



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**DISEÑO Y FABRICACIÓN DEL SUBSISTEMA DE  
COMUNICACIONES UHF PARA UN NANO  
SATÉLITE EXPERIMENTAL**

**TESIS**

Que para obtener el título de

**INGENIERO ELÉCTRICO-ELECTRÓNICO**

**P R E S E N T A**

**RAMÍREZ SANTIAGO PEDRO**

**DIRECTOR DE TESIS**

Dr. Esaú Vicente Vivas



Ciudad Universitaria, Cd. Mx., 2016



Universidad Nacional  
Autónoma de México



**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.

# Agradecimientos

---

*A mis padres Pedro y Antonieta de quienes siempre he recibido apoyo, confianza y sobre todo cariño, no hay mayor orgullo que el ser su hijo, gracias por ser mis primeros y mejores maestros.*

*A mi hermana Minerva, por mostrarme el significado de la amistad, gracias por confiar y compartir todas esas experiencias con las que crecimos juntos. Sabes que siempre podrás contar conmigo.*

*A mi abuela Julia y mi tía Esther por su comprensión, apoyo y cariño incondicional gracias por todo, este trabajo y logro también es de ustedes.*

*A mis amigos durante la carrera Bernardo del Moral, Gabriela Pérez, Jorge González, Moisés González, Tomás Alvarado, Ulises Bravo y Cesar Gómez por todo el tiempo compartido a lo largo de esta etapa tan satisfactoria de mi vida.*

*A Brianda Salas, Ricardo Mora, Josué Romero y Rodrigo Camarena por ser incondicionales en mi vida, gracias por su sincera amistad y compañía a pesar del tiempo y la distancia.*

*A mi asesor el Dr. Esaú Vicente Vivas (†) y al Maestro Emilio Jiménez por confiar en mí y permitirme formar parte de este gran equipo, mi más grande admiración y respeto hacia ustedes.*

*Al Dr. Ramón Gutiérrez y al Ing. Enrique Rosas por el apoyo brindado para la realización de este trabajo, fue un honor haberlos conocido y el haber trabajado con ustedes.*

*A la Universidad Nacional Autónoma de México por la oportunidad de formar parte de esta casa de estudios, complementando mi desarrollo profesional así como el de ser un mejor ser humano.*

*A mis primos y tíos, gracias por formar esta gran familia, la cual me ha brindado consejos, consuelo y alientos para superarme tanto en lo profesional como en lo personal.*

# Índice

---

**Agradecimientos.....i**  
**Índice.....ii**  
**Lista de figuras.....v**  
**Lista de tablas.....vii**  
**Resumen.....viii**

**Capítulo 1 Antecedentes al Diseño del Nanosatélite Experimental 1**

- 1.1 Introducción.....1
- 1.2 Satélites Cubesat.....2
- 1.3 Arquitectura de un nanosatélite.....3
  - 1.3.1 Computadora de vuelo.....3
  - 1.3.2 Subsistema de potencia.....3
  - 1.3.3 Subsistema de comunicaciones.....4
  - 1.3.4 Subsistema de orientación y control de apuntamiento .....4
  - 1.3.5 Estructura.....4
  - 1.3.6 Carga útil.....5
- 1.4 Cubesats en misiones satelitales.....5
  - 1.4.1 E1P-2(Explorer-1 PRIME-2).....5
  - 1.4.2 RAX-2.....6
  - 1.4.3 Aeneas.....8

**Capítulo 2 Subsistema de Comunicaciones 9**

- 2.1 Introducción.....9
- 2.2 Esquema de comunicación.....9
  - 2.2.1 Segmento terrestre.....10
  - 2.2.2 Segmento espacial .....11
- 2.3 Bandas de frecuencia en que han operado Cubesats.....12
- 2.4 Protocolos de comunicación en Cubesats.....13
- 2.5 Configuraciones comunes de transeptores.....14
  - 2.5.1 Transeptor COTS.....14
  - 2.5.2 Transeptores modificados.....14
  - 2.5.3 Diseño de transmisor-receptor.....15

**Capítulo 3      Diseño del Subsistema de Comunicaciones      16**

---

3.1 Introducción.....16

3.2 Parámetros de diseño.....16

3.3 Descripción del diseño propuesto.....17

    3.3.1      Interfaz de USB a UART.....19

    3.3.2      Sensor de temperatura DS1820.....21

    3.3.3      Protecciones contra efecto Latch-up.....21

    3.3.4      Cambiador de nivel NTB0104.....27

    3.3.5      Transceptor ADF7021.....28

    3.3.6      Transceptor CC1101.....30

    3.3.7      Amplificador de potencia RFPA0133.....33

    3.3.8      Microcontrolador MSP430F1611.....34

**Capítulo 4      Fabricación del Subsistema de Comunicaciones      37**

---

4.1 Introducción.....37

4.2 Elaboración de la tarjeta electrónica.....37

    4.2.1      Software de desarrollo Altium Designer.....38

    4.2.2      Consideraciones en la elaboración.....41

        4.2.2.1      Ubicación de componentes.....42

        4.2.2.2      Ruteo de pistas.....42

        4.2.2.3      Planos de tierra.....43

4.3      Ensamble de componentes.....44

    4.3.1      Recomendaciones en cuanto al ensamble.....45

**Capítulo 5      Software de operación y pruebas de Validación del Subsistema de Comunicaciones      47**

---

5.1 Introducción.....47

5.2 Software de desarrollo Code Composer Studio.....48

5.3 Cargado de programa al MSP430F1611.....49

5.4 Pruebas realizadas al subsistema.....51

    5.4.1      Pruebas de protección contra efecto latch up.....51

        5.4.1.1      Circuito protector mayor a 300mA.....51

        5.4.1.2      Circuito protector menor a 300mA.....52

    5.4.2      Prueba de comunicación mediante el bus USB.....53

        5.4.2.1      Software Com Port Toolkit.....56

    5.4.3      Prueba de comunicación por medio del bus I2C.....56

5.4.4	Comunicación inalámbrica entre radios.....	60
5.4.4.1	Smart RF Studio 7.....	60
5.4.4.2	ADF7021 Evaluation Software.....	61
5.4.4.3	Prueba de comunicación inalámbrica entre transceptores CC1101.....	62
5.4.4.4	Prueba de comunicación inalámbrica entre transceptores ADF7021.....	70
<b>Capítulo 6</b>	<b>Conclusiones y recomendaciones</b>	<b>75</b>
<hr/>		
	<b>Bibliografía.....</b>	<b>78</b>
	<b>Apéndice A. Protocolo RS232.....</b>	<b>78</b>
	<b>Apéndice B. Protocolo I2C.....</b>	<b>80</b>
	<b>Apéndice C. Protocolo SPI.....</b>	<b>82</b>
	<b>Apéndice D. Esquemáticos del proyecto.....</b>	<b>83</b>

# Lista de figuras

---

Figura 1.1 Esquema de un CubeSat .....	2
Figura 1.2 Partes de un CubeSat .....	3
Figura 1.3 CubeSats con P-Pod .....	4
Figura 1.4 CubeSat HRBE.....	6
Figura 1.5 CubeSat RAX-2.....	7
Figura 1.6 Esquema de subsistemas del Aeneas.....	8
Figura 2.1 Esquema de comunicación.....	10
Figura 2.2 Diagrama de bloques de Estación terrena.....	11
Figura 2.3 Diagrama de bloques del segmento espacial.....	11
Figura 2.4 Radio Helium 100 de AstroDev.....	14
Figura 2.5 Radio Alinco DJ-C5.....	15
Figura 2.6 Tarjeta de desarrollo para el CC1101 de Texas Instrument.....	15
Figura 3.1 Diagrama a bloques del subsistema de comunicaciones.....	17
Figura 3.2 Diagrama de bloques del circuito FT232R.....	19
Figura 3.3 Diseño electrónico para el FT232R.....	20
Figura 3.4 Diagrama de DS18B20.....	21
Figura 3.5 Diseño del BQ24314.....	22
Figura 3.6 Diagrama simplificado del MAX4372.....	23
Figura 3.7 Diseño del comparador.....	25
Figura 3.8 Estructura de un flip-flop JK.....	26
Figura 3.9 Diseño de protección Latch Up menor a 30mA.....	27
Figura 3.10 Diseño del NTB0104.....	28
Figura 3.11 Diagrama a bloques del ADF7021.....	29
Figura 3.12 Diseño del ADF7021.....	30
Figura 3.13 Diagrama a bloques del CC1101.....	31
Figura 3.14 Diseño propuesto del transceptor CC1101.....	32
Figura 3.15 <i>Diseño del RFPA0133</i> .....	33
Figura 3.16 Diagrama a bloques del MSP430F1611.....	34
Figura 3.17 Parte de polarización y programación del MSP430F1611.....	35
Figura 4.1 Editor de esquemáticos en Altium Designer.....	39
Figura 4.2 Editor de componentes PCB.....	40
Figura 4.3 Distribución de hojas de esquemáticos.....	41
Figura 4.4 Posicionamiento de los componentes.....	42
Figura 4.5 Vista superior e inferior de la tarjeta ruteada.....	43
Figura 4.6 Tarjeta del subsistema de comunicaciones.....	44
Figura 4.7 Vista superior e inferior de la tarjeta ensamblada.....	46
Figura 5.1 Editor de código Code Composer Studio.....	48

Figura 5.2 Conexión JTAG para el cargado del programa al MSP430F1611.....	49
Figura 5.3 Ventana de depuración de Code Composer Studio.....	50
Figura 5.4 Pruebas en protoboard.....	51
Figura 5.5 Respuesta del circuito BQ24314 a corrientes altas.....	52
Figura 5.6 Prueba a la protección contra efecto latch up a 200mA.....	53
Figura 5.7 Datos transmitidos por el MCU vía serie utilizando el FT232R.....	54
Figura 5.8 Datos recibidos por el MCU vía serie utilizando el FT232R.....	55
Figura 5.9 Ventana principal de Com Port ToolKit.....	56
Figura 5.10 Datos enviados de un dispositivo esclavo a un maestro utilizando I2C.....	59
Figura 5.11 Programa Smart RF Studio 7.....	61
Figura 5.12 Ventana de ADF7021 Evaluation Software.....	61
Figura 5.13 Operaciones de escritura y lectura de registros.....	62
Figura 5.14 Formato del paquete.....	65
Figura 5.15 Diagrama de estados del CC1101.....	66
Figura 5.16 Escritura y lectura de registros.....	68
Figura 5.17 Datos transmitidos hacia el CC1101 por medio del protocolo SPI.....	69
Figura 5.18 Tiempo que tarda en transmitir un paquete de datos el CC1101.....	69
Figura 5.19 Diagrama de tiempos de la interfaz serial.....	70
Figura 5.20 Diagrama de tiempos de la interfaz Readback.....	71
Figura 5.21 Diagrama de tiempos para transmitir en modo SPI/UART.....	72
Figura 5.22 Diagrama de tiempos para recepción en modo SPI/UART.....	72
Figura 5.23 Secuencia a seguir en modo transmisor.....	73
Figura 5.24 Secuencia a seguir en modo receptor.....	74
Figura 5.25 Desglose de registro de configuración Readback No.7.....	75
Figura 5.26 Obtención de la revisión del semiconductor.....	75
Figura 5.27 <i>Valores obtenidos mediante la función Readback</i> .....	76



# Lista de tablas

---

Tabla 3.1 Resistencia sugerida por el fabricante .....	24
Tabla 3.2 Tabla de verdad de 74HC112.....	26
Tabla 3.3 Parámetros de diseño del ADF7021 .....	30
Tabla 3.4 Parámetros de diseño del CC1101 .....	32
Tabla 3.5 Pines utilizados del MSP430F1611.....	36
Tabla 5.1 Pines de interfaz JTAG.....	49
Tabla 5.2 Descripción de banderas de interrupción.....	59
Tabla 5.3 Requisitos de tiempo de la interfaz SPI.....	63
Tabla 5.4 Descripción de estados del byte de estatus.....	63
Tabla 5.5 Registros mínimos requeridos para configurar Tx y Rx.....	72

# Resumen

---

Este trabajo presenta el proceso de diseño, ensamble y validación de un Subsistema de Comunicaciones aplicable a un nanosatélite, siendo este compatible con el estándar CubeSat en cuanto a forma, masa y dimensiones utilizadas, simultáneamente esta acoplado con el resto de los subsistemas diseñados que conforman al satélite.

Se describe el proceso de comunicación entre el nanosatelite con la estación terrena, detallando los elementos necesarios para su realización mostrando protocolos y bandas de frecuencias mayormente utilizadas en este tipo de misiones. Además de mencionarse las tendencias existentes en cuanto a su realización.

El subsistema de comunicaciones que se diseñó cumple con parámetros específicos como cierta banda de frecuencia, velocidad y potencia de transmisión, así como el contar con circuitos de protección contra efectos de radiación específicos, situación a la que va a estar sometido el satélite una vez colocado en órbita.

Con base en los parámetros se realizó una búsqueda de componentes y configuraciones de componentes con los cuales se elaboró el diseño del subsistema, realizando una breve descripción de estos y su funcionamiento.

Teniendo una vez los componentes necesarios, se realizó el diseño de la tarjeta electrónica utilizando el software Altium Designer. Describiendo su funcionamiento en general así como aspectos que fueron de utilidad para generar nuestro circuito impreso.

Se describe una metodología de montaje para los circuitos integrados y demás componentes que conforman la tarjeta electrónica, para que una vez terminado el ensamble se prosiga con el software básico de operación del subsistema.

Se detallan las herramientas de programación para el desarrollo del software básico de operación del subsistema, probándose cada uno de los periféricos involucrados en el microcontrolador MSP430F1611. Comprobándose así entre otros aspectos un buen soldado de los componentes en la tarjeta y una buena operación por parte de los mismos componentes.

# Capítulo 1. Antecedentes al Diseño del Nanosatélite Experimental

---

## 1.1 Introducción

Desde los tiempos más antiguos, los seres humanos han explorado tierras desconocidas, en busca de conocimiento. Así, cuando el hombre supo que las estrellas que brillaban en el cielo eran cuerpos reales y lejanos, quiso llegar a ellos [1]. Después de la segunda guerra mundial Estados Unidos y la entonces Unión Soviética desarrollarían la tecnología necesaria para lograrlo. Dando así, comienzo a lo que fue conocido como carrera espacial. Siendo un periodo de competencia entre estos dos países a lo largo de 15 años, finalizando en 1975. Las dos potencias buscaban mostrar sus capacidades tecnológicas para demostrar superioridad y avances en seguridad nacional.

Esta carrera espacial empujó y logró grandes avances tecnológicos a nivel mundial, como el colocar en órbita al Sputnik I en 1957 siendo el primer satélite artificial de la Tierra. Después de este le siguieron el Telstar I en 1962 y el Intelsat I en 1965; siendo estos de los primeros satélites destinados a comunicaciones comerciales. Comenzando así, con la exploración espacial y la era satelital.

Lo que empezó como una carrera de defensa nacional, competencia por poder con fines militares, se ha convertido actualmente en colaboración internacional, exploración espacial y grandes desarrollos científicos [2].

México a pesar de haber contado con satélites desde 1985, fecha en que se pusieron en órbita los satélites Morelos I y II, siempre ha estado ligado a tecnología espacial de terceros. Teniendo así, un escaso desarrollo aeroespacial dentro del país; pues casos como los satélites Solidaridad I y II, así como el más reciente satélite Bicentenario lanzado en 2012 fueron fabricados por la firma internacional Boeing. El ser consumidores de tecnología extranjera ha impedido el crecimiento de una industria propia capaz de generar tecnología en el rubro aeroespacial.

Los satélites en el espacio actualmente utilizan procesadores digitales, con los cuales se desarrolla software especializado para ejecutar y definir sus funciones y desempeño. Se vislumbra en un futuro no muy lejano el desarrollo de sistemas con mayores capacidades de movilidad, más versátiles, poderosos en funciones y de menor costo. Siendo un claro ejemplo de estos los nanosatélites y redes de nanosatélites.

El desarrollo de satélites pequeños se inició en los entornos académicos, sin embargo, en los últimos años existen instancias empresariales, tanto de gran envergadura, como emprendimientos (start-ups) que impulsan esa disciplina para incorporar dispositivos en una serie de aplicaciones en donde se observa una convergencia entre los campos aeroespaciales, robótica y manufactura avanzada a bajo costo.

## 1.2 Satélites Cubesat

El concepto de CubeSat comenzó en 1999 en dos universidades de Estados Unidos, la Universidad Politécnica del Estado de California y la Universidad de Stanford. El propósito de este proyecto fue crear una estandarización en cuanto al diseño de picosatélites, con el cual reducir costos y tiempos de investigación; impulsar el desarrollo espacial y generar lanzamientos exitosos con mayor frecuencia.

El estándar Cubesat (1U) en un principio señalaba como dimensiones un cubo de 10x10x10 cm teniendo como masa máxima 1.33 kg. Con el tiempo, debido a la funcionalidad de su diseño, se estandarizaron tamaños conocidos como 2U y 3U siendo sus alturas y sus masas la diferencia respecto al de 1U; siendo de 20cm y 30cm las alturas correspondientes.

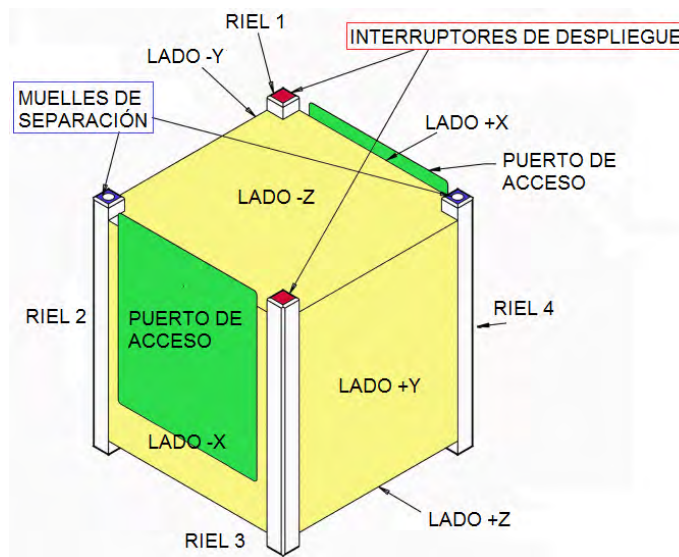


Figura 1.1 Esquema de un CubeSat

El estándar CubeSat incluye a su vez, un sistema de lanzamiento para este tipo de satélites, el P-POD (PolyPico-satellite Orbital Deployer) el cual es capaz de adaptarse a los distintos tamaños de CubeSats sin ninguna clase de complicaciones, siempre y cuando el diseño de estos satélites cumpla con los parámetros de diseño y de manufactura que dicta el estándar, como lo son las masas, los tipos de materiales a usar y conexiones dentro del CubeSat [3].

La forma en que está constituido un CubeSat trae con él grandes facilidades para su desarrollo. Su bajo costo, dimensiones y materiales hacen posible que entidades como las universidades y empresas tengan un crecimiento en lo que se refiere a tecnología aeroespacial, rubro que se limitaba a entes como lo eran la NASA o agencias espaciales de algunos países. Así como el poder validar nuevas tecnologías espaciales y el generar recursos humanos capaces de desarrollar dicha tecnología.

### 1.3 Arquitectura de un nanosatélite

Un nanosatélite está conformado por un conjunto de subsistemas. Cada subsistema tiene una tarea específica, algunos se encargan de la operación del propio satélite en forma remota siendo ejemplos de ello los subsistemas de computadora de vuelo, potencia, comunicaciones, orientación y control de apuntamiento y la estructura; mientras que existen otros subsistemas que se enfocan en brindar el servicio específico que presta el satélite al usuario [4].

La arquitectura del nanosatélite depende primordialmente del servicio que vaya a brindar la misión. De ahí se darán a conocer las necesidades que tenga que cubrir por medio de los subsistemas y que a continuación se describirán brevemente.

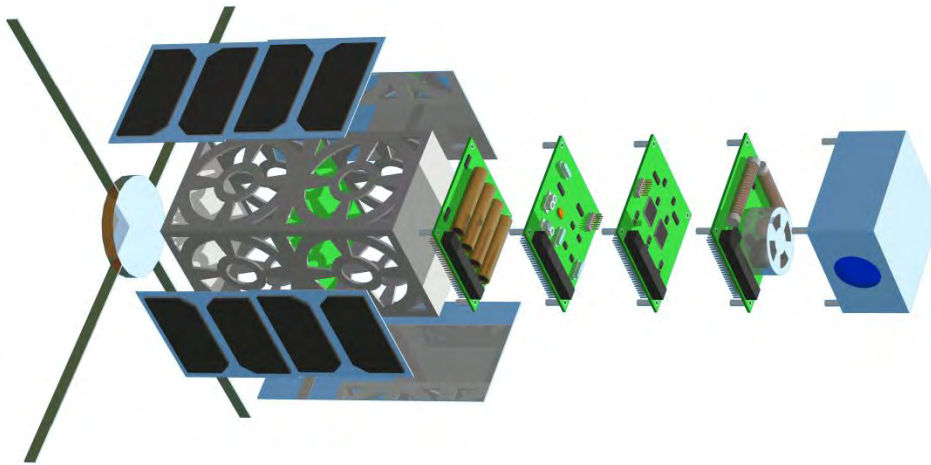


Figura 1.2 Partes de un CubeSat

#### 1.3.1 Computadora de vuelo

La Computadora de vuelo es la unidad principal de procesamiento, encargada de controlar y ejecutar las instrucciones externas e internas para la operación de la plataforma y de la carga útil. Dirige el flujo de información entre subsistemas, ordena la secuencia de actividades a realizar e interpreta información recibida del exterior por medio de otros subsistemas. Este subsistema inteligente debe estar preparado ante posibles fallas recurrentes en el espacio, tal es el caso del efecto “Latch Up” generado por radiación en el espacio. Generalmente conformado por un microcontrolador principal, un banco de memorias, protecciones contra efecto “Latch Up” y buses que le permitan interactuar con los demás subsistemas.

#### 1.3.2 Subsistema de potencia

El subsistema de potencia se enfoca en generar y distribuir la energía necesaria para la operación del satélite en órbita. Cuenta con baterías recargables, con las cuales alimenta a todos los subsistemas. Este subsistema es el primero en encender, lo cual ocurre por medio de interruptores posicionados estratégicamente dentro del satélite. Cuenta con reguladores y convertidores de DC a DC para generar los niveles de voltaje apropiados para cada subsistema. Los niveles de voltaje son transmitidos a través de un bus al subsistema correspondiente. El subsistema de potencia debe ser capaz de recargar las baterías ya que de no ser así, el satélite tendría un periodo reducido de vida.

Para ello se colocan paneles solares en las caras externas aprovechando la radiación solar que recibe y recargan las baterías.

### **1.3.3 Subsistema de comunicaciones**

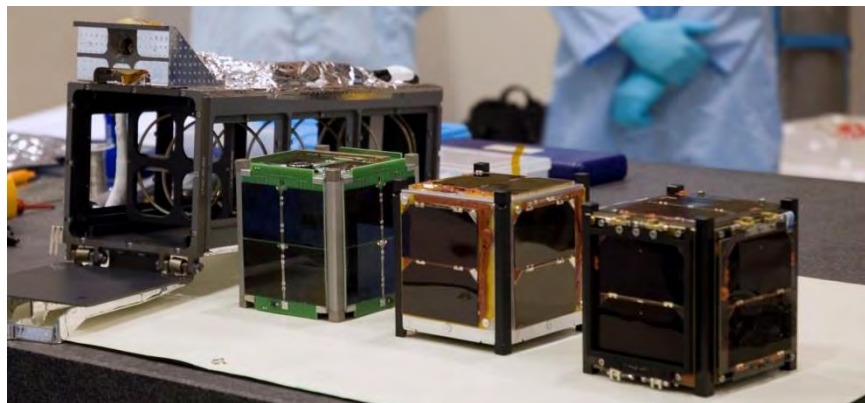
Este subsistema tiene la fundamental tarea de recibir y transmitir toda aquella información que esté direccionada al satélite. Mediante este subsistema el satélite se encarga de mandar la telemetría e información generada por la carga útil; la computadora de vuelo procesa la información y la transmite a este subsistema, siendo este el que modula y adapta la información para mandarla a la antena transmisora. En el caso inverso la estación terrena envía un comando el cual llega a la antena receptora, el subsistema de comunicaciones demodula el comando y lo envía a computadora de vuelo, quien lo interpreta y realiza las acciones pertinentes. Para realizar dicha operación el subsistema cuenta con transmisores y receptores que operan en los rangos de VHF, UHF y banda S.

### **1.3.4 Subsistema de orientación y control de apuntamiento**

Este subsistema se encarga de mantener al nanosatélite en cierta posición, direccionando sus caras de forma específica. Este subsistema posee sensores con retroalimentación como sensores de sol que le ayudan a orientarlo. Otra de las tareas de las que se encarga es la de estabilizar al satélite, para esto hay métodos pasivos y activos, como lo es el gradiente gravitacional y las ruedas inerciales respectivamente. El control de apuntamiento es necesario para un buen funcionamiento de antenas y celdas solares, así como el de la carga útil. Siendo esta última la de mayor influencia a la hora de elegir un sistema de estabilización, pues es la que dicta los grados de libertad que podría tener el subsistema.

### **1.3.5 Estructura**

La estructura tiene la difícil tarea de sostener y proteger a todos los subsistemas que conforman el satélite. Implica tomar en cuenta consideraciones como puntos de sujeción, materiales a usar y la vibración a la que va a estar sometido dicho material tanto en el lanzamiento así como su vida en órbita. Esta estructura funciona además como disipador de calor de componentes electrónicos a lo largo de todo el satélite. Al ser un CubeSat, el estándar dicta ciertos parámetros en cuanto al diseño de esta parte del satélite pues es la que está en contacto con el dispositivo de lanzamiento.



*Figura 1.3 Cubesats con P-Pod.*

### **1.3.6 Carga útil**

La carga útil, como ya se mencionó con anterioridad, es la razón por la cual se realiza la misión. El satélite tiene como objetivo brindar un servicio; siendo este de carácter científico, comercial o privado. Casos como lo es percepción remota llevan consigo cámaras multiespectrales o infrarrojas. Satélites experimentales tienen como carga útil sensores o aparatos especializados cuya finalidad es obtener mediciones en el espacio de cierto fenómeno. Misiones de comunicaciones primordialmente llevan equipos de antenas o dispositivos repetidores los cuales transmiten la información que reciben. Recordemos que uno de los principales objetivos de un CubeSat es el poder validar y generar tecnología espacial.

### **1.4 CubeSats en misiones satelitales**

Debido a su gran versatilidad y fácil construcción, los CubeSats han generado que compañías satelitales así como institutos de investigación y universidades hayan comenzado a desarrollar nanosatélites de este tipo.

En Junio de 2003, fueron lanzados al espacio los primeros CubeSats, siendo lanzados a una órbita a 810 km de altura. El lanzamiento fue coordinado por el grupo de vuelos espaciales de la Universidad de Toronto, en Plesetsk, Rusia. Dando así el comienzo a una nueva era en el campo satelital [5].

Hasta el año 2012 se tenía un aproximado de 65 CubeSats en el espacio [6], teniendo como resultado en la mayoría de los casos lanzamientos exitosos. Se mostrarán a continuación algunos CubeSats enviados y los resultados obtenidos de ellos.

#### **1.4.1 E1P-2(Explorer-1 PRIME-2)**

EL E1P-2 forma parte de la misión llamada HRBE (Hiscock Radiation Belt Explorer) siendo lanzado el 28 de octubre de 2011.

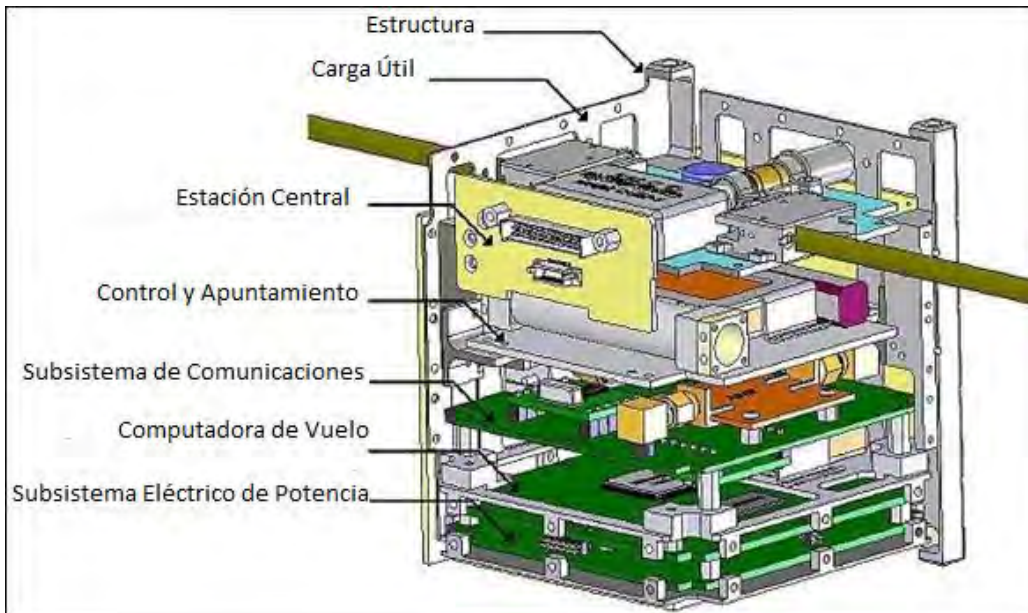
El objetivo científico de la misión es detectar los niveles de radiación en los cinturones de Van Allen, así como el medir la intensidad y la variabilidad de la energía de los electrones en orbitas bajas (LEO). Esta misión contribuyó al desarrollo de trabajo aeroespacial mediante la participación de estudiantes universitarios en el diseño de la nave espacial, el desarrollo y las operaciones realizadas por dicho artefacto [7].

El satélite, se ajusta a los estándares 1U CubeSat. Su subsistema de comunicación se basa en el chip CC1000 de *Texas Instrument* tomando un chip de receptor a 437.305 MHz y otro de transmisor a 437.505 MHz, el cual amplifica por medio de un RF5110G la salida a 850mW, enviándola a una antena del tipo monopolo. El empaquetamiento se lleva a cabo por el estándar AX.25 a 1200 bauds y con una modulación del tipo FSK. Además de contar con un Beacon (un radio transmisor) de telemetría continua a 435.505 MHz.

En el subsistema eléctrico de potencia se tienen 2 baterías de Li-ion, reguladores a 3.3V y 5V y 6 paneles solares, los cuales se encargan de suministrar la alimentación a todo el satélite. Mientras

que en la computadora de vuelo se tiene un microcontrolador HCSX12 y almacenamiento de datos en memorias flash. Por último en control y apuntamiento cuenta con un sistema pasivo que alinea al satélite con el campo magnético de la tierra.

La carga útil está compuesta por un sencillo tubo Geiger-Müller para controlar el flujo de electrones atrapados en los cuernos de los cinturones de radiación interiores y exteriores [8].



*Figura 1.4 Cubesat HRBE*

Hasta el 28 de febrero de 2014, después de 28 meses de operar, el satélite continúa con sus actividades de manera normal en órbita. La misión se consideró todo un éxito ya que hasta esa fecha ha recolectado 47000 telemetrías siendo decodificadas aproximadamente la mitad por la comunidad amateur y 210 horas de información científica, además de probar su utilidad como herramienta educativa y como plataforma científica

#### **1.4.2 RAX-2**

EL RAX 2 (Radio Aurora Explorer-2) es un CubeSat de 3U 10x10x34cm con un peso de 2.8kg [9]. Siendo una colaboración entre la Universidad de Michigan, la compañía International SRI. Fue lanzado el 28 de Octubre de 2011 como una carga secundaria por la NASA.

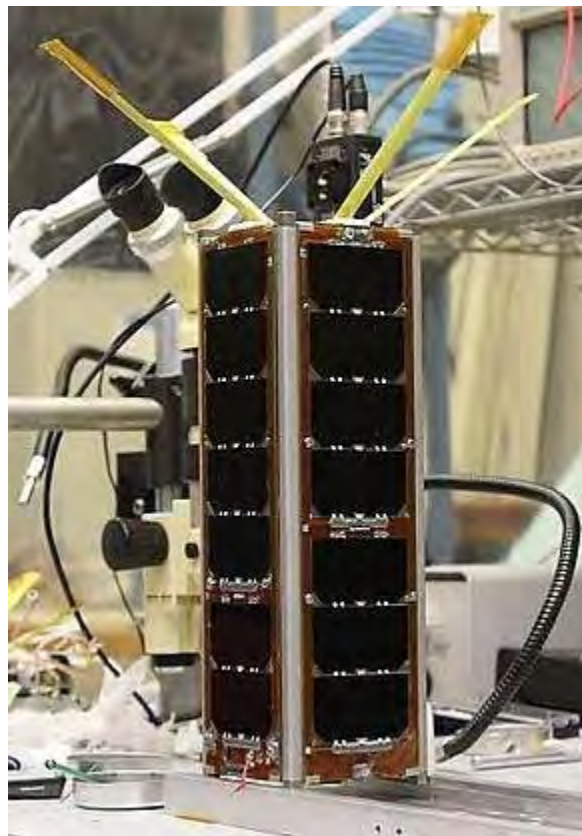
El objetivo de la misión fue el estudio de las condiciones climáticas del espacio. Encargándose de analizar las grandes cantidades de plasma en la ionosfera, siendo capaz de estudiar las perturbaciones que tiene esta capa de la tierra, ya que se sabe que las inestabilidades del plasma provocan irregularidades en el campo magnético las cuales se sabe que interrumpen la comunicación entre la Tierra y naves espaciales en órbita.

