



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

**ACTUALIZACIÓN DEL SUBSISTEMA DE COMUNICACIONES
INALÁMBRICAS PARA EL SATELITE EDUCATIVO SATEDU**

TESIS

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN TELECOMUNICACIONES

PRESENTA:

MANUEL ALEJANDRO CÓRDOBA MALDONADO



CIUDAD UNIVERSITARIA

DIRECTOR DE TESIS:

DR. ESAÚ VICENTE VIVAS

MÉXICO, D.F. 2011



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicado a mis hermanos Jesús Javier León Maldonado y Luz Aurora León Maldonado que este documento les sirva de ejemplo, que los motive con deseos de superación. Fíjense metas altas en la vida que nada es imposible, todo lo que necesitan es entusiasmo y deseos de lograrlo, nunca duden de sus habilidades y no permitan que alguna persona las ponga en duda, tengan confianza en sí mismos y logran tener éxito en la vida.

Agradecimientos

A mi mamá Amparo por la ayuda brindada por tantos años, por sus cuidados y siempre preocuparse por mí, gracias por todo.

A mi padre Carlos Córdoba por su apoyo, comprensión y su forma de ser que me ha motivado a lo largo de la realización de mis estudios.

A mi madre Luz Maldonado por saberme escuchar, apoyarme y darme consejos en momentos difíciles en la vida.

A mi abuela Bartola Salgado por sus consejos, cuidados y sus oraciones.

A mi hermano y mis primos por sus consejos, cuidados y por todos los años de convivencia, sé que ustedes nunca me fallaran y me darán su apoyo en momentos difíciles.

A mis tías y tíos por su apoyo, consejos y cuidados durante estos años.

A todos y cada uno de mis amigos que han hecho más fácil y alegre mi vida, a aquellos que conocí en el 404 y en las canchas saben que nunca los olvidare ya que llegaron para quedarse, a mis amigos de computación sin ustedes los primeros semestres no hubieran sido los mismos, y también a mis amigos de telecom que conocí en quinto semestre y desde ese momento los vi hasta en la sopa deben saber que los aprecio mucho. Y también a mis demás amigos de otras carreras o que conocí en la prepa que siempre han estado ahí cuando necesité de su apoyo, la lista sería interminable, gracias por formar parte de mi familia.

A mis compañeros y amigos del Instituto de Ingeniería de la UNAM por brindarme su apoyo y consejos en la realización de este trabajo, en especial quiero agradecer a Emilio Jiménez y Mario Mendoza por sus consejos y también a mis demás compañeros Dierk Lüders, Eduardo Vizcaino, Ignacio Mendoza, Luis Zepeda, Mario Hernández, Rodrigo Córdoba.

A mi asesor de tesis el Dr. Esau Vicente Vivas por su ayuda y apoyo en la elaboración de esta tesis y por permitirme formar parte de un grupo de trabajo único.

A los profesores que no solo me educaron como ingeniero sino que también cambiaron mi forma de ver la vida y me motivaron a ser mejor persona cada día María del Rosario Cabeza Luna, Erik Castañeda, Jaime Martínez Martínez, Roberto Mandujano, Jesús Reyes, Margarita Bautista, Oleksandr Martyniuk, José María Matías Maruri, Francisco García Ugaldé, Víctor García Garduño, Salvador Landeros.

A mi alma mater, la UNAM que me brindó una oportunidad de formar parte de una de las mejores universidades del mundo y la mejor en Latinoamérica.

A la Facultad de Ingeniería que me dio una educación de alta calidad y no solo me formó como ingeniero si no también me hizo ser una mejor persona.

A Fundación Telmex por el apoyo brindado durante la realización de mis estudios.

A mis sinodales M.I. Juan Fernando Solórzano Palomares, Dr. Esau Vicente Vivas, Ing. Jesús Reyes García, M.I. Juventino Cuellar González y la Dra. Fátima Mountabi por tomarse la molestia de revisar esta tesis y apoyarme en los trámites para el examen profesional.

"¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil nos aporta tan poca felicidad? La respuesta es esta, simplemente: porque aún no hemos aprendido a usarla con tino."

Albert Einstein

*“Para tener éxito hay que tener amigos, pero
Para tener mucho éxito hay que tener enemigos”*

Frank Sinatra

ÍNDICE

Dedicatoria	iii
Agradecimientos	iv
ÍNDICE	vii
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xiii
INTRODUCCIÓN	xiv
Capítulo 1 El Satélite Educativo SATEDU	
1.1 Introducción.....	1
1.2 Subsistemas de SATEDU.....	4
1.2.1 Subsistema Estructural.....	6
1.2.2 Subsistema de Potencia.....	6
1.2.3 Subsistema de Estabilización.....	7
1.2.4 Subsistema de Sensores.....	8
1.2.5 Subsistema de Computadora de Vuelo.....	9
1.2.6 Subsistema de Comunicaciones Inalámbricas.....	9
1.2.7 Subsistema de Carga Útil.....	10
1.2.8 Estación Terrena.....	11
Capítulo 2 Características de las Primeras Tarjetas de Comunicaciones Inalámbricas de SATEDU	
2.1 Introducción.....	13
2.2 Arquitectura de la Tarjeta de RF.....	14
2.3 Software operativo de la Tarjeta de RF.....	18
2.4 Operación de la Tarjeta de Comunicaciones de RF.....	21
2.5 Ventajas y Desventajas de la Tarjeta RF.....	22

Capítulo 3 Características del Circuito Integrado de Interfaz Bluetooth

3.1 Introducción.....	24
3.2 Circuito de Interfaz Bluetooth.....	25
3.3 Arquitectura del Circuito Bluetooth.....	27
3.4 Detalles de Programación del Circuito Bluetooth.....	30

Capítulo 4 Diseño del Nuevo Subsistema de Comunicaciones Inalámbricas

4.1 Introducción.....	34
4.2 Arquitectura Propuesta Para la Tarjeta de Comunicaciones Inalámbricas Bluetooth.....	35
4.3 Diseño Esquemático de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.....	38
4.4 Diseño del Circuito Impreso Para la Tarjeta de Comunicaciones Inalámbricas Bluetooth.....	40
4.5 Fabricación del Circuito Impreso y Pruebas Eléctricas Preliminares.....	44
4.6 Ensamble de la Tarjeta con Componentes Electrónicos.....	47
4.7 Pruebas Electrónicas Preliminares Aplicadas a la Tarjeta de Comunicaciones Inalámbricas Bluetooth.....	48

Capítulo 5 Software de Operación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

5.1 Introducción.....	50
5.2 Plataforma de Desarrollo Para la Interfaz de Usuario.....	51
5.3 Necesidades de Programación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.....	52
5.4 Desarrollo de la Interfaz de Usuario.....	53
5.5 Utilización de la Interfaz de Usuario.....	55
5.6 Cambios de Programación Requeridos en el Software de Estación Terrena y en el Subsistema de Computadora de Vuelo para Incorporar el Uso de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.....	59

Capítulo 6 Pruebas de Validación Preliminares de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

6.1 Introducción.....	61
6.2 Conexión de la Tarjeta de Comunicaciones Inalámbricas Bluetooth a la Computadora.....	61
6.3 Pruebas de Validación de la Interfaz de Configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.....	63
6.4 Pruebas Comunicación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth con un Microcontrolador.....	66

Capítulo 7 Pruebas de Validación Finales de la Tarjeta de Comunicaciones Inalámbricas Bluetooth con SATEDU

7.1 Introducción.....	71
7.2 Incorporación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth al conjunto de Tarjetas de SATEDU.....	71
7.3 Pruebas de Validación Operativa Entre la Tarjeta de Comunicaciones Inalámbricas Bluetooth y la Tarjeta de Computadora de Vuelo.....	73
7.4 Pruebas de Validación Operando la Rueda Inercial de SATEDU.....	76

Capítulo 8 Conclusiones y Recomendaciones

8.1 Conclusiones.....	79
8.2 Recomendaciones.....	80

Apéndice A Código de las Interfaces de Usuario Desarrolladas en Microsoft Visual Studio 2010 81

Apéndice B Programas y Diagrama de Bloques de las Pruebas con Microcontroladores..... 90

ÍNDICE DE FIGURAS

Figura 1.1	SPUTNIK I	1
Figura 1.2	Morelos I	2
Figura 1.3	CubeSat Delfi-C3 construido por estudiantes de la Universidad Tecnológica de Delf en los Países Bajos.....	3
Figura 1.4	Diseño de SATEDU hecho en Solid Edge V18	4
Figura 1.5	Subsistema de Potencia	7
Figura 1.6	Subsistema de Estabilización	8
Figura 1.7	Subsistema de Sensores.....	8
Figura 1.8	Subsistema de Computadora de Vuelo	9
Figura 1.9	Subsistema de Comunicaciones Inalámbricas.....	10
Figura 1.10	Software de Estación Terrena que controla las funciones de SATEDU desde una computadora personal	11
Figura 2.1	Subsistema de Comunicaciones Inalámbricas	13
Figura 2.2	Diagrama de bloques de la tarjeta conectada a la PC	16
Figura 2.3	Imagen de la tarjeta de RF conectada a la PC (ambas caras)	16
Figura 2.4	Diagrama de bloques de la tarjeta de RF conectada a SATEDU	17
Figura 2.5	Imagen de la tarjeta de RF conectada a SATEDU (ambas caras)	17
Figura 2.6	Comunicación entre SATEDU y la Estación Terrena	18
Figura 2.7	Diagrama de flujo del programa en el PIC18F2321	19
Figura 2.8	Diagrama de flujo de la interrupción por puerto serie	20
Figura 3.1	Logo del protocolo de comunicaciones Bluetooth	24
Figura 3.2	Circuito RN-41	26
Figura 3.3	Diagrama de bloques del circuito RN-41	27
Figura 3.4	Diagrama de bloques del chip BlueCore-04 External	28
Figura 3.5	Patigrama del circuito RN-41	29

Figura 4.1	Diagrama de Bloques de la Tarjeta de Comunicaciones Inalámbricas Bluetooth	35
Figura 4.2	LED's Indicadores	36
Figura 4.3	Reset Manual	36
Figura 4.4	DIP Switch	37
Figura 4.4	Divisor de Voltaje	37
Figura 4.5	Diagrama esquemático de la Tarjeta de Comunicaciones Inalámbricas Bluetooth..	39
Figura 4.6	Matriz de reglas de conexión	40
Figura 4.7	Colocación de las huellas de los componentes	41
Figura 4.8	Huella de impresión del circuito RN-41	41
Figura 4.9	PCB de la Tarjeta de Comunicaciones Inalámbricas Bluetooth	42
Figura 4.10	Tarjeta de Comunicaciones Inalámbricas Bluetooth	43
Figura 4.11	Impresión del diseño del PCB	44
Figura 4.12	Placa Fenólica y dibujo del PCB	45
Figura 4.13	Placa Fenólica con las pistas marcadas	45
Figura 4.14	Circuito Impreso	46
Figura 4.15	Tarjeta de Comunicaciones Inalámbricas Bluetooth, vista de frente	47
Figura 4.15	Tarjeta de Comunicaciones Inalámbricas Bluetooth, vista posterior	48
Figura 5.1	Microsoft Visual Studio 2010	51
Figura 5.2	Herramientas para el desarrollo de la interfaz de software	52
Figura 5.3	Desarrollo de la interfaz	54
Figura 5.4	Interfaz de configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth	55
Figura 5.5	Software de Estación Terrena de SATEDU	59
Figura 5.6	Subsistema de Computadora de Vuelo SATEDU rojo.....	60
Figura 6.1	Regulador de 3.3 Volts LF33CV	62
Figura 6.2	Nombre Inicial de la Tarjeta de Comunicaciones Inalámbricas Bluetooth	62
Figura 6.3	Código PIN de la Tarjeta de Comunicaciones Inalámbricas Bluetooth	62

Figura 6.4	Dispositivo agregado correctamente	63
Figura 6.5	Puertos Asociados a la Tarjeta de Comunicaciones Inalámbricas Bluetooth	63
Figura 6.6	Configuración inicial del circuito RN-41	64
Figura 6.7	Modificación de parámetros por medio de la interfaz gráfica	65
Figura 6.8	Configuración final del circuito RN-41	65
Figura 6.9	Nuevo nombre y código PIN de la Tarjeta de Comunicaciones Inalámbricas Bluetooth	66
Figura 6.10	Monitor de Puertos	67
Figura 6.11	Primer prueba realizada en protobord	67
Figura 6.12	Interacción exitosa con un microcontrolador	68
Figura 6.13	Caracteres ASCII	68
Figura 6.14	Segunda prueba realizada en protobord	69
Figura 6.15	LCD y Monitor de Puertos con el mensaje “Hola Mundo”	69
Figura 7.1	Incorporación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth a SATEDU	72
Figura 7.2	Energización del Simulador Satelital SATEDU	73
Figura 7.3	Subsistemas básicos para utilizar el software de Estación Terrena	74
Figura 7.4	Configuración del Canal de Comunicaciones	74
Figura 7.5	Configuración del puerto en la Estación Terrena	75
Figura 7.6	Pestaña de “Computadora de Vuelo” en el Software de Estación Terrena	76
Figura 7.7	Pestaña de “Estabilización” en el Software de Estación Terrena	77
Figura B.1	Diagrama de Bloques de la Conexión Para la Prueba 1	90
Figura B.2	Diagrama de Bloques de la Conexión Para la Prueba 2	92

ÍNDICE DE TABLAS

Tabla 3.1	Consumo de energía del circuito Bluetooth.....	27
Tabla 3.2	Descripción de los pines del circuito RN-41	29
Tabla 3.3	Valores de fábrica del circuito RN-41	30
Tabla 3.4	Set Commands	31
Tabla 3.5	Get Commands	32
Tabla 3.6	Change Commands	32
Tabla 3.7	Action Commands	32
Tabla 3.8	Configuraciones por hardware	33
Tabla 4.1	Pines ocupados del circuito RN-41	39
Tabla 5.1	Comandos implementados por la Interfaz	53
Tabla 6.1	Prueba validación interfaz gráfica	64

INTRODUCCIÓN

En la actualidad la formación y capacitación de recursos humanos en el área de diseño y desarrollo de tecnología para sistemas espaciales es muy importante, tanto que existen diversos simuladores satelitales desarrollados en diversos países que permiten realizar esta tarea.

En el caso de México, existe el simulador satelital SATEDU. Este sistema es un simulador satelital para fines educativos y didácticos, de menos de 800 gramos y que ha sido desarrollado al interior del grupo de desarrollo de sistemas aeroespaciales en el Instituto de Ingeniería de la UNAM, por alumnos de diferentes áreas de la ingeniería tanto de licenciatura, maestría y doctorado bajo la supervisión del Dr. Esaú Vicente Vivas.

El objetivo de SATEDU es formar recursos humanos en el área satelital; SATEDU está integrado por diferentes subsistemas desarrollados en tarjetas electrónicas inteligentes lo que lo hace versátil, ya que se pueden realizar diversos experimentos desde pruebas muy sencillas como transferencia de datos hasta pruebas más elaboradas como la orientación y estabilización de un satélite por medios activos.

En este simulador satelital se requiere realizar una mejora sustancial en el subsistema de comunicaciones inalámbricas. SATEDU tiene la limitante que cada computadora con la que quiera enlazarse, necesita de una tarjeta de comunicaciones propietaria, la cual se desarrolló específicamente en el Instituto de Ingeniería de la UNAM para este propósito.

Por tal razón, en esta tesis se muestra el diseño, construcción y validación operativa de un nuevo subsistema de comunicaciones inalámbricas, que permitirá a SATEDU conectarse con cualquier computadora que cuente con bluetooth y el software de estación terrena. Esto redundará en beneficios de simplicidad y practicidad de conexión, así como reducción en el costo del satélite educativo SATEDU.

Capítulo 1

El Satélite Educativo SATEDU

1.1 Introducción

Al conocer el desarrollo de la ciencia y la tecnología que actualmente existe en el mundo las personas quedan perplejas y fascinadas al saber que todos esos avances tecnológicos empezaron como una idea en la imaginación de alguien, una persona que tenía curiosidad, que se sorprendía día a día al ver su entorno. La imaginación y la curiosidad humana son las herramientas que han logrado los avances en ciencia y tecnología que hoy tenemos.

En algún momento en la historia un hombre observó las estrellas y se empezó a preguntar cosas acerca de ellas, lo invadió una necesidad por dominar su entorno y empezó a desarrollar lo que hoy conocemos como tecnología espacial.

El 4 de octubre de 1957 en el año Geofísico Internacional la Unión Soviética logró poner en órbita el primer satélite artificial de la Tierra, el Sputnik I, figura 1.1, que era un satélite con forma de esfera que pesaba 84 Kg y tenía 60 centímetros de diámetro, estuvo durante 6 meses en su órbita pero solo 21 días operando.



Figura 1.1 SPUTNIK I.

A partir de este acontecimiento empezó una nueva era para la humanidad, una era de conquista espacial en la que los satélites fueron una herramienta indispensable para esa conquista, el campo de la tecnología espacial no solo ha traído avances a este campo sino que también ha traído inventos que se ocupan en diferentes ámbitos como el horno de microondas por ejemplo.

La incursión de México en el ámbito espacial no es algo nuevo, en 1962 se creó la Comisión Nacional del Espacio Exterior (CONEE) misma que se extinguió en 1977, pero dos años más tarde en 1979 México hizo la petición a la UIT de posiciones orbitales geoestacionarias para poner un satélite en órbita; mismas que se le conceden en 1982, sin olvidar la transmisión a color por televisión de los juegos olímpicos de 1968 por medio de la estación terrena Tulancingo-I.

El 17 de junio de 1985 México pone su primer satélite artificial en órbita, el Morelos-I, figura 1.2, por medio del transbordador espacial Discovery y unos meses más tarde el 27 de noviembre del mismo año se pone en órbita el segundo satélite artificial de México, el Morelos-II por medio del transbordador espacial Atlantis misión en la que viaja Nery Vela el primer Astronauta Mexicano.

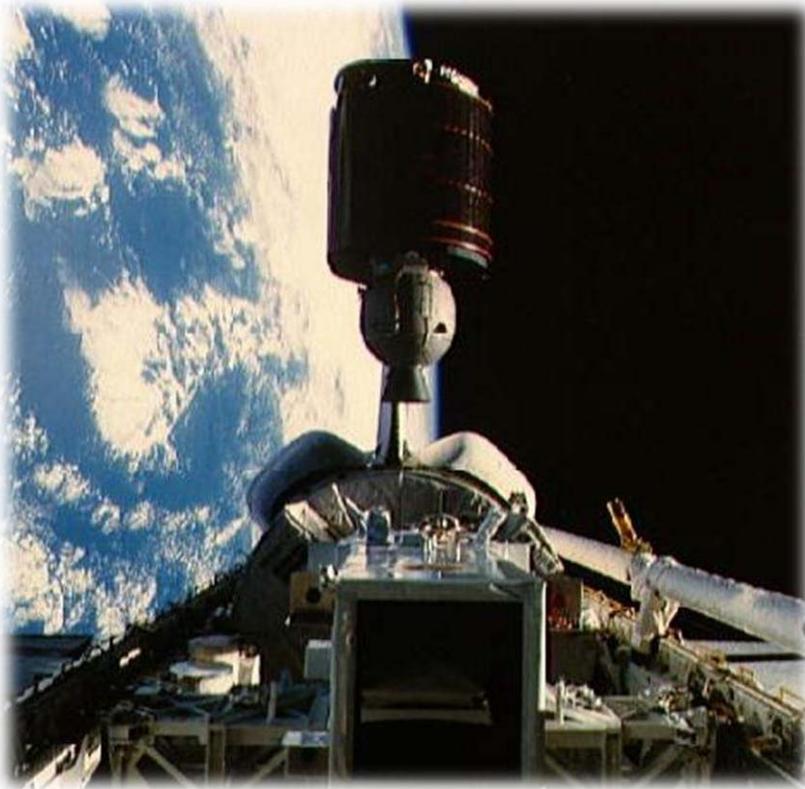


Figura 1.2 Morelos I.

En la actualidad se habla mucho de los pequeños satélites y de cómo los estudiantes participan en este tipo de experimentos y de esta manera obtienen mayores conocimientos al involucrarse en un proyecto de ingeniería como lo es la construcción de un satélite, debido a que dicha construcción requiere de un grupo de trabajo interdisciplinario que fomenta el trabajo en equipo.

Si todos los estudiantes interesados en el área satelital pudieran tener la oportunidad de involucrarse de un proyecto de ingeniería tan completo como lo es la construcción de un satélite esto les permitiría estar más preparados para enfrentarse al mundo laboral.

En un futuro lo ideal sería que estos satélites contruidos por estudiantes, figura 1.3, se vuelvan parte importante tanto de las telecomunicaciones como de la tecnología espacial, ya que se ha visto en diferentes misiones de este tipo de satélites que pueden tener una vida útil en el espacio mucho más grande de la esperada.

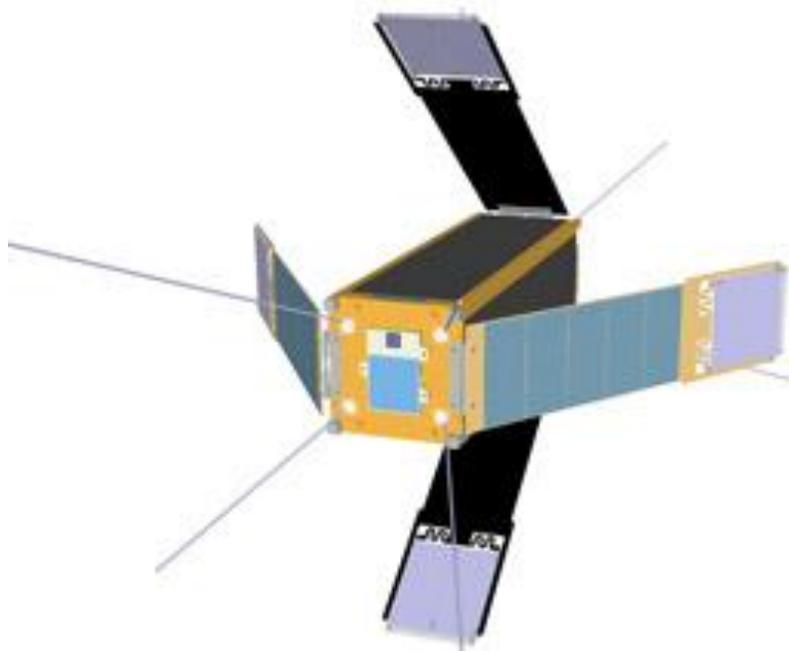


Figura 1.3 CubeSat Delfi-C3 construido por estudiantes de la Universidad Tecnológica de Delf en los Países Bajos.

El desarrollo de pequeños satélites podría ser una propuesta para reducir la brecha tecnológica que existe entre México y países más desarrollados pero para hacer esto posible se necesita capacitar recursos humanos, en este sentido SATEDU es un simulador satelital totalmente desarrollado y construido en el Instituto de Ingeniería de la UNAM por medio de estudiantes de diferentes áreas de la ingeniería y el objetivo de este simulador es el ser un satélite educativo para la formación de recursos humanos.

SATEDU es un simulador versátil que se puede utilizar en laboratorios de escuelas o en cursos de entrenamiento en el área satelital, tanto de manera general al dar demostraciones de funcionamiento para acercar a la juventud al área satelital y de esta manera despertar su interés por la ciencia y la tecnología. Pero también puede proporcionar entrenamiento especializado ya que este satélite educativo cuenta con todos los subsistemas que tiene un satélite real y cada uno de estos cuenta con un microcontrolador para realizar tareas específicas asignadas por medio de comandos que provienen de una computadora que cuenta con el software especializado desarrollado también en el Instituto de Ingeniería para fungir como Estación Terrena.

1.2 Subsistemas de SATEDU

Como se puede ver en la figura 1.4, SATEDU tiene una forma cilíndrica y dentro se encuentran sus subsistemas.

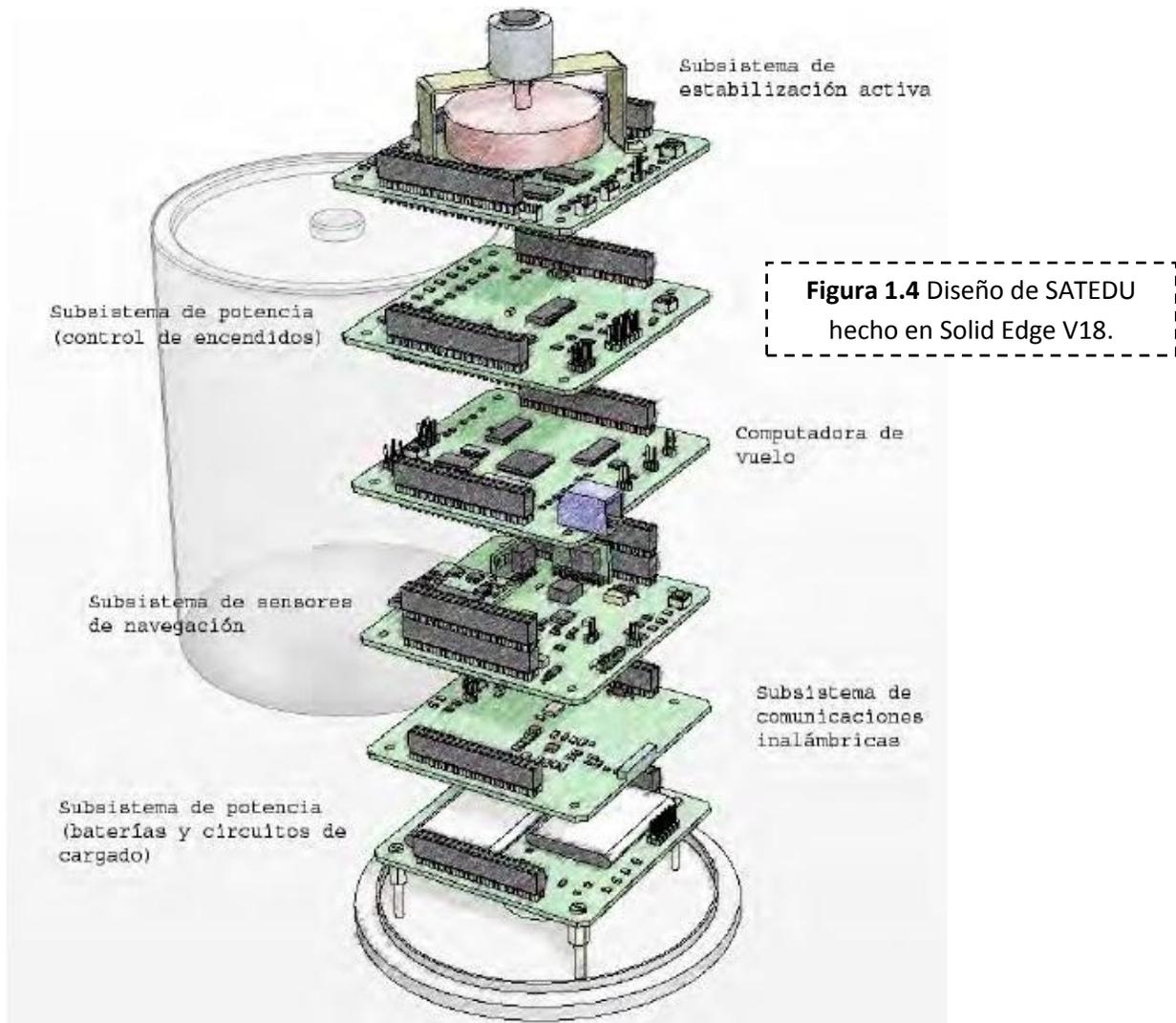


Figura 1.4 Diseño de SATEDU hecho en Solid Edge V18.

Un satélite real tiene los siguientes subsistemas:

- El subsistema estructural (lo tiene SATEDU)
- El subsistema de energía eléctrica o potencia (lo tiene SATEDU)
- El subsistema de control de posición y estabilización (lo tiene SATEDU)
- El subsistema de control térmico
- El subsistema de comando y telemetría (lo tiene SATEDU)
- El subsistema de computadora principal (lo tiene SATEDU)
- El subsistema de carga útil

Los primeros subsistemas constituyen el llamado BUS de un satélite es decir el satélite en sí y la CARGA ÚTIL representa la aplicación del satélite.

El único subsistema del BUS que no posee SATEDU es el de control térmico, el objetivo de este subsistema es el de mantener los demás subsistemas en un rango de temperatura en el cual puedan operar, y también protege parcialmente al satélite de la radiación cósmica por medio de películas reflejantes (control térmico pasivo) y por medio de resistencias y refrigerantes disipa calor (control térmico activo). SATEDU al ser un simulador satelital que se utiliza en laboratorios no necesita este tipo de protección. Esta es una diferencia muy marcada entre el simulador SATEDU y un satélite real ya que cada uno está hecho para trabajar en ambientes muy distintos.

El subsistema estructural de un satélite tiene la función de mantener cada uno de los subsistemas en una posición fija y de proveer una resistencia mecánica durante el lanzamiento, esta labor la hace el subsistema estructural de SATEDU.

El subsistema de energía eléctrica de un satélite real tiene la función de proveer de energía eléctrica de manera eficiente al satélite, para SATEDU este subsistema es el de potencia.

El subsistema de control de posición y estabilización de un satélite real tiene como tarea dar estabilidad a un satélite y poder mantenerlo en una posición específica, para SATEDU el encargado de esta labor es el subsistema de estabilización.

El subsistema de comando y telemetría de un satélite es el encargado de comandar al satélite y conocer el estado en el que se encuentra, este subsistema es el vínculo entre la estación terrena y el satélite y está integrado por un sistema de comunicaciones y un sistema de adquisición de datos de sensores de diversos tipos que ofrecen información operativa de los subsistemas que conforman el BUS satelital e importantes señales que permiten retroalimentar algoritmos que controlan la estabilización y apuntamiento de un satélite hacia puntos definidos del planeta, en SATEDU este subsistema es el de comunicaciones inalámbricas y el de sensores.

El subsistema de computadora principal de un satélite es el encargado de coordinar diversas acciones de los subsistemas que componen a un satélite, en SATEDU este subsistema es el de computadora de vuelo.

