



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS

“Implementación del prototipo del sistema de adquisición de datos para plataforma satelital utilizando Raspberry Pi”

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN TELECOMUNICACIONES

PRESENTA:

HAZAE MIGUEL MOCTEZUMA FLORES

DIRECTOR DE TESIS
Dr. Carlos Romo Fuentes

CIUDAD UNIVERSITARIA, Abril 2015



Los agradecimientos, desde lo imprescindible, lo legendario hasta lo fantástico.

Agradezco a mi mamá Isabel, por darme todo, más que la que más, por ser una dama estoica en cualquier situación, por enseñarme las lecciones más importantes que podría aprender, por darme retos y sueños a realizar.

A mi hermana Eli, por ser mi axiomático apoyo en la vida; por convertirse en mi más sincero cómplice, porque cada vez que volteo a verla deseo ser como ella cuando sea grande.

A mis tíos, Miguel, por ser un ejemplo de superación, perseverancia y constancia a seguir, también por apoyarme a no rendirme. A Lilia, por su comprensión y por motivarme siempre.

A mis primos Darío y Alexia por los maratones de terror. Sin el lazo de sangre, estoy seguro, aun seríamos amigos.

A Isa y Lía, por su tecnología y amor, por hacerme mejor persona. Yo mataré monstruos por ustedes.

A Liana de Winter, por aquellos momentos de catarsis y las crisis, por mirar lagartos conmigo y matar vampiros, y por los conciertos irreversibles.

A Pita, por ser mi compañera indudable ante el caos, en noches de desvelo y ante los procesos aleatorios

A Solezazo, por las largas caminatas, las carcajadas imparables, los consejos y el apocalipsis zombie.

A Fany, por las historias secretas y la búsqueda mítica, por le nom de guerre y the bittersweet distractor. A todos mis camaradas de aventuras y de rock & roll, los tullidos, la machimora, Fabricio, Saúl, Luis, Ángel, José. Y a Bulma, por la estrella fugaz.

A Ivette, porque en este y en todos los mundos paralelos, mi proyecto favorito es: nosotros.

A mi asesor, el Doctor Carlos Romo, por ser un ingeniero fuera de lo común, por ayudarme constantemente, por su paciencia y ser mi guía cuando las cosas se hacían complicadas. Al Dr. Alberto Ramírez por su capacidad de transmitir conocimiento y ser un gran mentor. Al Dr. Ferrer por sentarse a platicar conmigo. A los miembros del CAT por apoyarme en mis metas. Al Ing. José Antonio Sandoval de Red Uno, quien me guio en mi primer aventura en el mundo laboral.

A la UNAM y a la Facultad de Ingeniería, por ser uno de los lugares más maravillosos, ahí he pasado los mejores años de mi vida.

A todos los autores, realizadores y artífices que me han inspirado y han cambiado mi vida. Y por supuesto, a todos los cronopios y seres de media noche.

Y para una buena historia, un buen final..

En letra capital, agradezco a la persona más fuerte y valiente que conozca, Margarita Moctezuma Flores, por soportarme en las malas y nunca rendirse, por siempre creer y tener esperanza en mí. Gracias.

HAM

Dedico esta página a las parábolas con excéntrica mayor a uno, a las líneas paralelas que se cruzan y a los matemáticos empedernidos.

Esta página fue intencionalmente dejada en blanco. Bueno, casi. Y sí, también quiero que paren la tala innecesaria de árboles.

*Not explaining science seems to me perverse.
When you're in love, you want to tell the world*

Carl Sagan

Engineers turn dreams into reality

Hayao Miyazaki



Índice General

Índice de tablas	iii
Índice de figuras	iii
Acrónimos	v
Prefacio.....	viii
Introducción.....	x
Antecedentes	xi
Justificación	xii
Capítulo 1. Propuesta	1
1.1. Definición del problema.....	2
1.2. Objetivos	3
1.3. Hipótesis.....	3
1.4. Metodología	4
1.5. Alcances del trabajo.....	10
Capítulo 2. Diseño general del sistema	11
2.1. Computadora central	13
2.2. Sistema de comunicación satelital	15
2.3. Sistema de comunicación inalámbrica	20
2.4. Sistema de comunicación Ethernet.....	22
2.5. Sistema de sensores y adquisición de datos	23
2.6. Sistema de control y monitoreo.....	24
Capítulo 3. Características de los sistemas	27
3.1. Computadora central	28
3.1.1. Raspberry Pi.....	28
3.1.2. Hardware y software utilizado.....	45
3.2. Sistema de comunicación satelital	46
3.2.1. Satélites. ¿Qué son y para qué?	46
3.2.2. Ambiente espacial.....	66
3.2.4. Sistema SCADA.....	70
3.2.5. Aplicación y servicio de las constelaciones satelitales	72
3.2.6. Modem Satelital	76



3.3. Sistema de comunicación inalámbrica y adquisición de datos.....	79
3.3.1. Comunicaciones alámbricas e inalámbricas en la actualidad	79
3.3.2. Redes de sensores.....	80
3.4. Sistema de comunicación Ethernet.....	86
3.4.1. Diferencia entre red inalámbrica y red alámbrica Ethernet	87
3.4.2. Diseño a utilizar	88
3.4.3. Almacenamiento de información.....	88
3.5. Sistema de control y monitoreo.....	89
3.5.1. Adquisición de información.....	89
3.5.2. Procesamiento de información	90
3.5.3. Subsistema de potencia.....	97
Capítulo 4. Implementación del sistema.....	98
4.1. Pruebas individuales de los sistemas.....	100
4.2. Integración de los sistemas	110
Se realizaron pruebas donde todos los subsistemas se encontraban en operación.....	110
4.3. Validación del prototipo	110
Capítulo 5. Resultados y conclusiones	118
Capítulo 6. Trabajo a futuro	122
Apéndices	124
Bibliografía.....	144
Epilogo.....	149

Índice de tablas

Tabla 2. 1 Comparación entre plataformas de desarrollo de alta gama.....	14
Tabla 3.1 Diferentes controladores y puertos del chip Broadcom.....	33
Tabla 3.2 Características de los principales protocolos de comunicación.....	44
Tabla 3.3 Listado de interfaces a utilizar.....	45
Tabla 3.4 Eventos históricamente importantes en el desarrollo de satélites de comunicación...	49
Tabla 3.5 Principales sistemas LEO.....	57
Tabla 3.6 Características de orbitas de satélites artificiales.....	58
Tabla 3.7 Bandas usadas en comunicaciones con satélites.....	59
Tabla 3.8 Tecnologías utilizadas en constelaciones satelitales.....	65
Tabla 3.9 Uso de SCADA en redes satelitales.....	72
Tabla 3.20 Trama de datos de módulos inalámbricos.....	90
Tabla 3.11 Síntesis de consumo de energía	97
Tabla 4.3 Pruebas con sensores analógicos.....	101
Tabla 4.2 Resumen de pruebas sobre la red inalámbrica de sensores.....	113

Índice de figuras

Figura 1.1 Modelo Vee	4
Figura 1.2 Modelo general.....	5
Figura 1.3 Modelo conceptual del prototipo	5
Figura 1.4 Modelo conceptual de computadora central.....	6
Figura 1.5 Diagrama de evaluación de pruebas.....	8
Figura 2.1 Sistema básico de comunicación	12
Figura 2.2 Modelo de sistema de comunicación	12
Figura 2.3 Raspberry Pi	15
Figura 2.4 Computadora central y sus conexiones entre interfaces.	15
Figura 2.5 Esquema de conexión satelital	17
Figura 2.6 Diagrama de conexión serial entre Raspberry Pi y modem satelital	18
Figura 2.7 Esquema de conexión del sistema de comunicación satelital.....	18
Figura 2.8 Esquema de simulación del modem satelital	19
Figura 2.9 Diagrama de conexión serial entre interfaces Arduino y Raspberry Pi vía puertos UART	19
Figura 2.10 Diagrama de conexión serial entre interfaces Arduino y Raspberry Pi vía puertos UART	19
Figura 2.11 Esquema de conexión entre modulo inalámbrico y computadora central	20
Figura 2.12 Diagrama de conexión entre modulo inalámbrico y computadora central vía UART20	21
Figura 2.13 Diagrama de conexión entre modulo inalámbrico y computadora central vía I2C	21
Figura 2.14 Esquema de la red inalámbrica de sensores conectada al Raspberry Pi.....	21
Figura 2.15 Esquema de conexión a red de datos	22
Figura 2.16 Esquema del sistema de sensores.....	23
Figura 2.17 Diagrama de conexión entre la computadora central y dispositivo de expansión de periféricos	24
Figura 2.18 Diagrama general de conexión de interfaces	25
Figura 2.19 Diagrama de flujo del funcionamiento del prototipo.....	26
Figura 3.1 Diagrama de puertos del Raspberry Pi	29
Figura 3.2 Conectores GPIO	32
Figura 3.3 Bus de 8-bits controlado por un reloj, transmite cada byte por cada pulso de reloj.....	35



Figura 3.4 Interface serial, transmitiendo un solo bit cada pulso de reloj	35
Figura 3.5 Tamaño de bits en una trama de datos	36
Figura 3.6 Lecturas de diferencias de potencia en TTL	37
Figura 3.7 Lecturas de diferencias de potencia en el estándar RS-232	37
Figura 3.8 Circuitos de enlace entre interfaces RS-232 y TTL	38
Figura 3.9 Comparación de niveles lógicos en TTL al pasar a RS-232	38
Figura 3.10 Interfaces del estándar UART	39
Figura 3.11 Relación maestro-esclavo	40
Figura 3.12 Selección del esclavo	41
Figura 3.13 Sistema de 3 esclavos conectados	41
Figura 3.14 Esclavos conectados en serie	42
Figura 3.15 Tramas de dirección y de datos en I2C	43
Figura 3.16 Interfaces externas de los satélites	50
Figura 3.17 Utilización de servicios FSS.....	51
Figura 3.18 Uso de enlaces intersatelitales	52
Figura 3.19 Esquema básico de comunicación en MSS.....	52
Figura 3.20 Segmento espacial y terrestre	53
Figura 3.21 Red de servicios satelitales	54
Figura 3.22 Diferentes tipos de orbitas satelitales terrestres.....	55
Figura 3.23 a) Relación de altitud contra periodo de los satélites en orbitas LEO, MEO y GEO. b) Relación en órbita MEO. c) Misma relación para orbita LEO	58
Figura 3.24 Espectro electromagnético.....	59
Figura 3.25 Representación temporal y fasorial para una señal BPSK.....	61
Figura 3.26 Representación temporal y fasorial para una señal QPSK	61
Figura 3.27 Representación fasorial de señales 16-QAM y 16 APSK	62
Figura 3.28 Representación de FDMA	63
Figura 3.29 Representación de TDMA	63
Figura 3.30 Representación de CDMA	64
Figura 3.31 Desechos espaciales en órbita terrestre baja	67
Figura 3.32 Canales disponibles para Wi-Fi	69
Figura 3.33 Red satelital Globalstar.....	69
Figura 3.34 Constelación Globalstar	70
Figura 3.35 Diagrama de sistema SCADA	71
Figura 3.36 Aplicación de las redes satelitales con VSAT	73
Figura 3.37 Constelación de satélites LEO de Iridium	75
Figura 3.38a) Satélite con capacidad de un solo haz b) satélite con capacidad para multihaz	75
Figura 3.39 Probabilidades de error al transmitir paquetes de datos.....	78
Figura 3.40 Tipos de nodos en las redes inalámbricas de sensores.....	82
Figura 3.41 Red con un coordinador y tres nodos sensores	85
Figura 3.42 Redes de un solo salto y multi salto	85
Figura 3.43 Modelo de referencia OSI	86
Figura 3.44 Protocolos de TCP/IP.....	87
Figura 3.45 Red de datos simulada	88
Figura 3.46 Trama de PPP	91
Figura 3.47 Trama de protocolo de capa de enlace.....	92
Figura 3.48 Estructura de paquetes en IPv4	92
Figura 3.49 Diagrama de flujo para la simulación de PPP	95
Figura 3.50 Paquete para PING simulado.....	96



Figura 3.51 Sistema completo.....	96
Figura 4.1 a) Sistemas a probar. b) Protocolos de referencia.....	99
Figura 4.2 Estructura para la adquisición datos en la red alámbrica.....	101
Figura 4.3 Prototipo del nodo sensor de temperatura.....	102
Figura 4.4 Prototipo del nodo sensor de temperatura.....	102
Figura 4.5 Prototipo del nodo sensor de movimiento.....	103
Figura 4.6 PCB de sensor de temperatura.....	103
Figura 4.7 PCB de sensor de humedad.....	104
Figura 4.8 Prueba a 1 [m] (izquierda) y prueba a 3 [m] (derecha).....	104
Figura 4.9 Prueba a 0.5 [m].....	105
Figura 4.10 Red inalámbrica de sensores (izquierda) y nodo de temperatura (derecha).....	105
Figura 4.11 Conexiones del nodo coordinador.....	106
Figura 4.12 a) Conexión Ethernet entre red LAN y el Raspberry Pi (izquierda) transmisión de información entre ambos (derecha).....	106
Figura 4.13 Prueba de PING para la red de datos del CAT.....	107
Figura 4.14 Modulo GPS.....	107
Figura 4.15 Diagrama de conexión entre Arduino y computadora central.....	108
Figura 4.16 Circuito usado por la computadora central.....	108
Figura 4.17 Diagrama de proceso para comunicación satelital.....	109
Figura 4.18 Modem satelital.....	109
Figura 4.19 Prototipo de pruebas completo.....	110
Figura 4.20 Software X-CTU.....	111
Figura 4.21 Trama API enviada.....	111
Figura 4.22 Trama de datos recibida.....	111
Figura 4.23 Trama de datos ya procesada con la información accesible.....	112
Figura 4.24 Tramas de información recibidas de los nodos.....	112
Figura 4.25 Prueba de PING exitosa.....	113
Figura 4.26 Activación de VPN para LAN.....	114
Figura 4.27 Datos recibidos por el GPS.....	114
Figura 4.28 Ubicación en el D.F.....	114
Figura 4.29 Ubicación en Juriquilla.....	115
Figura 4.30 Recepción de los paquetes exitosa.....	115
Figura 4.31 Recepción de protocolos de internet y de capa de enlace.....	116
Figura 4.32 Graficas en computadora central de los datos adquiridos.....	116

Acrónimos

ARM Advanced RISC Machine



ASK	Amplitud-shift keying
BMC	Business Management Centre
BSS	Business Support System
BW	Bandwidth
CAT	Centro de Alta Tecnologia
CDMA	Code Division Multiple Access
COTS	Comercial Off-The-Shelf
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DCE	Data Communications Equipment
DSP	Digital Singal Processor
DTE	Data Terminal Equipment
FDMA	Frequency Division Multiple Access
FI	Facultad de Ingenieria
FLOSS	Free/Libre and Open Spurce Software
FSK	Frequency-shift keying
FSS	Fixed-Satellite Service
GEO	Geosynchronous Satellite
GES	Satellite-Ground Earth Station
GPIO	General-purpose inpit-output
HLR	Home Location Register
HMI	Human-Machine Interface
ICD	Interface Control Document
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISL	Inter Satellite Link
ISM	Industrial, Scientific and Medical
ISO	International Organizarion for Standardization
ITU	International Telecommunication Union
LAN	Local Area Network
LEO	Low Earth Orbit
MAN	Metropolitan Area Network
MEO	Medium Earth Orbit
MSC	Mobile Switching Centre
MSS	Mobile Satellite Services
MTU	Master Terminal Unit
M-VSAT	Mobile Very Small Aperture
NASA	National Aeronautics and Space Administration
NCC	Network Control Centre
NGEO	Non-geostationary satellite
OBP	On-board processing
OBS	On-board switching
OSI	Open System Interconnection
PPP	Point-to-Point Protocol
RTU	Remote Terminal Unit



SCADA	Supervisory Control And Data Adquisition
SCC	Satellite Control Centre
SPI	Serial Peripheral Interface
TCP	Tranmission Control Protocol
TDMA	Time Division Multiple Access
TT&C	Telemetry, tracking and command
TTL	Transistor-transistor logic
UART	Universal Asynchrponus Receiver-Transmitter
UES	Satellite User Earth Station
VLR	Visitor Location Register
VSAT	Very Small Aperture
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network



Prefacio

La elaboración de la presente tesis surgió del interés personal de profundizar en las implicaciones tecnológicas de la industria aeroespacial en México. Sin embargo, los objetivos iniciales tienden a perderse constantemente y considero de vital importancia dar a conocer los inicios, la motivación principal. Todo empezó con un refrigerador.

En el año de 1995 llegue a la Ciudad de México, no por primera vez, pero sí para permanecer ahí durante muchos, muchos años. Mi madre me llevo a la casa de una tía. Es muy probable que hubiera estado en aquella casa antes pero el primer recuerdo que tengo de ella fue hasta ese momento. Aquel día, una pequeña casa, la cual, desde entonces y hasta la fecha, ha estado repleta de una multitud de cosas y artefactos. Para los nuevos visitantes podrían decir que parece un museo. Todo era mágico, adornos por doquier, recuerdos de todo el mundo, lucia espectacular y claro, mi joven yo, un niño glotón y curioso tuvo como primer parada el refrigerador ubicado en el límite de la sala-comedor y la cocina. Ahí fue cuando obtuve el recuerdo de la primer foto que en verdad me gusto.

La imagen de una nave espacial (o eso creí en el momento) estaba pegada en la puerta del refrigerador. Para mí los viajes espaciales eran cosas de las cientos de películas para niños que sucedieron antes de ese momento. Algo que no podía comprender y que se me hacía tan común. Pero esa foto tenía algo más, no parecía salida de mis caricaturas o de los libros ilustrados que me leían, esa más real. Pregunte a mi madre con harta curiosidad sobre la procedencia y significado de la foto. Me entere que la gente de la realidad sí viajaba al espacio, sí llegaba a la luna, aun no viajaban a velocidad de la luz ni tenían sorprendente batallas en cruceros interestelares, pero algo es algo. También supe que la dichosa nave era un transbordador espacial y que si trabajaba duro, existía la remota posibilidad de algún día poder estar como aquel pequeño hombre de la foto. Y desde entonces pensé con el espacio infinidad de veces.

Fue así como mi camino se posiciono poco a poco en esta dirección. Y llegue a la Facultad de Ingeniería de la UNAM, debo confesar que con muchas deficiencias. Por fortuna encontré personas que me guiaron en la verdadera labor tecnológica y científica que se realiza en el país. Los ingenieros del anexo decían que si caíamos teníamos que levantarnos y sacudirnos el polvo, y vaya que se cumplió varias veces. De los divulgadores de ciencia aprendí que el verdadero viaje comienza aquí (posicione su dedo índice derecho en su sien, haciendo referencia al cerebro). Ya en el "principal" conocí las verdaderas historias de los viajes espaciales en México, veía imágenes y fotos ya no solo como espectador, aún más maravillado veía como se transformaban en dígitos, en señales electrónicas. Llegue a comprender la labor artesanal de los creadores de satélites y lo que implica posicionarse en el firmamento.

También conocí a buenos compañeros que con el tiempo se convirtieron en equipo. Y solo o acompañado, se intentó concebir nueva tecnología. Se planteó una ciencia para atravesar ventanas, una teoría de síntesis musical para conciertos a grandes dB, labores interdisciplinarias y un etcétera que solo aquellos involucrados conocen.

El refrigerador se ha convertido en esencial para la vida de las personas, sin dicha innovación tecnológica sería impensable concebir el mundo actual. Siempre en los momentos de necesidad y aun ahora, cuando hace falta a media noche se plantea en comprar uno, barato pues la economía no da para más. Puede que de esa forma la ingeniería se haya convertido en un esencial en mi vida. Un constante con la cual se podría vivir, pero qué clase de vida sería. Un Dr. dio una plática al inicio de la carrera, exhortando a los alumnos a vencer las probabilidades, todo indicaba que no lo lograríamos, que el titularse era un casi imposible. Él dijo que se podía vencer a Skynet y salvar el futuro y presente. Lo creí, así como también se nos demandó el pensar críticamente. Ahora tengo dos sensaciones que extrañamente están juntas: el orgullo y la gratitud. Con todo lo que representa esta tesis y con los que ayudaron a realizarla pero más de aquel día, en el que vi la fotografía y el refrigerador y mi madre, sin saber nada ingeniería, me dijo lo que podía hacer.





Introducción

El ser humano siempre ha tenido la necesidad de comunicarse, el simple hecho de tener conexiones neuronales adecuadas, da paso a la acción de pensar, gracias a eso se puede producir las ideas, y de ahí pasar a querer comentarlas, querer comunicar lo que se tiene en la mente, por lo tanto se tiene la base para establecer un sistema de comunicación. Esto puede ser tan sencillo como el hablar y tan complicado que requiera a la ciencia para su entendimiento. Poder transmitir algo siempre ha sido un imperativo en la vida cotidiana. Las formas de comunicación han cambiado constantemente y lo seguirán haciendo. A este cambio constante se le suele llamar evolución, ya sea de manera tecnológica o científica.

En el presente trabajo se da una visión general de algunos de los principales tipos de medios de comunicación en la actualidad como son: la transmisión de datos mediante redes cableadas, que concierne a las redes personales, redes de área local y hasta redes mundiales de información con uso de fibra óptica, microondas o radiofrecuencia. Dichas redes dieron origen al internet, catalogado como el mejor invento del siglo y no es para menos, desde cualquier punto del mundo puedes acceder a miles de recursos de manera remota, esta utilidad es primordial en el presente proyecto y se retomara en muchas ocasiones en los capítulos posteriores.

También se hará mención de un medio importante que lucha por ser el sucesor de las redes con cables, las de comunicaciones inalámbricas. ¿Quién en nuestros días no ha utilizado el bluetooth, mandado SMS desde su teléfono celular, utilizado un GPS para evitar perderse en alguna carretera del país? Todos esos ejemplos son medios inalámbricos de comunicación. En este caso se abordaran las redes inalámbricas que han tenido un gran empuje en términos de investigación durante las últimas décadas.

Las comunicaciones satelitales dieron mucho de qué hablar en su momento, para ser un poco más preciso, en la segunda mitad del siglo pasado. La razón principal fue todo lo que implicaban, viajes al espacio, ser astronauta, llegar más allá de lo que cualquier hombre ha llegado jamás, etc. Hoy en día pareciera que se encuentra en olvido general por la gente, aun cuando son de vital importancia en la vida diaria. En materia de enseñanza, los satélites en México están muy poco difundidos y podría pensarse que esta industria apenas están en pañales dentro del país, afortunadamente esto está cambiando. Cada vez hay más interesados en estos temas. Las universidades del mundo también han dedicado muchos esfuerzos a la promoción de dicha tecnología, un ejemplo muy claro es el protocolo CubeSat, siendo básicamente un nano satélite en forma de cubo. En México ya existen varios trabajando en más prototipos de este tipo de satélites y con diferentes propósitos.

Se ha llegado al punto que la comunicación no solo es entre humanos, ahora las maquinas también interactúan entre ellas. Vehículos en el espacio “hablan” con



máquinas aquí en tierra y transmiten a otras máquinas, como se menciona, a veces de manera inalámbrica, por el aire, pareciera magia. Todo esto podría bien ser un comentario proveniente de alguna buena novela de ciencia ficción, pero la innovación tecnológica es una realidad.

El CAT (Centro de Alta Tecnología) de la Facultad de Ingeniería de la UNAM es un instituto multidisciplinario, su principal objetivo es el desarrollo de la innovación tecnológica industrial para el crecimiento del país, para lo cual, vincula a los alumnos con algunas de las principales industrias. Dentro del CAT, una de las áreas de trabajo es la industria aeroespacial, se han desarrollado trabajos para varias misiones de diferentes tipos de satélites, entre ellas el Quetzal, Cóndor, Ulises 2.0. Siempre buscando experimentar y seguir implementando tecnología.

Con el apoyo del CAT y su equipo de trabajo, el presente trabajo muestra una síntesis de un sistema de comunicación que contempla varias formas de transmitir el mensaje de interés, dicho mensaje son datos, cuyo propósito será explicado más adelante en el texto. También se describen como se desarrolló teórica y prácticamente el sistema hasta llegar a ser un prototipo tecnológico listo para implantarse y ser probado en el mundo real.

Para el primer capítulo se presenta lo relacionado con la propuesta general del sistema, incluyendo el objetivo principal y particulares. La hipótesis del experimento y la metodología que se siguió para su construcción.

El estado de arte concerniente a la investigación se describe en el capítulo dos, y se retoma en el capítulo tres que también proporciona la parte teórica necesaria. A lo largo de los capítulos dos y tres se describen grosso modo las comunicaciones alámbricas, algunos tipos de comunicaciones inalámbricas que utilizan la atmósfera como medio para enviar mensajes. También se hará hincapié en las herramientas utilizadas para poder crear el prototipo, tarjetas de desarrollo, módulos electrónicos y mini computadoras.

Las pruebas del diseño, integración del sistema, validación y verificación del prototipo se presenta en el capítulo cuatro, seguido de las conclusiones y por supuesto, lo que queda pendiente por hacer en el último capítulo.

Antecedentes

El Centro de Alta Tecnología como instituto de vinculación entre el área industrial y el área académica creó un proyecto de colaboración en conjunto con la empresa Globalstar de tecnología espacial, que provee servicios de comunicación satelital en México, con el objeto de crear un sistema de telemetría en tierra para después considerarlo como sistema de comunicación en una plataforma de nano satélite. Como primera propuesta se realizó el prototipo a nivel terrestre, dentro del



sistema de telemetría se consideró diseñar un sistema fiable para poder adquirir y procesar datos paramétricos, como por ejemplo: temperatura, presión, humedad, etc. Se presentó la oportunidad dar solución a ambas necesidades en conjunto mediante la utilización de la tarjeta de desarrollo Raspberry Pi.

Hubo un acercamiento con la compañía Globalstar bajo el marco del proyecto donde se realizaron pruebas de enlace satelital con el modem satelital hacia la constelación en órbita de Globalstar. También se probaron diferentes antenas pasivas para el enlace RF entre el modem y el satélite, se probó el modem satelital en conjunto con varios celulares satelitales, con todos estos elementos se empezó a desarrollar la propuesta de subsistemas que requería el prototipo para su correcto funcionamiento.

Esta experiencia, me permitió conocer un poco más del estado de arte en las industrias modernas que tienden a utilizar mucho la comunicación de datos, con la finalidad de que estos interactúen con diferentes procesos, sistemas e instalaciones. Todo para que la industria sea más competitiva y así poder cumplir con unos de los requerimientos pilares: ser eficiente a un nivel de competitividad exigida en los procesos productivos actuales.

Se continuó con la labor de experimentar formas eficientes de adquisición de datos y también de transmitir dicha información con el fin de ser aplicadas en la industria, reduciendo costos en la producción de tecnología. Esta premisa sentó las bases para el presente proyecto.

Justificación

La importancia del presente proyecto radica en la selección de los materiales que tengan todas las capacidades contenidas dentro de un elemento de procesamiento. Al hablar de ciertas industrias, como son la aeroespacial, aeronáutica o militar como ejemplos, es necesario hablar de estándares y protocolos por lo que están regidos.

En general, toda la ingeniería está homologada por ciertas reglas, pero normalmente suelen ser más estrictas y complicadas para las industrias aeroespacial, aeronáutica o militar. Por esa misma razón, la tecnología fabricada en esas áreas suele ser muy robusta, como en el caso de la construcción de satélites, en el que su estructura y elementos en ella pasan por pruebas de resistencia, de temperatura o de compatibilidad electromagnética. También sus métodos de construcción tienden a mantenerse en secreto por años. Obviamente si se tuviera algo especial y único, buscaría preservarse en anonimato, pero gracias a la naturaleza inquisitiva del hombre se han podido romper estas fronteras.

En una noble labor de enseñanza, existen iniciativas que promueven el uso abierto de la información para crear nueva tecnología, ya sea en software o en hardware. La construcción de nano satélites, sistemas SCADA, maquinaria de



calidad industrial con tarjetas de prototipos que se producen en masas y que utilizan código abierto, ha creado todo un mundo lleno de nuevas oportunidades.

Ahora aprender programación es tan indispensable como saber leer, uno aprende a leer y luego lee para aprender. Debe ser más que obvio que al programar sucede el mismo, más importante aún es la aplicación. Enfocarse en diseños caseros que se aplican a un nivel industrial es el verdadero reto, ya no solo de la ingeniería, si no de cualquier entusiasta de la electrónica, de la programación, de la mecánica, robótica, neurociencia, medicina. Se habla ya de telemedicina usando Arduino, robots en líneas de producción programados en SoC (System-on-a-Chip), tarjetas Texas que leen patrones neuronales, estamos en el siglo donde la tecnología básica y hecha en masa sirve para tareas específicas y de alta especialización. Cuando en otras décadas no se podía pensar en esto, que solo los expertos podrían lograrlo, hoy se sabe que no cualquiera puede hacerlo, he ahí la meta. Poder realizar un prototipo sofisticado de ingeniería con materiales de bajo costo para la industria.

El análisis realizado en este trabajo se enfoca precisamente en la utilización de sistemas integrados que logren satisfacer todas las necesidades de interfaces y procesamiento de información para realizar sistemas complejos con las capacidades de reconfiguración a través de sistemas inalámbricos de comunicación.

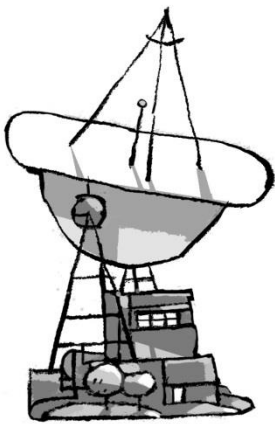
Capítulo 1. Propuesta

Beware of the man who works hard to learn something, learns it, and finds himself no wiser than before

Kurt Vonnegut

You can't try to do things. You simply must do things.

Ray Bradbury





1.1 Definición del problema

Actualmente los sistemas de adquisición de datos basados en elementos COTS (Commercial off-the-shelf) se encuentran en crecimiento y desarrollo constante. El hardware utilizado por dichos sistemas carece de puertos para anexas más dispositivos periféricos y también carece de una interfaz fácil de utilizar, para los cuales es necesario equiparlos con varios módulos para su correcto funcionamiento, lo cual implica un diseño de mayores dimensiones físicas, que requiera más espacio al momento de integrarlo.

El Raspberry Pi es una tarjeta de desarrollo que cuenta con la adaptabilidad para integrar más dispositivos periféricos, provee un lenguaje propio para su programación (Python), interfaz gráfica amigable (Raspbian) y un procesador con capacidad superior a los microcontroladores en comparación con su velocidad de procesamiento. Con este tipo de equipo se ahorra espacio y recursos ya que cuenta con las especificaciones de procesamiento y de periféricos (teclado, mouse, pantalla de visualización) necesarias para funcionar como una computadora de bajo costo y que se puede adaptar a las necesidades de proyectos de electrónica.

Mediante la utilización del Raspberry Pi se solucionaron problemas financieros y de espacio. Al final se obtuvo un prototipo económico, que resolvió el problema de comunicación por varias vías y la obtención de datos de manera eficaz en comparación a la velocidad obtención y procesamiento de datos, espacio necesario para la convergencia de los subsistemas.

En la actualidad las comunicaciones satelitales siguen vigentes y son una gran alternativa para poder resolver problemas precisos usando sus ventajas únicas; Una de ellas es la cobertura en casi cualquier zona del planeta. Usando dicho argumento se implementó un sistema eficiente tanto en adquisición de datos como en movilidad.

Sin el sistema propuesto existirían problemas como el tener un prototipo abultado en tamaño sin capacidad para ser portable, una velocidad de procesamientos lenta y poco adecuada para un ambiente espacial o un ambiente industrial que requiera un mayor volumen de trabajo.

1.2 Objetivos

El objetivo general de la tesis es generar un prototipo capaz de adquirir datos de parámetros físicos: humedad y temperatura provenientes de cualquier ambiente, a través de sensores para poder transmitir los valores obtenidos de manera segura y eficaz por medio de una configuración de cables hacia una red LAN así como también vía una constelación satelital.