El satélite está compuesto por subsistemas de comunicación RF, Computadora de vuelo, Eléctrico de potencia, Estabilización y Control de apuntamiento y por último una carga útil.



El subsistema de comunicaciones se encontraba integrado por transceptores ya fabricados, el Lithium-1 de la compañía *Astronautical Development* y el MHX2400 de *Microhard*. El primero operando en la banda del UHF a una velocidad de 8.4kbits/s encargado de bajar la telemetría y comandos del satélite. El segundo operando en banda S se utilizó para el manejo de datos científicos obtenidos de la carga útil. La computadora de vuelo, se compuso por un MSP430, encargado de procesar los datos generados por la carga útil; un FPGA el cual recopila los datos teniendo un almacenamiento de 8GB.

El subsistema de potencia cuenta con paneles solares de triple juntura, reguladores de voltajes a 3.3V y 5V, así como baterías ion-litio. Mientras que el subsistema de estabilización y control de apuntamiento cuenta con magnetómetros de tres ejes, sensores de sol en las seis caras del satélite, tres giróscopos, Filtro de Kalman extendido, un receptor GPS proporcionando precisión en la posición (menor a 20m) y el tiempo orbital (menor de 1 ms) [10].



*Figura 1.5 CubeSat RAX-2*

El estatus de la misión esta como inactivo, ya que se perdió contacto con el satélite el 01 de julio de 2013 por medio del MHX 2400. Mientras que en UHF se perdió comunicación desde el 20 de abril de ese mismo año, se cree que este aislamiento se debió a una anomalía en la corriente de la computadora de vuelo o en el subsistema de potencia. Aunque el cese de operaciones fue inesperado, RAX-2 completó plenamente su misión prevista de un año y ha proporcionado mediciones sin precedentes de las irregularidades de la ionosfera.

### **1.4.3 Aeneas**

Aeneas es un CubeSat de 3 unidades (3U), lanzado el 13 de septiembre de 2012 y desarrollado por estudiantes de la Universidad del Sur de California, junto con el apoyo del laboratorio de Investigación de la Fuerza Aérea de Kirtland, esta misión tuvo la finalidad de validar un programa de demostración de tecnología del Departamento de Seguridad Nacional para el seguimiento de los contenedores de carga en todo el mundo.

Este programa es el primero de varios esfuerzos de satélites y naves espaciales de desarrollo, financiados por la USC que proporcionarán a los estudiantes experiencia en el montaje, diseño e integración de satélites [11].

El satélite se compone de un subsistema de estabilización y control de apuntamiento, integrado por tres ruedas inerciales, bobinas de torque magnético, magnetómetros y sensores finos de sol montados en los paneles solares; un subsistema de potencia eléctrico desplegable, hecho por la compañía *ClydeSpace*, el cual cuenta con paneles solares y baterías del tipo ion-Litio, que se encargan de distribuir la alimentación a todo el satélite.

El subsistema de comunicaciones lo conforman dos transceptores de diferentes compañías, el primero es un MHX425 de la compañía *Microhard* utilizado para bajar telemetría y comandos del satélite a Tierra, mientras que el segundo es el *AstroDev Neon*, el cual funciona como un beacon. El subsistema tiene dos monopolos como antenas y los microprocesadores que utilizan para comunicarse son PIC'S de la familia 24.

La carga útil de la misión requiere un plato de banda S de despliegue de medio metro que ocupa 1.5U del satélite. Una vez enlazado a los contenedores de carga en el océano, la antena se cierra y los datos de identificación son transferidos y se utilizan para geolocalizar el contenedor en la superficie de la tierra [12].

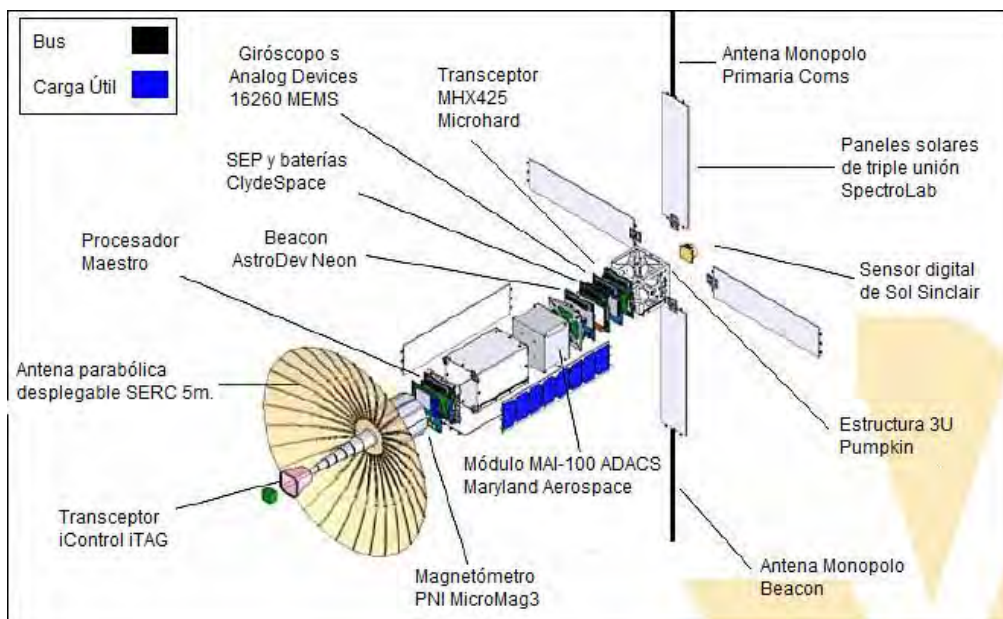


Figura 1.6 Esquema de subsistemas del Aeneas

El estatus de la misión es operando nominalmente hasta febrero de 2014.

# Capítulo 2

## Subsistema de Comunicaciones

---

### 2.1 Introducción

El subsistema de comunicaciones es uno de los factores limitantes para el desarrollo de un nanosatélite, pues tiene la gran responsabilidad de mantener en contacto al satélite con la estación terrena. Sin un buen funcionamiento de este subsistema se perdería todo enlace de comunicación con el satélite lo cual significaría un total aislamiento del mismo.

Este capítulo muestra como es el proceso de comunicación de un nanosatélite con la Tierra de manera general. Desde la generación de la información a ser enviada al Satélite estando en una estación terrena; pasando por una antena para ser transmitida hasta el ser recibida por el subsistema de comunicaciones dentro del satélite. Así como el tipo de protocolos y bandas de frecuencia que han utilizado este tipo de satélites.

Si bien el diseño de este subsistema es libre, cabe mencionar que existen tendencias en cuanto a su realización, siendo estas mencionadas al final de este capítulo.

### 2.2 Esquema de comunicación

El subsistema de Comunicaciones VHF-UHF, como se ha mencionado, tiene como objetivos primordiales el enviar la telemetría del satélite a Tierra y el recibir comandos generados de una estación terrena. Para que esto sea posible, la información debe pasar por ciertas etapas en las cuales se ve procesada para poder ser transmitida y/o recibida según sea el caso. A este proceso lo denominamos esquema de comunicación, siendo de vital importancia su conocimiento para así poder describir algunos parámetros o necesidades que se tendrán que cubrir.

Con el fin de dar una idea muy general en cuanto a este proceso, el esquema de comunicación satélite-Tierra está dividido en dos partes: la parte de comunicación terrestre y la parte de comunicación espacial. Los elementos involucrados se muestran en la figura 2.1.

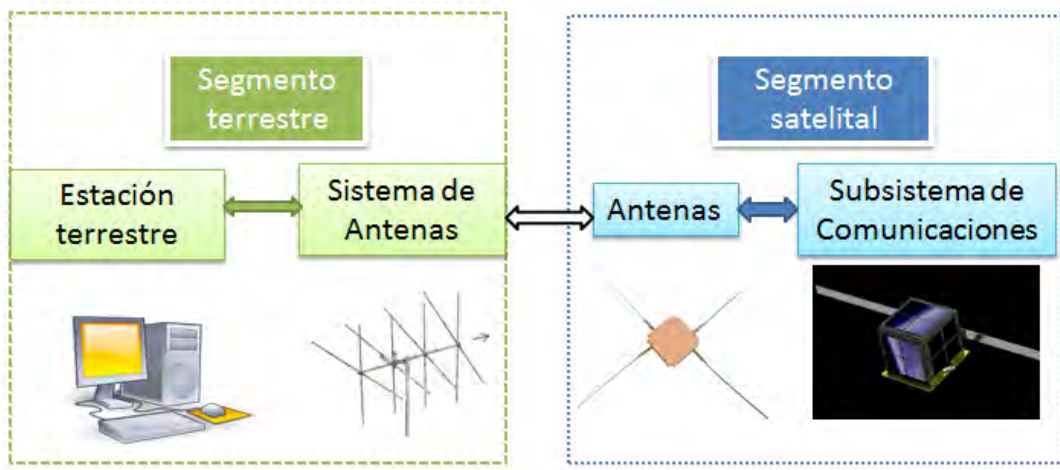


Figura 2.1 Esquema de comunicación.

### 2.2.1 Segmento terrestre

Como su nombre lo indica, este segmento de comunicación es el que se encuentra en Tierra; es el encargado de procesar comandos y telemetría que se desean transmitir y/o recibir del satélite a Tierra o viceversa. El segmento también es conocido como estación terrena [4].

Los componentes que integran una estación terrena varían de una a otra, esto se debe a la aplicación y el tipo de comunicación que se tiene con el satélite. Para el caso de satélites pequeños en órbita baja, se observa que las estaciones terrenas más completas, están integradas por:

- Rotadores con su respectivo controlador: con esto se manipula el azimut y la elevación de la antena para seguir el satélite durante su trayectoria.
- Antenas: encargadas de recibir y transmitir la señal en forma de ondas electromagnéticas a una cierta frecuencia.
- Transceptores para las bandas VHF y UHF. Integrados por módems y bloques de ganancia necesarios para la comunicación [13].

Bloques como UPS utilizados como circuitos de protección, ante una interrupción en la alimentación y receptores en banda S se consideran como opcionales, ya que depende del tener en el satélite un módulo que opere en este rango de frecuencias. Un ejemplo a forma de bloques se muestra en la siguiente figura:

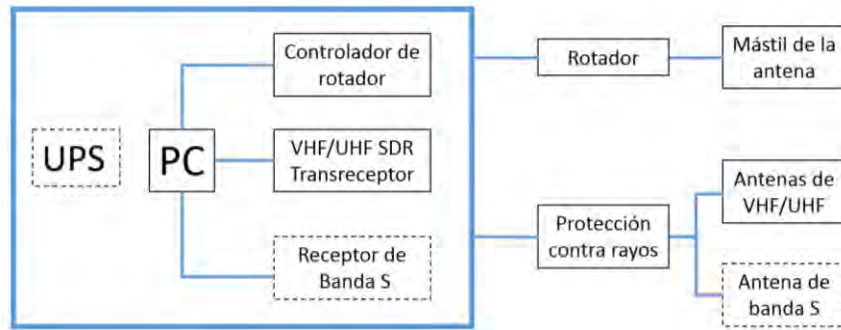


Figura 2.2 Diagrama de bloques de Estación Terrena

Las estaciones terrenas están ancladas a una Computadora, la cual tiene la tarea de procesar e interpretar la información a transmitir o recibir por los transmisores o receptores, para que el usuario pueda observar el comportamiento del satélite desde Tierra. Por lo general la PC se comunica por medio de un protocolo serial, lo que facilita la interacción con los demás componentes de la estación como los controladores de rotación o los transceptores.

### 2.2.2 Segmento espacial

La parte espacial de comunicación está conformada básicamente por el subsistema de comunicaciones y por la tarjeta de antenas del satélite. El proceso de comunicación dentro del satélite se muestra en la figura 2.3



Figura 2.3 Diagrama de bloques del segmento espacial

La antena recibe en forma de ondas electromagnéticas la información transmitida desde la estación terrena, convirtiéndola en una señal capaz de ser interpretada por los módems de comunicación, el cual se encarga de amplificar la señal, demodularla y mandarla como una trama de datos al microcontrolador del subsistema. Este es el encargado de procesar, verificar y retransmitir las tramas de datos al subsistema computadora de vuelo por medio del bus que comunica a todos los subsistemas del satélite. Siendo la computadora de vuelo quien interpreta el comando recibido desde la estación terrena y ejecuta las acciones correspondientes.

En el caso inverso, la computadora de vuelo genera las tramas de datos a enviar, redireccionándola al microcontrolador del subsistema de comunicaciones, el cual las almacena y las envía al modem para así, ser moduladas, amplificadas y enviadas a la tarjeta de antenas quien se encarga de

propagar la información en forma de ondas electromagnéticas para ser recibida por las antenas de la estación terrena.

### **2.3 Bandas de frecuencia en que han operado Cubesats**

En el proyecto CubeSat se propone utilizar las frecuencias de radioaficionados para la comunicación entre el satélite y la estación terrena. Generalmente se utilizan las bandas de 144MHz, la de 435MHz hasta 438MHz, de 900MHz y en algunos casos la de 2.4GHz para misiones en las que se necesitan una mayor tasa de transferencia de datos [6].

El servicio de radioaficionados es un servicio de comunicación por radio con el propósito de auto-formar, intercomunicar y generar estudios técnicos mediante personas debidamente autorizadas que se interesan en la radiotecnica con carácter exclusivamente personal y sin fines de lucro.

En la mayoría de los proyectos que involucran la comunicación con un CubeSat, se utilizan este tipo de bandas de frecuencias debido a cinco razones fundamentales:

- El obtener una licencia no genera un gran costo comparada con otras bandas de frecuencia.
- Cualquier persona que cuente con equipos de radioaficionado, puede recibir datos de telemetría del CubeSat. Pudiendo compartir datos con los radioaficionados de todo el mundo y poder analizarlos en conjunto. Teniendo más información del satélite que si tuviéramos solo contacto con la estación terrena.
- Para dar seguimiento al satélite en órbita inmediatamente después del lanzamiento pues se complica el identificar los parámetros orbitales del CubeSat, siendo de gran valor e importancia la ayuda de los radioaficionados al darle continuidad al satélite.
- Se pueden realizar experimentos de comunicación entre el satélite, la estación terrena y los radioaficionados, utilizando protocolos ya establecidos, mejorándolos en cuanto a la detección y corrección de errores, así como la validación de nuevos protocolos de comunicación.
- Muchos CubeSats utilizan frecuencias de radioaficionados debido al hardware fácilmente disponible para esas frecuencias. Sólo cuando el hardware del radio satelital y la estación terrena tenga mayor disponibilidad y sea probado en vuelos, los CubeSats emigrarán lejos de frecuencias satelitales para aficionados [14].

## 2.4 Protocolos de comunicación en Cubesats

Existe una amplia cantidad de protocolos de comunicación satelital diseñados para realizar esta comunicación inalámbrica. Varios protocolos han sido desarrollados de acuerdo a las necesidades de cada CubeSat, ya que un solo protocolo no siempre es el adecuado para la aplicación deseada.

Cerca del 55% de los CubeSats enviados hasta el año 2014 han utilizado el protocolo de comunicaciones AX.25. Otro porcentaje de las misiones o utilizaron un protocolo que es propiedad de la empresa encargada de elaborar el radio empleado o no dieron información relacionada al protocolo y otro porcentaje, aún menor utilizaron uno de los siguientes protocolos:

- ❖ AMSAT-DL. Este protocolo es utilizado en todos los satélites de la fase 3 de AMSAT. Es menos común que el AX.25 y requiere mayor software para procesar y enviar.
- ❖ RLP. Protocolo de enlace entre radios utilizado en la red GSM.
- ❖ IEEE802.2-LLC. Protocolo estándar para enlace de datos como lo son WiFi, GPRS y conexión inalámbrica de área local.
- ❖ XSTP (eXtended Satellite Transport Protocol). Protocolo desarrollado por la Universidad de Toronto para la serie de satélites Can-X. es un protocolo derivado del AX.25.
- ❖ SRLL (Simple Radio Link Layer). Basado en el protocolo AX.25, capaz de cancelar errores y corregirlos, fue desarrollado por el Instituto Tecnológico de Tokio [15].

El protocolo estándar AX.25 es definido como un estándar abierto, es utilizado para transmisiones digitales de radioaficionados, también conocido como radio por paquetes, es el más empleado para la transmisión de datos digitales en esta banda. Este protocolo no depende de la velocidad de transmisión y puede ser generado por un dispositivo llamado Controlador de Terminal de Nodo ó TNC por sus siglas en inglés, el cual convierte los datos digitales a una señal modulada y viceversa. Siendo que el TNC es conectado directamente a una computadora personal. En operaciones satelitales de radioaficionados el TNC opera de modo en el cual transmite de forma transparente los datos entre la PC y este mismo.

Una de las razones por las que se emplea primordialmente, es que varias misiones necesitan el utilizar protocolos de comunicación libre, para así poder recibir información adicional de los radioaficionados.

## 2.5 Configuraciones comunes de transceptores

Como se mencionó con anterioridad, el subsistema de comunicaciones es una parte vital de cualquier satélite. Sin ninguna forma de comunicarse, el CubeSat pronto fallaría en su misión. Al comenzar a investigar sobre subsistemas de comunicaciones, se observó que los nanosatélites enviados hasta ahora han tenido tres posibilidades en la construcción u obtención de dicho subsistema: el comprar un transceptor COTS, el comprar uno de uso terrestre y modificarlo o el diseñar un transmisor-receptor a partir de componentes individuales.

### 2.5.1 Transceptor COTS

El término COTS (Commercial Off-The-Shelf), hace referencia a productos comerciales que no han sido modificados para un uso en específico, como lo es el sector espacial. El comprar un radio de este tipo para un subsistema de comunicaciones simplifica su diseño, pues normalmente estos radios aceptan datos de forma serial, los propios radios empaquetan, verifican errores y retransmiten la información de manera desconocida, ya que la mayoría de los protocolos y modulaciones de estos dispositivos son propiedad de la compañía. Esto hace que, en nuestra estación terrena tengamos la necesidad de tener un radio idéntico y descarta cualquier red de estaciones terrenas a gran escala.

Existen varias compañías que fabrican transceptores de grado espacial, los cuales tienen un costo elevado, son pesados y son de gran tamaño para un CubeSat. El grupo Stensat fabrica transceptores específicamente para CubeSats cumpliendo con medidas y tamaños para este tipo de satélites los cuales ya han sido probados en el espacio. Empresas como AstroDev e ISIS comenzaron también, a vender radios diseñados para este propósito en particular.



*Figura 2.4 Radio Helium 100 de AstroDev.*

### 2.5.2 Transceptores Modificados

Esta alternativa para un subsistema de comunicaciones implica utilizar radios de uso terrestre. Este tipo de radios tendrían serios problemas en cuanto a su operación en el espacio, ejemplo de ello se tiene en el caso de la disipación térmica, a falta de aire para ser enfriados los amplificadores de estos equipos. El utilizar radios de este tipo implica modificaciones como lo son el remover la carcasa para reducir masa y tamaño, el programar el radio de acuerdo a las capacidades de potencia que se tienen, realizar interfaces entre el radio y el satélite para la manipulación del mismo, agujeros de



montaje, eliminación de pantallas LCD y botones, aspectos que inclusive requieren ayuda del fabricante.

Radios comerciales de las empresas Microhard y Alinco han sido probados con anterioridad y han tenido resultados exitosos.



Figura 2.5 Radio Alinco DJ-C5

### 2.5.3 Diseño de transmisor-receptor

En algunas misiones, principalmente en las que participan en el desarrollo del satélite las universidades, se decide construir componente por componente todo el transceptor. Esto permite tener un sistema personalizado con un mejor control de las especificaciones de nuestro subsistema. Generando así un trabajo desde cero, en el cual, las próximas generaciones de estudiantes tengan una motivación y una base sobre la construcción de circuitos de este tipo [6].

Los componentes de este tipo de transceptores incluyen un TNC, un chip de radiofrecuencia y amplificadores de potencia. Generalmente se utiliza un microcontrolador como TNC, el cual programa y configura los registros del transceptor. Los chips de radiofrecuencia más utilizados son de Texas Instruments, RF Microdevices y Analog Devices, los cuales operan sin ninguna complicación en las bandas de radioaficionados.



Figura 2.6 Tarjeta de desarrollo para el CC1110 de Texas Instrument

# Capítulo 3 Diseño del Subsistema de Comunicaciones

---

## 3.1 Introducción

Para dar comienzo con el diseño al subsistema de comunicaciones se necesitan conocer ciertas especificaciones con las cuales se pretende cumplir. Aspectos mecánicos, en cuanto a transmisión como la asignación de bandas de frecuencias, velocidad, esquemas de modulación o la potencia al transmitir, así como el tomar en cuenta aspectos externos a los que va a estar sometido el satélite una vez colocado en órbita, son fundamentales para poder obtener un diseño que realmente pueda satisfacer estas necesidades.

En la primera parte de este capítulo se mencionarán algunos parámetros que se tomaron como base para la generación del diseño. Tomando como referencia estos datos, en la segunda parte se describen los componentes y configuraciones de componentes con los cuales se pretende elaborar el subsistema, dando una breve descripción de estos y su funcionamiento.

## 3.2 Parámetros de diseño

Los parámetros seguidos para el diseño de este subsistema fueron planteados de acuerdo a las tareas que pretende realizar el nanosatélite experimental.

El subsistema de comunicaciones que se pretende elaborar forma parte de un CubeSat de tamaño 3U. Debido a las dimensiones de este tipo de satélites y para este caso en particular el subsistema debe estar contenido en una tarjeta PCB de 9.4x9.6 cm. El subsistema debe contar con un bus de datos, el cual es un conector de 50 pines, el cual se encarga de mantener la comunicación entre todos los subsistemas del satélite.

El protocolo que se adoptó para la comunicación entre subsistemas fue I2C, por lo tanto el dispositivo que se debía elegir para controlar el flujo de datos en el subsistema debería manejar entre sus periféricos un módulo capaz de manejar este protocolo para facilitar la programación del subsistema.

Como ya se mencionó, el ambiente espacial presenta diversos inconvenientes para mantener en buenas condiciones a un satélite. Uno de los eventos con mayor frecuencia en CubeSats es el efecto conocido como “Latch Up”, una especie de corto circuito producido por cierto tipo de radiación emitida por el Sol. Este efecto en ocasiones genera corrientes tan elevadas que el daño en componentes electrónicos es inevitable. El subsistema debe contener un circuito capaz de proteger al subsistema de este efecto.

Otra acción que se toma como medida de protección al subsistema es la de colocar sensores de temperatura. Mediante estos sensores de temperatura se pretende vigilar la temperatura del subsistema, ya que un incremento considerable en la temperatura del subsistema también podría dañar el funcionamiento de sus componentes.

Se ha observado en misiones anteriores que componentes de alta escala de integración son más susceptibles a daños provocados por efectos de radiación. Motivo por el cual se pide que los componentes que integran el subsistema sean de la menor escala de integración posible, ejemplo de estos son los microcontroladores.

El subsistema de comunicaciones debe contener un módulo con el cual pueda comunicarse a través de un cable con una computadora, con esta comunicación se busca por lo general evitar el uso de la parte de comunicación inalámbrica del subsistema. Este módulo tiene que realizar la misma función de comunicar al satélite con el software de estación terrena pero por medio de esta vía alámbrica.

En lo que se refiere a la selección de transceptores, en base a parámetros observados en otras misiones, se necesita que cumpla con una potencia de transmisión de 1W, que opere en bandas de radioaficionados en el rango UHF o VHF y que puedan modular en alguno de estos esquemas de modulación: 2FSK, 4FSK o GMSK.

### 3.3 Descripción del diseño propuesto

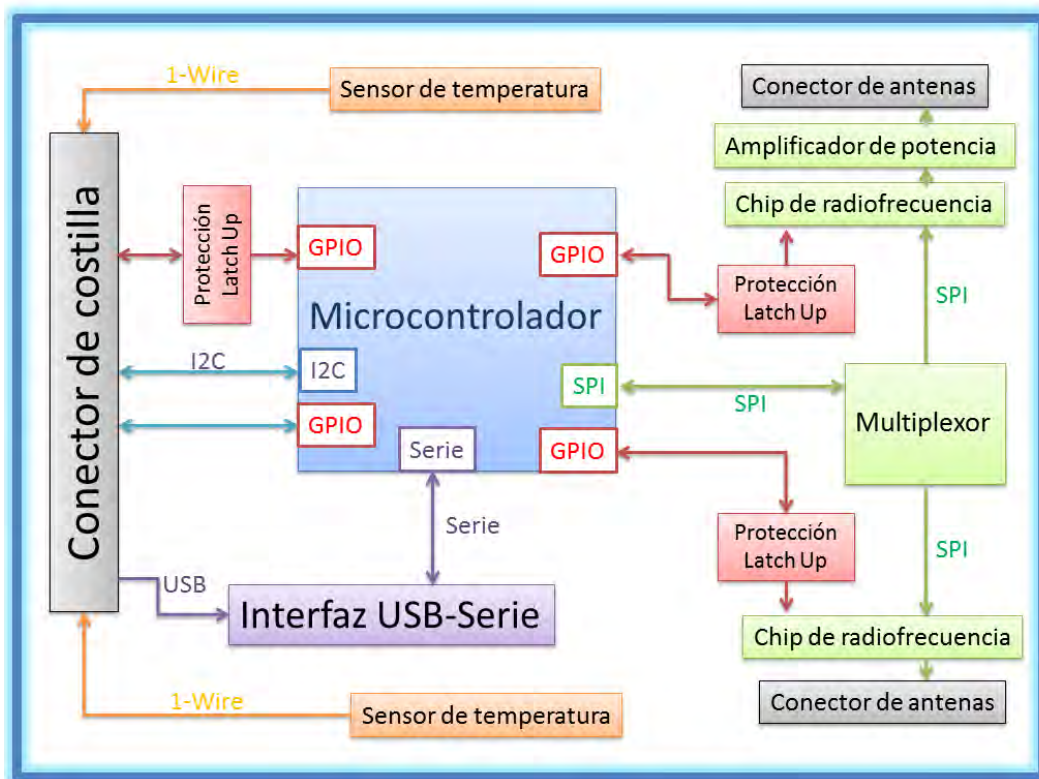


Figura 3.1 Diagrama a bloques del subsistema de comunicaciones.

La figura 3.1 muestra un diagrama a bloques de lo que es el subsistema de comunicaciones propuesto. El subsistema está conformado por un circuito integrado que realiza la conversión de protocolo USB a protocolo Serie, con el fin de realizar la comunicación entre el satélite y el software de estación terrena de manera alámbrica; cuenta con circuitos protectores contra sobre-corriente, los cuales están colocados de manera que puedan monitorear las corrientes eléctricas en los circuitos más vulnerables a radiación; sensores de temperatura, cuyo objetivo es sensar las temperaturas de los componentes de mayor uso en el subsistema; dos chips de radiofrecuencia, que por medio de protocolo SPI se comunican con un microprocesador y se encargan de transmitir y recibir comandos de forma inalámbrica, contando uno de ellos con un amplificador de potencia para lograr una potencia de 1W. Por último cuenta con un microcontrolador cuya tarea es el controlar el flujo de información y discernir señales que provienen de dentro y fuera del subsistema, por medio de protocolos como lo son I2C, SPI o UART, siendo este componente clave fundamental de nuestro subsistema.

En cuanto a la interacción que tiene este subsistema con los demás, se realiza por medio del conector de 50 posiciones. Este conector, lleva consigo los voltajes de alimentación, provenientes del subsistema de potencia; líneas de comunicación I2C para comunicarse con computadora de vuelo; línea de un protocolo llamado 1 WIRE, para la lectura de los sensores de temperatura y líneas conectadas a GPIO'S que hacen la función de banderas, para avisarle a computadora de vuelo y al subsistema de potencia si existe una posible sobrecorriente.

El subsistema opera dentro de la bandas VHF (Very High Frequency) y UHF (Ultra High Frequency), eligiendo las frecuencias entre los rangos de frecuencias amateur. Lo que simplifica trámites y documentación en lo que se refiere a permisos por parte de instancias regulatorias de telecomunicaciones, pues las frecuencias elegidas forman parte de un conjunto de frecuencias libres.

Respecto al sistema de antenas, no es crítico el tener unas antenas con una gran directividad, ya que el rango de frecuencias lo permite, teniendo menos conflictos en lo que se refiere a la parte de la elaboración de antenas; así como el no requerir de un gran ancho de banda y una velocidad grande para transmitir.

El diseño propuesto para este trabajo, se basa en una selección de componentes COTS, los cuales fueron el resultado de una larga búsqueda en misiones realizadas con anterioridad y en catálogos de empresas dedicadas a la distribución y desarrollo de componentes electrónicos. Estos componentes se eligieron ya que cumplían con los parámetros requeridos por la misión.

Para este proyecto, al no contar con una estación terrena con grandes capacidades, se optó por el simular una, mediante la generación de tarjetas PCB, las cuales contengan transmisores y receptores iguales a los del subsistema de comunicaciones del satélite. Con estas tarjetas se pretenden realizar las pruebas necesarias para la validación del subsistema. Posteriormente al lanzamiento se prevé la compra de una estación terrena para tener contacto con nuestro satélite y con otros satélites pequeños de órbita baja, los cuales operen en rangos de frecuencias VHF, UHF y banda S. Esta última banda de frecuencias comienza a ser muy utilizada en los últimos años, debido a los avances en cuanto a sistemas de apuntamiento de satélites pequeños.

### 3.3.1 Interfaz de USB a UART

La razón de utilizar una interfaz entre el protocolo de comunicación USB y el de comunicación Serial UART, es el tener una comunicación alámbrica entre el subsistema de comunicaciones con una computadora, facilitando así la interacción entre la computadora y el microcontrolador del subsistema.

Es de suma importancia tener esta alternativa de comunicación ya que si se requieren cambios de software, como actualizaciones o el monitoreo de algún subsistema del satélite en específico, se puede lograr mediante esta interfaz.

Además de tener un propósito preventivo, ya que se pueden presentar situaciones donde no se pueda transmitir vía inalámbrica, por lo que es forzoso utilizar una interfaz alámbrica. Esta interfaz está diseñada para ser utilizada en Tierra, pues al ser liberado el satélite al espacio, esta comunicación no volverá a ser utilizada.

Para el diseño de esta interfaz se utilizó el circuito integrado FT232R de la compañía *Future Technology Devices International Ltd*, el cual simplifica el diseño, ya que no se necesita de muchos componentes externos.

En cuanto al propósito del satélite experimental, el circuito como se observa en la Figura 3.2, se encargará de convertir las señales de datos del protocolo USB mediante el uso de un DPLL, generando señales de reloj y de datos como parámetros a la máquina de interfaz serial (SIE), la cual hace la conversión de datos serie- paralelo y viceversa de USB de acuerdo a la especificación USB 2.0, además de realizar un CRC de los datos. Para después ser almacenados en un buffer de recepción.

A su vez la trama de datos es pasada por la máquina de protocolos USB, donde se generan los comandos para el control de los parámetros funcionales de UART. Los cuales son parámetros para el controlador de los buffers de recepción de entrada y salida. Este controlador junto con el controlador UART manipulan los datos de entrada y salida de los buffers para así acoplarlas a la interfaz RS232. La tasa del baudaje es proporcionada por el generador en este último bloque [16].

