mode(short int r_mode){ // 1 normal, 2 error,
3 promiscuous mode
413: short int temp = 0;
414:
415: switch(r_mode)
416: {
417: case 1:
418: {
419: temp = read_ZIGBEE_short(RXMCR); // normal mode
420: temp = temp & (!0x03); // mask for normal mode
421: write_ZIGBEE_short(RXMCR, temp);
422: }
423: break;
424:
425: case 2:
426: {
427: temp = read_ZIGBEE_short(RXMCR); // error mode
428: temp = temp & (!0x01); // mask for error mode
429: temp = temp | 0x02; // mask for error mode
430: write_ZIGBEE_short(RXMCR, temp);
431: }
432: break;
433:
434: case 3:
435: {
436: temp = read_ZIGBEE_short(RXMCR); // promiscuous mode
437: temp = temp & (!0x02); // mask for promiscuous mode
438: temp = temp | 0x01; // mask for promiscuous mode
439: write_ZIGBEE_short(RXMCR, temp);
440:
441: break;
442: }
443: }
444:
445: /*
446:  * Frame format filter
447:  */
448: void set_frame_format_filter(short int fff_mode){ // 1 all
frames, 2 command only, 3 data only, 4 beacon only
449: short int temp = 0;
450:
451: switch(fff_mode)
452: {
453: case 1:
454: {
455: temp = read_ZIGBEE_short(RXFLUSH); // all frames
456: temp = temp & (!0x0E); // mask for all frames
457: write_ZIGBEE_short(RXFLUSH, temp);
458: }
459: break;
460:
461: case 2:
462: {

```

```

463: temp = read_ZIGBEE_short(RXFLUSH); // command only
464: temp = temp & (!0x06); // mask for command only
465: temp = temp | 0x08; // mask for command only
466: write_ZIGBEE_short(RXFLUSH, temp);
467: }
468: break;
469:
470: case 3:
471: {
472: temp = read_ZIGBEE_short(RXFLUSH); // data only
473: temp = temp & (!0x0A); // mask for data only
474: temp = temp | 0x04; // mask for data only
475: write_ZIGBEE_short(RXFLUSH, temp);
476: }
477: break;
478:
479: case 4:
480: {
481: temp = read_ZIGBEE_short(RXFLUSH); // beacon only
482: temp = temp & (!0x0C); // mask for beacon only
483: temp = temp | 0x02; // mask for beacon only
484: write_ZIGBEE_short(RXFLUSH, temp);
485: }
486: break;
487: }
488: }
489:
490: /*
491: * Flush RX FIFO pointer
492: */
493: void flush_RX_FIFO_pointer(void){
494: short int temp;
495:
496: temp = read_ZIGBEE_short(RXFLUSH);
497: temp = temp | 0x01; // mask for flush RX FIFO
498: write_ZIGBEE_short(RXFLUSH, temp);
499: }
500:
501: /*
502: * FIFO
503: */
504: void read_RX_FIFO(void){
505: unsigned short int temp = 0;
506: int i = 0;
507:
508: temp = read_ZIGBEE_short(BBREG1); // disable receiving packets
off air.
509: temp = temp | 0x04; // mask for disable receiving packets
510: write_ZIGBEE_short(BBREG1, temp);
511:
512: for(i=0; i<128; i++)
513: {
514: if(i < (1 + DATA_LENIGHT + HEADER_LENIGHT + 2 + 1 + 1))
515: data_RX_FIFO[i] = read_ZIGBEE_long(address_RX_FIFO + i); //
reading valid data from RX FIFO
516: if(i >= (1 + DATA_LENIGHT + HEADER_LENIGHT + 2 + 1 + 1))
517: lost_data = read_ZIGBEE_long(address_RX_FIFO + i); // reading
invalid data from RX FIFO
518: }
519:
520: DATA_RX[0] = data_RX_FIFO[HEADER_LENIGHT + 1]; // coping valid
data
521: DATA_RX[1] = data_RX_FIFO[HEADER_LENIGHT + 2]; // coping valid
data
522: DATA_RX[2] = data_RX_FIFO[HEADER_LENIGHT + 3]; // coping valid
data
523: LQI = data_RX_FIFO[1 + HEADER_LENIGHT + DATA_LENIGHT + 2]; //
coping valid data

```

```

524: RSSI2 = data_RX_FIFO[1 + HEADER_LENHT + DATA_LENHT + 3]; //
coping valid data
525:
526: temp = read_ZIGBEE_short(BBREG1); // enable receiving packets off
air.
527: temp = temp & (!0x04); // mask for enable receiving
528: write_ZIGBEE_short(BBREG1, temp);
529:
530: }
531:
532: void write_TX_normal_FIFO(void){
533: int i = 0;
534:
535: data_TX_normal_FIFO[0] = HEADER_LENHT;
536: data_TX_normal_FIFO[1] = HEADER_LENHT + DATA_LENHT;
537: data_TX_normal_FIFO[2] = 0x01; // control frame
538: data_TX_normal_FIFO[3] = 0x88;
539: //data_TX_normal_FIFO[4] = SEQ_NUMBER; // sequence number
540: data_TX_normal_FIFO[5] = PAN_ID_2[1]; // destinatoin pan
541: data_TX_normal_FIFO[6] = PAN_ID_2[0];
542: data_TX_normal_FIFO[7] = ADDRESS_short_2[0]; // destination
address
543: data_TX_normal_FIFO[8] = ADDRESS_short_2[1];
544: data_TX_normal_FIFO[9] = PAN_ID_1[0]; // source pan
545: data_TX_normal_FIFO[10] = PAN_ID_1[1];
546: data_TX_normal_FIFO[11] = ADDRESS_short_1[0]; // source address
547: data_TX_normal_FIFO[12] = ADDRESS_short_1[1];
548:
549: data_TX_normal_FIFO[13] = DATA_TX[0]; // data
550: data_TX_normal_FIFO[14] = DATA_TX[1];
551: data_TX_normal_FIFO[15] = DATA_TX[2];
552:
553: for(i = 0; i < (HEADER_LENHT + DATA_LENHT + 2); i++)
554: {
555: write_ZIGBEE_long(address_TX_normal_FIFO + i,
data_TX_normal_FIFO[i]); // write frame into normal FIFO
556: }
557:
558: set_not_ACK();
559: set_not_encrypt();
560: start_transmit();
561:
562: }
563:
564: /*
565: * Address
566: */
567: void set_short_address(short int * address){
568: write_ZIGBEE_short(SADRL, address[0]);
569: write_ZIGBEE_short(SADRH, address[1]);
570: }
571:
572: void set_long_address(short int * address){
573: short int i = 0;
574:
575: for(i = 0; i < 8; i++)
576: {
577: write_ZIGBEE_short(EADR0 + i, address[i]); // 0x05 address of
EADR0
578: }
579: }
580:
581: void set_PAN_ID(short int * address){
582: write_ZIGBEE_short(PANIDL, address[0]);
583: write_ZIGBEE_short(PANIDH, address[1]);
584: }
585:
586: /*
587: * Wake

```