Objetivos Específicos

- Tener prototipos de los sistemas de adquisición de datos, conmutación y multiplexión, sistema de comunicación inalámbrico, sistema de comunicación vía satelital, sistema de red conmutada por paquetes y de una plataforma funcional que integre todos los prototipos mencionados.
- Encontrar los medios, información y materiales necesarios para procesar los datos obtenidos por varios subsistemas del proyecto, en un solo medio de control
- Investigación las ventajas de usar las WSN como reemplazo de los sistemas SCADA, debido a la necesidad de encontrar medios más sustentables y suficientes que llenen las expectativas técnicas.
- Tomar parte de la discusión y extender el problema a la utilización de constelaciones satelitales en sistemas WSN.
- Investigar la posible capacidad del modem satelital como punto de acceso a internet
- Utilizar un paradigma FLOSS (Free-libre / Open Source Software) y COTS para el diseño del prototipo.
- Brindar las bases sólidas que sirvan como antecedente para el desarrollo de una plataforma satelital en estándar CubeSat donde se incorpore el modem satelital como sistema de comunicación

1.3. Hipótesis

Se puede diseñar, construir y probar un prototipo el cual sea capaz de monitorear parámetros físicos como temperatura, humedad, voltaje y que envíe los datos obtenidos a través de varios modos de comunicación como: comunicación inalámbrica, comunicación satelital o bien, utilizando una red Ethernet, basado en el sistema Raspberry Pi y materiales COTS.

1.4. Metodología

El desarrollo del proyecto se basó en la metodología del modelo Vee¹ (figura 1.1), comúnmente usado en el desarrollo de software pero también aplicado en gran parte de la industria tecnología para el desarrollo de prototipos de sistemas espaciales, el cual hace énfasis en los aspectos técnicos del proyecto y el código desarrollado para su funcionamiento.

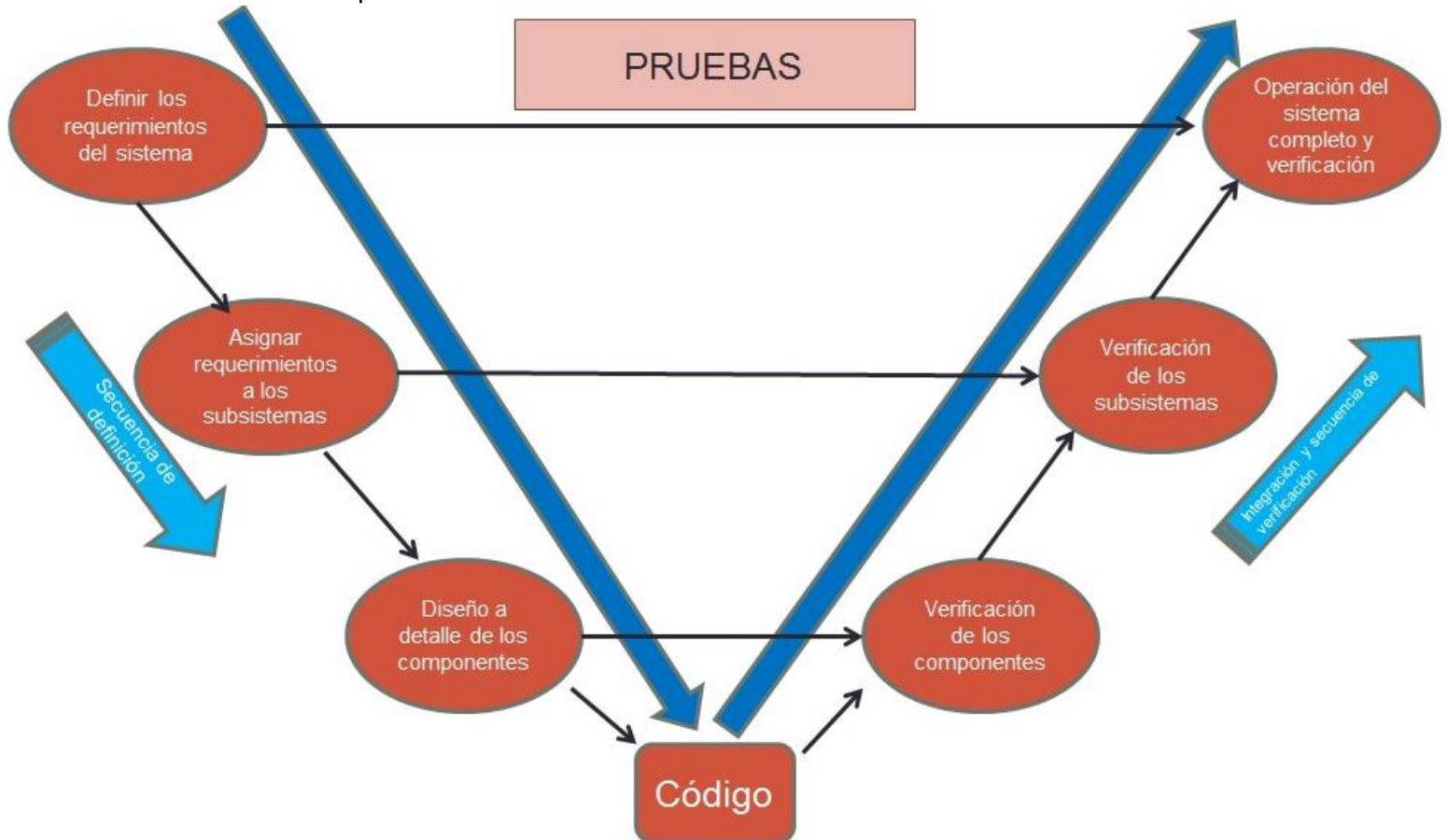


Figura 1.1 Modelo Vee [24]

El modelo Vee se conforma de los siguientes puntos a desarrollar:

- Definición de los requerimientos del sistema,
- Asignación de los requerimientos a los subsistemas
- Diseño a detalle de los componentes,
- Verificación de los componentes y subsistemas
- Operación del sistema completo

¹ Introducido por Kevin Forsberg y Hal Mozz en 1991, con modificaciones hechas por Duane W. Hyberston en *Model-Oriented Systems Engineering Science*.

Se presentan también otros puntos como son la parte de pruebas de funcionamiento de cada subsistema y sus componentes, así como de funcionalidad integral, además del desarrollo del código. A continuación se presenta una descripción de los pasos usados para la creación del prototipo.

Definición los requerimientos del sistema.

El modelo comienza con las necesidades del sistema en general. La idea que se tiene para dar solución a la problemática. Para lo cual se desarrolló una propuesta que solucionara las problemáticas presentadas, de ahí se establecieron los requerimientos generales para el proyecto: un módulo de telemetría que tendrá un sistema central de procesamiento capaz de transmitir datos de parámetros físicos (humedad, temperatura) a través de una constelación de servicio de comunicación satelital. En la figura 1.2 se muestra el modelo básico del sistema.

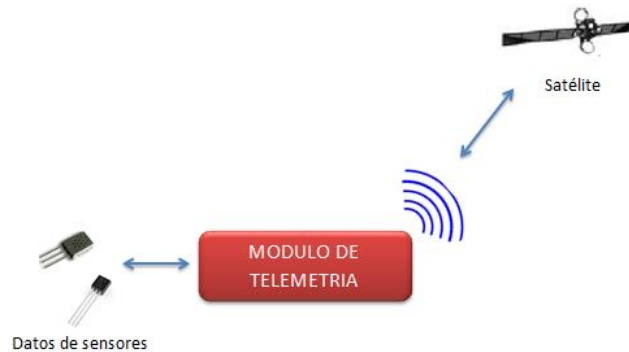


Figura 1.2 Modelo general

Asignación los requerimientos a los subsistemas.

Al tener los requerimientos generales se empieza a descomponer el sistema en subsistemas para solucionar problemas específicos.



Figura 1.3 Modelo conceptual del prototipo

En la figura 1.3 se presenta el modelo conceptual de los subsistemas dentro de la propuesta presentada. Cada requerimiento se traduce en un módulo dentro del sistema, que realizará tareas específicas para cumplir con algún requerimiento, considerando que tendrá un sistema central de procesamiento.

En el modelo conceptual se puede observar que se tienen los siguientes módulos que constituyen el sistema a desarrollar:

- Modem satelital.- Brinda el acceso de conexión hacia una constelación satelital para transmitir datos a través de su infraestructura.
- Sistema GPS.- provee de coordenadas geográficas para la ubicación exacta del prototipo.
- Sistema GMS (Telefonía celular).- permite enviar la información vía SMS a cualquier usuario con un dispositivo y servicio de telefonía celular.
- Sistema de Telemetría.- Encargado de obtener datos de parámetros físicos de algún ambiente.
- Sistema de comunicaciones VHF/UHF.- sirve como un control de operaciones de manera remota del prototipo.
- Sistema de potencia.- Encargado de regular las diferencias de potencial y corrientes de todos los dispositivos eléctricos conectados a la computadora central.
- Red de datos del CAT.- En una red de área local (LAN) que permite visualizar los datos adquiridos por el sistema de telemetría en una red con privada para garantizar seguridad en el traslado de información.
- Computadora central (Raspberry Pi)

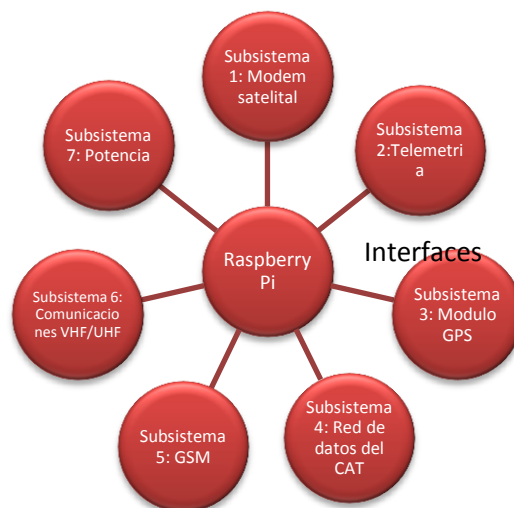


Figura 1.4 Modelo conceptual de computadora central

El modelo sistema de computadora central presenta todos los subsistema que involucra. Se encarga de conectar todo los subsistemas y hacer que converjan en un solo nodo donde la información sea procesada y transmitida a otro nodo de conexión, ya sea un nodo WAN de la constelación satelital, un nodo de una red de área local o a un nodo de red de telefonía celular.

Este trabajo se enfoca solo a las propuestas de prototipos para los siguientes módulos:

- Computadora central
- Modem satelital
- Conexión a Modulo GPS
- Sistema de telemetría
- Conexión a Red de datos del CAT

Diseño a detalle de los componentes

En esta etapa se definen las especificaciones técnicas de los subsistemas o módulos del proyecto, las cuales determinan el funcionamiento característico de cada módulo con el objetivo de que cumplan su proceso específico.

Para cumplir con las especificaciones se utilizaron componentes electrónicos, dispositivos de vanguardia tecnológica como plataformas de hardware libre, tarjetas de desarrollo, circuitos integrados, sensores, y cables de conexión además de conocimientos teóricos adquiridos durante la carrera, lo que permite hacer que este sistema de comunicaciones sea capaz de ser funcional.

Durante cada etapa de desarrollo (diseño a detalle, asignar requerimientos a los subsistemas y requerimientos de los subsistemas) dentro del modelo Vee, es necesario desarrollar pruebas para comprobar su correcto funcionamiento. Cada prueba exitosa tendrá como consecuencia la validación del componente o subsistema. Cuando llegaba un nuevo componente para instalar en el prototipo, se hacía una prueba separada para ese componente. Cuando se tuvo un subsistema funcionando, se realizó una prueba individual del subsistema.

La figura 1.5 muestra un procedimiento para la realización de pruebas e integración de módulos o componentes. Es comparable a la parte de validación e integración en el modelo Vee.

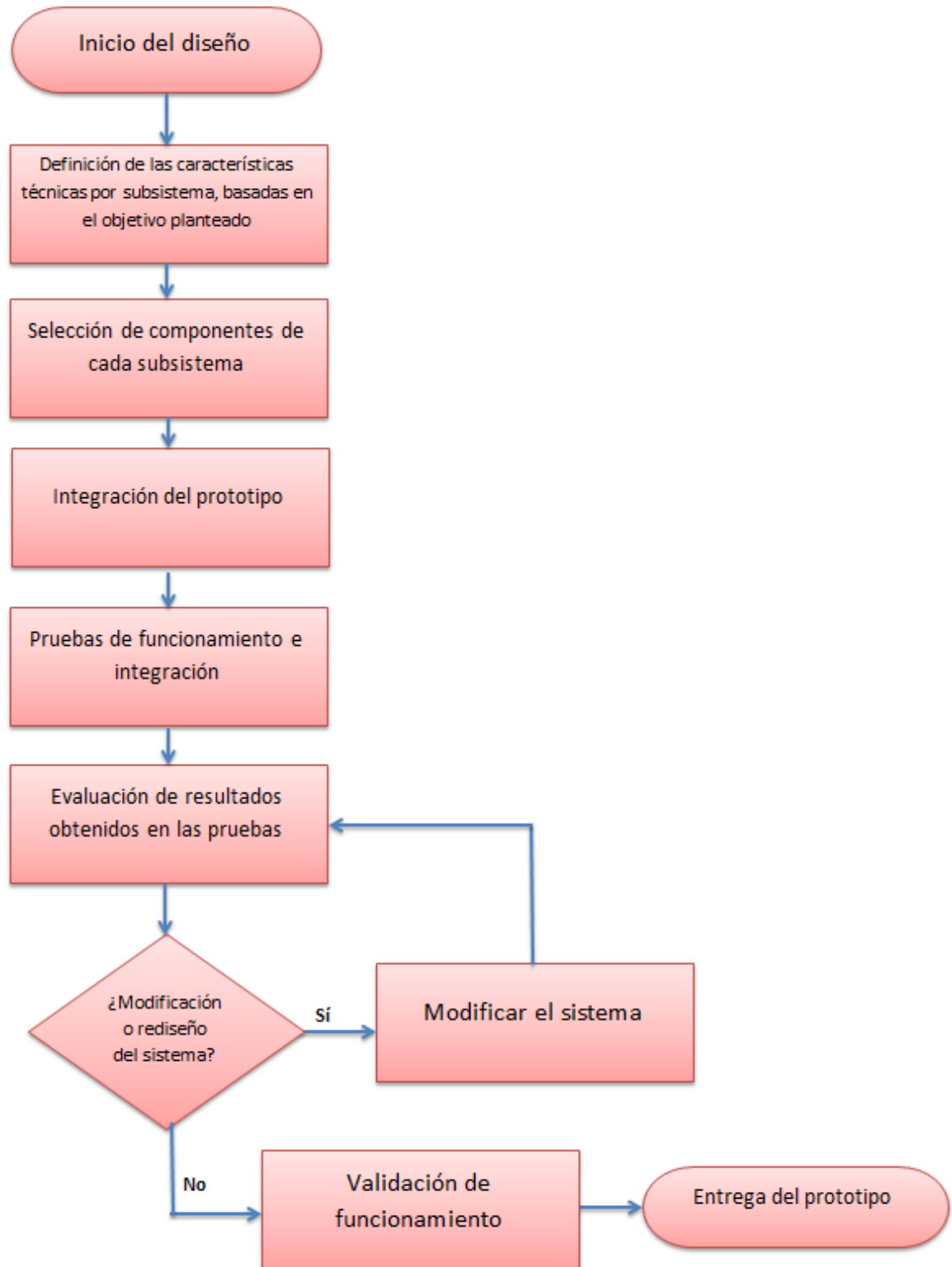


Figura 1.5 Diagrama de evaluación de pruebas

Código

El código es una parte esencial en el trabajo donde cada componente requiere sus propias líneas de código para su correcto funcionamiento, por ello es el vértice del modelo. El software desarrollado en este proyecto se elabora en lenguaje de programación que coincide en muchos casos con la programación orientada a objetos Java y C++.

Pruebas

Es de gran ayuda entender a la perfección qué es lo que hace cada componente en el sistema. Conocer cada parte utilizada ya sea elementos electrónicos, código de programación, etc. Cada prueba establece condiciones de entrada de acuerdo al resultado que se espera. Bajo premisas realizadas al inicio del proyecto como el envío de datos de manera ágil, eficiencia en la transmisión, dimensiones de prototipos, integridad de los paquetes de información, se experimentó para poder comprobarlas. Las pruebas también sirvieron para monitorear y observar el rendimiento individual y en conjunto, solo así se demostró con contundencia las capacidades de escalabilidad de la propuesta.

Verificación de los componentes y subsistemas.

Para validar los componentes y funcionalidad de los subsistemas se conectó el módulo (cada uno de los módulos se conectaron de manera individual y de manera integral) a la computadora central (se concretaron al Raspberry Pi).

En cada módulo se tiene una interface específica para intercambio de información, para lo cual se utilizaron las diferentes interfaces que tiene el RBP como Ethernet, usb, serial, etc. Se verificó que hubiera información transmitida a través de cada una de ellas de acuerdo al módulo que se conectara.

Los documentos de control de interface o ICD son la documentación que ayudará a saber como se conectan los diferentes sistemas y simplificar dicho proceso, estos documentos se presentan en el capítulo dos de esta tesis.

El ICD tiene información detallada de las entradas y salidas de un sólo sistema, la interface entre dos sistemas y el protocolo de interfaz que contenga los elementos físicos involucrados. Esto puede ser desde los niveles de voltaje de alguna conexión hasta la definición de la capa del modelo OSI que se está desarrollando en el sistema. Cabe aclarar que los ICD, sólo deben describir las interfaces usadas y no las características completas de los sistemas. Las funciones



tanto físicas como lógicas de los sistemas se presentarán en sus propios documentos de diseño (datasheets), en el caso del presente proyecto, se da una breve explicación en el capítulo dos y una más extensa en el tercero.

Otro proceso que resultó útil (y que viene en conjunto con algunos ya mencionados anteriormente) es evitar probar dos elementos juntos a la vez. La literatura de procedimientos² no recomienda hacer pruebas con más de dos elementos nuevos al mismo tiempo. Esto sirve para evitar errores futuros en el diseño y sobre todo para saber lo que cambia, e identificar una falla oportunamente.

Para finalizar la parte de diseño a detalle de los componentes, se tienen que integrar todos los módulos en una prueba conjunta. Lo mismo pasa con la parte de requerimientos de los subsistemas, se integraron y se comprobó que realizaran su función definida en las especificaciones técnicas, por lo que se validó el funcionamiento de los componentes.

Operación y verificación del sistema completo.

Se probaron los sistemas en conjunto (esto es, conectados simultáneamente al Raspberry Pi y funcionando) y se realizaron pruebas contra errores (como la saturación en la información de los sensores y repetición de datos).

Una vez finalizada esta etapa se corroboró la verificación (se comprobaron los datos enviados por cada módulo, recibidos y desplegados con el Raspberry Pi).

1.5. Alcances del trabajo

Se analizaron varias de las formas de comunicaciones electrónicas que existen para transmitir datos, aplicadas en un prototipo construido a base de COTS. Este trabajo está enfocado a integrar los siguientes módulos que se contemplan en la propuesta:

- Computadora central
- Modem satelital
- Conexión a Modulo GPS
- Sistema de telemetría
- Conexión a Red de datos del CAT

² NASA Systems Engineering Handbook

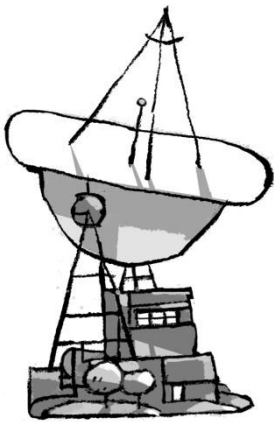
Capítulo 2. Diseño general del sistema

Any sufficiently advanced technology is indistinguishable from magic

Arthur C. Clarke

A quiere a B es, en gran medida, el producto de la autoestima de A multiplicada por la admiración que B siente por A

J. Wagensberg



Se planteó el sistema más básico de comunicación que cumpliera con el objetivo de poder transmitir datos. En la figura 2.1 se observa que dicho modelo cuenta con varios elementos. El transmisor es donde se genera el mensaje. Un mensaje es información que se tiene que transmitir al punto final del modelo. El canal de transmisión es el medio (cable, aire, etc) por el cual se envía la información. El receptor es donde se recibe la información y se le da un sentido.



Figura 2.1 Sistema básico de comunicación

El sistema se puede modificar según las necesidades del proyecto, para el prototipo desarrollado se presenta un modelo de sistema de comunicación (figura 2.2) que cubre áreas de importancia en las telecomunicaciones como son las modulaciones y codificaciones. Actualmente en todo tipo de comunicaciones existe algún tipo de modulación y de codificación, por ello este modelo se adapta a las necesidades del proyecto.

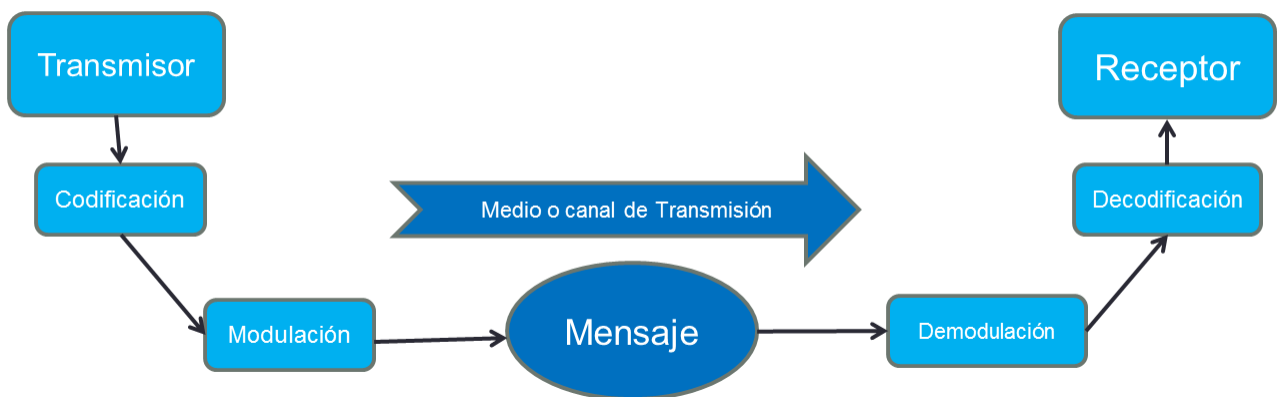


Figura 2.2 Modelo de sistema de comunicación

El mensaje a la salida del transmisor tiene que ser codificado, modulado y por consecuencia será demodulado y decodificado antes de llegar al receptor. Durante todo el desarrollo de la presente tesis se profundizó en conceptos como modulación, codificación, decodificación y demodulación.

Se diseñó el prototipo basado en una generalidad como es el sistema de comunicación básico para llegar a un dispositivo específico (prototipo de sistema de telemetría), siguiendo un modelo más complejo pero que continuara con los principios elementales para transmitir información.

Siguiendo el modelo de sistema de comunicación se buscó lo necesario para desarrollar un prototipo que no sólo cumpliera con los requerimientos planteados en la propuesta, también se consideraron aspectos técnicos importantes como son modulaciones y codificaciones por las ventajas que ofrecen.

Este capítulo bien podría fungir como Interface Control Documents³, ya que se abordará los sistemas que decidieron incluirse en el prototipo, también se describen las conexiones entre subsistemas e interfaces a utilizarse.

2.1. Computadora central

Es conveniente conocer el proceso de toma de decisiones, sólo así se podrá justificar correctamente los componentes utilizados en el desarrollo de los subsistemas. En una etapa muy temprana de este proyecto se pensó en utilizar el microcontrolador PIC16F887 como el responsable del procesamiento de la información. Después se comprobó que cumplía con los requerimientos básicos de portabilidad (no excede más de los 10 [cm]) y de potencia (fuente de alimentación de 2 a 5 [V] y corriente de 1[μ A]). Pero el procesamiento de información fue bastante problemático debido a que no poseía la capacidad de procesamiento necesaria, sólo llega a 3 [KHz] en ciclos de trabajo, lo cual no es adecuado si se trata de cantidades muy grandes de información. Posteriormente se optaron por otras tarjetas de desarrollo de prototipos.

Siguiendo la filosofía open source, que establece el apoyo de intercambio de conocimiento entre diferentes usuarios, se consideraron varias tarjetas de desarrollo que cumplieran el paradigma FLOSS. Para la decisión se tomaron en cuenta el sistema Arduino, Raspberry Pi, Beagle Bone, y PCduino.

Inicialmente se planteó en utilizar la plataforma Arduino pero el nivel de procesamiento de la tarjeta Arduino no era el óptimo, así como tampoco su eficiencia para agregar periféricos debido a que el esquema de conectar “módulo sobre módulo” para adquirir otros dispositivos periféricos hacía que las dimensiones

³ Adjuntos en la parte de anexos de la tesis

del prototipo se incrementaran. Se hizo la comparación con otras tarjetas Arduino⁴, pero básicamente son lo mismo ya que tiene una variante del mismo microcontrolador y con la misma capacidad para agregar dispositivos extras.

Se tomó en cuenta cuatro tarjetas de desarrollo:

- BeagleBone.- similar al Raspberry Pi, pero es más potente, con base en la TI Sitara AM335x, un SoC como procesador de aplicaciones que contiene un núcleo ARM Cortex-A8. Tiene más pines para poder controlar y agregar periféricos.
- PCduino.- tiene todas las características básicas de un Raspberry Pi, pero es más potente. En última instancia hace que sea más caro. Ya viene con 2 [GB] de memoria flash. Se le puede agregar una tarjeta SD de hasta 32 GB. Ventajoso en capacidad de almacenamiento.
- Pandaboard.- es una tarjeta de desarrollo de bajo costo, cuenta con una plataforma abierta de software. Es fácil de utilizar y altamente expandible. La base de su sistema es soportado por Linux al igual que el Raspberry Pi.

En la tabla 2.1 se muestra una comparación entre las características más importantes para tomar una decisión sobre que plataforma utilizar.

Tabla 2.1 Comparación entre plataformas de desarrollo de alta gama

Características	Raspberry Pi	BeagleBone	Panda Board	PCduino
Procesador/Frecuencia	Un solo núcleo/700 [MHz]	Un solo núcleo / 720[MHz]	Doble núcleo / 1 [GHz]	Un solo núcleo / 1 [GHZ]
Área	45[cm ²]	45[cm ²]	115[cm ²]	50[cm ²]
RAM	512[MB]	256[MB]	1[GB]	1[GB]
Precio en dólares	\$ 36.48	\$ 58.44	\$ 222	\$ 49.95

Se seleccionó a Raspberry Pi mostrado en la figura 2.3 debido a que es más barato y más popular dentro de las comunidades de intercambio de información de código abierto. Cuenta con las conexiones para periféricos necesarias (Ethernet, usb y serial). El pensar en utilizar un almacenamiento mayor se descarta debido a que solo se trabajarán con cadenas de datos pequeñas, sin que ocupen un valor significativo en medios de almacenamiento.

⁴ El Arduino YUN se descartó por la imposibilidad de adquirirlo dentro del país en el momento de creación de la propuesta para el prototipo



Figura 2.3 Raspberry Pi [71]

En el capítulo 3.1 se comentó más a fondo los convenientes, ventajas y características principales del Raspberry Pi. Las conexiones de interfaces utilizadas por la computadora central se observan en la figura 2.4

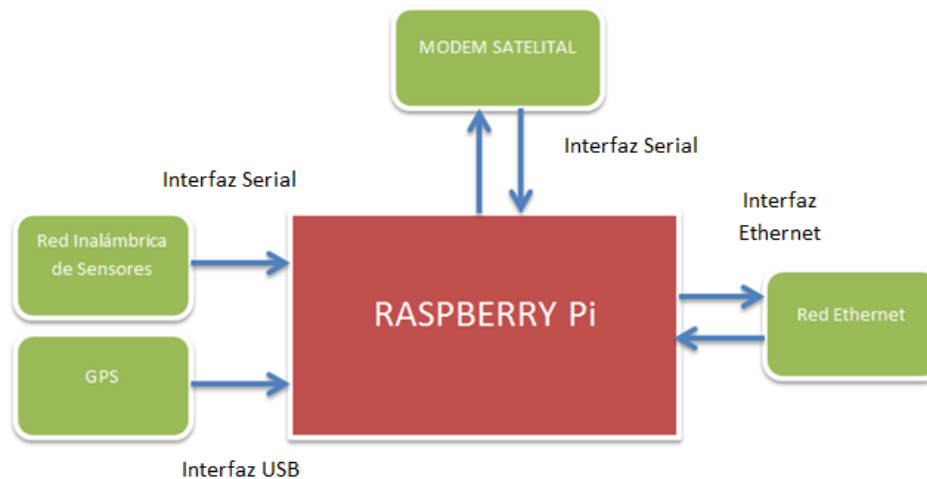


Figura 2.4 Computadora central y sus conexiones entre interfaces.

En las siguientes secciones se presentan las conexiones respectivas para cada subsistema.

2.2. Sistema de comunicación satelital

El uso de satélites artificiales en los sistemas de comunicación es un hecho dentro de la vida cotidiana, lo cual se evidencia por el número de hogares equipados con antenas parabólicas o “dish” utilizadas para la recepción de televisión por satélite. Lo que no puede ser tan conocido es que los satélites forman una parte

esencial de las telecomunicaciones alrededor del mundo, llevando grandes cantidades de datos, tráfico telefónico además de señales de televisión.

Estos vehículos espaciales no tripulados ofrecen una serie de características que no están disponibles con otros medios de comunicación. Grandes áreas de terreno del planeta Tierra son visibles desde un satélite, el satélite es el guía en la comunicación, dirigiendo las señales desde lo alto, vinculando de manera simultánea muchos usuarios que pueden estar ampliamente separados geográficamente. Esa misma cualidad permite a los satélites proporcionar enlaces de comunicaciones a comunidades remotas de baja densidad de población, donde las áreas son de difícil acceso por otros medios de comunicación (cable, fibra óptica, microondas terrestres).

Los satélites también se utilizan para la teledetección, siendo ejemplos la contaminación del agua en los océanos, mares y ríos, notificación de condiciones meteorológicas. Algunos de estos satélites de tele observación también forman un eslabón vital en las operaciones de búsqueda y rescate de aviones caídos. En la misma área de rastreo, los GPS son muy usados en la actualidad. Pero el dominio de interés ahora, es el de la comunicación y monitoreo. Como se mencionó, un satélite puede brindar un servicio de comunicación superior en áreas geográficas de difícil acceso, por ello se puede usar para enviar datos adquiridos a zonas estratégicas de importancia para el usuario final.

Para combinar el sistema de comunicación satelital se utilizó los puertos seriales. El modem cuenta con dos cables, uno de datos y otro de control. El que interesa es solamente el cable de datos. Usan conectores con el estándar de comunicación RS-232.

Se pueden usar dos modos de comunicación que ofrece el modem, modo asíncrono y modo por paquetes. Para este proyecto se eligió usar el modo de transmisión por paquetes por las facilidades y ventajas que este tiene. Dichas características se explicaran más a detalle en el capítulo 3.2.6.

El modem satelital toma el papel de DCE, un dispositivo que suministra los servicios de temporización y de datos a otro dispositivo. Estrictamente este proporciona un acceso a un enlace WAN. El Raspberry Pi toma el rol de DTE que es el dispositivo que recibe los servicios de datos desde el DCE y se ajusta a su temporización. Esta es la frontera entre el proveedor WAN y el cliente (CAT), la conexión entre el modem y el Raspberry Pi es también conocida como enlace de última milla, término utilizado por los proveedores de servicios (carriers)

En la figura 2.5 se observa el diseño de la conexión satelital. El DTE puede recibir y transmitir datos en el puerto de datos mediante comandos AT y al mismo tiempo comunicarse con el modem satelital en el puerto de control para actualizaciones en el software.

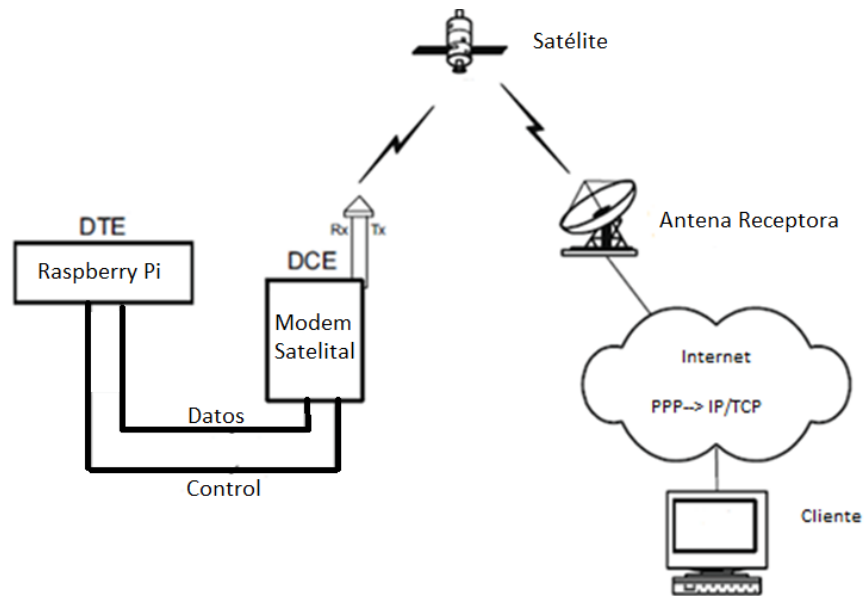


Figura 2.5 Esquema de conexión satelital [21]

Los comandos AT son una serie de símbolos que utilizan los módems para transmitir (codificar, decodificar y encriptar) información. El protocolo de red de capa dos (según el modelo OSI de referencia) fue PPP y en capa tres se usó empaquetamiento IP.