Cada uno de los subsistemas de SATEDU posee un microcontrolador lo que le permite realizar tareas en paralelo, pero el tipo de electrónica utilizado no es de clasificación militar como lo es en un satélite real comercial. Cabe señalar que existen actualmente muchos satélites experimentales pequeños que emplean electrónica comercial y han tenido excelentes resultados en vuelo orbital. También debe resaltarse que este tipo de casos emplea protecciones electrónicas contra efecto "lanch-up" para determinar cuando existen riesgos operativos en la electrónica debido al incremento de dosis de carga por radiación espacial. Este tipo de protecciones, también ya esta incluida en algunos de los subsistemas de SATEDU.

SATEDU tiene la forma de un cilindro pero cada uno de los subsistemas están contruidos sobre circuitos impresos cuadrados de 8.9 centímetros por lado.

En un principio se pretendía adoptar el tamaño de un cubesat (un cubo de 10 centímetros por lado) pero como SATEDU es solo un simulador se optó por una forma cilíndrica. La manera en que se interconectan cada uno de los subsistemas es a través de conectores de terminales largas o conectores de costilla uno sobre otro a manera de sándwich, esto fue un diseño heredado del proyecto SATEX. En estos conectores de costilla se realiza la comunicación entre los subsistemas por puerto serie.

1.2.1 Subsistema Estructural

Este subsistema desde el momento de su diseño de SATEDU se separó en dos partes, la primera un chasis que mantendría cada uno de los subsistemas de SATEDU en su lugar y la segunda que se uniría a la primer estructura para poder maniobrar el satélite. Inicialmente también se pensó en desarrollarle una estructura tipo giróscopo pero esta idea fue desechada debido a la dificultad de construir y balancear dicha estructura, con respecto a la primer estructura en un principio se deseaba hacer de acrílico pero la idea no se llevó a cabo debido a que este material requiere muchos cuidados en el momento de construir la estructura.

Actualmente para la primer estructura se optó por una estructura barata, fácil de conseguir y ya hecha como lo es un contener de varios discos compactos donde los subsistemas se fijan por medio de tornillos a la base del contenedor, este se cuelga de un hilo para poder realizar simulaciones en dos ejes.

Para realizar pruebas más serias de estabilización satelital en tres ejes se piensa utilizar otras plataformas donde será posible simular condiciones de cero fricción, por ejemplo una mesa suspendida en aire misma que ya tenemos construida y que actualmente instrumentamos en el Instituto de Ingeniería.

1.2.2 Subsistema de Potencia

Esté subsistema, figura 1.5, está compuesto por dos tarjetas, la tarjeta de baterías y la tarjeta de electrónica, la de baterías está compuesta por cuatro baterías Li-On, sus circuitos de carga y un botón de encendido aparte de componentes como resistencias, capacitores, led's y diodos. La alimentación de los circuitos de carga puede venir del tomacorriente o de celdas solares. Los led's son utilizados para mostrar de forma visual las condiciones del proceso de carga de las baterías.

La tarjeta de electrónica contiene un microcontrolador, sensores de temperatura, un convertidor DC-DC que convierte de 5 a 12 Volts y led's para mostrar de forma visual que subsistemas están energizados. El microcontrolador de este subsistema tiene la función de ejecutar los comandos que le envíe la computadora de vuelo para apagar o activar algún subsistema, este PIC controla los

encendidos por medio de interruptores, otra función que tiene es la de monitorear el voltaje de las baterías para conocer el estado en que se encuentran.



Figura 1.5 Subsistema de Potencia.

La tarjeta de electrónica es la conexión entre las baterías y el resto del simulador, esta recibe la energía de las baterías y con esta alimenta a todos los reguladores de voltaje de las demás tarjetas. La tarjeta puede proporcionar tres tipos de voltajes, 3.3 Volts, 5 Volts y 12 Volts dependiendo de las necesidades de los circuitos de cada tarjeta. La mayoría de los circuitos de SATEDU se alimentan con 5 Volts. El voltaje de 3.3 Volts es ocupado por los subsistemas de comunicaciones inalámbricas, sensores y para la memoria flash en la computadora de vuelo. El de 12 Volts es usado por el subsistema de estabilización.

1.2.3 Subsistema de Estabilización

Los componentes que integran a esta tarjeta son: un PIC18F4431, puentes H, circuitos protectores de corriente, una rueda inercial y un motor de DC, figura 1.6, además en el futuro se le agregarán bobinas de torque magnético.

El PIC se encarga de controlar el giro y velocidad del motor en un sentido o en otro para generar esquemas de estabilización en un eje, pero actualmente se están agregando otros dos motores para alcanzar esquemas de estabilización en tres ejes. El PIC cuenta con un módulo de retroalimentación de movimiento que le permite monitorear la velocidad del motor. Los puentes H se utilizan para proteger tanto al motor como a las bobinas de torque magnético mismas que se encuentran en desarrollo.



Figura 1.6 Subsistema de Estabilización.

1.2.4 Subsistema de Sensores

SATEDU cuenta con una tarjeta dedicada a sensores de navegación, que permiten monitorear sus movimientos y visualizar su posición en tiempo real. Esta tarjeta tiene un microcontrolador PIC18F2520 que cuenta con una memoria flash mayor para reducir el tiempo de adquisición de datos. Tiene también un acelerómetro triaxial, tres giróscopos electrónicos que son sensores de referencia inercial y una brújula electrónica como sensor de referencia, figura 1.7. Los giróscopos están colocados de manera ortogonal para monitorear los tres ejes de SATEDU.



Figura 1.7 Subsistema de Sensores.

Los datos de telemetría son adquiridos por el PIC y se transmiten a la estación terrena cuando esta los requiera para conocer la posición en la que se encuentra SATEDU. De esta forma resulta factible estabilizar o realizar el control de apuntamiento de SATEDU.

1.2.5 Subsistema de Computadora de Vuelo

Este subsistema tiene contacto con todos los demás subsistemas que componen a SATEDU debido a que la computadora de vuelo se encargada de asignar tareas a los demás subsistemas. La tarjeta cuenta con tres sensores de temperatura, un circuito tres estados, 2 multiplexores 8 a 1, una compuerta and, un flip flop, dos memorias RAM y dos microcontroladores, un SAB80C166 (SAB) de Siemens y un PIC16F876A de Microchip, figura 1.8.



Figura 1.8 Subsistema de Computadora de Vuelo.

El PIC es el intermediario entre la Estación Terrena (computadora personal con software especializado) y el SAB para cargar el programa de operación de SATEDU. El SAB almacena este programa en su memoria RAM de 256 Kb y tiene dos puertos series uno para comunicarse con el PIC (con la Estación Terrena) y otro para comunicarse con los demás subsistemas de SATEDU (de forma multiplexada). Los multiplexores sirven para seleccionar el subsistema al que se le enviará el comando.

1.2.6 Subsistema de Comunicaciones Inalámbricas

El subsistema de comunicaciones inalámbricas, figura 1.9, está compuesto por dos tarjetas una conectada a la computadora que hace las veces de Estación Terrena y otra instalada en SATEDU.

La arquitectura de ambas es muy parecida solo difieren en que una tiene un transceptor adicional para comunicarse por medio de USB a la computadora ya que las computadoras portátiles que hay actualmente en el mercado carecen de puertos seriales.

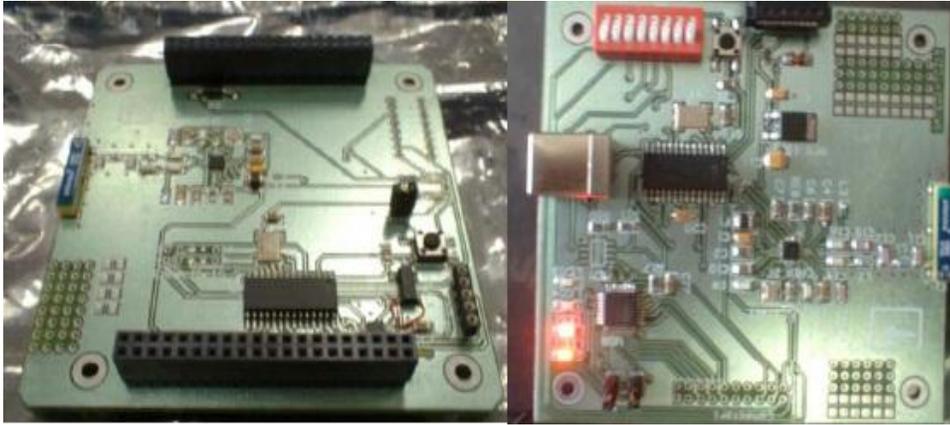


Figura 1.9 Subsistema de Comunicaciones Inalámbricas.

La función de estas dos tarjetas es enviar inalámbricamente los comandos generados en la estación terrena a SATEDU, más precisamente al subsistema de computadora de vuelo para que esta distribuya las órdenes a los diferentes subsistemas, y también para enviar datos de telemetría desde SATEDU a la estación terrena. Este subsistema será discutido dentro de esta tesis en el siguiente capítulo.

1.2.7 Subsistema de Carga Útil

En este momento SATEDU no cuenta con ninguna carga útil, sin embargo ya se encuentran varias de ellas en fase de desarrollo (telesalud, estabilización en tres ejes, cámara digital, etc) pero la ventaja de este simulador es que tiene una arquitectura versátil, ningún subsistema tiene un lugar fijo y aún quedan líneas de comunicación disponibles en el subsistema de computadora de vuelo (que ocupa para asignar tareas a cada subsistema).

Se tiene planeado crear un experimento donde se muestre principios de percepción remota, reconocimiento y seguimiento de patrones por medio de una cámara digital que pueda almacenar fotografías en la memoria flash de la computadora de vuelo. Además de procesar imágenes y de tomar decisiones de control de apuntamiento satelital de acuerdo con los resultados obtenidos, las imágenes se podrán mandar a la computadora que funge como Estación Terrena por medio del subsistema de comunicaciones Inalámbricas.

1.2.8 Estación Terrena

La estación terrena para SATEDU, figura 1.10, es cualquier computadora ya sea de escritorio o laptop en la cual se encuentre el software de estación terrena, este software es una interfaz de comunicación entre la computadora y SATEDU. Esta interfaz nos permite cargar el programa de funcionamiento del subsistema de computadora de vuelo para después mandarle comandos para los diferentes subsistemas de SATEDU, así como capturar telemetría satelital y visualizarla a distancia en tiempo real.

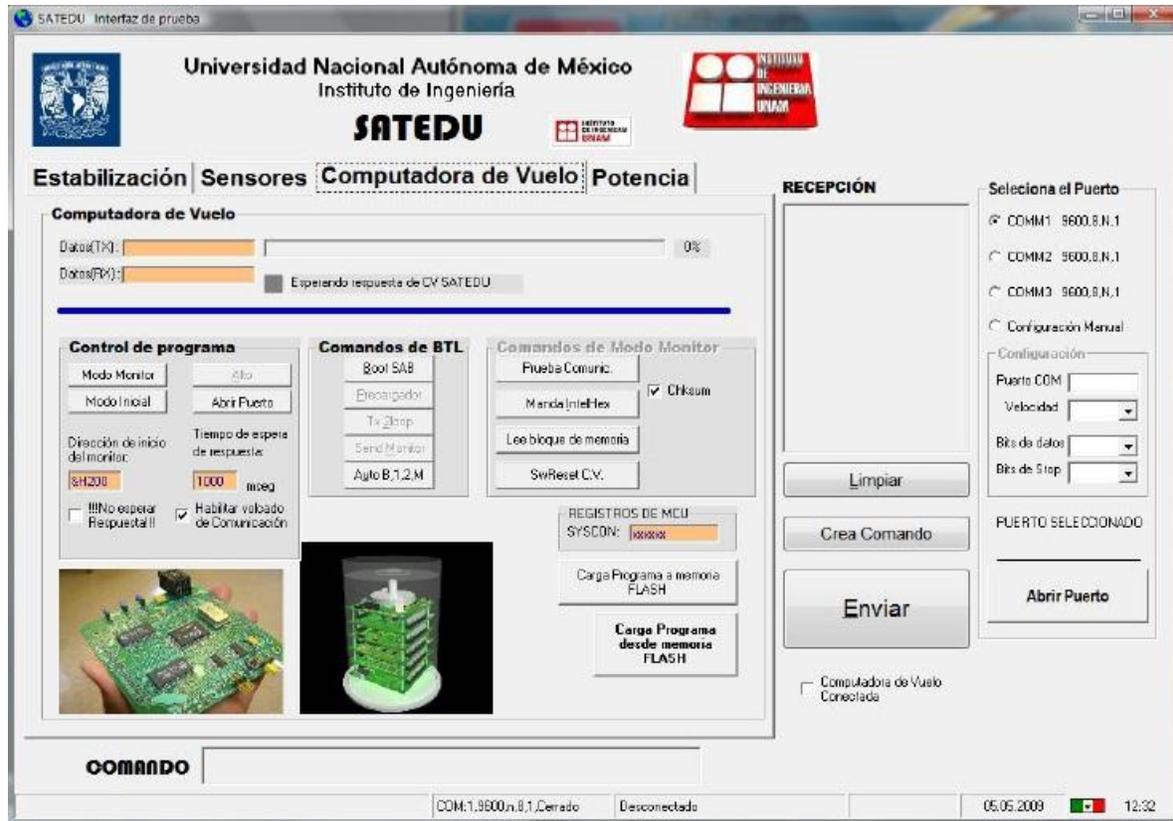


Figura 1.10 Software de Estación Terrena que controla las funciones de SATEDU desde una computadora personal.

Bibliografía:

- Apuntes de Radiocomunicaciones II, Elaborados por M. en I. José Luis García García. Semestre 2011-1.

- Jiménez Madrigal, Emilio Agosto, “Subsistemas de estabilización activa y sensores para un simulador satelital”, Tesis de licenciatura, Facultad de Ingeniería, UNAM. México D.F. 2009.
- Carrizales Muñoz, Zaira Lilian, “Subsistema inteligente de potencia para un satélite educativo”, Tesis licenciatura, Facultad de Ingeniería, UNAM. México D.F. 2009.

Capítulo 2

Características de las Primeras Tarjetas de Comunicaciones Inalámbricas de SATEDU

2.1 Introducción

Un satélite necesita comunicarse con su estación terrena para recibir comandos y para reportarle telemetría, como se vio en el primer capítulo esta tarea la realiza el Subsistema de Comando y Telemetría, el cual tiene la equivalencia en SATEDU con el Subsistema de Sensores para la telemetría y con el Subsistema de Comunicaciones Inalámbricas para la transmisión de datos.

Este subsistema es fundamental para que la operación de SATEDU se realice sin necesidad de cables externos, además de darle el atributo de portabilidad.

Este capítulo se dedica a mostrar la arquitectura, software y las funciones de una manera general para las dos tarjetas de comunicaciones de RF que constituyen al Subsistema de Comunicaciones Inalámbricas actual de SATEDU así como un análisis de sus ventajas, desventajas y las oportunidades que hay para mejorar el subsistema, figura 2.1.

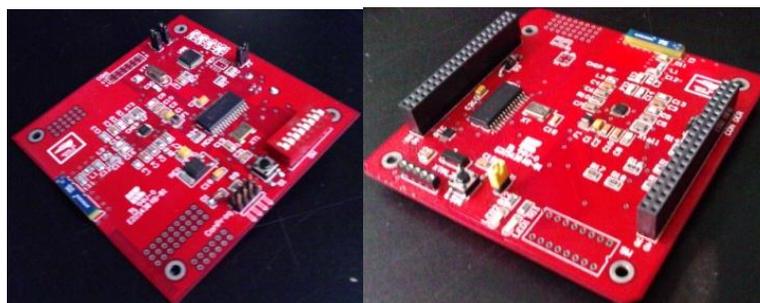


Figura 2.1 Subsistema de Comunicaciones Inalámbricas.

2.2 Arquitectura de la Tarjeta de RF

Como se dijo en el primer capítulo este subsistema se compone por dos tarjetas una conectada a una computadora personal y otra conectada a SATEDU, ambas tarjetas son muy similares excepto que la tarjeta conectada a la computadora tiene un transductor para conectarse con la computadora por medio de un puerto USB ya que la mayoría de equipos que están en el mercado hoy en día no cuentan con puertos seriales disponibles, otra diferencia es que la tarjeta conectada a SATEDU tiene conectores de costilla.

Las dos tarjetas de RF tienen básicamente la misma arquitectura, ambas tienen el microcontrolador PIC18F2321, el transceptor CC2500, una antena de RF de Antenova MICA 2.4 GHz, un oscilador de 10 MHz y un switch de selección de frecuencia. Las diferencias radican en que la tarjeta conectada a la PC tiene un convertidor de USB a Serial FT232BL, en tanto que la tarjeta conectada a SATEDU tiene un reloj en tiempo real Max6902.

El PIC18F2321 es de Microchip, es un microcontrolador de 8 bits, capaz de realizar hasta 10 millones instrucciones por segundo con un reloj de 40 MHz, es eficiente en compiladores en lenguaje C lo que facilita su programación; entre las características que tiene se pueden resaltar:

- Tiene un puerto de comunicaciones Serial síncrono y asíncrono.
- Trabaja en un rango de temperatura de -45°C a 85°C.
- WatchDogTimer con tiempo extendido.
- Múltiples opciones de Oscilador como osciladores de cristal, RC o un oscilador interno.
- Modulo MSSP (Master Synchronous Serial Port).
- Modulo HLVD (High Level Voltage Detection).
- Modulo CCP (Capture / Compare / PWM).
- 8 Kbytes de memoria flash.
- 512 bytes de memoria RAM.
- 256 bytes de memoria EEPROM.
- 25 entradas o salidas.
- Un temporizador de 8 bits y tres temporizadores de 16 bits.
- Convertidor analógico digital de 10 bits.

El transceptor CC2500 es un circuito integrado que trabaja en un rango de frecuencia de 2400 MHz a 2483.5 MHz que es una banda para aplicaciones industriales, científicas y médicas. Está diseñado para aplicaciones de bajo consumo de potencia y en distancias cortas, entre las características que tiene se pueden resaltar:

- Soporta modulación en amplitud, frecuencia y fase.
- Puede alcanzar tasas de transmisión de hasta 500 mil símbolos por segundo.
- Se controla por medio del BUS SPI.
- Modo de espera para un bajo consumo de energía.

Este transceptor puede confirmar si el canal a utilizar se encuentra libre y también tiene un indicador de la calidad del enlace establecido.

La antena de RF de Antenova MICA 2.4 GHz está diseñada para trabajar en un rango de frecuencia entre 2.4 GHz y 2.5 GHz, entre sus características se puede resaltar que:

- Utiliza polarización lineal.
- Impedancia de acoplamiento de 50 ohms.
- Opera entre un rango de temperatura que va de -40°C a 85°C.
- Pesa 0.5 gramos.
- Tiene forma de un paralelepípedo de 20.5 x 3.6 x 3.3 milímetros.
- Ganancia promedio de 1.9 dBi.
- Eficiencia promedio de 65%.

El oscilador de 10 MHz es un oscilador de cristal que utiliza el microcontrolador. El switch de selección de frecuencia está conectado al microcontrolador para hacer de manera manual el cambio de canal de transmisión con la posibilidad de seleccionar entre 255 canales diferentes.

El convertidor de USB a Serial FT232BL se utiliza debido a que las computadoras portátiles que se venden hoy en día no tienen puertos seriales. Este convertidor, crea un puerto serie virtual en la computadora, algunas de sus características son:

- Configuración del puerto serie síncrono y asíncrono.
- Buffer de recepción de 384 bytes.
- Buffer de transmisión de 128 bytes.
- Interfaz para niveles lógicos de 5 Volts y 3.3 Volts.
- Compatible con USB 1.1 y USB 2.0.
- Tiene un oscilador de 6 MHz.

El reloj en tiempo real Max6902 puede proveer año, mes, día del mes, día de la semana, hora, minutos y segundos. Se conecta con el PIC por medio de un puerto SPI. Se integró a SATEDU para que se le puedan programar tareas de manera preestablecida.

En la figura 2.2 se muestra la forma en que se conectan los componentes de la tarjeta de RF de la PC:

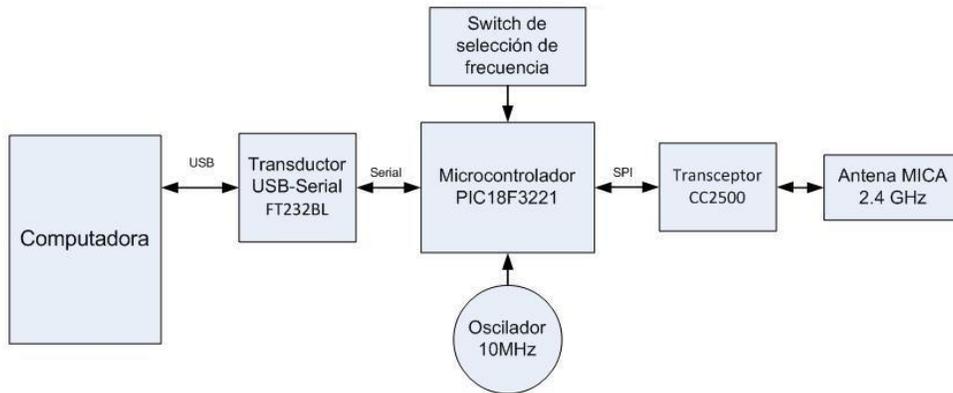


Figura 2.2 Diagrama de bloques de la tarjeta conectada a la PC.

La tarjeta que se conecta a la computadora, figura 2.3, obtiene su energía por medio del puerto USB, el microcontrolador se alimenta con 5 Volts, en tanto que el regulador de 3.3 Volts se usa para alimentar el transceptor CC2500.

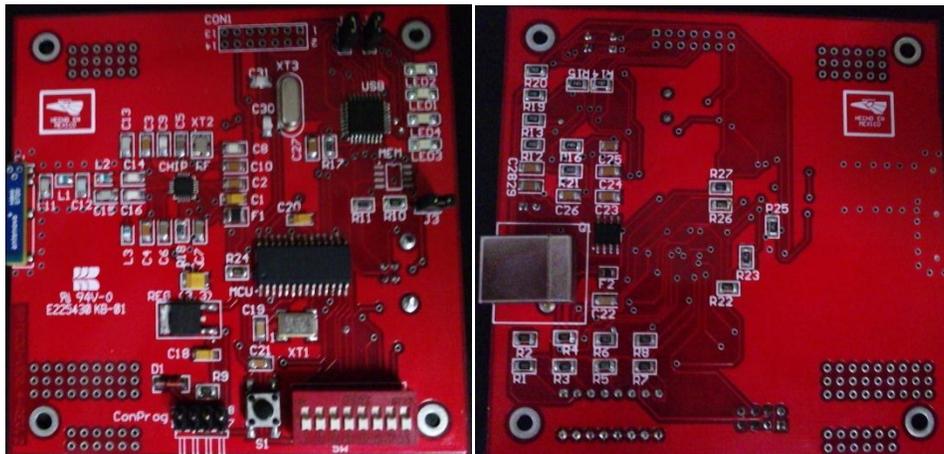


Figura 2.3 Imagen de la tarjeta de RF conectada a la PC (ambas caras).

En la figura 2.4 se muestra la forma en la que se conectan los componentes de la tarjeta de RF conectada a SATEDU:

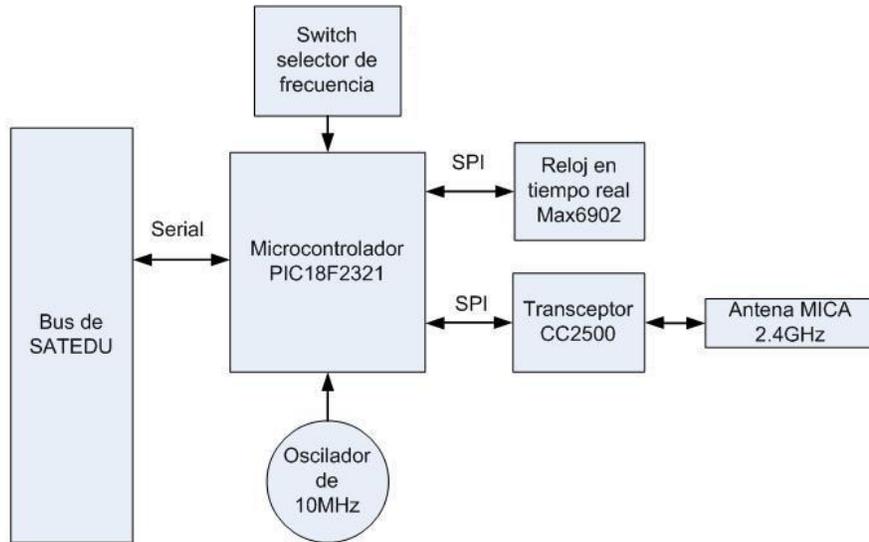


Figura 2.4 Diagrama de bloques de la tarjeta de RF conectada a SATEDU.

La tarjeta conectada a SATEDU, figura 2.5, obtiene su energía del voltaje suministrado al bus de SATEDU por el subsistema de potencia. El reloj en tiempo real y el transceptor CC2500 necesitan una alimentación de 3.3 Volts y el microcontrolador necesita 5 Volts.

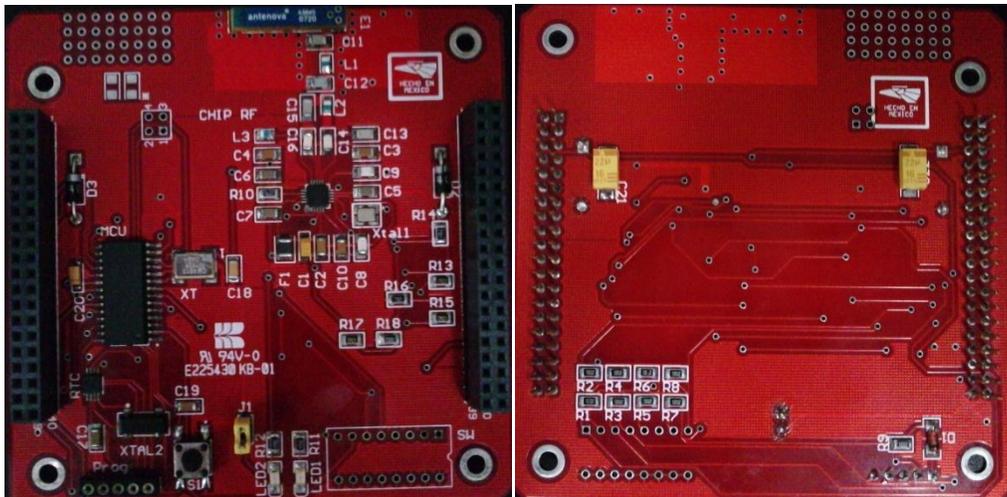


Figura 2.5 Imagen de la tarjeta de RF conectada a SATEDU (ambas caras).

Adicionalmente a los circuitos mostrados en los diagramas de bloques, cada tarjeta necesita de otros componentes pasivos como resistencias y capacitores, también las tarjetas tienen diodos para proteger los circuitos mostrados en las figuras anteriores.

2.3 Software operativo de la Tarjeta de RF

Todos los comandos que se envían a SATEDU son generados por el software de la estación terrena. En este caso, el Subsistema de Comunicaciones Inalámbricas es el encargado de llevar tal información desde la computadora hasta SATEDU, figura 2.6.

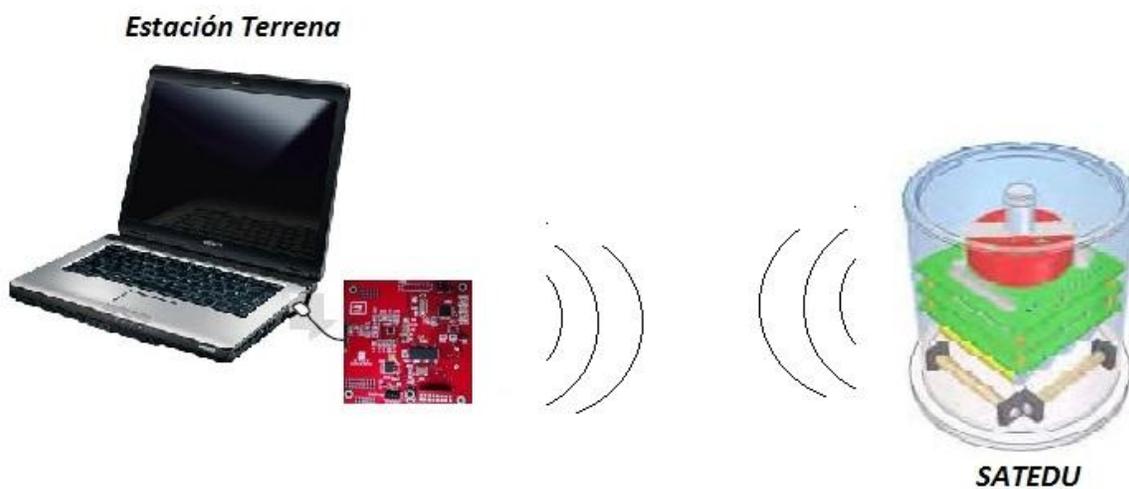


Figura 2.6 Comunicación entre SATEDU y la Estación Terrena.

Para el buen funcionamiento del transceptor CC2500 es necesario el microcontrolador y el software en el microcontrolador el cual es el mismo tanto para la tarjeta conectada a SATEDU como para la tarjeta conectada en la computadora, ya que aun no se implementa la utilización del reloj en tiempo real.

El software se desarrollo en lenguaje C y para la compilación se utilizó el compilador C18 de Microchip en MPLAB, la configuración del transceptor CC2500 es muy larga y tediosa ya que se tienen que configurar muchos registros para esta tarea se utilizó el programa smartRF studio que configura los registros de los dispositivos del tipo CCXXXX, el programa entrega un archivo *.c con las configuraciones de los registros.