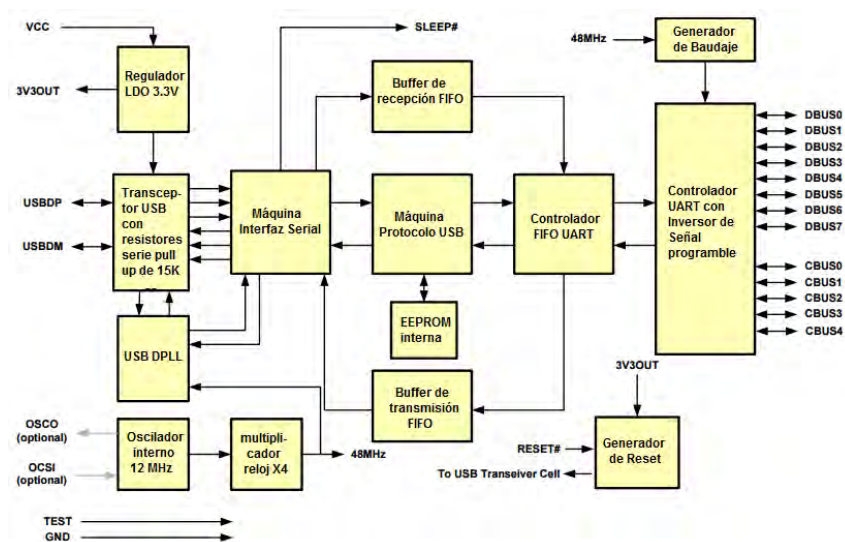


Figura 3.2 Diagrama de bloques del circuito FT232R

Diseño electrónico del FT232R para el subsistema de comunicaciones.

El circuito FT232R es capaz de convertir datos de USB a diversas interfaces seriales, como lo son el RS232, RS422 Y RS485. Sin embargo, para esta aplicación no es necesario el tener todas las líneas del RS232, pues el manejo de datos se realizará mediante un microcontrolador. En este trabajo se propusieron 4 líneas para la comunicación serial: RX, TX, CTS y RTS.

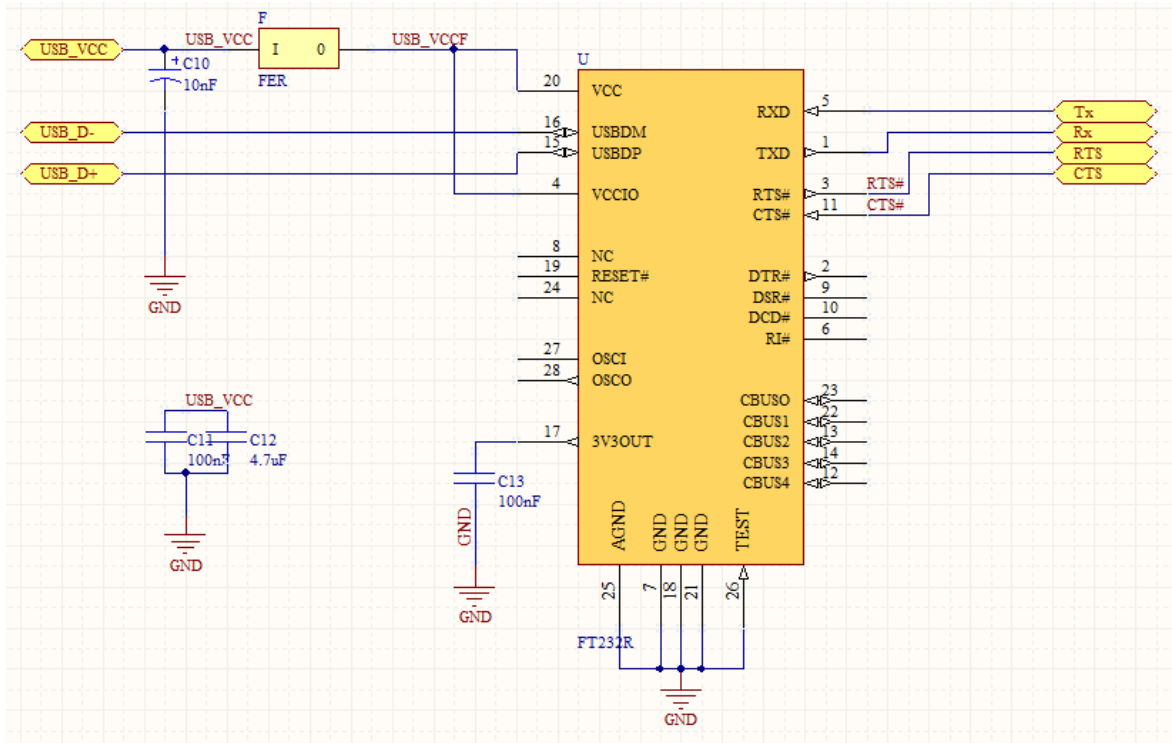


Figura 3.3 Diseño electrónico para el FT232R

Como se puede observar en la Figura 3.3 la alimentación del circuito se realiza directamente del voltaje que trae el conector del cable USB, siendo filtrada por medio de capacitores y una barra de ferrita, no se le colocan resistencias de pull-up o pull-down ya que las tiene integradas el circuito. El circuito es capaz de generar un voltaje de 3.3V y un reloj de 12MHz para alimentar a algún circuito externo. Además de tener la posibilidad de poner al circuito en modo de bajo consumo, que para nuestra aplicación no es necesario, pues al momento de ocuparlo es cuando se energizará, no habrá otra posibilidad.

El pin de alimentación de este circuito además de proveer el voltaje para el circuito, será conectado a un pin del microcontrolador, de forma que sirve como señal indicadora de que el sistema esta conectado vía USB.

### 3.3.2 Sensor de temperatura DS18B20

Se decidió colocar este tipo de sensores con la finalidad de ir monitoreando los cambios en la temperatura de la tarjeta, ya que al existir altas temperaturas se corre el riesgo de una avería o la pérdida definitiva de alguno de los componentes. Con los datos arrojados por estos sensores se podrían tomar medidas preventivas que eviten este tipo de incidentes.

El circuito que funge como termómetro es el DS18B20 de *Maxim Integrated* el cuál es un termómetro digital, capaz de entregar temperaturas en grados Celsius con tan solo 9 bits. Abarca un rango de operación de  $-55^{\circ}\text{C}$  hasta  $125^{\circ}\text{C}$  y puede medir temperaturas precisas desde los  $-10^{\circ}\text{C}$  hasta los  $85^{\circ}\text{C}$ ., además de poder ser alimentado con la misma línea de datos.

Utiliza el protocolo de comunicación 1-Wire, el cual como su nombre lo indica, consta de una sola línea de datos con un microcontrolador.

Cada DS18B20 contiene un código serial único de 64bits, lo cual permite tener más de un circuito en el mismo bus de comunicación. Funciona a manera de esclavo respondiendo a comandos mandados por el Maestro, siendo uno de estos el leer temperatura la cual está almacenada en dos registros de 8 bits [17].

Diseño electrónico DS18B20 para el subsistema de comunicaciones.

El diseño para este circuito es de lo más simple, al tener una sola línea de comunicación con el microcontrolador ahorra componentes, teniendo en sí, solo un capacitor de filtrado y una resistencia de pull-up la cual asegura el estado que está en la transmisión de datos.

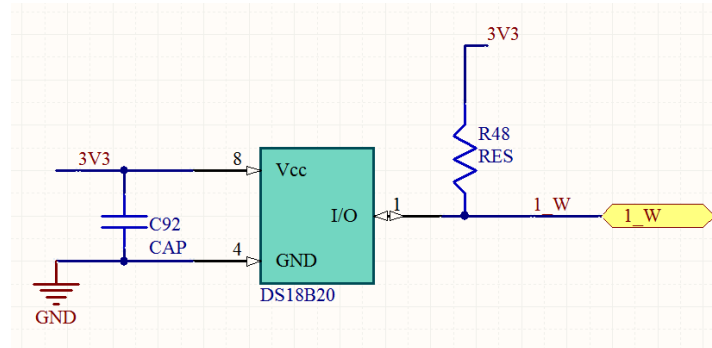


Figura 3.4 Diagrama de DS18B20

### 3.3.3 Protecciones contra efecto Latch-up

El circuito de protección contra efecto Latch Up se incluye en el diseño ya que es el efecto más frecuente y que más repercusiones causa a los circuitos electrónicos en el espacio, siendo más propensos a él los componentes de mayor escala de integración.

Este efecto se podría considerar como un corto circuito; generando así que se dispare la corriente en la tarjeta. En algunos casos, este efecto suele ser una condición temporal, sin embargo, en otros

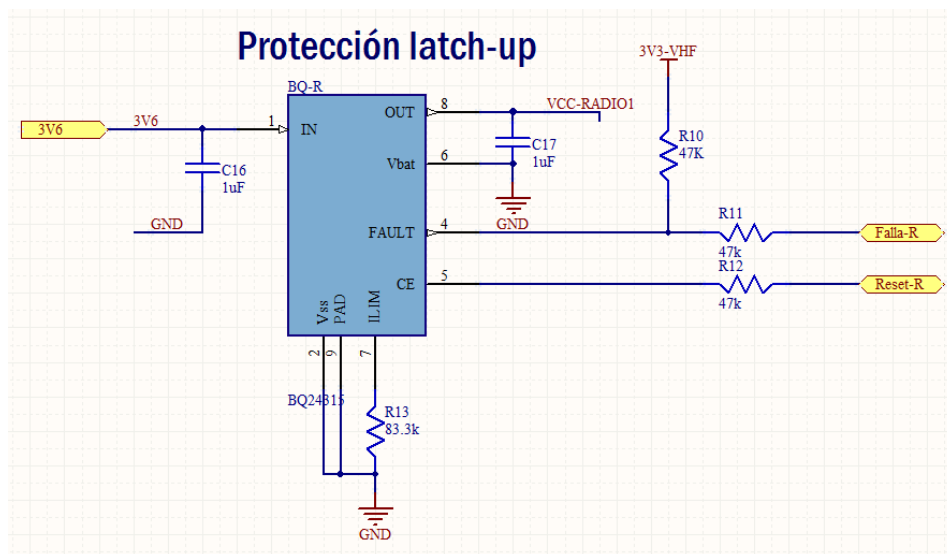
puede provocar hasta una falla total, generando que la electrónica sea dañada de manera permanente.

Para contrarrestar este problema se sensa la corriente que se le suministra a los componentes más críticos. Dado que cada componente utilizado en la tarjeta tiene un máximo valor de corriente distinto, se tomó la decisión de realizar dos circuitos de protección; uno para corrientes mayores de 300mA y otro para menores a esa corriente.

El circuito abre la alimentación del componente monitoreado, dejando pasar un determinado tiempo, para que se regularice el voltaje y corriente en el circuito debido al efecto Latch Up y vuelve a alimentarlo.

*Diseño electrónico del circuito mayor a 300mA.*

Para esta protección se encontró un circuito integrado de la compañía *Texas Instruments*, el BQ24314. El cual está diseñado para dar protección a baterías de Ion-Litio. El circuito monitorea constantemente el voltaje y la corriente de entrada y en caso de una sobre corriente o sobre voltaje abre la alimentación por medio de un switch interno [18]. El circuito intenta 15 veces volver a alimentar la carga, sin embargo, si persiste la condición de sobre corriente o sobre voltaje desenergiza definitivamente a la carga. Teniendo que aplicar un reset al circuito para poder alimentarla nuevamente.



*Figura 3.5 Diseño del BQ24314*

La condición de sobre corriente es generada cuando la corriente que demanda la carga es mayor a la corriente programada en el circuito. Esta corriente de referencia es programada por medio de una resistencia externa al circuito, dada de manera aproximada por la siguiente ecuación:



$$R_{LIM} = \frac{25}{I_{OCP}}$$

Donde:

$R_{LIM}$  = resistencia que limita la corriente

$I_{OCP}$  = Corriente máxima soportada por la carga

Diseño electrónico del circuito menor a 300mA.

Para este diseño, se requirió de una mayor cantidad de componentes, ya que las corrientes que se necesitan sensor son muy pequeñas, no se encontró un circuito como el BQ24314 para detener el suministro de energía, así que se hizo un arreglo propio, el cual fuera capaz de desconectar a la carga de la alimentación. La forma en que se explicará este diseño consta primeramente del desglose de componentes utilizados, para después ser descrita la funcionalidad del arreglo completo.

MAX4372

Este circuito de *Maxim Integrated* es un sensor de corriente, el cual monitorea el flujo de corriente que viene de la alimentación. Esta corriente fluye a través de una resistencia de sensado, generando un voltaje de sensado. Ya que las terminales de los amplificadores operacionales se consideran como altas impedancias, el voltaje en la entrada negativa de A1 es  $V_{in} - V_{sense}$ . A1 fuerza a su terminal positiva a acoplarse a su terminal negativa, dado por entendido, el voltaje de sensado sería  $R_{g1}(V_{in} - V_1)$ . Esto genera una corriente a través de  $R_{g1}$  igual a  $V_{sense}/R_{g1}$  [19]. El transistor y el espejo de corriente, amplifica la corriente por un factor B, haciendo que la corriente a la salida del espejo de corriente sea igual a:

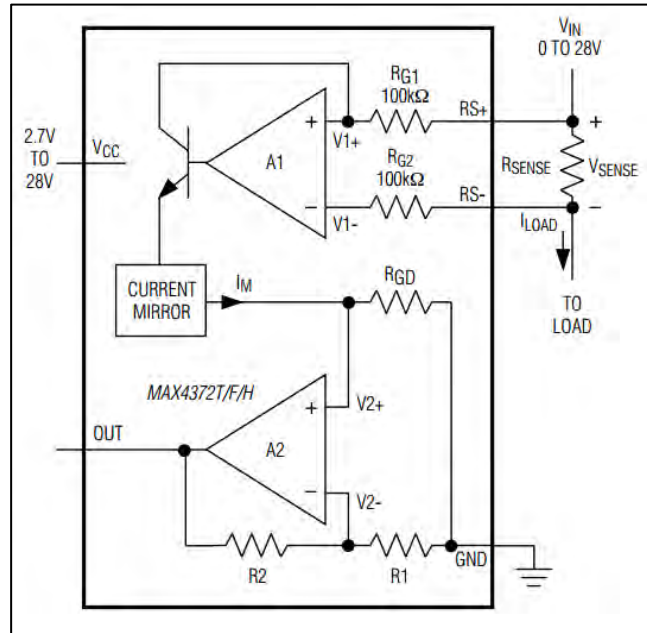


Figura 3.6 Diagrama simplificado del MAX4372

$$I_M = \frac{\beta V_{SENSE}}{R_{G1}}$$

En la terminal positiva de A2 se encuentra una alta impedancia, así que la corriente fluye a través de Rgd, dando como resultado

$$V_{2+} = \frac{R_{GD}\beta V_{SENSE}}{R_{G1}}$$

R1 y R2 ajustan la ganancia del bucle para A2, amplificando el voltaje de la terminal positiva, obteniéndose:

$$V_{2+} = \frac{R_{GD}\beta V_{SENSE}}{R_{G1} \left(1 + \frac{R_2}{R_1}\right)}$$

Teniendo la ganancia del dispositivo igual a

$$V_{OUT}/V_{SENSE} = \frac{R_{GD}\beta \left(1 + \frac{R_2}{R_1}\right)}{R_{G1}}$$

El proveedor proporciona una tabla de resistencias de sensado sugeridas para la elaboración del arreglo según la escala que se desea medir, la cual se puede observar a continuación.

Corriente a Carga Completa I <sub>carga</sub> (A)	R <sub>sense</sub> (mΩ)	Ganancia (V/V)	Voltaje de salida a escala completa V <sub>out</sub> (V)
0.1	1000	20	2.0
		50	5.0
		100	10.0
1	100	20	2.0
		50	5.0
		100	10.0
5	20	20	2.0
		50	5.0
		100	10.0
10	10	20	2.0
		50	5.0
		100	10.0

Tabla 3.1 Resistencia sugerido por el fabricante.

### LM6511

Este dispositivo de *Texas Instruments* es un comparador de voltaje. Tiene una salida a colector abierto que permite ser compatible con variedad de voltajes de alimentación a la salida.

El comparador de voltaje, utiliza un amplificador operacional en lazo abierto. Basta con que una de las dos señales de entrada sea ligeramente mayor para que cause que la salida del amplificador sea máxima, ya sea positiva o negativa [20]. Demostrándose con la siguiente ecuación

$$V_{OUT} = A(V_1 - V_2)$$

Donde

$$V_{OUT} = \text{voltaje de salida}$$

$$V_1 = \text{voltaje de entrada en la terminal inversora}$$

$$V_2 = \text{voltaje de entrada en la terminal no inversora}$$

$$A = \text{Ganancia del amplificador en lazo abierto}$$

En el caso de que uno de los dos voltajes sea fijo, la salida va a depender del comportamiento del otro voltaje. Por ejemplo: en caso de que en la terminal inversora se tenga el voltaje de referencia, cuando en la terminal no inversora el voltaje sea menor al voltaje de referencia la salida del amplificador se irá al voltaje de saturación negativa. Por otro lado, si es superior al voltaje de referencia se tiene una salida con el voltaje de saturación positivo.

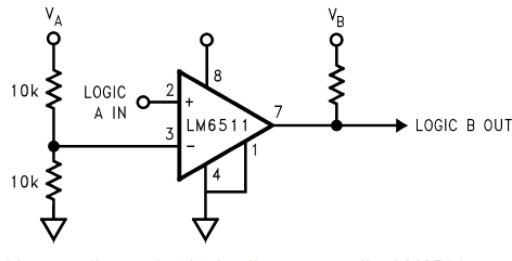


Figura 3.7 Diseño del comparador.

#### MIC9404

El MIC9404 es un switch de la compañía *Micrel* diseñado para operar entre los 1.7v y los 5.5v, cuenta con una terminal de habilitación la cual debe llevar una resistencia de pull down. Es un empaquetado muy pequeño (1.2mmx1.2mm) y de solo cuatro terminales.

#### 74HC112

Este componente es un Flip-Flop del tipo JK de *Texas Instruments*. Un flip flop es un dispositivo de dos estados, que funciona como memoria básica para las operaciones lógicas secuenciales. Son ampliamente usados para el almacenamiento y transferencia de datos digitales.

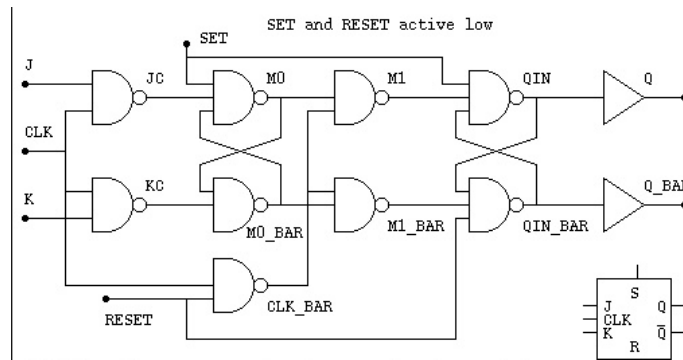


Figura 3.8 Estructura de un flip flop JK.

Al ser un dispositivo generado por compuertas lógicas NAND, el comportamiento a la salida es fiel reflejo de las entradas del flip flop. Las cuales se muestran en la tabla siguiente:

Tabla de verdad

Entradas					Salidas	
S	R	CP	J	K	Q	Q̄
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H	H
H	H	↓	L	L	No cambia	
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	Toggle	
H	H	H	X	X	No cambia	

H= Nivel Alto    X= No importa  
L= Nivel Bajo    ↓= Transición de Alto a Bajo

Tabla 3.2 Tabla de verdad de 74HC112.

### Arreglo del circuito protector Latch Up

Al haber abordado componente por componente de este circuito, lo que sigue es explicar el desarrollo del circuito protector completo. El voltaje de entrada a la carga es el voltaje que entra al switch MIC9404, en condición inicial el switch está habilitado, lo que hace que pase el voltaje por él. La alimentación sigue por la resistencia de sensado del MAX4372 y pasando esa resistencia se alimenta la carga (circuito a proteger). Cuando el MAX4372 detecte una corriente elevada y pueda generar un voltaje lo suficientemente grande como para rebasar el voltaje de referencia a la entrada no inversora del comparador, el comparador cambiará su salida del voltaje de saturación positivo al voltaje de saturación negativo. Dicha salida es suministrada como la entrada del reloj del Flip Flop, que a su vez, tiene todas sus entradas en un nivel alto para que, acorde a la tabla 3.2 se obtenga un cambio de estado de alto a bajo en la salida; al estar conectado a la habilitación del switch, hace que se desenergice la carga.

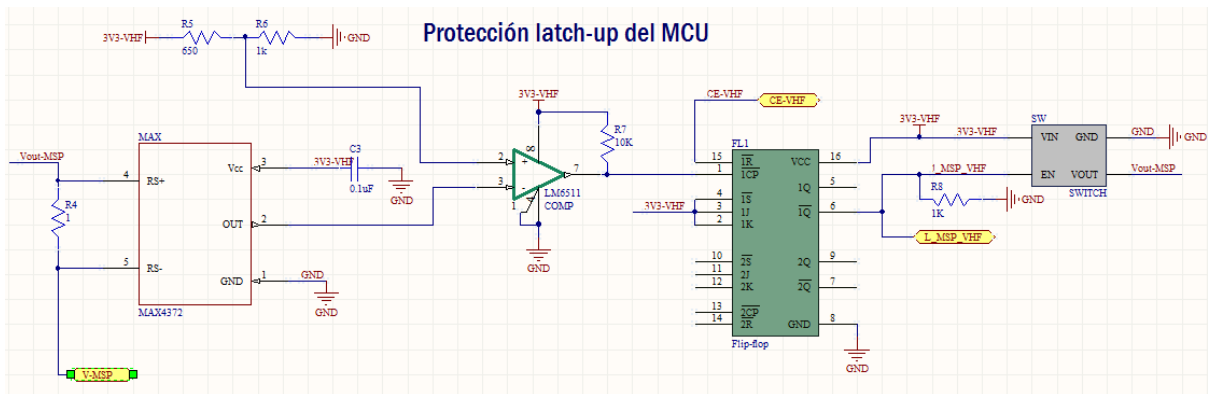


Figura 3.9 Diseño de protección Latch Up menor a 30mA.

El arreglo de protección cuenta con dos señales, la primera de ellas es la señal de falla, esta señal tiene el propósito de avisar al subsistema de potencia que el efecto Latch Up ocurrió, siendo tomada del cambio de estado a la salida del Flip Flop; la segunda es la señal de Reset, esta señal es tomada del Reset del Flip Flop, ya que si cambia de alto a bajo esta terminal, regresa a condiciones iniciales, regresando a alto la salida del comparador la del Flip Flop y la del habilitador del switch, las cuales corresponden a su condición inicial, energizando de nuevo el circuito protegido.

### 3.3.4 Cambiador de nivel NTB0104

Este dispositivo es un cambiador de nivel de voltaje, de cuatro puertos, dos alimentaciones y un habilitador. Este circuito es bidireccional y su principal función es cambiar el nivel de voltaje entre la entrada y la salida [21].

En este caso, el circuito fue requerido para poder reducir el nivel de voltaje a la salida del FT232R, el cual nos da respuestas en 5V, mientras que el microcontrolador a utilizar tiene como máximo 3.3V. De no hacer este paso entre un voltaje y otro podríamos dañar el microcontrolador al alimentarlo con un voltaje mayor.

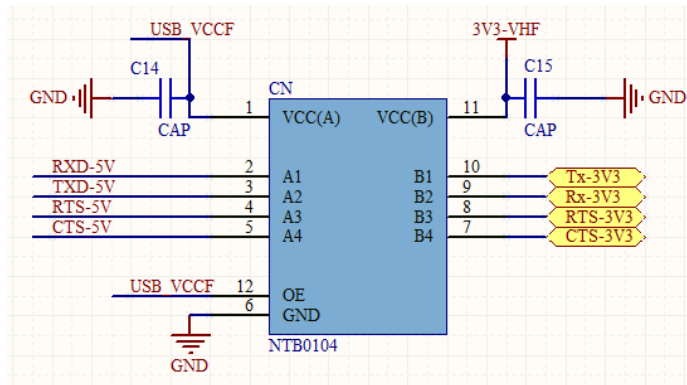


Figura 3.10 Diseño del NTB0104

### Diseño electrónico del NTB0104

Respecto al diseño del cambiador de nivel, se conectó un puerto del circuito a las salidas del FT232, alimentando ese puerto con 5V, mientras que el otro puerto va dirigido a las entradas del microcontrolador, siendo a su vez, alimentado con 3.3V. Para el uso de este componente solo fueron necesarios un capacitor de filtrado a la entrada de cada una de las alimentaciones.

### 3.3.5 Transceptor ADF7021

Este dispositivo es un transceptor de alto rendimiento, bajo consumo de energía y maneja esquemas de modulación 2FSK/3FSK/4FSK. Está diseñado para operar en banda angosta, bandas ISM sin licencias y en las bandas con licencia en rangos de frecuencia de 80MHz a 650MHz y de 862MHz a 950MHz. El dispositivo cuenta con filtros de transmisión de datos del tipo coseno cuadrado y de tipo gaussiano para mejorar aplicaciones en eficiencia espectral y anchos de banda estrechos.

Se puede elaborar un transceptor completo solo con unos cuantos componentes externos, lo cual lo hace adecuado cuando se busca economizar en precio y área.

La parte del transmisor cuenta con dos osciladores controlados por voltaje (VCO) y un PLL fraccional de bajo ruido con una resolución a la salida menor a 1ppm. Uno de los VCO utiliza un circuito interno LC tipo tanque para rangos de frecuencia de 431MHz a 475MHz y de 862MHz a 950MHz, mientras que el otro VCO utiliza un inductor externo como parte del circuito tanque para el rango de frecuencias de 80MHz a 650MHz. Los dos operan al doble de la frecuencia fundamental para reducir emisiones espurias.

El PLL permite a este dispositivo ser usado en sistemas de espectro ensanchado por salto de frecuencia (FHSS).

La potencia de salida del transmisor es programable en 63 pasos de -16dBm a +13dBm. La frecuencia de operación, espaciado de canales y la modulación son programables utilizando una interfaz de 3 líneas, opera con un voltaje de entrada de 3.3V y puede estar en bajo consumo cuando no sea utilizado.

El receptor cuenta con una arquitectura baja del tipo Frecuencia Intermedia (IF) lo que minimiza su consumo y componentes externos. Cuenta con un filtro IF, el cual, tiene ya programados anchos de banda de 12.5kHz, 18.75kHz y 25kHz.

Cuenta con un convertidor analógico digital que provee la lectura de un sensor de temperatura integrado, entrada analógica externa, voltaje de batería y una señal que indica la intensidad de la señal recibida (RSSI) [22].

Además de contar con pines para poderse comunicar mediante protocolos serial y SPI con un microcontrolador.

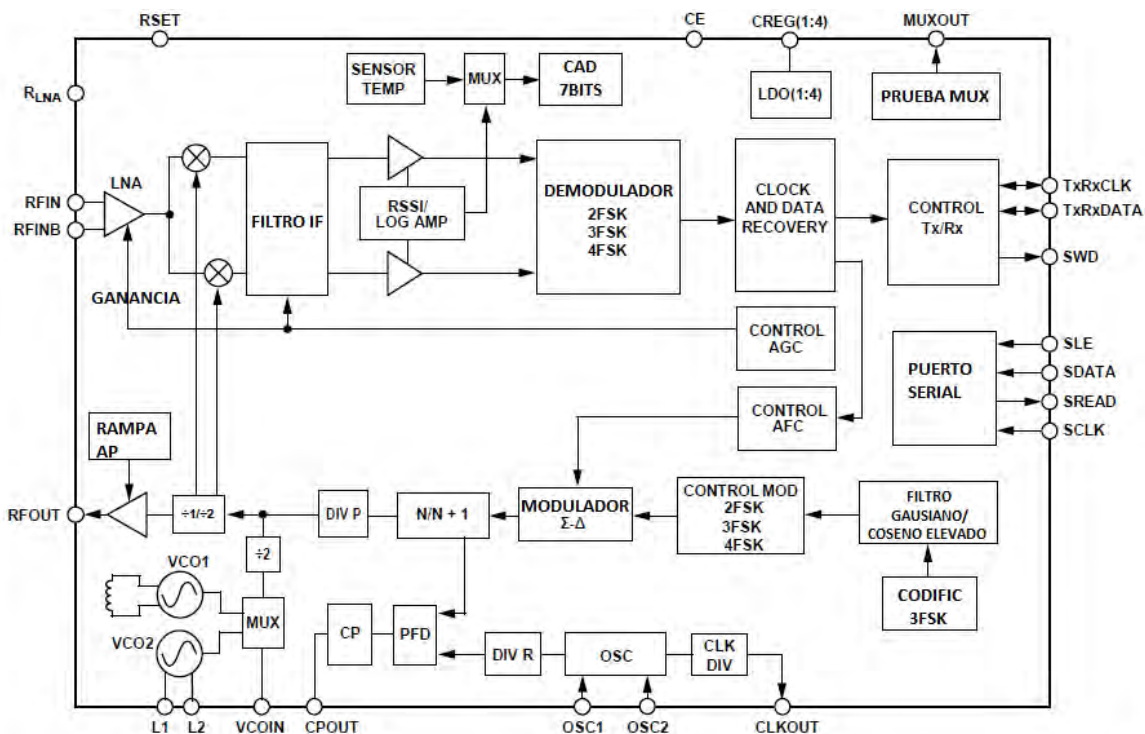


Figura 3.11 Diagrama a bloques del ADF7021

### Diseño electrónico del ADF7021

Para el diseño del transceptor ADF7021 se requirieron muy pocos componentes externos, en los que se incluyeron un inductor para completar el circuito tanque LC del oscilador controlado por voltaje que se utiliza a bajas frecuencias, un cristal que funge como entrada de referencia y capacitores, resistores e inductores que forman parte del acoplamiento hacia la antena y para filtros que requiere el transceptor. Los valores de dichos componentes fueron obtenidos mediante un software que el proveedor pone a disposición de los usuarios. El programa se encuentra en su página y se llama ADIsimSRD Design Studio. Dicho programa, como se puede observar en la figura 3.12, arroja los valores de los componentes externos, facilitándonos así la elaboración del diseño.

El programa pide como parámetros: el modelo de transceptor a utilizar, rango de frecuencia, ancho del canal, la velocidad a transmitir, tipo de modulación y si se va a ocupar solo como transmisor o como transceptor, datos que se observan en la tabla siguiente:

Parámetro	Valor
Dispositivo	ADF7021
Frecuencia central	146 MHz
Velocidad de datos	9.6Kbps
Modulación	GFSK
Ancho de canal	200KHz

Tabla 3.3 Parámetros del diseño del ADF7021.

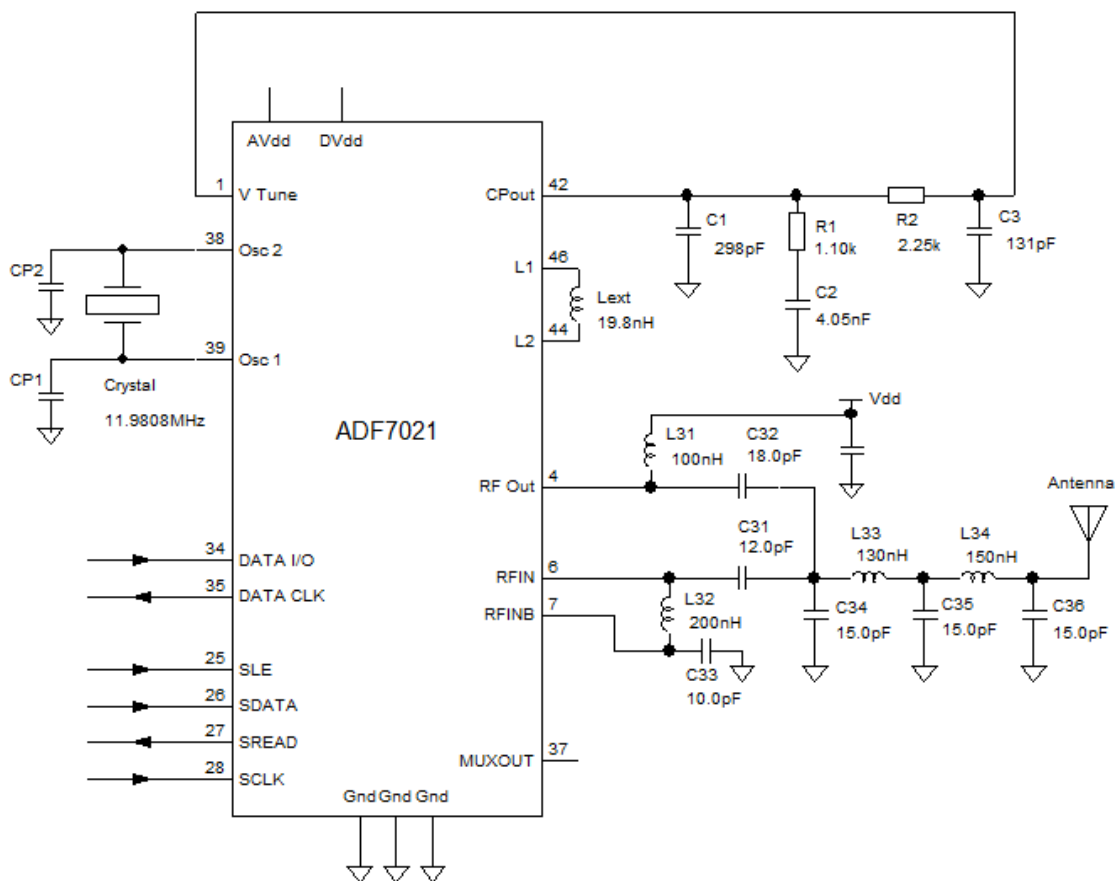


Figura 3.12 Diseño del ADF7021

### 3.3.6 Transceptor CC1101

El CC1101 es un transceptor de bajo costo y pequeñas dimensiones (4x4mm), diseñado para operar en bandas de frecuencias industriales, científicas y médicas (ISM) 300-348MHz, 387-464MHz y 779-928MHz. Está integrado por un modem base banda, soportando formatos de modulación como



2FSK, 4FSK, GFSK y MSK además de tener una tasa de velocidad configurable, pues tiene un rango de 0.6 a 600 kbps.