```

588: */
589: void set_wake_from_pin(void){
590: short int temp = 0;
591:
592: WAKE = 0;
593: temp = read_ZIGBEE_short(RXFLUSH);
594: temp = temp | 0x60; // mask
595: write_ZIGBEE_short(RXFLUSH, temp);
596:
597: temp = read_ZIGBEE_short(WAKECON);
598: temp = temp | 0x80;
599: write_ZIGBEE_short(WAKECON, temp);
600: }
601:
602: void pin_wake(void){
603: WAKE = 1;
604: Delay_ms(5);
605: }
606:
607: /*
608: * PLL
609: */
610: void enable_PLL(void){
611: write_ZIGBEE_long(RFCON2, 0x80); // mask for PLL enable
612: }
613:
614: void disable_PLL(void){
615: write_ZIGBEE_long(RFCON2, 0x00); // mask for PLL disable
616: }
617:
618: /*
619: * Tx power
620: */
621: void set_TX_power(unsigned short int power){ // 0-31 possible
variants
622: if((power < 0) || (power > 31)) power = 31;
623: power = 31 - power; //0 max, 31 min -> 31 max, 0 min
624: power = ((power & 0b00011111) << 3) & 0b11111000; // calculating
power
625: write_ZIGBEE_long(RFCON3, power);
626: }
627:
628: /*
629: * Init ZIGBEE module
630: */
631: void init_ZIGBEE_basic(void){
632: write_ZIGBEE_short(PACON2, 0x98); // Initialize FIFOEN = 1 and
TXONTS = 0x6
633: write_ZIGBEE_short(TXSTBL, 0x95); // Initialize RFSTBL = 0x9
634: write_ZIGBEE_long(RFCON1, 0x01); // Initialize VCOOPT = 0x01
635: enable_PLL(); // Enable PLL (PLLEN = 1)
636: write_ZIGBEE_long(RFCON6, 0x90); // Initialize TXFIL = 1 and
20MRECVR = 1
637: write_ZIGBEE_long(RFCON7, 0x80); // Initialize SLPCLKSEL = 0x2
(100 kHz Internal oscillator)
638: write_ZIGBEE_long(RFCON8, 0x10); // Initialize RFVCO = 1
639: write_ZIGBEE_long(SLPCON1, 0x21); // Initialize CLKOUTEN = 1 and
SLPCLKDIV = 0x01
640: }
641:
642: void init_ZIGBEE_nonbeacon(void){
643: init_ZIGBEE_basic();
644: set_CCA_mode(1); // Set CCA mode to ED and set threshold
645: set_RSSI_mode(2); // RSSI2 mode
646: enable_interrupt(); // Enables all interrupts
647: set_channel(11); // Channel 11
648: RF_reset();
649: }
650:

```



```

651: char debauns_INTn(void){
652: char i = 0, j = 0, intrn_d = 0;
653: for(i = 0; i < 5; i++)
654: {
655: intrn_d = INT;
656: if (intrn_d == 1) j++;
657: }
658: if(j > 2) return 1;
659: else return 0;
660: }
661:
662:
663: /*
664: * Main
665: */
666:
667: char RSSI2X;
668: char RSSI2Y;
669: char persona;
670: char personal;
671: char persona2;
672:
673: void main(){
674:
675: short int temp1 = 0;
676:
677: ANSEL = 0; // Configure AN pins as digital
678: ANSELH = 0;
679: C1ON_bit = 0; // Disable comparators
680: C2ON_bit = 0;
681: GIE_bit = 0; // Disable interrupts
682:
683: RSSI2 = 0;
684:
685: TRISA = 0x00; // Set direction to be output
686: TRISB = 0x00; // Set direction to be output
687: TRISC = 0x00; // Set direction to be output
688: TRISD = 0x00; // Set direction to be output
689:
690: CS_Direction = 0; // Set direction to be output
691: RST_Direction = 0; // Set direction to be output
692: INT_Direction = 1; // Set direction to be input
693: WAKE_Direction = 0; // Set direction to be output
694:
695:
696: Delay_ms(5);
697: // Initialize SPI module
698: SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV64,
699: _SPI_DATA_SAMPLE_MIDDLE, _SPI_CLK_IDLE_LOW, _SPI_LOW_2_HIGH);
700: pin_reset(); // Activate reset from pin
701: software_reset(); // Activate software reset
702: RF_reset(); // RF reset
703: set_WAKE_from_pin(); // Set wake from pin
704:
705: set_long_address(ADDRESS_long_2); // Set long address
706: set_short_address(ADDRESS_short_2); // Set short address
707: set_PAN_ID(PAN_ID_2); // Set PAN_ID
708:
709: init_ZIGBEE_nonbeacon(); // Initialize ZigBee module
710: nonbeacon_PAN_coordinator_device();
711: set_TX_power(31); // Set max TX power
712: set_frame_format_filter(1); // 1 all frames, 3 data frame only
713: set_reception_mode(1); // 1 normal mode
714:
715: pin_wake(); // Wake from pin
716:
717: Lcd_Init(); // Initialize LCD
718: Lcd_Cmd(_LCD_CLEAR); // Clear display
719: Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off

```