En la figura 2.7 se puede ver el diagrama de flujo del programa instalado en el microcontrolador:

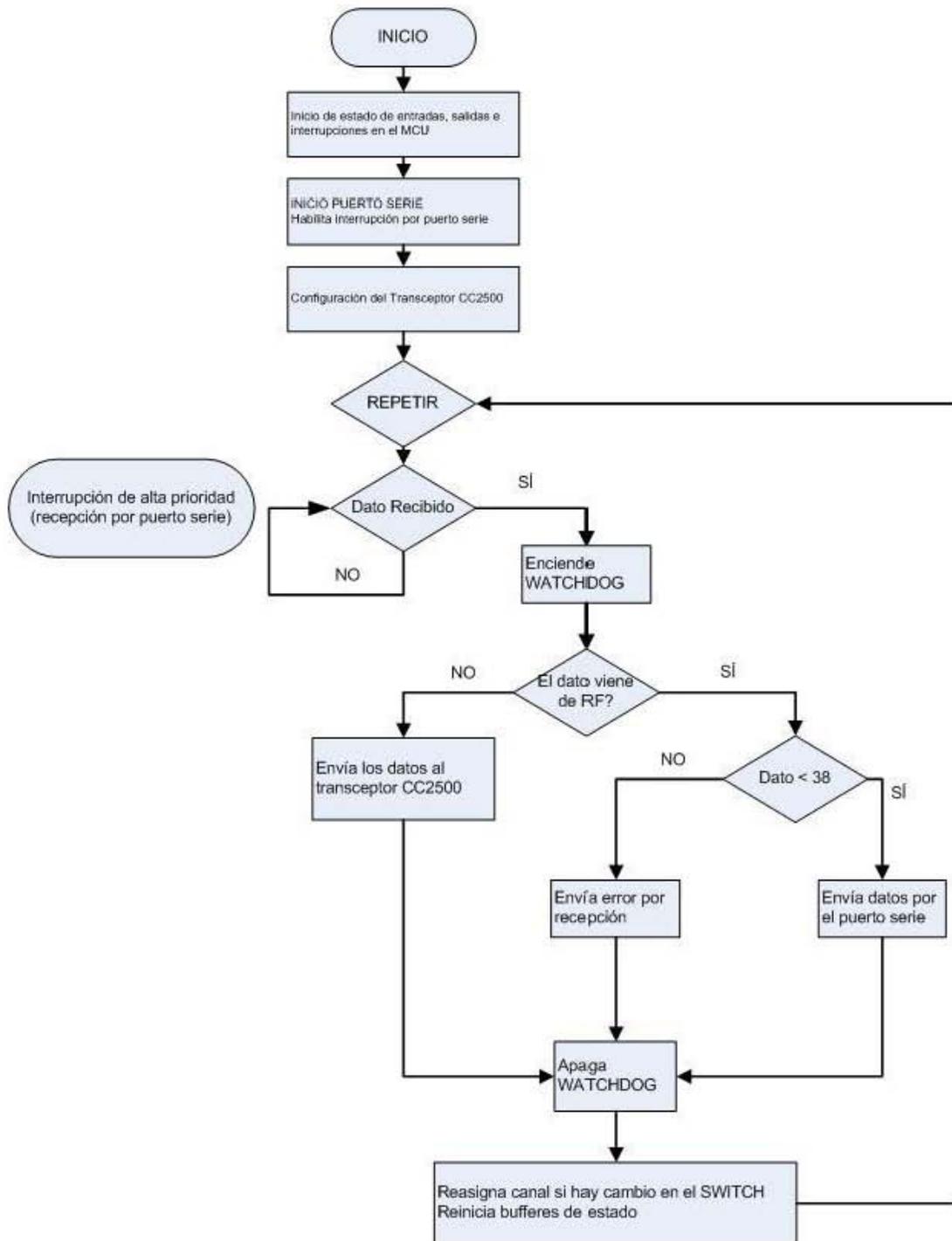


Figura 2.7 Diagrama de flujo del programa en el PIC18F2321.

Una vez que empieza la ejecución del programa, éste espera a recibir algún dato. Si la recepción se hace por medio del puerto serie entra la interrupción. El diagrama de flujo de la interrupción se muestra en la figura 2.8:

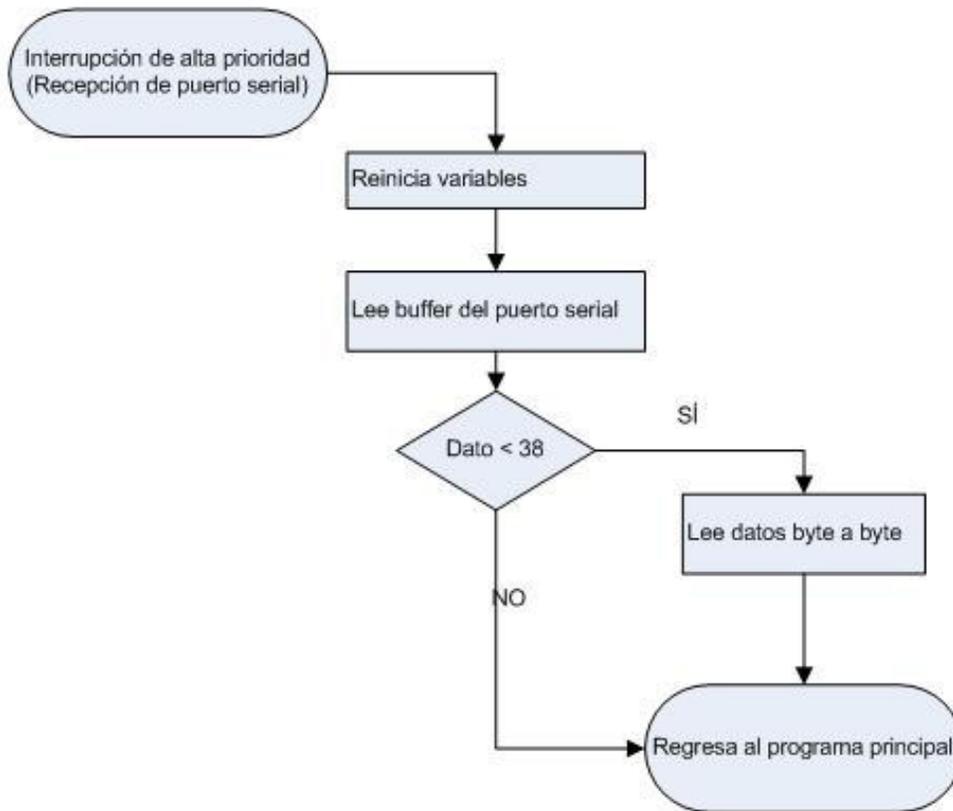


Figura 2.8 Diagrama de flujo de la interrupción por puerto serie.

Una vez que recibe algún dato se enciende el Wach Dog timer, este temporizador es un elemento muy importante en cualquier microcontrolador ya que en el momento que se desborda realiza una acción correctiva dado que se presentó una falla en la ejecución del programa, en este programa la acción correctiva es la de reiniciar el microcontrolador.

Otro aspecto que vale la pena comentar es que en ambos diagramas se indica que los datos tienen que ser menores a 38 bytes, esto es debido a que la trama que utiliza el transceptor CC2500 tiene una longitud máxima debido a su buffer FIFO (first in first out), este buffer es de 64 bytes pero no todos son para datos ya que también incluye unos bytes de preámbulo, palabra de sincronía, longitud de datos, y comprobación de redundancia cíclica.

2.4 Operación de la Tarjeta de Comunicaciones de RF

La función que tienen estas dos tarjetas es la de enviar los comandos que se generan en la computadora que simula la estación terrena a la computadora de vuelo de SATEDU, además de enviar la telemetría generada en SATEDU a la estación terrena.

Para enviar un comando primero se genera éste en la interfaz de estación terrena y después este se manda por un puerto COM a la tarjeta de RF conectada a la PC. Esta a su vez envía la información por medio del PIC y del chip de RF a la tarjeta de RF en SATEDU, esta tarjeta recibe la información y la envía a la computadora de vuelo para que esta distribuya la información entre los subsistemas.

En el caso de envío de telemetría se emplea el mismo proceso solo que en sentido inverso. Cabe señalar que por sus componentes la distancia máxima que puede haber entre SATEDU y la computadora es de 10 metros.

Para operar las tarjetas lo primero que se debe de hacer es conectar la tarjeta de RF de la computadora por medio del cable USB y conectar la tarjeta de RF de SATEDU. Una vez que se genera un comando con el software de estación terrena la información pasa por diferentes interfaces de comunicaciones:

- Primero los datos que salen del puerto USB se pasan a un formato serial asíncrono para que llegue el PIC.
- Una vez en el PIC los datos pasan del formato serial asíncrono a un formato SPI síncrono para interactuar con el transceptor CC2500.
- El transceptor CC2500 convierte estos datos a una forma de onda a una frecuencia de 2.4 GHz, con una modulación FSK.
- La información es recibida por el transceptor CC2500 en SATEDU y la forma de onda pasa de un formato SPI síncrono para ser entregada al PIC.
- El PIC convierte el formato SPI síncrono a formato serial asíncrono y entrega los datos a la computadora de vuelo.

Para el envío de telemetría se hace el mismo proceso pero en sentido inverso.

Con estas dos tarjetas se validaron las operaciones inalámbricas de:

- La transmisión de comandos entre la estación terrena (computadora) y el satélite (SATEDU).
- Recepción de telemetría básica en la estación terrena.
- Control de la rueda inercial de SATEDU.
- Visualización virtual de la posición de SATEDU en 3D en tiempo real de acuerdo a la orientación indicada por los sensores de navegación inercial.

2.5 Ventajas y Desventajas de la Tarjeta RF

Las ventajas que presenta el actual Subsistema de Comunicaciones Inalámbrico de SATEDU son las siguientes:

- No necesita que la computadora tenga un puerto serie disponible ya que se puede conectar por medio de un puerto USB.
- No cualquier dispositivo se puede comunicar con SATEDU por lo que es un sistema seguro ya que cada computadora que desee conectarse con él necesita una tarjeta de RF propietaria.
- Para transmitir tiene 255 canales disponibles lo que permitiría en algún momento tener varios satélites educativos funcionando en el mismo laboratorio sin interferencia entre ellos ya que cada uno trabajaría en una frecuencia diferente.
- El microcontrolador hace a este sistema versátil y flexible.

Una desventaja que tiene es que aunque cuenta con 255 canales disponibles es un poco laboriosa la configuración del sistema ya que se tienen que configurar muchos registros del transceptor CC2500 en software antes de poder ocupar cualquier canal de transmisión. Otra desventaja es que no se puede conectar con dispositivos que no cuenten con un puerto USB.

La función de las tarjetas es dar portabilidad a SATEDU esto se cumplió perfectamente debido a que con la tarjeta de RF de la computadora se puede conectar a cualquier laptop o PC, pero hoy en día existen dispositivos más pequeños que tienen capacidades de procesamiento similares a las de una computadora como los llamados teléfonos inteligentes y PDA's. SATEDU no puede conectarse con este tipo de dispositivos.

La oportunidad de mejora es actualizar el Subsistema de Comunicaciones Inalámbricas para que SATEDU se pueda conectar con dispositivos como teléfonos inteligentes y PDA's, sin perder la conectividad con computadoras.

En este sentido la propuesta que se hace y desarrolla en esta tesis, es la utilización del estándar Bluetooth para la comunicación entre la computadora y SATEDU debido a que esto daría la posibilidad en un futuro de controlar a SATEDU por medio de cualquier dispositivo que posea Bluetooth, esto daría una mayor conectividad al simulador satelital, que abre la puerta a simulaciones de otro tipo y desarrollo de nuevos experimentos como localización satelital por medio de dispositivos móviles o telesalud.

Bibliografía:

- Dr. Esaú Vicente Vivas, "Sistema portátil para entrenamiento de recursos humanos en tecnología de satélites pequeños primera fase". Proyecto 52297 de CONACYT, Instituto de Ingeniería, UNAM. México D.F. Mayo de 2008.

Capítulo 3

Características del Circuito Integrado de Interfaz Bluetooth

3.1 Introducción

Bluetooth es un protocolo de comunicaciones bastante aceptado para redes WPAN (Wireless Personal Area Network, red inalámbrica de área personal), trabaja en la banda de 2.4GHz, tiene un bajo consumo de energía y su mayor utilidad es en la transferencia de datos. Los dispositivos bluetooth, figura 3.1, son muy utilizados en computadoras de escritorio, laptops, PDA's y teléfonos celulares entre otros dispositivos.



Figura 3.1 Logo del protocolo de comunicaciones Bluetooth.

Al integrar bluetooth a SATEDU ampliamos el número de dispositivos a los cuales se puede conectar. Por ello, además de interactuar con computadorastambién podrá interactuar con PDA's y teléfonos celulares. Una ventaja de este trabajo es que el Subsistema de Comunicaciones Inalámbricas pasará de usar dos tarjetas a usar solo una, lo cual contribuye también a reducir el costo del satélite educativo.

Existen muchos dispositivos bluetooth en el mercado pero para las necesidades de SATEDU se requieren las siguientes características en el circuito:

- Pequeño, debido a que las tarjetas que se conectan en SATEDU tienen una dimensión de 8.9 x 8.9 centímetros.
- Económico, siempre es un factor determinante en cualquier proyecto.
- Bajo consumo de energía, debido a que SATEDU funciona con baterías.
- Montaje superficial.
- Fácil de soldar.
- Compatible con el puerto serie, puesto que el BUS principal de SATEDU utiliza este protocolo.
- Que tenga un alcance de por lo menos 10 metros; una forma de clasificar los circuitos bluetooth es por su alcance, potencia de transmisión y sensibilidad de recepción. Existen clase 1, clase 2 y clase 3, el dispositivo elegido tiene que ser clase 1 o clase 2 ya que la clase 3 solo tiene un rango aproximado de operación de un par de metros.

En la búsqueda realizada en esta tesis, se encontraron muchos circuitos bluetooth, pero algunos no cumplían con todas las características buscadas, principalmente no eran fáciles de soldar y la compatibilidad con el puerto serie era limitada, ya que solo era compatible con ciertas velocidades de transmisión.

En este capítulo se describe el circuito elegido para la nueva tarjeta del Subsistema de Comunicaciones Inalámbricas.

3.2 Circuito de Interfaz Bluetooth

El circuito elegido para la interfaz bluetooth es el circuito RN-41 de la compañía Roving Networks, figura 3.2, este es un modem bluetooth de bajo consumo, económico y que puede añadir capacidad inalámbrica por interfaz bluetooth a SATEDU. El modem puede tener velocidades de transmisión hasta de 3 Mbps en distancias menores a 100 metros según datos del fabricante y entre sus aplicaciones principales tiene la de sustituir cables de tipo serial, que es exactamente lo que se necesita en este proyecto.

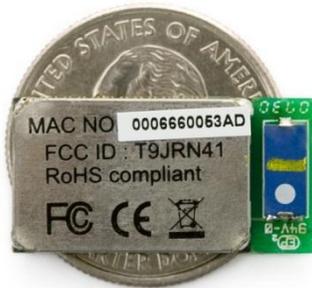


Figura 3.2 Circuito RN-41.

Las características del circuito bluetooth son las siguientes:

- Es compatible con las versiones 1.1, 1.2, 2.0 y 2.1 de Bluetooth.
- También es compatible con la versión 2.0+EDR lo que permite tener velocidades de hasta 3Mbps.
- Empaquetado pequeño de 13.4mm x 25.8mm x 2mm.
- Bajo consumo, con un máximo de 30 mA.
- Interfaz de conexión de datos con UART (SPP y HCI) y USB (HCI).
- Calificado Bluetooth SIG.
- Fácil montaje en PCB.
- Dispositivo Clase 1:
 - Hasta 100 metros de distancia.
 - 12 dBm en transmisión.
 - Sensibilidad de -80 dBm.
- Certificado FCC, ICS y CE
- Tasas de transmisión desde 1200 bps hasta 921 Kbps.
- Frecuencia de operación entre 2402 a 2480 MHz.
- Modulación FHSS/GFSK, 79 canales en intervalos de 1 MHz.
- Encriptación de 128 bits.
- Corrección de errores para garantizar la entrega de paquetes.
- Puede crear Piconets y Scatternet.

Este modem trabaja entre -40°C y 85°C, se alimenta con un voltaje de 3.3 Volts y su consumo de potencia dependiendo del estado en el que se encuentra se ve en la tabla 3.1:

Estado	Consumo promedio [mA]
Stanby/Idle (opción por default)	25
Conectado (modo normal)	30
Conectado (bajo poder)	8
Standby/Idle (dormido profundo)	2.5

Tabla 3.1 Consumo de energía del circuito Bluetooth.

3.3 Arquitectura del Circuito Bluetooth

A simple vista se puede ver un encapsulado y una antena en el circuito bluetooth. En el datasheet del circuito se encuentra el siguiente diagrama de bloques, figura 3.3:

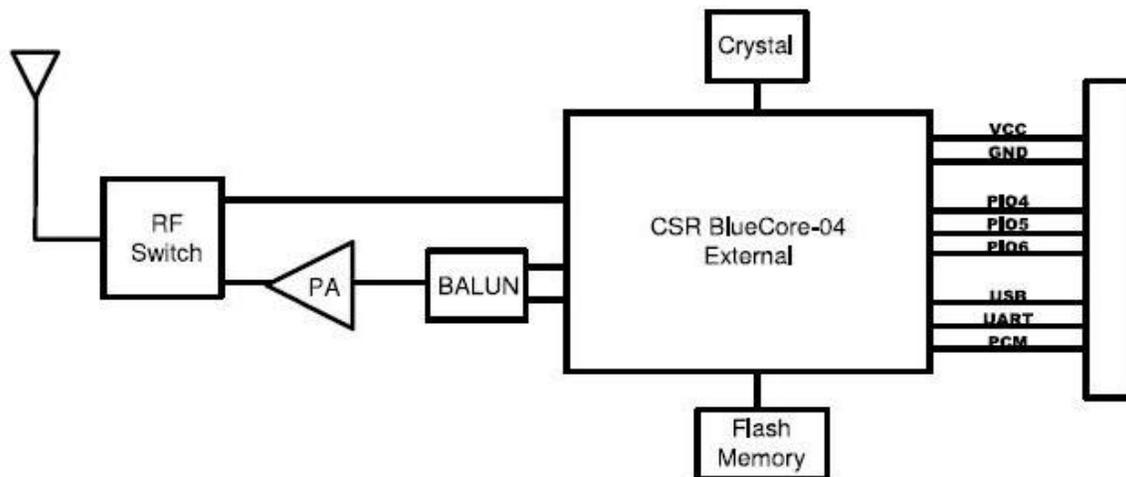


Figura 3.3 Diagrama de bloques del circuito RN-41.

CSR es una compañía que se dedica a diseñar y fabricar dispositivos de radio en un solo chip para el estándar bluetooth. En la figura 3.4, se muestra el diagrama bloques del BlueCore-04 External, este chip tiene integrados un microcontrolador, una memoria RAM, un DSP, interfaz de entrada y salida de datos con diferentes protocolos, una interfaz de Radio a 2.4 GHz y una memoria flash externa.

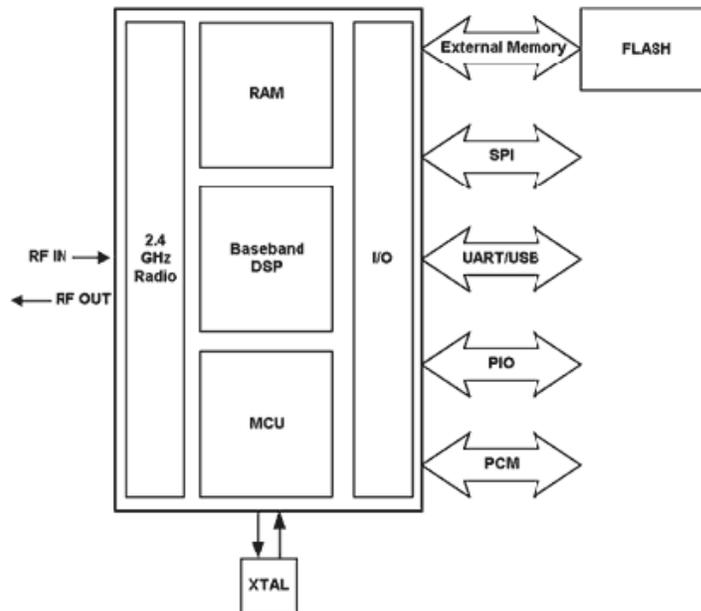


Figura 3.4 Diagrama de bloques del chip BlueCore-04 External.

La memoria RAM de 48 Kbytes permite una máxima transferencia de datos y la interfaz UART puede tener una velocidad de transmisión de 1.5 Mbaud; otras de sus características son las siguientes:

- Memoria flash externa de 8 Mbits.
- Bluetooth v2.0 +EDR (se puede actualizar a versión 2.1).
- Soporta Piconet y Scatternet.
- Puede estar en el mismo ambiente que el estándar 802.11.
- Con EDR Bluetooth se pueden transmitir imágenes de 1 Mb en 4 segundos.

Este chip es utilizado para controlar el envío y la recepción de datos, el microcontrolador se puede reprogramar por medio del puerto SPI y se utiliza un entorno de desarrollo de CRS.

El balun adapta la impedancia del transmisor con la impedancia de la antena. El amplificador permite un mayor alcance debido a que es un Bluetooth clase 1 y el switch de RF permite seleccionar entre transmisión y recepción.

El patigrama y función de cada uno de los pines se muestra tanto en la figura 3.5 como en la tabla 3.2:

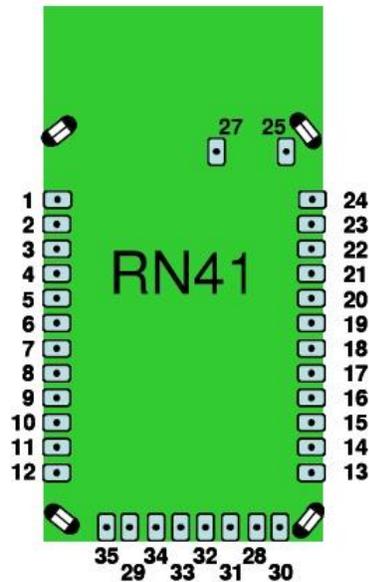


Figura 3.5 Patigrama del circuito RN-41.

PIN	Nombre	Descripción
1	GND	Tierra
2	SPI MOSI	Solo para programación
3	PIO6	Selección maestro/esclavo
4	PIO7	Selección de tasa de transmisión
5	RESET	Reinicia
6	SPI_CLK	Solo programación
7	PCM_CLK	Interfaz PCM
8	PCM_SYNC	Interfaz PCM
9	PCM_IN	Interfaz PCM
10	PCM_OUT	Interfaz PCM
11	VDD	Alimentación de 3.3 Volts
12	GND	Tierra
13	UART-RX	Recepción de datos seriales
14	UART-TX	Transmisión de datos seriales
15	UART_RTS	Pasa a alto para desactivar el transmisor del anfitrión
16	UART_CTS	Si está en alto deshabilita el transmisor
17	USB_D+	Puerto USB

18	USB_D-	Puerto USB
19	PIO2	Alto cuando se conecta, bajo en otro caso
20	PIO3	Descubrimiento automático
21	PIO5	Cambia en función del estado, bajo al conectar
22	PIO4	Valores de fabricación
23	SPI_CSB	Solo programación
24	SPI_MISO	Solo programación
25	GND	Tierra
26	NC	Opción de RF
27-29	GND	Tierra
30	AIO0	Entrada análoga opcional
31	PIO8	Estado de RF (transmitiendo o recibiendo)
32	PIO9	Entrada/Salida
33	PIO10	Entrada/Salida
34	PIO11	Entrada/Salida
35	AIO1	Entrada análoga opcional

Tabla 3.2 Descripción de los pines del circuito RN-41.

3.4 Detalles de Programación del Circuito Bluetooth

Para la programación del circuito RN-41 se necesita una computadora con bluetooth (integrado o externo) o la conexión del circuito a un puerto serie que tenga la misma configuración que tiene por default. Una vez programada la configuración no cambiará (aunque se desconecte el circuito) hasta que la configuración sea cambiada o se restauren los valores de fábrica, por default la configuración del puerto serie del circuito es el que indica la tabla 3.3:

Opción	Valor de fábrica
Perfil de Servicio Bluetooth	SPP (Serial Port Profile)
Modo	Esclavo
Tasa de Transmisión	115200 bps
Paridad	Ninguna
Bits de Datos	8bits
Bit de Parada	1 bit
Modo de Consumo	Modo de descubrimiento de bajo consumo
Nombre del Dispositivo	FireFly-xxxx
Clase de servicio	SPP
Autenticación	Deshabilitado
Encriptado	Deshabilitado
Descubrimiento	Se muestra por intervalos de 2.56 segundos
Tiempo para entrar a la configuración	60 segundos
Código	1234
Eco	Apagado

Tabla 3.3 Valores de fábrica del circuito RN-41.

Lo primero que se tiene que hacer para programar el circuito es hacer el enlace entre el modulo bluetooth y la computadora, el nombre del circuito es "FireFlyXXXX", la mayoría de los dispositivos que utilizan bluetooth requieren autenticación, el código por default del circuito es "1234".

Una vez realizado el enlace se le asignará al circuito un puerto COM y entrará en el modo de datos para envío y recepción de información, para la programación del circuito se necesita entrar al modo comando.

Para entrar al modo comando se tienen que enviar tres signos de peso "\$\$\$" desde la computadora a través del puerto serie asignado en los primeros 60 segundos después de encender el modulo, esté responderá con "CMD", para salir del modo comando se deben enviar tres signos de menos "---" y el modulo responderá con "END".

Cada uno de los caracteres enviados debe de estar en código ASCII. Los comandos validos obtienen "AOK" de respuesta, comandos no validos obtienen "ERR" y comandos que no se reconocen obtienen "?". Para enviar los comandos se requiere el programa HyperTerminal, algún programa parecido o realizar una aplicación propia.

Todos los comandos son de una o dos letras, no se distingue entre mayúsculas y minúsculas y están delimitados por una coma. El nombre del circuito y el código para autenticación distingue entre mayúsculas y minúsculas. Los comandos se dividen en 5 categorías:

- Set Commnads: Almacena información, los cambios no tienen efectos hasta reiniciar el modulo, ver tabla 3.4.

Comando	Descripción
S7,<1,0>	Modo 7 bits de datos, 1 para habilitar, 0 para desactivar
SA,<1,0>	Autenticación, 1 para habilitar, 0 para desactivar
SE,<1,0>	Encriptación,1 para habilitar, 0 para desactivar
SF,1	Establecer valores predeterminados de fabrica
SL,<E,O,N>	Establecer paridad, E = par, O = impar, N = ninguna
SN,<nombre>	Nombre del dispositivo, como máximo 20 caracteres
S-,<nombre>	Nombre del dispositivo agregando los dos últimos bytes de la dirección del dispositivo
SP,<clave>	Código de seguridad, máximo 20 caracteres
SS,<nombre>	Nombre del servicio, máximo 20 caracteres
ST,<número>	Tiempo para entrar a la configuración (número entre 0 y 255), 0 = no se puede configurar, 60 = dentro de los primero 60 segundos, 255 = todo el tiempo se puede entrar a configurar
SU,<Bauds>	Tasa de transmisión, solo es necesario poner los dos primeros números {1200,2400,4800,9600,19.2,28.8,38.4,57.6,115K,230K,460K,921K}
SW,<número>	Activa el modo de bajo consumo, el número debe de estar en milisegundos, el dispositivo despertará para ver la actividad de RF

SZ,<número>	Permite tasas de transmisión no estándar, número = baudrate * 0.004096
S\$,<carácter>	Permite cambiar el valor predeterminado \$\$\$ a algún otro carácter

Tabla 3.4 Set Commands.

- Get Commands: Recupera y muestra información almacenada, ver tabla 3.5.

Comando	Descripción
D	Muestra configuraciones básicas
E	Muestra configuraciones extendidas
GK	Estado de conexión, 1 = conectado, 0 = desconectado

Tabla 3.5 Get Commands.

- Change Commands: Cambia temporalmente valores como velocidad de transmisión, paridad, etc., ver tabla 3.6.

Comando	Descripción
U,<bauds>,<E,O,N>	Cambio temporal de la configuración de la comunicación serial. Este cambio es inmediato, en bauds se deben de poner los cuatro caracteres exactos: {1200,2400,4800,9600,19.2,28.8,38.4,57.6,115K,230K,460K,921K} La paridad debe de ir en mayúsculas: E = par, O = impar, N = ninguna

Tabla 3.6 Change Commands.

- Action Commands: Realiza acciones como búsquedas, conexiones, etc.

Comando	Descripción
\$\$\$	Entrar a modo comando
---	Salir de modo comando
+	Hacer un eco de la información que se recibe
H	Ayuda, imprime la lista de comandos
K	Mata la conexión actual
L	Calidad del enlace
I,<tiempo>	Busca dispositivos Bluetooth, el tiempo de búsqueda e entre 10 y 48 segundos.

Tabla 3.7 Action Commands.

- GPIO Commands: Configura y manipula las señales GPIO.

En las tablas anteriores se muestran algunos comandos y una breve explicación de cada uno de ellos pero cabe aclarar que no son todos los comandos; si se requiere ver otro comando se debe consultar el manual de usuario.

La configuración de ciertos parámetros también se puede hacer por medio de un DIP switch por medio de la interfaz PIO, un uno lógico debe ser de 3 Volts y un cero debe ser 0 Volts, por medio de esta interfaz se pueden restablecer los valores de fábrica, ver la tabla 3.8.

Configuración	Conexión	Apagado	Encendido
Valores de fabrica	PIO 4	Desactivado	Alternar tres veces apagado y encendido para habilitarlo.
Descubrimiento automático	PIO3	Desactivado	Activado
Conexión automática	PIO 6	Desactivado	Activado
Tasa de transmisión	PIO 7	Velocidad programada	9600 Bauds

Tabla 3.8 Configuraciones por hardware.

Bibliografía:

- Roving Networks, “ RN-Bluetooth–UM version 4.77”, Conjunto de comandos, Noviembre 2009.
- Roving Networks, “ DS-RN41-V3.1”, Hoja de especificaciones, Agosto 2009.
- <http://www.csr.com/products/29/bluecore4-ext>

Capítulo 4

Diseño del Nuevo Subsistema de Comunicaciones Inalámbricas

4.1 Introducción

El actual Subsistema de Comunicaciones Inalámbricas de SATEDU tiene limitantes, como la necesidad de una tarjeta de comunicaciones propietaria para cada computadora que se quiere conectar con SATEDU, o para cualquier otro dispositivo personal que se quiera conectar necesita.