Sus parámetros de funcionamiento principal, configuración y sus buffers de 64 bytes de transmisión y recepción pueden ser controlados mediante protocolo SPI, siendo utilizado junto a un microcontrolador y muy pocos componentes pasivos adicionales.

Cuenta con un indicador digital de potencia de señal recibida RSSI, son programables los anchos de banda del canal, así como la potencia de salida hasta de +12dBm. Incluye soporte para las configuraciones del canal, manejo de paquetes y almacenamiento de datos.

El CC1101 está integrado por un receptor bajo de frecuencias intermedias (IF). La señal recibida es amplificada por un amplificador de bajo ruido y convertida en cuadratura (I/Q) a frecuencias intermedias. Las señales I/Q son digitalizadas por convertidores analógicos digitales (ADC). El control automático de ganancia, el filtrado de canales, la demodulación y la sincronización de paquetes es realizada digitalmente.

La parte de transmisión está basada en la síntesis directa de la Radiofrecuencia (RF). El sintetizador de frecuencia incluye completamente un Oscilador Controlado por Voltaje (VCO) y un defasador de 90° para generar las señales I y Q.

Utiliza un cristal externo que genera una frecuencia de referencia para el sintetizador así como para los relojes de los ADC y toda la parte digital [23].

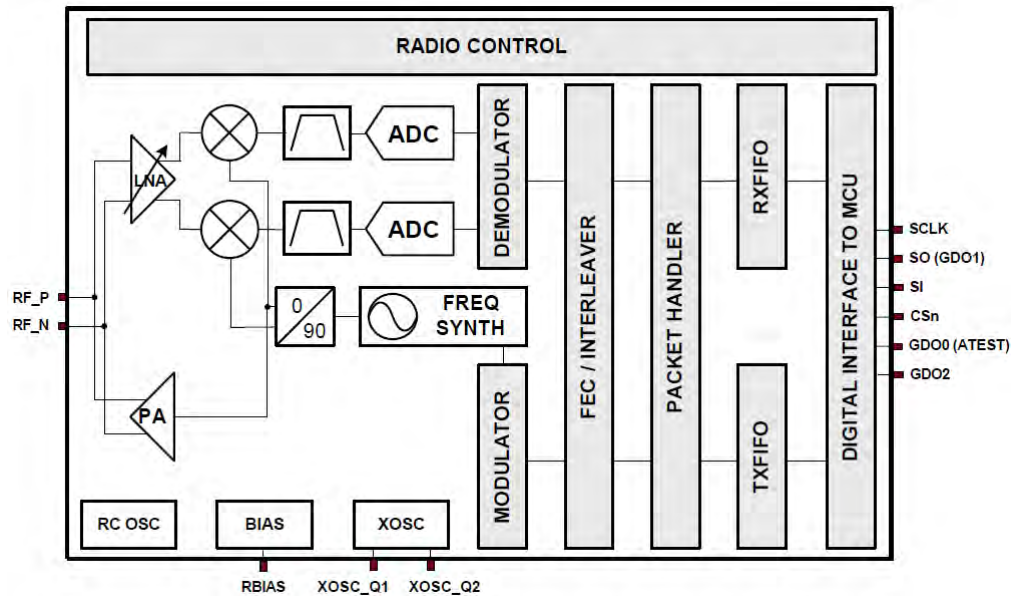


Figura 3.13 Diagrama a bloques del CC1101.

### Diseño electrónico del CC1101

La entrada y salida de la señal balanceada de radiofrecuencia comparten dos pines en común, para esto se necesita pasar de una señal de forma diferencial a una sola señal, para llegar a esto se tiene que diseñar un simple y barato acoplamiento y una red balun en el circuito impreso. Siendo este último el encargado de convertir la señal diferencial de RF a una sola. Los componentes que conforman esta red son capacitores e inductores que a su vez acoplan la impedancia de la carga a 50  $\Omega$ , lo cual facilita demasiado la salida que va a la antena.

El CC1101 necesita un cristal en el rango de frecuencias de 26 a 27MHz para generar su frecuencia de referencia, este cristal necesita de un par de capacitores de carga. Para frecuencias por debajo de los 870 MHz se recomienda un cristal de 26 MHz y para superiores de 869 MHz se recomienda uno de 27 MHz.

Para el diseño del acoplamiento del CC1101 se siguieron las recomendaciones que da el fabricante, pues ya hay un circuito propuesto para la frecuencia a la que se deseaba que funcionara, así como también recomienda los fabricantes de capacitores e inductores a utilizar.

En cuanto a los parámetros de diseño como la frecuencia de operación, ancho del canal a utilizar y tipos de frecuencia se observan en la siguiente tabla:

Parámetro	Valor
Dispositivo	CC1101
Frecuencia central	434 MHz
Velocidad de datos	100 KBauds
Modulación	GFSK
Ancho de canal	200KHz

Tabla 3.4 Parámetros de diseño del CC1101.

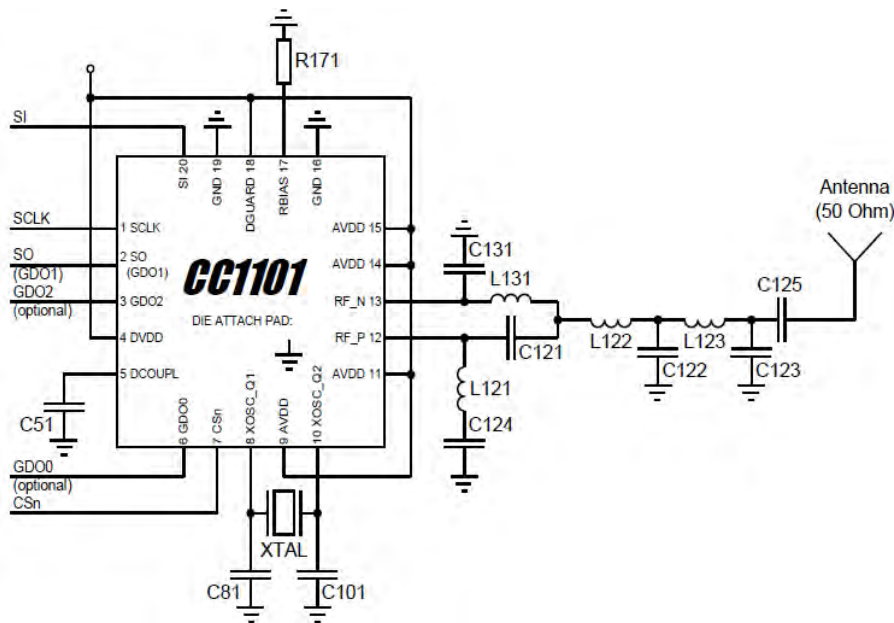


Figura 3.14 Diseño propuesto del transceptor CC1101.

### 3.3.7 Amplificador de potencia RFPA0133

Este amplificador de potencia se utilizó debido a que el chip de radiofrecuencia no generaba la potencia deseada a la salida. Este circuito se alimenta con 3.3V o 5V, es de alta eficiencia, más del 60% en un rango de frecuencias de 380MHz a 960MHz y cuenta con una ganancia programable con la cual puede obtener a la salida del circuito una potencia de entre 0.5W y 1W [24].

#### Diseño electrónico del amplificador RFPA0133

En cuanto al diseño de este circuito el proveedor cuenta con esquemáticos de dos tarjetas de prueba, una para los rangos de frecuencia que van de 860MHz a 930MHz y la otra de 430MHz a 470MHz. En las cuales propone valores de capacitores, resistores e inductores para su elaboración. Para este trabajo el diseño se basó en la tarjeta que corresponde a las frecuencias de 430MHz a 470MHz. En la siguiente figura se muestra el esquemático utilizado para su elaboración.

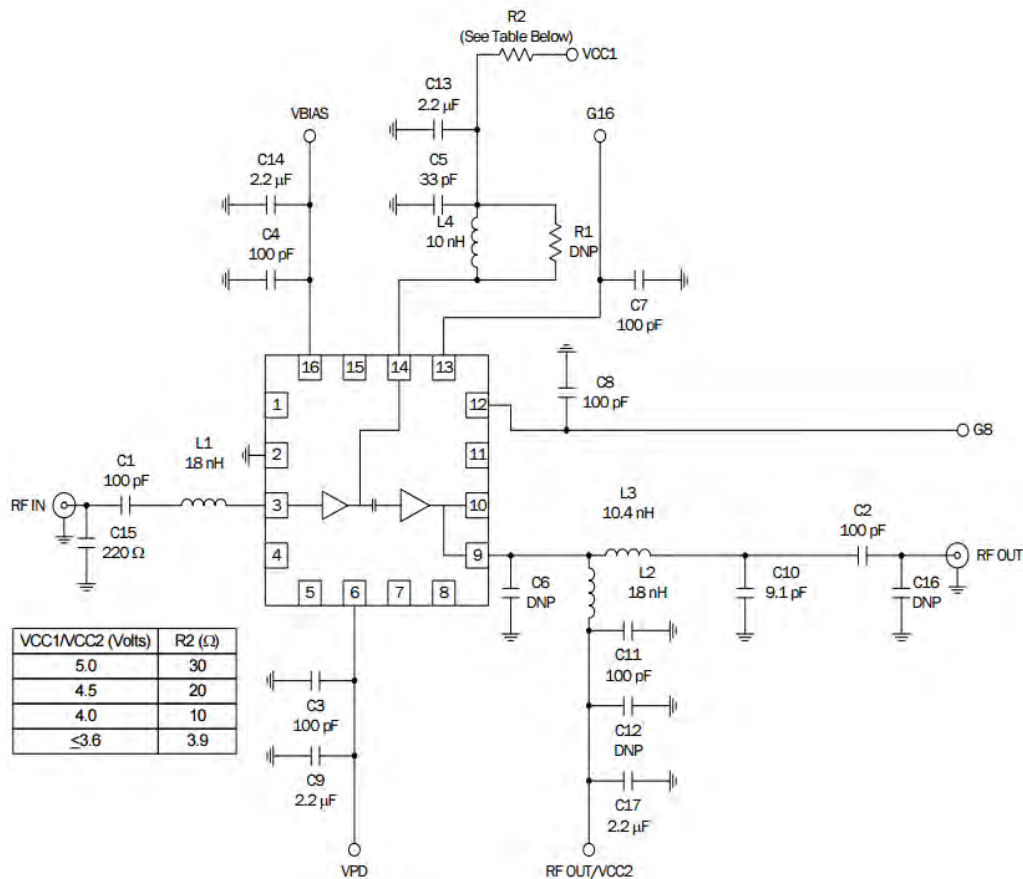


Figura 3.15 Diseño del RFPA0133

Además de contener el esquemático de la tarjeta de prueba, también nos sugiere la distribución de los componentes, pues al colocar pistas de mayor o menor tamaño se alteran los valores de inductancias requeridas, por ello es de suma importancia seguir las recomendaciones del fabricante al pie de la letra y colocar los componentes los más próximos al circuito integrado.

### 3.3.8 Microcontrolador MSP430F1611

Se eligió este microcontrolador ya que proviene de una familia de muy bajo consumo de energía, contienen varios periféricos para múltiples aplicaciones, además de ya haber sido lanzado en otras misiones de CubeSats como lo son SwissCube, Delfi-C3, Libertad-1 y Goliat teniendo un buen funcionamiento en dichas misiones.

La arquitectura del MSP430F1611 acompañada de sus 5 modos de bajo consumo lo hacen excelente para extender la vida de las baterías. Cuenta con un CPU del tipo RISC de 16 bits, registros de la misma longitud y generador de constantes que ayudan a una mayor eficiencia en el código.

Entre los periféricos con los que cuenta están contadores de 16 bits, convertidores Analógicos/Digitales de 12 bits, interfaces de comunicación serial síncrona y asíncrona, I2C y pines de entradas y salidas de propósito general. Además de incluir un módulo de emulación embebido, que permite realizar el debug y programación de una manera más fácil [25].

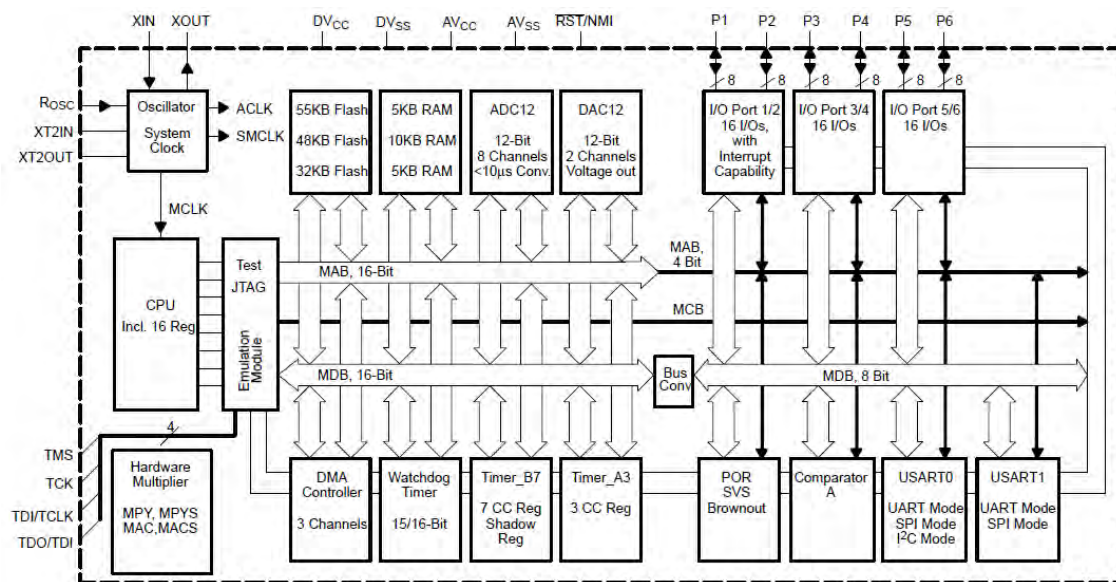


Figura 3.16 Diagrama a bloques del MSP430F1611

El microcontrolador, se podría decir, que es el “cerebro” del subsistema. Ya que este es el encargado de recibir y mandar todas las señales a tratar en el subsistema. Como se mencionó con anterioridad, el subsistema puede tener contacto con el exterior de dos formas: por vía cordón umbilical y por radiofrecuencia. La primera de ellas la integra el circuito FT232 que convierte la forma de comunicación de protocolo USB a Serial y el cambiador de nivel NTB0104 para la reducción del voltaje entrante al microcontrolador. La comunicación por radiofrecuencia implica la interacción directa de los transceptores con el microcontrolador vía puerto serial.

Para esta aplicación en particular se utilizaron los módulos USART0, USART1 y pines de propósito general.

### Diseño electrónico del microcontrolador MSP430F1611

Para el diseño y funcionamiento del MSP430F1611, como se puede observar en la figura 3.17 siguiente fueron necesarios: en la parte de la polarización se colocaron capacitores de filtrado a la entrada de la alimentación, así como un cristal de 8 MHz el cual se utiliza como un oscilador de uso interno. Por otro lado se dejan las líneas de programación JTAG del microcontrolador conectadas a un conector, por el cual, se programará más adelante.

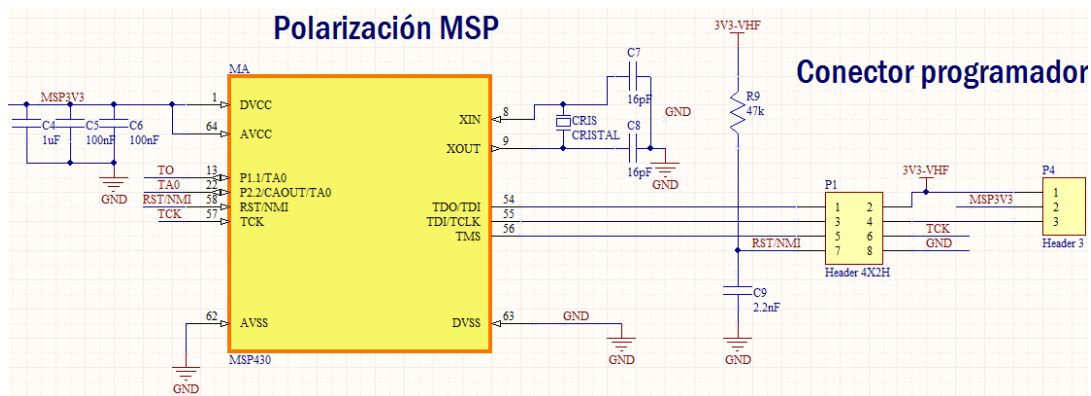


Figura 3.17 Parte de polarización y programación del MSP430F1611

La siguiente parte del diseño del MSP430F1611 es la que interactúa con alguno de los periféricos o pines de propósito general.

El microcontrolador tiene pines específicos para comunicarse con cualquiera de sus periféricos. En nuestro caso los pines que se utilizaron fueron los encargados de entablar el protocolo de I2C (pines 29 y 31), puerto serial (pines 32, 33, 34 y 35) y los encargados de manejar el protocolo SPI.

Los otros pines conectados al microcontrolador como se observa en la tabla 3.5 son banderas utilizadas en la comunicación serial (RTS y CTS); banderas de falla y reset de las protecciones contra efecto latch up de cada uno de los transceptores, así como de banderas que generan interrupciones necesarias para alguna rutina, tal es el caso de la bandera que se alimenta del voltaje del protocolo USB, pues con este pin de propósito general se configura el puerto USART0 a SPI o a protocolo Serial, puesto que se ocupan los dos tipos de comunicación en diferentes situaciones.

PIN	NOMBRE	DESCRIPCIÓN
2	P6.3/A3	Fault ADF
3	P6.4/A4	GDO0
4	P6.5/A5	GDO2
14	P1.2/TA1	Re-AP
15	P1.3/TA2	Fa-AP
16	P1.4/SMCLK	Reset_CC1101
17	P1.5/TA0	Fault_CC1101
24	P2.4/CA1/TA2	ADF_SWD

26	P2.6/ADC12CLK/DMAE0	USB_VCC
27	P2.7/TA0	INT_UHF_VHF
29	P3.1/SIMO0/SDA	I2C_SDA
31	P3.3/UCLK0/SCL	I2C_SCL
32	P3.4/UTXD0	RTS
33	P3.5/URXD0	CTS
34	P3.6/UTXD1	Tx
35	P3.7/URXD1	Rx
36	P4.0/TB0	FLAG1
37	P4.1/TB1	FLAG2
38	P4.2/TB2	FLAG3
40	P4.4/TB4	ADF_CE
41	P4.5/TB5	ADF_SLE
42	P4.6/TB6	ADF_SDATA
43	P4.7/TBCLK	ADF_SREAD
44	P5.0/STE1	ADF_SCLK
45	P5.1/SIMO1	MOSI
46	P5.2/SOMI1	MISO
47	P5.3/UCLK	SCLK
48	P5.4/MCLK	CSn
50	P5.6/ACLK	SW_AP
61	P6.2/A2	Reset_ADF

*Tabla 3.5 Pines utilizados del MSP430F1611.*

Para este proyecto, al no contar con una estación terrena con grandes capacidades, se optó por el simular una, mediante la generación de tarjetas PCB, las cuales contengan transmisores y receptores iguales a los del subsistema de comunicaciones del satélite. Con estas tarjetas se pretenden realizar las pruebas necesarias para la validación del subsistema.

Posteriormente se prevé la compra de una estación terrena para tener contacto con el satélite, una vez lanzado y con otros satélites pequeños de órbita baja, los cuales operen en frecuencias de las bandas VHF, UHF y S.

# Capítulo 4 Fabricación del Subsistema de Comunicaciones

---

## 4.1 Introducción

Al tratarse del diseño y la fabricación de un circuito impreso de dos capas, espesores estándar en cuanto a la tarjeta y el cobre, desde el principio se tuvo la idea de mandar a fabricar dicha tarjeta electrónica. Para lograr esto se debía contar con alguna plataforma que contuviera las herramientas necesarias para poder generar archivos que nos facilitara esta tarea.

Siendo de suma importancia el conocer todo el procedimiento que conlleva el realizar el diseño de un circuito impreso, ya que, se debe implementar cierta metodología para obtener como producto terminado la tarjeta PCB.

En este capítulo se abordará el cómo se desarrolló el diseño de la tarjeta electrónica, usando el software comercial Altium Designer. Se describirá su funcionamiento general, aspectos del cual nos serán de utilidad para generar nuestro circuito impreso, como el uso de librerías de componentes, el diseño de esquemáticos y finalmente el ruteo de componentes y pistas en un archivo PCB. Tomando las consideraciones necesarias para que en su elaboración se obtengan los mejores resultados.

## 4.2 Elaboración de la tarjeta electrónica.

Para dar inicio a la elaboración de la tarjeta impresa fue necesario el adentrarse al software de desarrollo que se eligió, en este caso Altium Designer. Al tener una propuesta de diseño bien clara lo que se necesitaba era plasmarlo en un circuito impreso capaz de contener todos los componentes y hacerlos interactuar en una sola tarjeta de tamaño 9.4x9.6 cm.

Dado que la tarjeta se mandaría a fabricar, era necesario tener en claro algunos parámetros que podrían retrasar su elaboración. El tomar en cuenta aspectos como lo son el largo y ancho de la tarjeta, el ancho mínimo de pistas, el espacio entre pistas y tamaño de perforaciones, llevó a elegir el empaquetado de componentes específicos, debido a que el ensamble del prototipo se hace manualmente se hubieran presentado complicaciones.

#### 4.2.1 Software de desarrollo Altium Designer

Altium Designer reúne un conjunto de herramientas en un solo software. Es utilizado para el diseño electrónico en todas sus fases, ya que es capaz de realizar esquemáticos, simulaciones, diseño de circuitos impresos, implementación de FPGA's y desarrollo de código de circuitos embebidos.

En este software puedes generar proyectos de los siguientes tipos:

- **Proyectos PCB.** Este tipo de proyectos genera los documentos necesarios para la fabricación de un Circuito impreso. El diseño del circuito es capturado como un esquemático, construido a partir de componentes provenientes de librerías siendo colocados y alambrados juntos. El diseño es transferido al editor de PCB donde el manejo de los componentes se realiza a través de huellas y de conexiones punto a punto. Especificaciones como el tamaño de la tarjeta, reglas de ruteo de componentes, así como el ancho de la pista y el espacio entre ellas son necesarias en el editor.
- **Proyectos FPGA.** Los proyectos FPGA generan los archivos que se necesitan para poder programar un campo programable de arreglo de compuertas. El diseño es capturado por medio de esquemáticos o también por medio de código HDL. El diseño es trasladado a un lenguaje de bajo nivel de compuertas en un formato estándar conocido como EDIF.
- **Proyectos embebidos.** Estos proyectos son generados cuando se requiere producir una aplicación de software que puede ser embebida con su procesador ejecutable en un producto electrónico. En este tipo de proyectos el diseño se captura por medio de lenguaje C o ensamblador, para así convertirlo en lenguaje máquina y ser mapeado en un espacio de memoria específico produciendo un solo archivo de salida
- **Proyectos Core.** Producen una representación EDIF de un componente funcional que puede ser implementado en un FPGA. El diseño es capturado por medio de esquemáticos o código HDL. A este tipo de proyectos se le adjuntan archivos donde se especifican los dispositivos soportados. Siendo un componente dibujado para poder representar en un esquemático al código EDIF correspondiente a las referencias EDIF de cada uno.
- **Librerías integradas.** Este tipo de proyectos generan una librería integrada. Símbolos esquemáticos son dibujados en un editor de librerías y modelos de referencia definidos. Los modelos de referencia pueden incluir huellas PCB, simulaciones y modelos en tercera dimensión. Los símbolos esquemáticos y modelos requeridos son compilados juntos, en un solo archivo, referidos a esta librería integrada [26].

Para este trabajo, los tipos de proyectos que se utilizaron fueron de dos tipos: uno del tipo librería integrada, el cual fue necesario para generar todos los componentes utilizados en la tarjeta y otro del tipo PCB, del cual obtuvimos como resultado el modelo de nuestro circuito impreso, listo para ser fabricado.

En el caso del uso de librerías se creó toda una nueva del tipo esquemática y PCB, en la cual, se fueron creando uno por uno los componentes utilizados.

Altium Designer cuenta con una interfaz de usuario muy descriptiva, la cual nos simplifica el poder familiarizarse con el programa. En las figuras 4.1 y 4.2 se muestran ventanas del editor de



esquemáticos así como del editor PCB, en las cuales se observan los principales espacios y herramientas de cada uno de los ambientes.

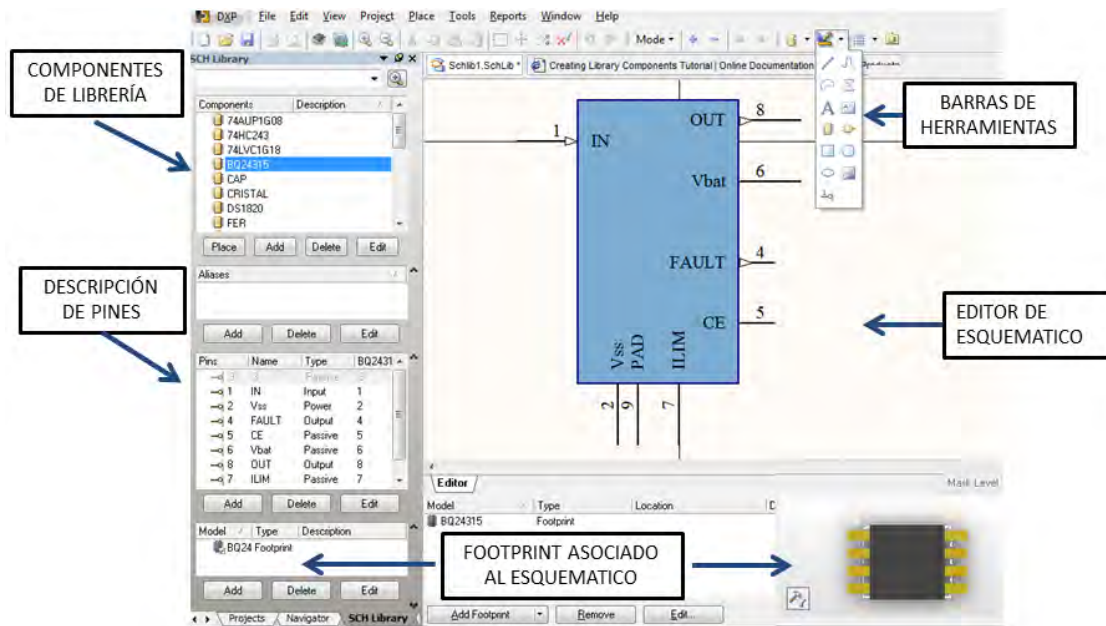


Figura 4.1 Editor de esquemático en Altium Designer

Para la parte de los esquemáticos se cuenta con un panel en el cual, se observan los componentes que se tienen en la librería, además de contener la descripción de cada uno de los pines del componente e información del footprint que tiene asociado.

Además contiene menús de herramientas, los cuales permiten generar un componente con el mayor número de detalles posibles, como lo son: descripción de pines, entradas, salidas, figuras geométricas, cuadros de textos; así como dividir el diseño de un componente en caso de que sea muy grande el patigrama de este.

En el editor PCB hay una herramienta muy valiosa para la generación de las huellas o 'footprints' llamada PCB Component Wizard, la cual facilitó la realización de huellas de componentes con diversos números de pines como lo son los del tipo BGA, DIP, QUAD, SOP, SBGA y SPGA. Esta herramienta solicita como parámetros para generar las huellas, distancias entre los pads, la cantidad de pines que tiene el componente, así como la forma y el tamaño de estos.

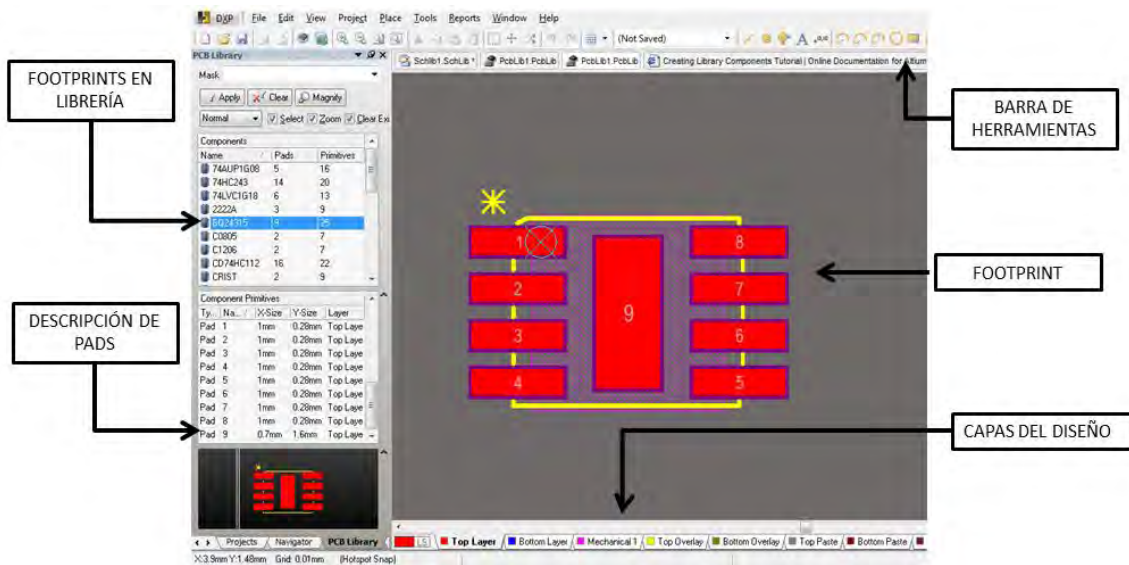


Figura 4.2 Editor de Componentes PCB.

Al igual que en el editor de esquemáticos, el editor PCB cuenta con herramientas como lo son agujeros, formas geométricas, líneas, vías y cuadros de texto, lo cual facilita el generar un nuevo diseño lo más aproximado a las especificaciones dadas por los fabricantes de los componentes utilizados. Así como un panel donde se aprecian todos los componentes de la librería y la descripción de los pines, como se muestra en la figura 4.2.

Una característica de suma ayuda en este editor es la opción de poder elegir la capa en la que se desea trabajar. Basta con seleccionar la capa y la herramienta deseada, puedes trabajar sobre una capa a la vez. Así, se puede generar desde los textos que sirven como referencia para los componentes hasta poder colocar los pads o los agujeros en el editor si son necesarios.

Con la finalidad de tener una idea del cómo quedará el diseño de la tarjeta una vez terminada, al componente se le puede asociar un archivo del tipo STEP, el cual es una representación en tercera dimensión del componente. Obteniendo una vista final de la tarjeta al terminar de rutear y colocar todos los componentes.

Para la generación de la tarjeta electrónica, como se mencionó con anterioridad, se necesitó un proyecto del tipo PCB. Siendo que este tipo de proyectos se diseñan en un esquemático y una vez teniendo este tipo de archivos, se vacía el diseño en uno PCB, listo para que los componentes sean organizados en la tarjeta.

En el diseño de esta tarjeta, al ser un diseño algo complejo, se procedió a realizar el diseño en varias hojas de esquemáticos. Respecto al uso de múltiples hojas de esquemáticos Altium Designer, tiene la facilidad de manejar varias hojas, las cuales pueden irse ligando de acuerdo a la forma en que se organice el diseño, teniendo así hojas de esquemáticos padres e hijos. Herramientas como puertos, buses, símbolos de hojas y entradas a hojas, permiten ir representando el diseño contenido en hojas de esquemáticos y así poder enlazarlas a una sola hoja padre. Teniendo la posibilidad de tener los niveles que sean necesarios para realizar el diseño.