```

719: Lcd_Out(1,1,"Received :"); // Write on LCD
720:
721: ByteToStr(0,&txt);
722: UART1_Init(9600);
723: while(1){ // Infinite loop
724: if( debauns_INtn() == 0 ){ // Polling line INT
725: temp1 = read_ZIGBEE_short(INTSTAT); // Read and flush register
INTSTAT
726: read_RX_FIFO(); // Read receive data
727: ByteToStr(DATA_RX[2],&txt); // Convert third byte to string
728: Lcd_Chr(2,1,DATA_RX[0]); // Display first received byte
729: Lcd_Chr(2,2,DATA_RX[1]); // Display second received byte
730: Lcd_Out(2,5,txt); // Display third received byte
731: ByteToStr(RSSI2,&txt2); // Convert third byte to string
732: Lcd_Out(2,9,txt2);
733:
734:
//*****
*****
735: persona=DATA_RX[2];
736:
737:
738: if(persona==1){
739: //para definir si es usuario 1 o 2
740: personal=persona;
741: RSSI2X=RSSI2;
742: if(RSSI2X>=139 && RSSI2X<=255){
743: Lcd_Out(2,14,"U1"); // Display third received
byte//UART1_Write_Text("U1");
744: UART1_Write_Text("U1");
745: }
746: if(RSSI2X>=80 && RSSI2X<=100){
747: Lcd_Out(2,14,"U2");//UART1_Write_Text("U2");
748: UART1_Write_Text("U2");
749: }
750:
751: if(RSSI2X>=12 && RSSI2X<=47){
752: Lcd_Out(2,14,"U3");//UART1_Write_Text("U2");
753: UART1_Write_Text("U3");
754: }
755:
756: if(RSSI2X>=0 && RSSI2X<=9){
757: Lcd_Out(2,14,"U4");//UART1_Write_Text("U2");
758: UART1_Write_Text("U4");
759: }
760:
761: }
762:
763:
764:
765:
766:
767: if(persona==2){
768: //para definir si es usuario 1 o 2
769: persona2=persona;
770: RSSI2Y=RSSI2;
771: if(RSSI2Y>=139 && RSSI2Y<=255){
772: Lcd_Out(2,14,"V1"); // Display third received
byte//UART1_Write_Text("U1");
773: UART1_Write_Text("V1");
774: }
775: if(RSSI2Y>=80 && RSSI2Y<=100){
776: Lcd_Out(2,14,"V2");//UART1_Write_Text("U2");
777: UART1_Write_Text("V2");
778: }
779:
780: if(RSSI2Y>=12 && RSSI2Y<=47){
781: Lcd_Out(2,14,"V3");//UART1_Write_Text("U2");
782: UART1_Write_Text("V3");

```

```
783: }
784:
785: if(RSSI2Y>=0 && RSSI2Y<=9){
786: Lcd_Out(2,14,"V4");//UART1_Write_Text("U2");
787: UART1_Write_Text("V4");
788: }
789:
790: }
791:
792:
793: //*****
794: *****
795:
796: }
797: }
```

Anexo N.2

Código de programación del dispositivo móvil

Código de programación del dispositivo móvil

```
1:
2:
3: #include "registers.h"
4:
5: // EasyBee3 connections
6: sbit CS at RA0_bit; //CS pin
7: sbit RST at RA1_bit; //RST pin
8: sbit INT at RA2_bit; //INT pin
9: sbit WAKE at RA3_bit; //WAKE pin
10:
11: sbit CS_Direction at TRISA0_bit; //CS pin direction
12: sbit RST_Direction at TRISA1_bit; //RST pin direction
13: sbit INT_Direction at TRISA2_bit; //INT pin direction
14: sbit WAKE_Direction at TRISA3_bit; //WAKE pin direction
15:
16: const unsigned short int DATA LENGHT = 3;
17: const unsigned short int HEADER LENGHT = 11;
18:
19: int address_RX_FIFO = 0x300;
20: int address_TX_normal_FIFO = 0x000;
21: short int data_RX_FIFO[1 + HEADER LENGHT + DATA LENGHT + 2 + 1 +
1];
22: short int lost_data = 0;
23:
24: short int ADDRESS_short_1[] = {1,1}; // Source address
25: short int ADDRESS_long_1[] = {1,1,1,1,1,1,1,1};
26: short int ADDRESS_short_2[] = {2,2}; // Destination address
27: short int ADDRESS_long_2[] = {2,2,2,2,2,2,2,2};
28: short int PAN_ID_1[] = {3,3}; // Source PAN ID
29: short int PAN_ID_2[] = {3,3}; // Destination PAN ID
30:
31: short int DATA_RX[DATA LENGHT], DATA_TX[DATA LENGHT],
data_TX_normal_FIFO[DATA LENGHT + HEADER LENGHT + 2];
32: short int LQI = 0, RSSI2 = 0, SEQ_NUMBER = 0x23;
33:
34: /*
```

```

35: * Functions for reading and writing registers in short address
memory space
36: */
37:
38: // write data in short address register
39: void write_ZIGBEE_short(short int address, short int data_r){
40: CS = 0;
41:
42: address = ((address << 1) & 0b01111111) | 0x01; // calculating
addressing mode
43: SPI1_Write(address); // addressing register
44: SPI1_Write(data_r); // write data in register
45:
46: CS = 1;
47: }
48:
49: // read data from short address register
50: short int read_ZIGBEE_short(short int address)
51: {
52: short int data_r = 0, dummy_data_r = 0;
53:
54: CS = 0;
55:
56: address = (address << 1) & 0b01111110; // calculating addressing
mode
57: SPI1_Write(address); // addressing register
58: data_r = SPI1_Read(dummy_data_r); // read data from register
59:
60: CS = 1;
61: return data_r;
62: }
63:
64: /*
65: * Functions for reading and writing registers in long address
memory space
66: */
67:
68: // Write data in long address register
69: void write_ZIGBEE_long(int address, short int data_r){
70: short int address_high = 0, address_low = 0;
71:
72: CS = 0;
73:
74: address_high = (((short int)(address >> 3)) & 0b01111111) | 0x80;
// calculating addressing mode
75: address_low = (((short int)(address << 5)) & 0b11100000) | 0x10;
// calculating addressing mode
76: SPI1_Write(address_high); // addressing register
77: SPI1_Write(address_low); // addressing register
78: SPI1_Write(data_r); // write data in register
79:
80: CS = 1;
81: }
82:
83: // Read data from long address register
84: short int read_ZIGBEE_long(int address){
85: short int data_r = 0, dummy_data_r = 0;
86: short int address_high = 0, address_low = 0;
87:
88: CS = 0;
89:
90: address_high = ((short int)(address >> 3) & 0b01111111) | 0x80;
//calculating addressing mode
91: address_low = ((short int)(address << 5) & 0b11100000);
//calculating addressing mode
92: SPI1_Write(address_high); // addressing register
93: SPI1_Write(address_low); // addressing register
94: data_r = SPI1_Read(dummy_data_r); // read data from register
95:

```