La idea básica del funcionamiento del sistema es sencilla: el Raspberry Pi tiene el objetivo de adquirir y transmitir datos, los cuales se envían por el puerto de datos al modem satelital. Enseguida los datos pasan por los protocolos de capa tres y capa dos, llegan al modem y este los envía por microondas a través de la antena hacia el satélite, que se encuentre en línea de vista de la constelación satelital. De ahí se envía la información, igualmente por un enlace de microondas a la antena en la estación terrena que recibe la información y la procesa. Hace la trama de capa dos y encapsula en capa tres los datos para poder enviar todo por internet a una ubicación remota, a la cual tendrá acceso el CAT o la empresa que proporcione el servicio satelital.

Para hacer la conexión físicamente se investigó y se optó por usar un circuito integrado MAXIM 232 para hacer la conversión de niveles de lógicos. El diagrama

mostrado en la figura 2.6 detalla las conexiones entre las interfaces UART del Raspberry Pi y el cable DCE.

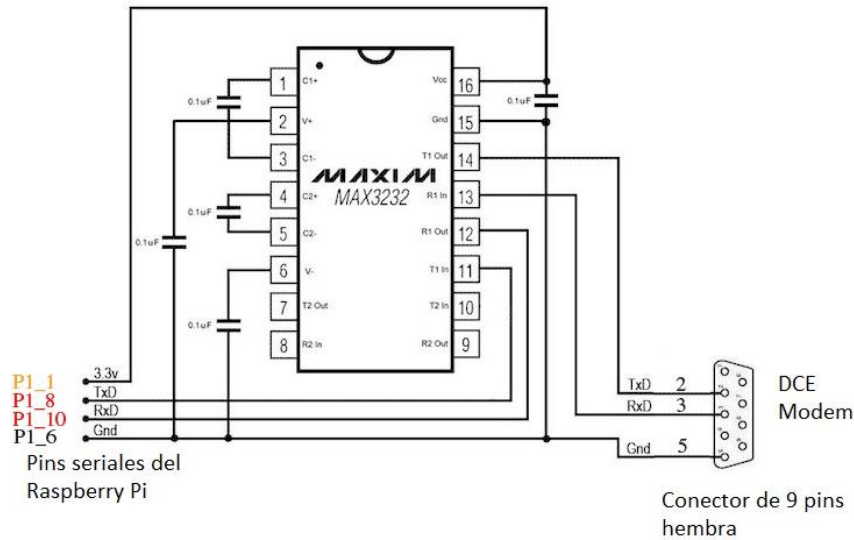


Figura 2.6 Diagrama de conexión serial entre Raspberry Pi y modem satelital

Un diagrama de bloques se muestra en la figura 2.7, de esta forma es más fácil entender el concepto de conexión entre interfaces.

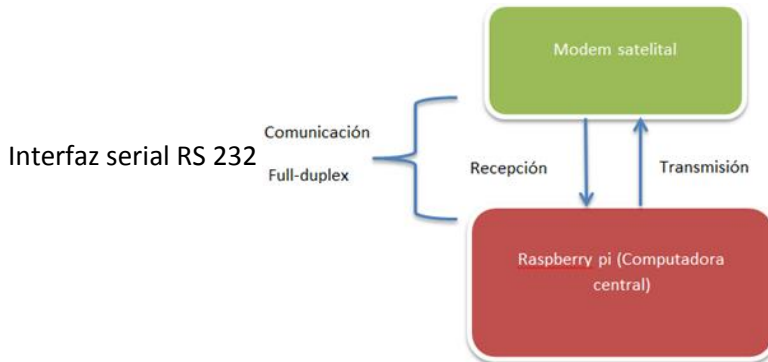


Figura 2.7 Esquema de conexión del sistema de comunicación satelital

Para una parte de experimento se realizaron pruebas simulando el funcionamiento de un modem satelital con una tarjeta de desarrollo de prototipos Arduino. Se implementó un sistema embebido PPP en Arduino para hacer la conexión con la computadora central, obsérvese en la figura 2.8 la conexión entre interfaces. La conexión con las interfaces entre Arduino y el Raspberry Pi solo requiere un cambiador de voltajes para asegurar una correcta conexión entre interfaces y evitar que los circuitos se dañen a causa de diferencias de potencial.

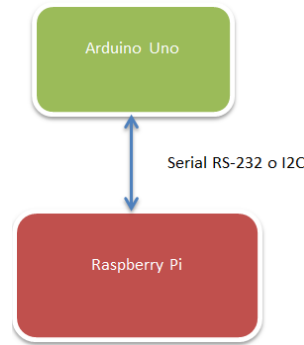


Figura 2.8 Esquema de simulación del modem satelital

Con propósitos de investigación, experimentación y para obtener el mejor resultado, se probó dos modalidades diferentes de conectar el sistema PPP programado, una de ellas fue mediante conexión serial UART (Figura 2.9) y otra por medio de los puertos I2C (figura 2.10).

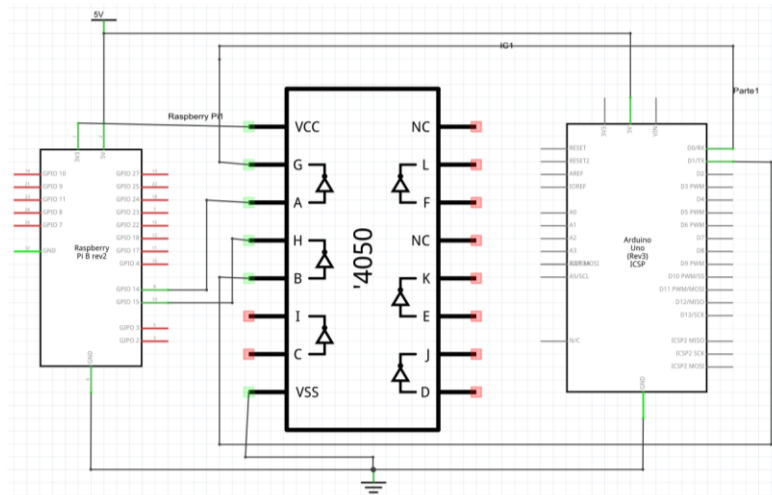


Figura 2.9 Diagrama de conexión serial entre interfaces Arduino y Raspberry Pi vía puertos UART

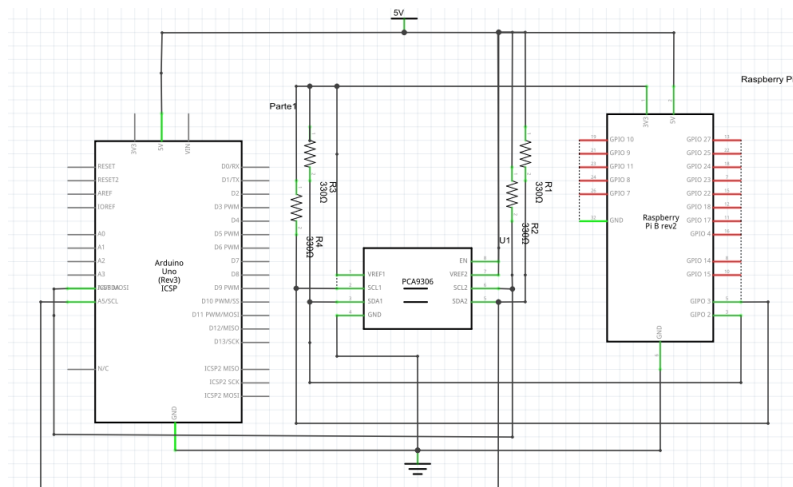


Figura 2.10 Diagrama de conexión serial entre interfaces Arduino y Raspberry Pi vía puertos I2C

Al final se optó por utilizar los puertos UART por las ventajas que ofrece. Así se propusieron los diferentes modos de comunicación utilizados en el experimento y en las pruebas de validación y verificación.

2.3. Sistema de comunicación inalámbrica

En la parte inalámbrica se utilizó una conexión entre Raspberry Pi y un módulo XBee, dicho modulo funcionara como nodo agregador. La conexión se hace mediante los puertos GPIO del Raspberry Pi, más específicamente los I2C. En la figura 2.10 se tiene el diagrama conceptual básico.

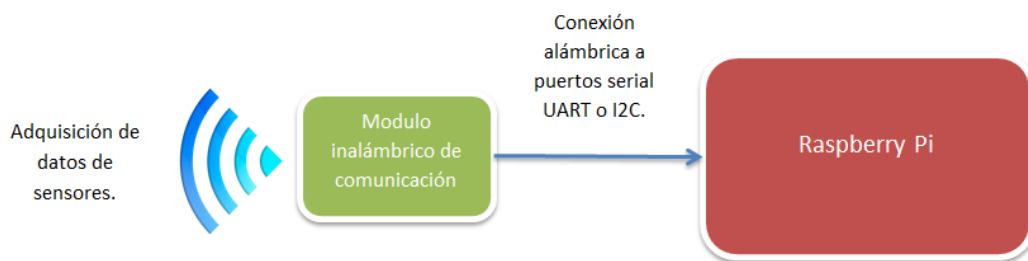


Figura 2.11 Esquema de conexión entre modulo inalámbrico y computadora central

Como en el caso del subsistema anterior, existe la posibilidad de conectar el modulo inalámbrico vía puertos UART o I2C. Se establecen las dos configuraciones entre las conexiones de interfaces, una mostrada en la figura 2.11 los pines UART, y en la figura 2.12 por medio de pines I2C.

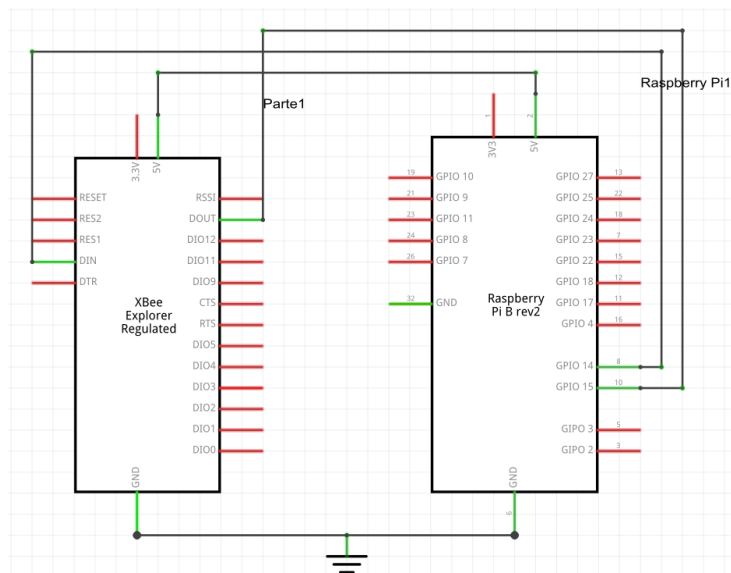


Figura 2.12 Diagrama de conexión entre modulo inalámbrico y computadora central vía UART

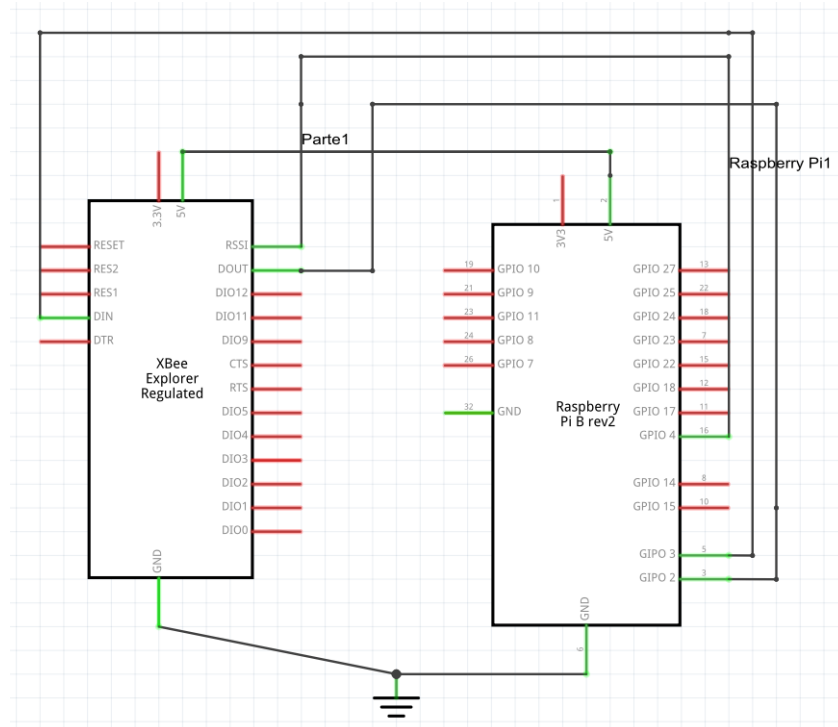


Figura 2.13 Diagrama de conexión entre modulo inalámbrico y computadora central vía I2C

El modulo inalámbrico en conjunto con el sistema de sensores, puede traducirse en una red inalámbrica de sensores, WSN por sus siglas en inglés, aparece en la figura 2.13. Las WSN son actualmente muy utilizadas para el tratamiento y adquisición de datos y es el sistema de telemetría destinado para este prototipo.

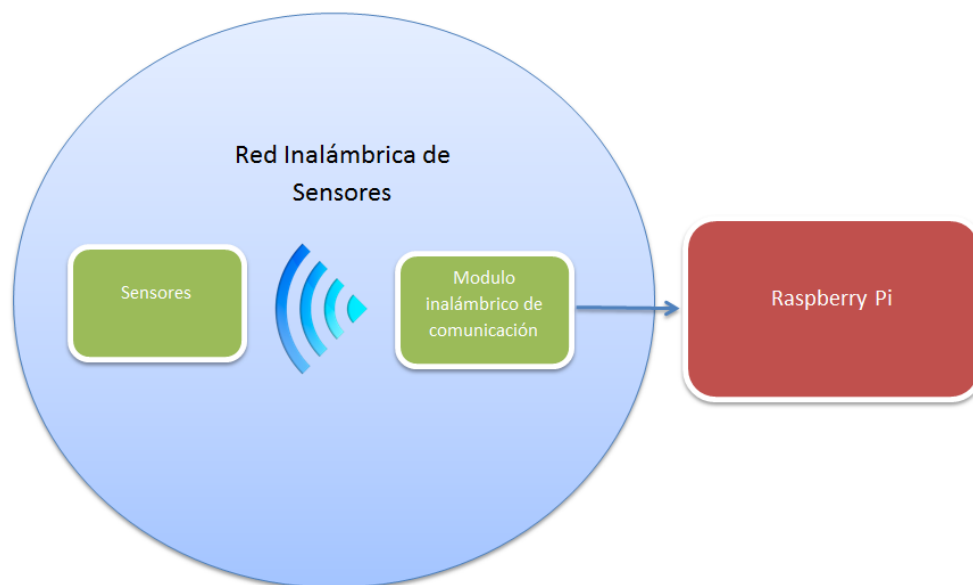


Figura 2.14 Esquema de la red inalámbrica de sensores conectada al Raspberry Pi

2.4. Sistema de comunicación Ethernet

A pesar de los avances tecnológicos actuales, las comunicaciones alámbricas siguen siendo el medio más usado para poder compartir información. En los últimos años han existido nuevas tecnologías que quieren encarar a las redes cableadas: desde los satélites, tecnologías inalámbricas, lo cierto es que las redes alámbricas aún son el medio más fiable y barato para transmitir información.

Se pensó en una manera auxiliar para poder transmitir los datos adquiridos, en el caso de que la comunicación por satélite fallara o presentara problemas. Para ello se utilizó el puerto Ethernet RJ45 disponible en el Raspberry Pi. Como destino final se presenta una red de datos, la cual es operada por el CAT de la UNAM.

El esquema de la figura 2.14 deja en claro la conexión entre interfaces, así como también la idea de la transmisión de datos por este medio es de manera full-duplex.

Cuando la información sale de la computadora central, se considera que está entrando a una red LAN, en este caso, una LAN administrada por el CAT. La utilidad de tener un sistema de respaldo de comunicación full-dúplex es poder controlar de manera remota la computadora central, un usuario debidamente autorizado puede ingresar al Raspberry Pi desde cualquier computadora conectada a la red del CAT y realizar cambios en su configuración.

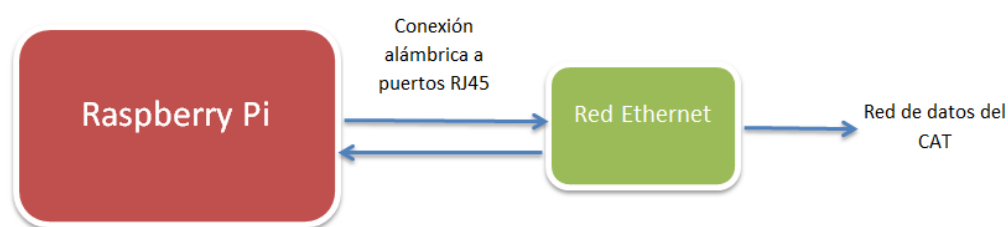


Figura 2.15 Esquema de conexión a red de datos

Si se desea acceder a los datos de los sensores, o modificar algún parámetro relacionado con ellos, solo basta con tener disponible cualquier computadora, en cualquier parte del mundo, y si se tienen los requerimientos (contraseñas de autenticación, dirección ip, usuario) se podrá tener ingreso a la computadora central. Obviamente esto es poco factible si el Raspberry Pi no está conectado a internet vía Ethernet, pero es un buen método de respaldo en caso de ausencia de enlace satelital.

2.5. Sistema de sensores y adquisición de datos

Los sensores son parte primordial de este proyecto, sin ellos no se tendría un mensaje para transmitir y se han convertido en un medio muy utilizado en la actualidad.

Bonganay (2014) argumenta que actualmente la lectura de medición automática (AMR en inglés) está en apogeo y utiliza sensores para cumplir sus objetivos. Esa es la tecnología que permite recoger de datos de forma automática a partir de la medición de energía, y agua, principalmente en casa habitación, y la transferencia de los datos recogidos para la facturación y el análisis. El principal motor para este tipo de tecnología es la recolección de datos difíciles de obtener, por lo que es adaptable a condiciones específicas del área industrial y aeroespacial.

Se observa en la figura 2.15 que hay tres diferentes sensores (humedad, temperatura y movimiento). Estos se comunican de manera inalámbrica con el módulo inalámbrico que está paralelamente conectado al Raspberry Pi.

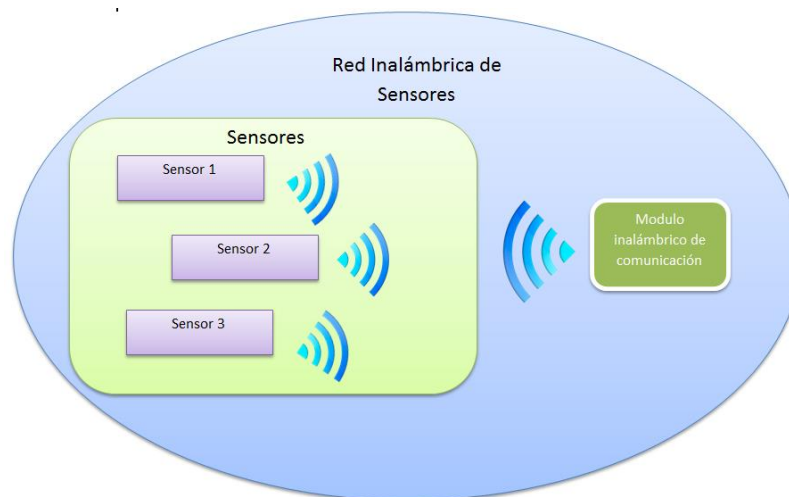


Figura 2.16 Esquema del sistema de sensores

Los sensores, guiados y controlados por la computadora central, cuentan con la capacidad de mandar los datos de manera ordenada, así se puede discriminar entre cual información pertenece a cada sensor y evitar fallas.

2.6. Sistema de control y monitoreo

Existen dos diferentes sistemas de control: sistema centralizado y sistema distribuido. Un sistema centralizado concentra el control en un solo dispositivo, en un solo sistema. El sistema distribuido es cuando se realiza el control a través de diversos sistemas esparcidos en toda la red.

El centralizado es ideal cuando el sistema general no es muy grande y brinda las ventajas de fácil mantenimiento ya que solo existe un único controlador y también se eliminan los problemas de compatibilidad con otros microcontroladores. La única desventaja es cuando llegara a fallar el sistema de control todo se detendría, para eso se realizaran pruebas para visualizar los posibles escenarios de fallas.

Para que los datos sean correctamente tratados se necesita un sistema que los maneje y regule. El microprocesador del Raspberry Pi brinda la capacidad necesaria para fungir como un sistema de control en el prototipo. En una etapa más temprana de este proyecto, se pensó en otro microcontrolador que cumpliera con esta función (sistema distribuido). Se concluyó que no era necesario el tener otro, debido al espacio extra que implicaría. El no tener otro microcontrolador también implica que habrá menos espacio para conectar otros periféricos, dispositivos, etc. Para resolver dicho problema, se pensó la manera de expandir esos límites. El diagrama con la solución se observa en la figura 2.16, el circuito integrado visible está conectado a la computadora central y permite agregar, por ejemplo, más sensores.

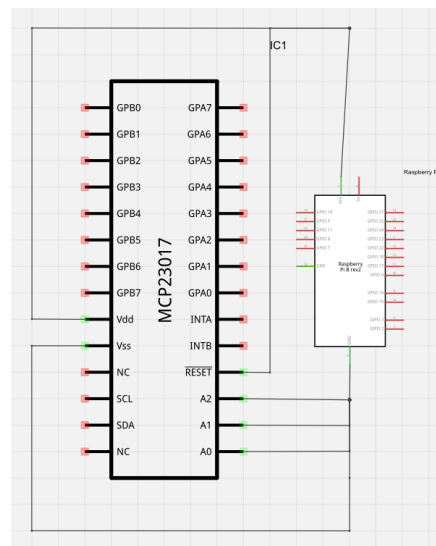


Figura 2.17 Diagrama de conexión entre la computadora central y dispositivo de expansión de periféricos

En colaboración con un compañero del CAT se desarrolló un sistema GPS. Se integró al prototipo. La finalidad principal del sistema fue proveer una ubicación exacta de donde se encuentra el prototipo.

Como se puede observar hay una multitud de dispositivos que convergen en un solo sistema. Lo más básico es tener un monitor, teclado y recomendable, mouse, conectados al Raspberry Pi. Los demás periféricos se posicionan en interfaces auxiliares de la computadora central como se muestra en la figura 2.18.

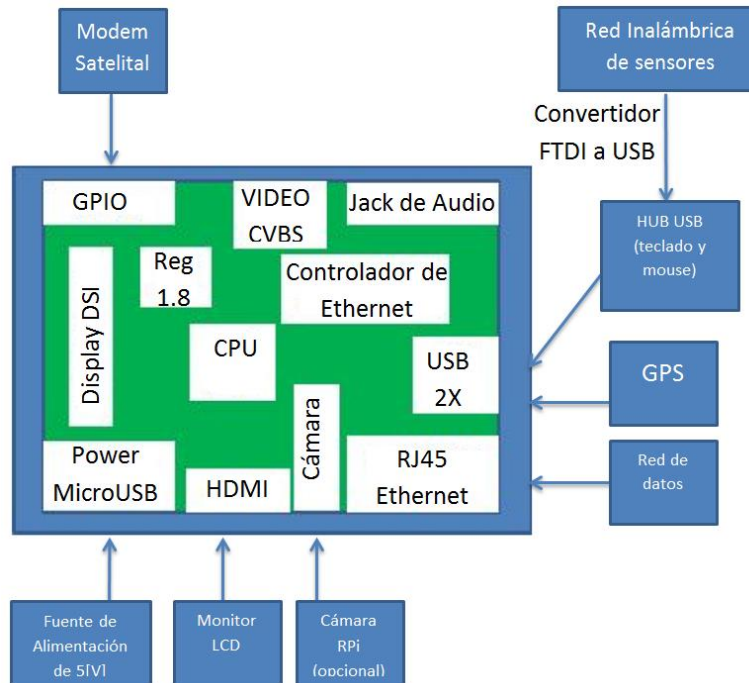


Figura 2.18 Diagrama general de conexión de interfaces

El funcionamiento general del sistema (figura 2.19.) es el siguiente: se obtienen los datos paramétricos del ambiente censado. Los sensores proceden a guardarlos y los envían al nodo de transmisión inalámbrico para que sean transmitidos al módulo inalámbrico conectado al Raspberry Pi. Se almacena la información y existen las opciones de poder enviar vía satelital usando el modem mediante la interface serial o vía alámbrica hacia una red de datos Ethernet.

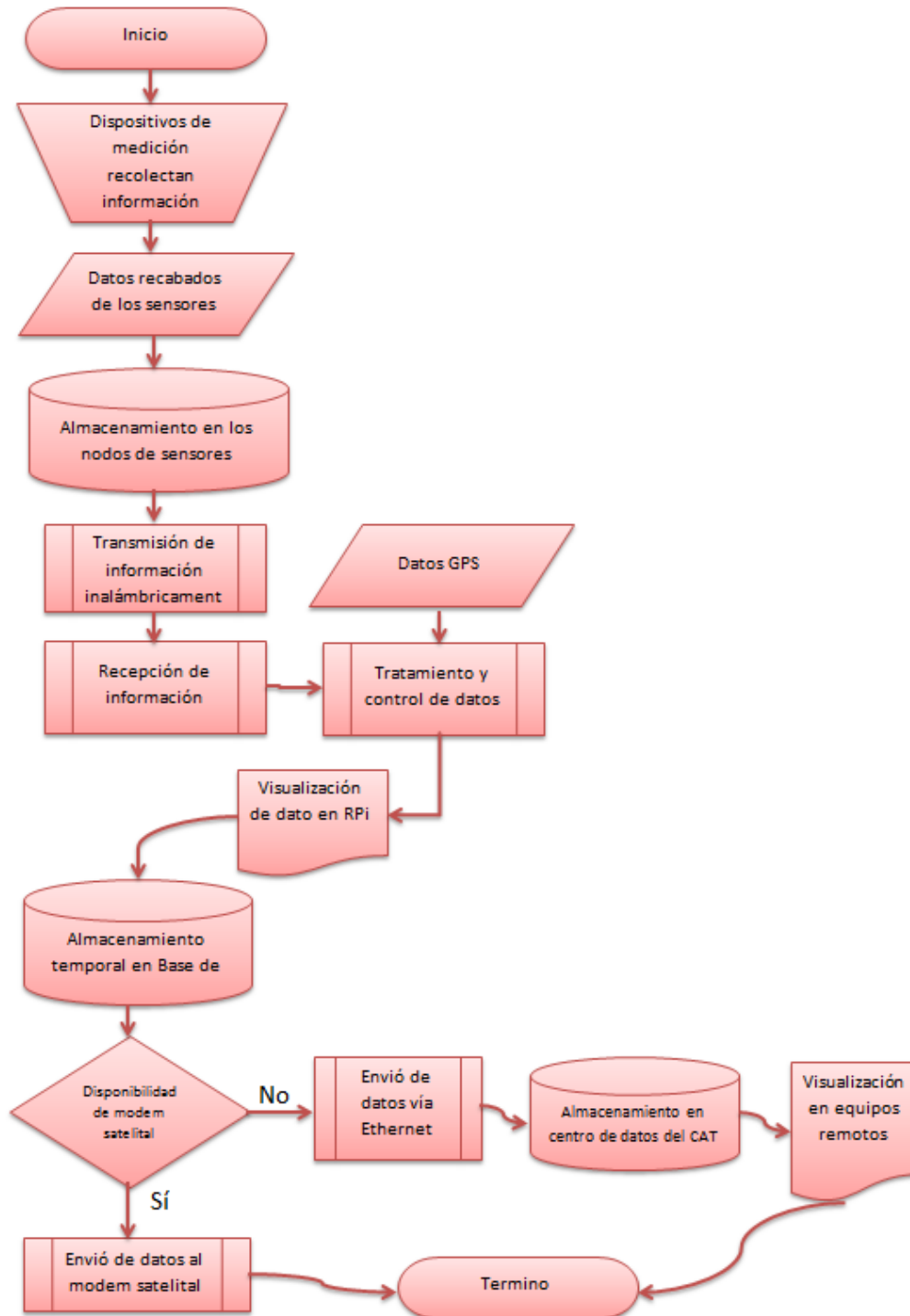


Figura 2.19 Diagrama de flujo del funcionamiento del prototipo

El diagrama de flujo describe un funcionamiento muy general del prototipo. Detalles más específicos de abordan en el capítulo tres, así como también otros diagramas utilizados en la programación de diferentes subsistemas del prototipo.

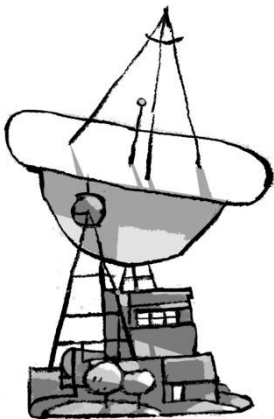
Capítulo 3. Características de los sistemas

Science and technology multiply around us. To an increasing extent they dictate the languages in which we speak and think. Either we use those languages, or we remain mute.

J. G. Ballard

La computadora central era una esfera de gelatina roja del tamaño de una casa, suspendida del techo por millones de cables...

Bernardo Fernández, Bef



¿Qué son los satélites? ¿Por qué usarlos en la actualidad? ¿Qué se puede medir con ellos y cuáles son sus límites? ¿Medir en tierra o en órbita espacial? ¿Qué se usa para medir los parámetros? Y aún más importante ¿Cómo se usan?

En el siguiente capítulo se exploraron las peculiaridades de los sistemas utilizados para el prototipo. Se explicó breve, pero de manera consistente, las razones por las que se optaron en usar ciertas interfaces, diseños, herramientas y protocolos en el proyecto. Será un ejercicio de argumentar para poder concluir la mejor solución. ¿Qué funciones desarrolla el microprocesador central? ¿Cuáles son sus fortalezas y debilidades? Se empieza a describir el Raspberry Pi y mencionar las posibilidades que tiene como plataforma de desarrollo. Luego se dio una descripción general de los sistemas de comunicación satelital en la actualidad. También está presente la descripción del sistema alámbrico utilizado, los protocolos más relevantes que fueron implementados: PPP, IP. De igual manera se aborda los diferentes sistemas de medición de parámetros existentes en la actualidad, los sensores y como se manifiestan en redes inalámbricas de sensores. Es importante resaltar que muchas pruebas se usaron con sensores conectados de manera alámbrica. También se abordan lo concerniente a protocolos para el cableado de los sensores, cubriendo I2C, TTL, en serie y SPI.

3.1 Computadora central

En esta tesis, la computadora corresponde a un sistema de procesamiento que controla todos los subsistemas del proyecto. Otros diseños en el mercado pueden diferir de este, al tener varios microprocesadores a cargo de subsistemas diferentes. Se optó por el tipo arquitectura convergente en un solo microprocesador debido a las dimensiones pequeñas que debe tener el prototipo para facilitar su movilidad. En este caso es más fácil controlar todo en un nodo donde convergen varios sistemas en vez de tener muchos nodos. Así se ahorra espacio y recursos (ahorro de periféricos y cables a conectar).