En nuestro caso como se puede ver en la figura 4.3, se utilizaron solo una hoja padre y seis hojas hija. En la hoja padre, se colocaron los componentes con mayor conectividad, como lo son el

microcontrolador y el conector de costilla de 50 posiciones, del cual se derivan diversas señales entre subsistemas; los símbolos de hojas que representan a las hojas hija conteniendo en ellas una parte del subsistema, circuito de USB, los radios en VHF, UHF y las protecciones contra efecto Latch Up.

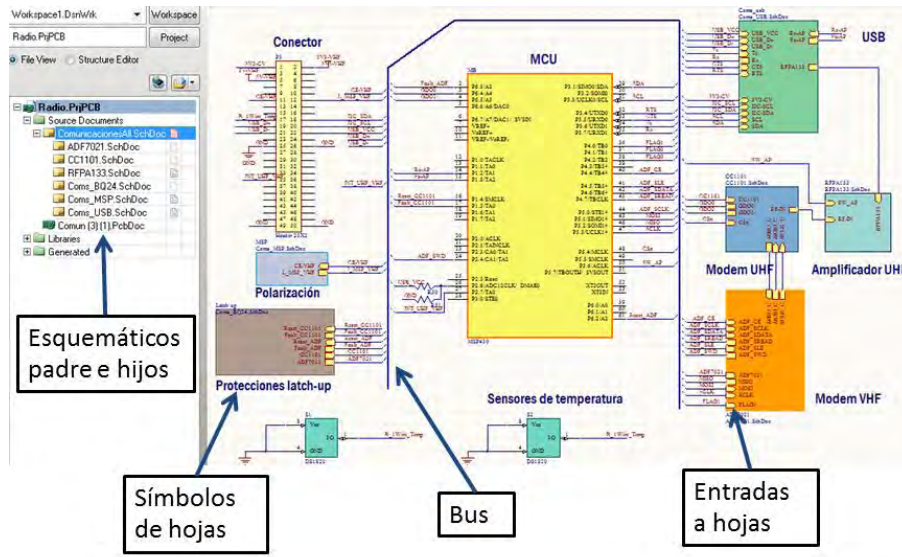


Figura 4.3 Distribución de hojas de esquemáticos

En cuanto se tiene el diseño en forma de esquemático se puede generar un reporte de errores, con la finalidad de checar violaciones y poder checar las más severas entre conexiones eléctricas de los componentes. Ejemplos de estos errores son conexiones entre pines, puertos y entradas de hojas. Este reporte se genera con la finalidad de pasar a la elaboración de la PCB del circuito [27].

#### 4.2.2 Consideraciones en la elaboración

Al pasar del esquemático al PCB se tomaron en cuenta aspectos que podrían traer problemas. El ruteo y ancho de las pistas, así como espacios entre ellas, fueron algunos aspectos que se tomaron en cuenta, ya que, al ser una buena cantidad de componentes, el espacio de la tarjeta para poder interconectar a los componentes se ve reducido drásticamente.

En cuanto a la posición de componentes dentro del espacio propuesto, se debió seguir cierto orden, pues al no seguirlo, se observó que el ruteo de pistas se volvía más laborioso y complicado, trayendo como consecuencia el volver a comenzar con dicho proceso, retrasando el tiempo de entrega del diseño.

##### 4.2.2.1 Ubicación de componentes

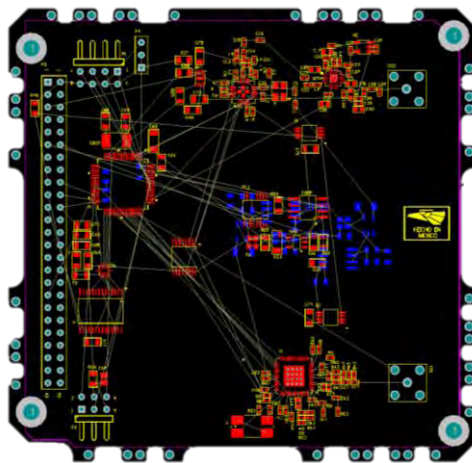
En lo que se refiere a la ubicación de los componentes se optó por seguir el mismo orden con el que se trabajó en los esquemáticos. Los componentes se fueron colocando por secciones, de tal manera, que se fueron generando partes de la PCB donde se colocaban los componentes del radio de transmisión, en otra los componentes del radio de recepción, en una parte para el circuito de interfaz USB a serie y en una última para las protecciones contra efecto Latch Up.

En el caso del microcontrolador al ser el de mayor conexión, se optó por colocarlo de tal forma que a la hora de rutearlo con las demás secciones tuviera la menor dificultad, siendo una buena posición la parte del centro y teniendo a las demás secciones a su alrededor.

Para el conector de 50 posiciones su posición fue definida debido al mutuo acuerdo entre subsistemas, ya que al ser un bus de entradas y salidas entre subsistemas debía permanecer en un mismo lugar en todas las tarjetas de los subsistemas.

En cuanto a los componentes de cada uno de los transceptores, el tamaño de los componentes recomendados por los fabricantes fueron del empaquetado 0402 milésimas de pulgada y nos brinda una plantilla con la posición de cada uno. Este tamaño se justifica debido al tamaño del chip de radiofrecuencia, al ser de un tamaño tan pequeño, al momento de acomodar componentes de mayor tamaño, no podrían ser colocados a la distancia señalada por el fabricante.

En la figura 4.4 Se puede observar la distribución final de los componentes basándose en los puntos anteriores.



*Figura 4.4 Posicionamiento de los componentes.*

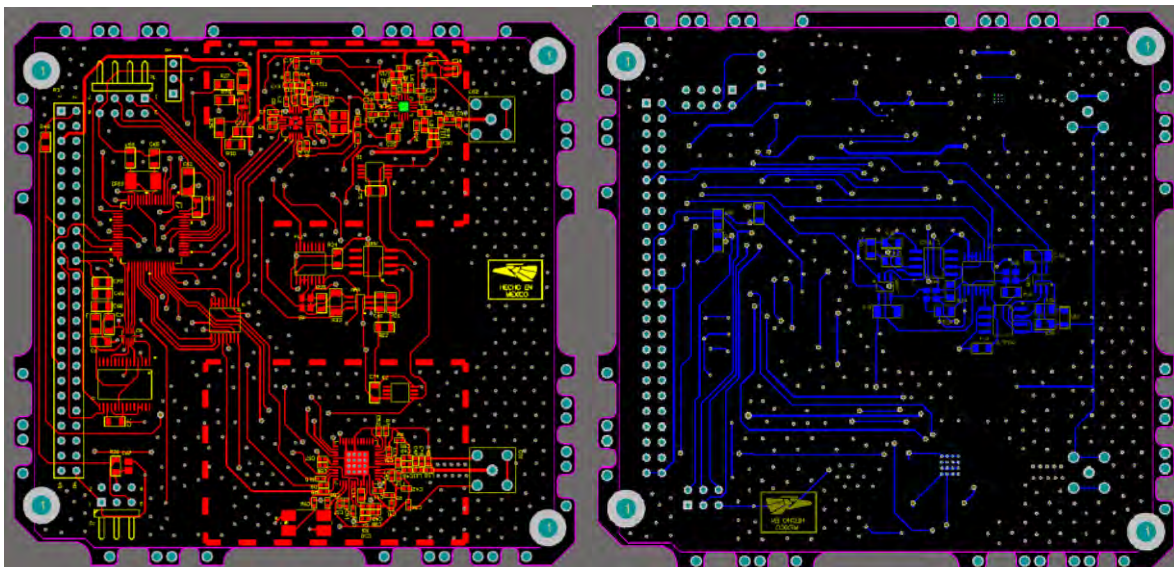
#### **4.2.2.2 Ruteo de pistas**

Para el ruteo de las pistas se buscó el tener las rutas más rápidas y directas entre componentes. Esto con el fin de no generar posibles interferencias entre líneas de distintos protocolos. Para que esto fuera posible el punto anterior fue de gran ayuda, ya que al tener los componentes de cada sección separados, su ruteo se volvió menos laborioso.

Se tomaron con mayor prioridad pistas en las que se manejaban los protocolos I2C, SPI y SERIE, las cuales se colocaron de tal forma en que no se cruzaran entre ellas y llegaran de principio a fin lo más directo posible.

Para el ruteo de los componentes de cada uno de los transceptores se siguieron las recomendaciones hechas por los fabricantes, el hecho de no seguirlos y trazar pistas más gruesas o más largas generarían inductancias y capacitancias parasitas lo que nos repercutiría en el funcionamiento del circuito.

En cuanto el grosor de las pistas, se manejaron tres tamaños, el primero de ellos fue de 0.01" y fue utilizado para la mayoría de pistas. Protocolos de comunicación, señales de interrupción, alimentación de 3.3V y señales de sensores fueron ejemplo de su uso. Para la alimentación de 5V, se ocupó una pista de mayor grosor (0.02"), ya que se espera un valor de corriente mayor. Por último, se utilizó una pista de 0.025" para las señales de radiofrecuencia, las cuales salen de uno de los circuitos de RF y del amplificador de potencia hacia el conector que va hacia la antena.



*Figura 4.5 Vista superior e inferior de la tarjeta ruteada.*

#### **4.2.2.3 Planos de tierra**

La tarjeta lleva planos de tierra por ambas caras, esto con el propósito de mitigar posibles ruidos entre pistas de distintos protocolos. Los planos son sólidos y cubren la mayor parte de la superficie de la tarjeta. Cuenta con orificios en los sitios donde no hay ni componentes ni pistas, eliminando así, valores de capacitancias parasitas en el cobre del plano además de absorber ruidos que serían perjudiciales en nuestro diseño [28].

En cuanto al ruido, algo que se observó en el diseño de chips de radiofrecuencia fue la colocación de una tipo armadura que cubriera el circuito de RF dentro de la misma tarjeta. Ésta cubierta tiene

la finalidad de proteger al circuito del radio de ruido generado en el exterior de la tarjeta, así como el de no causar interferencia con los demás subsistemas.

Una vez realizado el diseño de la tarjeta, se generaron los archivos correspondientes a cada capa del circuito y con estos se mandó a fabricar. Datos como tamaños de orificios, revisar bien la posición de componentes en cada capa, continuidad de pistas, vías y las conexiones de los pads o pines de componentes, son de suma importancia, ya que si alguno de estos aspectos se descuida nuestro diseño podría no funcionar, teniendo que mandar a fabricar otro, implicando más tiempo y gastos de diseño.

### 4.3 Ensamble de componentes

Al ya contar con el circuito impreso se revisó la tarjeta con el fin de verificar la fabricación de la misma, ya que puede haber ocurrido que este mal fabricada, pistas abiertas y cortos circuitos entre pistas suelen ser defectos que podemos encontrar.



*Figura 4.6 Tarjeta del subsistema de comunicaciones.*

Una vez realizada la revisión de la tarjeta el siguiente paso fue ensamblar los componentes. Para el ensamble se tomó la decisión de comenzar con los arreglos que se hicieron para proteger a los circuitos contra efecto Latch Up, pues se realizarían pruebas de este circuito antes de colocar componentes como lo son chips de radiofrecuencia y el microcontrolador. Estas pruebas se detallarán en el siguiente capítulo.

Una vez colocados los arreglos contra efecto Latch Up se prosiguió a ensamblar los circuitos de radiofrecuencia, el ADF7021, el CC1101 y el RFPA0133 junto con sus respectivos arreglos de capacitores, inductores y resistores, los cuales fueron los componentes más pequeños que se tuvieron que ensamblar siendo complicado, pues al ser un empaquetado del tipo 0402 milésimas de pulgada las dimensiones eran de 0.5mm x 1mm.

Teniendo soldados los circuitos de RF y los arreglos contra efecto Latch Up, se continuó con los sensores de temperatura y con el circuito de interfaz USB a SERIE, los cuales son de mayores dimensiones y no presentaron ningún inconveniente a la hora de soldarlos.

A la tarjeta se le colocaron conectores provisionales en lugar de colocarle el conector de 50 posiciones, esto con la finalidad de realizar pruebas antes de colocarle el conector final.

Por último se colocó el microcontrolador junto con sus componentes adicionales, el cristal y sus capacitores, la razón para colocar al microcontrolador al final fue preventiva, ya que al poner en contacto en varias ocasiones a la tarjeta con las herramientas de ensamble, ponía en riesgo al componente, que es susceptible a estos cambios de temperatura.

#### **4.3.1 Recomendaciones en cuanto al ensamble**

Durante el proceso de ensamble fue necesario tomar en cuenta algunas consideraciones para poder soldar los componentes a la tarjeta. Diferencias entre componentes, empaquetados diversos y aspectos de prevención son algunas de las razones por las que se debieron tomar estas consideraciones. En la siguiente lista se mencionan los aspectos que se consideraron:

- En general se ocupó a lo largo de todo el ensamble el utilizar una pulsera antiestática, esto con el fin de proteger a los circuitos más susceptibles a descargas estáticas.
- A los componentes con una cantidad considerable de terminales, se les inmovilizaba dentro de su huella para que estos no se movieran y quedaran alineados dentro de ella, de no hacerlo se corría el riesgo de moverse y dejar unas terminales fuera del área para soldar, trayendo como consecuencia una mala operación o inclusive un circuito inservible.
- Debido a las dimensiones tan pequeñas que tienen algunos componentes como los chips de radiofrecuencia (entre 4mm y 6mm), se optó por utilizar un microscopio para poder soldarlos, ya que realizándolo a simple vista era en ocasiones imposible verificar si se habían logrado soldar los componentes. Así como el uso de puntas de cautín más delgadas, con la finalidad de alcanzar y soldar los pads más pequeños y juntos.
- Componentes como el ADF7021 o el CC1101 utilizan el área debajo del chip para referenciarlo a tierra. Para esta parte metalizada debajo del encapsulado se utilizó soldadura en pasta, la cual facilitaba la tarea de soldarlo que con soldadura convencional. Una pistola de aire caliente fue necesaria para la aplicación de dicha pasta.
- Es importante el utilizar cantidades adecuadas de pasta o líquidos para soldar tal es el caso de flux, así como el mantener limpia las puntas de cautín utilizadas, ya que al no hacerlo se puede presentar el caso que al momento de soldar se presente un sobrecalentamiento que perjudique a la tarjeta.

En la figura 4.7 se muestra la tarjeta con todos sus componentes ensamblados.

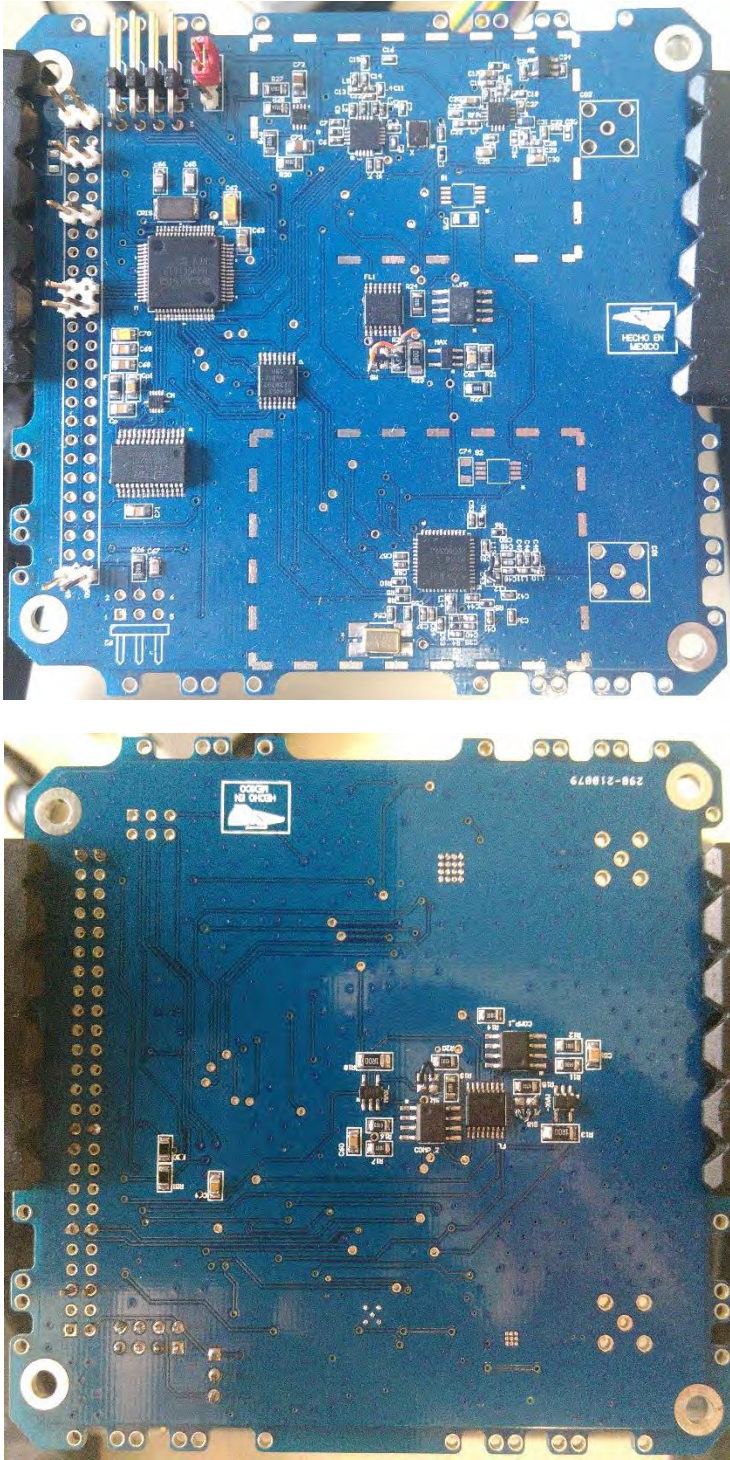


Figura 4.7 Vista superior e inferior de la tarjeta ensamblada



# Capítulo 5 Software de operación y pruebas de validación del Subsistema de Comunicaciones

---

## 5.1 Introducción

Una vez terminado el subsistema en lo que a hardware se refiere, lo siguiente a realizar fue desarrollar el software de operación del subsistema. Para poder desarrollar dicho software fue necesario efectuar pruebas por cada uno de los periféricos que se utilizaron del MSP430F1611, esto con el propósito de asegurarse de que los componentes soldados en la tarjeta no presentaran ningún inconveniente y que pudieran responder sin ninguna dificultad en cuanto a la programación que se les diera.

Para el caso de los circuitos de protección se tuvieron que validar antes de ser colocados en el diseño de la tarjeta, pues se debía asegurar su funcionamiento para que a la hora de probarlos en la tarjeta, solo fuera cuestión de verificarlos y ya no de tener la expectativa de su funcionamiento.

En lo que se refiere a software de operación hablamos de software para el microcontrolador, pues de este depende el funcionamiento del subsistema en su totalidad. Podríamos hablar de programar los chips de radiofrecuencia, pero lo que se realiza en ellos es configurar registros y comunicarse con ellos mediante el uso de protocolos, solo respondiendo instrucciones ordenadas por el microcontrolador.

A lo largo de este capítulo se describirán las pruebas de validación del subsistema, mostrando diseños implementados en protoboard o circuitos impresos realizados para ese fin. Se referirá al ambiente en el que se trabajó para realizar el software de operación y por último se explicará el software elaborado para poder interactuar con el subsistema de comunicaciones.

## 5.2 Software de desarrollo Code Composer Studio

Para la programación del microcontrolador MSP430F1611 se utilizó una plataforma elaborada por el mismo fabricante (*Texas Instrument*) que lleva por nombre Code Composer Studio.

Code Composer Studio es un entorno de desarrollo integrado (IDE por sus siglas en inglés), que soporta todos los microcontroladores de la familia MSP. Cuenta con un conjunto de herramientas integradas de software las cuales se utilizan para desarrollar y depurar aplicaciones embebidas. Incluye un compilador optimizado en C/C++, un editor de código fuente, entorno de construcción de proyectos, un depurador y muchas otras características.

Este intuitivo IDE proporciona una única interfaz al usuario, la cual lo va guiando a través de todas las etapas de desarrollo de su aplicación. Herramientas familiares y útiles además de sus interfaces, permiten al usuario comenzar más rápido. Code Composer Studio combina las ventajas de utilizar una plataforma basada en Eclipse con capacidades de depuración avanzada de *Texas Instrument* dando como resultado un entorno de desarrollo rico en características de peso para desarrolladores de sistemas embebidos [29].

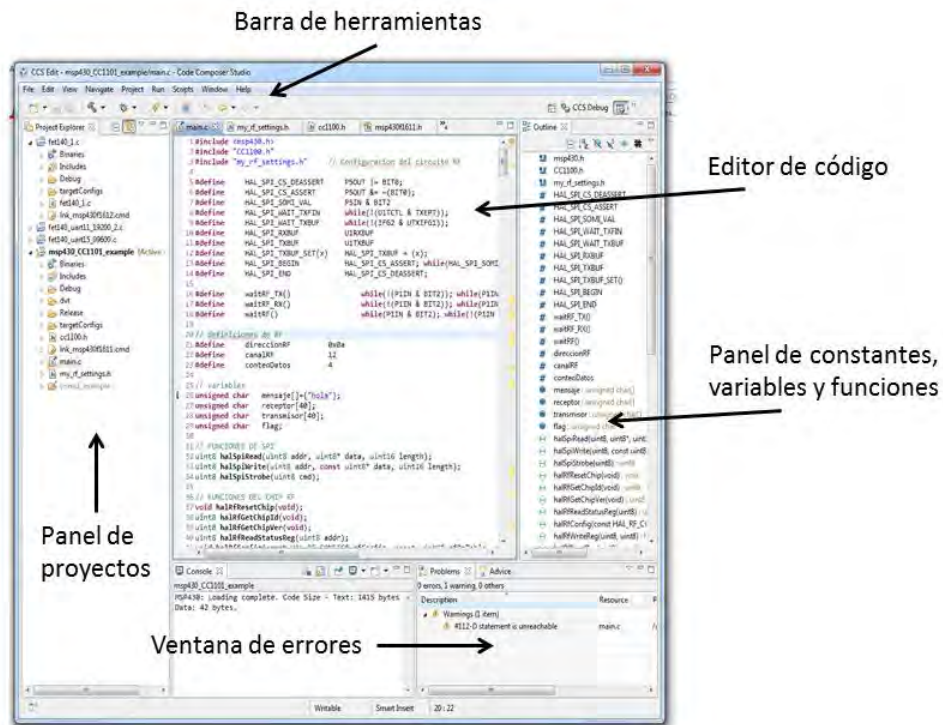


Figura 5.1 Editor de código Code Composer Studio

### 5.3 Cargado de programa al MSP430F1611

Para borrar, programar y verificar el módulo de memoria del MSP se utiliza la interfaz de comunicación JTAG.

Esta interfaz consta de 4 líneas para enviar y recibir datos. En los dispositivos más pequeños se tiene una quinta línea (TEST), la cual se utiliza para habilitar la interfaz, pues los pines del dispositivo se comparten con alguno de sus periféricos. En este dispositivo se tienen pines dedicados a esta interfaz. La tabla siguiente muestra los pines y su función [30].

Pin	Dirección	Descripción
54.TDO	Salida	Salida de datos JTAG
55.TDI	Entrada	Entrada de datos JTAG y entrada TCLK
56.TMS	Entrada	Señal que controla la máquina de estados JTAG
57.TCK	Entrada	Entrada de reloj JTAG
**TEST	Entrada	Pin que habilita interfaz JTAG

Tabla 5.1 Pines de interfaz JTAG

La señal TCLK es una señal de reloj proveniente de una fuente externa a la tarjeta, la cual se utiliza internamente como el reloj del sistema. El diagrama de conexión se muestra en la figura 5.2 [31].

Para el cargado de software al MSP430 se utilizó el programador MSP430-JTAG-TINY-V2, producido por una empresa llamada Olimex. Dicho programador se encarga de realizar la comunicación por medio de esta interfaz JTAG. Este programador solo requiere descargar de la página del fabricante los drivers necesarios para que la computadora reconozca el dispositivo y así poder tener una buena comunicación con el microcontrolador.

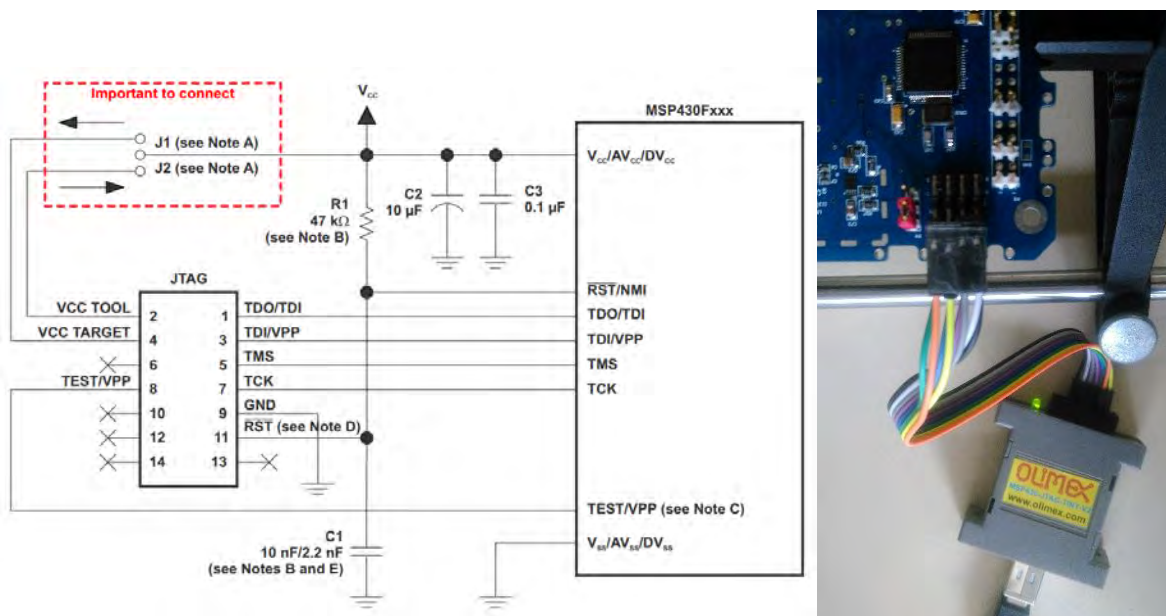



Figura 5.2. Conexión JTAG para el cargado de programa al MSP430F1611

Una vez conectado el microcontrolador a la PC y poder programarlo con el software Code Composer Studio, se realiza lo siguiente:

Lo primero que se tiene que verificar es si el programador está siendo detectado por la computadora. Esto se realiza en administración de dispositivos en el panel de control, debe aparecer el puerto virtual que se genera al instalar los drivers del programador.

Una vez abierto el software se crea un nuevo proyecto, indicando información del dispositivo así como el tipo de proyecto que se desea elaborar.

Una vez terminado el proyecto se compila y se carga en el microcontrolador por medio de la herramienta de depuración  direccionándonos a la ventana donde se puede ver como se ejecuta el programa, mostrándose en la figura 5.3.

En esta ventana abierta se observan paneles en donde se puede ver la ejecución del código del programa, otro donde muestra cambios en variables, registros y expresiones. Además se puede pausar, detener o correr el programa a pasos verificando así el funcionamiento de nuestro código.

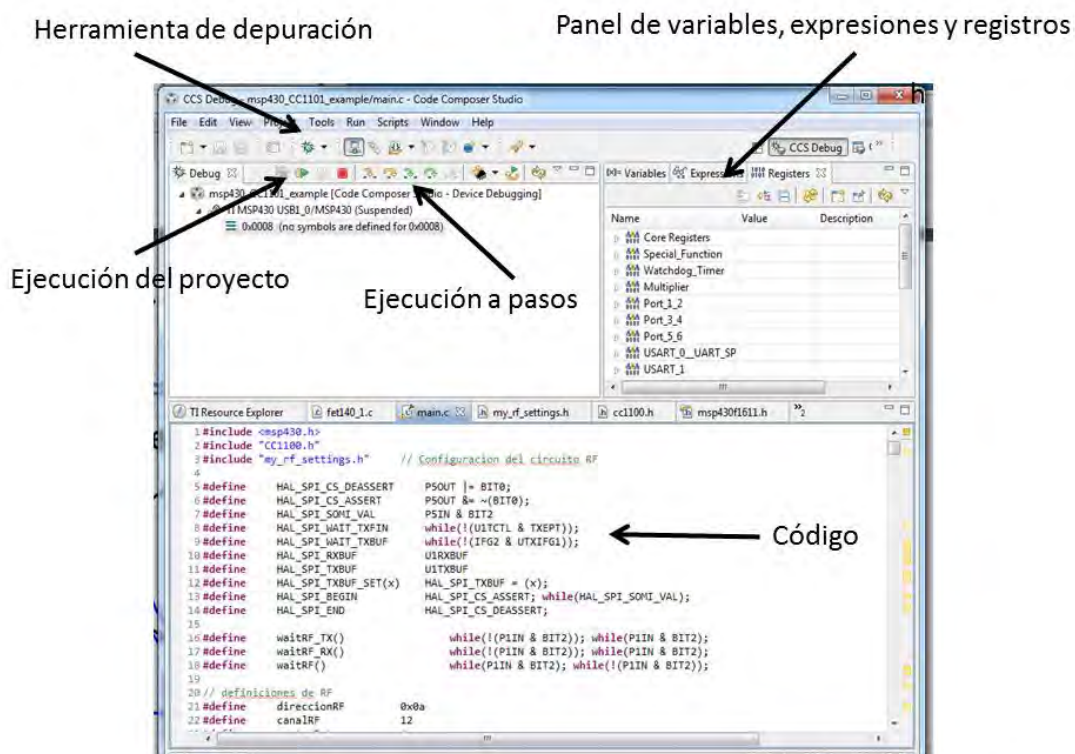


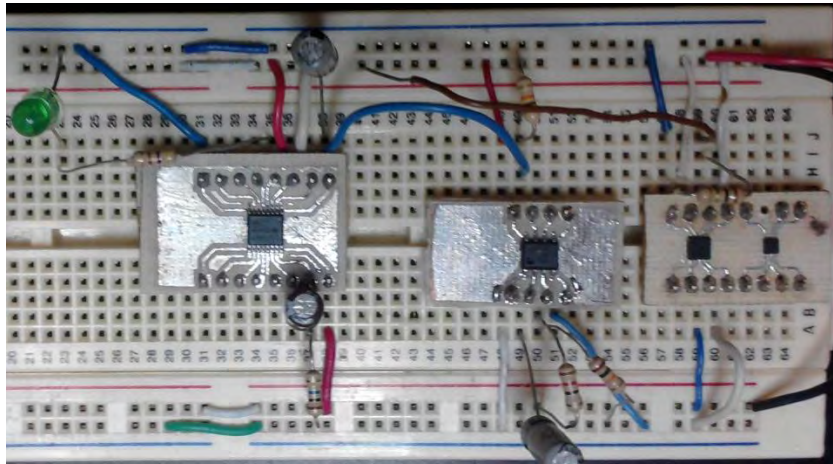
Figura 5.3 Ventana de depuración de Code Composer Studio

## 5.4 Pruebas realizadas al subsistema

El subsistema de comunicaciones fue sometido a una serie de pruebas, tanto en software como en hardware, con la finalidad de verificar el buen funcionamiento de los dispositivos que componen la tarjeta. Las pruebas que se realizaron en hardware básicamente fueron en los circuitos de protección, asegurándonos que estas protecciones funcionaran de acuerdo a lo ya establecido. En lo que a pruebas de software se refiere, se realizaron códigos de programas muy específicos en cuanto a los protocolos a probar entre el microcontrolador y el dispositivo que se deseaba validar, como lo son el FT232R, el CC1101, el ADF7021 y el bus de I2C que comunica al MCU con la Computadora de Vuelo.

### 5.4.1 Pruebas de protección contra efecto latch up

La realización de estas pruebas se llevó a cabo antes y después del diseño de la tarjeta. Fue necesario realizar pruebas antes, ya que era necesario corroborar que el diseño funcionara y pudiera integrarse al diseño final. Para estas primeras pruebas se diseñaron tarjetas impresas para cada componente, pues no se tenía asegurado un arreglo en particular. Una vez ensamblados se montaron sobre una protoboard con la finalidad de interconectarlos entre sí para formar el arreglo propuesto y verificar su funcionalidad.