La actualización del subsistema de comunicaciones hará más versátil a SATEDU ya que se podrá conectar de forma simple con cualquier computadora o algún otro dispositivo que cuente con bluetooth, además de que la actualización redundará en beneficios económicos y simplicidad de conexión.

En este capítulo se muestra el diseño del nuevo Subsistema de Comunicaciones Inalámbricas de SATEDU, este nuevo subsistema solo empleará una tarjeta de comunicaciones en SATEDU y se comunicara vía bluetooth con la estación terrena que ahora podrá ser emulada por una computadora personal, pero también por otros dispositivos como teléfonos celulares o una PDA que tengan bluetooth.

Cabe aclarar que para utilizar un teléfono celular o una PDA como estación terrena se necesitará desarrollar posteriormente el software para el envío de comandos (software de estación terrena) dependiendo del sistema que tenga el dispositivo; por ejemplo si es un teléfono inteligente con Android se necesita desarrollar una aplicación compatible con este sistema operativo, cosa que no resulta difícil ya que existen una gran cantidad de programas para el desarrollo de aplicaciones para teléfonos en la actualidad.

4.2 Arquitectura Propuesta Para la Tarjeta de Comunicaciones Inalámbricas Bluetooth

El hardware de cada uno de los subsistemas de SATEDU es de alta calidad. Para el desarrollo de todas las tarjetas que constituyen a SATEDU se han empleado herramientas de diseño asistido por computadora. Una vez desarrollada la versión final del circuito impreso este se envía a un proveedor que ha dado servicio al Instituto de Ingeniería desde hace tiempo.

En el caso de la nueva tarjeta de comunicaciones inalámbricas para SATEDU, su corazón es el circuito RN-41 de la compañía Roving Networks; como se vio en el capítulo anterior este circuito cuenta con antena, un switch de RF, memoria flash, oscilador y un chip que tiene dentro un microcontrolador, una interfaz de RF y una interfaz de puerto serie entre otras cosas. El circuito opera con 3.3 Volts y en sus entradas no soporta más de 3.6 Volts, se debe de tener mucho cuidado con el voltaje suministrado en cualquiera de sus pines.

El circuito RN-41 utiliza la interfaz bluetooth para conectarse inalámbricamente con la computadora y una interfaz serial para conectarse con el subsistema de computadora de vuelo de SATEDU. Debido a que la conexión con este subsistema es por medio de un microcontrolador que trabaja con niveles de voltaje de 5 Volts, se debe cuidar que el voltaje a la entrada del circuito RN-41 no sea mayor a 3.6 para no dañarlo y también cuidar que el voltaje en las salidas del RN-41 este por arriba del umbral de decisión del microcontrolador.

El diagrama de bloques de la Tarjeta de Comunicaciones Inalámbricas Bluetooth se muestra en la figura 4.1:

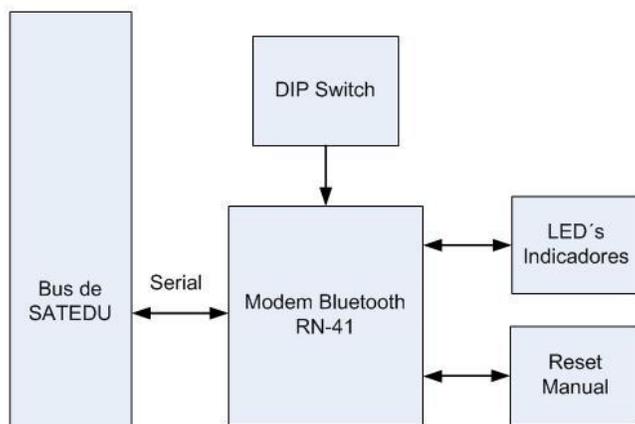


Figura 4.1 Diagrama de Bloques de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Como se puede ver el diagrama es muy sencillo, de hecho el circuito podría operar solo con la conexión al bus de SATEDU, la alimentación de 3.3 Volts y de tierra. A continuación se explica cada uno de los bloques del diagrama:

- Los LED's indicadores, figura 4.2, son dos y se utilizan para mostrar visualmente el estado en el que se encuentra la conexión, uno de ellos se enciende cuando ésta se realiza y se apaga cuando no existe conexión con ningún dispositivo. El otro parpadea normalmente y se apaga cuando se realiza una conexión.

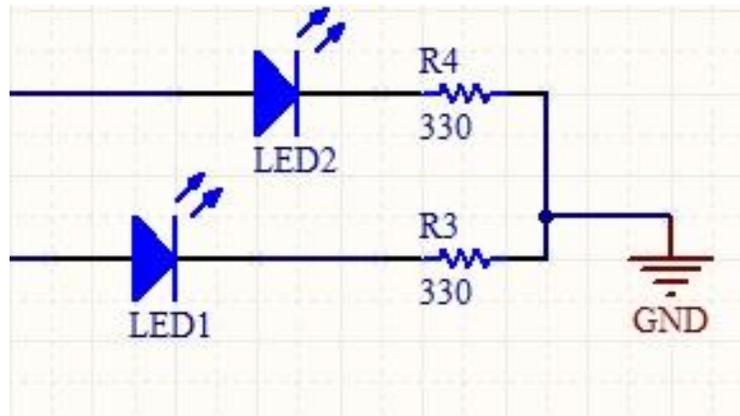


Figura 4.2 LED's Indicadores.

- El reset manual, figura 4.3, solo apaga y prende el circuito, dado que es un reset con nivel bajo de energía (0 Volts) se conecta una resistencia entre el pin y 3.3 Volts para mantener un uno lógico y por medio de un switch se hace el reset.

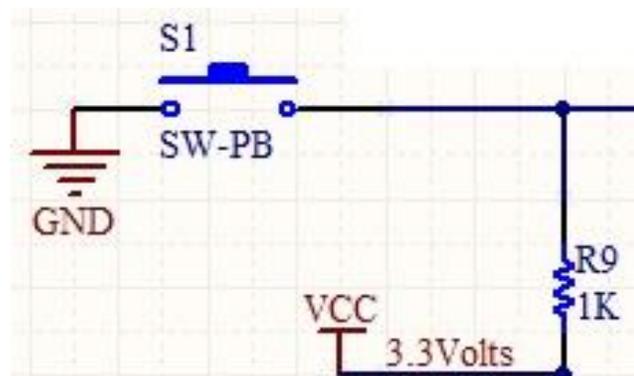


Figura 4.3 Reset Manual.

- El DIP switch, figura 4.4, como se vio en el capítulo tres se utiliza para modificar ciertos parámetros del circuito por hardware; como restaurar los valores de fábrica, tasa de transmisión, conexión automática y descubrimiento automático.

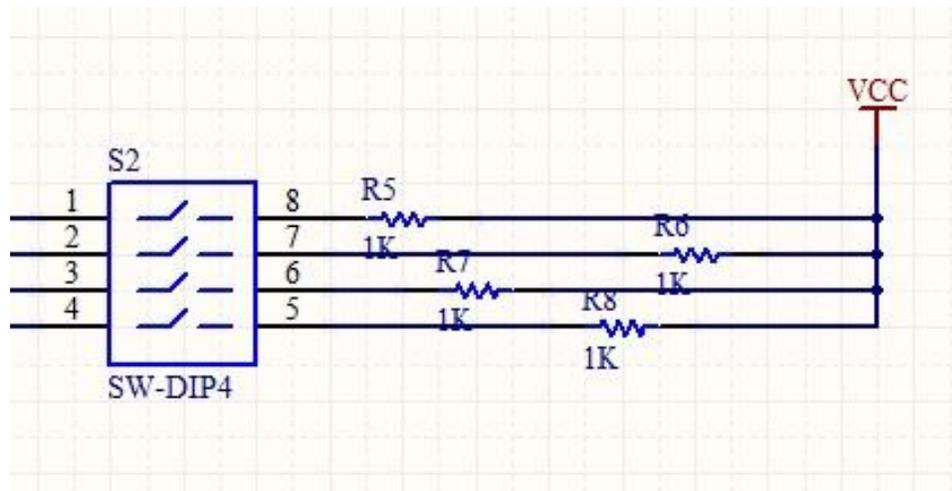


Figura 4.4 DIP Switch.

- El BUS SATEDU corresponde a dos conectores de costilla que tiene cada tarjeta en los que residen señales de comunicación y niveles de voltaje como se vio en el capítulo 1.

La conexión con el puerto serie requiere atención, ya que como se mencionó anteriormente el circuito no soporta más de 3.6 Volts a su entrada, para la recepción por puerto serial se usa un divisor de voltaje, figura 4.4, conectando dos resistencias en serie, una de 10 K Ω y otra de 20 K Ω . En vista de que el voltaje en el transmisor del puerto serial del microcontrolador es de 5 Volts con el divisor de voltaje se tendrá en el receptor un voltaje de 3.3 Volts.

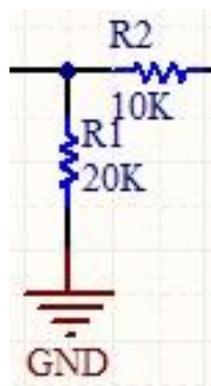


Figura 4.4 Divisor de Voltaje.

El umbral de recepción para distinguir entre un uno y un cero lógico de un microcontrolador depende del voltaje con el que es alimentado, este generalmente es del orden del 25% del Voltaje con el que es alimentado. Para nuestro caso el umbral del uno lógico esta alrededor de los 1.2 Volts, por lo cual no se tendrá problema entre el transmisor serial del circuito RN-41 y el receptor serial del microcontrolador.

El programa de diseño utilizado para la realización del esquemático y circuito impreso fue Protel DXP de la compañía Altium, este programa ha sido utilizado para realizar otros circuitos impresos con buenos resultados en el Instituto de Ingeniería.

La metodología de diseño utilizada con el programa Protel DXP fue la siguiente:

1. Creación del proyecto de la nueva tarjeta.
2. Diseño del circuito esquemático.
3. Verificación del circuito esquemático.
4. Generación de la lista de conexiones.
5. Generación de la tarjeta impresa con las dimensiones adecuadas.
6. Posicionamiento de los componentes en la tarjeta.
7. Ruteo de conexiones.
8. Verificación final de las reglas de diseño.
9. Generación de archivos de salida para la manufactura.

4.3 Diseño Esquemático de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

El esquemático de la Tarjeta de Comunicaciones Inalámbricas Bluetooth es una representación de los componentes en el circuito impreso y como se conectaron, este tipo de representaciones de un circuito impreso son importantes ya que para cambios o modificaciones es más fácil ver este tipo de figuras que analizar el circuito impreso.

En la figura 4.5 se muestra la forma en que se conecta cada componente de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

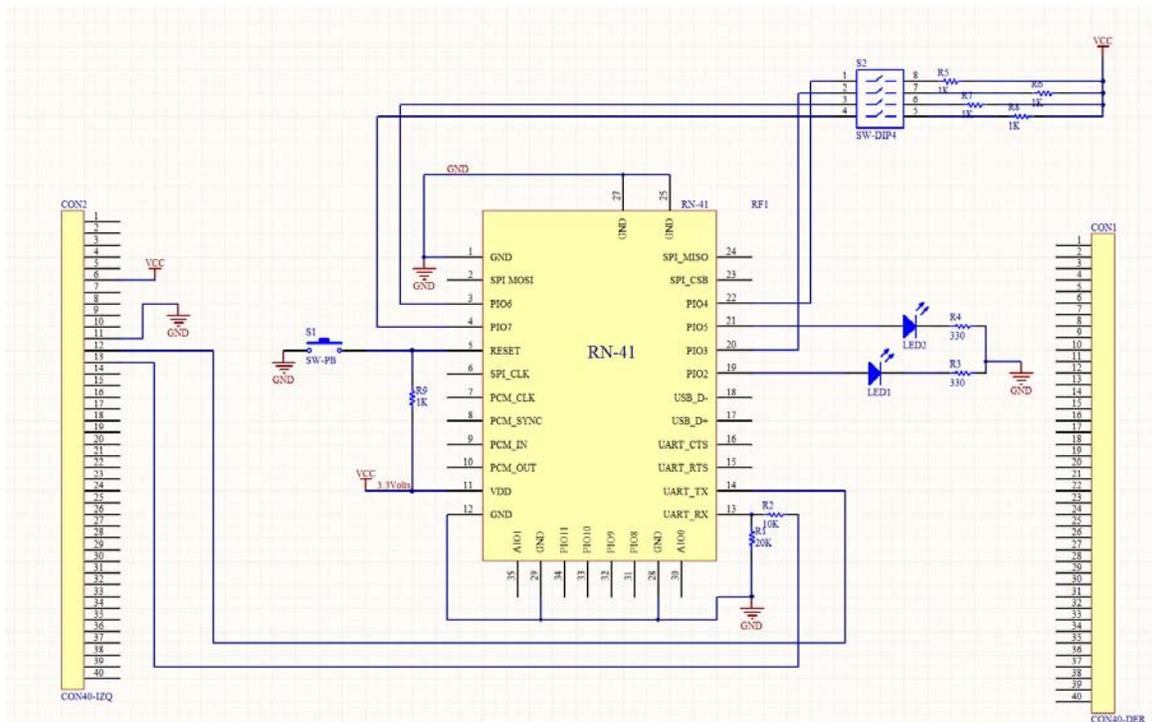


Figura 4.5 Diagrama esquemático de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Los pines del circuito que se ocuparon se muestran en la tabla 4.1.

PIN	Nombre	Conexión
1	GND	Tierra
3	PIO6	DIP Switch 3
4	PIO7	Dip Switch 4
5	RESET	Switch tipo Push Button
11	VDD	Nivel de 3.3 Volts del bus de SATEDU
12	GND	Tierra
13	UART-RX	Transmisor Serial de la computadora de vuelo
14	UART-TX	Receptor Serial de la computadora de vuelo
19	PIO2	LED 1
20	PIO3	DIP Switch 2
21	PIO5	LED 2
22	PIO4	DIP Switch 1
25, 27-29	GND	Tierra

Tabla 4.1 Pines ocupados del circuito RN-41.

4.4 Diseño del Circuito Impreso Para la Tarjeta de Comunicaciones Inalámbricas Bluetooth

Una vez que se tiene el diagrama esquemático se debe de revisar que todo esté bien conectado. Para realizar esta tarea Protel DXP tiene una matriz de conexiones en la que se muestra si éstas están bien realizadas o no según las reglas de diseño que se establezcan. Como se ve en la figura 4.6, cada una de estas conexiones tiene un nivel de importancia pero este se puede cambiar modificando las reglas dependiendo de lo que el diseñador requiera.

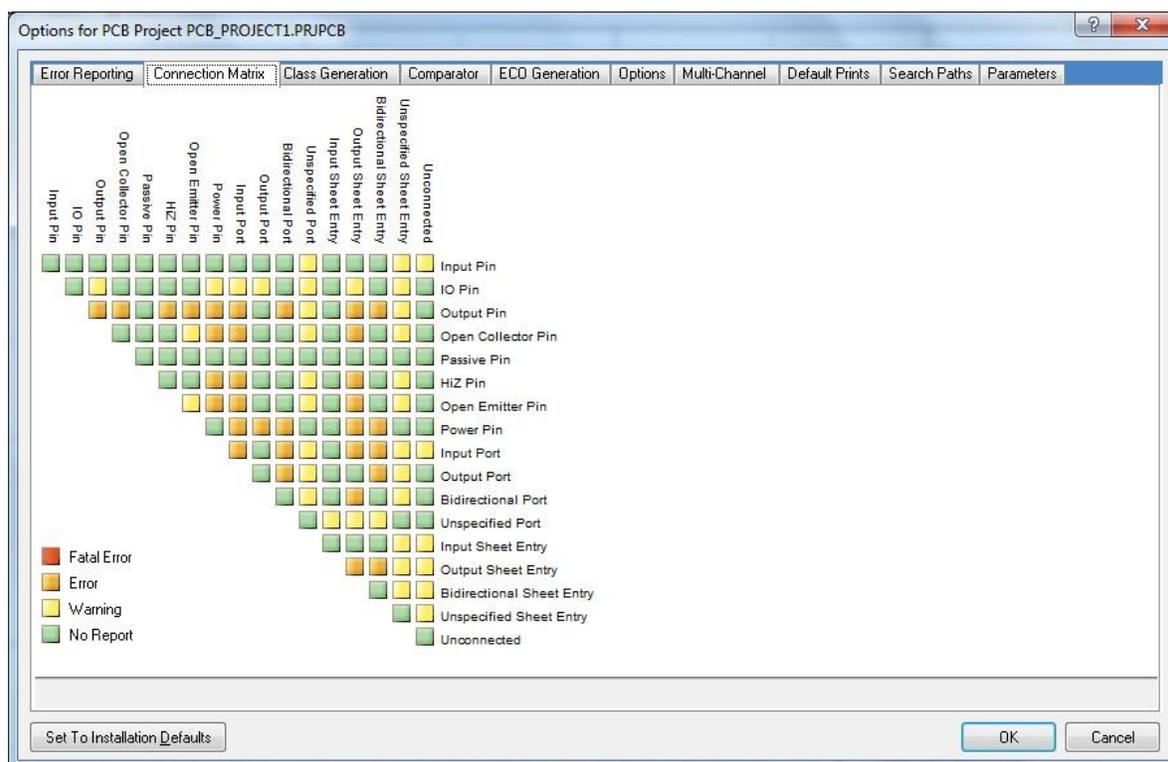


Figura 4.6 Matriz de reglas de conexión.

Una vez que se revisan estas reglas se compila el proyecto, si al compilar no se presenta ningún tipo de error se procede a generar la tarjeta impresa.

Nuestra tarjeta es sencilla así que se realizó el PCB (Printed Circuit Board, circuito impreso) con solo una capa y totalmente compatible con SATEDU, es decir una tarjeta cuadrada de 8.9 centímetros por lado y con conectores de costilla. Protel DXP tiene una herramienta que realiza esta tarea de manera rápida y fácil, que conduce al usuario paso a paso y configura el tipo de circuitos que se colocan en la tarjeta, el número de capas que tendrá la tarjeta impresa, el tamaño y la forma de la tarjeta y el tamaño de las vías; esta herramienta se llama PCB BoardWizard.

Una vez generada la tarjeta se deben de posicionar los componentes en ella, pero primero se asocia el PCB con el proyecto para posteriormente desde el diagrama esquemático cargar los componentes a la tarjeta. La huella de cada uno de los componentes se coloca fuera del PCB para su posterior colocación al interior, figura 4.7.

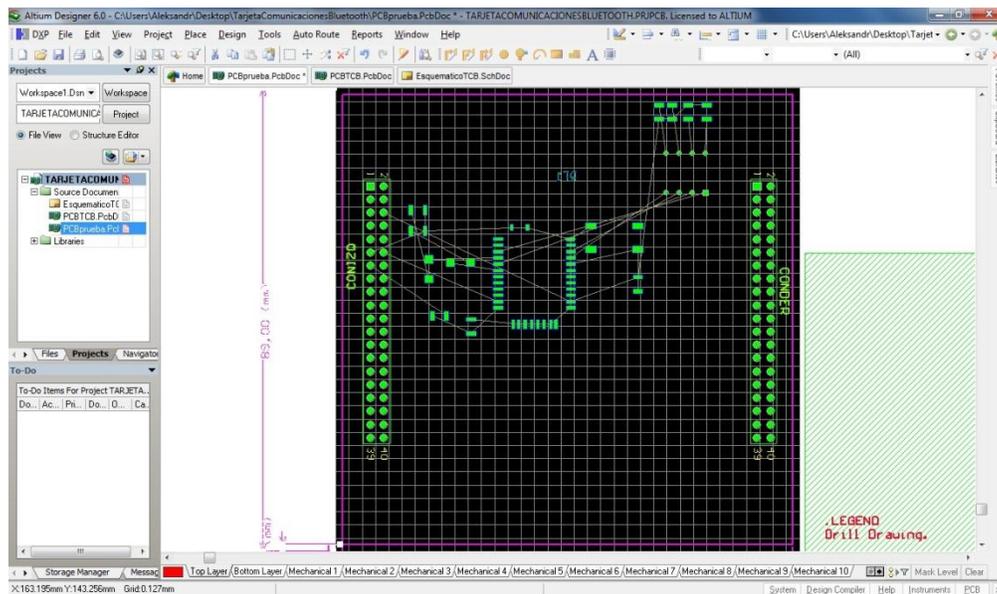


Figura 4.7 Colocación de las huellas de los componentes.

Cabe mencionar que Protel DXP también permite crear nuevas huellas de circuitos que no tenga en su base de datos, esta herramienta se utilizó para crear la huella del circuito RN-41, ya que sus librerías contienen solo circuitos básicos. Esta huella se tomó de la hoja de especificaciones del RN-41, figura 4.8.

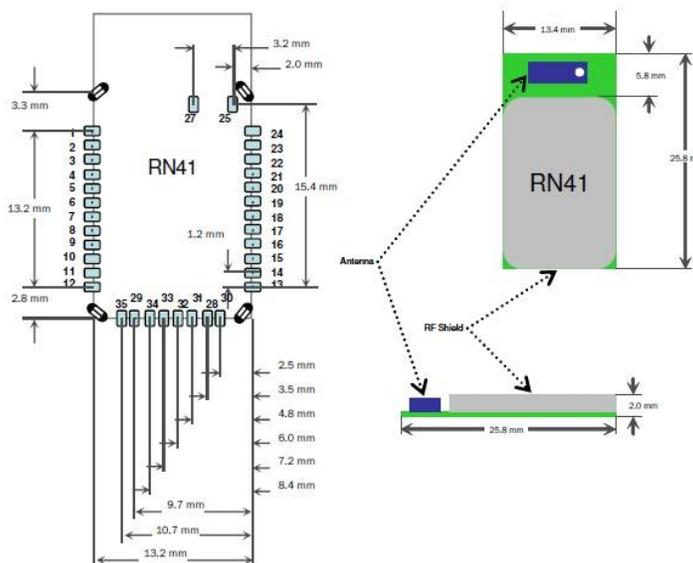


Figura 4.8 Huella de impresión del circuito RN-41.

La colocación de componentes puede ser difícil porque se desconoce cuál es el lugar correcto para colocarlos. En la colocación de componentes lo que se hizo fue poner todos los componentes de cada uno de los bloques lo más junto posible para realizar conexiones cortas y fáciles.

Una vez colocados cada uno de los componentes se deben de trazar las pistas que los unen, esta es una de las partes más importantes en el diseño de los circuitos impresos ya que se debe optimizar el espacio en la tarjeta, Protel DXP cuenta con herramientas para ruteo manual o ruteo automático. Se utilizó el ruteo manual con pistas de 0.25 milímetros de ancho.

La hoja de especificaciones del circuito RN-41 establece que no se debe de colocar ninguna pista debajo del lugar donde está la antena. Una vez ruteada la tarjeta se tiene el circuito impreso listo para su construcción, figura 4.9.

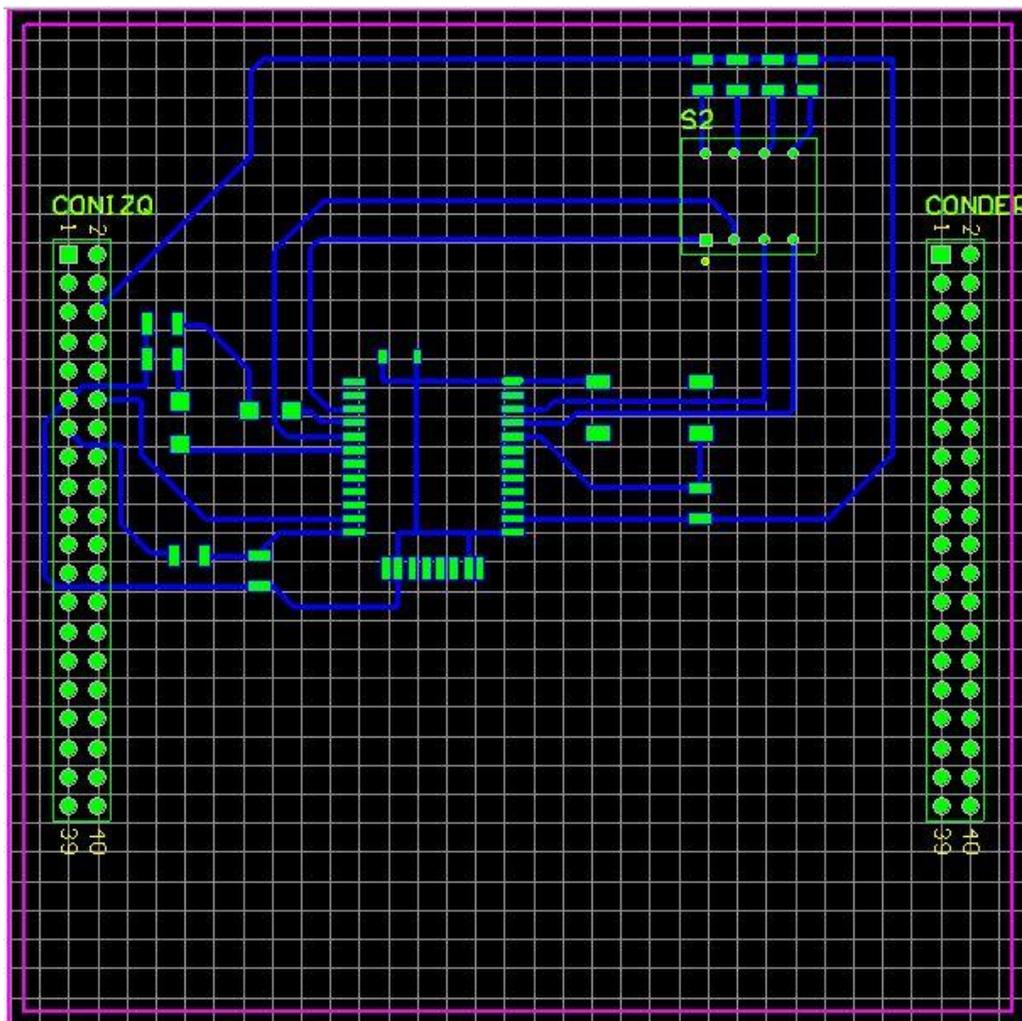


Figura 4.9 PCB de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Adicionalmente Protel DXP permite la visualización en 3D del circuito impreso, figura 4.10.

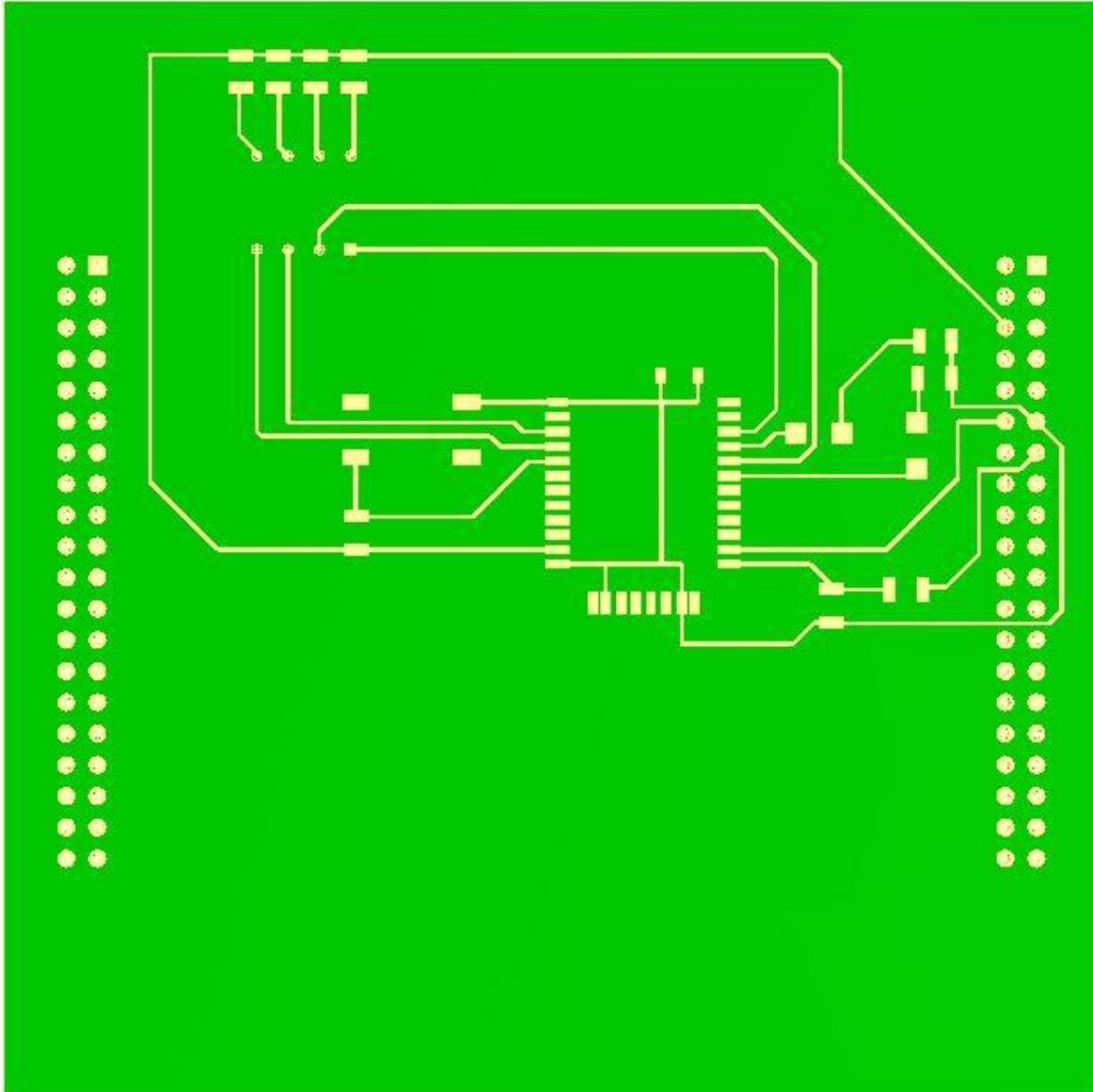


Figura 4.10 Tarjeta de Comunicaciones Inalámbricas Bluetooth.