3.1.1. Raspberry Pi

El Raspberry Pi es una minicomputadora, su sistema de procesamiento es un chip ARM1176JZF-S a 700 [MHz] de velocidad de procesamiento. Cuenta con un sistema gráfico visualilla desde una salida HDMI. Puede conectarse un teclado y un monitor, cargar una de las distribuciones Linux como sistema operativo, y requiere menos conocimientos técnicos para comenzar a usarse que otros sistemas

embebidos comerciales como son PandaBoard o BeagleBone. Es una plataforma increíblemente potente en un encapsulado muy pequeño de tamaño y perfecta para sistemas embebidos, o proyectos que requieren una mayor interactividad y procesamiento de energía

Fue en 2011 cuando se sucinto el lanzamiento del Raspberry Pi, desde entonces ha encontrado un papel importante como un equipo computadora de muy bajo costo basado en Linux y como plataforma de informática embebida. Ha demostrado ser popular entre los educadores y aficionados por igual, con millones de unidades vendidas alrededor del mundo.

Michalak (2014) habla de un análisis comparativo de sus elementos clave como son el procesador y batería usada, también trabajo con algunos de los nodos de sensores inalámbricos (Texas Instruments y FPGA's) actualmente existentes, y ha demostrado que a pesar de algunos inconvenientes, el Raspberry Pi sigue siendo un equipo de bajo costo con gran éxito en el dominio de red de sensores y diversas gamas de aplicaciones de investigación.

Pero, en verdad, ¿qué es? Raspberry Pi es una placa pequeña, potente, barata, hackeable y orientada a la educación en el campo de la informática (Figura 3.1). Una computadora de tamaño aproximado al de una tarjeta de crédito con muchos funcionamientos y asequible para interactuar con muchos dispositivos.

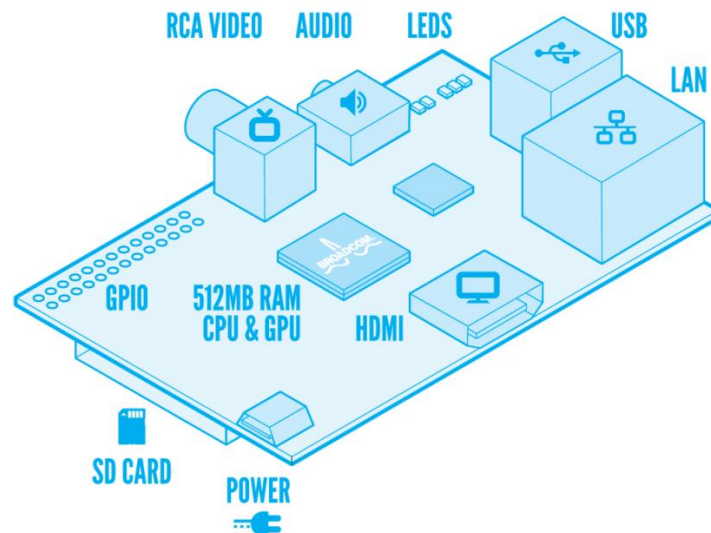


Figura 3.1 Diagrama de puertos del Raspberry Pi [71]

Todos los modelos Raspberry Pi tienen el mismo CPU llamado BCM2835, que es barato, de gran potencia, y que no consume mucha energía. Opera en la misma forma que una computadora estándar, lo que requiere un teclado para la entrada de

comandos, una unidad de visualización (monitor HDMI) y una fuente de alimentación. La unidad se alimenta a través del conector micro USB. La conectividad a internet puede ser a través de un cable Ethernet / LAN o a través de un dongle USB (conectividad WiFi).

Al igual que cualquier otro equipo, el Raspberry Pi también utiliza un sistema operativo el cual es una distribución de Linux llamada Raspbian. Así se aprovechan las ventajas de Linux, como sistema libre y de código abierto. Una de las mejores cosas del Raspberry Pi es el contar con una amplia gama de aplicaciones ya probadas: trabajos con sensores, pantallas y motores.

Vujovic y Maskimovic (2014) también afirman que las redes de sensores son sólo un ejemplo de cómo se pueden utilizar estos componentes pequeños, potentes y de bajo costo. El construir una red ZigBee XBee-Mesh resulta no tan tediosos comparado con otros sistemas embebidos (MicaZ, Telosb, Iris, Cricket, Lotus).

La posibilidad de utilizar conjunto el Raspberry Pi como un servidor de automatización ZigBee vuelve la tarea mucho más atractiva para los desarrolladores comunes y para los más expertos. Por otro lado, puede ser configurado para soportar una gran variedad de tareas diversas en el área de la domótica, incluyendo aire acondicionado, iluminación con control remoto, y comprobar si las puertas están abiertas o cerradas para brindar mecanismos de seguridad. Trabajos electrónicos según Costanzo (2013), proponen la combinación del Raspberry Pi con el poder de la plataforma Arduino (igualmente con código abierto), y presentan numerosos proyectos de domótica y redes de sensores.

Existen diferentes distribuciones de Linux corriendo sobre el Raspberry Pi, en el estudio realizado por Milenkovic (2013) hace comparación entre Pidora, Moebius, Raspbian, Arch Linux ARM. De las cuales se opta por usar Raspbian gracias a su facilidad y al código (Python) nativo con el que cuenta. Así es como se decidió usar como sistema operativo.

Uno de las metas del prototipo es que sea de bajo consumo, lo cual se adapta perfectamente a la estructura poco demandante de energía con la que cuenta. Un funcionamiento de ultra bajo consumo de energía sólo puede lograrse mediante la combinación de componentes de hardware de baja potencia y un funcionamiento de bajos ciclos de trabajo.

Kochlan y Cechvic (2014) comprobaron la capacidad del procesador al trabajar con varios tipos de programación, como la programación en paralelo y



aplicaron el sistema a un monitoreo en tiempo real, lo que siempre demanda un constante uso del procesador.

El BCM2835 basado en ARM es la razón por la Raspberry Pi es capaz de funcionar con sólo la fuente de alimentación de 5[V] a 1 [A], provista por el puerto incorporado USB micro. El Raspberry Pi para operar requiere desde 700[mA]. Se puede energizar con una amplia gama de fuentes de alimentación (suponiendo que son capaces de proporcionar suficiente corriente de hasta 1 [mA] en consumo máximo) como: Puerto USB de la computadora, hubs USB, baterías de teléfonos celulares, pilas alcalinas (seis pilas AA recargables y un regulador de tensión o baterías recargables de alta capacidad), o sistema de celdas solares (para uso en exteriores). Para movilidad, S. Monk (2014) argumenta haber realizado pruebas en donde funciona alrededor de 2 horas con solamente baterías, lo cual dependerá de los periféricos conectados, esto brinda otra ventaja al trabajar. Como se observa hasta ahora, hay muchas maneras diversas de energizar la computadora central.

El Raspberry Pi cuenta con cuatro modos de potencia distintas:

- El modo de ejecución lo que significa que el CPU y todas las funcionalidades del núcleo ARM11 están disponibles y encendidas.
- El modo de espera, en donde los principales relojes de núcleo están apagados (las partes del CPU que procesan las instrucciones ya no están ejecutándose) a pesar de que los circuitos de potencia en el núcleo están activos para que pueda encender. En este modo el núcleo se puede prender rápidamente por un proceso generado por una llamada espacial del CPU, la cual se le hace llamar llamada de interrupción. Esta interrupción detendrá cualquier otro proceso en ejecución actual y hará solamente lo que el proceso que llamada este solicitando. También se le conoce como como Wait For Interrupt. Este proceso es muy rápido y utilizado solo para tareas embebidas muy específicas.
- El modo de apagado. En este no hay nada de poder que alimente al sistema.
- El modo inactivo. El núcleo está apagado y todos los sistemas de memoria y están activos.

Es importante tener en cuenta que no hay disco duro en el Raspberry Pi; todo se almacena en una tarjeta Secure Digital (SD). Aunque las grandes tarjetas SD están disponibles y alcanzan capacidades de 32 [GB], 64 [GB] o más, el tamaño mínimo requerido de la tarjeta SD es de 2 [GB], dependiendo de las exigencias de distribución del sistema operativo. Este almacenamiento se puede ampliar mediante el uso de dispositivos que proporcionan un disco duro adicional en el uso de los puertos USB.

Otro de los puntos importantes y definitivos a la hora de escoger al Raspberry Pi, fue su flexibilidad. Su capacidad para conectar otros periféricos es de suma importancia. Existen dos puertos USB 2.0 permitiendo dispositivos expandibles con hubs. Salida HDMI para pantallas de televisión de nueva generación. Puede soportar DSI (Display Serial Interface) para agregar una pequeña pantalla y mantener la portabilidad. Cuenta de la misma forma con CSI (Camera Serial Interface) en donde es factible agregar una cámara.

Una parte muy importante son las interfaces GPIO, y son la principal manera de conectarse con otras tarjetas electrónicas, sensores, motores, etc. Se observa en la figura 3.2 y se compone de 26 pines dispuestos en dos filas que contienen 13 pines cada una. La fila de la izquierda contiene los pines pares, y la fila los pines impares. Los pines GPIO pueden aceptar comandos de entrada y salida, por lo tanto se pueden programar en el Raspberry Pi de forma nativa. Es importante tener en cuenta que existe cierta diferencia entre los pines. Algunos de pines GPIO (tabla 3.1) se pueden utilizar como entradas o salidas digitales y como interfaces para protocolos incrustados. Los tres protocolos más importantes que se trabajaran a lo largo del desarrollo son: I2C, SPI, UART.

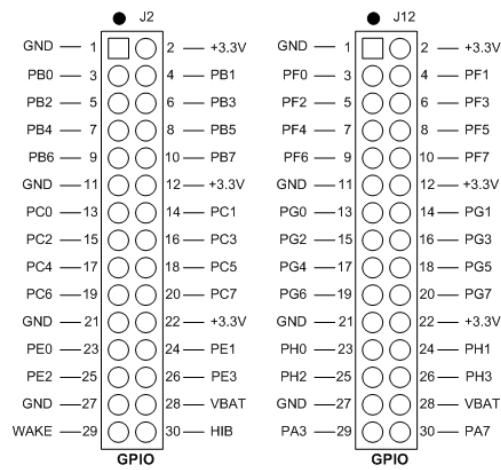


Figura 3.2 Conectores GPIO [69]

Tabla 3.1 Diferentes controladores y puertos del chip Broadcom [79]

Nombre del puerto	Función
SDA0	Línea de datos para bus de control serial 0
SCL0	Línea de reloj para bus de control serial 0
SDA1	Línea de datos para bus de control serial 1
SCL1	Línea de reloj para bus de control serial 1
GPCLK0	Reloj de propósito general 0
GPCLK1	Reloj de propósito general 1
GPCLK2	Reloj de propósito general 2
SPI0_CE1_N	Controlador de SPI 1
SPI0_CEO_N	Controlador de SPI 0
SPI0_MISO	MISO SPI 0
SPI0_MOSI	MOSI SPI 0
SPI0_SCLK	Reloj de SPI 0
PW Mx	Modulador de ancho de pulso
TXD0	Transmisor de datos UART 0
RXD0	Receptor de datos UART 0
CTS0	Línea de limpiar para enviar, UART 0
RTS0	Línea de preguntar para enviar, UART 0
PCM_CLK	Reloj PCM
PCM_FS	Trama de sincronización PCM
PCM_DIN	Datos de entrada PCM
PCM_DOUT	Datos de salida PCM
SAX	Bus de direcciones de memoria secundaria
SOE_N/SE	Controles de memoria secundaria
SWE_N/SRW_N	Controles de memoria secundaria
SDx	Bus de datos de memoria secundaria
BSCLS SDA/ MOSI	Línea de datos de SPI
BSCLS SLC/ SCLK	Línea de reloj de SPI
BSCLS- / MISO	MISO SPI
BSCLS- /CE_N	CSn SPI
SPI1_CEx_N	Seleccionador de SPI 1 [0-2]
SPI1_MISO	MISO SPI 1
SPI1_MOSI	MOSI SPI 1
SPI1_SCLK	Reloj de SPI 1
TXD0	Transmisor de datos UART 1
RXD0	Receptor de datos UART 1
CTS0	Línea de limpiar para enviar, UART 1
RTS0	Línea de preguntar para enviar, UART 1
SPI2_CEx_N	Seleccionador de SPI 2 [0-2]
SPI2_MISO	MISO SPI 2
SPI2_MOSI	MOSI SPI 2
SPI2_SCLK	Reloj de SPI 2

Un protocolo de comunicación es un conjunto de reglas. En este caso define como los bits viajan a través del medio, incluyendo el momento cuando una



computadora puede transmitir, la tasa de bits que se enviarán, y en qué orden serán transmitidos los bits.

Dos equipos que quieren intercambiar datos deben ponerse de acuerdo sobre si los dos extremos pueden transmitir a la vez o si los equipos tienen que esperar su turno. La mayoría de cables encargados de vincular a los dos equipos son full-duplex: ambos equipos pueden transmitir y recibir a la vez al mismo tiempo. Muchos enlaces inalámbricos son half-duplex: los equipos deben tomar turnos para transmitir.

Un protocolo de comunicación puede incluir el uso de líneas de estado y de control. Estas líneas pueden indicar cuando un transmisor tiene datos para enviar o cuando un receptor es capaz de aceptar nuevos datos. El proceso de intercambio de información se llama control de flujo. Los dispositivos también pueden utilizar el control de flujo de software para proporcionar la misma información mediante el envío de códigos definidos, típicamente en la misma ruta de acceso utilizada para los datos.

Las comunicaciones seriales a menudo intercambian mensajes que consisten en bloques de datos con formatos definidos. Un protocolo de mensajes puede especificar el tipo de datos que contiene un mensaje y cómo la información se estructura dentro del mensaje. Los dispositivos de una red necesitan una manera de detectar qué equipo es el receptor previsto para los datos transmitidos. Las redes suelen asignar una dirección a cada equipo e incluir la dirección del receptor en cada mensaje. Por ejemplo, un mensaje muy básico podría consistir en dos bytes: un byte para identificar el receptor y un byte que contiene datos. Cuando se habilita un equipo receptor para detectar el inicio y el final de un mensaje, un mensaje puede incluir códigos para indicar estos eventos o un encabezado que almacena el mensaje longitud. Un mensaje también puede incluir uno o más bytes que el receptor del equipo utiliza en la comprobación de errores.

La comunicación serial o mediante los pines UART/RS-232 son asíncronos, requieren relojes alineados en ambos lados. La comunicación paralela transfiere múltiples bits al mismo tiempo. Usualmente requiere varios buses de datos para transmitir a través de ocho, dieciséis o más cables. Se necesita un cable para transmitir cada uno de los diferentes bits además de un reloj que envía los pulsos y que ordene el envío de bits cada vez que este envíe un pulso, como se muestra en la figura 3.3.

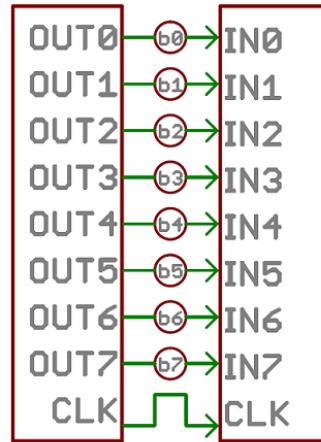


Figura 3.3 Bus de 8-bits controlado por un reloj, transmite cada byte por cada pulso de reloj [6]

El envío de datos de manera serial es diferente, un solo bit se envía a la vez, en un solo lapso. Estas interfaces pueden operar con un solo cable que transmita los datos y uno como portador del pulso de reloj (Figura 3.4).

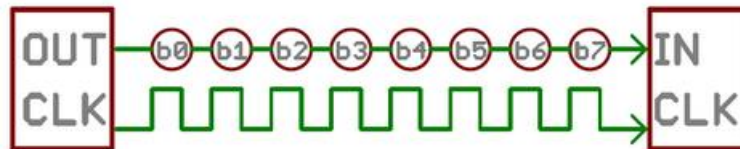


Figura 3.4 Interface serial, transmitiendo un solo bit cada pulso de reloj [6]

Una buena analogía que siempre ayuda es pensar en estos buses de información como autopistas. La comunicación paralela es una gran autopista con 8 carriles, mientras que la comunicación serial es más como una carretera de un pueblo rural con solo dos carriles. Es cierto que se puede enviar más información por la gran autopista, pero con menos recurso, se puede cumplir su objetivo que es simplemente el enviar información de manera segura, este caso sería el escenario de la carretera rural. La comunicación paralela tiene beneficios, es rápida, fácil de seguir, pero requiere muchas líneas de entrada y salida de datos, lo que resulta engorroso y burdo al momento de conectar. Es por ello la gran mayoría de dispositivos electrónicos para ahorrar espacio y conservar portabilidad, han optado por usar comunicación serial.

Muchos protocolos han sido diseñados para satisfacer las necesidades particulares de los sistemas embebidos. USB y Ethernet son un par de interfaces seriales muy conocidas. Este tipo de interfaces son asíncronas. La diferencia radica en que un tipo de interface síncrona siempre estará emparejada en su línea de datos con una señal de reloj, por lo que todos los dispositivos que usen comunicación

síncrona en un mismo bus de datos tendrán que usar un reloj en común. La consecuencia es que existirá una transferencia más sencilla y a menudo más rápida, pero también requiere un cable adicional entre dispositivos de comunicación. Este tipo de interfaces son los protocolos I2C y SPI.

Asíncrono quiere decir que los datos se transfieren con el apoyo de una señal de reloj externa. Este método de transmisión de datos es más adecuado cuando se quiere minimizar los cables y pines de entrada o salida, pero significa que hay que poner esfuerzo extra en transferir y recibir datos de forma fiable. Normalmente este tipo de comunicación tiene ciertas reglas en su protocolo que evitan los errores al momento de enviar información, así se asegura que lleguen de manera segura al destino. Uno de ellos es la velocidad de transmisión, que especifica que tan rápidamente se envían los datos a través del cable en protocolo serial. Ambos dispositivos (transmisor y receptor) tiene que funcionar a la misma velocidad, normalmente también tienen que tener las mismas especificaciones en protocolos para que funcionen de manera correcta. De no ser así, los datos pueden recibirse con errores o simplemente no recibirse.

Cada bloque de datos transmitidos se envía en realidad en un paquete o trama, se designan esos nombres dependiendo del nivel de capa OSI en donde se estén analizando los datos. A las tramas que llevan los datos se les agrega bits de sincronización y bits de paridad. En algunos casos también hay bits para determinar cuándo es donde la trama termina, en la figura 3.5 se observa un pequeño esquema de una trama genérica.



Figura 3.5 Tamaño de bits en una trama de datos [6]

La información realmente importante, lo que se transmite se encuentra en la zona de datos. El tamaño de dicho trozo puede variar, pero el tamaño estándar es de 8 bits por paquete. Los bits de sincronización son dos o tres bits transferidos especialmente en cada fragmento de datos. El bit de inicio (start) y el bit para final (stop), son los que indican donde inicia y donde termina la trama. Puede haber más de un bit de final pero siempre hay un solo bit de inicio. Los bits de paridad son una forma de comprobar si hay errores de una manera muy simple. Casi no es utilizada debido a que hace más lento el tiempo de transferencia.

La transmisión serial consiste en dos cables, Transmisor (Tx) y receptor (Rx) y en modos de transmisión hay dos tipos: full-duplex y half-duplex. El primero se refiere al envío y recepción información al mismo tiempo, en cambio el segundo solo puede realizar una de estas dos actividades a la vez. El modem satelital funciona de manera full-duplex, por lo que es conveniente este tipo de interface para su conexión.

El modo en que el protocolo interpreta o “lee” la información es sencillo. Se hace mediante la lectura de niveles lógicos. Por ejemplo, en el hardware TTL se usan diferencias de potencial de 0 [V], 3.3 [V] hasta 5[V]. Una señal con nivel VCC (3.3 [V] o 5 [V]) puede significar un bit de fin, un bit de valor 1 o simplemente una pulso inactivo de datos. Una señal de 0[V] puede representar un bit de inicio o un valor de 0. Normalmente se estandarizan las señales dependiendo los protocolos usados, así no habrá errores al momento de transmitir. El ejemplo usando lógica TTL aparece en la figura 3.6.

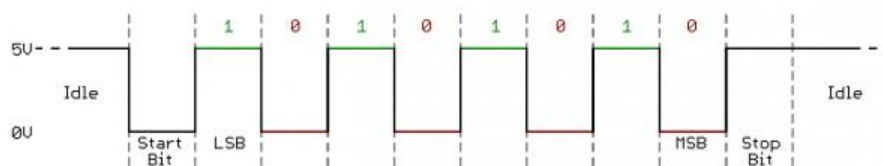


Figura 3.6 Lecturas de diferencias de potencia en TTL [6]

También se usan el estándar RS-232, la diferencia principal son las diferencias de potencial que maneja de van en rango de -15[V] hasta 15[V]. Para hacer mayor hincapié en esta importante diferencia se ilustrara en la figura 3.7.

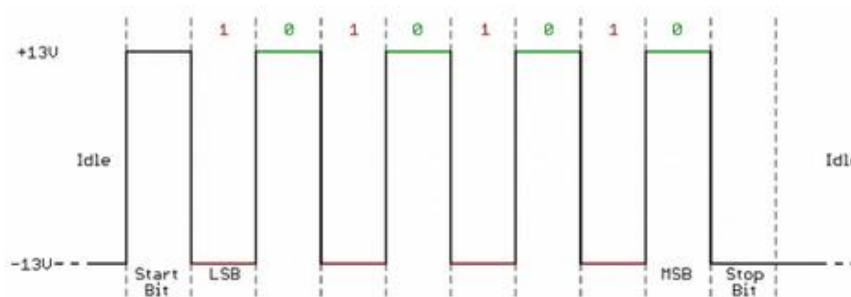


Figura 3.7 Lecturas de diferencias de potencia en el estándar RS-232 [6]

Las diferencias de potencial entre ambos estándares (TTL y RS-232) varían y lo hacen por mucho. No es posible conectar así como así un dispositivo TTL a uno RS-232. Para ello es necesario usar un convertidor de tensiones. Un simple circuito (figura 3.8) o un integrado MAX232 (que no hacen más que invertir las señales con

transistores comunes) donde el más común, bastan para arreglar el problema y poder conectar ambos estándares sin ningún problema. La cuestión aquí radica que el modem satelital tiene un estándar RS-232 (-12[V] a +12[V]) y el Raspberry Pi trabaja con niveles lógicos menores, de la familia TTL. Como se indica con cualquiera de los circuitos se pudo arreglar el problema, y se probó con ellos. Pero al hablar de un modem transmitiendo las cosas cambian y las velocidades de trasmisión son muy importantes.

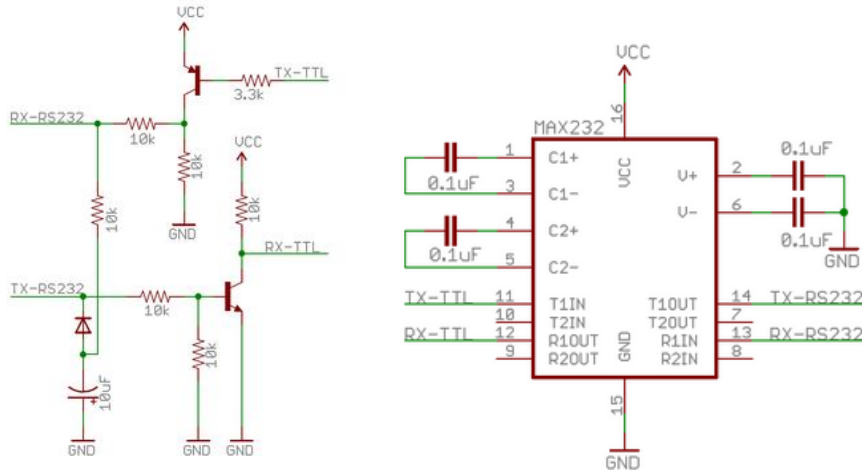


Figura 3.8 Circuitos de enlace entre interfaces RS-232 y TTL [6]

No importa si los niveles en las diferencias de potencias corresponde perfectamente En figura 3.9 se encuentran equivalencias en los niveles lógicos de ambos estándares) si la velocidad no es la adecuada. Con los circuitos e integrados comunes se llega a tener velocidades de hasta 200[Kbps], este número es perfecto para hablar de datos pequeños. Por las velocidades requeridas por el modem, se optó por usar el circuito integrado MAX232 que proporciona velocidades de transferencia de hasta 1[Mbps] (Figura 2.5).

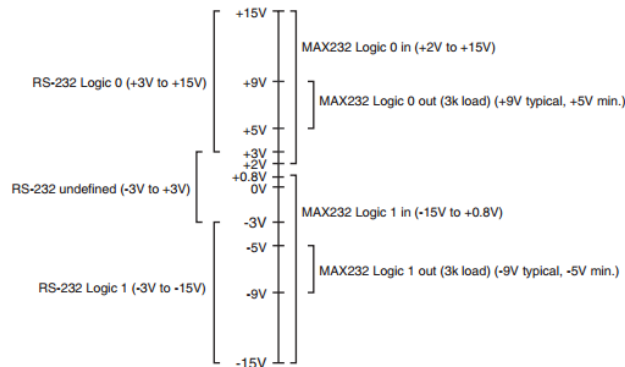


Figura 3.9 Comparación de niveles lógicos en TTL al pasar a RS-232 [6]

Existe un controlador para los datos recibidos en el hardware, que son los pines UART, un bloque de circuitería responsable de la aplicación de comunicación en

serie. Actúa como intermediaria entre las interfaces paralelo y seria. En un extremo del UART (figura 3.10), es un bus de ocho líneas de datos y en el otro son los cables de la comunicación serial (TX, RX).

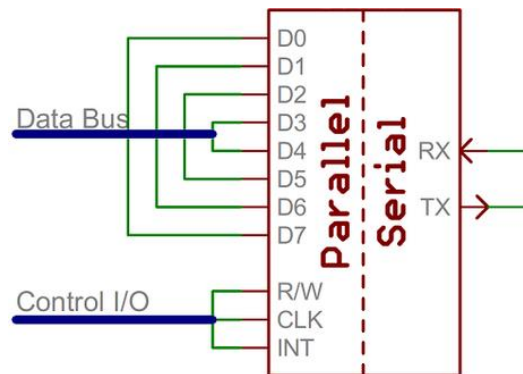


Figura 3. 10 Interfaces del estándar UART [6]

El chip BCM2835 del Raspberry Pi trabaja con mini UART, que son UART de baja velocidad. La velocidad de transmisión depende del reloj del sistema, que está directamente ligado con la velocidad del procesador. Las tasas de transmisión pueden ser altas, hablando de hasta [Mbps], pero sería forzar mucho al procesador, por eso se les clasifica de esa manera. Otra diferencia es que no tiene detección de errores en la trama, no bit de paridad, su bit de inicio y bit de final también es 1. El chip cuenta con la capacidad de tener dos puertos transmisión y de recepción (Tabla 3.1). Se tienen dos RX's (receptores) y TX's (transmisores).

La desventaja más evidente de este tipo de protocolo es el requerir de hardware específico UART en cada extremo, el cual es relativamente complejo y difícil de implementar en conjunto con el software. También se debe de tener un bit de inicio y uno de final, esto representa que por cada 8 bits de datos, habrá dos extra, lo que representa 10 bits en total, al final esto termina por comerse la velocidad de datos final transmitida (throughput).

Otra falla fundamental es que solo se adaptan a una comunicación entre dos dispositivos. Si bien es posible conectar múltiples dispositivos a un solo puerto serie, siempre representa un problema, sobre todo al momento de enviar los datos y colocar las cabeceras en cada tipo de dato. Por último, la tasa de datos es un problema, si bien no hay un límite teórico, las mayorías de las comunicaciones UART solo admiten un cierto conjunto de velocidades de transmisión fijas, y la más alta de ellas, es por lo general alrededor de 23[Mbps].

SPI es un protocolo comúnmente utilizada para enviar datos entre microcontroladores y pequeños periféricos (sensores). Utiliza por separado una

línea de reloj y una de datos, además de una línea para elegir el dispositivo con el que se comunicara. El SPI nace ante el problema de la comunicación asíncrona en la comunicación serial, como no se garantiza que los relojes en ambos sitios (Rx y Tx) estén completamente alineados, puede causar fallar al momento de comunicarse entre ellos. Esto se puede arreglar con los bits de inicio y final, el inconveniente es que las velocidades de muestreo tienen que ser las mismas en ambos extremos del sistema de comunicación, de lo contrario se obtendrá un resultado erróneo. Así se tiene la manera síncrona de trabajar mediante SPI, por ello utiliza líneas de datos y de reloj separadas, esta última mantiene ambos lados en perfecta sincronía. El reloj es una señal oscilante que le dice al receptor exactamente cuándo va a muestrear los datos. El muestreo lee los niveles de lógicos para saber cuándo cambiar de símbolo.

Una de las características importantes es que SPI solo genera la señal de reloj (CLK o SCK para reloj en serie) de un lado, en transmisión o en recepción. Ese lado que genera el reloj se denomina maestro o “master” (muchos de la información con la que se trabaja son anglicismos, se intentará hacer el menor uso de ellos en el presente trabajo, al menos que sea absolutamente necesario utilizarlos) y el otro lado se llama esclavo o “slave”. Siempre hay un solo maestro (casi siempre es un microcontrolador), pero puede haber varios esclavos. Cuando se envían datos desde el maestro al esclavo, se envía en una línea de datos llamada MOSI (Master out/Slave in). Si el esclavo tiene que enviar una respuesta al maestro, el maestro seguirá generando un número preestablecido de ciclos de reloj, y el esclavo pondrá los datos en una tercera línea de datos llamada MISO (Master in/ Slave out), como se ilustra en la figura 3.11.

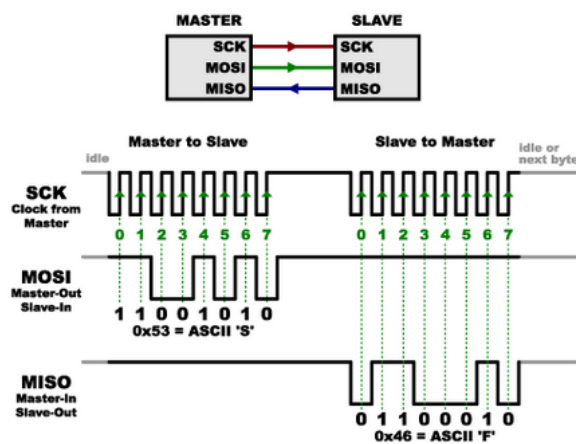


Figura 3.11 Relación maestro-esclavo [6]

SPI es full-duplex, y por lo tanto, en ciertas ocasiones puede transmitir y recibir datos al mismo tiempo. Por último, existe una línea de datos más que hay que tomar en cuenta. SS (Slave Select), que se muestra en la figura 3.12. Esta le

dice al esclavo que debe despertar y comenzar a recibir o enviar datos, también se utiliza cuando hay varios esclavos presentes y se desea seleccionar un en específico para hablar.

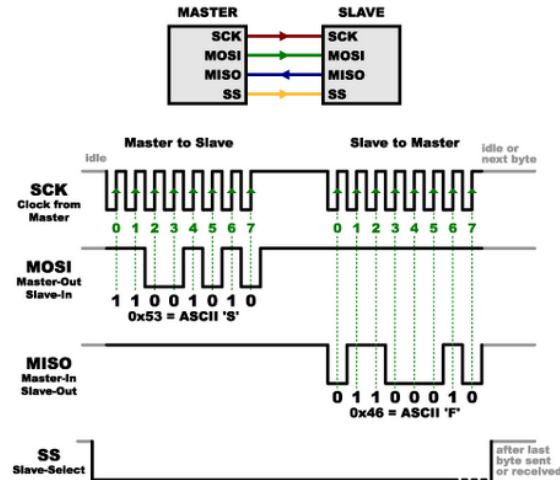


Figura 3. 12 Selección del esclavo [6]

La ventaja de este tipo de interfaz radica en su facilidad de conectar más periféricos en serie (esclavos), y seleccionar al esclavo específico con el que se desee entablar comunicación. En general, cada esclavo necesitará una línea SS por separado. Se hace así para hablar con un esclavo en particular, que va a hacer la línea SS de ese esclavo (activo) y mantener el resto de ellos en estado desactivado (no se quiere dos esclavos activados al mismo tiempo ya que pueden tratar de hablar en la misma línea MISO y el resultado sería datos ilegible).

Muchos esclavos requerirán muchas líneas SS, se tiene el caso de tres esclavos en la figura 3.13, se observa que el numero puede aumentar, dependiendo de las necesidades, claro que tiene su límite.