```

96: CS = 1;
97: return data_r;
98: }
99:
100: /*
101:  * Reset functions
102:  */
103:
104: // Reset from pin
105: void pin_reset(void)
106: {
107: RST = 0; // activate reset
108: Delay_ms(5);
109: RST = 1; // deactivate reset
110: Delay_ms(5);
111: }
112:
113: void PWR_reset(void){
114: write_ZIGBEE_short(SOFTRST, 0x04); //0x04 mask for RSTPWR bit
115: }
116:
117: void BB_reset(void){
118: write_ZIGBEE_short(SOFTRST, 0x02); //0x02 mask for RSTBB bit
119: }
120:
121: void MAC_reset(void){
122: write_ZIGBEE_short(SOFTRST, 0x01); //0x01 mask for RSTMAC bit
123: }
124:
125: void software_reset(void){ // PWR_reset, BB_reset and MAC_reset at
once
126: write_ZIGBEE_short(SOFTRST, 0x07);
127: }
128:
129: void RF_reset(void){
130: short int temp = 0;
131: temp = read_ZIGBEE_short(RFCTL);
132: temp = temp | 0x04; //mask for RFRST bit
133: write_ZIGBEE_short(RFCTL, temp);
134: temp = temp & (!0x04); //mask for RFRST bit
135: write_ZIGBEE_short(RFCTL, temp);
136: Delay_ms(1);
137: }
138:
139: /*
140:  * Interrupt
141:  */
142: void enable_interrupt(void)
143: {
144: write_ZIGBEE_short(INTCON_M, 0x00); //0x00 all interrupts are
enable
145: }
146:
147: /*
148:  * Set channel
149:  */
150: void set_channel(short int channel_number){ // 11-26 possible
channels
151: if((channel_number > 26) || (channel_number < 11)) channel_number
= 11;
152: switch(channel_number){
153: case 11:
154: write_ZIGBEE_long(RFCON0, 0x02); //0x02 for 11. channel
155: break;
156: case 12:
157: write_ZIGBEE_long(RFCON0, 0x12); //0x12 for 12. channel
158: break;
159: case 13:
160: write_ZIGBEE_long(RFCON0, 0x22); //0x22 for 13. channel

```

```

161: break;
162: case 14:
163: write_ZIGBEE_long(RFCON0, 0x32); //0x32 for 14. channel
164: break;
165: case 15:
166: write_ZIGBEE_long(RFCON0, 0x42); //0x42 for 15. channel
167: break;
168: case 16:
169: write_ZIGBEE_long(RFCON0, 0x52); //0x52 for 16. channel
170: break;
171: case 17:
172: write_ZIGBEE_long(RFCON0, 0x62); //0x62 for 17. channel
173: break;
174: case 18:
175: write_ZIGBEE_long(RFCON0, 0x72); //0x72 for 18. channel
176: break;
177: case 19:
178: write_ZIGBEE_long(RFCON0, 0x82); //0x82 for 19. channel
179: break;
180: case 20:
181: write_ZIGBEE_long(RFCON0, 0x92); //0x92 for 20. channel
182: break;
183: case 21:
184: write_ZIGBEE_long(RFCON0, 0xA2); //0xA2 for 21. channel
185: break;
186: case 22:
187: write_ZIGBEE_long(RFCON0, 0xB2); //0xB2 for 22. channel
188: break;
189: case 23:
190: write_ZIGBEE_long(RFCON0, 0xC2); //0xC2 for 23. channel
191: break;
192: case 24:
193: write_ZIGBEE_long(RFCON0, 0xD2); //0xD2 for 24. channel
194: break;
195: case 25:
196: write_ZIGBEE_long(RFCON0, 0xE2); //0xE2 for 25. channel
197: break;
198: case 26:
199: write_ZIGBEE_long(RFCON0, 0xF2); //0xF2 for 26. channel
200: break;
201: }
202: RF_reset();
203: }
204:
205: /*
206: * Set CCA mode
207: */
208: void set_CCA_mode(short int CCA_mode){
209: short int temp = 0;
210: switch(CCA_mode){
211: case 1: //ENERGY ABOVE THRESHOLD
212: {
213: temp = read_ZIGBEE_short(BBREG2);
214: temp = temp | 0x80; //0x80 mask
215: temp = temp & 0xDF; //0xDF mask
216: write_ZIGBEE_short(BBREG2, temp);
217:
218: write_ZIGBEE_short(CCAEDTH, 0x60); //Set CCA ED threshold to -69
dBm
219: }
220: break;
221:
222: case 2: //CARRIER SENSE ONLY
223: {
224: temp = read_ZIGBEE_short(BBREG2);
225: temp = temp | 0x40; // 0x40 mask
226: temp = temp & 0x7F; // 0x7F mask
227: write_ZIGBEE_short(BBREG2, temp);
228:

```