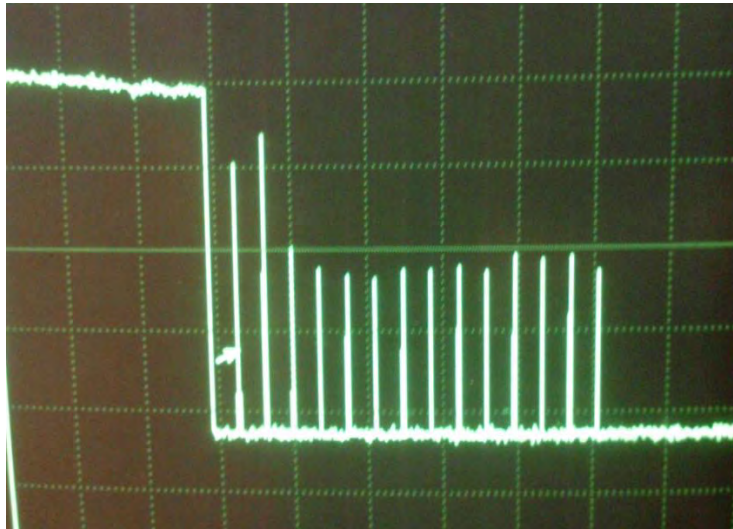
*Figura 5.4 Pruebas en protoboard*

Una vez realizadas las pruebas y obteniendo resultados satisfactorios, se procedió a hacer el diseño de las protecciones en la tarjeta.

#### 5.4.1.1 Circuito protector mayor a 300mA

Las pruebas que se realizaron al BQ24314 fueron sencillas, básicamente fueron mediciones de corrientes y voltajes a la entrada del circuito a proteger. Para este caso del protector mayor a 300mA se analizó mediante el osciloscopio la señal que alimentaba la carga (el chip de RF). Esta carga era simulada por medio de una resistencia variable, con la cual se hacía variar la corriente demandada.

El resultado se puede apreciar en la Figura 5.5, mostrando el comportamiento de la protección. El circuito BQ24314, al notar el incremento en la corriente consumida y rebasar el máximo al que está programado realizaba 15 intentos por apagar y encender el voltaje de nuestra carga, siendo que al quinceavo intento no se ha disminuido la corriente, el circuito de protección corta el suministro de alimentación, apagándolo en ese momento y envía una señal de aviso al microcontrolador del subsistema de potencia. El circuito permanece apagado hasta que por medio de software, el subsistema de potencia resetea al circuito protector volviendo a suministrar el voltaje a la carga.



*Figura 5.5. Respuesta del circuito BQ24314 a corrientes altas.*

#### **5.4.1.2 Circuito protector menor a 300mA**

Para el caso del arreglo menor a 300mA las pruebas que se realizaron fueron muy similares al circuito anterior. Al tratarse de un arreglo, se identificaron a las terminales que interesaba observar su respuesta, en este caso se observaron:

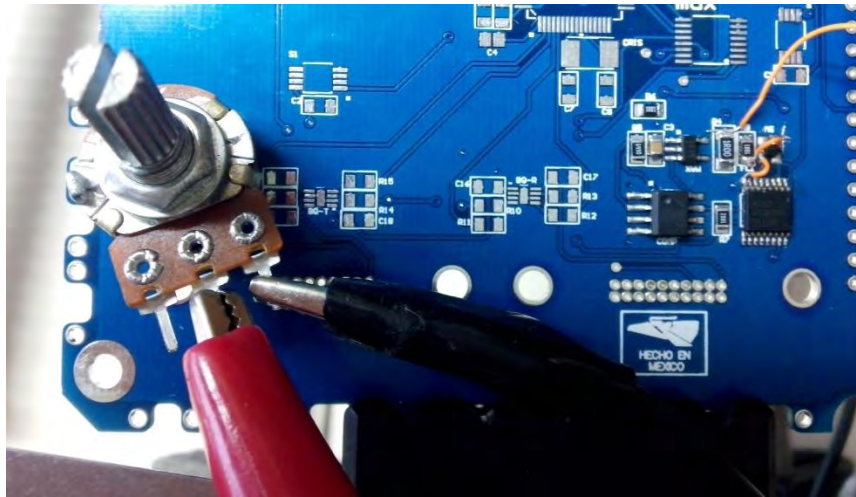
Vout\_MSP: voltaje a la salida del arreglo protector en general. Esta terminal es la salida del switch MIC9404. Se espera que el voltaje en esta terminal sea 0V si se presenta una corriente elevada, en caso contrario se esperan los 3.3V con los que se alimentan las cargas.

L\_MSP\_VHF: señal que desactiva el flujo de alimentación hacia la carga. Proveniente de la salida del Flip Flop JK, cambia de 5V a 0V cuando ocurre una sobrecorriente. Esta terminal alimenta el pin de habilitación del Switch MIC9404, el cual al tener ese cambio en su voltaje deshabilita el swith desconectando a la carga de su fuente, además de enviarse como aviso al subsistema de potencia, con la finalidad de informarle que se ha desconectado la carga.

CE\_CHIP: esta terminal proviene de la señal de Reset del Flip Flop JK. Al cambiar de estado el Flip Flop JK al presentarse una corriente elevada, deja desconectada la carga de su alimentación. Se monitorea esta terminal ya que debe ocurrir un cambio de nivel lógico (de 5V a 0V) para que se

pueda reconectar la alimentación a la carga. Esta es la única manera de hacerlo, siendo el encargado de ello el subsistema de potencia.

La prueba que se realizó ya en la tarjeta impresa tuvo los resultados esperados, se tomó como carga a una resistencia variable y se midieron las terminales antes mencionadas, presentándose el comportamiento antes mencionado. La protección al desaparecer el aumento en la corriente era activado nuevamente gracias al cambio de nivel lógico en la terminal CE\_CHIP.



*Figura 5.6 Prueba a la protección contra efecto Latch Up menor a 200mA.*

#### **5.4.2 Prueba de comunicación mediante el bus USB**

Para poder realizar las pruebas en esta parte de la tarjeta se tuvo que desarrollar un poco de software, con la finalidad de comprobar que en realidad se tuviera una comunicación entre la computadora y el microcontrolador.

Como se mencionó en el capítulo 3 el circuito FT232R es el que realiza la interfaz entre protocolo USB a comunicación serial. Para que el dispositivo pueda interactuar con la computadora como un puerto virtual del tipo serial es necesario cargarle a la computadora unos controladores. Estos controladores pueden ser descargados de la página del fabricante sin ningún problema resolviendo esta parte, quedando así, pendiente la comunicación serial proveniente del circuito FT232R hacia el MCU.

Para poder recibir estos datos enviados vía serial el MCU cuenta con un módulo llamado USART, el cual, puede manejar dos tipos de comunicaciones seriales, las de tipo síncrono así como las de tipo asíncrono. Siendo del tipo asíncrona el tipo de comunicación que arroja el FT232R.

Para comunicaciones asíncronas el módulo USART debe ser configurado para entrar en un modo llamado UART (Universal Asynchronous Receiver/Transmit), el cual conecta al MSP con cualquier sistema externo mediante dos pines, el UTXD Y URXD.

Las características en este modo de comunicación son las siguientes:

- Son 7 u 8 bits de dato con paridad par, impar o sin ella.
- Registros de corrimiento independientes de transmisión y recepción
- Registros de almacenamiento de transmisión y recepción separados.
- LSB primer dato en transmisión y recepción
- Puede despertar de Modos de bajo consumo.
- La tasa de baudaje es programable.
- Banderas de estado para la detección de errores.
- Habilidad de interrupciones independientes para recibir y transmitir

En este modo, el MCU transmite y recibe caracteres a cierta tasa asíncrona a otro dispositivo. El tiempo entre cada carácter es basado en la tasa de baudaje seleccionado del módulo. La transmisión y recepción utilizan esa misma frecuencia.

### Para mayor Información sobre comunicación serial diríjase al anexo A

Para poder utilizar el módulo UART en este tipo de comunicación hay que configurarlo, para lo cual se siguen esta serie de pasos:

- En primera instancia hay que activar este modo lo cual se hace al limpiar el bit SYNC en el MSP
- Una vez hecho esto hay que reiniciar el módulo USART, colocando el bit SWRST.
- Seguido de inicializar todos los registros teniendo SWRST=1.
- Habilitar el modulo vía MEx SFRs (URXEx y/o UTXIEx).
- Se limpia SWRST vía software.
- Se habilitan interrupciones (opcional) [32].

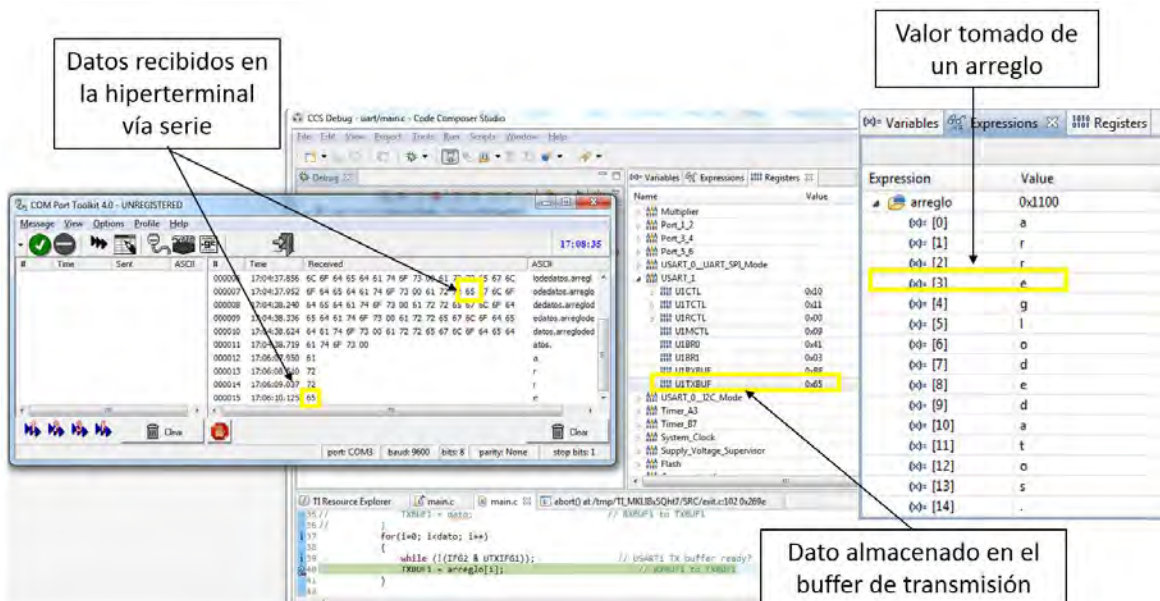


Figura 5.7 Datos transmitidos por el MCU vía serie utilizando el FT232R



Lo que se programó en un principio para esta comunicación fue un código de configuración de registros del módulo USART, con los cuales el módulo fuera capaz de comunicar los datos recibidos del FT232R. Registros ligados a la activación del periférico, la velocidad de transmisión de datos, el número de bits y la paridad fueron los que se vieron modificados. Además de activar la interrupción al recibir datos por medio de este puerto.

Una vez realizado esto se procedió a guardar los datos recibidos, para que una vez almacenados sean procesados por la computadora de vuelo. El subsistema de comunicaciones se encarga solo de recibir la información y enviarla a la computadora de vuelo, la información no se ve alterada al pasar por el subsistema.

En las figuras 5.7 y 5.8 se muestran ejemplos de la transmisión y recepción de datos por medio del FT232R, mostrándose énfasis en el buffer de transmisión y recepción del módulo UART, así como en la variable transmitida.

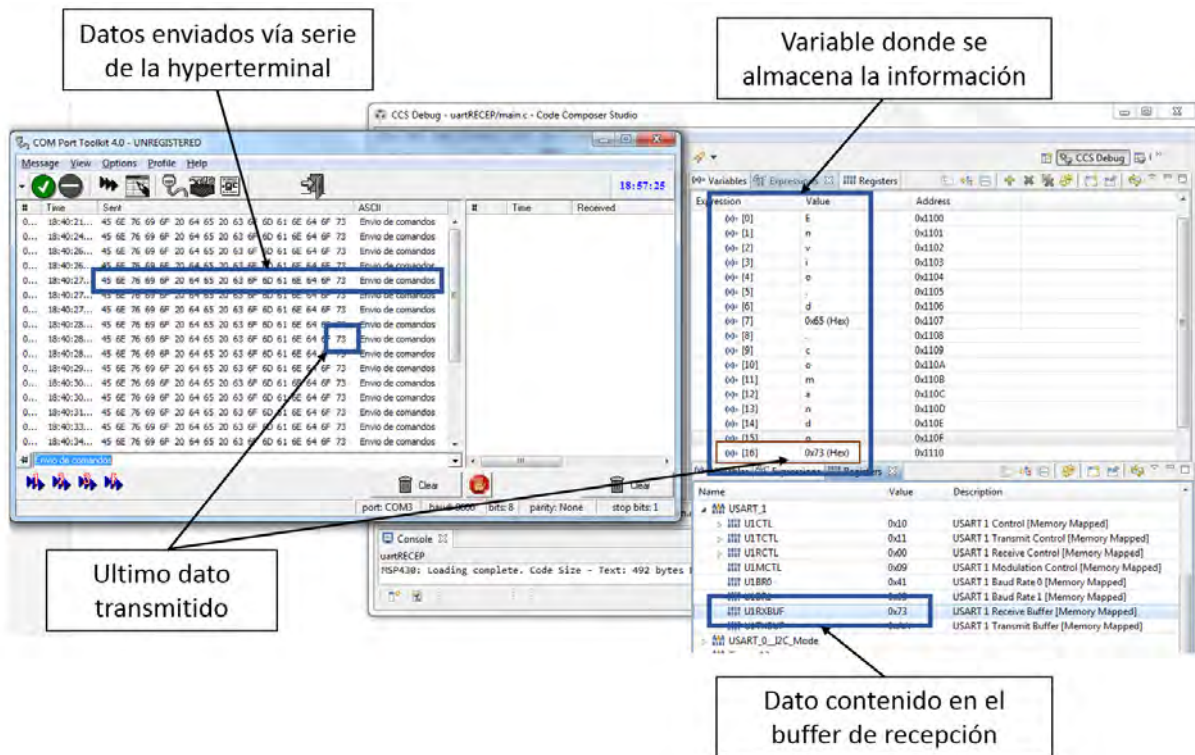


Figura 5.8 Datos recibidos por el MCU vía serie utilizando el FT232R

### 5.4.2.1 Software Com Port Toolkit

La interacción entre la computadora y el FT232R se realizó a través de una herramienta llamada *COM Port Toolkit*, la cual permite reconocer al dispositivo FT232R como un puerto serial. Mediante esta herramienta se pueden enviar y recibir datos de la computadora al MCU.

Esta herramienta tiene una interfaz amigable. Como se puede observar en la figura 5.7 cuenta con menús en los cuales se puede manipular los datos a recibir, así como el poder variar los parámetros de la comunicación serial tales como el baudaje, número de bits, el tipo de paridad, los bits de parada y la selección del puerto serie a elegir. El envío y recepción de datos se da de manera sencilla y se observan los datos en la interface.

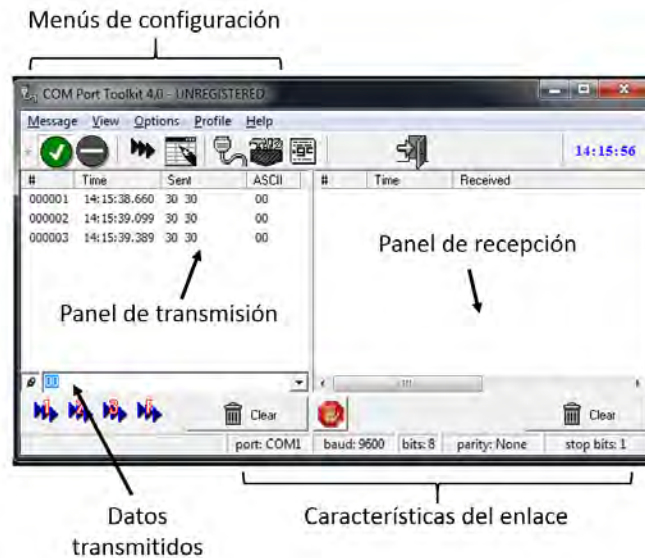


Figura 5.9 Ventana principal de Com Port Toolkit.

Para poder entablar una comunicación entre el programa *Com Port Toolkit* y el MCU se deben colocar los mismos parámetros en los dos. En este caso, se utilizó un baudaje de 9600 baud, datos de 8 bits, sin paridad y un bit de parada. Esta herramienta fue de gran ayuda ya que se pudo comprobar el buen funcionamiento del FT232R mediante el flujo de datos que había entre el microcontrolador y la computadora.

### 5.4.3 Prueba de comunicación por medio del bus I2C

Como ya se mencionó anteriormente, el módulo USART del MCU es el encargado de comunicarlo mediante protocolos del tipo serie, síncronos o asíncronos. En el punto anterior se detalló la comunicación del tipo asíncrono mediante dos líneas UTXD Y URXD. Ahora bien, toca el turno de un protocolo del tipo síncrono, el cual lleva por nombre I2C (Inter-Integrated Circuit), cuyo objetivo fue el poder comunicarse con el subsistema de Computadora de Vuelo.

Para esto, el módulo provee una interfaz entre el MCU y dispositivos compatibles con el protocolo I2C. Los componentes externos conectados al bus, transmiten o reciben datos desde la USART a través de la interfaz de dos hilos de este protocolo.

Este módulo cuenta con las siguientes características:

- Es compatible con las especificaciones v2.1 I2C de Phillips Semiconductor
  - Formato de transferencia byte/Word
  - Modos de direccionamiento de dispositivos de 7 y 10 bits
  - Llamada general
  - Condiciones de START/RESTART/STOP
  - Modo Multi-maestro transmisor/ esclavo receptor
  - Modo Multi-maestro receptor / esclavo transmisor
  - Modos combinados maestro transmisor/receptor y receptor/transmisor
  - Soporte en Modo estándar de hasta 100kbps y modo rápido de 400kbps
- Registros de almacenamiento FIFO en cuanto a lectura y escritura.
- Generación de reloj programable.
- datos con un ancho de 16 bits para maximizar el rendimiento del bus.
- Contador automático de bytes
- Diseñado para baja potencia
- Esclavo receptor capaz de salir del modo bajo consumo con condición de START.
- Gran capacidad de interrupciones.

El módulo I2C soporta cualquier esclavo o maestro compatible. Cada dispositivo es reconocido por una única dirección y puede operar como receptor o transmisor. Un maestro inicializa una transferencia de datos y genera una señal de reloj SCL, cualquier dispositivo direccionado por un maestro se considera como esclavo.

Los datos son comunicados utilizando la línea de datos serial (SDA) y la línea del reloj serie (SCL). Ambas líneas son bidireccionales y deben de tener una resistencia de pull-up.

Para inicializar el módulo I2C, al ser parte del periférico USART, se deben configurar bits individuales dentro de sus registros, los cuales lo definen como el modo I2C, dentro de la unidad USART, diferentes del SPI y de la comunicación UART.

Para configurar el módulo el proceso es el siguiente:

- Hay que activar dos bits en el registro de control U1CTL: SYNC e I2C.
- Deshabilita el módulo I2C
- Configurar el módulo I2C con I2CEN =0
- Activar el I2CEN vía software.

El protocolo consta de un dispositivo maestro y uno o varios esclavos. En este caso el dispositivo maestro es el subsistema de computadora de vuelo, mientras que el subsistema de comunicaciones fungirá como esclavo. El subsistema de comunicaciones será el encargado de avisarle a la computadora de vuelo que ha recibido datos por medio de un pin que funciona como bandera externa al protocolo de I2C.

Al ser un dispositivo esclavo la operación de recepción y transmisión son controladas automáticamente por el módulo I2C.

En el modo esclavo receptor los datos son recibidos de forma serial en la línea SDA para sincronizarse con el maestro se usa una señal de reloj, el dispositivo esclavo no puede generar dicha

señal pero puede mantener baja la línea de SCL si la intervención del CPU es requerida después de haber recibido un byte. En este modo cada byte recibido será reconocido.

Mientras que en el modo esclavo transmisor es iniciado cuando la dirección del esclavo a elegir es transmitida por el maestro y un bit de lectura/escritura ha sido transmitido, indicando al esclavo que debe responder enviando datos al maestro. Siendo así el esclavo hace los corrimientos necesarios para transmitir los datos por la línea SDA sincronizados por los pulsos del reloj generado por el maestro.

Algo de gran utilidad es el vector de interrupciones que tiene este módulo, pues es capaz de generar 8 banderas de interrupción. Cada bandera de interrupción tiene su propio bit de habilitación [32]. Las 8 banderas se muestran en la tabla siguiente:

Bandera de interrupción	Condición de interrupción
<b>ALIFG</b>	Arbitraje perdido. Ocurre cuando 2 o más transmisores empiezan una transmisión simultáneamente, o cuando los intentos por software para inicializar una transferencia I2C mientras I2CBB=1.
<b>NACKIFG</b>	Interrupción por No-Acknowledge. Esta bandera se enciende cuando una condición de Acknowledge es esperada pero no es recibida en el modo maestro. Solo aplica en modo maestro.
<b>OAIFG</b>	Interrupción por propia dirección. Esta bandera es encendida cuando otro maestro tiene direccionado el módulo I2C. Solo aplica en modo esclavo.
<b>ARDYIFG</b>	Interrupción por acceso inmediato a registros. Esta bandera se activa en alguna de las siguientes condiciones: <b>Maestro transmisor</b> , I2CRM=0: todos los datos enviados. <b>Maestro transmisor</b> , I2CRM=1: todos los datos enviados y esta encendido I2CSTP. <b>Maestro receptor</b> , I2CCRM=0: I2CNDAT número de bytes recibidos y todos los datos leídos de I2CDR. <b>Maestro receptor</b> , I2CRM=1: último byte de datos recibido, I2CSTP encendido y todos los datos leídos de I2CDR. <b>Esclavo transmisor</b> : condición de STOP detectada. <b>Esclavo receptor</b> : condición de STOP detectada y todos los datos leídos de I2CDR.
<b>RXRDYIFG</b>	Interrupción por recepción lista. Esta bandera es activada cuando el módulo I2C ha recibido un nuevo dato. Esta bandera es automáticamente limpiada cuando I2CDR es leída y el buffer de recepción está vacío. Aplica solo en modo recepción.
<b>TXRDYIFG</b>	Interrupción por transmisión lista. Esta bandera se enciende cuando el módulo está listo para una nueva transmisión de datos (maestro) o cuando algún maestro solicita datos (esclavo). TXRDYIFG es automáticamente limpiada cuando I2CDR y el buffer de transmisión está lleno.

<b>GCIFG</b>	Interrupción por llamada general. Esta bandera es activada cuando el módulo recibe la dirección de llamada general (00h). Solo aplica en modo receptor.
<b>STTIFG</b>	Interrupción por detección de una condición de START. Esta bandera es encendida cuando el módulo detecta una condición de START en el modo esclavo. Esto permite al MCU estar en un modo de bajo consumo con la fuente del reloj I2C desactivada hasta que un maestro inicia la comunicación. Aplica solo en modo esclavo.

Tabla 5.2 Descripción de banderas de interrupción.

El microcontrolador fungirá como un dispositivo del tipo esclavo, siendo el dispositivo maestro la computadora de vuelo pues este es el encargado de procesar la información de los demás subsistemas. La información recibida por el subsistema mediante el cordón umbilical o los chips de radiofrecuencia se almacena en el microcontrolador para después ser enviada hacia el buffer de transmisión del protocolo y por medio de este periférico es transferida hacia la computadora de vuelo una vez que lo solicite este subsistema.

De igual manera la computadora de vuelo le envía información por medio de este protocolo, recibiendo dicha información en el buffer de recepción de I2C, se almacena en el microcontrolador para después ser enviada por los chips de radiofrecuencia o por el cordón umbilical.

Para la realización de esta comunicación, se utilizaron las banderas de interrupción RXRDYIFG y TXRDYIFG, las cuales corresponden a recepción y transmisión del bus I2C.

El tamaño de las variables que se deseaba ocupar, así como con las opciones que tiene el maestro de manipular al esclavo como ponerlo en modo receptor o transmisor, fueron parámetros que se fueron probando para comprobar la correcta operación del protocolo.

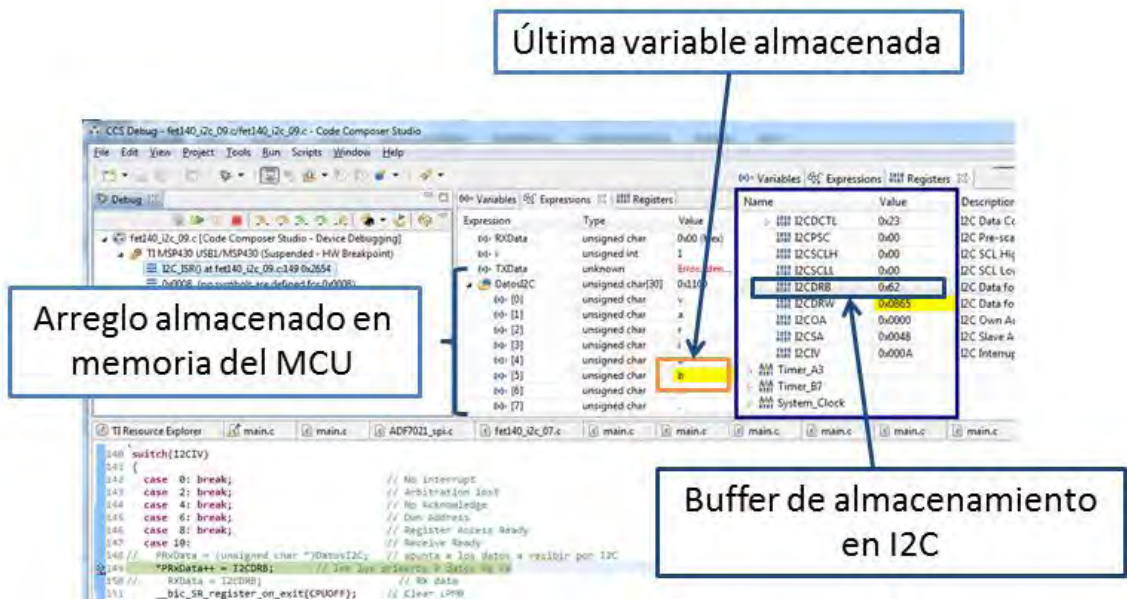


Figura 5.10 Datos enviados de un dispositivo esclavo a un maestro utilizando I2C

#### **5.4.4 Comunicación inalámbrica entre radios**

Al no contar con una TNC con la cual se pudiera comprobar la recepción y transmisión de datos a través de los radios, se ensamblaron dos tarjetas. Siendo una de ellas con la que se simuló la parte de estación terrena a la cual no se le colocó el amplificador de potencia. Esto con el fin de validar software básico en cuanto a los chips de radiofrecuencia que se utilizaron.

Cabe señalar, que para realizar estas pruebas, fue necesario crear una tarjeta donde se colocaron las antenas. En esta tarjeta solo se realizó el acople de impedancias mediante un software llamado Microwave Office, para el cual se utilizó un arreglo del tipo PI, el valor de los capacitores e inductores se ajustó para obtener el mejor coeficiente de transmisión a la frecuencia que fueron utilizadas.

Para poder realizar pruebas con cada uno de los circuitos de radiofrecuencia, es necesario tener claro los parámetros a configurar como frecuencia, esquema de modulación, ancho de banda y número de canales. Una vez teniendo claros estos parámetros los fabricantes de los circuitos integrados; cuentan con una herramienta la cual facilita la configuración de los registros de control.

##### **5.4.4.1 Smart RF Studio 7**

Esta aplicación de *Texas Instruments* para Windows, se enfoca en la evaluación y configuración de sus circuitos de radiofrecuencia de baja potencia. Esta aplicación facilita la tarea a los diseñadores de sistemas de Radiofrecuencia, ya que específicamente se utiliza para generar o configurar los valores de los registros para pruebas prácticas de los chips, así como para encontrar los valores óptimos de componentes externos [33].

Para este trabajo, esta herramienta se utilizó para la configuración de los registros, pues dependiendo los parámetros que se insertaran en la aplicación, los registros se cambiaban. Aspectos como la frecuencia a operar, el ancho de banda deseado, el tipo de modulación, así como el número de canales a utilizar fueron algunos de los parámetros que se colocan al utilizar este software.

Al tener todos los parámetros dentro de la aplicación, el software te muestra los registros y como se van modificando al cambiar alguno de los parámetros. Por último, genera un archivo cuyo contenido son los valores que deben ir en cada registro.



Figura 5.11 Programa Smart RF Studio 7

#### 5.4.4.2 ADF7021 Evaluation Software

Analog Devices cuenta con este software para probar sus circuitos de RF, así como sus tarjetas de prueba. A partir de este se puede seleccionar desde el tipo de comunicación que se desea tener con el transceptor, así como elegir el oscilador controlado por voltaje a utilizar, esquemas de modulación y frecuencia empleada.

Conforme se van modificando los parámetros, se observa el cambio de valores en los registros de configuración, los cuales se encuentran en la parte inferior del panel.

Este software está diseñado para obtener de una manera más precisa los valores de los registros, simplificando la obtención de estos valores. Además de contar con una interfaz muy amigable con el usuario ya que se encuentra dividido de tal forma que los parámetros a modificar sean fáciles de comprender.

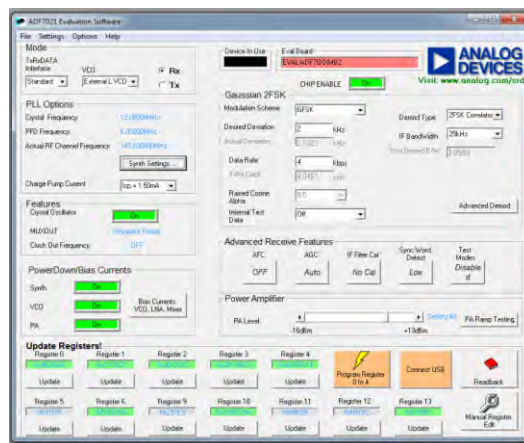


Figura 5.12 Ventana de ADF7021 Evaluation Software

### 5.4.4.3 Prueba de comunicación inalámbrica entre transceptores CC1101

Para la configuración del CC1101, es utilizada la interfaz SPI, con dicha interfaz se puede programar:

- Modo transmisor o receptor
- La selección de canales de radiofrecuencia
- Taza de transmisión
- Formato de modulación
- La potencia de salida

Después de reiniciar el circuito, todos los registros tienen un valor por default por lo tanto se necesita configurar a través de la interfaz SPI.

La configuración es vía cuatro líneas, compatible con la interfaz SPI (SI, SO, SCLK y CS), donde el CC1101 es un esclavo. Esta interfaz también es utilizada para leer y escribir datos almacenados y todas las transferencias comienzan por el bit más significativo MSB por sus siglas en inglés.

Todas las transacciones en SPI comienzan con un byte de encabezado conteniendo un bit de lectura/escritura, uno de acceso ráfaga y una dirección de 6 bits.

El pin CS debe mantenerse en un nivel bajo durante la transferencia de datos en el bus. En caso contrario la transmisión se verá cancelada. En la figura 5.13 se observa un diagrama de tiempos donde se muestran los tiempos requeridos en las líneas del SPI, referenciados a la tabla 5.3.

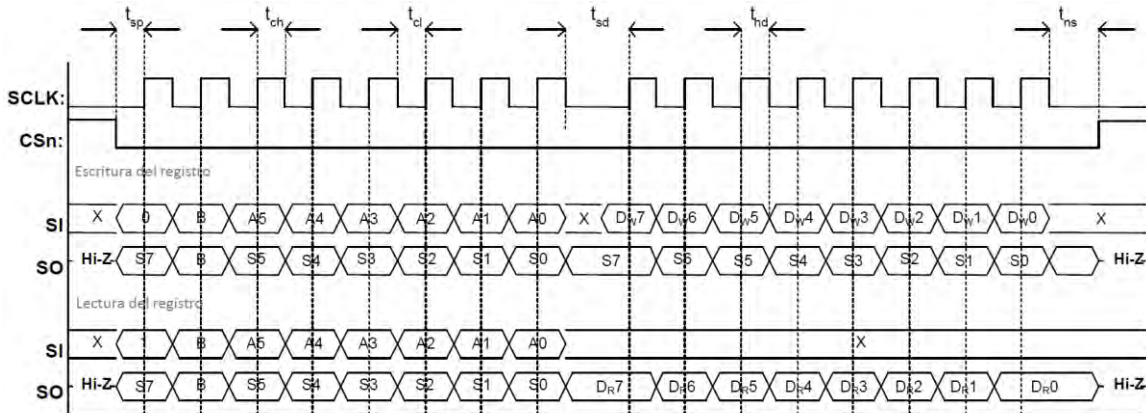


Figura 5.13 Operaciones de escritura y lectura de registros.