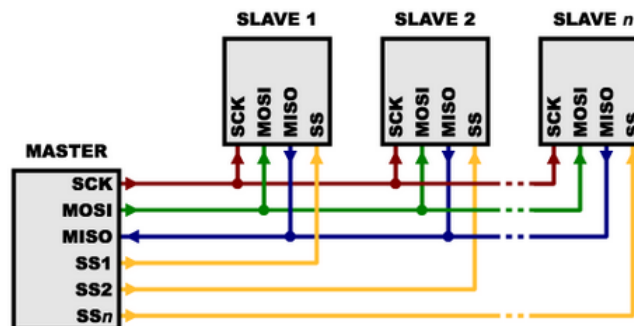


Figura 3. 13 Sistema de 3 esclavos conectados [6]

Por otro lado, también es posible conectarlos en serie, con el MISO (salida) de uno de los esclavos, al MOSI (entrada) del siguiente esclavo. En este caso existe una sola línea SS para todos los esclavos (figura 3.14). Una vez que se envíen todos

los datos, la línea SS se desactiva, lo que hace que todas las demás entradas se activen.

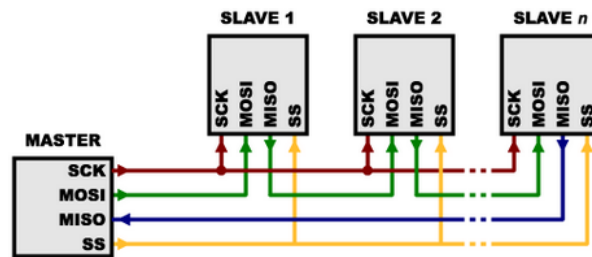


Figura 3. 14 Esclavos conectados en serie [6]

Este protocolo aunque incluye varias ventajas como son: velocidad superior respecto a comunicación asíncrona serial, una alta tasa de transmisión que soporta relojes a velocidades de hasta 10 [MHz] (10 millones de bits por segundo), simplicidad en el hardware receptor para leer los datos, y poder soportar múltiples esclavos, fue con el que menos se trabajó en el proyecto debido a sus desventajas más obvias: requiere muchos cables o líneas de datos que otros protocolos de comunicación, el maestro debe controlar todas las comunicaciones, y que por lo general se requieren líneas SS separadas a cada esclavo, lo que poder ser problemático si se necesitan numerosa cantidad de esclavos (sensores en nuestro caso). Por ello se utilizó más para pruebas de sensores conectados alambreadamente al Raspberry Pi.

Según Michalak (2014) el poder de procesamiento de la computadora no se ve afectado al usar constantemente dos protocolos de comunicación a la vez (SPI y UART).

I2C (Inter-integrated circuit) o muchas veces llamada 2wire, es una interface de bus en serie que soporta múltiples dispositivos y solo requiere de dos cables para la comunicación, a diferencia del SPI, es para trabajar en velocidades relativamente bajas. Permite que varios circuitos integrados (esclavos) se puedan comunicarse con uno o más maestros. Diseñado igualmente que SPI para comunicaciones a corta distancia en un solo dispositivo. Requiere sólo dos cables, como en comunicación serial asíncrona, pero esos dos cables pueden soportar hasta muchos más dispositivos esclavos. A diferencia de SPI, I2C puede apoyar un sistema multimaestro, lo que permite tener más de un maestro para comunicarse con todos los dispositivos del bus (aunque los dispositivos maestros no pueden hablar entre sí a través del bus y debe tomar turnos para hablar).

Las tasas de transmisión decaen en serial asíncrona y SPI; la mayoría de los dispositivos I2C pueden comunicarse a 100 [kHz] o 400 [kHz]. Hay menos sobrecarga de datos con I2C; por cada 8 bits de datos a enviar, un bit más de

señalización (ACK / NACK) debe ser transmitido. El hardware necesario para implementar I2C es más complejo que SPI, pero menos que en serial asíncrona. Puede ser bastante trivial implementarlo en software.

En cada bus I2C existen dos tipos de señales, SCL y SDA. SCL es la señal del reloj, y SDA es la señal de datos. La señal de reloj se genera siempre por el maestro del bus, algunos dispositivos esclavos pueden forzar a alentar el reloj para retrasar el envío de datos del maestro. Esto se conoce como "reloj de estiramiento". Los bits de datos van por SDA, por cada bit de información es necesario un pulso de SCL, así los datos solo pueden cambiar cuando SCL está a nivel bajo. Los datos se transfieren en 8 bits (byte), cada byte enviado requerirá una respuesta de confirmación. A esto se denomina ACK o NACK. El ACK sucede cuando el destinatario, ya sea maestro o esclavo, mantiene SDA a nivel bajo durante un tiempo de bit, si no lo hace entonces es NACK. De esta manera el maestro genera un pulso de SCL. Los mensajes se dividen en dos tipos de trama: una trama de dirección, donde el maestro indica al esclavo que se envía el mensaje, y una o más tramas de datos, que son mensajes como ya se mencionó, de 8 bits. Los datos se colocan en la línea SDA y luego la línea SCL pasa a nivel bajo.

La trama de dirección (figura 3.15.) se inicializa, el maestro deja al SCL en nivel alto y al SDA en nivel bajo. Esto pone a todos los dispositivos esclavos sobre aviso de que una transmisión está a punto de comenzar. Si dos dispositivos maestros desean tomar posesión del bus otra vez, el dispositivo que primero pone al SDA en nivel bajo es el que gana el control del bus.

Como es posible observar, la trama de dirección es siempre la primera en toda nueva secuencia de comunicación. Para una dirección de 7 bits se registra el bit más significativo en primer lugar, seguido por un bit de señalización R/W (Read / Write) que indica si se trata de una operación de lectura (1) o una operación de escritura (0).

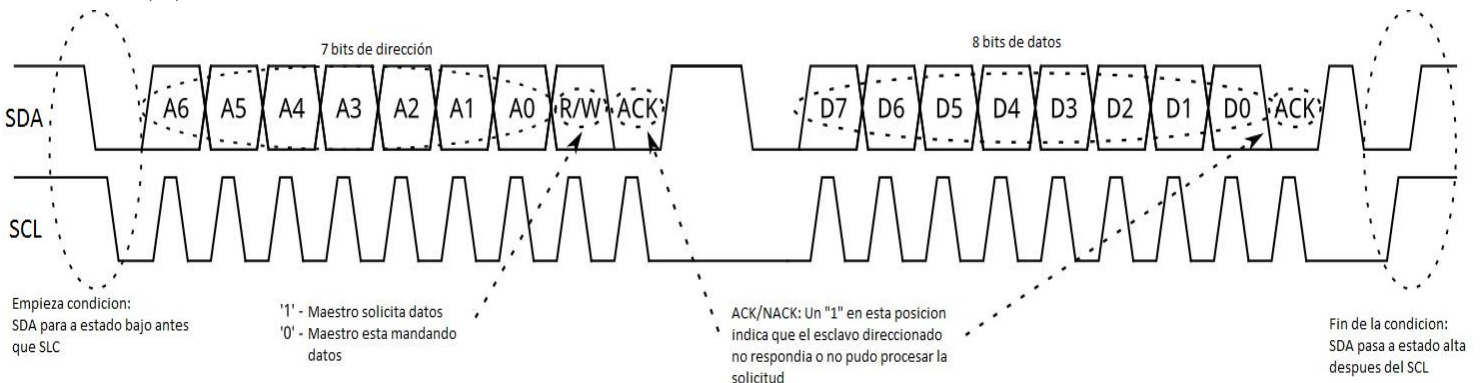


Figura 3. 15 Tramas de dirección y de datos en I2C [69]

Después de que la trama de dirección ha sido enviada, los datos pueden comenzar a ser transmitidos. El maestro tiene la instrucción inherente de continuar generando pulsos de reloj en intervalos regulares, y los datos se colocaran en SDA por el maestro o por el esclavo, dependiendo si el bit R/W indica lectura o escritura. El número de tramas de datos es arbitrario debido a que la mayoría de los dispositivos esclavos tienen la opción de auto incrementar su registro interno, lo que significa la trama subsecuente leerá o escribirá lo que se encuentre en el siguiente registro.

Una vez que todas las tramas de datos han sido enviadas, el maestro generara una condición de alto, para detener el proceso. Las condiciones de parada se definen por un cambio en los niveles lógicos de 0 a 1, o sea de bajo a alto, esto sucede en el SDA después de una transición de 0 a 1 en el SCL, con el SCL restante en nivel lógico alto. Durante el funcionamiento normal de escritura de datos, el valor de SDA no debe cambiar cuando SCL es alta, para evitar condiciones de alto que sean falsas.

La tabla 3.2 presenta un resumen de los protocolos antes mencionados, así como sus características relevantes para este proyecto.

Tabla 3.2 Características de los principales protocolos de comunicación [6]

Interfaz	Formato	Numero de dispositivos máximos a conectar	Distancia máxima de alcance [Ft]	Velocidad máxima de transición [bps]	Uso típico
RS-232	Serial asíncrono	2	50-100	20[Kbps]	Modems, comunicación básica
Ethernet	Serial asíncrono	1024	1600	10 [Gbps]	Comunicaciones de redes computacionales
I2C	Serial síncrono	40	18	3.4[Mbps]	Comunicación entre microcontroladores
SPI	Serial síncrono	8	10	2.1[Mbps]	Comunicación entre microcontroladores
USB	Serial asíncrono	127	16	1.5[Mbps], 12[Mbps], 480[Mbps]	Periféricos de PC

3.1.2 Hardware y software utilizado

Como se planteó en el capítulo dos (figura 2.8), se realizarán diferentes conexiones para conectar los diferentes sistemas del proyecto. Para dicha conexión de interfaces se utilizarán cables como principal medio de transmisión. La tabla 3.3 describe los periféricos y cables utilizados para todo el proyecto.

Tabla 3.3 Listado de interfaces a utilizar

Conexión	Medio de transmisión	Imagen
Raspberry Pi - Modem satelital	Cable RS-232	
Raspberry Pi -Nodo sensor de la red de sensores	Cable FTDI- USB	
Raspberry Pi- Fuente de alimentación	Cable de corriente mini USB	
Raspberry Pi- Monitor LCD para visualizar proceso	Cable HDMI	
Raspberry Pi- Red de datos	Cable Ethernet	
Raspberry Pi-GPS	Cable USB tipo B	

El puerto de conexión Ethernet es la principal entrada que tiene una computadora para comunicarse con otros dispositivos a internet. En este caso dicho puerto funciona con detección automática que significa que puede estar conectado a un router o directamente a otro ordenador. Con el cable conectado, la

computadora central recibirá automáticamente detalles que necesita para acceder a internet a través de DHCP, asignándole dirección IP y señalando la puerta de enlace predeterminado que se tiene para acceder al modem, como el caso de Chaklander, Alam, Islam y Sabbir (2013) que implementaron un servidor web de bajo costo con la misma tecnología se utiliza en este prototipo.

Milenkovic, Markovic, Jankovic y Rajkovic (2013) argumenta la decisión de utilizar un sistema serial RS-232 para adquirir datos y de la misma manera enviar mediante Ethernet la información recabado, si es cierto que el objeto de estudio es bastante diferente (enzimas biológicas), el objetivo no cambia y por lo tanto es relevante seguir invirtiendo recursos en experimentar y más importante aún, en perfeccionar esta tecnología.

Otra de las ventajas de estar conectados a la red de internet que el propio Raspberry Pi puede fungir como sitio web estático, también puede generar contenido dinámico utilizando las bases de datos y aplicaciones web. Las ventajas de servidores web se han contado en experimentos externos (Tso, White, Jouet, Singer y Pezaros, 2013) donde la potencia del Raspberry Pi se pone a prueba para toda una red de servidores cuyo fin es proporcionar servicio en la nube. Además, incluso puede facilitar el acceso a los puertos GPIO a través de tecnologías web.

3.2. Sistema de comunicación satelital

Las definiciones más básicas de satélite lo nombran como un objeto físico que orbita alrededor de un cuerpo celeste. Por ejemplo, la Tierra y los otros planetas son astros que giran alrededor del Sol, una estrella solar, y al mismo tiempo la luna es un satélite natural de la Tierra. El principio básico menciona un equilibrio entre la inercia del satélite que gira y la atracción gravitacional del cuerpo orbitado mantienen al satélite en su órbita. Así es como surgen los satélites artificiales. Con la posibilidad de poner en órbita satélites artificiales con diferentes propósitos. Una de las aplicaciones principales son las telecomunicaciones.

3.2.1. Satélites. ¿Qué son y para qué?

En esta tesis no se toma en cuenta el término de satélite natural y cada vez que se hable de un satélite se referirá a un satélite artificial. Un satélite puede definirse como un vehículo fabricado por el hombre que se encuentra orbitando la Tierra. Al tener este elemento en el espacio surgieron las comunicaciones por satélite cuyo principal objetivo es lograr cada vez mayores rangos y capacidad con los más bajos costos posibles.

Los satélites brindan ciertas características que otros sistemas de comunicación no tienen. Poseen capacidades únicas para proporcionar cobertura en grandes áreas. Algunas redes de satélites poseen la habilidad de interconectividad entre nodos geográficos extensos, por ejemplo, las centrales telefónicas de diferentes países se conectan mediante satélites. La conectividad directa entre usuarios se observa en teléfonos móviles o en emisiones de televisión y de radio. En muchas áreas el proveedor de servicios puede usar infraestructura existente adicional (por ejemplo fibra óptica) para proporcionar las mejores soluciones que converjan en costo y rendimiento. Un ejemplo particular es la empresa Iridium la cual era un fracaso espectacular desde el punto de vista de negocios, pero es uno de los sistemas más capaces jamás diseñados e implementados para comunicaciones móviles globales.

El modem satelital utilizado en el proyecto tiene el propósito de funcionar como parte de una constelación satelital de comunicación del tipo MSS. Trabaja en conjunto con satélites ubicados en órbita baja terrestre y funciona mediante comunicación asíncrona o por paquetes. El conocer cómo funcionan las redes satelitales es de vital importancia en el presente trabajo porque dan la causa y la consecuencia para desarrollar este tipo de tecnología, por ellos se debe de conocer que hay detrás de ellos de una manera general, solo así se puede entender el porqué de ciertos parámetros implícitos en el modem satelital.

Entendiendo el ambiente que rodea al desarrollo satelital es la única manera en la que realmente se podrá desarrollar conocimiento aplicado que ayude a mejorar el ya existente. Poner en contexto el trabajo es la base de la construcción científica.

Fue Arthur C. Clark quien se dio cuenta del potencial de los cohetes V2 como un medio para poner en marcha un sistema geoestacionario para proveer servicios de difusión en todo el mundo. Desde entonces hasta la fecha han ido en constante evolución las comunicaciones y los satélites.

Los satélites tienen una visión amplia, que es especialmente útil para otros usos que las comunicaciones como son meteorología, comunicación, navegación, astronomía, monitoreo terrestre y usos militares.

Existen varias maneras de clasificar a los satélites. Por el tipo de órbita, por el tamaño, o por la aplicación. En la parte del tipo de órbita se encuentran clasificaciones dependiendo su forma (elíptica, circular), inclinación (polar, ecuatorial, inclinada) o altura (órbita baja, órbita media o geoestacionarios).

Otra manera es por medio del tamaño, normalmente está directamente ligada al tipo de aplicación que desempeñara, debido a que un satélite muy pequeño no puede llevar un gran sistema de comunicaciones, o sería un gasto de recursos innecesarios implementar experimentos que requieren pocos recursos en un satélite de gran tamaño. En general (varía dependiendo de la literatura) se pueden ordenar de la siguiente manera: Picosatélites menores a 1 [Kg], nanosatélites entre 1 y 10[Kg], microsátélites entre 10 y 100 [Kg], minisatélites entre 100 y 500[Kg], satélites mediano entre 500 y 1000[Kg] y macro satélites más de 1000[Kg]

Según Rodolfo Neri Vela, un satélite tiene varios subsistemas, los cuales en conjunto se encargan del correcto funcionamiento de un aparato tan complicado como es el satélite. Dichos subsistemas son:

- Antenas: Se encarga de recibir y transmitir las señales a las zonas de cobertura disponibles del satélite.
- Comunicaciones: Multiplexean, amplifican y procesan las frecuencias de las señales para entregarlas a las antenas.
- Energía eléctrica: Proporciona electricidad (corriente y diferencia de potencial) a todos los subsistemas del satélite.
- Control térmico: Regula la temperatura del satélite.
- Posición y orientación: Estabiliza y orienta al satélite para que siga la trayectoria orbital correcta.
- Propulsión: Suministra los incrementos de velocidad para mantener y corregir la posición.
- Telemetría: Enviar y recibir información del centro de control en la tierra para corroborar el correcto funcionamiento del satélite.
- Estructural: Soportar a todo los equipos y proporcionar la rigidez necesaria para controlar el satélite.
- Carga útil: Todo lo que se usara en la misión espacial que no sea primordial para el funcionamiento del satélite, excepto claro, los tripulantes.

Ahora que se tiene una idea más específica que es un satélite, para que sirva y como se clasifican, se enfocara este texto en las aplicaciones de comunicaciones móviles.

Los sistemas satelitales de comunicaciones han evolucionado constantemente por ello es de gran ayuda conocer el contexto y marco histórico. En la tabla 3.4 se presenta una breve síntesis de los acontecimientos más relevantes en materia de sistemas de constelaciones satelitales.

Tabla 3.4 Eventos históricamente importantes en el desarrollo de satélites de comunicación
[10][11][23][25]

Año	Acontecimiento	Responsable
1957	Lanzamiento del primer satélite	Unión Soviética
1965	Lanzamiento del primer satélite en órbita geoestacionaria	Comsat,USA
1976	Primer demostración de un enlace intersatelital	Satélites LES-8
1977	Experimentos de comunicaciones móviles usando ATS-6	NASA
1978	Introducción del sistema de posicionamiento global (GPS)	Departamento de defensa de los EUA
1982	Introducción de GLONASS	Unión soviética
1982	Primeros sistemas civiles de comunicaciones móviles	Inmarsat
1987-1989	Se propone arquitectura para comunicaciones por satélite en órbita baja (LEO)	Universidad de Surrey
1990	Primer sistema comercial de emisión de radio satelital	CD Radio
1990	Primer sistema comercial portátil no geoestacionario es anunciado	Motorola/ Iridium
1990	Servicios satelitales móviles(MSS) son introducidos	Inmarsat
1992	Sistema GSM es introducido	Europa
1992	Primer sistema satelital comercial tipo Little-LEO	ORBCOMM
1994	Primer servicio tipo FSS en órbita no geoestacionaria	Teledesic Corporation
1994-1996	Anuncio de un sistema “súper geoestacionario” que cubrirá múltiples regiones	Agrani, APMT y ACes Thuraya
1996	Servicios de búsqueda son introducidos	Inmarsat
1997	Terminales móviles de tamaño escritorio son introducidas	Inmarsat
1997	Primer sistema no geoestacionario en órbita Little-Leo es presentado	ORBCOMM
1997	Lanzamiento del primer lote de constelación satelital para comunicación de datos de baja velocidad en orbita Little-LEO	ORBCOMM
1997	Utilizando tecnología satelital de comunicaciones avanzadas	NASA
1997	Disponibilidad en capacidad de superposición en órbita geoestacionaria para sistema de navegación	Inmarsat
1997	Comienzo de utilización de haz puntual a nivel mundial para comunicaciones MSS	Inmarsat
1998	Introducción del primer sistema de satélites en big-LEO	Iridium
1998	Introducción de los teléfonos satelitales combinando comunicación vía satelital y terrestre	Iridium
1999	Introducción de los satélites de procesamiento a bordo para MSS	Iridium
1999	Introducción de Globalstar	Globalstar
2005	Introducción de sistema de comunicaciones satelital de banda ancha móvil terrestre	Inmarsat
2006-2008	Ampliación del sistema de banda ancha portales para plataformas móviles	Inmarsat
2010	Anuncio de los sistemas de próxima generación	Inmarsat, Iridium, , Globalstar
2009	LightSquared propone servicios de tecnología avanzada en comunicaciones en USA	LightSquared
2014	Introducción de comunicaciones de banda ancha de alta gama en MSS provistos en banda Ka	Inmarsat

Las redes satelitales tienen dos tipos de interfaces externas para poder comunicarse: una es entre el satélite y las terminales de los usuarios finales (UES) y

otro es entre el satélite y las estaciones terrenas (GES), observadas en la figura 3.16. Ambos tipos de interfaces de conexión tienen la misma finalidad, que es establecer una comunicación entre la constelación completa de satélites y los usuarios. La diferencia entre las interfaces es el hardware y software empleados para poder procesar la información, la parte de UES generalmente requiere mayor sofisticación. Actualmente los sistemas de procesamiento digital de señales (DSP) son muy avanzados y esto se logra sin problemas. El hablar de DSP hace referencia a las últimas capas (sesión y aplicación) del modelo de referencia OSI.

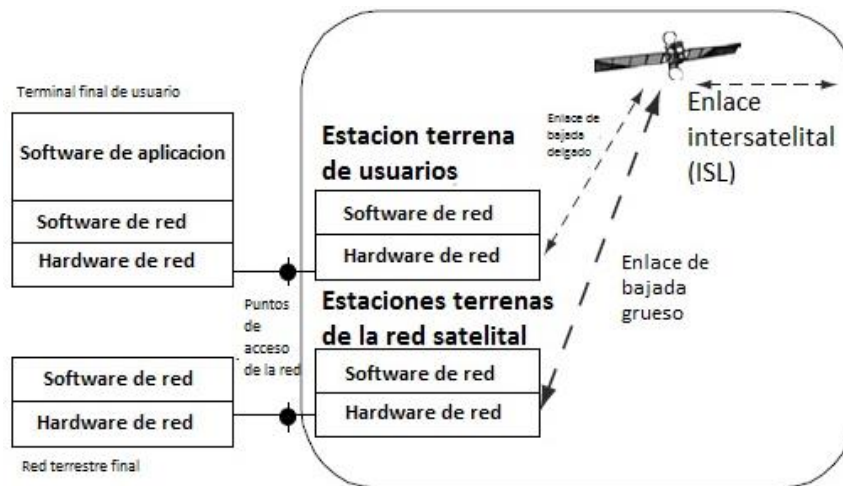


Figura 3. 16 Interfaces externas de los satélites [53]

El modem satelital se encuentra dentro de la clasificación GES por ser un dispositivo pensado para trabajar en tierra.

Se tiene acceso a una constelación satelital la cual proporciona un enlace de bajada, pero el enlace satelital varía en ancho de banda (BW) dependiendo el destino. Si es un enlace a una estación terrena se tendrá un mayor ancho de banda en comparación con un enlace de destino directo a usuario. En lugar de tener una estación terrena en donde recibir la información del satélite, se cuenta con el conjunto de modem satelital-Raspberry Pi que cuenta con un ancho de banda limitado. De ahí se tendrá el punto de acceso a la red del CAT, y podrá enviarse información a las terminales finales de usuario, que será cualquier computadora con acceso a internet que se encuentre en el CAT.

El hardware corresponde a la capa física, el software a la capa dos y tres, enlace de datos y de red respectivamente. El software de aplicación a las capas de la cuatro a la siete.

La tecnología en la transmisión de datos en redes satelitales pueda variar, se puede configurar para conexiones punto a punto, punto a multipunto, o multipunto a multipunto.

Los servicios proporcionados por satélites y avalado por la ITU son:

- *FSS (Fixed Satellite Services)*: Usados para comunicación entre puntos fijos en tierra.
- *MSS (Mobile Satellite Services)*: Usado para comunicaciones con terminales móviles.
- *BSS (Broadcast Satellite Services)*: Usado por los servicios de emisión de televisión y de radio.

FSS es un servicio de radiocomunicación (figura 3.17) entre posiciones dadas en la superficie de la tierra. Las estaciones terrenas de este tipo de comunicación varían dependiente de la aplicación para que sea destinado el servicio, por ejemplo, para emisiones internacionales se usan antenas de alta cobertura, de unos 11 a 30 [m] de diámetro con un robusto sistema RF, para uso de consumidores pequeños o móviles, se usan antenas de 0.5 a 3 [m] de diámetro y obviamente no cuentan con antenas para el seguimiento del satélite.

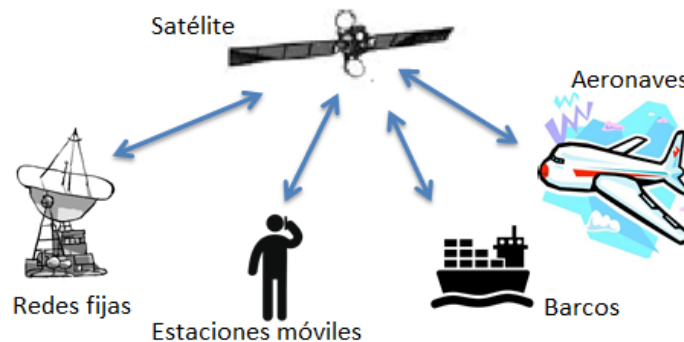


Figura 3. 17 Utilización de servicios FSS

En este tipo de comunicación es muy usual el uso de enlaces intersatelitales o ISL (Inter-satellite link), cuyo principal objetivo es reducir el segmento de tierra en las conexiones de una red satelital. Sin ISL, el número de estaciones terrenas aumentaría al tener que vincular más satélites en conjunto. Especialmente para satélites en LEO y GEO, que son constelaciones donde se encuentran en continuo movimiento a través del espacio.

Se sabe que la topología de una red satelital cambia también con el movimiento de la constelación. Si las posiciones de los satélites son “relativamente” estables, se puede vincular las constelaciones de satélites entre sí para formar una

red de satélites mucho mayor. Esto se puede traducir en beneficios como cobertura. Y reducción de estaciones terrenas como se observa en la figura 3.18.

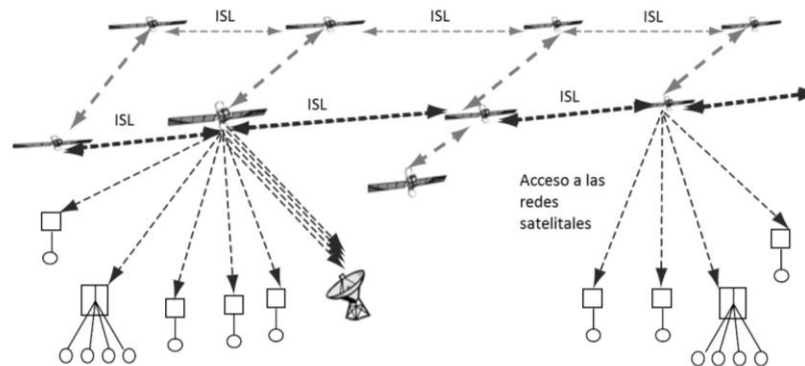


Figura 3.18 Uso de enlaces intersatelitales [53]

También es posible considerar como otra ventaja de usar ISL, la comunicación directa entre satélites siempre y cuando estos tengan línea de vista disponible, así se disminuye el tráfico generado entre el segmento de tierra y el del espacio, aunque esto requiere un mayor grado de sofisticación en ruteo y en procesamiento de la información. Un ISL es una clasificación a parte, no entra en el enlace de subida (uplink) y en enlace de bajada (downlink) de un satélite.

Un MSS (figura 3.19) se define específicamente como un servicio de radiocomunicaciones entre estaciones terrenas móviles y uno o más satélites. Se cataloga en esta clasificación los servicios aeronáuticos, marítimos o terrestres. Por cuestiones de movilidad, las terminales terrestres móviles son comúnmente pequeñas y algunas son terminales de mano.

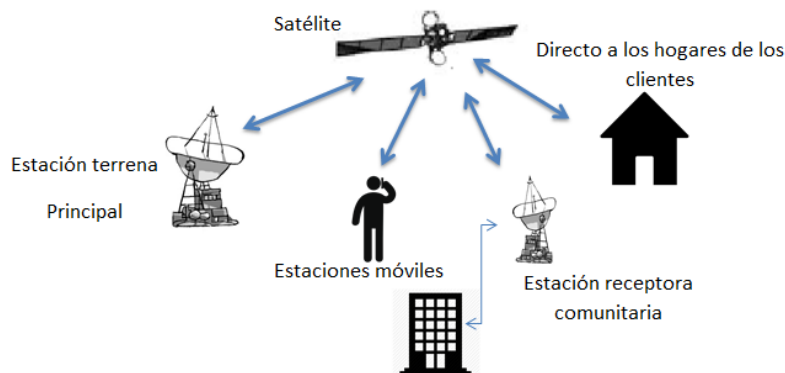


Figura 3.19 Esquema básico de comunicación en MSS

Un BSS es un servicio de radiocomunicación en el cual las señales emitidas o retransmitidas por satélite están destinadas a la recepción directa por los usuarios

de TV que utilizan una antena tipo TVRO. Los satélites usados para este tipo de comunicación son llamados satélites de transmisión directa o DBS por sus siglas en inglés.

Las constelaciones satelitales constan básicamente de dos partes: el segmento espacial y el segmento terrestre (figura 3.20). Como su nombre lo indica, el segmento terrestre está conformado por las estaciones receptoras de los usuarios, tienen un transmisor RF o receptor, amplificador de potencia o de bajo ruido, dependiendo si se trata de un enlace de subida o de bajada. Y equipo encargado de modular y de retransmitir la información a una ubicación específica en tierra. Las estaciones terrenas pueden estar conectadas a diferentes tipos de redes para ofrecer un amplio abanico de servicios.

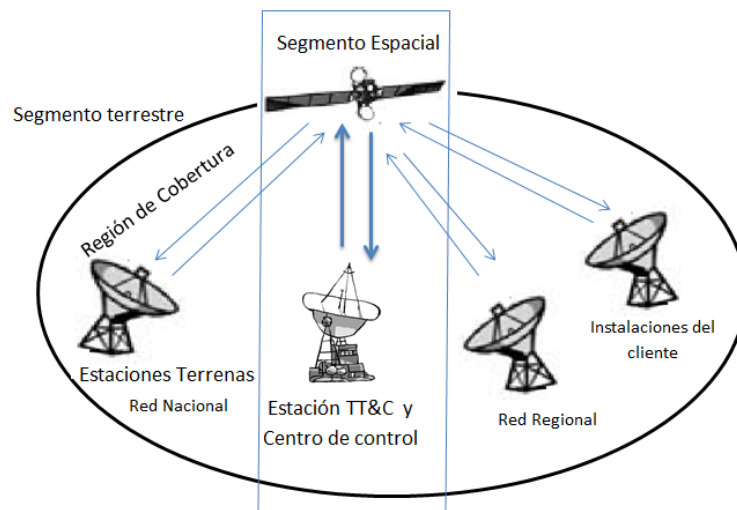


Figura 3. 20 Segmento espacial y terrestre [53]

El segmento espacial está conformado por el satélite en su órbita adecuada, las correspondientes estaciones terrenas que son el comando primario de seguimiento y control del satélite, estaciones de mantenimiento y el equipo necesario para el control, telemetría y rango (TT&C). Todas las estaciones tienen como objetivo monitorear el estado de salud del satélite, en otras palabras, que todo marche bien allá arriba y que cumpla su misión. En la práctica, la mayoría de los satélites de comunicación están respaldados por uno o más satélites.

En el caso específico de servicios MSS, el segmento terrestre consiste en diferentes tipos de terminales móviles conectadas a nodos de redes fijas vía satelital. El tamaño de las antenas varía dependiendo del tipo de comunicación móvil que se maneje (aeronáutica, con base en tierra, marítima) y del ancho de banda que se necesite.

Una red de satélites puede tener alguno de cientos a miles estaciones terrestres, por lo que se tuvo la necesidad de encontrar esquemas de acceso para equilibrar el ancho de banda y la potencia para una aplicación determinada. Las redes satelitales crecieron a sistemas tan completos como se muestra en la figura 3.21, una arquitectura de una constelación satelital de servicios móviles satelitales.