```

229: temp = read_ZIGBEE_short(BBREG2); // carrier sense threshold
230: temp = temp | 0x38;
231: temp = temp & 0xFB;
232: write_ZIGBEE_short(BBREG2, temp);
233: }
234: break;
235:
236: case 3: //CARRIER SENSE AND ENERGY ABOVE THRESHOLD
237: {
238: temp = read_ZIGBEE_short(BBREG2);
239: temp = temp | 0xC0; //0xC0 mask
240: write_ZIGBEE_short(BBREG2, temp);
241:
242: temp = read_ZIGBEE_short(BBREG2); // carrier sense threshold
243: temp = temp | 0x38; // 0x38 mask
244: temp = temp & 0xFB; // 0xFB mask
245: write_ZIGBEE_short(BBREG2, temp);
246:
247: write_ZIGBEE_short(CCAEDTH, 0x60); //Set CCA ED threshold to -69
dBm
248: }
249: break;
250:
251: }
252:
253: }
254:
255: /*
256: * Set RSSI mode
257: */
258: void set_RSSI_mode(short int RSSI_mode){ // 1 for RSSI1, 2 for
RSSI2 mode
259: short int temp = 0;
260:
261: switch(RSSI_mode){
262: case 1:
263: {
264: temp = read_ZIGBEE_short(BBREG6);
265: temp = temp | 0x80; //0x80 mask for RSSI1 mode
266: write_ZIGBEE_short(BBREG6, temp);
267: }
268: break;
269:
270: case 2:
271: write_ZIGBEE_short(BBREG6, 0x40); //0x40 data for RSSI2 mode
272: break;
273: }
274: }
275:
276: /*
277: * Set type of device
278: */
279: void nonbeacon_PAN_coordinator_device(void){
280: short int temp = 0;
281:
282: temp = read_ZIGBEE_short(RXMCR);
283: temp = temp | 0x08; // 0x08 mask for PAN coordinator
284: write_ZIGBEE_short(RXMCR, temp);
285:
286: temp = read_ZIGBEE_short(TXMCR);
287: temp = temp & 0xDF; // 0xDF mask for CSMA-CA mode
288: write_ZIGBEE_short(TXMCR, temp);
289:
290: write_ZIGBEE_short(ORDER, 0xFF); // B0, S0 are 15
291: }
292:
293: void nonbeacon_coordinator_device(void){
294: short int temp = 0;
295:

```



```

296: temp = read_ZIGBEE_short(RXMCR);
297: temp = temp | 0x04; // 0x04 mask for coordinator
298: write_ZIGBEE_short(RXMCR, temp);
299:
300: temp = read_ZIGBEE_short(TXMCR);
301: temp = temp & 0xDF; // 0xDF mask for CSMA-CA mode
302: write_ZIGBEE_short(TXMCR, temp);
303:
304: write_ZIGBEE_short(ORDER, 0xFF); // B0, S0 are 15
305: }
306:
307: void nonbeacon_device(void){
308: short int temp = 0;
309:
310: temp = read_ZIGBEE_short(RXMCR);
311: temp = temp & 0xF3; // 0xF3 mask for PAN coordinator and
coordinator
312: write_ZIGBEE_short(RXMCR, temp);
313:
314: temp = read_ZIGBEE_short(TXMCR);
315: temp = temp & 0xDF; // 0xDF mask for CSMA-CA mode
316: write_ZIGBEE_short(TXMCR, temp);
317: }
318:
319: /*
320: * ACK request
321: */
322: void set_ACK(void){
323: short int temp = 0;
324:
325: temp = read_ZIGBEE_short(TXNCON);
326: temp = temp | 0x04; //0x04 mask for set ACK
327: write_ZIGBEE_short(TXNCON, temp);
328: }
329:
330: void set_not_ACK(void){
331: short int temp = 0;
332:
333: temp = read_ZIGBEE_short(TXNCON);
334: temp = temp & (!0x04); //0x04 mask for set not ACK
335: write_ZIGBEE_short(TXNCON, temp);
336: }
337:
338: /*
339: * Encrypt
340: */
341: void set_encrypt(void){
342: short int temp = 0;
343:
344: temp = read_ZIGBEE_short(TXNCON);
345: temp = temp | 0x02; // mask for set encrypt
346: write_ZIGBEE_short(TXNCON, temp);
347: }
348:
349: void set_not_encrypt(void){
350: short int temp = 0;
351:
352: temp = read_ZIGBEE_short(TXNCON);
353: temp = temp & (!0x02); // mask for set not encrypt
354: write_ZIGBEE_short(TXNCON, temp);
355: }
356:
357: /*
358: * Transmit packet
359: */
360: void start_transmit(void){
361: short int temp = 0;
362:
363: temp = read_ZIGBEE_short(TXNCON);

```

```

364: temp = temp | 0x01; // mask for start transmit
365: write_ZIGBEE_short(TXNCON, temp);
366: }
367:
368: /*
369:  * Interframe spacing
370:  */
371: void set_IFS_recomended(void){
372:     short int temp = 0;
373:
374:     write_ZIGBEE_short(RXMCR, 0x93); // Min SIFS Period
375:
376:     temp = read_ZIGBEE_short(TXPEND);
377:     temp = temp | 0x7C; // MinLIFSPeriod
378:     write_ZIGBEE_short(TXPEND, temp);
379:
380:     temp = read_ZIGBEE_short(TXSTBL);
381:     temp = temp | 0x90; // MinLIFSPeriod
382:     write_ZIGBEE_short(TXSTBL, temp);
383:
384:     temp = read_ZIGBEE_short(TXTIME);
385:     temp = temp | 0x31; // TurnaroundTime
386:     write_ZIGBEE_short(TXTIME, temp);
387: }
388:
389: void set_IFS_default(void){
390:     short int temp = 0;
391:
392:     write_ZIGBEE_short(RXMCR, 0x75); // Min SIFS Period
393:
394:     temp = read_ZIGBEE_short(TXPEND);
395:     temp = temp | 0x84; // Min LIFS Period
396:     write_ZIGBEE_short(TXPEND, temp);
397:
398:     temp = read_ZIGBEE_short(TXSTBL);
399:     temp = temp | 0x50; // Min LIFS Period
400:     write_ZIGBEE_short(TXSTBL, temp);
401:
402:     temp = read_ZIGBEE_short(TXTIME);
403:     temp = temp | 0x41; // Turnaround Time
404:     write_ZIGBEE_short(TXTIME, temp);
405: }
406:
407: /*
408:  * Reception mode
409:  */
410: void set_reception_mode(short int r_mode){ // 1 normal, 2 error,
3 promiscuous mode
411:     short int temp = 0;
412:
413:     switch(r_mode)
414:     {
415:     case 1:
416:     {
417:         temp = read_ZIGBEE_short(RXMCR); // normal mode
418:         temp = temp & (!0x03); // mask for normal mode
419:         write_ZIGBEE_short(RXMCR, temp);
420:     }
421:     break;
422:
423:     case 2:
424:     {
425:         temp = read_ZIGBEE_short(RXMCR); // error mode
426:         temp = temp & (!0x01); // mask for error mode
427:         temp = temp | 0x02; // mask for error mode
428:         write_ZIGBEE_short(RXMCR, temp);
429:     }
430:     break;
431:

```