Parámetro	Descripción	Min	Max	Unidades
$f_{SCLK}$	Frecuencia SCLK Retardo de 100ns entre un byte de dirección y uno de datos (acceso unitario), o entre dirección y dato y entre dato y dato (acceso ráfaga).	-	10	MHz
	Frecuencia SCLK, acceso unitario No hay retardo entre bytes de dirección y dato.	-	9	



	Frecuencia SCLK, acceso ráfaga No hay retardo entre bytes de dirección y dato, o entre bytes de datos.		-	6.5	
<b>t<sub>sp,sd</sub></b>	CS bajo a flanco de subida en SCLK en modo apagado		150	-	μs
<b>t<sub>sp</sub></b>	CS bajo a flanco de subida en SCLK en modo activo		20	-	ns
<b>t<sub>ch</sub></b>	Reloj en alto		50	-	ns
<b>t<sub>cl</sub></b>	Reloj en bajo		50	-	ns
<b>t<sub>rise</sub></b>	Tiempo de subida del reloj		-	40	ns
<b>t<sub>fall</sub></b>	Tiempo de bajada del reloj		-	40	ns
<b>t<sub>sd</sub></b>	Configuración de datos ( bajo SCLK) hasta un nuevo flanco de subida en SCLK (Aplica entre bytes de dirección y datos, y entre datos).	Acceso unitario	55	-	ns
		Acceso ráfaga	76	-	
<b>t<sub>hd</sub></b>	Dato mantenido en alto después de un flanco de subida en SCLK		20	-	ns
<b>t<sub>ns</sub></b>	Flanco de bajada en SCLK para CS en alto		20	-	ns

*Tabla 5. 3 Requisitos de tiempo de la interfaz SPI.*

Cada vez que es enviado un byte de cabecera, un dato o un byte de comando instantáneo el CC1101 envía por el pin SO un byte de Status. Este byte contiene información de utilidad al MCU, la cual se despliega de la siguiente forma:

1. El primer bit es la señal CHIP\_RDYn, esta señal nos indica que el cristal opera con normalidad.
2. Los siguientes tres bits nos indican el modo en el que se encuentra la máquina de estados:

Valor	Estado	Descripción
<b>000</b>	IDLE	Estado IDLE
<b>001</b>	RX	Modo receptor
<b>010</b>	TX	Modo transmisor
<b>011</b>	FSTXON	Transmisión rápida
<b>100</b>	CALIBRATE	Calibración de sintetizador de frecuencia en proceso
<b>101</b>	SETTLING	PLL es configurado
<b>110</b>	RXFIFO_OVERFLOW	El buffer de recepción se desbordó. Leer cualquier dato útil y después limpiar el buffer con SFRX.
<b>111</b>	TXFIFO_UNDERFLOW	El buffer de transmisión se desbordó. Se reconoce con SFTX.

*Tabla 5.4 Descripción de estados del byte de estatus*

3. Los últimos cuatro bits contiene el valor del registro FIFO\_BYTE\_AVAILABLE, el cual indica la cantidad de bytes disponibles en los buffers de transmisión o recepción, esto depende del bit de lectura/escritura contenido en el byte de cabecera que se le envía al chip:
  - ✓ Bit de lectura/escritura = 1, muestra la cantidad de bytes libres en el buffer de recepción.
  - ✓ Bit de lectura/escritura = 0, muestra la cantidad de bytes libres en el buffer de transmisión.

El bit de lectura/escritura es el que controla si un registro debe ser leído o escrito. Cuando se escribe en un registro, el byte de status es enviado en la terminal SO cada que se envía un byte de cabecera, o de datos en SI. Cuando se lee de los registros el byte de status se envía cada que es recibido un byte de cabecera en SI.

Los registros que cuentan con una dirección consecutiva pueden ser accedidos de una forma más eficiente, configurando el bit de ráfaga contenido en el byte de cabecera.

Existen unos comandos conocidos como comandos “estrobo” los cuales son instrucciones de un solo byte, estos comandos son ejecutados inmediatamente y son utilizados para deshabilitar el cristal, habilitar el modo receptor o hacer que despierte el radio, cambiando el estado en el que se encuentra en cualquier momento.

El CC1101 cuenta con buffers de transmisión y recepción por separado. Cuando el bit R/W es 0 se accede al buffer de transmisión, mientras que si es 1 se accede al buffer de recepción. Se puede acceder de modo unitario y modo ráfaga. Para el primer modo se espera una cabecera sin habilitar el bit ráfaga y con un solo byte de dato, seguido de una nueva cabecera. En el modo de ráfaga se espera una sola cabecera y bytes consecutivos de datos hasta que se termina el acceso cuando CS va a alto.

Los siguientes bytes de cabecera acceden a los buffers:

- 0x3F: acceso de un solo byte al buffer de transmisión.
- 0x7F: acceso de modo ráfaga al buffer de transmisión.
- 0xBF: acceso de un solo byte al buffer de recepción.
- 0xFF: acceso de modo ráfaga al buffer de recepción.

0x3E es la dirección para configurar el PATABLE y se utiliza para seleccionar la configuración de control y alimentación del PA (AMPLIFICADOR DE POTENCIA). Primero es esperada la dirección del registro seguida de un byte en el cual contiene el valor de la potencia que se requiere. En modo SLEEP se pierde este valor.

Adicionalmente a los pines de SPI, se integran dos pines adicionales de control GDO0 y GDO2, los cuales contienen información de utilidad en cuanto a software de control. Estos dos pines pueden ser configurados para generar interrupciones en el MCU.

El CC1101 cuenta con soporte en hardware de paquetes orientado a protocolos de radio. Para el modo transmisor, puede ser configurado para agregarle los siguientes elementos al paquete almacenado en el buffer de transmisión:

- \* Un número programable de Bytes de preámbulo.
- \* Una palabra de sincronización de 2 bytes, inclusive se puede duplicar para ser de 4 bytes.
- \* Un CRC sobre los datos del campo, dentro de este campo se pueden implementar un blanqueamiento de datos con una secuencia PN9.
- \*

El formato que maneja el CC1101 puede ser configurado y consiste en los siguientes campos:

- Preámbulo. Es una secuencia de unos y ceros alternos, su longitud es programada y es lo primero que se envía al estar en modo TX.
- Palabra de Sincronización. Una vez que es enviado el número programado de bytes de preámbulo, se transmite la palabra de sincronización. Esta palabra tiene un valor de 2 bytes localizados en los registros SYNC0 y SINC1, proporcionando sincronía en los paquetes que van entrando.
- Byte de longitud (opcional). Este byte indica el número de bytes que se pretende recibir en el modo de longitud variable.
- Byte de dirección (opcional). Este byte contiene una dirección la cual debe coincidir con una dirección válida, de ser así el paquete es recibido.
- Datos. Es la información que se pretende enviar.
- CRC de 2 bytes (opcional). Se encarga de verificar la información, calculando un checksum sobre todos los datos del buffer.

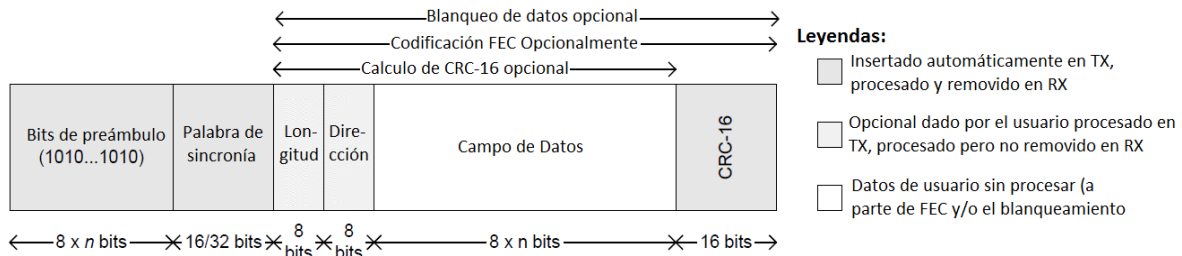


Figura 5.14 Formato del paquete

Cabe señalar que si el buffer se encuentra vacío y se intenta enviar algo, el radio entra en un estado llamado UNDERFLOW del cual solo podrá salir emitiendo un comando estrobo SFTX.

El demodulador y el facilitador de paquetes buscará un preámbulo válido y una palabra de sincronización, cuando la encuentra el demodulador obtiene los dos y recibe los datos por último comprueba el CRC.

### Control del radio

El CC1101 cuenta con una máquina de estados, la cual va cambiando entre diferentes estados operacionales. El cambio entre estos estados se puede observar en la figura 5.15. Estos cambios de estado se realizan a través de comandos Estrobos o por eventos internos como los desbordamientos de los buffers de almacenamiento.



Figura 5.15 Diagrama de estados del CC1101

Cuando por primera vez le es suministrado voltaje al CC1101 el sistema debe ser reiniciado. Esto se logra por medio de dos formas una que es mediante un reinicio automático y la otra es por medio de una manual. La primera de ellas se realiza mediante un circuito de reinicio incluido en el CC1101 realizando una secuencia interna donde CHIP\_RDYn cambia de un nivel lógico alto a bajo. Esto se observa en la salida del esclavo (SO) después de que el pin CS va a un nivel lógico bajo. Mientras que el reinicio manual consta de los siguientes pasos:

1. Se coloca a SCLK=1 y a SI=0 para evitar problemas con el modo control.
2. Realizar un cambio de bajo a alto súbito.
3. Mantener CS en bajo y después en alto al menos 40µs para después ponerlo en bajo.
4. Colocar a CS bajo y esperar a que SO vaya a bajo

5. Emitir el comando estrobo SRES en la línea SI.
6. Cuando SO vaya a bajo de nuevo, el proceso de reinicio es completado.

Una vez que se reinicia por completo el circuito el radio se encuentra en el estado IDLE. Los dos modos activos transmisión y recepción son activados mediante los comandos estrobo SRX y STX enviados vía SPI o automáticamente por una función llamada Wake On Radio (WOR).

El sintetizador de frecuencia debe ser calibrado regularmente. Existen cuatro formas de hacerlo, una manual aplicando el comando estrobo SCAL y otras tres automáticas controladas por un registro, las cuales ocurren cuando cambia de estado IDLE a RX o TX, de TX o RX a IDLE o cada cuatro cambios entre estos estados. El microcontrolador puede cambiar de TX a RX y viceversa utilizando los comandos estrobo correspondiente.

La función Wake On Radio (WOR) habilita al CC1101 para que periódicamente despierte del modo bajo consumo (SLEEP) al escuchar un paquete entrante sin la necesidad de interactuar con el MCU. Este modo comienza al ser enviado el comando estrobo SWOR, una vez que CS este en un nivel alto, el CC1101 entrará en modo bajo consumo.

Un contador será el encargado de cambiar entre los estados RX y SLEEP. Se programa el tiempo que se desea permanezca en el estado RX y después pase al estado SLEEP, si no recibió dato alguno. Para salir de este estado se debe regresar al estado IDLE [34].

Los buffers de transmisión y recepción pierden su contenido en el estado SLEEP.

Las pruebas realizadas con el circuito de radiofrecuencia, se basaron en desarrollar un software básico tomando en cuenta la información antes citada. Primero se realizó la asignación y declaración de los GPIOs que fueron utilizados, una vez hecho esto se crearon funciones, mediante las cuales se realizaba un procedimiento en específico. Acciones como el configurar los pines que se utilizan en el puerto SPI, reiniciar al CC1101 o leer datos y escribirlos sobre un registro en particular son ejemplo de estas funciones.

Al tener las funciones, se procedió a organizarlas de acuerdo a la secuencia de inicio del circuito de radiofrecuencia. La secuencia que se siguió es la siguiente:

1. Asignación de GPIOs de acuerdo al uso que tengan: entradas, salidas o que sean parte de algún periférico.
2. Escribir sobre los registros de configuración del CC1101, los valores obtenidos con la herramienta *Smart RF Studio*.
3. Crear una rutina en la cual se transmitieran o se recibieran los datos.

La primera prueba que se realizó fue el verificar que se estaban escribiendo de manera correcta los registros de configuración. Para lo cual, una vez que se escribió sobre los registros se procedía a leer el dato que contenía el mismo registro. Si el valor obtenido en la lectura del registro era el mismo que se había escrito anteriormente, esto significaba que la configuración de registros era correcta y que las funciones funcionaban correctamente.

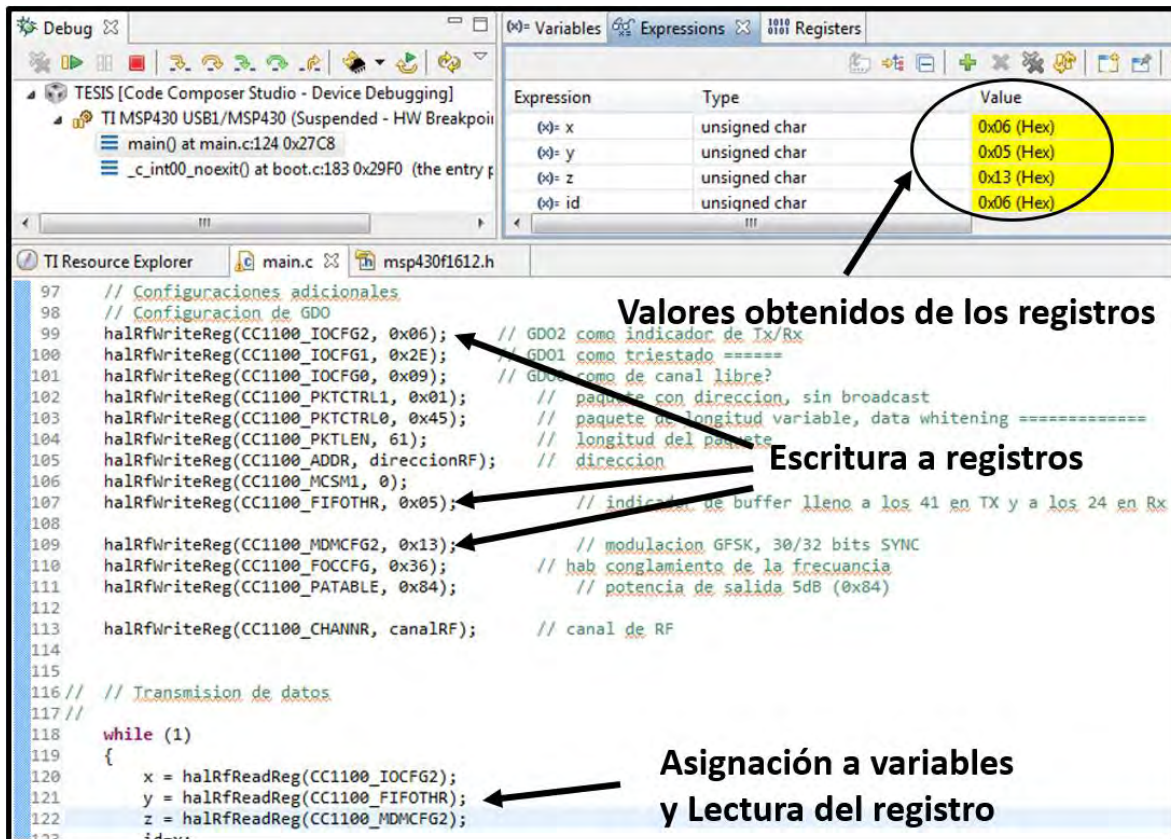


Figura 5.16 Escritura y lectura de registros.

Una vez verificada la lectura y escritura de registros, se comenzó con rutinas para transmitir o recibir datos. Para el caso de transmisión, se realizó lo siguiente:

1. Se comenzó con limpiar el buffer de transmisión mediante el comando estrobo SFTX.
2. En un arreglo se coloca la información que se desea transmitir.
3. El arreglo es colocado en el buffer de transmisión del CC1101.
4. El arreglo es transmitido por medio del comando estrobo STX.

Una señal de suma importancia es la señal de control GDO2 la cual es configurada para mandar pulsos al microcontrolador una vez que se transmitía o recibía información en el CC1101.

En el caso de la recepción se procedió de la siguiente manera:

1. Se limpia el buffer de recepción por medio del comando estrobo SFRX.
2. Activa el modo de recepción mediante el comando estrobo SRX.
3. Lectura del número de bytes contenidos en el buffer de recepción.
4. Se almacenan los datos del buffer en una variable y se vuelve a limpiar el buffer.

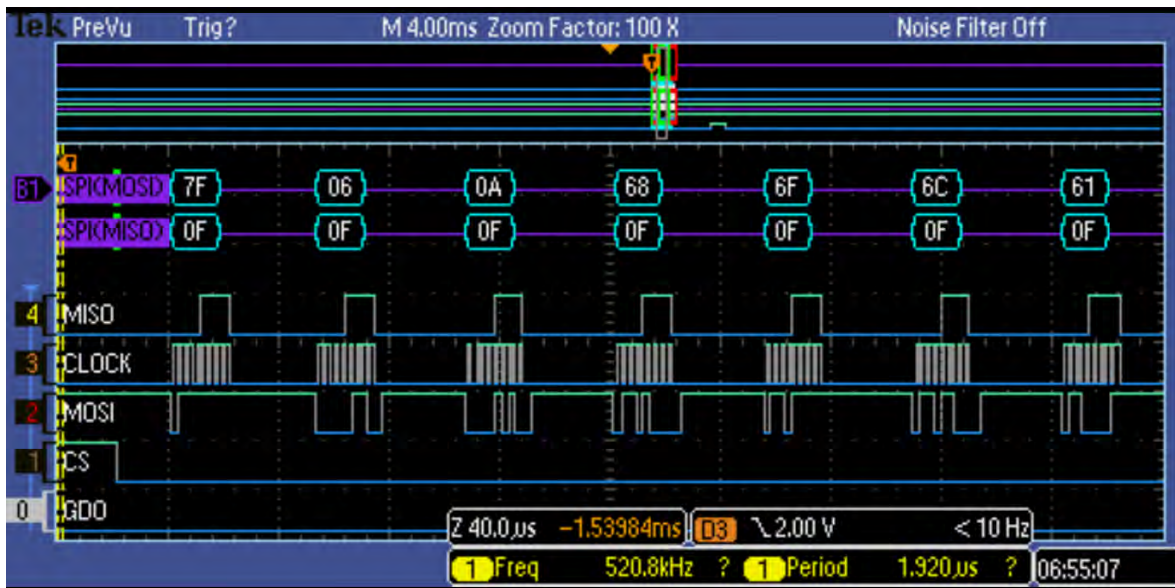


Figura 5.17 Datos transmitidos hacia el CC1101 por medio de protocolo SPI

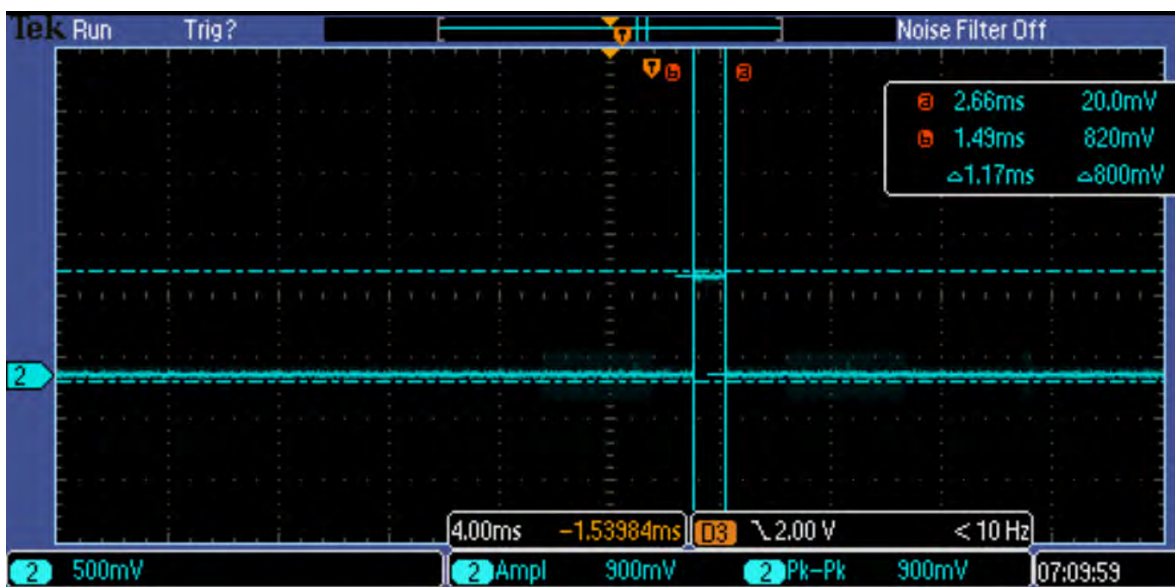


Figura 5.18 Tiempo que tarda en transmitir un paquete de datos el CC1101

En las figuras 5.17 se observa el comportamiento de algunas de las señales que se utilizaron para enviar información al CC1101, especialmente la señales del protocolo SPI. La señal GDO2 envía un pulso al ser finalizada la transmisión de datos a través del CC1101, esto se comprobó midiendo el espacio que se tenía entre el último dato enviado SPI y dicho pulso. A su vez en la figura 5.18 se muestra una lectura que se realizó en la salida del CC1101, en esta lectura se aprecia un pulso de 1.17ms de duración, duración que coincide con el tiempo que tarda en ser enviada la señal de aviso del GDO2.

#### 5.4.4.4 Prueba de comunicación inalámbrica entre transceptores ADF7021

Para poder entablar una comunicación exitosa entre los transceptores ADF7021 y el microcontrolador existen varias opciones. El circuito cuenta con una interfaz del tipo serial capaz de programar los registros de 16 y 32 bits, dicha interfaz está compuesta por tres líneas (SCLK, SDATA y SLE) y consiste en un cambiador de nivel, un registro de desplazamiento de nivel y 16 latches. Las señales deben ser compatibles con CMOS y la interfaz es desactivada cuando CE está en un nivel lógico bajo.

Los datos son almacenados en el registro siendo los primeros bits en ser almacenados los más significativos. Los datos son transferidos a uno de los 16 latches identificando al latch destino por medio del valor de los bits de control, los cuales son los cuatro bits menos significativos del registro. Los datos solo pueden ser leídos en el pin SREAD. La figura 5.19 muestra el diagrama de tiempos de este modo serial.

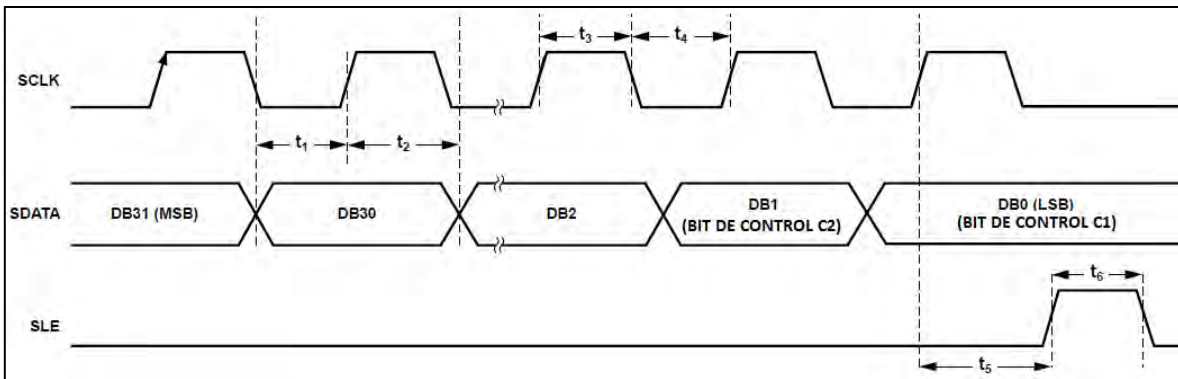


Figura 5.19 Diagrama de tiempos de la interfaz serial

Existe un proceso llamado lectura de retorno (readback). Esta operación es iniciada escribiendo una palabra de control válida al registro de lectura de retorno y habilitando el bit READBACK. Esta operación puede empezar una vez que la palabra de control haya sido registrada por la señal SLE. SLE debe permanecer en alto mientras los datos están siendo leídos. Por cada pulso de reloj que sale en el pin SCLK la lectura se va tomando en el pin SREAD comenzando con el bit más significativo. Se necesitan 18 ciclos de reloj para poder realizar esta operación de lectura, uno para que los datos se carguen al registro, 16 para transmitir los datos y uno más para regresar al pin SREAD a una posición de tres estados. Una vez realizados estos 18 ciclos de reloj SLE debe estar en un nivel bajo.



Mediante esta operación se pueden verificar valores como:

- RSSI: esta lectura contiene comprimido el indicador de potencia de señal recibida (RSSI), la ganancia del filtro de corriente y la ganancia de corriente del amplificador de bajo ruido.
- VOLTAJE DE LA BATERIA y SENSOR DE TEMPERATURA: mide el voltaje en el pin VDD4 arrojando la lectura en los bits RV1 al RV7 junto con la temperatura censada.
- REVISION DEL SEMICONDUCTOR (SILICIO): Este valor es válido sin configurar algún otro registro. Contiene el código del producto el cual es 0x210 y el código de revisión igual a 0x4
- CALIBRACION DEL ANCHO DE BANDA DEL FILTRO: este valor de 8 bits puede ser utilizado para poder ajustar el filtro manualmente evitando la necesidad de hacer una calibración al filtro.

La siguiente figura muestra el diagrama de tiempos donde se observa el comportamiento de las líneas SCLK, SDATA, SLE y SREAD, involucradas en esta operación.

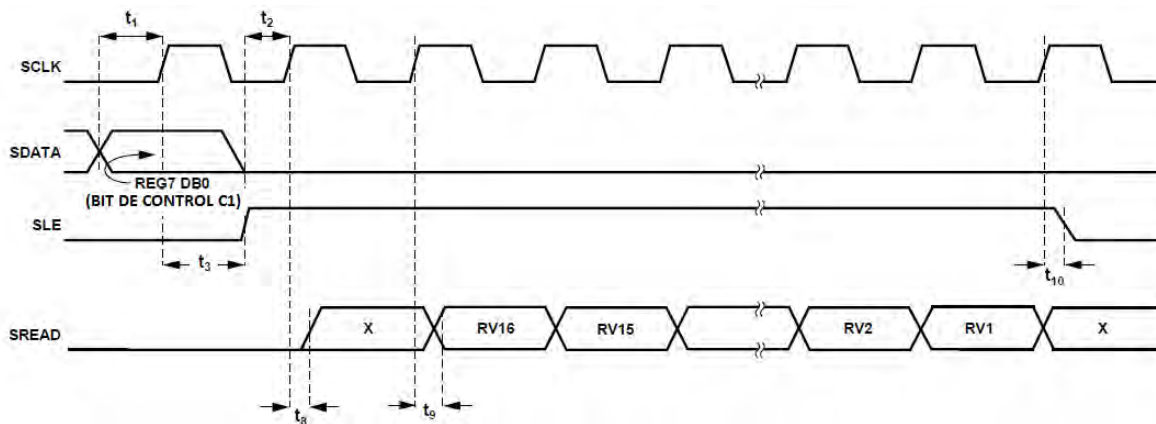


Figura 5.20 Diagrama de tiempos de la interfaz Readback

Otra forma de comunicarse es vía SPI, dicho protocolo es habilitado configurando el bit 28 del registro R0 a alto y escribir el valor de 0x07 en los bits del 17 al 19 del registro R15. Por medio de esta interfaz de comunicación, el pin TxRxCLK es configurado como una entrada transmisora de datos en el modo transmisor. Mientras que en el modo de recepción, los datos son recibidos en el pin TxRxDATA. La señal de reloj en ambos modos transmisión y recepción se genera en el pin CLKOUT, siendo los datos transmitidos o recibidos en cada

período de esta señal. En las figuras 5.21 y 5.22 se muestran los diagramas de tiempos para el modo transmisor y el modo receptor.

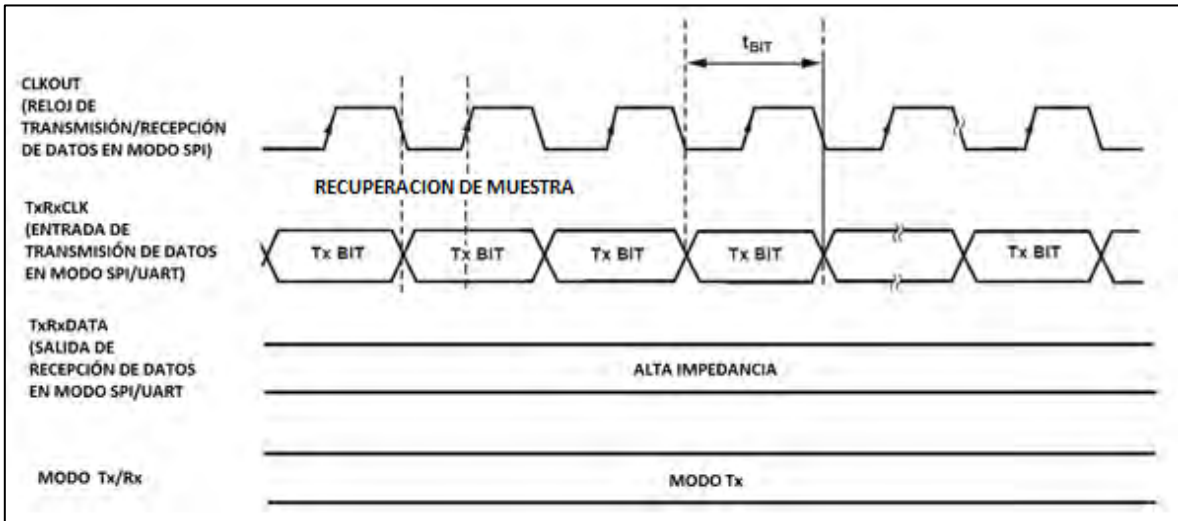


Figura 5.21 Diagrama de tiempos para transmitir en modo SPI/UART

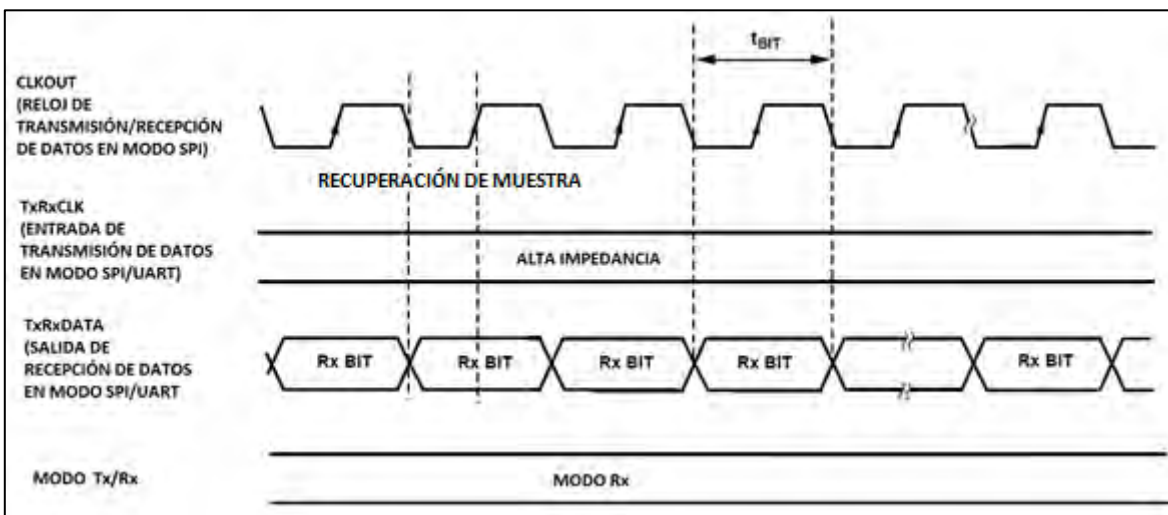


Figura 5.22 Diagrama de tiempos para recepción en modo SPI/UART

Para poder entrar en los modos de transmisión y recepción, se deben configurar en un orden específico los registros del ADF7021. En la tabla 5.5 se observan los registros que se necesitan configurar para poder habilitar estos dos modos, después de que el pin CE va a alto, así como en la figuras 5.23 y 5.24 donde se observa la secuencia de configuración de los registros [35].