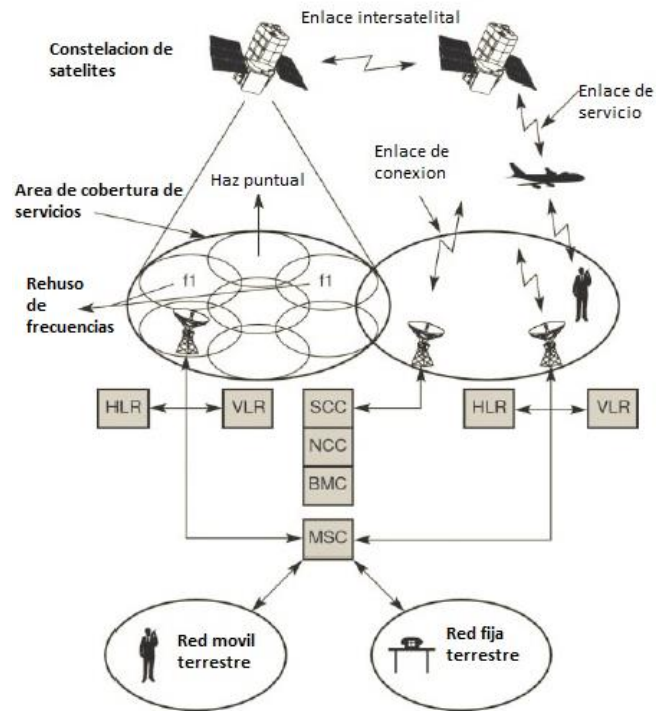


Figura 3. 21 Red de servicios satelitales [59]

El NCC gestiona el flujo de tráfico, establecimiento de la llamada y la liberación, los recursos de radio y señalización. Al recibir una petición de llamada, el administrador de recursos de radio asigna el recurso de radio deseada por la duración de la llamada. El registro de posiciones base (HLR) y el registro de posiciones de visitantes (VLR) son responsables de la gestión de la movilidad. El centro de conmutación móvil (MSC) gestiona la conmutación de llamadas entre la red fija y la red móvil. Un centro de gestión empresarial (BMC) se encarga de la facturación, nuevas activaciones, atención al cliente, y otras funciones relacionadas con el negocio. El satélite tiene sus transpondedores para recibir las señales de la antena de transmisión en la estación terrena. Los satélites modernos cuentan con un sistema de procesamiento abordo (OBP) y multiplexadores abordo (OBS), que sirven para hacer el procesamiento de las señales y canalizarlas a la estación terrena o satélite asignada. En un sistema N GEO, una llamada (o una sesión de datos) en curso, debe ser redirigida a otro satélite para mantener la conectividad del haz puntual, debido al constante movimiento de los satélites y a una posible pérdida de

visibilidad. La empresa Globalstar, proveedora del modem satelital, utiliza este tipo de arquitectura.

Las orbitas (figura 3.22), son de vital importancia como un recurso al momento de diseñar una constelación satelital, cada orbita tiene ventajas y desventajas.

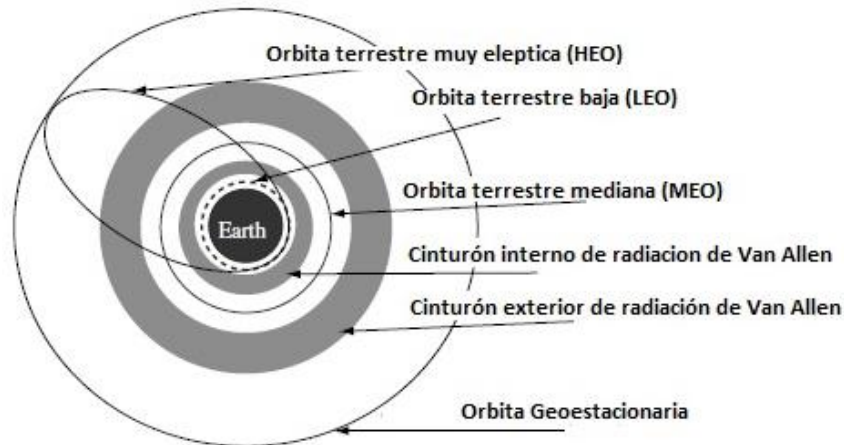


Figura 3. 22 Diferentes tipos de orbitas satelitales terrestres

Para órbita geoestacionaria es necesario tener tres satélites posicionados a 35786 [Km] para una cobertura global de la tierra. Independientemente de su inclinación, un satélite en esta orbita siempre será geosíncrono. Si, además tiene una inclinación y su excéntrica es de 0° , se trata de una órbita circular alrededor de la tierra, y es un satélite en órbita geoestacionaria. Todos los satélites que no sean geoestacionarios (LEO, HEO, MEO) se les denominada N GEO.

Orbitas geoestacionarias tienen ventajas como un buen establecimiento para las comunicaciones, los satélites están fijos respecto a las estaciones terrestres. Minimizan los requisitos de la estación TT&C. Las pérdidas de trayectoria son invariantes al existir el mínimo de efecto Doppler y la interferencia es predecible y su cobertura es adecuada para la mayoría de las áreas pobladas debido a su latitud de $\pm 76^\circ$.

Entre las principales desventajas que ofrece se encuentran un significativo retardo de la señal (de 250 [ms] a 500 [ms]), no tienen cobertura en latitudes muy altas (grandes regiones de Europa), especialmente para las comunicaciones móviles esto representa un gran inconveniente. Sus ángulos de elevación son bajos, tiene mala calidad de la señal debido a grandes pérdidas de propagación.

Los satélites de órbita baja LEO (low earth orbit) están en un rango de altitud generalmente de 500[Km] a 1500[Km] pero algunos casos especiales conocidos como big LEO, y pueden llegar a menos de 5000[Km]. El periodo del satélite es de 95 a 120 [minutos]. Su principal objetivo es garantizar una alta elevación para cada lugar de la tierra, así se podrá proporcionar un enlace de comunicación de alta calidad. Cada satélite solo es visible desde la tierra por alrededor de diez minutos. Los servicios en esta órbita son utilizados principalmente en la teledetección y la prestación de servicios de comunicaciones móviles (debido a una menor latencia). Una fortaleza en este tipo de órbita es el poder usar una transmisión inferior para la comunicación de voz, así permite atender a terminales móviles con antenas omnidireccionales utilizando potencia de transmisión baja, lo cual se traduce básicamente en un ahorro de energía. También tiene un haz puntual que abarca un área más pequeña permitiendo una mejor reutilización de frecuencias. Las constelaciones en órbita baja terrestre pueden proporcionar mucha mayor elevación en las regiones polares y así una mejor cobertura mundial.

Los inconvenientes en LEO existen, afortunadamente no son tan significativos. Uno de ellos es la necesidad de tener muchos satélites (más de 40) para una cobertura mundial, aproximadamente más de 50. Tienen corto tiempo de visibilidad con una alta elevación, es necesario tener enlaces cruzados complejos y capacidad en el sistema de tx y rx de los satélites para pasar la señal a otro satélite visible. Los satélites tienen unas vidas más cortas debido a la resistencia atmosférica y radiación del cinturón interior de Van Allen. Por último es necesario un enrutamiento de paquetes de datos de satélite, si los usuarios quieren comunicarse alrededor del mundo. Esta última se puede traducir en ventaja dependiendo de la aplicación.

Es necesario que los satélites en órbita baja terrestre sean desplegados en constelación de varios, debido a que un solo satélite proporciona un haz puntual pequeño que cubre un área chica y se mueve a una alta velocidad angular. Esto puede reducir los problemas de retardo y la pérdida, pero introducir una mayor complejidad en el mantenimiento de enlaces de comunicación entre las estaciones terrenas y satélites debido al movimiento rápido de los satélites.

Satélites en órbita terrestre media se localizan entre los satélites LEO y GEO, con periodos de 6 a 8 horas, y a una altitud de 5000 a 20 000 [Km]. Este tipo de constelación solo requiere de doce satélites para una cobertura mundial, lo cual es mucho más que en un sistema geostacionario, pero es menor que en un sistema

LEO. Dependiendo de la inclinación se puede cubrir grandes poblaciones, por lo que se requieren menos traspasos de señales entre satélites para conservar la señal.

La evolución de sistema para órbita terrestre baja (LEO) ha dado lugar a una serie de sistemas satelitales globales propuestos, que pueden agruparse en tres tipos distintos: Little LEO, Big LEO, y LEO de banda ancha. La tabla 3.5 muestra las principales características de los sistemas.

Tabla 3.5 Principales sistemas LEO

	Little LEO	Big LEO	LEO de banda ancha
Sistemas actuales	ORBCOMM, Starsys	IRIDIUM, Globalstar, ICO	Teledestic
Aplicaciones típicas	E-mail, fax	Teléfonos de voz, transmisión de datos a baja velocidad	Acceso a internet, transmisión de datos a alta velocidad, multimedia.

Entre las principales desventajas de los sistemas en MEO, está el retraso en la señal (70-80[ms]) debido a la larga distancia que existe entre los satélites y la tierra, y el satélite requiere mayor potencia para poder transmitir, en especial las antenas para pequeñas áreas.

Los satélites también se pueden clasificar por su tipo de órbita. Esto está directamente relacionado con el periodo de un satélite, que es cuánto tiempo tarda en recorrer su órbita. Por lo tanto, está vinculado con la velocidad con la que se mueve respecto a la tierra y su altitud. El periodo de un satélite en órbita circular está definido por la siguiente ecuación

$$T^2 = 4\pi^2(R + h)^3/\mu$$

Donde R es el radio de la tierra, h= la altitud del satélite y μ el parámetro de gravitación universal = 398 600.5 [km³]. En la figura 3.23 se presenta la relación entre la altitud y el periodo de los diferentes sistemas satelitales.

De las gráficas se concluye que en LEO el periodo es mucho menor, análogamente es evidente que la velocidad del satélite aumentara conforme la altitud de la órbita sea menor.

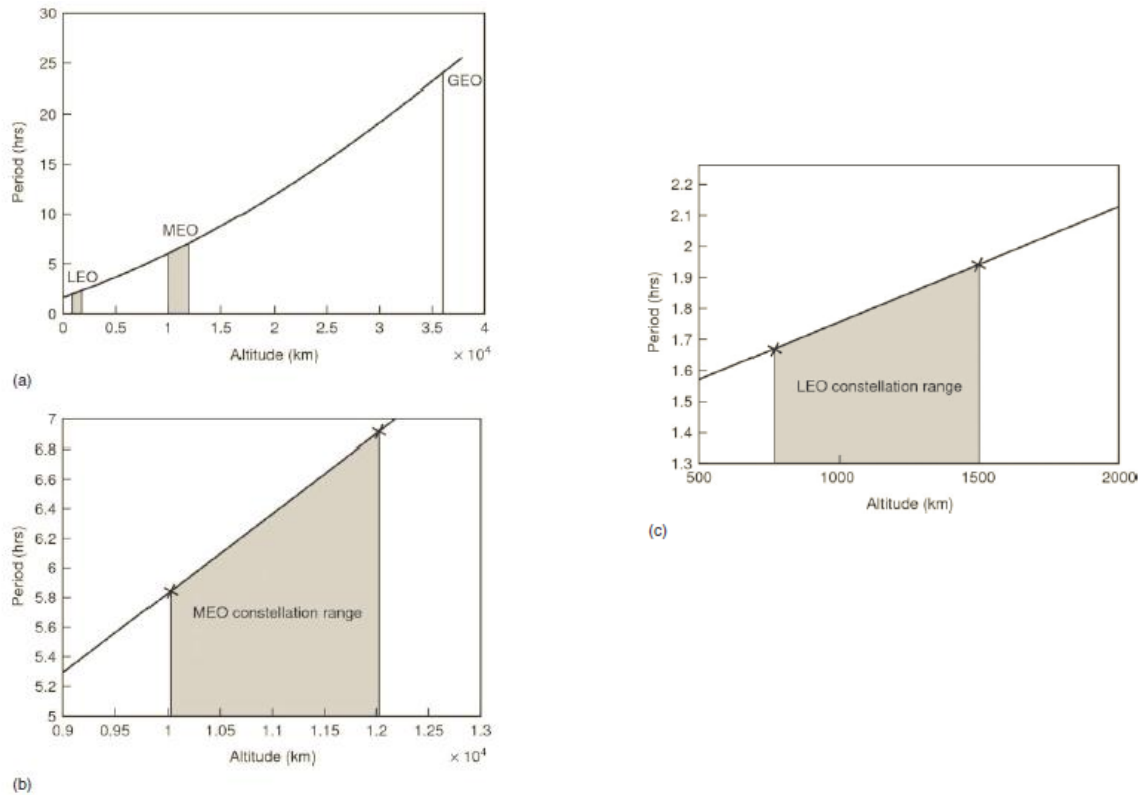


Figura 3.23 a) Relación de altitud contra periodo de los satélites en orbitas LEO, MEO y GEO. b) Relación en órbita MEO. c) Misma relación para orbita LEO [53]

En resumen se comprueba que el periodo es inversamente proporcional a la velocidad del satélite.

La tabla 3.6 presenta un resumen de las características más importantes de las constelaciones satelitales.

Tabla 3.6 Características de orbitas de satélites artificiales [53][35][36][61]

Característica	Orbita GEO	MEO	LEO
Numero de satélites para cobertura mundial	3-4	10-12	40-300
Cobertura regional	Bien adaptado	Se necesita diseño orbital específico	Se necesita diseño orbital específico
Limitaciones de cobertura	Entre latitud de $\pm 76^\circ$	Ninguna	Ninguna
Complejidad y costo en operación	Baja	Media-alta	Alta
Vida aproximada del satélite	15	7-10	5-7
Retardo en transmisión [ms]	250	55-80	3.5-15

Todas las señales se transmiten a través del aire, por medio de ondas en diferentes frecuencias. Las frecuencias conocidas por el hombre están catalogadas en el espectro electromagnético (figura 3.24). Desde las luz visible, hasta rayos gamma. La parte del espectro que sirve para las radiocomunicaciones se llama espectro de radiofrecuencia y a la vez se divide en ciertas bandas de frecuencia. Las bandas de frecuencia para satélites de comunicación son las UHF, SHF Y EHF. Las principales bandas usadas en satelices están en la tabla 3.7.

Tabla 3.7 Bandas usadas en comunicaciones con satélites [38]

Banda	Rango de frecuencias
HF	3 -30 [MHz]
VHF	30-300[MHz]
UHF	300-1000[MHz]
L	1-2 [GHz]
S	2-4[GHz]
C	4-8[GHz]
X	8-12[GHz]
Ku	12-18[GHz]
K	18-27[GHz]
Ka	27-40[GHz]
V	40-75[GHz]
W	75-100[GHz]
mm	110-300[GHz]

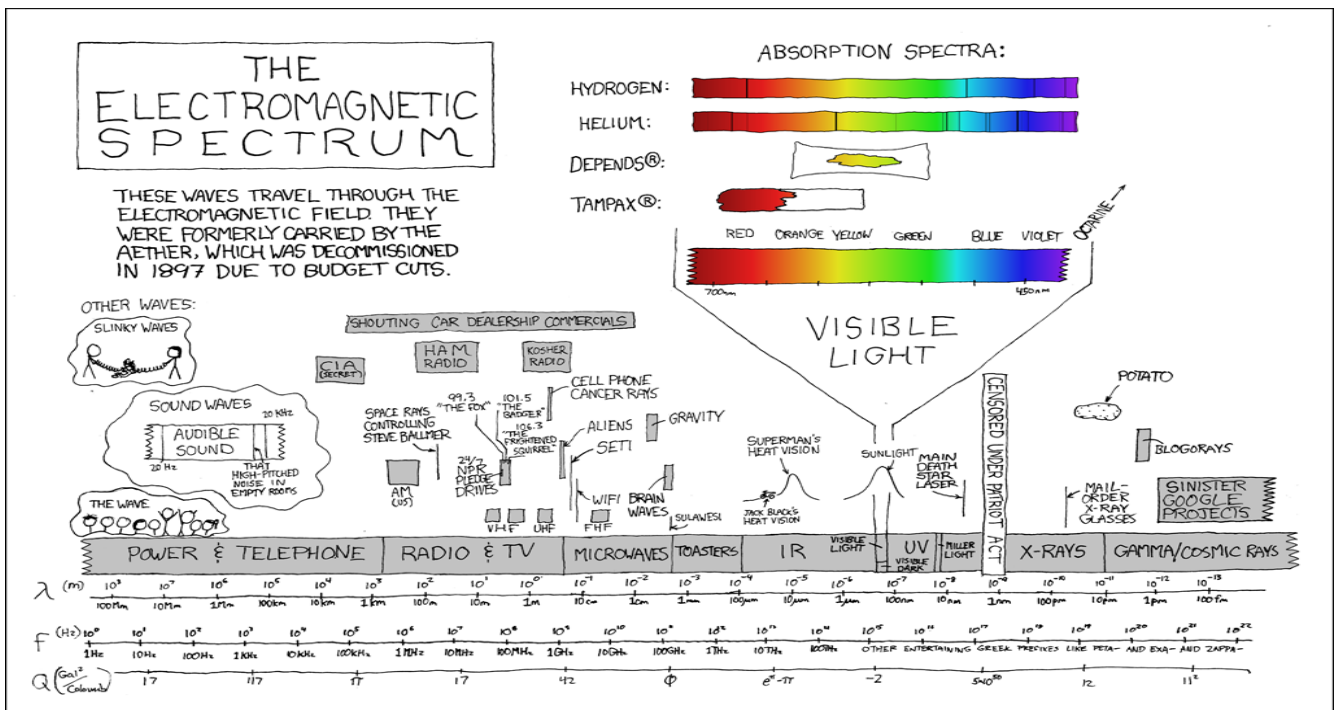


Figura 3.24 Espectro electromagnético [86]

Las modulaciones son parte necesaria debido a que el modem y la red inalámbrica de sensores las utilizan, si bien no se puede cambiar ese aspecto, se puede adaptar con una mejor codificación para hacer más robusto el enlace, evitando los imperfectos que tenga la modulación en uso.

La modulación es el proceso de difusión de información de banda de base a una portadora. Las señal de banda base son las señales de la portadora de información, voz, datos, video, etc. transmisión. Básicamente, modulación es el proceso en el que algunas características (fase, amplitud, frecuencia) de la portadora a alta frecuencia se varían de acuerdo con la señal de banda base.

Los debates se centrarán en los aspectos a nivel de sistema de modulación debido a la extensión de este tema. Un requisito fundamental para la transmisión eficiente en los sistemas de RF es que la antena debe acercarse a la longitud de onda de la señal. No se pueden construir antenas de ese tamaño. La solución más práctica es traducir la señal a una frecuencia mucho más alta, por lo que se puede construir antenas prácticas y transmitir la información. Normalmente la señal es convertida a una frecuencia intermedia, típicamente 70 [MHz]. Debido a que desde un punto de vista práctico, es más fácil llevar a cabo la modulación a la frecuencia intermedia.

La modulación ocupara cierto espectro de la transmisión así que se tiene que ser cuidadoso. Existen dos tipos de modulaciones generalmente, analógica y digital. No se profundizo en la analógica. Generalmente demodular es más difícil a causa de que la señal ya está corrompida después de haber viajado por el medio de transmisión. Y existen tres principales tipos de modulaciones digitales:

- ASK(Amplitude Shift Keying)-Modulación por desplazamiento de amplitud
- PSK(Phase Shift Keying)-Modulación por desplazamiento de fase
- FSK(Frecuency Shift Keying)-Modulación por desplazamiento de frecuencia

ASK no es usada en satélites porque es muy susceptible a perturbaciones, a interferencia y degradación. PSK es la más usada en comunicaciones satelitales. Adicionalmente los esquemas de modulación han desarrollado QAM(Quadrature Amplitude Modulation) para la optimización de las señales digitales, esta es un hibrido ente los esquemas de amplitud y de fase.

La ecuación que define a PSK está dada por

$$v(t) = A \cos(\omega_0 t + \theta_m)$$

Donde θ_m es el desplazamiento de fase y A la constante de amplitud. θ_m es variado de acuerdo con la señal de información, ya sea "0" o "1". Entonces se tiene

$$\theta_m = \frac{(2m+1)\pi}{M}$$

M Es el número de fases, y m va de 0 a $(M - 1)$. Cuando N información de bits se combinan, hay posibilidad de $M = 2^N$ estados de fase. Con un bit $N = 1$, $M = 2$, $m = 0,1$. Entonces $\theta_0 = \frac{\pi}{2}$ y $\theta_1 = \frac{3\pi}{2}$. Cuando fase de la portadora se cambia por 180° (figura 3.25) para cada cambio de bits, este tipo de modulación es BPSK.

La modulación por desplazamiento de fase binaria es, como se mencionó, cuando $M=2$. La modulación es alcanzada cuando la fase de la portadora cambia en 0 y en π dependiendo del dígito binario.

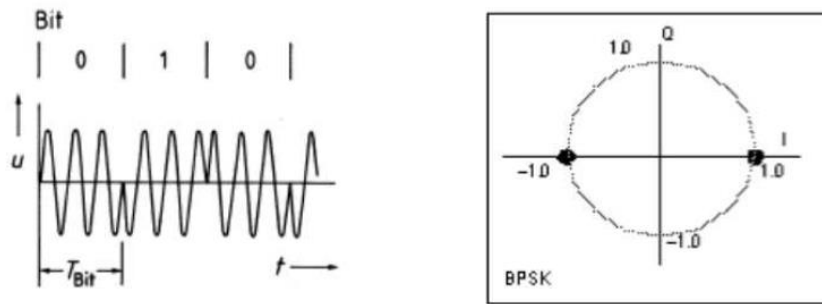


Figura 3.25 Representación temporal y fasorial para una señal BPSK [38]

Luego se tiene la modulación por desplazamiento de fase en cuadratura (QPSK) que está en la figura 3.26, es usada para una forma de onda de $M=4$ símbolos, la fase adyacente se desplazara siendo equidistante en 90° , así hay 4 formas de onda correspondientes a una modulación. Cada símbolo (00, 01, 1 y 11) estará representado por una fase, obviamente esto dependerá del valor de θ_m .

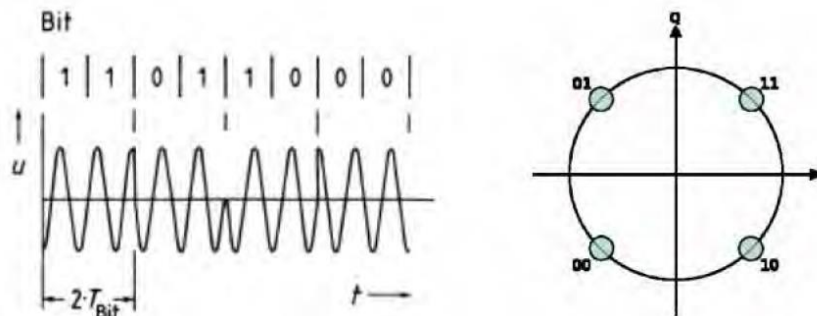


Figura 3.26 Representación temporal y fasorial para una señal QPSK[38]

Cuando un sistema presenta problemas en sus implementación con sistemas de modulación como 8-PSK, lo más común es cambiara esquemas duales como QAM, así se alcanzan mayores distancias con mayor fiabilidad y eficiencia espectral. Algunos de los estándares de transmisión de televisión digital usan 16APSK, 16-QAM o 64-QAM, los primeros dos se muestran en la figura 3.27.

De igual manera, no se tiene control sobre el método para multiplexear las señales al satélite. Pero es importante conocer los principales métodos de multiplexaje que existe, estos son comúnmente conocidos como métodos de acceso al medio.

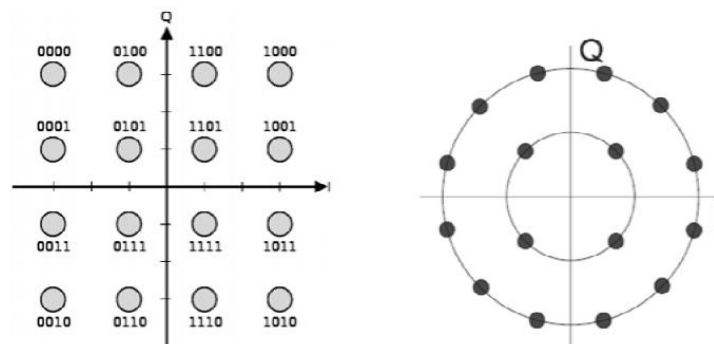


Figura 3.27 Representación fasorial de señales 16-QAM y 16 APSK [38]

El multiplexaje es cuando múltiples señales o cadenas de datos digitales se combinan en una señal a través de un medio compartido. El objetivo es compartir un solo recurso (canal) el cual puede ser el enlace, o una línea de fibra óptica, etc, dependerá la aplicación y los medios, mediante la combinación de las señales banda base. El mutiplexaje en el satélite se hace en el transpondedor a bordo y proviene de varias estaciones terrenas.

Una forma de acceso al medio es como la información que es recibida va a entrar a nuestro dispositivo final y por lo tanto como se va a procesar dicha información. Existen varias técnicas de acceso al medio, a nivel de enlaces satelitales, cada una provee diferentes características.

FDMA (figura 3.28) es la más usada por la sencillez que representa, muy utilizada en servicios FSS y BSS.

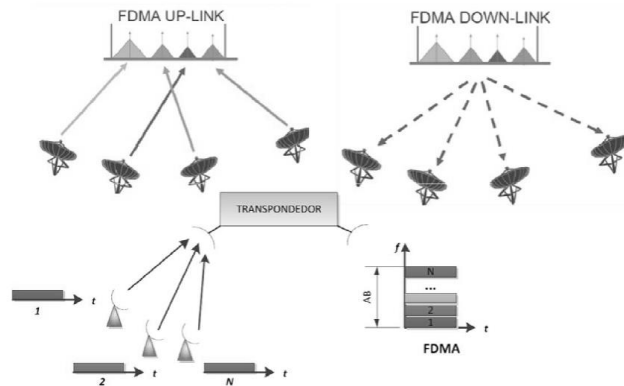


Figura 3.28 Representación de FDMA [38]

Una de sus desventajas es la presencia de intermodulación en los diferentes canales de frecuencia causado por la amplificación no lineal de los amplificadores en el satélite. También está la imposibilidad de manejar grandes cantidades de tráfico por un menor ancho de banda asignado.

La técnica de TDMA es para redes de medios compartidos. Que permite a varios usuarios compartir el mismo canal de frecuencia dividiendo la señal en intervalos de tiempo diferentes, de manera que las ráfagas de información no se traslapen de manera temporal una con otra. Una ilustración ejemplifica su utilización en la figura 3.29.

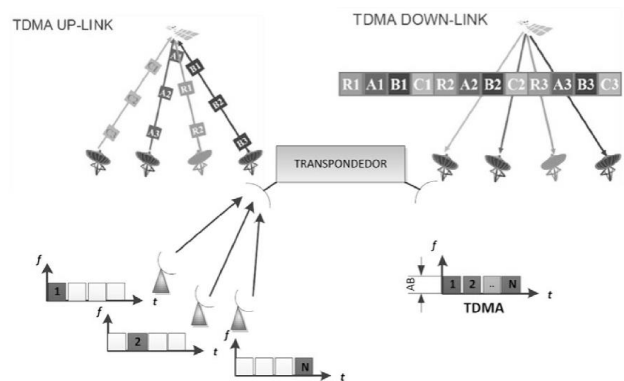


Figura 3.29 Representación de TDMA [38]

TDMA es utilizada para la transmisión de datos, voz y servicios basados en paquetes, debido a que proporciona una forma más eficiente de transmitir portadoras moduladas digitalmente con PSK.

SDMA, método de acceso basado en la creación de canales espaciales paralelos al lado de los canales de mayor capacidad a través de la multiplicación y la diversidad espacial.

CDMA (figura 3.30) es un medio donde varios transmisores pueden enviar información simultáneamente a través de un único canal de comunicación. Esto permite a varios usuarios compartir una banda de frecuencias. Para lograr esto sin interferencias indebidas entre los usuarios se utiliza la tecnología de espectro ensanchado o espectro disperso (SS), el nombre varía dependiendo de la literatura. Basa su funcionamiento en la división por código.

La interferencia entre las señales se resuelve en el receptor mediante una operación lógica entre la señal recibida y un código binario identificado con el transmisor, en la actualidad se utilizan receptores Rake que solucionan este problema. Se le caracteriza por su alta capacidad y celdas de radio pequeñas y por ser muy eficiente en potencia.

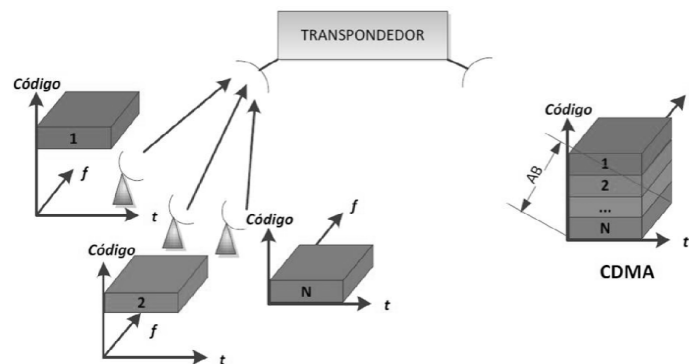


Figura 3.30 Representación de CDMA [38]

A cada estación (o nodo que envía) se le asigna un código que permite identificar la transmisión de dicha estación. El código se basa en una secuencia pseudoaleatoria (PN). Existen dos modalidades, en secuencia directa (DS) y en salto de frecuencia (FH).

En la tabla 3.8 muestra una síntesis de los principales proveedores de servicios satelitales en el mercado actual. Globalstar es empresa encargada del servicio de datos y voz del modén satelital usado en el proyecto. Por ello se le dará un seguimiento.

Tabla 3.8 Tecnologías utilizadas en constelaciones satelitales

Parámetro del sistema	Sistema			
	Iridium	Globalstar	ICO	Sistema regional
Servicio (Velocidad de enlace [Kbps])	Telefonía satelital y datos móviles (4.8)	Telefonía satelital y datos móviles (hasta 9 dependiendo de las condiciones del medio)	Telefonía satelital y datos móviles (4.8)	Telefonía satelital y datos móviles (2.4-4.8)
Tipo de orbita	LEO-Polar	LEO-Inclinada	MEO	Geoestacionaria
Numero de haz puntuales / número de satélites	48/66	16/48	163/10	100-300/1
Capacidad nominal por satélite	1100	2400	4500	16 000
Área de servicio	Global	Global	Global	Regional
Frecuencias del enlace de subida/ enlace de bajada [GHz]	1.616-1.6265/1.616-1.6265	1.62-1.63/2.48-2.5	1.98-2.01/2.17-2.2	1.6/1.5
Frecuencia de enlace de conexión, enlace de subida/ enlace de bajada [GHz]	30/20	5.1/6.9	5.2/6.9	14/12
Ancho de banda por canal para multiplexado [KHz]	TDMA 31.5	CDMA 1250	TDMA 25.2	TDMA 5-10
Potencia [W] para entrada modulada en estación terminal	QPSK 0.45	SS/QPSK 0.5	QPSK 0.625	QPSK 0.5
Figura de mérito (G/T) en antena de entrada [Db/K]	24.5	28.5	26.6	37
Figura de mérito (G/T) en terminal de usuario [Db/K]	-23	-22	-23.8	-23.8

Es fácil notar como la variedad de redes satelitales disponibles en la actualidad, cada una con sus ventajas y desventajas.

3.2.2. Ambiente espacial

Se puede catalogar como ambiente espacial a todas aquellas perturbaciones atmosféricas presentes a partir de la ionosfera, esto incluye: temperatura, húmeda, presión, arrastre atmosférico, presión atmosférica y solar, radiación solar, etc.

La investigación de efectos del ambiente espacial se realizó respecto a constelaciones en órbita satelital baja, a causa de que el modem utilizado en el prototipo pertenece a ese tipo de sistema. El segundo motivo se centra en la utilización de nano satélites y CanSat que son puestos, en su gran mayoría, en órbita baja y para finalizar, que los actuales sistemas que ofrecen MSS están en esta modalidad.

Los satélites de órbita baja se encuentran en la ionosfera (parte de la termosfera) y la exosfera afectan el ambiente. La primera se le llama así porque es un plasma muy delgado de átomos cargados eléctricamente (iones) y electrones a causa de la radiación ultravioleta del sol. Va desde 50 a 1000 [Km], y aunque la órbita terrestre baja comienza 150 [Km] por debajo de eso, no es posible mantener una órbita estable.

La exosfera comienza después de la ionosfera. En la ionosfera la luz ultravioleta causa ionización, por lo que provocara radiación y que las partículas (electrones y protones) estén eléctricamente cargados. Especialmente los protones son los que pueden causar daño temporal y permanente a los instrumentos electrónicos, aumentando el ruido en ellos. Si se quiere enviar sensores allá arriba en esta orbita, es necesario asegurar que el nivel de sensibilidad sea adecuado para el nivel de la señal que se está tratando de medir y de tener un rango dinámico que permita extraer los datos significativos. El sol es la mayor fuente de radiación electromagnética, por lo tanto, cada vez que aparece en el rango de vista de un satélite, puede causar hasta la interrupción del servicio debido a la fuerte radiación que recibe.