4.5 Fabricación del Circuito Impreso y Pruebas Eléctricas Preliminares

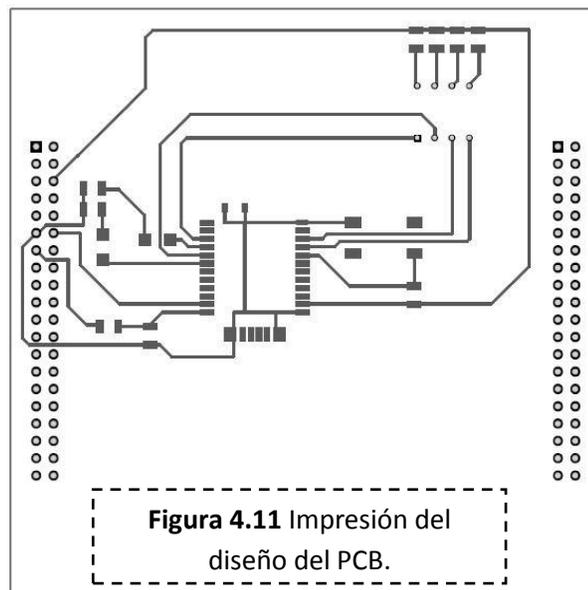
Antes fabricar la Tarjeta de Comunicaciones Inalámbricas Bluetooth se realizó un prototipo de la tarjeta en una placa fenólica, esto es para realizar pruebas preliminares con este PCB y hacer correcciones si es que se necesitan o realizar cambios, de tal manera que el circuito impreso final este totalmente probado y no se necesite hacer ninguna modificación o corrección.

Para la fabricación del circuito impreso se ocupó el siguiente material:

- Papel couche.
- Placa fenólica.
- Guantes de látex.
- Un pedazo de tela delgada.
- Cinta adhesiva.
- Cloruro férrico.
- Algodón y alcohol isopropílico.
- Un recipiente.

El procedimiento para la fabricación artesanal del circuito impreso se describe a continuación:

1. Primero se imprimió el diseño de las pistas en el lado brillante del papel couche, la impresora debe ser de tipo laser para que la tinta se adhiera bien al cobre de la placa fenólica. Protel DXP nos proporciona la impresión del circuito impreso, figura 4.11.



- Después se lijo y limpió la cara de cobre de la placa fenólica con alcohol isopropílico para remover grasa, manchas o cualquier cosa que afecte el funcionamiento del cloruro férrico sobre la placa, para esto se utilizaron guantes de latex para evitar manchar la placa fenólica con la grasa de nuestra piel.
- Una vez que se tuvo el dibujo de las pistas y la placa limpia, figura 4.12, se colocó el dibujo sobre la placa fenólica del tamaño apropiado, después se acomodó y aseguró con cinta adhesiva para asegurar que no se mueva.

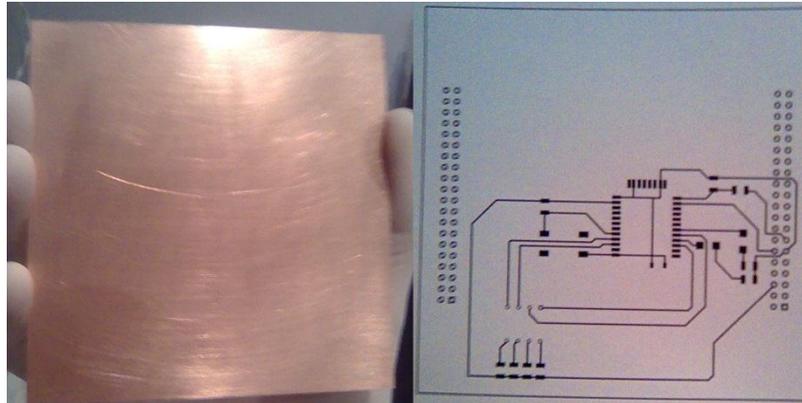


Figura 4.12 Placa Fenólica y dibujo del PCB.

- Después se cubrió el papel couche con un pedazo de tela delgada y se le paso una plancha por encima, esto es para calentar el cobre de la placa y que la tinta se adhiriera al cobre; se debe cuidar que el papel couche no se queme, una vez calentado el cobre se retira el papel, la tinta queda en la placa fenólica marcando las pistas del PCB, figura 4.13, adicionalmente si no queda bien marcada alguna pista se puede corregir con un marcador de tinta permanente.

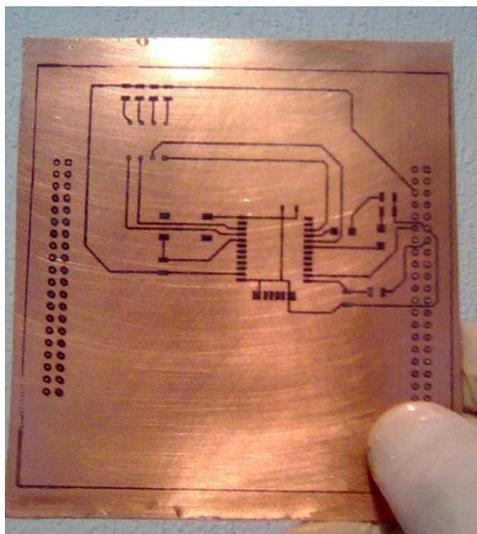


Figura 4.13 Placa Fenólica con las pistas marcadas.

5. Se llenó un recipiente con cloruro férrico acto seguido se metió la placa fenólica; el cloruro férrico produce la oxidación del cobre que no está protegido por la tinta haciéndolo desaparecer, para acelerar el proceso de oxidación se puede calentar a baño maría el cloruro férrico, agregar agua oxigena (se debe de usar cubre bocas) o mover el recipiente para que el cloruro fluya sobre la placa.
6. Una vez que se observa que las pistas están bien definidas, se debe de sacar la placa fenólica del cloruro férrico.
7. Después se limpió bien la placa fenólica con un estropajo y agua para limpiar la tinta, adicionalmente se limpió el cobre con un algodón mojado con alcohol isopropílico, figura 4.14.

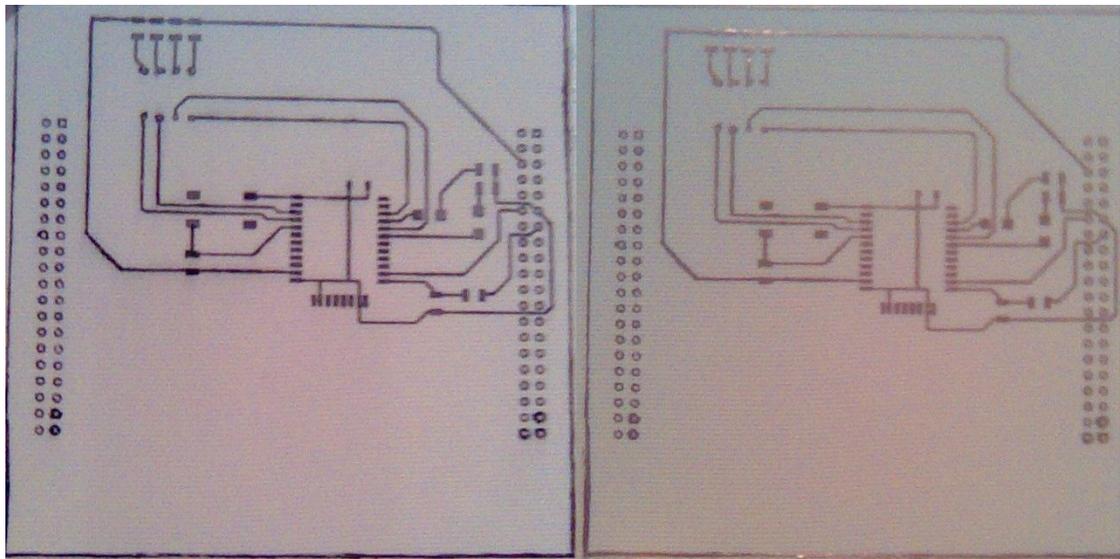


Figura 4.14 Circuito Impreso.

8. Una vez con la placa limpia, se verificó la continuidad de las pistas con un multímetro, si alguna no tuviese continuidad esto se podría reparar con un poco de soldadura; para los cortos circuitos lo que se hace es remover el cobre con un desarmador.
9. Para finalizar se le hicieron los orificios de los conectores de costilla y el DIP switch, para esto primero se marcó cada orificio con un clavo para que no se escape la broca al realizar los orificios con un taladro; por último se checo continuidad de las pistas nuevamente.

Una vez fabricado el circuito impreso, se empieza el montaje de cada uno de los componentes.

4.6 Ensamble de la Tarjeta con Componentes Electrónicos

Los componentes de la nueva tarjeta de comunicaciones inalámbricas deben ser de montaje superficial dado que de esta manera está diseñada la tarjeta, éstos se enumeran a continuación:

1. Circuito RN-41.
2. LED rojo.
3. LED verde.
4. Dos resistencias de 330 ohms.
5. Cinco resistencias de 1,000 ohms.
6. 1 DIP Switch de 4 terminales.
7. 1 Push Button.
8. 1 resistencia de 10,000 ohms.
9. 1 resistencia de 20,000 ohms.
10. 2 conectores de costilla de 40 terminales compatibles con SATEDU.

Cada una de las componentes se soldó a la Tarjeta de Comunicaciones Inalámbricas Bluetooth con caudín, flux, pinzas y soldadura. Esta parte se debe realizar con mucho cuidado dado que si se el caudín se tiene mucho tiempo en una pista esta se puede desprender o se puede dañar el pin del circuito.

Para el soldado primero se pusieron las resistencias, al terminar de soldar cada una de ellas se verificaba la continuidad de las pistas para evitar cortos circuitos o discontinuidades provocadas por el calor del caudín. Después se soldaron los LED's siguiendo el mismo procedimiento, posteriormente se soldaron el push button y el Dip Switch, figura 4.15, y por último el circuito RN-41. Para soldar de manera más fácil este último circuito lo que se hizo fue colocarlo y después asegurarlo con un pedazo de cinta adhesiva para después soldar dos de sus patas y finalmente soldar las demás.



Figura 4.15 Tarjeta de Comunicaciones Inalámbricas Bluetooth, vista de frente.

4.7 Pruebas Electrónicas Preliminares Aplicadas a la Tarjeta de Comunicaciones Inalámbricas Bluetooth

Una vez soldados los componentes a la tarjeta, se inspeccionó visualmente para verificar que no existía ningún defecto y por último también se verificó la continuidad de las pistas, en busca de defectos o cortos circuitos.

En caso de encontrar algún defecto como por ejemplo que no se tiene continuidad en las pistas se recomienda reparar con un poco de soldadura y un pequeño cable de par trenzado para remplazar la pista.

Finalmente al no encontrar ningún defecto en la tarjeta con los componentes montados, se tiene lista la Tarjeta de Comunicaciones Inalámbricas Bluetooth, figura 4.16, para probarla inicialmente por separado y después integrarla a SATEDU para validar la comunicación con la Estación Terrena.

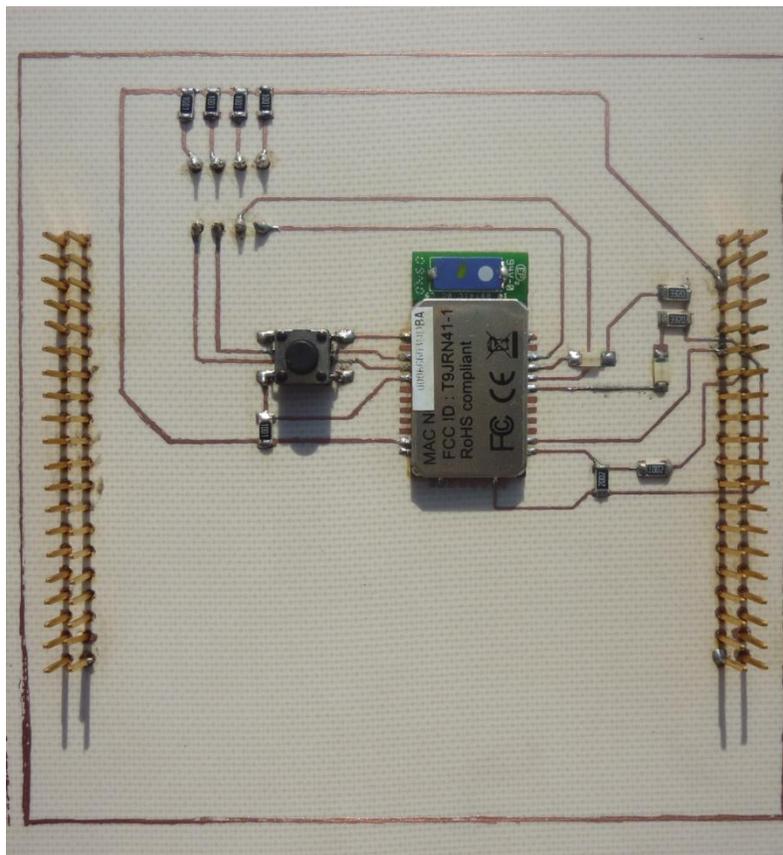


Figura 4.15 Tarjeta de Comunicaciones Inalámbricas Bluetooth, vista posterior.

En el siguiente capítulo se hablará del software necesario para la configuración del circuito RN-41.

Bibliografía:

- Altium, “Getting Started with PCB Desing”, Tutorial, Noviembre 2005.
- Roving Networks, “ DS-RN41-V3.1”, Hoja de especificaciones, Agosto 2009.

Capítulo 5

Software de Operación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

5.1 Introducción

Para la comunicación entre SATEDU y su estación terrena se necesita ajustar ciertos parámetros de comunicación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth como la tasa de transmisión, el número de bits de información y la paridad de datos entre otros. Como se vio en el capítulo 3 estos parámetros se configuran en el circuito RN-41; para realizar esta configuración se necesita una computadora con bluetooth o la conexión del circuito a un puerto serie.

Como ya se había mencionado las computadoras personales actuales no cuentan con puertos series disponibles por tal razón la configuración del circuito es más factible por medio de una computadora que cuente con bluetooth ya que si se desea configurar por medio de un puerto serie se tendría que ocupar un PIC o un adaptador para dotar a la computadora de un puerto serie virtual. Además, cada vez que se necesite modificar algún parámetro se tendría que conectar a la computadora nuevamente y como el caso es dotar a SATEDU de un sistema de comunicaciones inalámbrico, para la configuración de la nueva tarjeta de comunicaciones por Bluetooth se optó por desarrollar una interfaz de software.

En este capítulo se muestra como se desarrolló dicha interfaz para la configuración del circuito RN-41, este software se desarrolló en el lenguaje C# con el programa Microsoft Visual Studio 2010. La interfaz desarrollada se buscó que sea amigable con el usuario e intuitiva al manejar.

La razón de desarrollar este programa en Microsoft Visual Studio 2010 es que es muy intuitivo y su interfaz gráfica permite facilidad en el desarrollo.

5.2 Plataforma de Desarrollo Para la Interfaz de Usuario

La plataforma de desarrollo para la interfaz de configuración de los parámetros de comunicación de la Tarjeta de Comunicaciones Bluetooth es Microsoft Visual Studio 2010, este programa constituye un potente IDE (Integrated Development Environment, Ambiente de Desarrollo integrado).

Un IDE es un programa que está compuesto por un conjunto de herramientas para la programación, en este caso Visual Studio 2010, figura 5.1, soporta varios lenguajes de programación como el famoso Visual Basic por ejemplo, pero para el desarrollo de la interfaz de configuración se utiliza el lenguaje de programación Visual C#.

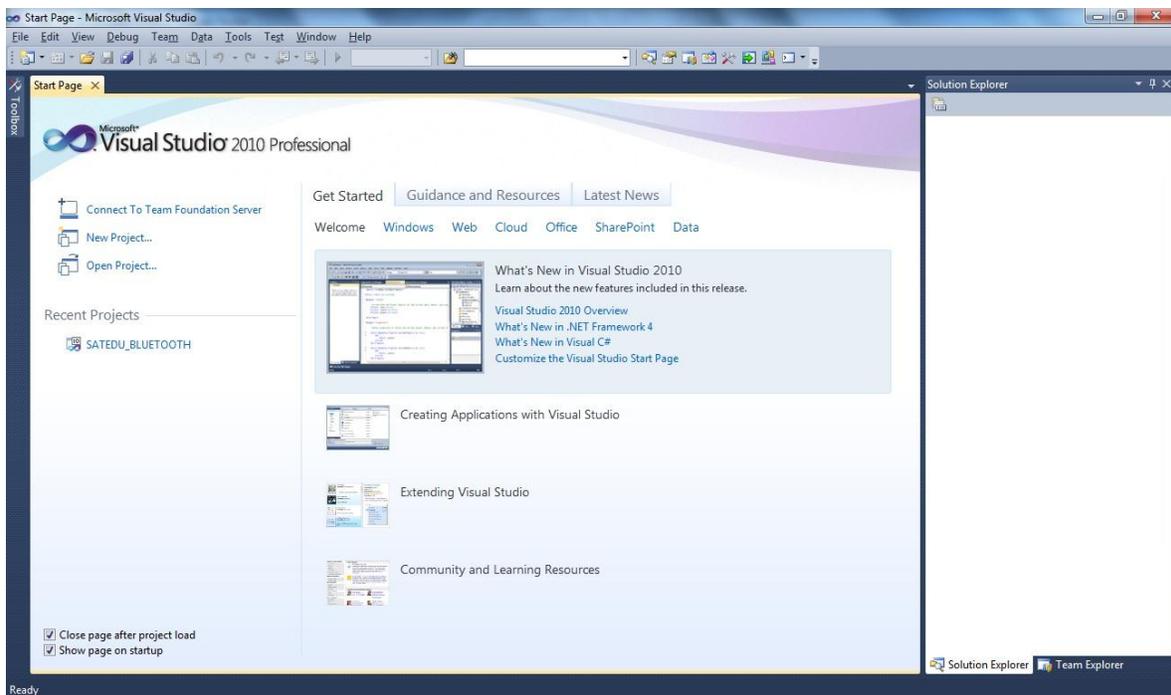


Figura 5.1 Microsoft Visual Studio 2010.

El IDE está compuesto por:

- **Editor de Código:** Es un editor de texto en el cual se escribe el código del programa, este editor contiene la función de autocompletar y coloreado de sintaxis que permite ver cuando existen errores de sintaxis y corregirlos fácilmente.
- **Depurador:** El depurador es el encargado de verificar que el código del programa contenga instrucciones validas o acceda a lugares de memoria permitidos, para esto el depurador revisa cada línea de código y muestra los errores en pantalla para su corrección.

- **Compilador:** Un compilador es un programa que traduce de un lenguaje de programación (en el cual se escribe el código fuente) a otro lenguaje que pueda entender la computadora (lenguaje maquina).
- **Un constructor de interfaz gráfica:** Es un programa que por medio de imágenes u objetos representa acciones disponibles en la programación, es una forma de programación visual, su principal función es una interacción amigable usuario-maquina.

Esta última característica es la que facilita el rápido desarrollo de aplicaciones gráficas. Visual C# es un lenguaje orientado a objetos desarrollado por Microsoft, la combinación de las interfaces gráficas y la programación orientada a objetos permite lo que se conoce como programación dirigida a eventos.

Una programación dirigida a eventos permite que la interfaz gráfica responda a eventos, este evento puede ser dar clic en un botón, o bien, en el caso del desarrollo de la interfaz de configuración para la Tarjeta de Comunicaciones Inalámbricas Bluetooth la recepción o transmisión por puerto serie.

5.3 Necesidades de Programación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

Para una buena comunicación entre la estación terrena (una computadora) y SATEDU se tienen que configurar ciertos parámetros; la interfaz desarrollada en este capítulo será capaz de configurar ciertos parámetros. Estos parámetros y sus respectivos códigos así como la acción que realizan se muestran en la tabla 5.1.

Comando	Descripción
S7,<1,0>	Modo 7 bits de datos, 1 para habilitar, 0 para desactivar
SA,<1,0>	Autenticación, 1 para habilitar, 0 para desactivar
SE,<1,0>	Encriptación,1 para habilitar, 0 para desactivar
SF,1	Establecer valores predeterminados de fabrica
SL,<E,O,N>	Establecer paridad, E = par, O = impar, N = ninguna
SN,<nombre>	Nombre del dispositivo, como máximo 20 caracteres
S-,<nombre>	Nombre del dispositivo agregando los dos últimos bytes de la dirección del dispositivo
SP,<clave>	Código de seguridad, máximo 20 caracteres
ST,<número>	Tiempo para entrar a la configuración (número entre 0 y 255), 0 = no se puede configurar, 60 = dentro de los primero 60 segundos, 255 = todo el tiempo se puede entrar a configurar
SU,<Bauds>	Tasa de transmisión, solo es necesario poner los dos primeros números {1200,2400,4800,9600,19.2,28.8,38.4,57.6,115K,230K,460K,921K}

SW,<número>	Activa el modo de bajo consumo, el número debe de estar en milisegundos, el dispositivo despertará para ver la actividad de RF
E	Muestra configuraciones extendidas
U,<bauds>,<E,O,N>	Cambio temporal de la configuración de la comunicación serial. Este cambio es inmediato, en bauds se deben de poner los cuatro caracteres exactos: {1200,2400,4800,9600,19.2,28.8,38.4,57.6,115K,230K,460K,921K} La paridad debe de ir en mayúsculas: E = par, O = impar, N = ninguna
\$\$\$	Entrar a modo comando
---	Salir de modo comando
+	Hacer un eco de la información que se recibe
H	Ayuda, imprime la lista de comandos

Tabla 5.1 Comandos implementados por la Interfaz.

Adicionalmente se contará con una ventana de texto para introducir cualquier código que sea válido y no esté implementado en la interfaz de configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

5.4 Desarrollo de la Interfaz de Usuario

Para empezar a desarrollar la interfaz primero se abre el programa Visual Studio 2010, una vez en este crea un nuevo proyecto, en este escogemos el lenguaje C# y la planilla Aplicación de Windows Form, y posteriormente se guarda el proyecto.

El programa muestra dos ventanas Form1.cs [Desing] y la página de inicio, del lado derecho se pueden ver los demás archivos que contiene el proyecto, de todos los que aparecen utilizaremos Form1.cs que es el editor de texto para programar la interfaz grafica.

Form1 es la ventana de nuestra aplicación, es el constructor de interfaz grafica que nos permite programar visualmente. Si seleccionamos la ventana Form1 [Desing], en la esquina inferior derecha nos aparecerán sus propiedades y en ese mismo lugar se puede modificar cualquiera de ellas. Se modifica el nombre y el texto de la ventana por el de "SATEDU TCB".

Del lado izquierdo de la ventana aparece un cuadro de herramientas que se utiliza para agregar diferentes objetos como botones, puertos serie, relojes, menús, cuadros de dialogo, etc, figura 5.2.

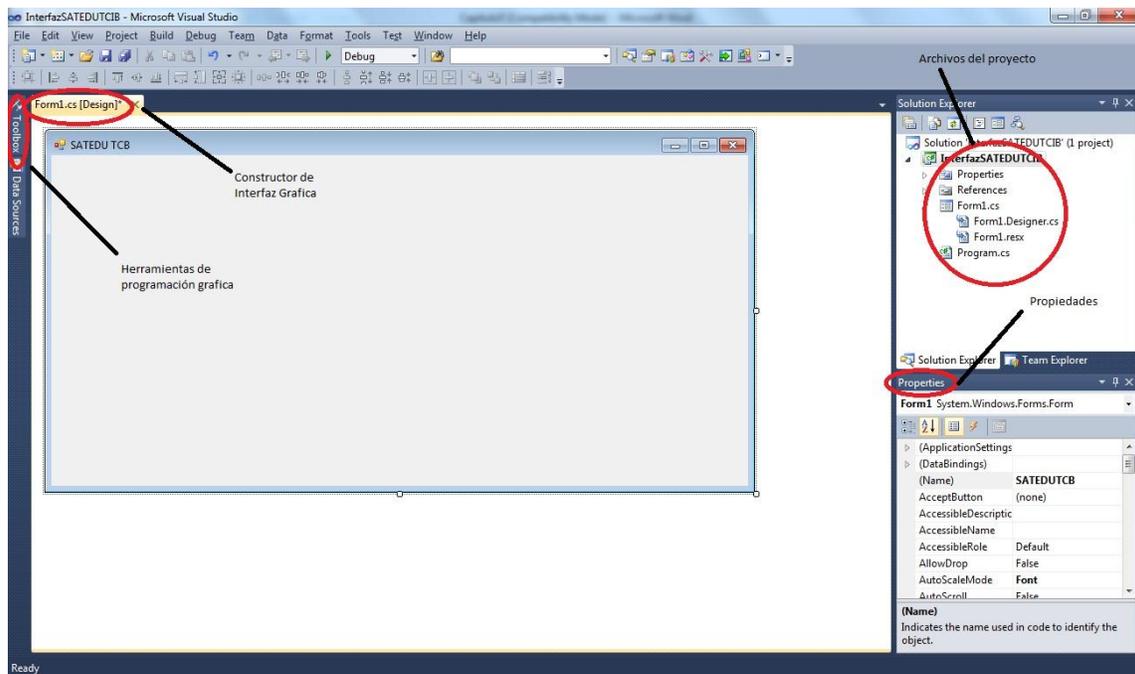


Figura 5.2 Herramientas para el desarrollo de la interfaz de software.

Se agregó un puerto serie, dos cajas de opciones y un botón para abrir el puerto serie que se le asigne a la Tarjeta de Comunicaciones Inalámbricas Bluetooth en la computadora. En las cajas de opciones (combo box) se colocan los puertos COM habilitados en la computadora y en el otro la velocidad de transmisión, también se coloca el botón que abrirá el puerto serie. De igual modo, se agrega un botón para salir de la interfaz, figura 5.3.

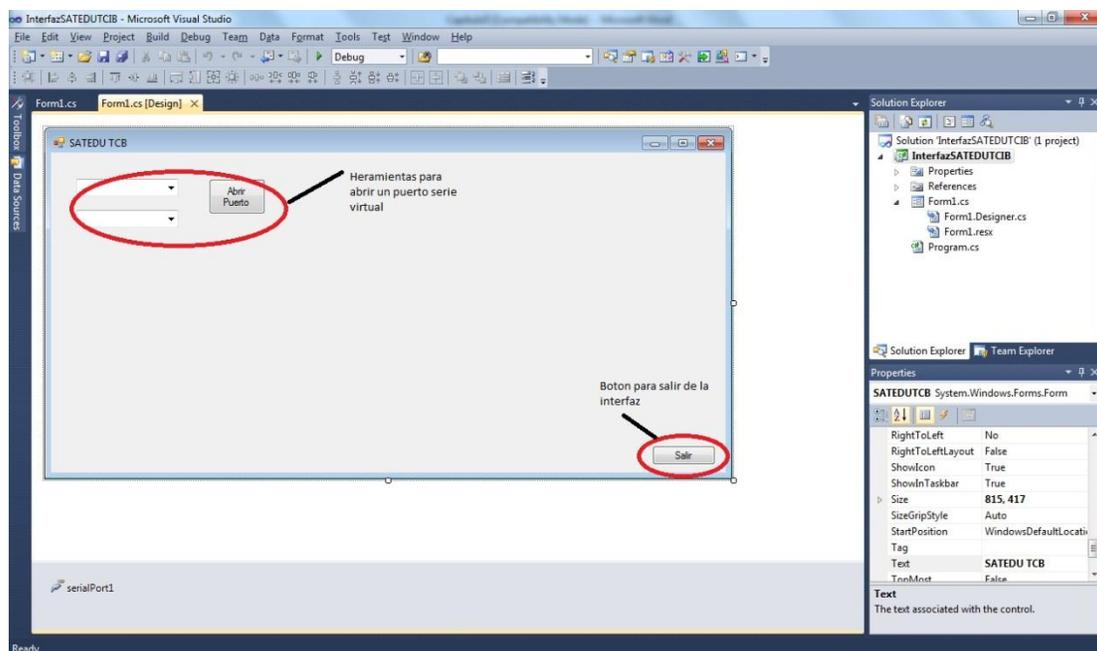


Figura 5.3 Desarrollo de la interfaz.

Después se agregó un botón para entrar y salir del modo comando, pero se cuidó que este botón solo se habilite cuando el puerto serie virtual este abierto. Al presionar el botón Modo Comando lo que se hace es enviar a la Tarjeta de Comunicaciones Inalámbricas Bluetooth tres signos de pesos “\$\$\$” para entrar en el modo comando y cambiar el nombre del botón al de “Salir Modo Comando”. Al pulsar este se mandan tres signos menos “---” para salir del modo comando. También se agregaron botones para habilitar el eco, modo 7 bits, ayuda, restablecer valores de fábrica, habilitar autenticación, habilitar encriptación, establecer paridad, establecer velocidad de transmisión, cambiar tiempo para entrar en modo comando, establecer paridad y velocidad de transmisión de forma momentánea, ver la configuración básica y extendida, cambiar el nombre del dispositivo y cambiar el código pin. Estos dos últimos no pueden ser mayores de 20 caracteres, por eso en sus cuadros de texto no se pueden poner más de 20 caracteres.

También se le agrego un cuadro de texto y un botón para enviar comandos extra que no estén programados en la interfaz pero que se conozca de ellos a través del manual de usuario.

Para finalizar la interfaz se agregó el título de la interfaz y un par de imágenes. Esta es una descripción muy básica de cómo se desarrolló la interfaz si se desea conocer más acerca del desarrollo se sugiere ver el código en el Apéndice A.

5.5 Utilización de la Interfaz de Usuario

La interfaz desarrollada, figura 5.4, es muy fácil de usar y muy intuitiva, lo único que se debe de tener presente en el momento de utilizarla es el tipo de comando que se está utilizando, ya que algunos de los cambios solo tienen efecto después de reiniciar el circuito RN-41.



Figura 5.4 Interfaz de configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Para utilizar la interfaz lo primero que se debe de hacer es seleccionar el puerto al que está asociado la Tarjeta de Comunicaciones Inalámbricas Bluetooth y la velocidad de transmisión. En el momento de pulsar el botón “Abrir Puerto” se habilitará el botón “Modo comando” y las opciones para enviar comandos escribiendo directamente el código.