```

432: case 3:
433: {
434: temp = read_ZIGBEE_short(RXMCR); // promiscuous mode
435: temp = temp & (!0x02); // mask for promiscuous mode
436: temp = temp | 0x01; // mask for promiscuous mode
437: write_ZIGBEE_short(RXMCR, temp)
438: }
439: break;
440: }
441: }
442:
443: /*
444: * Frame format filter
445: */
446: void set_frame_format_filter(short int fff_mode){ // 1 all
frames, 2 command only, 3 data only, 4 beacon only
447: short int temp = 0;
448:
449: switch(fff_mode)
450: {
451: case 1:
452: {
453: temp = read_ZIGBEE_short(RXFLUSH); // all frames
454: temp = temp & (!0x0E); // mask for all frames
455: write_ZIGBEE_short(RXFLUSH, temp);
456: }
457: break;
458:
459: case 2:
460: {
461: temp = read_ZIGBEE_short(RXFLUSH); // command only
462: temp = temp & (!0x06); // mask for command only
463: temp = temp | 0x08; // mask for command only
464: write_ZIGBEE_short(RXFLUSH, temp);
465: }
466: break;
467:
468: case 3:
469: {
470: temp = read_ZIGBEE_short(RXFLUSH); // data only
471: temp = temp & (!0x0A); // mask for data only
472: temp = temp | 0x04; // mask for data only
473: write_ZIGBEE_short(RXFLUSH, temp);
474: }
475: break;
476:
477: case 4:
478: {
479: temp = read_ZIGBEE_short(RXFLUSH); // beacon only
480: temp = temp & (!0x0C); // mask for beacon only
481: temp = temp | 0x02; // mask for beacon only
482: write_ZIGBEE_short(RXFLUSH, temp);
483: }
484: break;
485: }
486: }
487:
488: /*
489: * Flush RX FIFO pointer
490: */
491: void flush_RX_FIFO_pointer(void){
492: short int temp;
493:
494: temp = read_ZIGBEE_short(RXFLUSH);
495: temp = temp | 0x01; // mask for flush RX FIFO
496: write_ZIGBEE_short(RXFLUSH, temp);
497: }
498:
499: /*

```

```

500: * FIFO
501: */
502: void read_RX_FIFO(void){
503: unsigned short int temp = 0;
504: int i = 0;
505:
506: temp = read_ZIGBEE_short(BBREG1); // disable receiving packets
off air.
507: temp = temp | 0x04; // mask for disable receiving packets
508: write_ZIGBEE_short(BBREG1, temp);
509:
510: for(i=0; i<128; i++)
511: {
512: if(i < (1 + DATA_LENGTH + HEADER_LENGTH + 2 + 1 + 1))
513: data_RX_FIFO[i] = read_ZIGBEE_long(address_RX_FIFO + i); //
reading valid data from RX FIFO
514: if(i >= (1 + DATA_LENGTH + HEADER_LENGTH + 2 + 1 + 1))
515: lost_data = read_ZIGBEE_long(address_RX_FIFO + i); // reading
invalid data from RX FIFO
516: }
517:
518: DATA_RX[0] = data_RX_FIFO[HEADER_LENGTH + 1]; // coping valid
data
519: DATA_RX[1] = data_RX_FIFO[HEADER_LENGTH + 2]; // coping valid
data
520: DATA_RX[2] = data_RX_FIFO[HEADER_LENGTH + 3]; // coping valid
data
521: LQI = data_RX_FIFO[1 + HEADER_LENGTH + DATA_LENGTH + 2]; //
coping valid data
522: RSSI2 = data_RX_FIFO[1 + HEADER_LENGTH + DATA_LENGTH + 3]; //
coping valid data
523:
524: temp = read_ZIGBEE_short(BBREG1); // enable receiving packets off
air.
525: temp = temp & (!0x04); // mask for enable receiving
526: write_ZIGBEE_short(BBREG1, temp);
527:
528: }
529:
530: void write_TX_normal_FIFO(void){
531: int i = 0;
532:
533: data_TX_normal_FIFO[0] = HEADER_LENGTH;
534: data_TX_normal_FIFO[1] = HEADER_LENGTH + DATA_LENGTH;
535: data_TX_normal_FIFO[2] = 0x01; // control frame
536: data_TX_normal_FIFO[3] = 0x88;
537: //data_TX_normal_FIFO[4] = SEQ_NUMBER; // sequence number
538: data_TX_normal_FIFO[5] = PAN_ID_2[1]; // destinatoin pan
539: data_TX_normal_FIFO[6] = PAN_ID_2[0];
540: data_TX_normal_FIFO[7] = ADDRESS_short_2[0]; // destination
address
541: data_TX_normal_FIFO[8] = ADDRESS_short_2[1];
542: data_TX_normal_FIFO[9] = PAN_ID_1[0]; // source pan
543: data_TX_normal_FIFO[10] = PAN_ID_1[1];
544: data_TX_normal_FIFO[11] = ADDRESS_short_1[0]; // source address
545: data_TX_normal_FIFO[12] = ADDRESS_short_1[1];
546:
547: data_TX_normal_FIFO[13] = DATA_TX[0]; // data
548: data_TX_normal_FIFO[14] = DATA_TX[1];
549: data_TX_normal_FIFO[15] = DATA_TX[2];
550:
551: for(i = 0; i < (HEADER_LENGTH + DATA_LENGTH + 2); i++)
552: {
553: write_ZIGBEE_long(address_TX_normal_FIFO + i,
data_TX_normal_FIFO[i]); // write frame into normal FIFO
554: }
555:
556: set_not_ACK();
557: set_not_encrypt();

```