Los problemas de radiación también se puede deber a la presencia de campo magnético. La radiación aumentara con la altitud, eso es un hecho, por eso es una ventaja operar en este tipo de orbita. La intensidad de radiación también varía dependiendo del campo magnéticos de la tierra y del cinturón de Van Allen, aumentando por encima del ecuador y disminuyendo por debajo de los polos.

La temperatura en el espacio oscila entre -269 [°C] a más de 400 [°C]. Según estudios recientes (Antunes, Sandy en 2013), se ha comprobado que la temperatura de una placa de metal (obviamente de algún satélite) posicionada en

órbita baja terrestre llega a tener un ciclo de temperatura de $-170[^\circ\text{C}]$ a $123[^\circ\text{C}]$ con presión de $10^{-5}[\text{Pa}]$, dependiendo de cuánto tiempo este sometida a la radiación solar. Hay que recordar que una órbita tiene aproximadamente la mitad de su tiempo en la luz solar y la otra mitad en la sombra de la tierra. Los satélites modernos utilizan calentadores y enfriadores incluidos en sus sistemas de control térmico. Si se hace referencia a la electrónica común (no diseñado específicamente para la industria espacial), generalmente alcanzan temperaturas en rango típico de $-30[^\circ\text{C}]$ a $90[^\circ\text{C}]$, es importante tener esto en consideración, así se puede obtener la electrónica que se adapte mejor a una misión espacial específica.

Afortunadamente no todo tiene una solución complicada. Los efectos gravitatorios causados por el sol son prácticamente imperceptibles en órbita terrestre baja y media. Sin embargo la fuerza que ejerce la tierra sobre ellos sí podría resultar perjudicial.

Existe basura en el espacio, cientos de materiales de satélites pasados y en actual desuso. A esto se les llama desechos espaciales. Y pueden causar una colisión con el satélite, lo cual no es muy probable ya que hay muchos estudios y cálculos para que eso no pase, pero aun así, es posible. En la figura 3.31 se muestra una simulación de los desechos espaciales en órbita terrestre baja arriba de los $2000[\text{Km}]$.

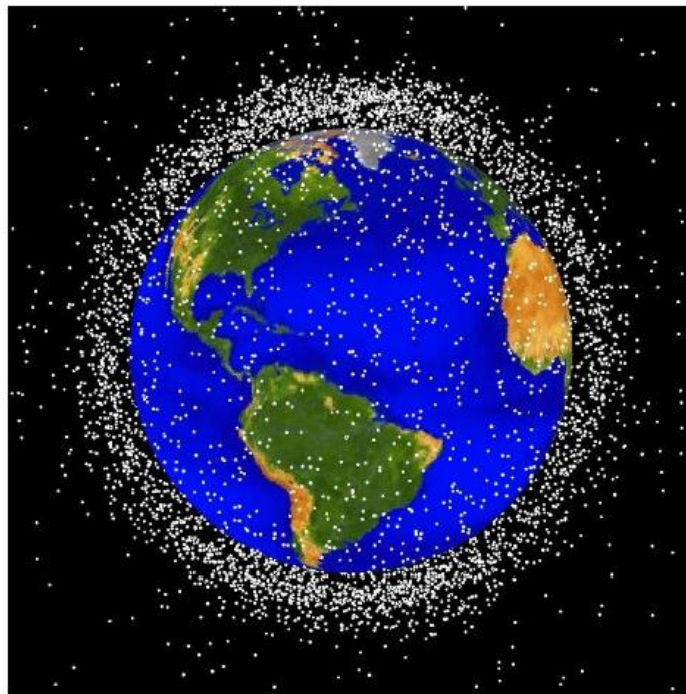


Figura 3.31 Desechos espaciales en órbita terrestre baja [4]

El arrastre atmosférico es causado por la fricción con el cuerpo de un satélite, principalmente con las partes altas de la atmósfera terrestre. Por lo tanto, los satélites en órbita terrestre baja sufren mayor resistencia atmosférica. Por debajo de los 180 [km], la fricción causa calor excesivo sobre la superficie de un satélite de tal manera que los satélites pueden llegar a calentarse vertiginosamente.

3.2.3. Globalstar. Historia y futuro

El sistema de servicios móviles satelitales de Globalstar empezó a dar funcionamiento como tal en el año de 1999, con una primera generación de satélites, que, aunque perdió 11 en la primera oleada, consiguió operar funcionalmente.

Sus principales características es usar tecnología CDMA con espectro disperso (SS) además de un control eficiente de potencia incluido. Alta eficiencia en codificación de voz. Haces puntuales con buena ganancia en antena. Ofrece servicios de voz y de datos generalmente de 7.4 [Kbps] hasta 9 [Kbps]. El sistema abarca casi el 80% de la tierra, principalmente latitudes medias. Una característica importante es su configuración de red prevista para que cada proveedor de servicios pueda ofrecer servicios de manera independiente a través de puntos de acceso bajos su jurisdicción y control.

La constelación Globalstar brinda tres servicios principales: fijo para lugares residenciales o rúlaes, móviles para el acceso móvil a los componentes del satélite y de modo dual portátil que permite acceder a sistemas satelitales y terrestres.

Tráfico de voz y datos de baja velocidad serán enviados desde teléfonos portátiles a través de uno o más satélites de pase y luego a las puertas de enlace en tierra según Smith y Hendrickson (1995).

Desde el 2003 tuvo intentos de proporcionar Wi-Fi mediante el acceso a su constelación satelital, sin embargo el proyecto no se concluyó. Comenzó a desplegar satélites de segunda generación desde octubre de 2010, esto sin afecta la estructura normal de la red o su funcionamiento. Termino de lanzar dichos satélites en 2013. Actualmente propone utilizar un canal para el ancho de banda de Wi-Fi y poder brindar este servicio de manera más eficiente. A través de TPLS (Terrestrial Low Power Service) que utilizara la norma 802.11 y será una extensión de la banda ISM de 2.4 [GHz] (espectro utilizado para el Wi-Fi común). En la figura 3.32 se observan los canales 1, 6 y 11 que proveen Wi-Fi a los dispositivos comunes, Globalstar posee la concesión de la banda terrestre destinada al canal 14, el cual pretende usar como nuevo medio de acceso a internet.

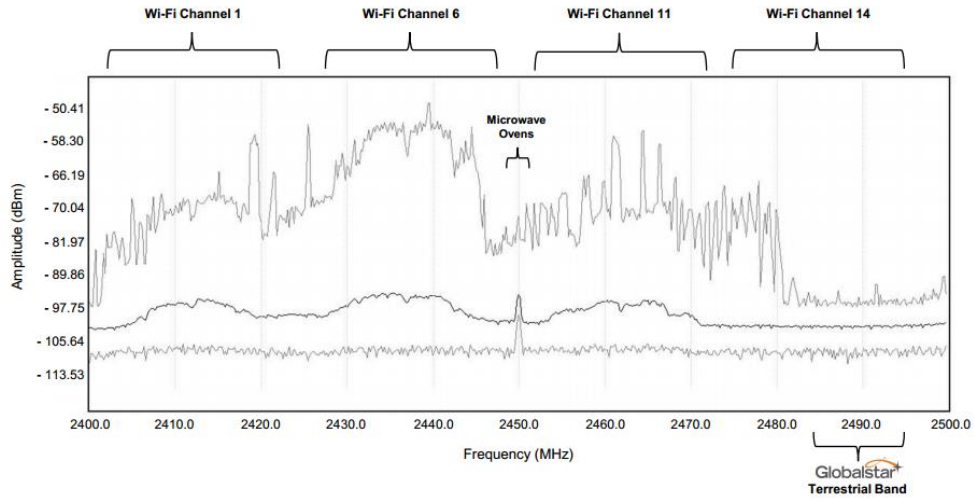


Figura 3.32 Canales disponibles para Wi-Fi [52] [21]

El proyecto aún sigue en pie y se esperan avances significativos en los siguientes años.

3.2.3.1. Arquitectura de la constelación

Su sistema completo comprende básicamente tres segmentos (figura 3.33) que son el segmento de usuario, segmento espacial y segmento terreno conectados a redes públicas y privadas.

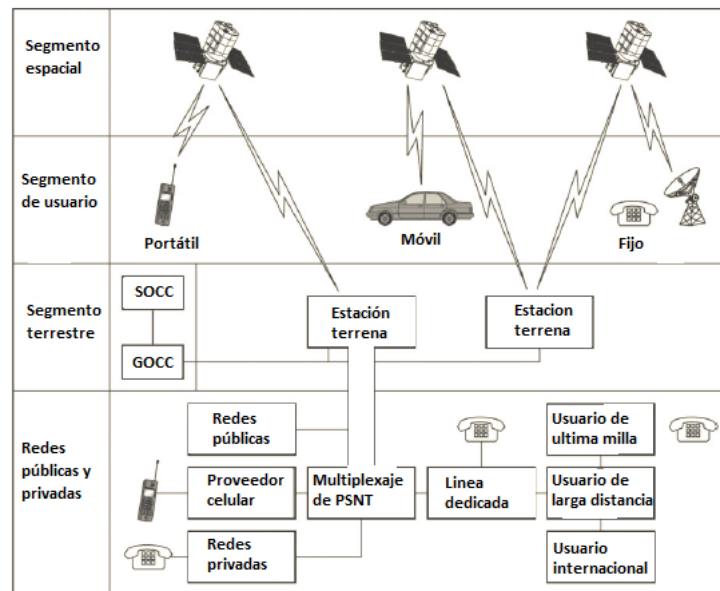


Figura 3.33 Red satelital Globalstar[52]

La información se canaliza a la estación terrena designada, la cual gestiona la información al centro de control del proveedor de servicios (SPCC), al centro de control Globalstar (GOCC) y aún más importante, al usuario final. Dichas conexiones (GOCC y SPCC) utilizan la red mundial de datos de Globalstar. No tiene acceso a ello otras redes privadas. El SPCC gestiona la nave espacial y la constelación. GOCC por su parte asigna la estación terrena capaz de cubrir el servicio y recolecta los datos operaciones y de facturación.

El segmento espacial tiene una constelación tipo Walker de 48 satélites como se muestra en la figura 3.34. A una altura de 1 414 [Km] en órbita circular inclinada a 52° y con periodo orbital de 114 [min]. La constelación no cuenta con capacidad para proporcionar ISL.



Figura 3.34 Constelación Globalstar [52]

La eficiencia del servicio de Globalstar se antepone a otros competidores como Iridium, quien igual compete en el ámbito de los MSS en órbita Big Leo. Según un estudio de Frost & Sullivan (2002). Globalstar tuvo un éxito del 100% en sus conexiones en transmisión de datos, contra un 70 % de Iridium. A igual que el número de llamadas con teléfonos satelitales de ambas compañías, Globalstar logró un 100% de éxito en sus enlaces en contra de 97% que tiene Iridium.

3.2.4. Sistema SCADA

Un sistema SCADA es de las principales aplicaciones que brindan las constelaciones satelitales, debido a su simplicidad e importancia. Básicamente es un

sistema de adquisición de datos y supervisor de control. Tradicionalmente se usan como sistema que permite supervisar una planta o un proceso por medio de una estación central que hace de Master o MTU, y una o varias unidades remotas, generalmente llamadas RTU, por medio de las cuales se hace el control y la adquisición de datos desde el medio que se está monitoreando.

Las topologías que manejan SCADA han cambiado pero generalmente siguen la forma presente en la figura 3.35. En donde el proceso objeto de control es lo que se desea supervisar, el origen de los datos que se quieren recolectar y distribuir. La adquisición de datos se hace con un conjunto de instrumentos de medición (sensores) y alguna interface que permita su interconexión. SCADA es la combinación de hardware y software que permitirá la colección y visualización de los datos proporcionados por el área de dispositivos. Al final vienen los clientes que son los usuarios finales donde la información es presentada. Esta última etapa utiliza HMI(Human-Machine Interface) para poder visualizar los datos, que es una aplicación con interfaz gráfica amigable para el usuario poco especializado.

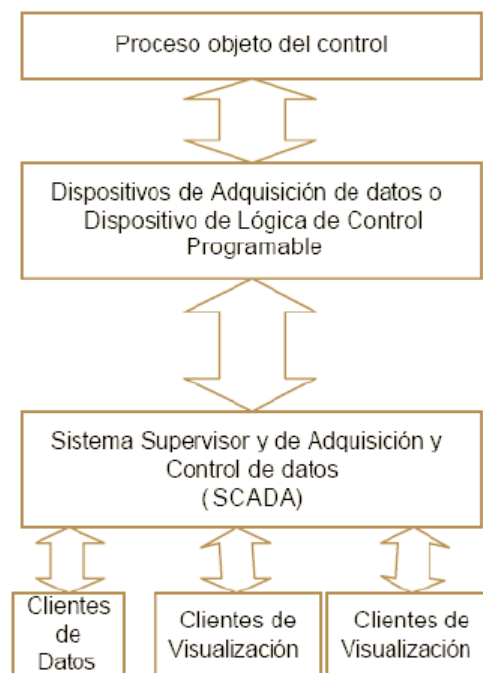


Figura 3.35 Diagrama de sistema SCADA [75]

La tabla 3.9 muestra una síntesis de los servicios móviles así como el ancho de banda utilizado. Se observa que Globalstar es un sistema capaz para esta aplicación.

Tabla 3.9 Uso de SCADA en redes satelitales

Categoría	Ancho de banda	Aplicaciones típicas	Ejemplo del sistema en uso
Básica	Pocos bits por segundo hasta 1 [Kbps]	Transferencia de mensajes. Comunicación maquina a máquina, SCADA, monitoreo remoto	EutelTRACS (GEO); ORBCOMM(LEO)
Baja tasa de transferencia	1 -10 [Kbps]	Voz, e-mail, internet básico, SCADA	Globalstar (LEO); Iridium (LEO)
Mediana tasa de transferencia	10- 100 [Kbps]	Voz, e-mail, internet, multimedia	Inmarsat (GEO), Thuraya (GEO),
Banda ancha básica	100-3000 [Kbps]	Internet, multimedia	Inmarsat (GEO), Thuraya (GEO), Iridium Next(LEO)
Banda ancha común	1-10 [Mbps]	Internet de calidad terrestre y multimedia	-
Banda ancha de gama alta	10-50 [Mbps]	Internet rápido de calidad terrestre y multimedia	Inmarsat (GEO) (Inicio en 2014)

El término de SCADA es muy antiguo al igual que su funcionalidad. Hoy en día con el desarrollo de las redes digitales de datos, la definición tiene que modificarse para incluir esta nueva conectividad que será abordada en la parte final del presente capítulo.

3.2.5. Aplicación y servicio de las constelaciones satelitales

Hay muchos más servicios que pueden ofrecer las redes satelitales. Desde GPS, telefonía satelital, y aplicaciones como tele educación, telemedicina, combinaciones de video o de voz, servicios multimedia, de aplicaciones militares, exploración profunda del espacio, diseño de nano satélites. Cada tipo de constelación tienen sus ventajas. Una de ellas es el poder proporcionar, dependiendo de sus características, diferentes tipos de servicios y aplicaciones a los usuarios. Entre los servicios se encuentra especialmente la rama de servicios de red, que incluyen servicios usados directamente por el consumidor, como teléfono, fax, video y servicios de datos. Este tipo de servicios requieren un gran ancho de banda para poder transmitirse y también buena calidad de servicio (QoS), el cual es un factor de gran importancia en las redes de datos y tiene un uso idéntico en las redes satelitales.

Para Hendrickson (1998) las aplicaciones de MSS no solo se limitan a usuarios comunes, también tienen beneficios para industrias militares. Si se

desarrolló un sistema que sirve para aplicaciones militares, que necesitan una cierta robustez y rendimiento avanzado, no hay razón para creer que no pueda servir para aplicaciones más comunes y de tráfico de datos menos encriptado.

Los satélites suelen usar comúnmente el protocolo IP el cual es usado por internet. Con el crecimiento del internet se diseñó primeramente el IPv4. Ya con esta forma de presentar internet es posible acceder a servicios como son de WWW (World wide web), FTP, Telnet, DNS, e-mail y voz sobre IP (VoIP). Hasta hace unos años, las direcciones IPv4 eran suficientes para cubrir la demanda. Cuando ya no lo fueron surgió el IPv6, el cual, aumentaba las direcciones de host en todo el mundo.

Ahora no es tan usado IPv6 ya que aún se considera “nuevo”, pero en un futuro no muy lejano, todos los sistemas satelitales usaran esta convención. Sobre todo las aplicaciones que incluyen VSAT, como la mostrada en la figura 3.37, gracias a la portabilidad de estas terminales muchas investigaciones que relacionan constelaciones satelitales y redes de datos se están desarrollando actualmente. Los sistemas M-VSAT con la quimérica combinación entre los servicios móviles (MSS) y la tecnología VSAT, operando en banda Ku y Ka.

La desventaja de usar VSAT es que solo se puede recibir información, pero según Hrychiewicz, B. Koch (2002), es más fácil implementar sistemas LAN de alta velocidad que con otro tipo de tecnología satelital. Dicha tecnología sigue en evolución.

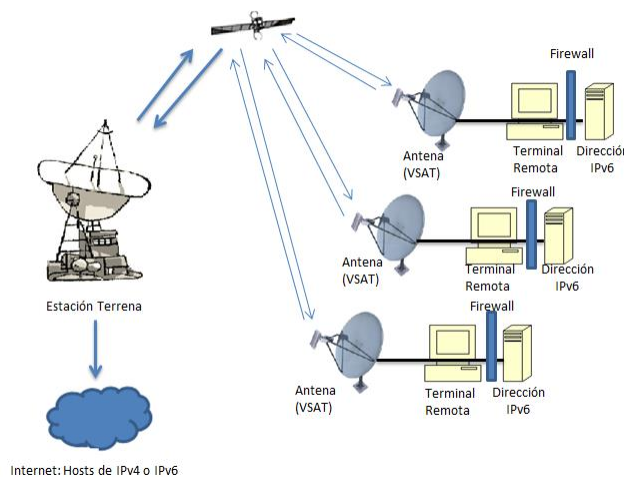


Figura 3.36 Aplicación de las redes satelitales con VSAT [28]



La distribución de video para las compañías de cable que ofrecen el “Triple Play” o “Cuádruple Play”, recepción de video DTH por los consumidores, son algunas de las aplicaciones que en el futuro estará disponible con IPv6.

La agencia espacial europea (ESA) está trabajando en un proyecto llamado “Preparación para IPv6 en satélites de comunicación”. El objetivo de este proyecto es investigar en introducir el protocolo. Hasta ahora con la arquitectura establecida se ha podido usar para la emisión de video (DVB-RCS y DVB-S/SCPC).

El proyecto SATSIX igualmente es un proyecto de comunicaciones basado en IPv6 con satélites que también son capaces de transmitir video DVB-RCS y DVB-S2 pero a bajo costo gracias a la utilización de terminales con Wi-Fi para general redes WiMax y a su arquitectura de solo enviar mensajes multicast para así reducir los costos.

Se ha creado “La fuerza especial europea de IPv6”, la cual ha tenido grandes avances y son líderes en el campo de investigación de las comunicaciones con satélite de dos vías, los servicios móviles satelitales y distribución de video HDTV sobre DVB-S2.

“Anywhere, anytime internet access” es un proyecto creado por la agencia espacial nacional de Francia con colaboración con la universidad de Strasbourg y la empresa Astrium para identificar y aplicar IPv6 en diferentes escenarios que podrían beneficiar la comunicación y movilidad en el uso de constelaciones satelitales.

Una colaboración muy importante surgió hace tiempo. NASA, Cico Systems y Tecnologías Satelitales Surrey (SSTL) ha realizado experimentos que incluyen satélites LEO ya equipados con tecnología para soportar IPv6 y poder monitorear la tierra, mandando imágenes a estaciones terrenas. Gracias a Cisco y sus routers es como se puede hacer un ruteo en IPv6.

La mayoría de los satélites tienden a estar en órbita geosíncrona, se prefiere geoestacionaria (42400 [Km]) por sus ventajas como aparecer de forma estándar respecto a la Tierra, minimiza los requisitos operacionales de las estaciones terrestres, simplifica el diseño de las estaciones terrenas, el rastreo es más simple, no tiene desplazamientos en frecuencia debidos al efecto Doppler, y su área de cobertura es más extensa (aproximadamente un tercio de la supervise de la tierra es cubierta). También hay ciertas desventajas, a latitudes de $\pm 84^\circ$, lo satélites ya no son visibles. No son fiables por debajo de 5° de elevación debido a que la señal recibida es insuficiente para la comunicación de buena calidad, causada por la

combinación de incremento en la degradación troposférica y reflexiones en tierra que causan fluctuaciones en la señal.

Una constelación LEO (figura 3.37) para comunicaciones fijas o móviles al principio no parecía factible debido al gran número de estaciones en tierra requeridas para su correcto funcionamiento.

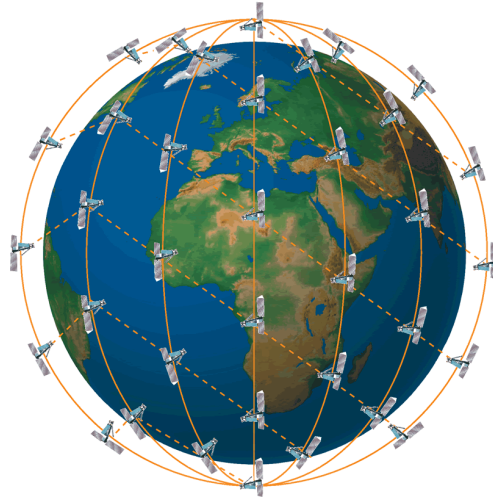


Figura 3. 37 Constelación de satélites LEO de Iridium [38]

Con el paso de los años la tecnología ha madurado y se están aprovechando sus ventajas como son: el uso de transceptores portátiles (como por ejemplo un modem satelital), la cercanía a la tierra significando una señal más fuerte y más libre de dispersiones, distorsiones o ruido, mayor rehusó de frecuencias debido a que el área de cobertura es mucho menos y la reducción del retardo de transmisión proporcionado por un sistema de multihaz(es)(figura 3.38).

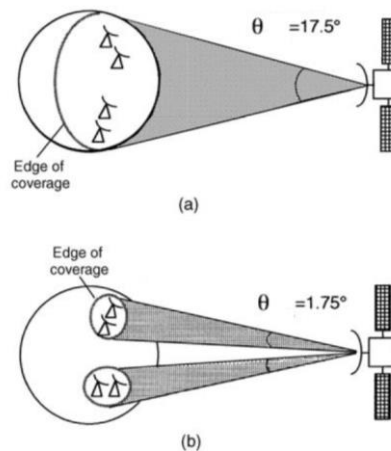


Figura 3.38 a) Satélite con capacidad de un solo haz. b) satélite con capacidad para multihaz(es) [38]

Se observa como un sistema multihaces tiene sus ventajas, como una reutilización de frecuencias, que consiste en usar la misma banda de frecuencias varias veces de manera que sea posible aumentar la capacidad del sistema para un determinado ancho de banda, además de conseguir un mejor ajuste a la zona de servicio respecto al caso de un único haz.

En general los sistemas de emisión móviles (MSS y BSS) necesitan transmitir a mucha mayor potencia, ya que va directamente al usuario y que el servicio sufra caídas no es tolerado por el consumidor. Y al tener múltiples haces es posible tener un canal de retorno de la información, aprovechando que se tiene comunicación full dúplex, esto, según Clausen, Linder y Collini-Nocker (1999) puede hacer más eficiente la tasa de transmisión, permitiendo enviar hasta video MPGE-2.

A veces la comunicación entre redes satelitales que usan el protocolo IP llega a tener un mayor desempeño que las redes terrestres de información. En el estudio realizado por Tsunoda, Ohta, Kato y Nemoto (2004) se comprobó estas redes tienen rendimiento mayor debido a un hangover más rápido entre estaciones, y a una movilidad más amplia.

3.2.6. Modem Satelital

Un modem es un dispositivo que tiene tanto modulador como demodulador. Convierte las señales binarias en señales análogas que pueden transmitirse por las redes telefónicas, las redes de datos actuales, o redes satelitales como es el caso del prototipo. Esta es la parte más importante del proyecto ya que brinda acceso de comunicación full-duplex a nivel mundial.

Existen dos modos en los que trabaja el modem. Envío de paquetes o envío de datos asíncronos.

En transmisión asíncrona de datos, tiene que conectarse forzosamente a una red telefónica conmutada, y se transmite a un modem de acceso telefónico. Normalmente se establece la conexión entre 30 [s] a 60 [s] debido a la negociación del modem y el tiempo de formación. Se cobraría un cargo por conexiones a larga distancia por razones de roaming.

Por otro lado, se tiene el envío de paquetes. Hace un enlace a través de la red satelital del proveedor al internet. Luego los paquetes de internet son transmitidos a un servidor. La conexión se completa siempre y cuando exista línea de vista, es casi instantánea estableciéndose entre 2[s] a 3 [s]. Sin embargo, la degradación del servicio (la transmisión de paquetes) puede ocurrir, dependiendo de las propias



conexiones a internet y del enrutamiento. Puede requerir una VPN para garantizar la seguridad. Además, se tiene acceso a una dirección IP dinámica o estática. Se usan comandos AT para establecer la comunicación.

El modem trabajando en modo de computación de paquetes es más útil para este proyecto. Por ello se seleccionó esta forma de enviar informacional sobre la asíncrona, debido a que en dicha comunicación sería necesario conectarnos a la red conmutada telefónica, dicha red queda fuera del alcance de este proyecto.

En una red de datos, es importante ser capaz de identificar donde la transmisión de datos se inicia y donde termina la transmisión. Los datos, junto con los identificadores del inicio y final de los datos, se llaman tramas, esto es análogo a la parte de comunicación síncrona. Además, se añaden direcciones, control contra errores y otra información para que el equipo emisor puede decirle a la computadora receptora qué hacer basado en un protocolo enviado cuando se recibe la trama.

A nivel de bit o byte, los errores de trasmisión se solucionan mediante el aumento de potencia o del ancho de banda de la trasmisión en curso, un método efectivo pero más complicado es el usar mejores (por lo tanto, más robustas) técnicas de codificación de canal y de modulación. Se sabe que al trabajar en sistemas reales, es imposible eliminar por completo los errores de bits, al final se propagaran a los niveles de paquetes.

Existen mecanismos de retransmisión que se utilizan para recuperar los errores (paquetes perdidos). Por lo tanto, la transmisión de paquetes puede hacerse fiable incluso si la transmisión de bits no es fiable. Característica determinante que diferencia a la comunicación asíncrona del envío por paquetes, ya que en asíncrona, por tratarse de una comunicación simple, el software y hardware no es muy complejo y no permite una redundancia completa al momento de transmitir bits. Sin embargo, esta capacidad de recuperación de error es adicional y afectara el costo, tiempo y flujo de la transmisión. También se basa en esquemas de detección eficientes y paquetes de confirmación (en recepción) para asegurar la llegada exitosa del paquete. Para el esquema de retransmisión, la eficiencia de la utilización del canal se puede calcular como:

$$\eta = t_t / (t_t + 2t_p + t_r)$$

Donde t_t es el tiempo para la transmisión de un paquete dentro de los canales, t_p es el tiempo para la propagación del paquete a lo largo del canal para el

receptor, y t_r es el tiempo de procesamiento del paquete de confirmación por parte del receptor. Se observa que los largos tiempos de trasmisión de paquetes, o pequeños tiempos de propagación con tiempos de procesamiento de paquetes son buenos para el rendimiento de la transmisión de paquetes. Esto es de vital importancia porque se espera que el prototipo diseñado, si bien no requiere mucho ancho de banda para enviar la información, si requiere un constante envío de información, por lo tanto, el envío de paquetes es la solución perfecta al momento de escoger.

Por otro lado, un paquete de gran tamaño también puede conducir una probabilidad de error mayor. Se tiene que P_b es la probabilidad de un error de bit, la probabilidad de error de paquete es P_p de N bits se puede calcular por la ecuación

$$P_p = 1 - (1 - P_b)^n$$

Según los estudios recientes (figura 3.39), muestran que un paquete de 8 bits (tamaño de la trama a enviar en este proyecto) tiene una probabilidad de error de 10^{-3} en envío de paquetes.

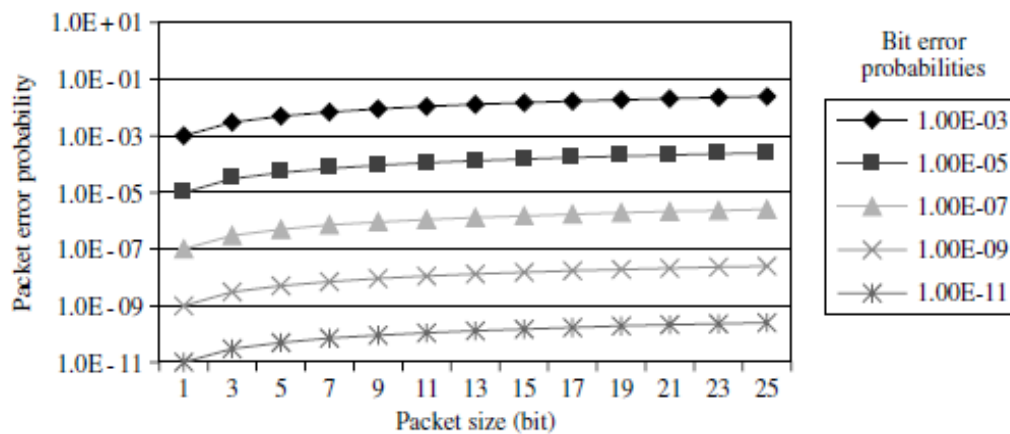


Figura 3.39 Probabilidades de error al transmitir paquetes de datos [53]

Según Sandhoo y Hernández (2014), un buen sistema satelital tiene una probabilidad de error BER= 10^{-2} , ya con la redundancia y el código necesario. Por lo que aún se está al margen y por lo tanto el sistema de transmisión de paquetes es la mejor elección.

Walpert, Guinasso y Lee III(2002) afirman que el modem satelital en una versión anterior se probó en condiciones inhóspitas del golfo de México como un sistema de monitoreo y alerta instalado en boyas. La eficiencia de las transmisiones fue del 95 %. Bastante bueno, a la altura de comunicaciones alámbricas actuales.

Usar PPP como protocolo de capa dos es uno de los puntos fuertes en este sistema ya que pruebas hechas con el mismo dispositivo (en conjunto con una suite de protocolos TCP/IP) en comunicación por paquetes, lograron un retardo de 0.25 [s] según Tardif (1994).

Estudios realizados por Raines y Davis (1995) ha demostrado que la comunicación de servicios en órbita baja terrestre tiene una mayor capacidad en cuanto a envío de paquetes se refiere, superando a los satélites geoestacionarios. Esto sí, y solo si se minimiza el retardo promedio de paquetes en la red. Lo cual es posible al enviar poca información concretamente de manera continua, para no sobrecargar el tráfico de la red.

3.3. Sistema de comunicación inalámbrica y adquisición de datos

3.3.1. Comunicaciones alámbricas e inalámbricas en la actualidad

Las redes de datos han crecido de manera impresionante. Un tipo de red muy estudiada son las redes de área local o LAN, son redes relativamente “pequeñas” que se encuentra en un área física delimitada. Las MAN o redes de área metropolitana, obviamente, por extensión son más grandes. Hasta llegar a las redes de área amplia (WAN). Estas interconectan varias LAN's, por lo que su extensión pueden llegar a ser igual o mayor en comparación con una MAN.

Un caso particular es PAN o red de área personal, en donde solo está vinculado un usuario y un dispositivo final, lo que simplifica mucho el diseño de la red. Las definiciones anteriores son específicamente para redes alámbricas. Pero no son las únicas que puede haber. Hasta ahora, la redes alámbricas son las usadas en todo el mundo, usando como medio de transmisión cable coaxial, par trenzado o fibra óptica entre los principales.

Existen redes que utilizan como medio de transmisión el ambiente. A este tipo de comunicación se les llama inalámbrica y al igual que las comunicaciones por satélite, utilizan cierta parte del espectro radioeléctrico para poder compartir información a través de ondas. Las hay diversas como su contraparte alámbrica: WPAN, WLAN, WiMax para redes inalámbricas de área personal, redes inalámbrica de área local y redes área amplia inalámbricas respectivamente.