La interfaz cuenta con 20 botones, 8 cuadros de texto, y tres cuadros de opciones, la descripción de cada uno de ellos se describe a continuación:

- **Botón “Abrir Puerto”:** Este botón se utiliza en conjunto con los dos cuadros de opciones a su izquierda para abrir el puerto vinculado con la Tarjeta de Comunicaciones Inalámbricas Bluetooth, se escoge el puerto COM y la velocidad de transmisión, después se pulsa el botón para abrir el puerto. Al realizar esta acción se habilita el botón “Modo Comando”, el botón “Enviar” y el cuadro de texto asociado a este botón.
- **Botón “Modo Comando”:** Este botón se encuentra deshabilitado hasta que se abre el puerto serie virtual. Para entrar en este modo se debe de pulsar el botón en los primeros 60 segundos después de energizar la Tarjeta de Comunicaciones Inalámbricas Bluetooth (este parámetro se puede cambiar), al pulsar este botón se habilitan todos los botones disponibles y se cambia el nombre del botón por el de “Salir Modo Comando”, al pulsar éste se deshabilitan los botones habilitados y se regresa al nombre original del botón.
- **Botón “Enviar”:** Este botón está asociado con el cuadro de texto arriba de este, ambos permanecen deshabilitados hasta que se abre el puerto serie asociado a la tarjeta. El cuadro de texto permite enviar cualquier comando escribiendo directamente el código y después pulsando el botón enviar. También se podrían mandar comandos a SATEDU por medio de esta opción.
- **Botón “Configuración Básica”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, este botón está asociado al cuadro de texto Configuración, al pulsarlo se obtiene la configuración básica de la tarjeta.
- **Botón “Configuración Extendida”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando; está asociado al cuadro de texto Configuración, al pulsarlo se obtiene la configuración extendida de la tarjeta.
- **Botón “Limpiar”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando; está asociado al cuadro de texto Configuración, al pulsarlo se borran todos los datos contenidos en el cuadro de texto.

- **Botón “Ayuda”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando; está asociado al cuadro de texto Ayuda, al pulsarlo se obtiene una guía rápida de comandos del circuito RN-41.
- **Botón “Activar Eco”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de texto Respuesta, al pulsarlo todo lo que recibe la Tarjeta de Comunicaciones Inalámbricas Bluetooth lo muestra en el cuadro de texto.
- **Botón “Modo 7 bits”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, al pulsarlo se configuran 7 bits de datos y se cambia el nombre del botón por el de “Modo 8 bits”, al pulsar este se configura a 8 bits de datos y el botón regresa a su nombre inicial.
- **Botón “Valores de Fabrica”:** el botón se encuentra deshabilitado hasta que se entra en modo comando, al pulsarlo se restablecen los valores de fábrica del circuito RN-41 (ver tabla 3.3).
- **Botón “Habilitar Encriptación”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, al pulsarlo se habilita la encriptación de datos y se cambia el nombre del botón por el de “Deshabilitar Encriptación”, al pulsar este se deshabilita la encriptación de datos y el botón regresa a su nombre inicial.
- **Botón “Habilitar Autenticación”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, al pulsarlo se habilita la autenticación en el momento de crear un enlace y se cambia el nombre del botón por el de “Deshabilitar Autenticación”, al pulsar este se deshabilita la autenticación y el botón regresa a su nombre inicial.
- **Botón “Cambiar Nombre”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de texto Nombre, al pulsarlo se cambia el nombre de la Tarjeta de Comunicaciones Inalámbricas Bluetooth por el escrito en el cuadro de texto, como máximo este nombre puede tener 20 caracteres.
- **Botón “Cambiar Nombre +MAC”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de texto Nombre, al pulsarlo se cambia el nombre de la Tarjeta de Comunicaciones Inalámbricas Bluetooth por el escrito en el cuadro de texto agregando los últimos 4 caracteres de la MAC del circuito RN-41 utilizado.
- **Botón “Cambiar Código PIN”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de texto Código PIN, al pulsarlo se cambia el código PIN de la Tarjeta de Comunicaciones Inalámbricas Bluetooth por el escrito en el

cuadro de texto, como máximo este código puede tener 20 caracteres y distingue entre mayúsculas y minúsculas.

- **Botón “Establecer Velocidad de Transmisión”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de texto Velocidad de Transmisión, al pulsarlo se cambia la velocidad de transmisión por el valor escrito en el cuadro de texto, solo es necesario escribir los dos primeros números del valor (por ejemplo si es 9600 se escribe 96).
- **Botón “Establecer Paridad”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de opciones Paridad, al pulsarlo se cambia la paridad de la comunicación por el valor elegido en el cuadro de opciones.
- **Botón “Configuración Temporal”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de texto Velocidad de Transmisión y al cuadro de opciones Paridad, al pulsarlo se cambia la paridad por la seleccionada en el cuadro de opciones y la velocidad de transmisión por el valor escrito en el cuadro de texto, es necesario escribir cuatro caracteres (por ejemplo si es 115000 se escribe 115K, ver tabla 5.1).
- **Botón “Establecer Tiempo”:** El botón se encuentra deshabilitado hasta que se entra en el modo comando, está asociado al cuadro de texto Tiempo, al pulsarlo se cambia el tiempo en el cual se puede ingresar al modo comando por el escrito en el cuadro de texto. Este valor debe ser entre 0 y 255, por ejemplo 0 = no se puede configurar, 60 = dentro de los primeros 60 segundos, 255 = todo el tiempo se puede entrar a el modo comando.
- **Botón “Salir”:** Al pulsarlo se cierra la interfaz de configuración.

Todos los botones están asociados al cuadro de texto respuesta ya que cada vez que se envía un comando en este cuadro aparece la respuesta del circuito RN-41, si se entra en modo comando se recibe “CMD”, si se sale del modo comando se recibe “END”, si el comando es correcto se recibe “AOK”, si el comando es erróneo se recibe “ERR” y si el comando no se reconoce se recibe “?”.

5.6 Cambios de Programación Requeridos en el Software de Estación Terrena y en el Subsistema de Computadora de Vuelo para Incorporar el Uso de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

El software de estación terrena para SATEDU es una interfaz de comunicación entre la computadora y SATEDU. Esta interfaz permite cargar el programa de funcionamiento del subsistema de computadora de vuelo para después mandarle comandos dirigidos a los diferentes subsistemas de SATEDU, así como capturar telemetría, para hacer esto la interfaz nos pide un puerto COM, figura 5.5.

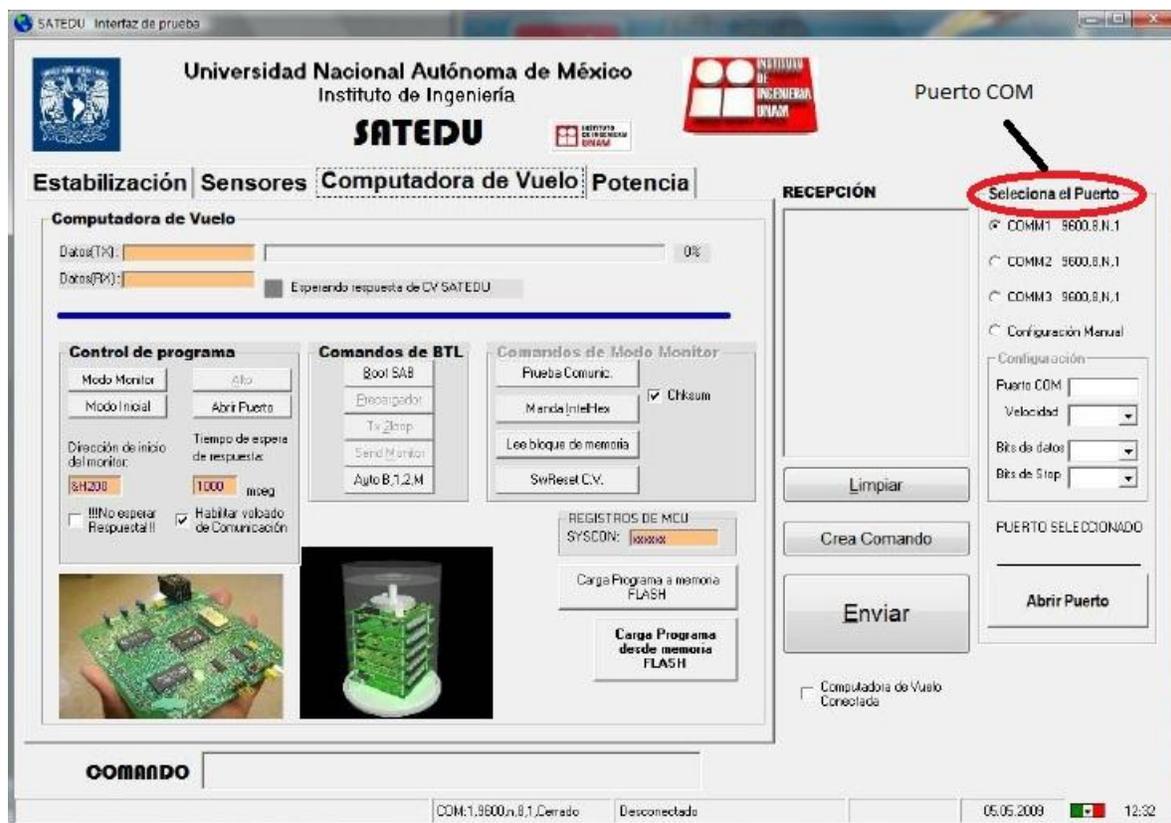


Figura 5.5 Software de Estación Terrena de SATEDU.

Debido a que la Tarjeta de Comunicaciones Inalámbricas Bluetooth también se asocia a la computadora por un puerto COM, no se tiene que hacer modificación ninguna al Software de Estación Terrena, solo se tiene que configurar con la interfaz de configuración desarrollada en este capítulo para no tener problemas de comunicación.

Capítulo 6

Pruebas de Validación Preliminares de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

6.1 Introducción

Una vez fabricada y armada la Tarjeta de Comunicaciones Inalámbricas Bluetooth y desarrollada la interfaz de configuración para la tarjeta se deben realizar pruebas que comprueben el buen funcionamiento de la tarjeta y cada uno de los componentes que la integran.

En este capítulo se describen las pruebas realizadas para la validación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth. Al realizar pruebas que garanticen el buen funcionamiento de la tarjeta diseñada se valida tanto el hardware como el software desarrollado en esta tesis.

Inicialmente se hicieron pruebas a la tarjeta por separado, para posteriormente integrarla a SATEDU para la realización de pruebas de validación finales.

6.2 Conexión de la Tarjeta de Comunicaciones Inalámbricas Bluetooth a la Computadora

Como se dijo en el capítulo 3 lo primero que hay que hacer para utilizar la tarjeta es vincularla con una computadora para esto se tiene que buscar la tarjeta con el software del bluetooth que esté integrado a la computadora para agregar este dispositivo.

Para energizar la Tarjeta de Comunicaciones Inalámbricas Bluetooth se utilizó un regulador LF33CV, figura 6.1, un par de caimanes y dos headers, una vez energizada la tarjeta se ve como empieza a parpadear el LED naranja, esto indica que está listo para vincularse con algún dispositivo.

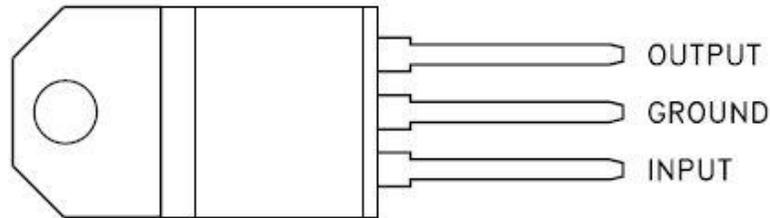


Figura 6.1 Regulador de 3.3 Volts LF33CV.

Desde la computadora se busca a la Tarjeta de Comunicaciones Inalámbricas Bluetooth el nombre del circuito RN-41 inicialmente es FireFly DDBA, figura 6.2.

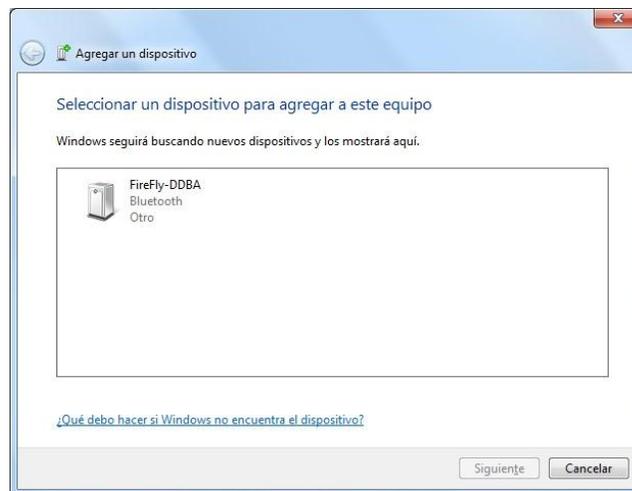


Figura 6.2 Nombre Inicial de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Después se debe de escoger el dispositivo y vincular la tarjeta escribiendo el código PIN del dispositivo, este código es 1234, figura 6.3.

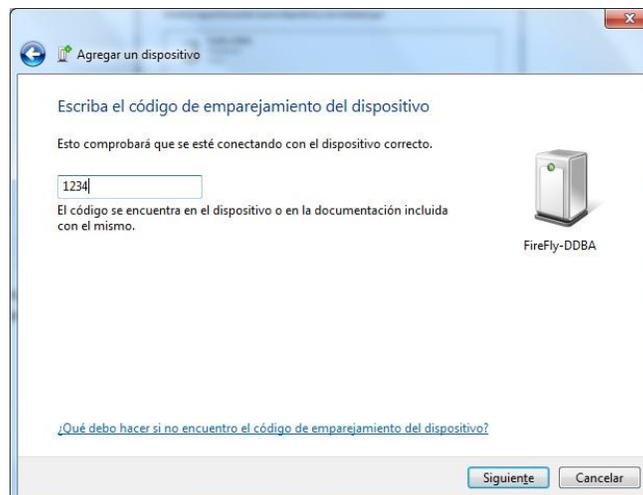


Figura 6.3 Código PIN de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Una vez que se agregó la tarjeta correctamente, figura 6.4, la computadora instalará los controladores y nos mostrará en la pantalla los puertos COM a los que se vinculó el dispositivo, figura 6.5, en este caso es al COM18 y COM 19.

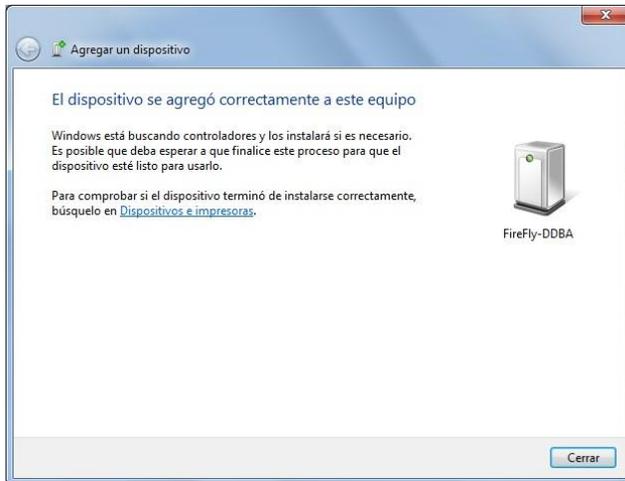


Figura 6.4 Dispositivo agregado correctamente.

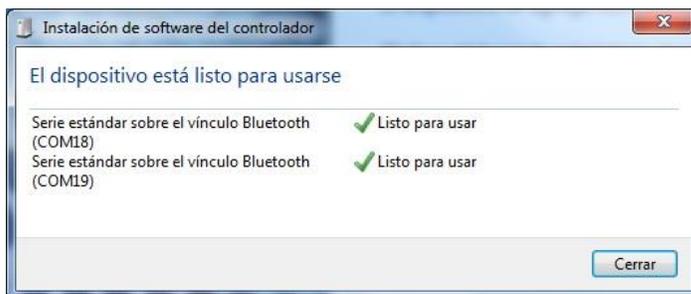


Figura 6.5 Puertos Asociados a la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Una vez que se agregó la Tarjeta de Comunicaciones Inalámbricas Bluetooth se pueden empezar las pruebas de validación.

6.3 Pruebas de Validación de la Interfaz de Configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

En el capítulo anterior se mostró el desarrollo de una interfaz gráfica, para validar el buen funcionamiento de esta interfaz se cambiará la configuración inicial de la tarjeta por una nueva configuración, tabla 6.1.

Opción	Configuración Inicial	Configuración Final
Nombre del Dispositivo	FireFly-DDBA	SATEDU-DDBA
Tasa de Transmisión	115200 bps	9600
Paridad	Ninguna	PAR

Bits de Datos	8bits	7bits
Bit de Parada	1 bit	1 bits
Autenticación	Deshabilitado	Habilitado
Encriptado	Deshabilitado	Habilitado
Código	1234	TCB1
Tiempo para entrar a la configuración	60 segundos	Todo el tiempo

Tabla 6.1 Prueba validación interfaz gráfica.

Primeramente, se selecciona el puerto COM18 con una velocidad de transmisión de 115000 bauds y se entra en el modo comando antes de que transcurran 60 segundos a partir de que se energiza la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

En el momento en que se abre el puerto el led rojo se prende y el naranja deja de parpadear, una vez que se entra al modo comando el led rojo permanece encendido y el led naranja empieza a parpadear.

Una vez en el modo comando primero se observa la configuración básica y extendida de la tarjeta, figura 6.6.

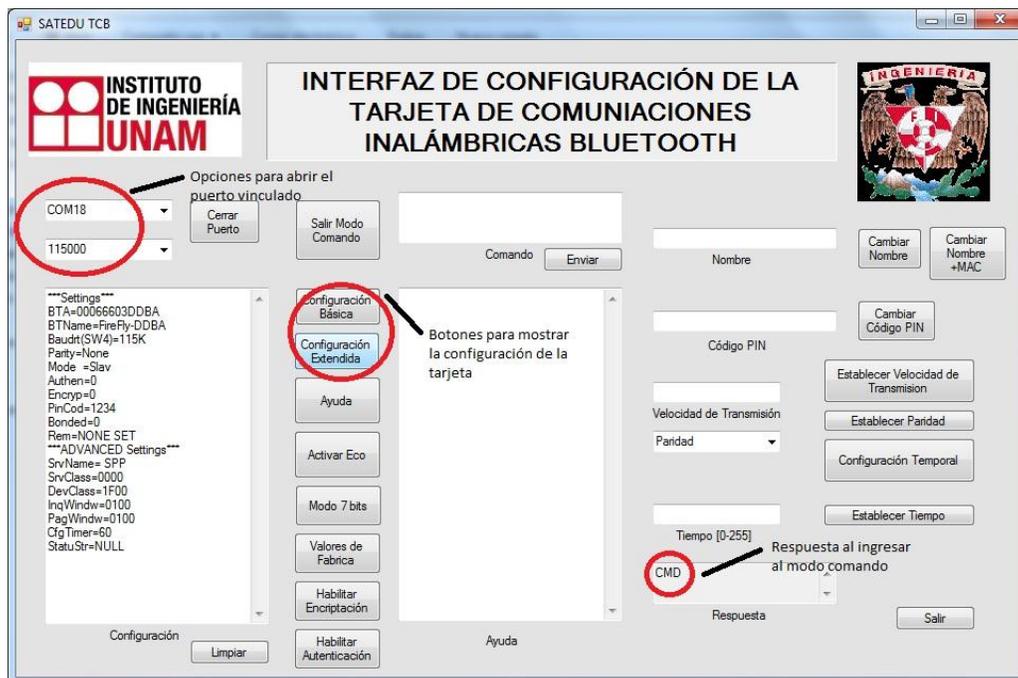


Figura 6.6 Configuración inicial del circuito RN-41.

En el cuadro de texto Configuración de la figura 6.6 se puede ver que la configuración inicial es la misma que la de la tabla 6.1, también en el cuadro de texto Respuesta se puede ver la respuesta del circuito al ingresar al modo comando.

Después se modifican los parámetros para llegar a la configuración final deseada, figura 6.7.

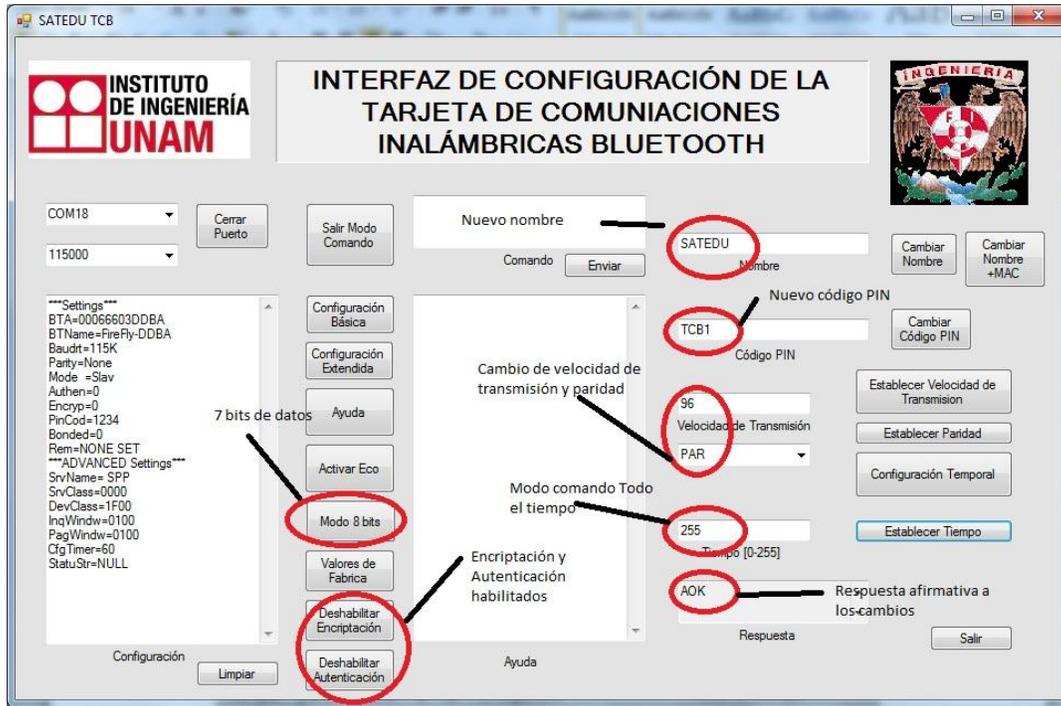


Figura 6.7 Modificación de parámetros por medio de la interfaz gráfica.

Finalmente se limpia el cuadro de texto Configuración y se obtiene la configuración básica y extendida de la tarjeta después de la modificación, figura 6.8.

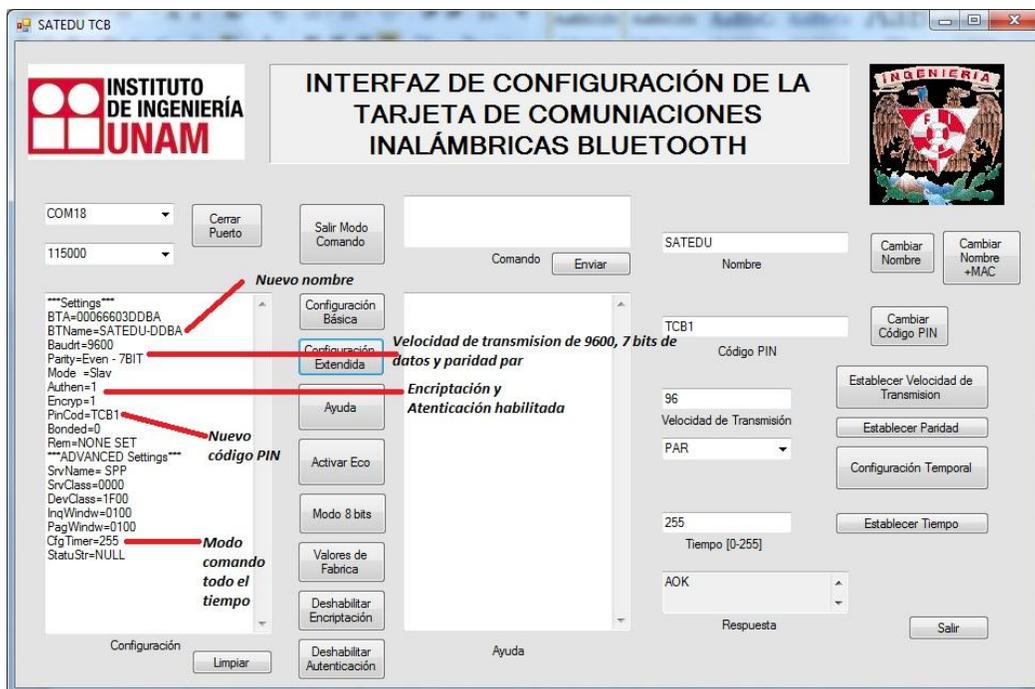


Figura 6.8 Configuración final del circuito RN-41.

Para comprobar la configuración del nombre se quita, el dispositivo vinculado y se vincula nuevamente, figura 6.9. Los nuevos puertos COM asociados al dispositivo son COM10 y COM14.

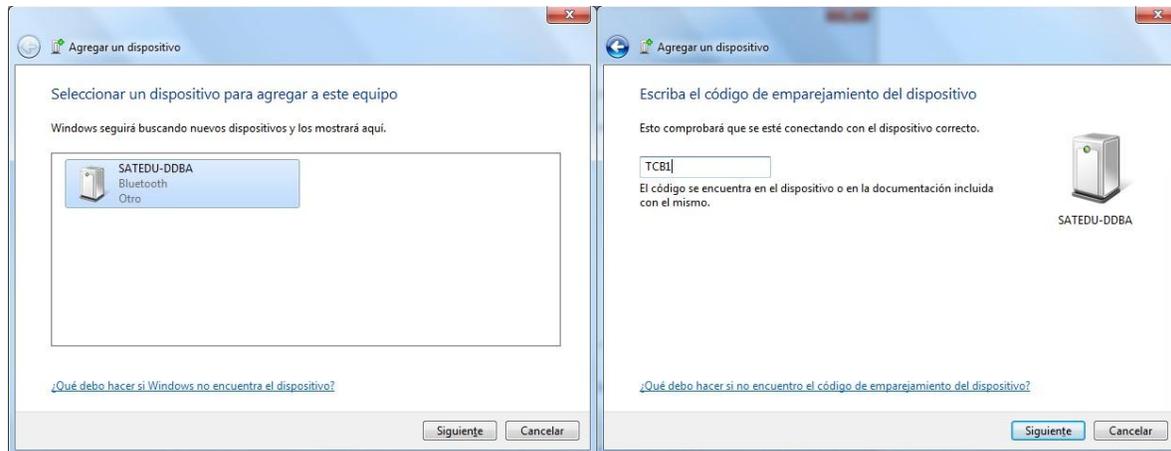


Figura 6.9 Nuevo nombre y código PIN de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

Con esta prueba quedó validada la interfaz gráfica para la configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth.

6.4 Pruebas Comunicación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth con un Microcontrolador

La Tarjeta de Comunicaciones Inalámbricas Bluetooth tiene comunicación por un lado con la computadora personal, dicha comunicación se validó con la prueba anterior, por otro lado la tarjeta se comunica con un microcontrolador en la tarjeta de computadora de vuelo, para validar la comunicación entre la tarjeta y un microcontrolador se desarrollaron dos pruebas utilizando protobord. Para estas pruebas se ocupó el microcontrolador PIC18F2321.

- La primer prueba consiste en mandar un carácter al PIC18F2321 a través de una interfaz gráfica más sencilla desarrollada para estas pruebas, una vez que el microcontrolador recibe el carácter hace un eco, es decir devuelve el carácter que le fue enviado, cada vez que recibe un carácter el microcontrolador prende y apaga un led.
- La segunda prueba consiste en mandar una cadena de caracteres al PIC18F2321, una vez que el microcontrolador reciba la información imprimirá la cadena de caracteres en un LCD y hará un eco, mientras el microcontrolador este energizado prenderá y apagará un led cada medio segundo aproximadamente.

Para el desarrollo del programa del PIC18F2321 para ambas pruebas se utilizó el programa MPLAB, este programa se utiliza para desarrollar aplicaciones para los microcontroladores de Microchip, en especial se utilizó el complemento MPLAB C18 que es un compilador de lenguaje C optimizado para microcontroladores de la familia PIC18. Los dos programas para realizar las pruebas se pueden ver en el apéndice B.

Para realizar las pruebas de validación se utilizó un monitor de puertos desarrollado con Visual C# de manera similar a la interfaz de configuración, este monitor de puertos cuenta con dos cuadros de opciones y un botón para seleccionar y abrir un puerto, un cuadro de texto y un botón para enviar datos y otro cuadro de texto con un botón asociado para ver y limpiar los datos recibidos, figura 6.10.



Figura 6.10 Monitor de Puertos.

Al realizar la primer prueba, figura 6.11, se conectó el receptor y el transmisor del puerto serie de la Tarjeta de Comunicaciones Inalámbricas Bluetooth al transmisor y receptor del puerto serie del PIC18F2321 por medio de un par de cables. Después con el monitor de puertos se abrió el puerto asociado a la tarjeta, que en este caso es el COM18 a 9600 bauds y se mandó carácter a carácter el abecedario y los números del 0 al 9 y algunos caracteres al azar.

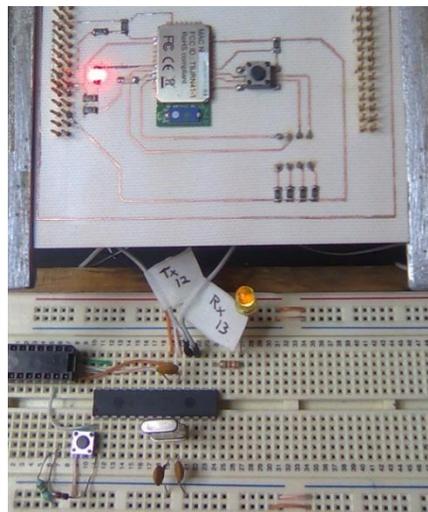


Figura 6.11 Primer prueba realizada en protobord.

Cada uno de los caracteres mandados se recibió en el cuadro de texto de Recepción, figura 6.12, y el led prendía y apagaba cada que se mandaba un carácter, con lo cual se concluye que se realizó con éxito la prueba número 1.