```

558: start_transmit();
559:
560: }
561:
562: /*
563:  * Address
564: */
565: void set_short_address(short int * address){
566: write_ZIGBEE_short(SADR_L, address[0]);
567: write_ZIGBEE_short(SADR_H, address[1]);
568: }
569:
570: void set_long_address(short int * address){
571: short int i = 0;
572:
573: for(i = 0; i < 8; i++)
574: {
575: write_ZIGBEE_short(EADR0 + i, address[i]); // 0x05 address of
EADR0
576: }
577: }
578:
579: void set_PAN_ID(short int * address){
580: write_ZIGBEE_short(PANID_L, address[0]);
581: write_ZIGBEE_short(PANID_H, address[1]);
582: }
583:
584: /*
585:  * Wake
586: */
587: void set_wake_from_pin(void){
588: short int temp = 0;
589:
590: WAKE = 0;
591: temp = read_ZIGBEE_short(RXFLUSH);
592: temp = temp | 0x60; // mask
593: write_ZIGBEE_short(RXFLUSH, temp);
594:
595: temp = read_ZIGBEE_short(WAKECON);
596: temp = temp | 0x80;
597: write_ZIGBEE_short(WAKECON, temp);
598: }
599:
600: void pin_wake(void){
601: WAKE = 1;
602: Delay_ms(5);
603: }
604:
605: /*
606:  * PLL
607: */
608: void enable_PLL(void){
609: write_ZIGBEE_long(RFCON2, 0x80); // mask for PLL enable
610: }
611:
612: void disable_PLL(void){
613: write_ZIGBEE_long(RFCON2, 0x00); // mask for PLL disable
614: }
615:
616: /*
617:  * Tx power
618: */
619: void set_TX_power(unsigned short int power){ // 0-31 possible
variants
620: if((power < 0) || (power > 31)) power = 31;
621: power = 31 - power; //0 max, 31 min -> 31 max, 0 min
622: power = ((power & 0b00011111) << 3) & 0b11111000; // calculating
power
623: write_ZIGBEE_long(RFCON3, power);

```

```

624: }
625:
626: /*
627:  * Init ZIGBEE module
628:  */
629: void init_ZIGBEE_basic(void){
630: write_ZIGBEE_short(PACON2, 0x98); // Initialize FIFOEN = 1 and
TXONTS = 0x6
631: write_ZIGBEE_short(TXSTBL, 0x95); // Initialize RFSTBL = 0x9
632: write_ZIGBEE_long(RFCON1, 0x01); // Initialize VCOOPT = 0x01
633: enable_PLL(); // Enable PLL (PLLEN = 1)
634: write_ZIGBEE_long(RFCON6, 0x90); // Initialize TXFIL = 1 and
20MRECVR = 1
635: write_ZIGBEE_long(RFCON7, 0x80); // Initialize SLPCLKSEL = 0x2
(100 kHz Internal oscillator)
636: write_ZIGBEE_long(RFCON8, 0x10); // Initialize RFVCO = 1
637: write_ZIGBEE_long(SLPCON1, 0x21); // Initialize CLKOUTEN = 1 and
SLPCLKDIV = 0x01
638: }
639:
640: void init_ZIGBEE_nonbeacon(void){
641: init_ZIGBEE_basic();
642: set_CCA_mode(1); // Set CCA mode to ED and set threshold
643: set_RSSI_mode(2); // RSSI2 mode
644: enable_interrupt(); // Enables all interrupts
645: set_channel(11); // Channel 11
646: RF_reset();
647: }
648:
649: /*
650:  * Main
651:  */
652: void main(){
653: short int temp1 = 0;
654:
655: ANSEL = 0; // Configure AN pins as digital
656: ANSELH = 0;
657: C1ON_bit = 0; // Disable comparators
658: C2ON_bit = 0;
659: GIE_bit = 0; // Disable interrupts
660:
661: TRISA = 0x00; // Set direction to be output
662: TRISB = 0x00; // Set direction to be output
663: TRISC = 0x00; // Set direction to be output
664: TRISD = 0x00; // Set direction to be output
665:
666: CS_Direction = 0; // Set direction to be output
667: RST_Direction = 0; // Set direction to be output
668: INT_Direction = 1; // Set direction to be input
669: WAKE_Direction = 0; // Set direction to be output
670:
671: Delay_ms(5);
672: // Initialize SPI module
673: SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV64,
_SPI_DATA_SAMPLE_MIDDLE, _SPI_CLK_IDLE_LOW, _SPI_LOW_2_HIGH);
674: pin_reset(); // Activate reset from pin
675: software_reset(); // Activate software reset
676: RF_reset(); // RF reset
677: set_WAKE_from_pin(); // Set wake from pin
678:
679: set_long_address(ADDRESS_long_1); // Set long address
680: set_short_address(ADDRESS_short_1); // Set short address
681: set_PAN_ID(PAN_ID_1); // Set PAN_ID
682:
683: init_ZIGBEE_nonbeacon(); // Initialize ZigBee module
684: nonbeacon_PAN_coordinator_device();
685: set_TX_power(31); // Set max TX power
686: set_frame_format_filter(1); // 1 all frames, 3 data frame only
687: set_reception_mode(1); // 1 normal mode

```

```
688:
689: pin_wake(); // Wake from pin
690:
691: DATA_TX[0] = 'm'; // First byte of data for sending
692: DATA_TX[1] = 'E'; // Second byte of data for sending
693: DATA_TX[2] = 0; // Third byte of data for sending
694:
695: PORTD = 0;
696:
697: while(1){ // Infinite loop
698: write_TX_normal_FIFO(); // Transmitting
699: Delay_ms(1000);
700: //DATA_TX[2]++; // Incremeting value
701: //PORTD++; // Showing value that is sent on PORTD
702: DATA_TX[2]=1;
703: }
704: }
```

Anexo N.3
Código de programación de la aplicación

Código de programación de la aplicación

```
Form1 - 1
Dim st As String
Dim S As String
Private Sub form_load()
MSComm1.PortOpen = True
Timer1.Interval = 1
End Sub
Private Sub Timer1_Timer()
Text2.Text = MSComm1.Input
If Text2.Text <> "" Then
'Text1.Text = Text2.Text
st = Text2.Text
S = Mid(st, 1, 1) '.....
If S = "U" Then Text1.Text = Mid(st, 2, 2)
If S = "V" Then Text3.Text = Mid(st, 2, 2)
End If
End Sub

Form1 - 1
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0";
"MSCOMM32.OCX"
Begin VB.Form Form1
Caption = "Form1"
ClientHeight = 11010
ClientLeft = 60
ClientTop = 450
ClientWidth = 11670
BeginProperty Font
Name = "MS Sans Serif"
Size = 18
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
LinkTopic = "Form1"
ScaleHeight = 11010
ScaleWidth = 11670
StartupPosition = 3 'Windows Default
Begin VB.TextBox Text3
Alignment = 2 'Center
BeginProperty Font
Name = "@Arial Unicode MS"
Size = 72
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 2355
Left = 7080
TabIndex = 8
Top = 4680
Width = 1935
End
Begin VB.TextBox Text2
Height = 495
Left = 15360
TabIndex = 4
Text = "Text2"
Top = 2760
Width = 735
End
```