Modo	Registros				
Tx	Reg 1	Reg 3	Reg 0	Reg 2	
Rx	Reg 1	Reg 3	Reg 0	Reg 5	Reg 4
De Tx a Rx y viceversa	Reg 0				

Tabla 5.5 Registros mínimos requeridos para configurar Tx y Rx

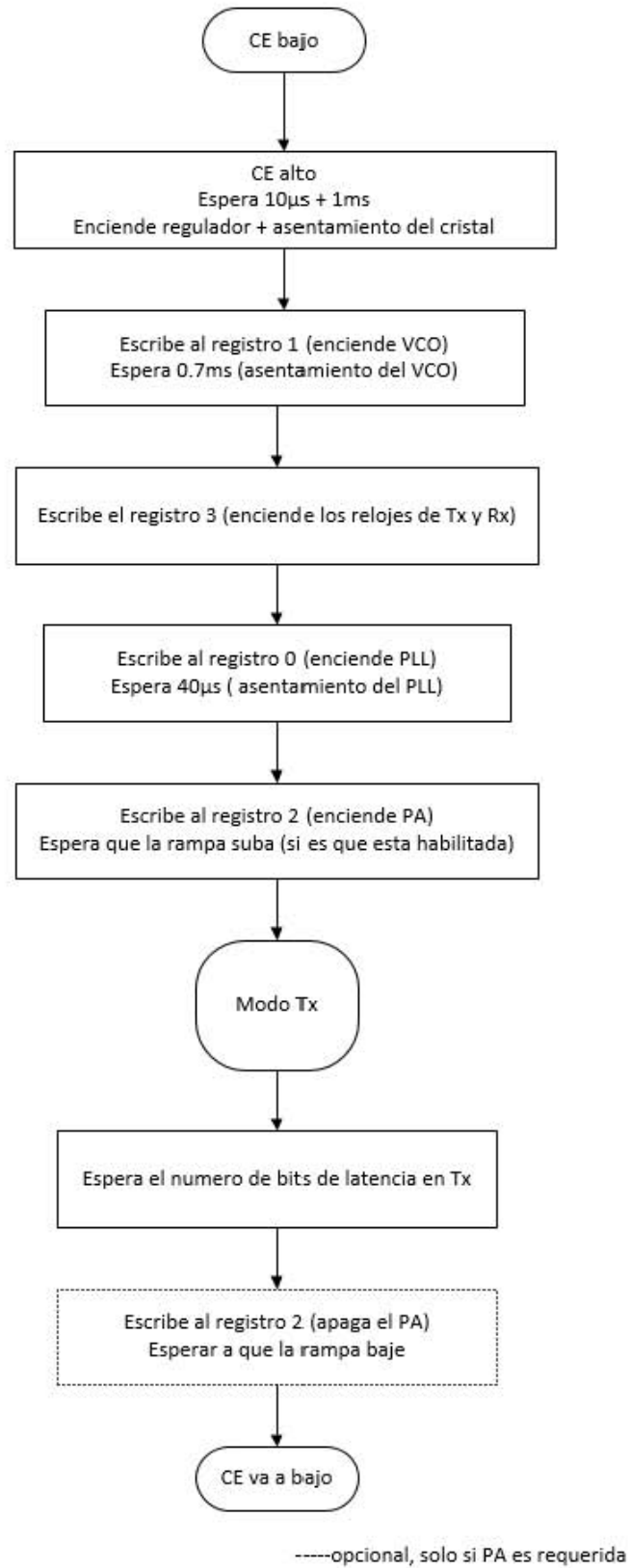


Figura 5.23 Secuencia a seguir en modo transmisor

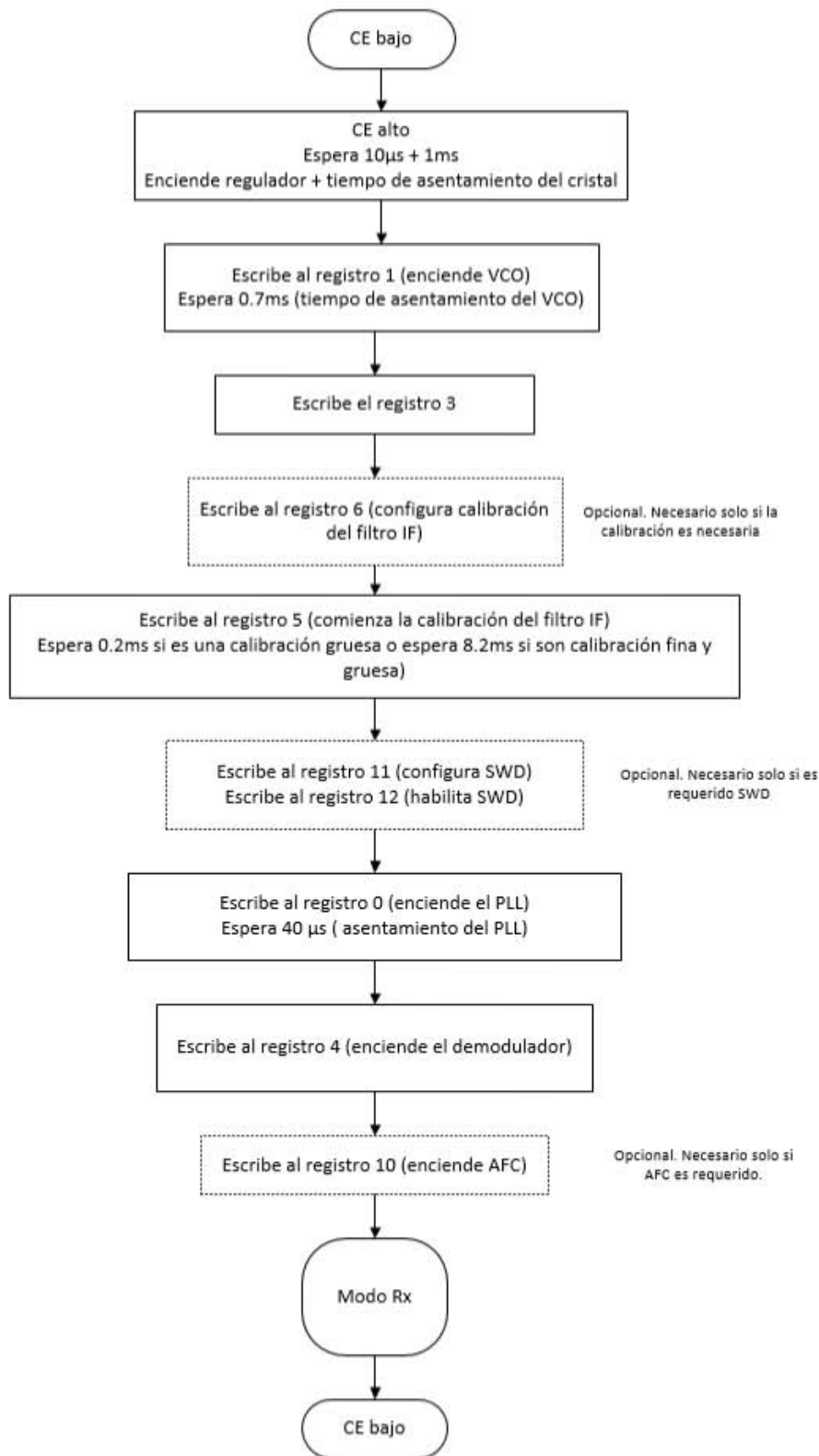


Figura 5.24 Secuencia a seguir en modo receptor

En cuanto a pruebas de software realizadas a estos chips de radiofrecuencia se comenzó con la lectura de registros por medio de la función “readback”. Siguiendo el diagrama de tiempos de este modo, el cual se muestra en la figura 5.20, se realizó la programación necesaria para obtener las lecturas de algunos registros, en particular se realizaron las lecturas de la revisión del semiconductor, el voltaje de batería y del sensor de temperatura.

En la figura 5.25 se observa el registro No. 7, el cual se configura de acuerdo al registro que se desea leer por medio de la función readback.

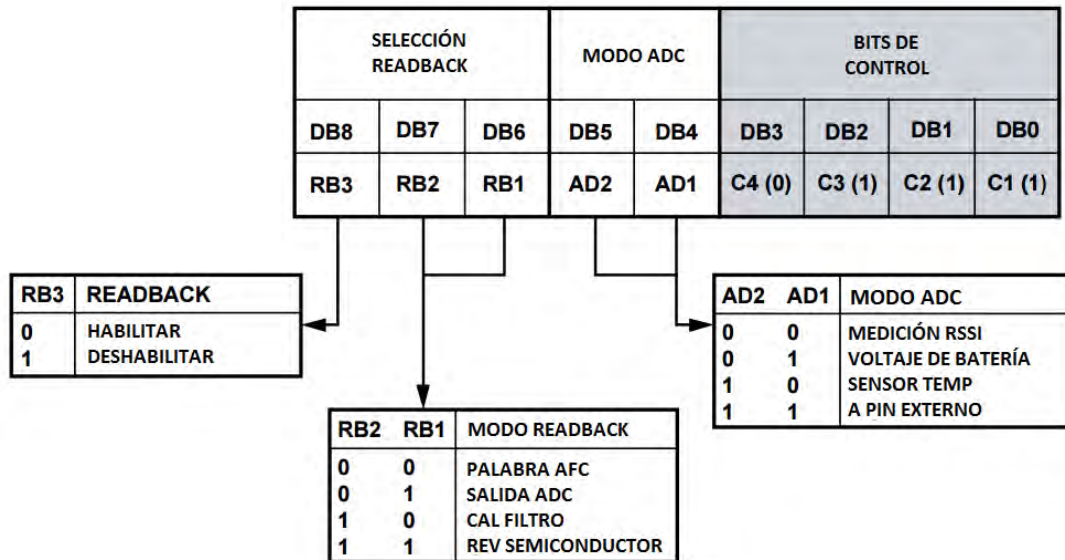


Figura 5.25 Desglose de registro de configuración Readback No.7

Para la revisión del semiconductor al registro No. 7 se le carga el número 0x1C, con el cual se seleccionan la función READBACK y el pin de REVISIÓN DEL SEMICONDUCTOR. La respuesta fue de manera exitosa ya que se obtuvieron los valores que la hoja de especificaciones nos brinda, para el código del producto se obtuvo el valor de 0x2104 mientras que para el código de revisión el número obtenido fue 0x04.

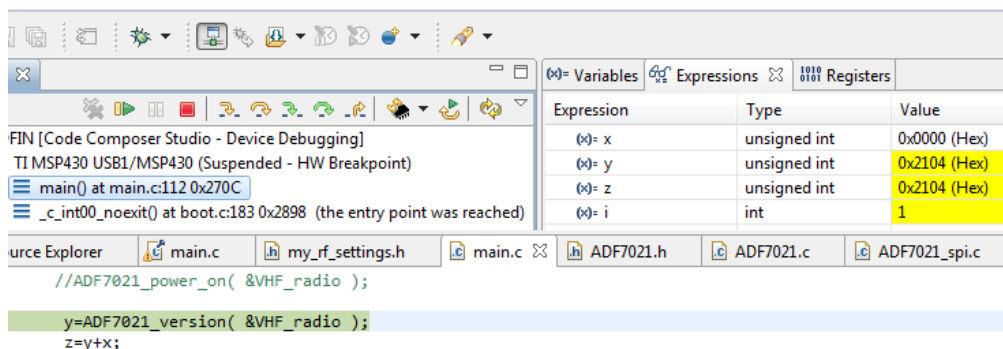


Figura 5.26 Obtención de la revisión del semiconductor.

Para obtener la lectura del sensor de temperatura se colocó en el registro No.7 el valor hexadecimal 0x16, con el cual se obtuvo el valor decimal 62. Con este valor se obtiene mediante la siguiente fórmula el valor de la temperatura en grados centígrados:

$$Temperatura [^{\circ}C] = 469.5 - (7.2 \times TEMP\_READBACK)$$

$$Temperatura = 23.1^{\circ}C$$

Dicho resultado fue comprobado con un sensor de temperatura que se encontraba en el laboratorio, en la figura 5.27 se observa el resultado obtenido del sensor de temperatura, así como del voltaje de batería.

Para la lectura en el voltaje de batería se le cargó el valor 0x015 al registro de configuración, obteniendo un valor de 70, dicho valor se ingresa a la siguiente fórmula, obteniendo como resultado:

$$V_{bateria}[V] = \frac{VOLTAJE_{BATERIA}}{21.1}$$

$$V_{bateria} = 3.3175 V$$

Valor que concuerda, ya que dicho valor se puede medir físicamente en el pin No. 9 que suministra el voltaje al amplificador de bajo ruido (LNA) y al bloque mezclador, voltaje conocido como  $V_{DD4}$ .

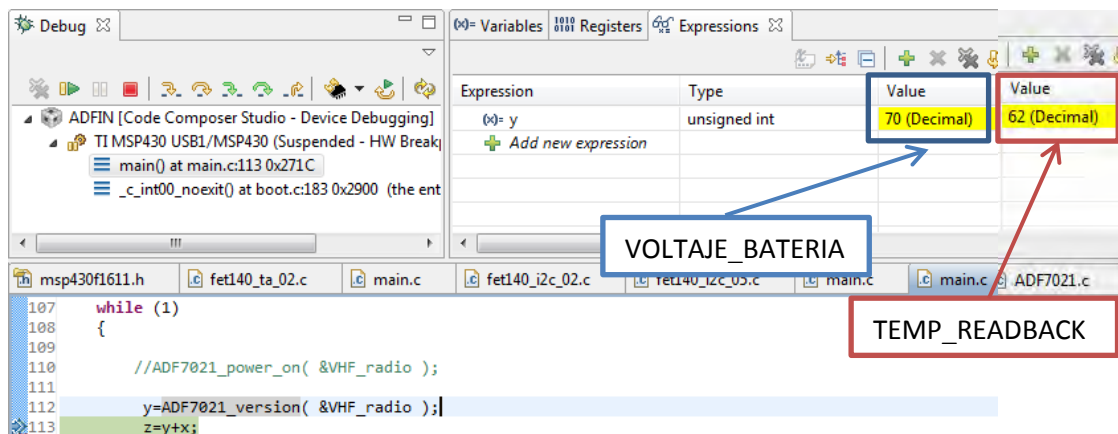


Figura 5.27 Valores obtenidos mediante la función Readback.

Actualmente se tiene la documentación necesaria para seguir trabajando en dichas pruebas sin embargo debido a diversas circunstancias y primordialmente cuestiones de tiempo no se han podido finalizar, quedando fuera de este trabajo, no sin antes mencionar que se sigue trabajando para poder obtener una transmisión satisfactoria.

# Conclusiones y Recomendaciones

---

## 8.1 Conclusiones

Del trabajo de tesis se concluye con:

- Se realizó el diseño de una tarjeta electrónica compatible con el estándar CubeSat en dimensiones y forma. Además de no presentar ningún tipo de inconveniente mecánico respecto a la interacción con los demás subsistemas del nanosatélite.
- El diseño del subsistema de comunicaciones contempla circuitos de protección contra efecto Latch Up, dichos circuitos se generaron para dos casos en cuanto a la corriente consumida por los circuitos protegidos. Estos circuitos de protección fueron probados en hardware y software teniendo un resultado favorable.
- Se logró la interacción entre el subsistema y el puerto USB de una computadora, esto gracias a la programación del periférico USART contenido dentro del microcontrolador MSP430F1611 y al circuito FT232R que sirvió de interfaz USB a comunicación serial. Pudiendo comunicarse como un puerto serie convencional.
- Se comprobó la comunicación entre el subsistema de comunicaciones con el subsistema de computadora de vuelo, mediante el uso del protocolo I2C, el cual fue implementado por la unidad USART integrado en el MSP430F1611. El subsistema actúa a manera de esclavo, respondiendo conforme el dispositivo maestro lo indique.
- Se logró la transmisión por medio del chip de radiofrecuencia CC1101, siendo esta transmisión controlada a través del microcontrolador MSP430F1611 utilizando el protocolo SPI, desarrollándose software elemental para comprobar el buen funcionamiento del chip de radiofrecuencia.
- La señal transmitida por el chip CC1101 tiene una potencia aproximada de 1W, siendo esta potencia la necesaria para poder ser recibida por alguna estación terrena si el nanosatélite se lanzara a una órbita baja (LEO).
- La transmisión fue recibida por otra tarjeta simulando ser la estación terrena. Se programó una rutina donde el chip estuviera de modo continuo en estado receptor, esperando a que le llegaran datos, una vez que los datos son recibidos, son reenviados al subsistema de computadora de vuelo.
- En lo que se refiere a la programación del transceptor ADF7021 se pudieron leer registros internos por medio de la función readback. El desarrollo de este software se continúa depurando con la finalidad de poder generar rutinas donde se transmita y se reciba información por medio de este chip.

## 8.2 Recomendaciones

- Continuar con el desarrollo de software operativo, seguir su depuración así como el constante trabajo en conjunto con los demás subsistemas, primordialmente con la computadora de vuelo.
- Generar el software necesario para una adecuada comunicación con el transceptor ADF7021, si bien se pudieron leer algunos registros en específico hace falta el desarrollo de más software en esta parte.
- En cuanto al software, se podría buscar robustecerlo mediante el uso de contadores, esto con la finalidad de evitar que el software quede a la espera de cierta bandera o respuesta después de un tiempo específico.
- En lugar de utilizar dos chips de radiofrecuencia, se puede utilizar un solo transceptor conectado a un circuito especial el cual se coloca entre el transceptor y el arreglo de las antenas. Este circuito denominado Front End Module contiene un amplificador de potencia utilizado en la transmisión de datos y un amplificador de bajo ruido en la recepción de información. El utilizar este circuito ahorraría tanto consumo de energía, ya que solo se tendría un solo circuito así como en espacio ocupado en la tarjeta.
- Al contar con circuitos Front End en cada uno de los dos transceptores, se podrían utilizar cada uno de los transceptores independientemente y no como se tiene actualmente (una frecuencia de subida y otra de bajada).



# Bibliografía y Mesografía

---

- [1] González de la Vara Fernán, *Las ideas del siglo XX*, México 1990, pp. 36
- [2] Carmen Felix, *Mi encuentro con Rusia Breve historia de la Unión Soviética en el Espacio*, Hacia el Espacio, Octubre 2013. [En línea] Available:  
<http://haciaelespacio.wix.com/haciaelespacio007#!la-union-sovietica/cakb>
- [3] CubeSat Project. [En línea] Available: <http://www.cubesat.org/index.php/about-us>
- [4] Roberto Conte Galván, *Tecnología de satélites de comunicaciones para distintas aplicaciones*, México 2014.
- [5] Bryan Klofas, Jason Anderson, *A Survey of CubeSat Communication Systems*, California Polytechnic State University, 2008.
- [6] Bryan Klofas, *CubeSat Communications System Table*, [En línea] Available:  
<http://www.klofas.com/comm-table/>
- [7] Montana Space Grant Consortium, *Hiscock Radiation Belt Explorer (HRBE)*, [En línea] Available:  
<http://spacegrant.montana.edu/Explorer1Prime.html>
- [8] EoPortal Directory, *E1P-2 (Explorer-1 PRIME-2) / HRBE CubeSat Mission*, [En línea] Available:  
<https://directory.eoportal.org/web/eoportal/satellite-missions/e/e1p-2>
- [9] Amateur Radio-PEOSAT, *Radio Aurora Explorer (RAX-2)*. [En línea] Available:  
<http://www.pe0sat.vgnet.nl/satellite/cube-nano-picosats/rax-2/>
- [10] EoPortal Directory, *RAX-2 (Radio Aurora Explorer-2)*. [En línea] Available:  
<https://directory.eoportal.org/web/eoportal/satellite-missions/r/rax-2>
- [11] USC Viterbi School of Engineering, *Viterbi School's AENEAS Satellite program blasting off*. [En línea] Available: <http://viterbi.usc.edu/news/news/2010/viterbi-serc-aeneas.htm>
- [12] EoPortal Directory, *Aeneas nanosatellite of USC/SERC*. [En línea] Available:  
<https://directory.eoportal.org/web/eoportal/satellite-missions/a/aeneas>
- [13] ISIS- Innovative Solutions In Space, *ISIS Small Ground Station Datasheet, Versión 13.6* [En línea] Available:  
[http://www.cubesatshop.com/index.php?page=shop.product\\_details&flypage=flypage.tpl&product\\_id=118&category\\_id=3&option=com\\_virtuemart&Itemid=72](http://www.cubesatshop.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=118&category_id=3&option=com_virtuemart&Itemid=72)
- [14] Tokyo Institute of Technology Lab. For Space Systems, *The use of amateur frequency band for the cubesat project*, [En línea] Available:  
<http://lss.mes.titech.ac.jp/ssp/cubesat/frequency/ama.html>
- [15] Dwi Hartanto, *Reliable Ground Segment Data Handling System for Delfi-n3Xt Satellite Mission*, Netherlands 2009.

- [16] Future Technology Devices International Ltd. [En línea] Available: [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf)
- [17] Maxim integrated. [En línea] Available: <https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS1820.html>
- [18] Texas Instrument. [En línea] Available: <http://www.ti.com.cn/cn/lit/ds/symlink/bq24314.pdf>
- [19] Maxim integrated. [En línea] Available: <https://www.maximintegrated.com/en/products/analog/amplifiers/MAX4372H.html>
- [20] Texas Instrument. [En línea] Available: <http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=lm6511&fileType=pdf>  
<http://www.ti.com/product/cd74hc112>
- [21] NXP Products. [En línea] Available: <http://www.nxp.com/products/discretes-and-logic/logic/dual-supply-translating-transceiver-auto-direction-sensing-3-state:NTB0104>
- [22] Analog Devices. [En línea] Available: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADF7021.pdf>
- [23] Texas Instrument [En línea] Available: <http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=cc1101&fileType=pdf>
- [24] rfmd. [En línea] Available: [http://www.rfmd.com/store/downloads/dl/file/id/27813/rfpa0133\\_data\\_sheet.pdf](http://www.rfmd.com/store/downloads/dl/file/id/27813/rfpa0133_data_sheet.pdf)
- [25] Texas Instrument [En línea] Available: <http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=msp430f1611&fileType=pdf>
- [26] Altium [En línea] Available: <https://techdocs.altium.com/display/ADOH/Altium+Designer>
- [27] Altium [En línea] Available: <https://techdocs.altium.com/display/ADOH/Tutorial+-+Getting+Started+with+PCB+Design>
- [28] E.A. Jiménez Madrigal, subsistema de estabilización activa y sensores para un simulador satelital, México 2009.
- [29] Texas Instrument [En línea] Available: <http://www.ti.com/tool/ccstudio>
- [30] Texas Instrument, MSP430 programming via the JTAG interface, User's Guide, 2010.
- [31] Texas Instrument, MSP430 Hardware tools, User's guide, 2009.
- [32] Texas Instrument, MSP430x1xx Family User's Guide, 2006.
- [33] Texas Instrument, [En línea] Available: <http://www.ti.com/tool/smartrftm-studio>
- [34] Texas Instrument, CC1101 Low Power Sub-1 GHz RF Transceiver, 2015.
- [35] Analog Devices, High Performance Narrow-Bnad Transceiver IC ADF7021, datasheet, 2014.

[36] B. Del Moral Perea, Diseño y fabricación de computadora de 32 bits para satélite experimental, México, 2014.

La EIA (Electronic Industry Association) elaboró la norma RS-232 que define la interfaz mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial.

El RS232 es un protocolo de comunicación serie orientado a palabras de 8 bits, es decir un protocolo donde toda la información es enviada por un solo canal bit a bit (un canal para enviar información y otro para recibirla), y donde lo que se envían son caracteres.

Este protocolo está diseñado para distancias cortas, de unos 15 metros, y se puede trabajar de forma asíncrona o síncrona. Además de contar con canales del tipo simplex, halfduplex y fullduplex.

Los conectores más utilizados para una conexión RS232 son los DB9 y DB25. En la siguiente tabla se observan las señales con mayor frecuencia utilizadas en comunicaciones de este tipo.

No. Conector DB9	Pin	No. Conector DB25	Pin	Nombre	Descripción
1		8		DCD	Línea por la que el receptor informa al transmisor que tiene una portadora entrante.
2		3		RD	Línea de datos del receptor al transmisor.
3		2		TD	Línea de datos del transmisor al receptor.
4		20		DTR	Línea por donde el receptor informa al transmisor que está activo y en buen estado.
5		7		GND	Valor a 0V.
6		6		DSR	Línea por donde transmisor informa al receptor que está activo y en buen estado.
7		4		RTS	Línea en la que el transmisor indica que quiere enviar algo al receptor.
8		5		CTS	Línea en la que se informa que el receptor está preparado para recibir datos.
9		22		RI	Línea en la que se indica que se ha detectado una portadora.

La conexión más sencilla se puede realizar con solo tres de las señales anteriores, TD, RD y GND.

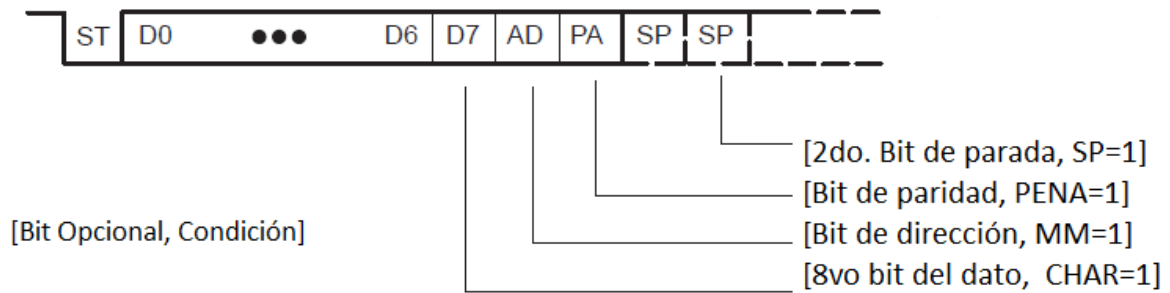
Para que esta comunicación se pueda llevar a cabo es necesario que se cumplan ciertos parámetros tanto en el dispositivo que recibe como en el que transmite, los cuales se mencionaran enseguida.

**Protocolo serie:** en cuanto a este parámetro, se ve involucrado el número de bits a utilizar, la paridad y los bits de parada utilizados. La paridad se utiliza para comprobar la calidad de los datos recibidos. Los bits de datos pueden estar entre los 5 bits y los 8, por último los bits de para consisten en uno o dos bits puestos a un nivel lógico alto.

**Velocidad del puerto:** esta norma puede transmitir los daos a unas velocidades ya determinadas, las cuales van de los 4800 a los 115200 bps.

**Protocolo de control de flujo:** este control de flujo puede ser mediante hardware gracias a las señales RTS y CTS, o por software mediante el XON/XOFF.

Para este trabajo el formato de los datos es el siguiente:



*Figura A.1 formato de datos comunicación serial*

Durante la década de 1980, Phillips desarrolló el bus de dos alambres entre circuitos integrados, nombrado I2C (por sus siglas en inglés Inter-Integrated Circuit) para poder conectar de manera sencilla periféricos múltiples a una unidad de procesamiento central (CPU o MCU) en aplicaciones de televisión.

Como los circuitos se fueron haciendo más complejos con muchas conexiones entre periféricos, se necesitó de un método para simplificar los diseños y reducir los costos. Este protocolo cumple con este requisito por medio de la limitación de pistas en el circuito impreso y reduciendo el uso de estradas-salidas en el microprocesador.

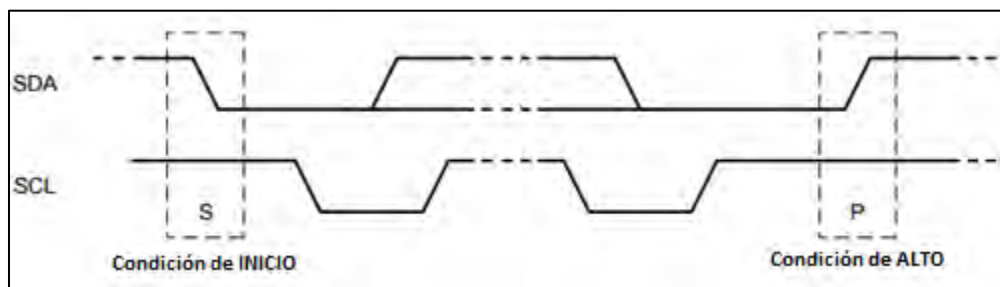
Las principales características del bus I2C son las siguientes:

- Requiere de un dispositivo maestro (microcontrolador) y uno o más dispositivos esclavos.
- Está formado por dos líneas SDA (Serial Data) y SCL (Serial Clock) siendo una forma de comunicación halfduplex.
- Cada dispositivo tiene una dirección única.
- Es un bus bidireccional de 8 bits.
- Trabaja a velocidades desde los 100kbps (estándar), 400kbps (modo rápido) y hasta 3.4 Mbps (alta velocidad).
- Las líneas de conexión necesitan de una resistencia de pull-up para polarizarse y permanecer estables.

Los datos en la línea SDA únicamente son válidos cuando la línea SCL mantiene una frecuencia controlada por el maestro.

La secuencia de comunicación básica siempre es iniciada y terminada por el maestro mediante condiciones de INICIO y ALTO así como es el encargado de indicar el tipo de transmisión de datos que se hará mediante un bit R/W (lectura o escritura).

Las condiciones de INICIO y ALTO se muestran en la siguiente figura. Como se puede observar la condición de INICIO se presenta cuando en la línea SDA se encuentra en un nivel bajo y SCL está en alto. En cambio la condición de ALTO ocurre cuando SCL está en alto justo antes de SDA.



*Figura B.1 Condiciones de INICIO Y ALTO.*

El protocolo I2C transmite los datos a partir del bit más significativo y cada mensaje es seguido por un bit de reconocimiento (Acknowledge) en modo receptor. El primer byte después de la condición de START siempre consiste de 8 bits que comprenden 7 bits de direccionamiento y un bit para lectura o escritura (R/W) u ocho bits de datos.

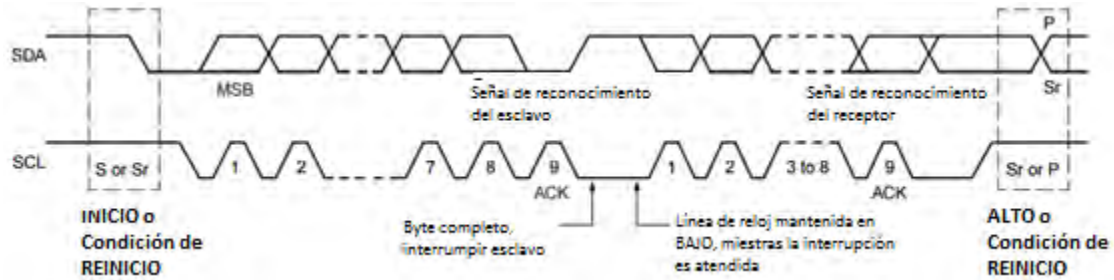


Figura B.2 Diagrama de tiempos I2C.

El bus SPI es un estándar de comunicaciones desarrollado por Motorola, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos siendo del tipo full duplex, síncrono de tipo serial. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier electrónica digital que acepte un flujo de bits serie regulado por un reloj.

Este tipo de bus necesita de un dispositivo maestro y uno o más esclavos por lo que requiere de 4 líneas de conexión:

**MISO (Master Input Slave Output):** esta línea envía los datos desde el esclavo hasta el maestro. Cuando el dispositivo con el que se desea la comunicación no está habilitado permanece en alta impedancia para evitar interferencias.

**MOSI (Master Output Slave Input):** envía los datos del maestro al esclavo. El circuito maestro pone la línea MOSI medio ciclo antes del final del flanco de alta impedancia para evitar interferencias.

**SCLK (Serial Clock):** el reloj con el que los datos son sincronizados en las líneas anteriores.

**CS (Chip Select):** los esclavos son seleccionados por el dispositivo maestro a través de esta línea. Para habilitar el dispositivo se tiene que poner en bajo esta línea. Solo puede haber un dispositivo habilitado.

Las primeras tres líneas son compartidas por el maestro y todos los dispositivos esclavos, mientras que por cada dispositivo esclavo el maestro deberá contener una línea CS, con la cual podrá activar o desactivar el dispositivo con el que se desea comunicar. Los dispositivos pueden leer los datos en el flanco de subida del reloj y otros en el flanco de bajada. La escritura se realiza casi siempre en la dirección opuesta de la dirección del movimiento del reloj.

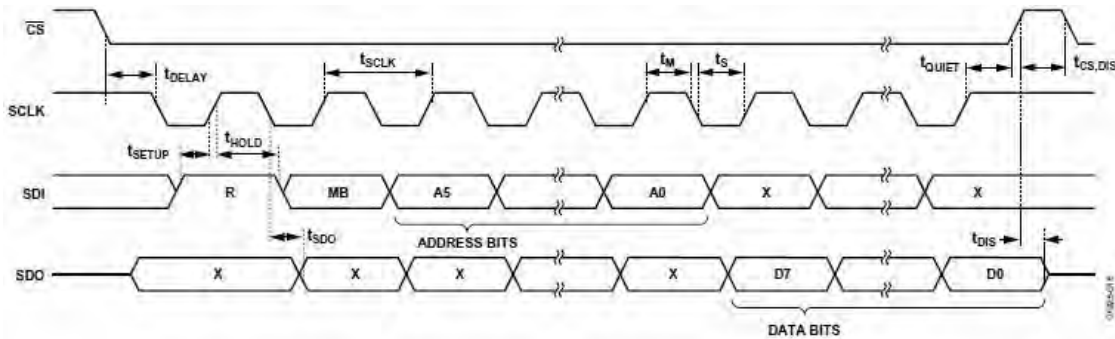


Figura C.1 Diagrama de tiempos en el protocolo SPI



# Apéndice D

# Esquemáticos del proyecto