Figura 6.12 Interacción exitosa con un microcontrolador.

En la imagen de arriba se puede ver que todos los caracteres se recibieron y se reenviaron por el PIC18F2321 de manera exitosa excepto el carácter 'ñ', esto es porque el PIC trabaja con código ASCII, figura 6.13.



Figura 6.13 Caracteres ASCII.

Para la segunda prueba además de conectar la Tarjeta de Comunicaciones Inalámbricas Bluetooth se conectó un LCD de 2 x 16 al PIC18F2321 para visualizar en este los caracteres enviados desde el monitor de puertos a través de la tarjeta, figura 6.14.

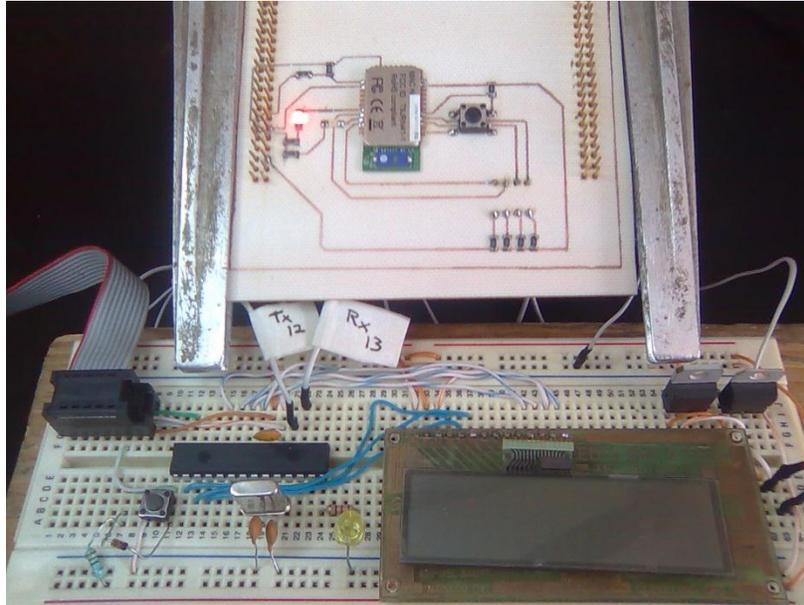


Figura 6.14 Segunda prueba realizada en protobord.

Se enviaron palabras desde el monitor de puertos vía bluetooth, una vez que estas palabras llegaron a la tarjeta está los envió al microcontrolador vía puerto serie, una vez en el microcontrolador se hizo un eco de la información y se imprimió en la primer línea del LCD, de tal forma que la palabra enviada se mostraba en el monitor de puertos y en el LCD, en la figura 6.15 se observa cómo se imprime en el LCD el mensaje “Hola Mundo”.

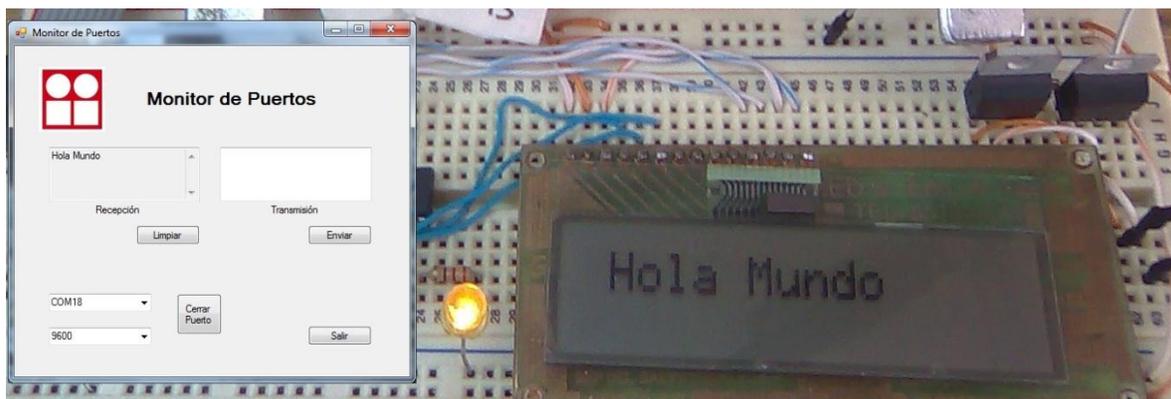


Figura 6.15 LCD y Monitor de Puertos con el mensaje “Hola Mundo”.

Adicionalmente en la realización de estas dos pruebas se probó el push button, y el DIP Switch colocados en la tarjeta, y ambos funcionaron de manera adecuada.

Al desarrollar las dos pruebas anteriores con éxito, se validó el funcionamiento de la nueva tarjeta para el Subsistema de Comunicaciones Inalámbricas, esta tarjeta se encuentra lista para pruebas de validación finales ya integrada en SATEDU, mismas que se desarrollaran en el siguiente capítulo.

Bibliografía:

- <http://msdn.microsoft.com/es-es/>
- Microchip Technology. "MPLAB C18 C CCOMPILER User's GUIDE", 2005.
- Microchip Technology. "MPLAB C18 C CCOMPILER GETTING STARTED", 2005.
- Microchip Technology. "MPLAB C18 C CCOMPILER LIBRARIES", 2005.
- Microchip Technology. "PIC18F4321 Family", Hoja de especificaciones, 2006.
- AND. "Intelligent Alphanumeric Application Notes", 1999.
- ST. "LF00 Series", Hoja de especificaciones, 2003.
- ST. "L78 Series", Hoja de especificaciones, 2003.

Capítulo 7

Pruebas de Validación Finales de la Tarjeta de Comunicaciones Inalámbricas Bluetooth con SATEDU

7.1 Introducción

En los capítulos anteriores se mostró el desarrollo de la Tarjeta de Comunicaciones Inalámbricas Bluetooth, iniciando con la búsqueda de los componentes electrónicos, pasando por el diseño y la fabricación del circuito impreso para el posterior montaje de componentes eléctricos y electrónicos y finalmente realizando pruebas preliminares de validación para verificar el buen funcionamiento de la nueva tarjeta del Subsistema de Comunicaciones Inalámbricas. Todo esto se realizó con el fin de dotar a SATEDU de comunicación inalámbrica vía bluetooth.

En este capítulo se integra la tarjeta desarrollada en esta tesis a SATEDU y se realizan pruebas que validen el funcionamiento de la tarjeta con comandos representativos generados por el software de Estación Terrena.

Con la validación desarrollada en este capítulo también se valida el remplazo del actual subsistema de comunicaciones de SATEDU por un subsistema de comunicaciones más barato, sencillo, de mayor alcance y con la posibilidad de conexión con teléfonos celulares, PDA's o cualquier dispositivo que tenga integrado un bluetooth y pueda soportar SPP (Serial Port Profile, perfil puerto serie).

7.2 Incorporación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth al conjunto de Tarjetas de SATEDU

La Tarjeta de Comunicaciones Inalámbricas Bluetooth es compatible con las demás tarjetas de SATEDU, cuenta con los conectores de costilla para conectarse a cualquier tarjeta de SATEDU.

La primer prueba de validación final fue agregar la Tarjeta de Comunicaciones Inalámbricas Bluetooth al "sándwich" del simulador satelital. Las tarjetas de SATEDU se apilan por medio de los conectores de costilla, cabe señalar que ninguna de ellas tiene un lugar fijo excepto por la tarjeta

del Subsistema de Estabilización que por su tamaño siempre se coloca hasta arriba, esto es por la rueda inercial que contiene. Al integrar la tarjeta Bluetooth a SATEDU, figura 7.1, se cuidó que el conector izquierdo fuera conectado con el conector izquierdo y el conector derecho con el derecho, dado que el conectarlo al revés podría provocar cortos y dañar a SATEDU de manera permanente.

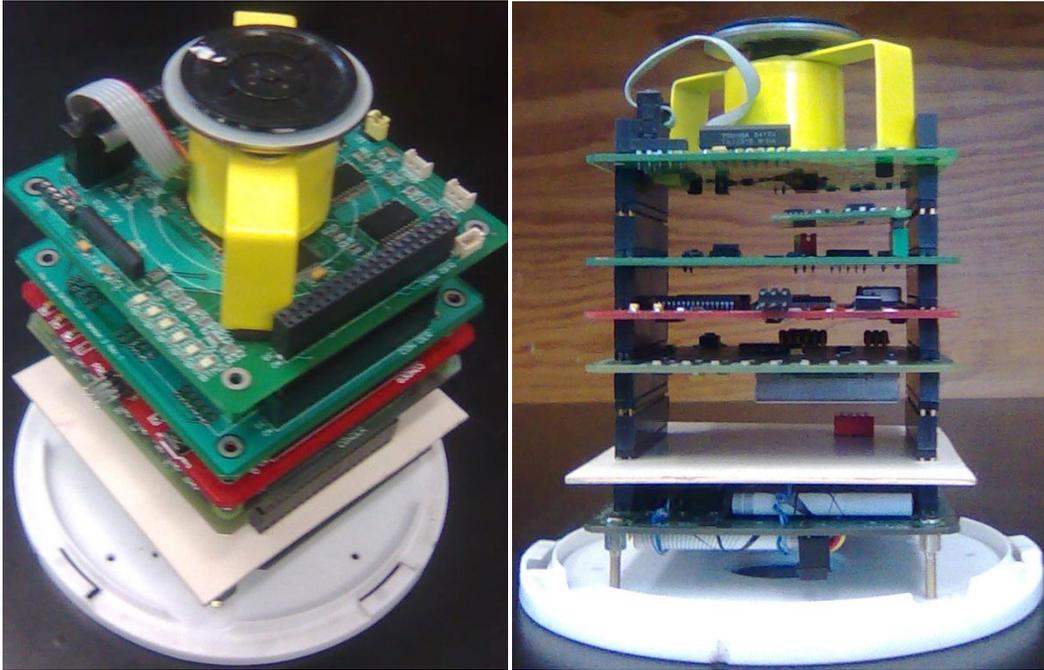


Figura 7.1 Incorporación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth a SATEDU.

La tarjeta no tuvo problemas al integrarse con las demás tarjetas que componen al simulador satelital SATEDU, la única observación es que el impreso se realizó en una placa fenólica cuadrada de diez centímetros por lado y no en una placa con las medidas exactas de las tarjetas de SATEDU, y tampoco cuenta con los orificios para atornillar a la base de la estructura de SATEDU.

Estos pequeños detalles se resolvieron con el mismo taladro con el cual se hicieron los orificios a la tarjeta, se cortó las orillas sobrantes de la tarjeta de tal manera que quedara del tamaño exacto para caber en la estructura de SATEDU, en cuanto a los orificios para colocar tornillos en la base del sándwich, no son indispensables ya que la base de SATEDU por convención es la tarjeta de baterías del Subsistema de Potencia y es ésta la que se atornilla a la base de la estructura.

7.3 Pruebas de Validación Operativa Entre la Tarjeta de Comunicaciones Inalámbricas Bluetooth y la Tarjeta de Computadora de Vuelo

El Subsistema de Computadora de Vuelo es el encargado de controlar todas las funciones que realiza SATEDU, como el apagado y encendido de los subsistemas, la asignación de tareas, la recepción de comandos desde la computadora que funge como Estación Terrena y el envío de telemetría a la Estación Terrena.

La principal tarea a realizar con la computadora de vuelo es el cargado de nuevo programa, ya que sin este no puede operar SATEDU, el SAB guarda el programa de operación en una memoria RAM y cada vez que SATEDU se apaga el programa guardado en la memoria RAM se borra y cuando se vuelve encender se tiene que cargar otra vez. Como se puede apreciar, esta tarea resulta indispensable para la operación de SATEDU, ya que este es el primer paso para poder utilizarlo.

Para el cargado del nuevo programa lo primero que se hizo fue energizar a SATEDU para después vincularlo con la computadora que fungirá como estación terrena. Este vínculo se realizó de la misma manera que se documentó en el capítulo anterior, si SATEDU ya está vinculado y no recordamos el puerto al cual está vinculado, esto lo podemos ver en el *Panel de Control* de Windows en el *Administrador de Dispositivos*. Al energizar a SATEDU se observa cómo cada uno de los Subsistemas se activa, esto se ve por medio de leds en las tarjetas, figura 7.2.

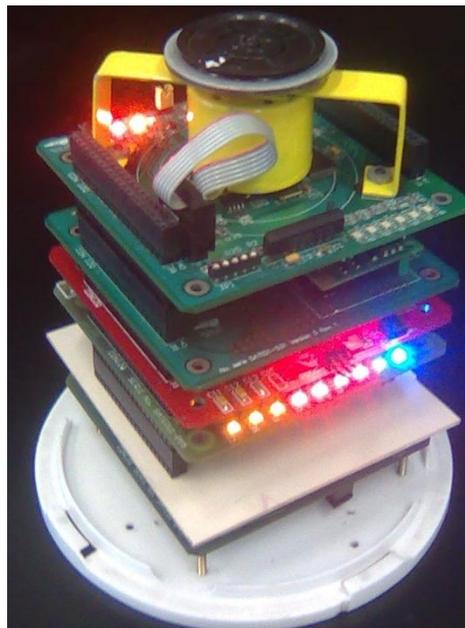


Figura 7.2 Energización del Simulador Satelital SATEDU.

Cabe mencionar que para utilizar la interface de Estación Terrena no es necesario que todos los Subsistemas que componen a SATEDU estén conectados, bastaría con la Computadora de Vuelo, las tarjetas de Potencia y el Subsistema de Comunicaciones Inalámbricas, figura 7.3.

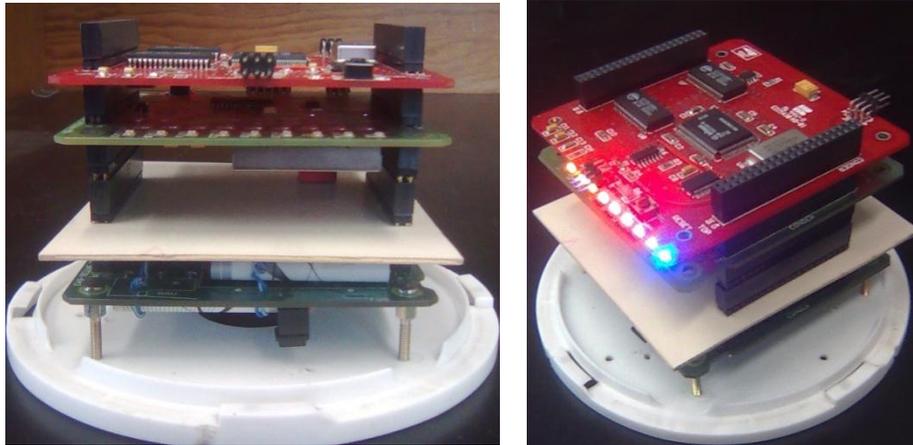


Figura 7.3 Subsistemas básicos para utilizar el software de Estación Terrena.

Una vez vinculado SATEDU a la Estación Terrena se configuró el canal de comunicación con la interfaz gráfica desarrollada en el capítulo 5, el canal se configuró con las siguientes características, figura 7.4:

- 8 bits de datos.
- 1 bits de stop.
- Sin paridad.
- Velocidad de Transmisión de 9600 bauds.



Figura 7.4 Configuración del Canal de Comunicaciones.

Una vez configurada la Tarjeta de Comunicaciones Inalámbricas Bluetooth, se abrió el software de Estación Terrena por medio del programa Visual Basic 6 y se configuró el puerto COM asociado a la tarjeta en la vinculación con la misma configuración que el Subsistema de Comunicaciones Inalámbricas y se abrió el puerto, figura 7.5.

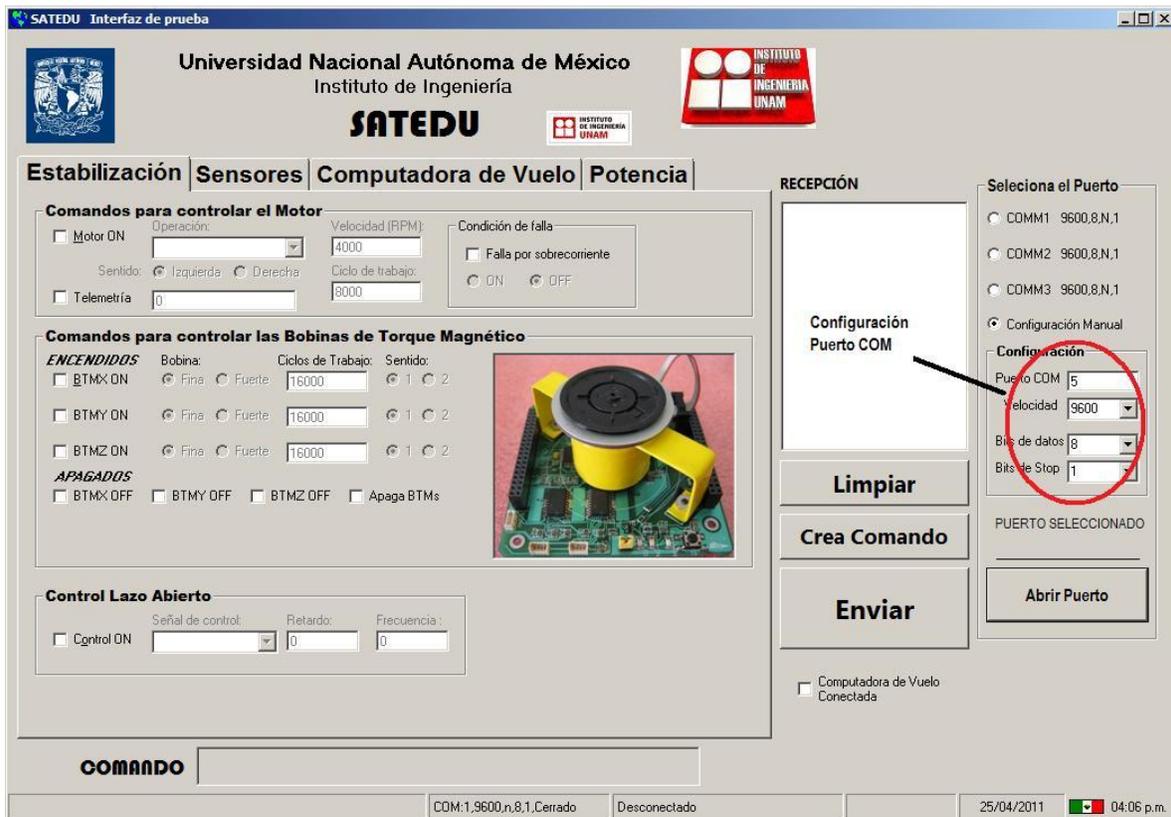


Figura 7.5 Configuración del puerto en la Estación Terrena.

Después se escogió la pestaña de “Computadora de Vuelo” en el software de Estación Terrena, una vez en esta pestaña se siguieron los siguientes pasos:

1. En la parte de “Comandos BT”, se deben de accionar uno por uno los botones “Boot SAB”, “Precargador”, “TX 2loop”, “Send Monitor”.
2. Después en la parte de “Control de Programa” se debe de accionar el botón “Modo Monitor”, después de esto el programa avisará que el microcontrolador se encuentra en modo monitor y se activará la parte de “Comandos Modo Monitor”.
3. En la parte de “Comandos Modo Monitor” primero se debe accionar el botón “Prueba de Comunic” y después se acciona el botón “Manda IntelHex” para enviar el programa de operación, una vez terminado el cargado del programa se verá un aviso.

- Para finalizar en la parte “Comandos Modo Monitor” se acciona el botón “SWReset CV”.

En la parte de “Computadora de Vuelo” se puede observar el proceso de cargado mediante una barra de estado progresiva. En la figura 7.6 se pueden ver las diferentes partes de la pestaña de “Computadora de Vuelo”.

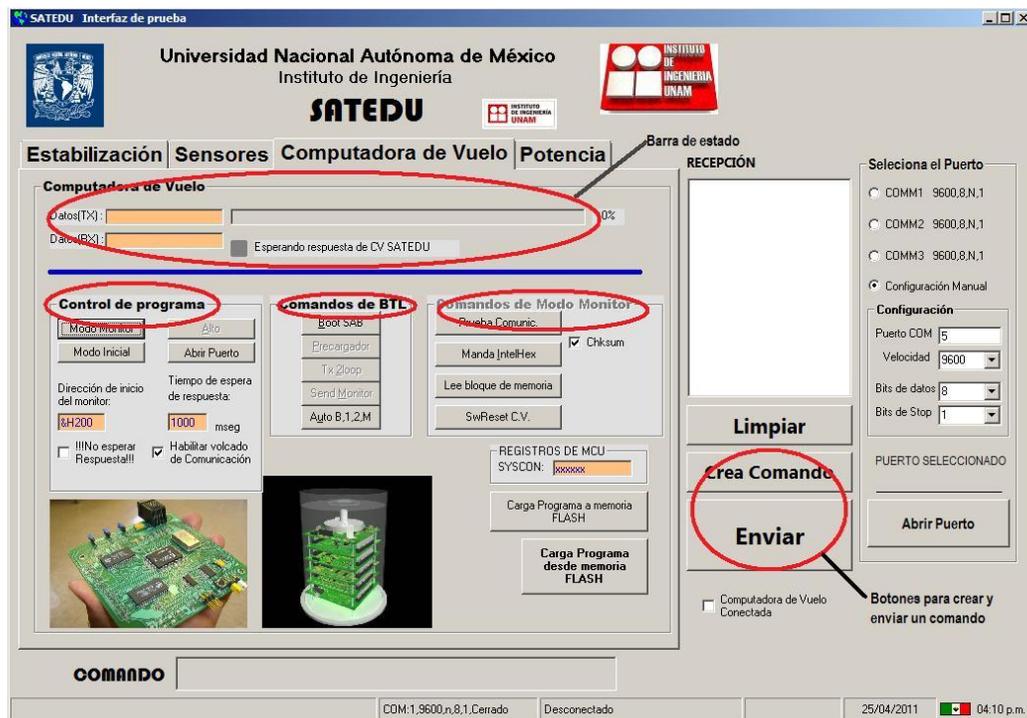


Figura 7.6 Pestaña de “Computadora de Vuelo” en el Software de Estación Terrena.

El programa se ejecuta inmediatamente después de ser cargado y de esta manera es posible empezar a mandar comandos a SATEDU. Esta prueba se desarrolló con éxito.

7.4 Pruebas de Validación Operando la Rueda Inercial de SATEDU

La estabilización activa de SATEDU se puede hacer por dos métodos, uno por medio de las bobinas de torque magnético (actualmente en desarrollo) y el otro por medio de una rueda inercial. La rueda inercial es una pequeña masa circular sujeta a un motor, por medio de esta se pueden realizar experimentos de visualización inmediata.

Un buen ejemplo de comando para mandar a SATEDU es mover la rueda inercial tanto a la izquierda como a la derecha ya que este comando se puede visualizar de manera inmediata al ver el giro de la rueda inercial de SATEDU.

Para hacer funcionar la rueda inercial, primero se debe hacer el cargado del programa de operación de SATEDU en la tarjeta de Computadora de Vuelo como ya se comentó anteriormente, después en el software de Estación Terrena se debe de ir a la pestaña de “Estabilización” y seguir los siguientes pasos en la parte de “Comandos para controlar el Motor”:

1. Se debe de activar la casilla de “Motor ON”.
2. Después en el cuadro de opciones “Operación” se debe elegir la opción “Fijar CT a motor” y se tiene que poner un número en el cuadro de texto “Ciclo de Trabajo”, para esta prueba se pondrá el número 1000.
3. Después se debe de elegir el sentido de giro activando cualquier casilla ya sea “Izquierda” o “Derecha”.
4. Después se acciona el botón “Crea Comando” que se encuentra abajo del cuadro de texto “RECEPCIÓN”, una vez creado el comando aparecerá en la parte inferior de la ventana en el cuadro de texto “COMANDO”.
5. Para finalizar se acciona el botón “Enviar” localizado debajo del botón “Crear comando”, y en el cuadro de texto “Recepción” se observará si todo se ejecutó de manera apropiada.

Al realizar esta serie de pasos se vio como empezaba a girar el motor en el sentido que se le había programado, por lo que se concluyó que se realizó con éxito el envío del comando a SATEDU. En la figura 7.7 se muestra como se realizó la configuración en el software de Estación Terrena.

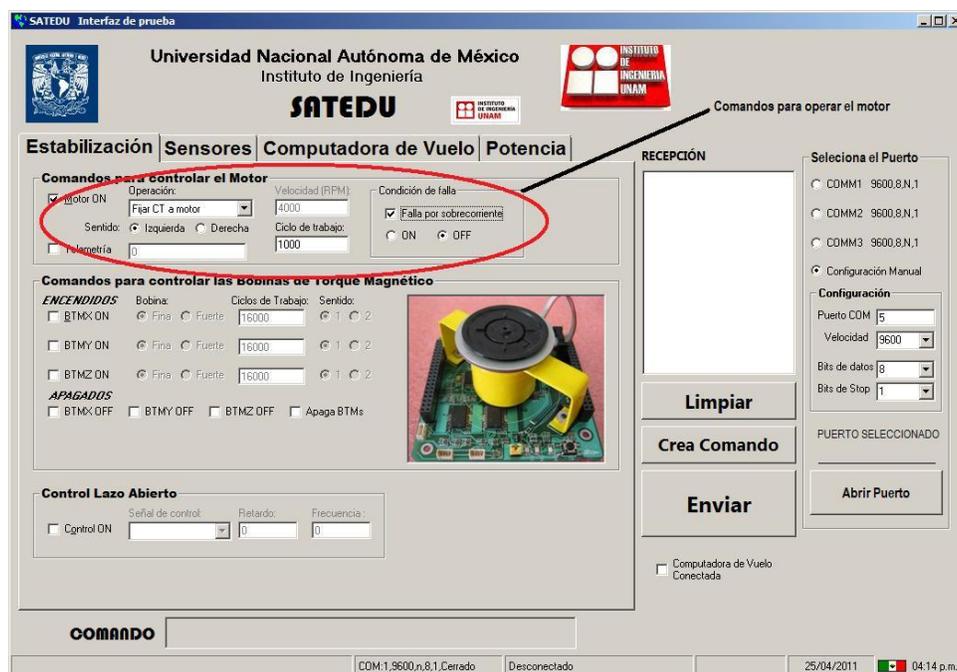


Figura 7.7 Pestaña de “Estabilización” en el Software de Estación Terrena.

En el desarrollo de esta prueba se tomaron ciertas precauciones, como no dar un ciclo de trabajo muy grande ya que esto puede ocasionar que se presente un pico de corriente activando la protección por sobre corriente que apaga el motor, para que el motor no se apague se debe de ir incrementando poco a poco el ciclo de trabajo o bien desactivando la protección mediante un comando en la casilla de "Condición de falla".

7.5 Elaboración de Material de Video Demostrativo de Operación de la Tarjeta de Comunicaciones Inalámbricas Bluetooth con SATEDU Para Uso del Instituto de Ingeniería, UNAM

Durante el desarrollo de las pruebas finales de validación se tomaron videos donde se demuestran las pruebas realizadas, estos videos serán utilizados por el Instituto de Ingeniería de la UNAM en conferencias y exposiciones a fin de mostrar el funcionamiento del Simulador Satelital SATEDU.

Bibliografía:

- Dr. Esaú Vicente Vivas, "Sistema portátil para entrenamiento de recursos humanos en tecnología de satélites pequeños primera fase". Proyecto 52297 de CONACYT, Instituto de Ingeniería, UNAM. México D.F. Mayo de 2008.

Capítulo 8

Conclusiones y Recomendaciones

8.1 Conclusiones

En esta tesis se diseñó, fabricó, ensambló y validó un sistema de comunicaciones versátil que utiliza el estándar bluetooth. Se pudieron enfrentar y resolver con éxito todos los problemas presentados durante el desarrollo de esta tesis. Al concluir el desarrollo de esta tesis se realizó de manera exitosa la actualización del protocolo de comunicaciones por el cual SATEDU se comunica y realiza transferencia de información con la computadora que funge como Estación Terrena.

Se fabricó también un circuito impreso totalmente compatible con el diseño de las tarjetas actualmente existentes en el simulador satelital SATEDU, el cual es capaz de sustituir a las tarjetas de RF del antiguo Subsistema de Comunicaciones Inalámbricas, ya que puede realizar todas las tareas que anteriormente realizaban dos tarjetas de RF.

El nuevo Subsistema de Comunicaciones Inalámbricas emplea solamente una tarjeta, esto reduce el costo del subsistema significativamente, ya que lo más caro de este subsistema es la fabricación del circuito impreso. Ahora cada dispositivo que requiera fungir como Estación Terrena de SATEDU solo necesita tener integrado un bluetooth ya interno o externo y ya no se necesita de una tarjeta especial propietaria como el anterior Subsistema de Comunicaciones Inalámbricas.

Con la integración de esta tarjeta a SATEDU adquirió una conectividad mayor, ya que ahora se puede conectar con cualquier dispositivo que tenga bluetooth y soporte SPP. Lo que abre la puerta a nuevos desarrollos de experimentos con la utilización de teléfonos celulares y PDA's.

El nuevo sistema de comunicaciones de SATEDU le proporciona simplicidad de conexión sin dejar de lado la seguridad del sistema ya que para conectarse con SATEDU se requiere del código PIN y este se puede cambiar para hacer tan complejo el acceso como se quiera.

El protocolo de comunicaciones Bluetooth es muy usado en redes de área personal, el desarrollo de esta tarjeta también le da la oportunidad a SATEDU de participar en una piconet con la cual se podría desarrollar comunicación entre dos simuladores satelitales SATEDU.

8.2 Recomendaciones

En un futuro cercano se piensa actualizar el BUS de SATEDU de un protocolo por puerto serie a el protocolo I²C, para hacer esto se necesitaría agregar un microcontrolador a la Tarjeta de Comunicaciones Inalámbricas Bluetooth, se recomienda que se le agregue el PIC18F2321, ya que es fácil de utilizar y cuenta con un puerto serie y un puerto I²C disponible y la conexión es bastante sencilla como se mostró en el desarrollo de las pruebas de validación preliminares.