```

Begin VB.Timer Timer1
Interval = 1000
Left = 15480
Top = 2280
End
Begin MSCommLib.MSComm MSComm1
Left = 15480
Top = 3360
_ExtentX = 1005
_ExtentY = 1005
_Version = 393216
DTREnable = -1 'True
End
Begin VB.TextBox Text1
Alignment = 2 'Center
BeginProperty Font
Name = "@Arial Unicode MS"
Size = 72
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 2070
Left = 7200
TabIndex = 0
Top = 1320
Width = 1815
End
Begin VB.Label Label6

Form1 - 2
Caption = "Localizador 2 está en zona :"
Height = 495
Left = 5520
TabIndex = 7
Top = 3840
Width = 5175
End
Begin VB.Label Label5
Caption = "ZONA 4"
BeginProperty Font
Name = "MS Sans Serif"
Size = 24
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 855
Left = 2040
TabIndex = 6
Top = 9240
Width = 2175
End
Begin VB.Line Line26
X1 = 5160
X2 = 5160
Y1 = 8640
Y2 = 10680
End
Begin VB.Line Line25
X1 = 720
X2 = 5160
Y1 = 10680
Y2 = 10680
End

```

```

Begin VB.Line Line24
X1 = 720
X2 = 720
Y1 = 8640
Y2 = 10680
End
Begin VB.Line Line23
X1 = 5160
X2 = 5160
Y1 = 8640
Y2 = 6120
End
Begin VB.Line Line22
X1 = 720
X2 = 5160
Y1 = 8640
Y2 = 8640
End
Begin VB.Line Line21
X1 = 720
X2 = 720
Y1 = 6120
Y2 = 8640
End
Begin VB.Label Label4
Caption = "ZONA 3"
BeginProperty Font
Name = "MS Sans Serif"
Size = 24
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 615
Left = 2040
TabIndex = 5
Top = 7080
Width = 2055
End

Form1 - 3
Begin VB.Label Label3
Caption = "ZONA 2"
BeginProperty Font
Name = "MS Sans Serif"
Size = 24
Charset = 0
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 495
Left = 2040
TabIndex = 3
Top = 4560
Width = 2055
End
Begin VB.Label Label2
BackStyle = 0 'Transparent
Caption = "ZONA 1"
BeginProperty Font
Name = "MS Sans Serif"
Size = 24
Charset = 0
Weight = 700
Underline = 0 'False

```

```

Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 615
Left = 2400
TabIndex = 2
Top = 1440
Width = 1935
End
Begin VB.Label Label1
Caption = "Localizador 1 está en zona :"
Height = 375
Left = 5520
TabIndex = 1
Top = 720
Width = 4935
End
Begin VB.Line Line20
BorderColor = &H000000FF&
X1 = 4920
X2 = 4320
Y1 = 6120
Y2 = 5640
End
Begin VB.Line Line19
X1 = 5160
X2 = 4920
Y1 = 6120
Y2 = 6120
End
Begin VB.Line Line4
X1 = 720
X2 = 4200
Y1 = 6120
Y2 = 6120
End
Begin VB.Line Line18
BorderColor = &H000000FF&
X1 = 2280
X2 = 1800
Y1 = 2760
Y2 = 2160
End
Begin VB.Line Line17
BorderColor = &H000000FF&
X1 = 2880
X2 = 3360
Y1 = 2760
Y2 = 2280
End
Begin VB.Line Line16

Form1 - 4
BorderColor = &H000000FF&
X1 = 3600
X2 = 4080
Y1 = 4440
Y2 = 4080
End
Begin VB.Line Line15
X1 = 720
X2 = 2280
Y1 = 4440
Y2 = 4440
End
Begin VB.Line Line14
X1 = 2280
X2 = 2280
Y1 = 2760

```

```
Y2 = 4440
End
Begin VB.Line Line13
X1 = 5160
X2 = 4080
Y1 = 4440
Y2 = 4440
End
Begin VB.Line Line12
X1 = 3360
X2 = 3600
Y1 = 4440
Y2 = 4440
End
Begin VB.Line Line11
X1 = 3360
X2 = 3360
Y1 = 3840
Y2 = 4440
End
Begin VB.Line Line10
X1 = 2880
X2 = 3360
Y1 = 3840
Y2 = 3840
End
Begin VB.Line Line9
X1 = 2880
X2 = 2880
Y1 = 2760
Y2 = 3840
End
Begin VB.Line Line8
X1 = 720
X2 = 2280
Y1 = 1080
Y2 = 1080
End
Begin VB.Line Line6
X1 = 2280
X2 = 2280
Y1 = 1080
Y2 = 2160
End
Begin VB.Line Line7
X1 = 2880
X2 = 2280
Y1 = 2160
Y2 = 2160
End
Begin VB.Line Line5
X1 = 5160
X2 = 2880
Y1 = 2760
Y2 = 2760
End
Begin VB.Line Line3
X1 = 5160
X2 = 5160
Y1 = 600
Y2 = 6120

Form1 - 5
End
Begin VB.Line Line2
X1 = 720
X2 = 5160
Y1 = 600
Y2 = 600
```

```
End
Begin VB.Line Line1
X1 = 720
X2 = 720
Y1 = 600
Y2 = 6120
End
End
```

Encuesta al personal de la Empresa LR Telecomunicaciones

Buenos días, estamos realizando una encuesta para evaluar el nuevo sistema de posicionamiento para la seguridad de la empresa.

Le agradeceremos brindarnos un minuto de su tiempo y responder las siguientes preguntas:

1.- ¿Estaría dispuesto a probar un nuevo sistema de seguridad?

- Si
- No

2.- ¿Le importaría usar un dispositivo electrónico como una tarjeta de seguridad?

- Si
- No

3.- El nuevo sistema de seguridad rastrea sus movimientos en su lugar de trabajo durante las horas laborables. ¿Lo gustaría que le rastreen sus movimientos?

- Si
- No

4.- ¿Cree usted que el nuevo sistema de posicionamiento, implementado en la empresa ha mejorado la seguridad del personal y de los objetos de valor?

- Si
- No

5.- ¿Qué problemas a mejorado el nuevo sistema de posicionamiento dentro de la

empresa?