El diseño del circuito impreso, se realizó en una solo capa dado que se iba a fabricar inicialmente un prototipo de manera artesanal pero al mandarse a fabricar se recomienda hacer el impreso en dos capas, de esta manera se ahorraría mucho espacio y el diseño sería más eficiente, aparte de que el botón de reset manual se podría colocar a la orilla del circuito impreso. Adicionalmente, se recomienda que en la misma tarjeta que se vaya a desarrollar se incorporen ambos sistemas de comunicación, por RF y por Bluetooth (anterior y nuevo) para aprovechar el espacio disponible en el PCB y para poder ofrecer más posibilidades didácticas de SATEDU a sus usuarios en términos de comunicaciones inalámbricas.

En la tarjeta hay mucho espacio libre mismo que se puede ocupar para agregar algún otro desarrollo en la misma tarjeta, antes de mandarla a fabricar. Este desarrollo podría ser uno de los dos motores que se piensa agregar a SATEDU, pero para hacer esto se necesita idear otro diseño que no incluya los conectores de costilla e incluya otra forma de conectarse al BUS de SATEDU.

Los led's incluidos en la tarjeta son para mostrar visualmente que está funcionando pero si en algún momento se desea ahorrar más energía se podrían quitar o agregar jumper's para decidir en qué momento se utilizan.

En el momento de configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth se recomienda tener cuidado al cambiar el código PIN de ésta ya que si este es olvidado o cambiado en el momento de alguna operación se perdería el contacto con SATEDU y se tendrían que restablecer los valores de fábrica de manera manual por medio de hardware para poder intercambiar información con SATEDU.

Apéndice A

Código de las Interfaces de Usuario

Desarrolladas en Microsoft Visual Studio 2010

Interfaz de Configuración de la Tarjeta de Comunicaciones Inalámbricas Bluetooth

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;

namespace InterfazSATEDUTCIB
{
    public partial class SATEDUTCIB : Form
    {
        int opcion = 0; //variable de seleccion
        char c = "\r"; //variable retorno de carro
        public SATEDUTCIB()
        {
            InitializeComponent();
        }

        private void SATEDUTCIB_Load(object sender, EventArgs e)
        {
            puerto.SelectedIndex = 0; //seleccion de la opcion cero del combobox1
            baudrate.SelectedIndex = 0; //seleccion de la opcion cero del combobox2
            paridad.SelectedIndex = 0; //seleccion de la opcion cero del combobox3
            foreach (String s in SerialPort.GetPortNames()) //obtencion de los puertos COM disponibles
            {
                puerto.Items.Add(s);
            }
            botonmc.Enabled = false; //deshabilita el boton modo comando
        }
    }
}
```

```

botonenviar.Enabled = false;//deshabilita el boton enviar
comandotex.Enabled = false;//deshabilita el cuadro de texto de comandos
botoneco.Enabled = false;//deshabilita el boton activar eco
botonayuda.Enabled = false;//deshabilita el boton ayuda
botonec.Enabled = false;//deshabilita el boton configuracion extendida
botoncn.Enabled = false;//deshabilita el boton cambiar nombre
botoncne.Enabled = false;//deshabilita el boton cambiar nombre +mac
tbnombre.Enabled = false;//deshabilita el cuadro de texto del nombre
botoncp.Enabled = false;//deshabilita el boton de cambiar codigo pin
tbcp.Enabled = false;//deshabilita el cuadro de texto cambiar codigo pin
botonmb.Enabled = false;//deshabilita el boton modo 7 bits
botonvf.Enabled = false;//deshabilita el boton restablecer valores de fabrica
botonvt.Enabled = false;//deshabilita el boton establecer velocidad de transmision
botonp.Enabled = false;//deshabilita el boton establecer paridad
paridad.Enabled = false;//deshabilita el combobox4
veltrans.Enabled = false;//deshabilita el cuadro de texto de baud rate
botonhe.Enabled = false;//deshabilita el boton habilita encriptacion
botonha.Enabled = false;//deshabilita el boton habilita autentificacion
botontc.Enabled = false;//deshabilita el boton establecer tiempo
tbtc.Enabled = false;//deshabilita el cuadro de texto de tiempo
botoncm.Enabled = false;//deshabilita el boton de configuracion temporal
botoncb.Enabled = false;//deshabilita el boton de configuracion basica
botonlimpiar.Enabled = false;//deshabilita el boton de limpiar
}

private void botonAC_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == false)//si el puerto esta cerrado
    {
        serialPort1.PortName = puerto.Text;//nombre del puerto seleccionado
        serialPort1.BaudRate = int.Parse(baudrate.Text);//velocidad de transmision seleccionada
        serialPort1.Open();//abre el puerto
        botonAC.Text = ("Cerrar Puerto");//cambia el nombre del boton
        botonmc.Enabled = true;//habilita el boton modo comando
        botonenviar.Enabled = true;//habilita el boton enviar
        comandotex.Enabled = true;//habilita el textbox para enviar otros comandos o datos
    }
    else
    {
        serialPort1.Close();//cierra el puerto
        botonAC.Text = ("Abrir Puerto");//cambia el nombre del boton
        botonmc.Enabled = false;//deshabilita el boton modo comando
        botonenviar.Enabled = false;//deshabilita el boton enviar
        comandotex.Enabled = false;//deshabilita el cuadro de texto de comandos
        botoneco.Enabled = false;//deshabilita el boton activar eco
        botonayuda.Enabled = false;//deshabilita el boton ayuda
        botonec.Enabled = false;//deshabilita el boton configuracion extendida
        botoncn.Enabled = false;//deshabilita el boton cambiar nombre
        botoncne.Enabled = false;//deshabilita el boton cambiar nombre +mac
        tbnombre.Enabled = false;//deshabilita el cuadro de texto del nombre
        botoncp.Enabled = false;//deshabilita el boton de cambiar codigo pin
        tbcp.Enabled = false;//deshabilita el cuadro de texto cambiar codigo pin
        botonmb.Enabled = false;//deshabilita el boton modo 7 bits
        botonvf.Enabled = false;//deshabilita el boton restablecer valores de fabrica
        botonvt.Enabled = false;//deshabilita el boton establecer velocidad de transmision
        botonp.Enabled = false;//deshabilita el boton establecer paridad
    }
}

```

```

    paridad.Enabled = false;//deshabilita el combobox4
    veltrans.Enabled = false;//deshabilita el cuadro de texto de baud rate
    botonhe.Enabled = false;//deshabilita el boton habilita encriptacion
    botonha.Enabled = false;//deshabilita el boton habilita autentificacion
    botontc.Enabled = false;//deshabilita el boton establecer tiempo
    tbtc.Enabled = false;//deshabilita el cuadro de texto de tiempo
    botoncm.Enabled = false;//deshabilita el boton de configuracion temporal
    botoncb.Enabled = false;//deshabilita el boton de configuracion basica
    botonlimpiar.Enabled = false;//deshabilita el boton de limpiar
}
}

private void botonsalir_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true)//cierra el puerto si esta abierto
    {
        serialPort1.Close();
    }
    Close();//cierra la aplicacion
}

private void botonmc_Click(object sender, EventArgs e)
{
    opcion = 0;//imprime en respuesta
    if (botonmc.Text == "Modo Comando")
    {
        serialPort1.Write("$$$");//manda tres signos de peso por el puerto serie
        botonmc.Text = "Salir Modo Comando";//cambia el nombre del boton
        botoneco.Enabled = true;//habilita el boton activar eco
        botonayuda.Enabled = true;//habilita el boton ayuda
        botonec.Enabled = true;//habilita el boton configuracion extendida
        botoncn.Enabled = true;//habilita el boton cambiar nombre
        botoncne.Enabled = true;//habilita el boton cambiar nombre +mac
        tbnombre.Enabled = true;//habilita el cuadro de texto del nombre
        botoncp.Enabled = true;//habilita el boton de cambiar codigo pin
        tbcpc.Enabled = true;//habilita el cuadro de texto cambiar codigo pin
        botonmb.Enabled = true;//habilita el boton modo 7 bits
        botonvf.Enabled = true;//habilita el boton restablecer valores de fabrica
        botonvt.Enabled = true;//habilita el boton establecer velocidad de transmision
        botonp.Enabled = true;//habilita el boton establecer paridad
        paridad.Enabled = true;//habilita el combobox4
        veltrans.Enabled = true;//habilita el cuadro de texto de baud rate
        botonhe.Enabled = true;//habilita el boton habilita encriptacion
        botonha.Enabled = true;//habilita el boton habilita autentificacion
        botontc.Enabled = true;//habilita el boton establecer tiempo
        tbtc.Enabled = true;//habilita el cuadro de texto de tiempo
        botoncm.Enabled = true;//habilita el boton de configuracion temporal
        botoncb.Enabled = true;//habilita el boton configuracion basica
        botonlimpiar.Enabled = true;//habilita el boton limpiar
    }
    else
    {
        serialPort1.Write("---" + c);//manda tres menos para salir del modo comando
        botonmc.Text = "Modo Comando";//cambia el nombre del boton
        botoneco.Enabled = false;//deshabilita el boton activar eco
        botoneco.Enabled = false;//deshabilita el boton activar eco
    }
}

```

```

        botonayuda.Enabled = false;//deshabilita el boton ayuda
        botonec.Enabled = false;//deshabilita el boton configuracion extendida
        botoncn.Enabled = false;//deshabilita el boton cambiar nombre
        botoncne.Enabled = false;//deshabilita el boton cambiar nombre +mac
        tbnombre.Enabled = false;//deshabilita el cuadro de texto del nombre
        botoncp.Enabled = false;//deshabilita el boton de cambiar codigo pin
        tcbcp.Enabled = false;//deshabilita el cuadro de texto cambiar codigo pin
        botonmb.Enabled = false;//deshabilita el boton modo 7 bits
        botonvf.Enabled = false;//deshabilita el boton restablecer valores de fabrica
        botonvt.Enabled = false;//deshabilita el boton establecer velocidad de transmision
        botonp.Enabled = false;//deshabilita el boton establecer paridad
        paridad.Enabled = false;//deshabilita el combobox4
        veltrans.Enabled = false;//deshabilita el cuadro de texto de baud rate
        botonhe.Enabled = false;//deshabilita el boton habilita encriptacion
        botonha.Enabled = false;//deshabilita el boton habilita autentificacion
        botontc.Enabled = false;//deshabilita el boton establecer tiempo
        tbtc.Enabled = false;//deshabilita el cuadro de texto de tiempo
        botoncm.Enabled = false;//deshabilita el boton de configuracion temporal
        botoncb.Enabled = false;//deshabilita el boton configuracion basica
        botonlimpiar.Enabled = false;//deshabilita el boton limmpiar
    }
}

private void botonenviar_Click(object sender, EventArgs e)
{
    String comando;//variable para el comando extra
    opcion = 0;//imprime en respuesta
    comando = comandotex.Text;//comando extra
    serialPort1.Write(comando + c);//manda el comando extra
    comandotex.Clear();//limpia el textbox despues de enviar
}

private void botoneco_Click(object sender, EventArgs e)
{
    opcion = 0;//imprime en respuesta
    if (botoneco.Text == "Activar Eco")
    {
        serialPort1.Write("+" + c);//manda comando para activar eco
        botoneco.Text = "Desactivar Eco";//cambia nombre del boton
    }
    else
    {
        serialPort1.Write("+" + c);//manda comando para activar eco
        botoneco.Text = "Activar Eco";//cambia nombre del boton
    }
}

private void botonayuda_Click(object sender, EventArgs e)
{
    opcion = 1;//opcion 1
    serialPort1.Write("H" + c);//envia el comando para la ayuda
}

private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    String mensaje;//variable mensaje
    mensaje = serialPort1.ReadExisting();//lee el mensaje proveniente de la tarjeta
}

```

```

if (opcion == 0)
{
    textboxeco.AppendText(mensaje);//imprime mensaje en el cuadro de texto respuesta
}
if (opcion == 1)
{
    tbayuda.AppendText(mensaje);//imprime en el cuadro de texto de ayuda
}
if (opcion == 2)
{
    tbedc.AppendText(mensaje);//imprime en el cuadro de texto de configuracion
}
}

private void botonec_Click(object sender, EventArgs e)
{
    opcion = 2;//opcion 2
    serialPort1.Write("E" + c);//envia el comando para el estado de la configuracion
}

private void botoncn_Click(object sender, EventArgs e)
{
    String nombre;//variable nombre
    opcion = 0;//opcion 0
    nombre = tnombre.Text;//nuevo nombre
    serialPort1.Write("SN," + nombre + c);//cambio del nombre de la tarjeta
}

private void botoncne_Click(object sender, EventArgs e)
{
    String nombre;//variable nombre
    opcion = 0;//opcion 0
    nombre = tnombre.Text;//nuevo nombre
    serialPort1.Write("S-" + nombre + c);//cambio del nombre de la tarjeta mas ultimos 4 numeros de la direccion mac
}

private void botoncp_Click(object sender, EventArgs e)
{
    String codigo;//variable codigo
    opcion = 0;//opcion 0
    codigo = tcbp.Text;//nuevo codigo
    serialPort1.Write("SP," + codigo + c);//cambia el codigo PIN
}

private void botonmb_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (botonmb.Text == "Modo 7 bits")
    {
        serialPort1.Write("S7,1" + c);//manda comando modo 7 bits
        botonmb.Text = "Modo 8 bits";//cambia el nombre del boton
    }
    else
    {
        serialPort1.Write("S7,0" + c);//entra en modo de 8 bits
        botonmb.Text = "Modo 7 bits";//cambia el nombre del boton
    }
}

```

```

}
private void botonvf_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    serialPort1.Write("SF,1" + c);//restablece los valores de fabrica
}
private void botonp_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (paridad.Text == "PAR")
    {
        serialPort1.Write("SL,E" + c);//manda la paridad seleccionada
    }
    if (paridad.Text == "IMPAR")
    {
        serialPort1.Write("SL,O" + c);//manda la paridad seleccionada
    }
    if (paridad.Text == "NINGUNA")
    {
        serialPort1.Write("SL,N" + c);//manda la paridad seleccionada
    }
}
private void botonvt_Click(object sender, EventArgs e)
{
    int velocidad;//variable velocidad
    opcion = 0;//opcion 0
    velocidad = int.Parse(veltrans.Text);//guarda velocidad de transmision
    serialPort1.Write("SU," + velocidad + c);//envia velocidad de transmision
}

private void botonhe_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (botonhe.Text == "Habilitar Encriptación")
    {
        serialPort1.Write("SE,1" + c);//habilita encriptacion
        botonhe.Text = "Deshabilitar Encriptación";//cambia nombre del boton
    }
    else if (botonhe.Text == "Deshabilitar Encriptación")
    {
        serialPort1.Write("SE,0" + c);//deshabilita encriptacion
        botonhe.Text = "Habilitar Encriptación";//cambia nombre del boton
    }
}

private void botonha_Click(object sender, EventArgs e)
{
    opcion = 0;//opcion 0
    if (botonha.Text == "Habilitar Autenticación")
    {
        serialPort1.Write("SA,1" + c);//habilita autenticacion
        botonha.Text = "Deshabilitar Autenticación";//cambia nombre del boton
    }
    else if (botonha.Text == "Deshabilitar Autenticación")
    {
        serialPort1.Write("SA,0" + c);//deshabilita autenticacion
    }
}

```

```
        botonha.Text = "Habilitar Autenticación";//cambia nombre del boton
    }
}

private void botontc_Click(object sender, EventArgs e)
{
    int tiempo;//variable tiempo
    opcion = 0;//opcion cero
    tiempo = int.Parse(tbtc.Text);//guarda tiempo
    serialPort1.Write("ST," + tiempo + c);//envia instruccion para cambiar el tiempo para entrar a el modo comando
}

private void botoncm_Click(object sender, EventArgs e)
{
    int velocidad;//variable velocidad
    opcion = 0;//opcion 0
    velocidad = int.Parse(veltrans.Text);//guarda velocidad de transmision
    if (paridad.Text == "PAR")
    {
        serialPort1.Write("U," + velocidad + ",E" + c);//manda la configuracion temporal del puerto serie
    }
    if (paridad.Text == "IMPAR")
    {
        serialPort1.Write("U," + velocidad + ",O" + c);//manda la configuracion temporal del puerto serie
    }
    if (paridad.Text == "NINGUNA")
    {
        serialPort1.Write("U," + velocidad + ",N" + c);//manda la configuracion temporal del puerto serie
    }
}

private void botoncb_Click(object sender, EventArgs e)
{
    opcion = 2;//opcion 0
    serialPort1.Write("D" + c);//envia comando de configuracion basica
}

private void botonlimpiar_Click(object sender, EventArgs e)
{
    tbedc.Clear();
}
}
}
```

Monitor de Puertos

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;

namespace MonitorPuertos
{
    public partial class MonitorPuertos : Form
    {
        public MonitorPuertos()
        {
            InitializeComponent();
        }

        private void MonitorPuertos_Load(object sender, EventArgs e)
        {
            comboBox1.SelectedIndex = 0; //seleccion de la opcion cero del combobox1
            comboBox2.SelectedIndex = 0; //seleccion de la opcion cero del combobox2
            foreach (String s in SerialPort.GetPortNames()) //obtencion de los puertos COM disponibles
            {
                comboBox1.Items.Add(s);
            }
            button2.Enabled = false; //deshabilita boton enviar
            button3.Enabled = false; //deshabilita boton limpiar
            textBox2.Enabled = false; //deshabilita cuadro de texto recepcion
            textBox1.Enabled = false; //deshabilita cuadro de texto transmision
        }

        private void button4_Click(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen == true) //cierra el puerto si esta abierto
            {
                serialPort1.Close();
            }
            Close(); //cierra la interfaz
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen == false) //si el puerto esta cerrado
            {
                serialPort1.PortName = comboBox1.Text; //nombre del puerto seleccionado
                serialPort1.BaudRate = int.Parse(comboBox2.Text); //velocidad de transmision seleccionada
                serialPort1.Open(); //abre el puerto
                button1.Text = ("Cerrar Puerto"); //cambia el nombre del boton
                button2.Enabled = true; //habilita boton enviar
                button3.Enabled = true; //habilita boton limpiar
            }
        }
    }
}
```

```
        textBox2.Enabled = true;//habilita cuadro de texto recepcion
        textBox1.Enabled = true;//habilita cuadro de texto transmision
    }
    else
    {
        serialPort1.Close();//cierra el puerto
        button1.Text = ("Abrir Puerto");//cambia el nombre del boton
        button2.Enabled = false;//deshabilita boton enviar
        button3.Enabled = false;//deshabilita boton limpiar
        textBox2.Enabled = false;//deshabilita cuadro de texto recepcion
        textBox1.Enabled = false;//deshabilita cuadro de texto transmision
    }
}
private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    String mensaje;//variable
    mensaje = serialPort1.ReadExisting();//guarda en mensaje lo que llega al rx del puerto serial
    textBox2.AppendText(mensaje);//agrega el texto que se recibe
}
private void button3_Click(object sender, EventArgs e)
{
    textBox2.Clear();//limpia el cuadro de texto recepcion
}
private void button2_Click(object sender, EventArgs e)
{
    String mensaje;//variable
    mensaje = textBox1.Text;//guarda el contenido del cuadro de texto transmision
    serialPort1.Write(mensaje);//envia el mensaje
    textBox1.Clear();//limpia el cuadro de texto transmision
}
}
}
```

Apéndice B

Programas y Diagrama de Bloques de las Pruebas con Microcontroladores

Prueba 1

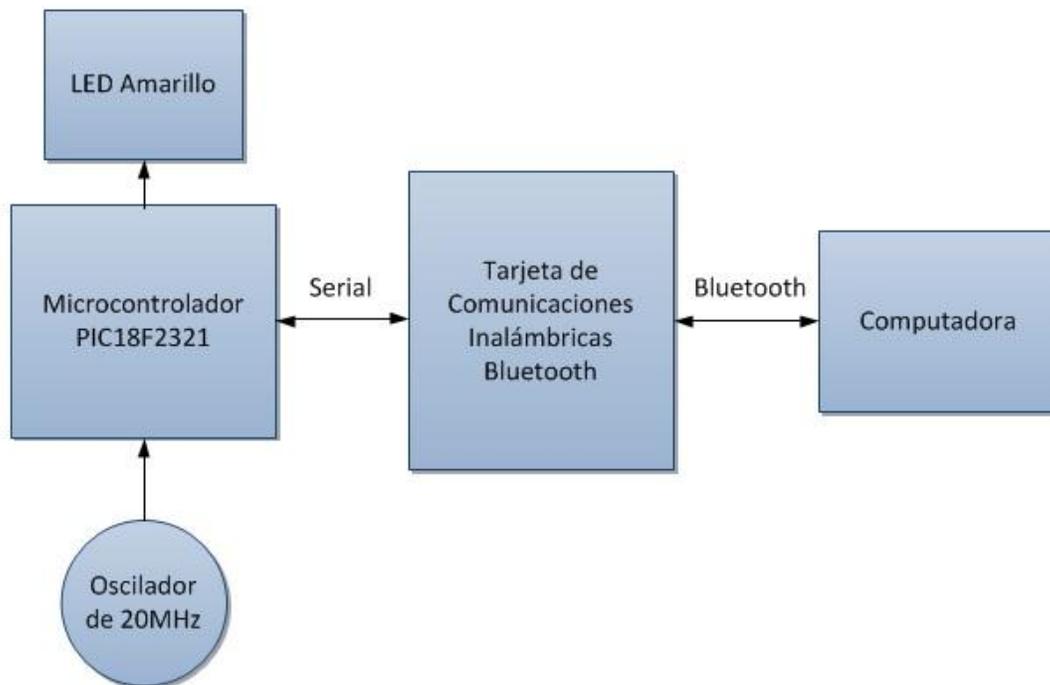


Figura B.1 Diagrama de Bloques de la Conexión Para la Prueba 1.

```
#include <p18f2321.h>  
#include <USART.h>  
#include <delays.h>
```

```
#pragma config OSC = INTIO1, WDT = OFF, LVP = OFF
```

```

char eco;

void envio (void);
void led (void);

#pragma code envio = 0x08//indica la direccion de memoria donde esta la interrupcion de alta prioridad
void la (void)
{
    _asm GOTO envio _endasm
}
#pragma code
#pragma interrupt envio
void envio (void)//interrupt
{
    PIE1bits.TXIE = 1;//habilita la interrupcion por transmision
    eco = ReadUSART();//lee el dato
    Delay10KTCYx(10);//retardo
    WriteUSART(eco);// pone el eco en el bufer de salida
}

#pragma code led = 0x18//indica la direccion de memoria del registro de baja prioridad
void lb (void)
{
    _asm GOTO led _endasm
}
#pragma code
#pragma interruptlow led
void led (void)//interruptlow
{
    PIE1bits.TXIE = 0;//deshabilita la interrupcion por transmision
    PORTCbits.RC4 ^= 1;//XOR para apagar el LED
}

void main (void)// programa principal
{
    OSCCON |= 0x70;//configura el oscilador a 8 [MHz]
    TRISCbits.TRISC4 = 0;//pin del led
    TRISCbits.TRISC6 = 1;//recepcion
    TRISCbits.TRISC7 = 1;//transmision

    OpenUSART(USART_TX_INT_OFF & //configuracion del puerto serie
              USART_RX_INT_OFF &
              USART_ASYNC_MODE &
              USART_EIGHT_BIT &
              USART_CONT_RX &
              USART_BRGH_HIGH, 51);

    RCONbits.IPEN = 1; //habilita la prioridad de las interrupciones
    INTCONbits.GIEH = 1;//habilita todas las interrupciones altas
    INTCONbits.GIEL = 1;//habilita todas las interrupciones bajas
    PIE1bits.RCIE = 1;//habilita interrupcion por recepcion
    IPR1bits.TXIP = 0;//interrupcion por transmision como baja
    while (1);
}

```

Prueba 2

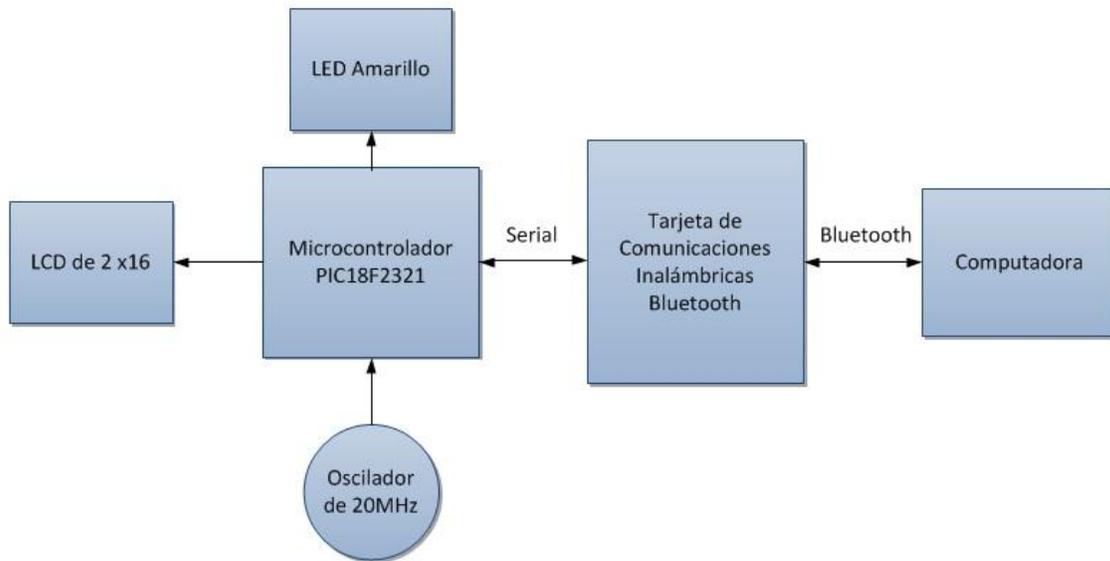


Figura B.2 Diagrama de Bloques de la Conexión Para la Prueba 2.

```

#include<p18f2321.h>
#include<usart.h>
#include<delays.h>
#include<stdlib.h>

#pragma config WDT = OFF, OSC = HS, LVP = OFF

#define RS PORTAbits.RA0
#define RW PORTAbits.RA1
#define DB4 PORTBbits.RB0
#define DB5 PORTBbits.RB1
#define DB6 PORTBbits.RB2
#define DB7 PORTBbits.RB3
#define I PORTCbits.RC3

void led(void);

#pragma code led = 0x08//indica la direccion de la interrupcion de alta prioridad
void la (void)
{
    _asm GOTO led _endasm
}
#pragma code
#pragma interrupt led
void led (void)//interrupcion para prender y pagar el led
{
    I ^= 1;
    INTCONbits.TMR0IF = 0;
}
    
```

```

#pragma code

void inicia_LCD (void);
void letra (char dato);
void E (void);

int tr = 6;
char eco;//variable que regresa lo escrito

void main (void){

    INTCONbits.PEIE = 1;//Habilita las interrupciones de los perifericos
    INTCONbits.GIE = 1;//Habilita las interrupciones
    INTCONbits.TMR0IE = 1;//Habilita la interrupcion por overflow

    TRISAbits.TRISA0 = 0;
    TRISAbits.TRISA1 = 0;
    TRISAbits.TRISA2 = 0;
    TRISBbits.TRISB0 = 0;//lcd 4 bits
    TRISBbits.TRISB1 = 0;
    TRISBbits.TRISB2 = 0;
    TRISBbits.TRISB3 = 0;
    TRISCbits.TRISC3 = 0;//led
    TRISCbits.TRISC6 = 1;//habilita puerto para Transmision
    TRISCbits.TRISC7 = 1;//habilita puerto para Recepcion

    Delay1KTCYx(80);//retardo adicional de mas de 15[ms]

    OpenUSART(USART_TX_INT_OFF &
               USART_RX_INT_OFF &
               USART_ASYNCH_MODE &
               USART_EIGHT_BIT &
               USART_CONT_RX &
               USART_BRGH_HIGH, 129);//configuracion de USART

    ADCON1 |= 0x0F;//salidas digitales en PORTA
    PORTA = 0;
    PORTB = 0;
    inicia_LCD();
    TOCON = 0x94;
    I = 1;//prende el led
    while (1)
    {
        while(!DataRdyUSART());//ve si hay dato en RCREG
        eco=ReadUSART();//lee el dato
        letra(eco);//pone la letra en el lcd
        WriteUSART(eco);//lo pone en el registro de salida
        while(BusyUSART());//espera a que se desocupe el registro de salida
    }
    CloseUSART();
}

void inicia_LCD (void){
    RS = 0;
    RW = 0;
}

```

```
Delay1KTCYx(80); // retardo de mas de 15ms
```

```
PORTB = 0x03;//1  
E();  
Delay100TCYx(tr);
```

```
Delay10TCYx(3);//retardo de mas de 4.7 us
```

```
PORTB = 0x03;//2  
E();  
Delay100TCYx(tr);
```

```
Delay100TCYx(6);//retardo de mas de 100 us
```

```
PORTB = 0x02;//3  
E();  
Delay100TCYx(tr);
```

```
PORTB = 0x02;//4  
E();  
Delay100TCYx(tr);
```

```
PORTB = 0x08;  
E();  
Delay100TCYx(tr);  
PORTB = 0x00;
```

```
E();  
Delay100TCYx(tr);  
PORTB = 0x0C;  
E();
```

```
Delay100TCYx(tr);  
PORTB = 0x00;  
E();  
Delay100TCYx(tr);
```

```
PORTB = 0x01;  
E();  
Delay1KTCYx(80);
```

```
PORTB = 0x00;  
E();  
Delay100TCYx(tr);  
PORTB = 0x06;  
E();  
Delay100TCYx(tr);
```

```
//initialization end
```

```
PORTB = 0x00;//display on/off control  
E();  
Delay100TCYx(tr);  
PORTB = 0x0C;  
E();  
Delay100TCYx(tr);
```

```
PORTB = 0x08;//posicion cursor  
E();
```

```
    Delay100TCYx(tr);
    PORTB = 0x05;
    E();
    Delay100TCYx(tr);
}

void letra (char dato){
    RS = 1;
    PORTB = dato >> 4;
    E();
    Delay100TCYx(tr);
    PORTB = dato;
    E();
    Delay100TCYx(tr);
    RS = 0;
}

void E(void)
{
    PORTAbits.RA2 = 1;
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    PORTAbits.RA2 = 0;
}
```