El Wi-Fi es un mecanismo para conectar ciertos dispositivos a internet, es el caso más conocido de utilización de comunicación inalámbrica. También hay muchos dispositivos que funcionan en ciertas frecuencias que no son destinadas para radiocomunicación, un mouse inalámbrico hasta el control remoto de la televisión son variados tipos de aplicaciones. Es de especial interés el caso especial de las WPAN.

Las WPAN son investigadas por el grupo de trabajo de la IEEE 802.15. Dicho tipo de redes se tienen la característica de una corta distancia entre sus terminales o nodos transmisor y receptor. Existe un amplio uso en otros dispositivos portátiles como son laptops, celulares o tablets.

El estándar IEEE 802.15.4 define una comunicación a nivel capa dos del modelo OSI, donde su principal objetivo es brindar comunicación entre dos dispositivos. En la capa de red del modelo OSI se encuentra el conjunto de protocolos de alto nivel llamado coloquialmente ZigBee. Su objetivo principal es la creación de una topología de red (jerarquía) para permitir que un número de dispositivos se comuniquen entre ellos, y así establecer funciones de comunicación adicionales, tales como la autenticación, encriptación, y de asociación a los servicios de aplicación de la capa superior.

3.3.2. Redes de sensores

Las redes de sensores están en todas partes, por lo general están consideradas como complicados sistemas de monitoreo destinadas al área industrial o médica. Sin embargo, son más sencillas de lo que se cree y mucho más fácil de utilizarlas.

Hay sensores automotrices, la mayoría de los automóviles tienen una sofisticada red de sensores para monitorear el rendimiento del motor y de otros subsistemas, el aire acondicionado, temperatura, presión e incluso la proximidad de otros vehículos u objetos. En el medio ambiente hay sensores para medir factores atmosféricos relevantes y se usan con fines de pronóstico meteorológico. También se utilizan para el área de seguridad, un circuito cerrado de cámaras por ejemplo, para monitorear y detectar posibles intrusos. La agricultura, domótica, tecnología militar y demás son otros posibles lugares donde se pueden encontrar.

3.3.2.1. Fundamentos de redes inalámbricas de sensores

La mayoría de los sensores son componentes discretos que tienen tensión y producen datos digitales o analógicos, normalmente requieren otro componente para leer los datos y enviarlo al destino, ya sea una base de datos o algún otro tipo de medio para visualizar los resultados, la nube en la web sería un posible destino para monitoreo en tiempo real y brinda la posibilidad de acceder a los datos de manera remota, aunque no del todo seguro y privado. En este proyecto se tiene al Raspberry Pi como el elemento que leerá y procesará los datos de los sensores.

Ahora que se tiene una idea de que es una red de sensores y el conocimiento de las comunicaciones inalámbricas, se unen estos conceptos. Las llamadas redes de sensores inalámbricas o WSN por sus siglas en inglés, han surgido como un novedoso y eficiente sistema para monitoreo, que hasta puede llegar a reemplazar a los SCADA.

La topología básica de una red de sensores se define en tres términos: nodo sensor o también llamado por ciertos autores como dispositivo terminal, nodo de datos (nodo router o simplemente router) y nodo agregador (coordinador).

El nodo sensor se refiere a los dispositivos que obtienen los datos, no tienen la capacidad de almacenar ni de manipular los datos capturados, simplemente los pasan a otro nodo de la red, normalmente son instalados alrededor de un fenómeno a monitorear.

Si tienen la capacidad de almacenar los datos, se le llama nodo de datos. En redes complejas se encargan de rutear (dirigir y ordenar) toda la información enviada por los nodos sensores al o a los dispositivos de almacenamiento designados, ya sean computadores o un nodo coordinador. Las bases de datos o dispositivos de almacenamiento externo son herramientas utilizadas por estos nodos y pueden visualizar los datos con ayuda de otro dispositivo terminal. Ambos nodos transmiten los datos a una red de forma alámbrica o inalámbrica. Tanto el nodo sensor como el nodo de datos pueden tener varios sensores conectados en ellos, esto para hacer más fluida la transmisión de información.

El nodo agregador, utiliza típicamente un dispositivo de comunicación y un dispositivo de grabación y no tiene sensores conectados, en cambio está comunicado con varios nodos sensores o de datos, y se encarga de procesar cantidades de información más grandes hasta su destino final. Él es el maestro, el que gobierna la red y se encarga de decidir cómo se procesa la información. La comunicación entre nodos también se puede ser de manera alámbrica y en algunos

casos híbrida, usando los dos modos de comunicación, pero el fin de este estudio es deshacerse de los cables e implantarlo inalámbricamente por cuestiones de movilidad.

La posibilidad de utilizar redes mesh (figura 3.40) para mejorar la redundancia y la adquisición de información es una habilidad que solo el mejor tipo de redes poseen.

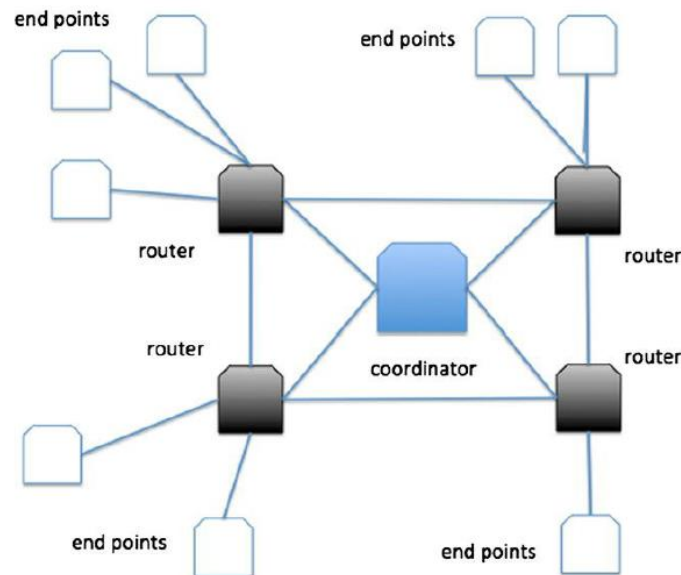


Figura 3.40 Tipos de nodos en las redes inalámbricas de sensores [64]

Se observan como el coordinador es el centro de inteligencia de la red. En el caso del prototipo desarrollado para esta tesis también es la computadora central.

3.3.2.2. Tipos de sensores

Por definición hay dos tipos de familias de sensores: analógicos y digitales. Los analógicos son dispositivos que generan una diferencia de potencial entre 0 a 5 [V]. Se necesita un circuito A/D (analógico a digital) para convertir la diferencia de potencial a un bits. La mayoría de los microcontroladores en el mercado tienen esta característica. Pero no es tan fácil su tratamiento. Los sensores analógicos funcionan como resistencias, y cuando son conectados al microcontrolador casi siempre requieren otra resistencia para evita los cambios en la diferencia de potencial que pudieran causar los propios sensores. Esto debido a que la tensión que fluye a través de una resistencia es continua en el tiempo y amplitud. Por lo tanto, incluso cuando el sensor no está generando un valor de medición, todavía hay un flujo de electrones (traducido en diferencia de potencial) a través del sensor que puede causar lecturas erróneas.

Al muestrear, la lectura de la diferencia de potencial debe ser interpretada como un valor en el rango específico para el sensor. El usar sensores analógicos no es malo ni poco útil, solo es un poco más laborioso y dependerá del objetivo del proyecto a desarrollar.

Los sensores digitales están diseñados para producir una cadena de bits que utilizan la transmisión serial de un bit a la vez. Sin embargo, algunos sensores producen datos a través de transmisión paralela. Los bits son representados como tensión, donde la alta tensión es “prendido” y representa 1. La baja tensión representa “apagado” y es un 0. Estas secuencias de valores 0 y 1 son llamados valores discretos. Este tipo de sensores se puede muestrear más fácilmente debido a que generan datos mucho más rápido, y porque no se necesitan circuitos adicionales para leer los valores. Como resultado los sensores digitales son generalmente más precisos, y fiables que los analógicos, y un poco más costosos.

Existen una multitud de sensores. Los más importantes, utilizados en la industria del monitoreo remoto y hasta en la industria espacial son los siguientes:

- **Acelerómetros:** Estos sensores miden el movimiento. Están diseñados para detectar el variaciones diversas del movimiento (velocidad, la inclinación, la vibración, y así sucesivamente) en varios ejes. Algunos incluyen características giroscópicas. La mayoría son sensores digitales.
- **Sensores de audio:** los micrófonos se utilizan para medir el sonido. La mayoría son analógico, pero algunos de los mejores sensores de seguridad y vigilancia tienen variantes digitales para una mayor compresión de los datos transmitidos.
- **Sensores de gas:** Hay un gran número de tipos de sensores de gas. Algunos miden gases potencialmente dañinos como el LP y el metano y otros gases como el hidrógeno, el oxígeno, y así sucesivamente. Otros sensores de gas se combinan con sensores de luz para detectar humo o contaminantes en el aire.
- **Sensores de ubicación:** Los smartphones modernos tienen sensores GPS para detectar la ubicación y, por supuesto, los dispositivos GPS utilizan la tecnología GPS para ayudarle a navegar. Afortunadamente, sensores GPS están disponibles en formas de bajo costo, lo que le permite añadir detección ubicación a la red de sensores. Los sensores GPS generan datos digitales en la forma de longitud y latitud, pero algunos también pueden sentir la altitud. El presente prototipo utiliza un sensor GPS.
- **Sensores meteorológicos:** Son sensores de temperatura, presión barométrica, precipitación, humedad, velocidad del viento, y así

sucesivamente están clasificados como sensores meteorológicos. La mayoría generar datos digitales y se pueden combinar para crear redes integrales de sensores ambientales.

- Los sensores de proximidad: A menudo pensados como sensores de distancia, usan ondas infrarrojas o de sonido para detectar la distancia o el rango de un objeto.

3.3.2.3. Transmisión y recepción de información

La transmisión y recepción en teoría es sencilla, porque los módulos hacen el trabajo difícil (modulación, acceso al medio, mitigar interferencias), trabajando a una frecuencia de 2.4 [GHz] (banda ISM) y hasta con tasa de transmisión de 250 [Kbps], solo se necesita que el módulo esté configurado correctamente y en línea de vista al otro módulo, que puede ser un nodo router o un nodo coordinador.

Una de las principales ventajas por la que esta forma de comunicación es buena contra la interferencia de ruido electromagnético es debido a la utilización del espectro disperso en secuencia como forma de modulación. Básicamente, cada bit de información a transmitir es modulado en 4 señales diferentes (otro tipo de alfabeto bits), este proceso causa que el total de información que es transmitida ocupe un mayor ancho de banda pero que use la menor densidad de potencia espectral para cada señal, esto causa menor interferencia en las bandas de frecuencia utilizadas y mejora la relación señal a ruido en el receptor porque es más fácil detectar y decodificar el mensaje que está siendo enviado por el transmisor, dicho efecto es una ventaja que otros protocolos de comunicación en radiofrecuencia no brindan.

Otro punto fuerte contra interferencias es la utilización de CSMA-CA el cual es de los mejores métodos para elegir un dispositivo en una determinada red. La mayor utilidad (además del tamaño) viene dada por la potencia necesaria para trabajar. El transceptor tiene la capacidad de estar durmiendo la mayor parte del tiempo (según pruebas hasta un 90 %) mientras que la recepción, el envío de tramas se pueden configurar para tomar solo una pequeña parte de la energía de los nodos.

El modelo utilizado en este proyecto tiene una sensibilidad de recepción de -100 [dBm] o 10 [pW]. Una cantidad de energía comparada con otras antenas de comunicación RF, que en las mismas condiciones llegan a tener un consumo de hasta -35 [dBm], cantidad bastante considerablemente alta.

El modelo de pequeña red (figura 3.41) utilizado en el prototipo cuenta con tres dispositivos terminales y un coordinado o nodo agregador. En una estructura de un solo salto como se ilustra en la figura 3.42.

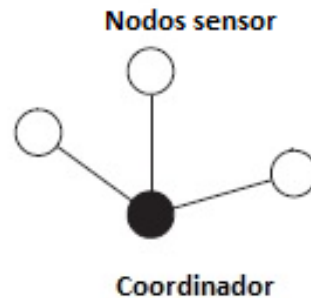


Figura 3.41 Red con un coordinador y tres nodos sensores [63]

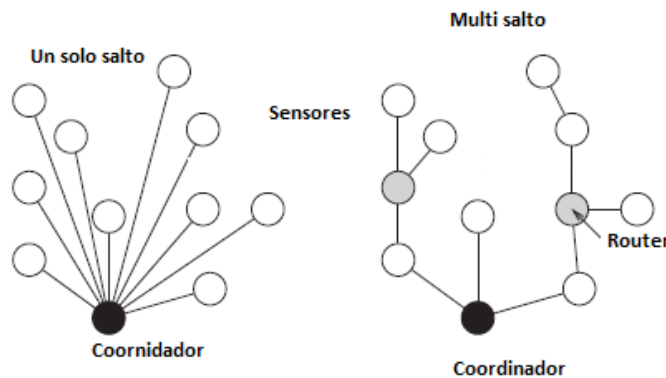


Figura 3.42 Redes de un solo salto y multi salto [63]

3.3.2.4. Tratamiento de datos adquiridos

Los módulos escogidos trabajan bajo dos paradigmas o códigos AT y API. AT significa que el módulo acepta comandos AT a través de su conexión serial y muestra información en un formato legible por humanos usando un derivado del conjunto de órdenes de módem Hayes. API significa que el módulo está configurado para enviar y recibir datos a través de su pila de protocolos. En este proyecto se utilizó API debido a la facilidad que brindan las tramas de información.

El API es un firmware muy utilizado en la formación de redes de malla o redes mesh, y es especialmente diseñado para implementar el protocolo ZigBee completo, lo que significa que los mensajes de datos se encapsulan dentro de un paquete con varias secciones, incluyendo la cabecera. La principal diferencia entre

los comandos AT, radica en que API manda mensajes binarios, y AT se transmite como mensaje de textos o caracteres. En programación, los String o cadenas de caracteres ocupan mayor espacio en memoria de procesamiento que los “integer” o enteros usados en API.

3.4. Sistema de comunicación Ethernet

Una red se considera buena cuando esta brindada con seguridad, que sea adaptable y escalable a cualquier modificación, una velocidad de transferencia de información que cubra las necesidades del cliente y que tenga redundancia en la transición de datos para los datos no se pierdan. Para cubrir las últimas dos necesidades se decidió instalar una forma de respaldo para compartir la información recabada. Fue la decisión más obvia y con el mayor número de beneficios. Hablo de una red Ethernet que es conectada al Raspberry Pi.

Ethernet se basa en el stack de protocolo TCP/IP que a la vez están basados en el modelo de referencia OSI definido por la ISO. Este modelo (figura 3.43) ha servido como guía para la interconexión de dispositivos electrónicos.



Figura 3.43 Modelo de referencia OSI [18]

En el stack TCP/IP (Figura 3.44) solo existen cuatro capas, no siete como en el modelo OSI. Las primeras dos, física y enlace de datos corresponden a la capa de acceso a la red, la capa de red es análoga a la capa de internet. Transporte tienen el mismo nombre en ambos paradigmas y al final la capa de aplicación en TCP/IP representa las capas de sesión, presentación y aplicación del modelo OSI.

La capa de física es la que está relacionada con las especificaciones mecánicas, los dispositivos eléctricos y niveles de tensión para representar los bits se encuentran en esta parte. Básicamente es el medio de trasmisión en contacto con el canal por donde se transmite. La parte de modulación también se encuentra en esta parte porque se trata de leer bits y convertirlos en algo coherente.

La capa dos de enlace de datos dice como accederá la información al dispositivo si utiliza FDMA, TDMA o cualquier otro sistema de acceso al medio, también asigna una dirección MAC que es un identificador único para cada dispositivo.



Figura 3.44 Protocolos de TCP/IP[18]

La capa de red simboliza el ruteo en la red lo cual es el correcto direccionamiento de información. En el protocolo IP brinda una dirección IP de cuatro octetos, para poder hacer el repartimiento de datos en el destino correspondiente. La capa de transporte proporciona seguridad al servicio. Las capas restantes (sesión, presentación y aplicación) son el software diseñado para hacer que los datos sean presentables y no solo una secuencia de 0 y 1.

3.4.1. Diferencia entre red inalámbrica y red alámbrica Ethernet

Las redes cableadas como las inalámbricas se basan en el mismo modelo que es el OSI, ambas tienen aplicaciones totalmente diferentes. Toda red inalámbrica está respaldada invariablemente por un cable. Las redes Ethernet dominan en la actualidad por todos los beneficios que brindan: seguridad, confiabilidad, escalabilidad. Cosas que las redes inalámbricas aún les falta mejorar.

El cable sigue siendo un impedimento de movilidad por lo que se busca desaparecerlo con el tiempo, por lo menos a nivel de usuario de última milla. La infraestructura siempre estará respaldada por una vasta red alámbrica.

3.4.2. Diseño a utilizar

La red del centro de datos del CAT fue diseñada por un compañero del CAT y se muestra en la figura 3.45.

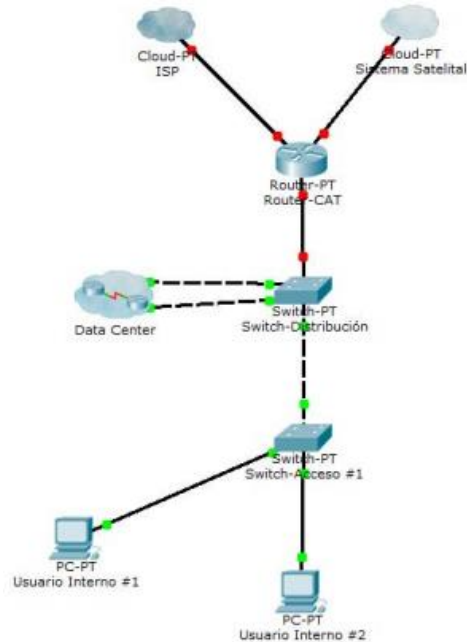


Figura 3.45 Red de datos simulada⁵

El diseño se simuló en software y se probó con la computadora central desarrollada en la presente tesis.

3.4.3. Almacenamiento de información

Como parte de las mejoras y posibles soluciones a problemas de almacenamiento, surgió la oportunidad de implantar una base de datos preprogramada en el Raspberry Pi para almacenar la información. Una base de datos grosso modo es una colección de datos ordenada. La importancia radica en una posible falla en el sistema satelital. El que no exista línea de vista puede limitar las capacidades del sistema y perder la conexión para enviar los datos y el cable Ethernet le quita movilidad al prototipo, por ello es mejor contar con algún medio para almacenar la información que fiarse de un enlace externo (vía satelital o LAN).

Se le decidió guardar en memorias individuales (EEPROM) los datos recabados por cada sensor (tres en total: temperatura, humedad y movimiento), así cada sensor tendrá su respaldo propio de información y una base de datos en el Raspberry Pi, en el supuesto que la red del CAT llegará a fallar.

⁵ Colaboración del Ing. Rafael Escalante Alfonso

3.5. Sistema de control y monitoreo

La computadora central también se encarga de controlar y multiplexar todas las señales, dato e interfaces que convergen en ella.

En el sistema satelital, los radio enlaces son el medio físico de transmisión de los datos, la modulación QAM y los canales usados en frecuencia así como el muestreo de bits corresponde a la capa física. Para la capa de enlace de datos está el modem satelital que hace de intermediario y brinda un protocolo dentro de capa dos. PPP representa una conexión de interfaz WAN, un modo de acceso al medio que es el CDMA y hasta una dirección MAC. El modem también tiene capa física que está relacionada con la antena y es recibida la información proveniente del satélite. En cuando a aplicación, sesión y presentación están relacionadas directamente con el software necesario para visualizar todos los datos recibidos, o sea el Raspberry Pi.

Para el caso de la red inalámbrica de sensores la capa física es la antena y como dispositivo por donde entra los datos recibidos y el medio de transmisión es el aire, para ser exacto, la banda ISM. En capa dos se te tienen los protocolos de acceso al medio como el CSMA-CA y el estándar IEEE 802.15.4. A nivel de capa tres el protocolo para redes mesh ZigBee. Estos tres se encuentran solo en los módulos inalámbricos. Igualmente el Raspberry Pi proporciona la parte de las capas restantes.

Y para el sistema Ethernet el cable UTP brinda lo necesario para las primeras dos capas. Los protocolos ya preinstalados del Raspberry Pi se encargan de otros controladores relacionados con la conexión LAN.

La convergencia, el control y el multiplexado de los datos obtenidos por el prototipo se gestionan manualmente de un subsistema a otro. Los datos son adquiridos de manera automática mediante el uso de un código programado, pero el control de flujo de los datos no es dinámico debido a las diferentes maneras de empaquetamiento para las vías LAN y satelital.

3.5.1. Adquisición de información

El hablar de WSN como reemplazo de sistemas SCADA es un hecho, así como también su aplicación en redes satelitales para hacer más eficiente la transmisión de datos a gran escala.

Los sensores adquieren la información del medio ambiente a estudiar. Esa información es recabada por el modulo inalámbrico que funciona en conjunto como un Arduino para su procesamiento. Como se mencionó anteriormente, se trabajó con códigos API que posean la trama de información mostrada en la tabla 3.10.

Tabla 3.10 Trama de datos de módulos inalámbricos

Campo	Longitud	Ejemplo
Delimitador	8 bits	7E
Longitud	16 bits	00 10
Comando API	8 bits	01
Identificador de trama	8 bits	10
Dirección de destino	64 bits	00 00 00 00 00 00 00 00
Dirección de red de destino	16 bits	00 00
Radio	8 bits	00
Opciones	8 bits	00
Datos de RF	Variable	99 99
Checksum	8 bits	BC

De esta forma la información se transmite entre nodos (nombrados en este proyecto como RTU 1, 2 y 3) y llegara hasta en nodo coordinador que está directamente conectado al Raspberry Pi. La trama de datos es muy parecida a otras tramas como son las utilizadas por protocolos de enlace de datos.

3.5.2. Procesamiento de información

Los datos adquiridos por los sensores se transmitieron hacia una red de datos del CAT o hacia una constelación satelital. Para ello los datos tuvieron que tener el formato adecuado, desempaquetar la trama API y volver a empaquetar la información en un formato de protocolo de enlace de datos para transmitirse a su destino final.

La descripción de la trama API es más sencilla en comparación con la trama PPP, protocolo seleccionado para enlazar los datos entre Computadora Central-Modem satelital y Computadora Central-Red de datos del CAT.

La longitud de cada campo en la trama API está dada en bits pero por convención de programación, al momento de ingresarla en un código programable se expresa en hexadecimal.

El Delimitador tiene como tarea marcar el inicio y fin de la trama. La Longitud marca el tamaño en bytes de la trama desde el propio campo de Longitud hasta el de Checksum. El Comando API indica que tipo de solicitud de transmisión se enviara. El campo de Identificador de trama es para nombrar a la trama y que la interfaz UART lo reconozca. En dirección de destino va la dirección del módulo coordinador de la red y tiene 64 bits. Dirección de red de destino de 16 bits del dispositivo final al que llegara la información, en el caso de nuestra red es el mismo dispositivo pero no la misma información que va en cada campo. El radio es el número máximo de saltos o nodos que puede tener la información para llegar a su destino final. Las opciones es información extra referente a la manera de transmitir la información. Un campo muy importante es el de Datos RF ya que esa toda la carga de información y en el ira la información recabada por los sensores. El Checksum solo es para comprobar que la trama no tenga perdidas al momento de llegar al receptor.

Una vez que la trama ha llegado exitosamente al Raspberry Pi, se ordenara siguiendo una secuencia TDMA para todos los mensajes y se anexara en tramas PPP como la mostrada en la figura 3.46.

Nombre del campo	Señalizador	Destino	Control	Protocolo	Datos	FCS
Tamaño en bytes	1 byte	1 byte	1 byte	2 bytes	variable	2 o 4 bytes

Figura 3.46 Trama de PPP

Los datos adquiridos anteriormente (datos de sensores en tramas API) se posicionan en el campo de Datos (tamaño variable) de PPP. El Señalizador sirve para delimitar cada trama, por default está constituido por el número binario 01111110, el Frame Check Secuence es para asegurar de que la información ha llegado al lugar correcto y está completa, de lo contrario la trama es rechazada y se pide su reenvió. El campo Destino tiene la dirección estándar PPP para enviar la información a todos los dispositivos (broadcast) de la red, en este caso solo es un nodo más en la red. El campo de Control es un único byte con la secuencia binaria 00000011. En protocolo esta indicado que clase de protocolo esta encapsulado en el campo de datos de la trama, en este caso se trata de IP.

El paquete anterior es genérico de PPP, sin embargo en RFC 1661 y RFC 1662 identifica otro tipo de trama parecida que sirve específicamente como trama de la capa de enlace de datos o de enlace de control. Dicha trama con su código se muestra en la figura 3.47 así como también los valores de trama a simular.

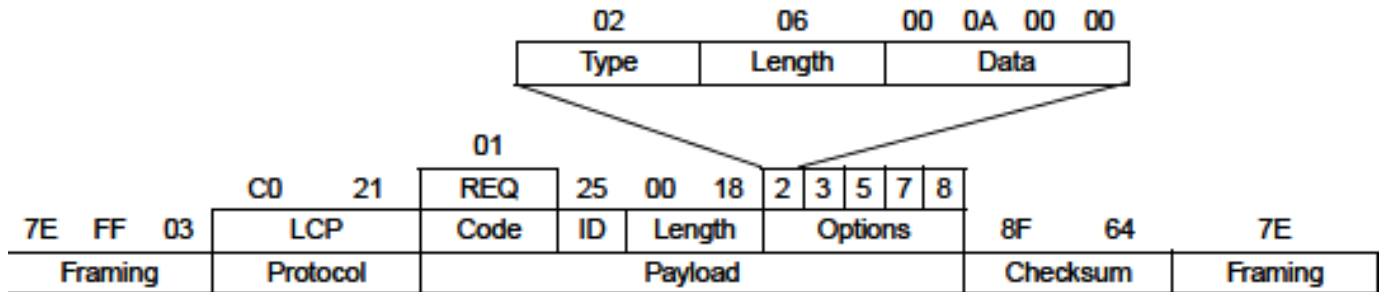


Figura 3.47 Trama de protocolo de capa de enlace

Los valores están en código hexadecimal. El encabezado normal de PPP de 0x7E 0xFF 0x03 seguido de 0XC0 0x21 que indican el protocolo LCP. El modem trabaja con comandos AT, que básicamente es código ASCII, así que dichos comandos se convierten en código binario y de ahí en hexadecimal para poder enviarse y empezar la comunicación.

IP es un protocolo de capa tres, también llamado del mejor esfuerzo, que va ligado con otros protocolos de capas superiores como el TCP-HTTP y que no garantiza la recepción de datos en el destino final. No es la finalidad de este proyecto emular el complicado sistema de comunicación que tiene internet. Únicamente la parte de PPP, por lo que se emulara esa trama en programación. Sin embargo, también es de importancia conocer un poco de que se IP debido a que también se emulara el tener una dirección IP. En la figura 3.48 se muestra el encabezado de IPv4, esta versión es la utilizada en el diseño de la red.

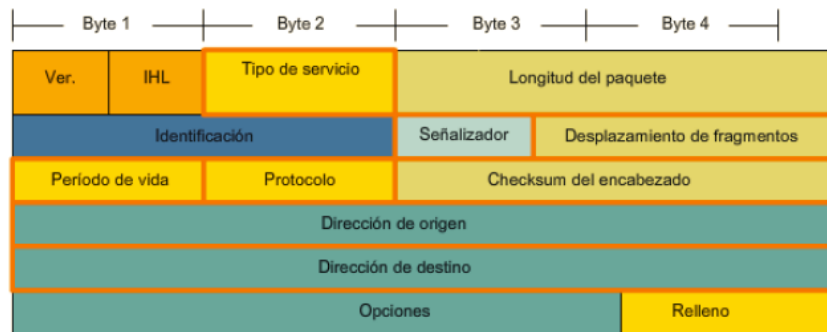


Figura 3.48 Estructura de paquetes en IPv4

Los campos de interés para el proyecto son la Dirección de Origen que es la dirección IP que se le asignará al Raspberry Pi. Y el ICMP, el cual es un protocolo de capa tres que forma parte del conjunto de aplicaciones TCP/IP. El valor de 8 bits en el campo de Protocolo indica la activación de ICMP. Los demás campos no

representan un uso de importancia para este prototipo y solo se simularan con valores por default, que no afectan el desarrollo del prototipo.

El funcionamiento del algoritmo para la obtención de información consta en posicionar los datos en los paquetes especialmente marcados que se transmiten de un dispositivo a otro. Los paquetes indican el tipo de datos que contienen, en diversos casos será una parte de una página web o de correo electrónico, en este solo son datos, simple información. Cada trama se instala en un paquete marcado y delimitado para conseguir su fácil visualización en el programa adecuado del dispositivo receptor.

El tipo de datos de aplicación (mensajes, datos web, VoIP) determinan si la capa siguiente en el modelo OSI es del tipo más simple (UDP) o TCP que es un tipo más robusto de protocolo. Los paquetes TCP generan campos de datos adicionales para abrir y cerrar la conexión, también para que en caso de que exista pérdida de datos, estos se vuelvan a enviar al receptor.

Cada equipo recibe una dirección única en Internet que se parece a 10.225.255.255, y funciona como una dirección de correo postal. El sobre se metió en un sobre más grande con las direcciones de origen y de destino escritas en el frente. Los paquetes en este algoritmo están orientados a obtener un ping.

El protocolo PPP requiere el formato de datos en serie de ocho bits de datos, sin paridad. Los datos PPP se envía en forma de paquetes que inician y paran con el carácter (~) o en hexadecimal 0x7E. El carácter ~ tiene un significado especial, cualquier otra instancia de ~ se sustituye por el }^, que es secuencia de escape. La secuencia de escape funciona transmitiendo dos caracteres en lugar del original, o en hexadecimal 0x20. Debido a que el } tiene un significado especial, también se puede escapar con la secuencia de caracteres 2}].

La conexión PPP se puede dividir en varias fases. En primer lugar, si el enlace está muerto se usa detección de portadora (si hay datos que enviar o no) desde el módem, que es uno de los estímulos que inicia la fase de establecimiento del enlace. Esta fase utiliza Control de Enlace Protocolo (LCP) para detectar y negociar opciones de enlace con el Raspberry Pi. A continuación, la fase de autenticación verifica la ID de usuario y la contraseña (si las hubiera) con Password Authentication Protocol (PAP). La fase final es el protocolo de capa de red. Cada protocolo como IP, se configura con su protocolo de control como IPCP.

Los protocolos de control son muy similares para el LCP, PAP, CCP, y IPCP pero el campo de protocolo es diferente y las opciones tienen diferentes

significados. Cada paquete puede pedir, rechazar, o aceptar una lista de opciones. La negociación comienza con un extremo (Raspberry Pi) y otro lado que solicita una lista de opciones en una solicitud (REQ) de paquetes. Cada opción consta de un byte de tipo de opción, byte de longitud, y los parámetros de opción. Se responde con un reconocimiento (ACK) si se desean las opciones de control.

El PING es una herramienta usada por IP para corroborar la correcta conexión entre los dispositivos terminales a nivel de la capa de red. Es la principal utilidad para comprobar que la conexión está establecida, si no hay PING significa que hay problemas en las capas 1,2 o 3 y se debe realizar troubleshooting.

El ICMP es un tipo de protocolo que se adjunta junto con los paquetes IP y sirve como implementación del PING. La parte correspondiente al ICMP está situada en después de la dirección de destino y origen de red en el encabezado IP. PING usa un datagrama de solicitud de eco ICMP. Si el receptor en la dirección de destino especificada recibe la solicitud eco, este responde con un datagrama de respuestas de eco ICMP. En cada paquete enviado, el PING mide el tiempo requerido para la respuesta.

A medida que se recibe la cada respuesta, el PING muestra el tiempo entre el envío del PING y la recepción de la respuesta. Esta es una medida del rendimiento de la red. De esta manera se utiliza el mensaje eco del ICPM para determinar si el destino (modem satelital) está en funcionamiento. El uso de los mensajes de eco de ICMP es la base de la utilidad ping.

También es posible programar el destino inalcanzable de ICMP para notificar a la computadora central que el modem o la red de datos es inalcanzable. El paquete de destino inalcanzable tendrá códigos que indican el motivo por el cual el paquete no pudo ser enviado.

El diagrama de flujo del algoritmo implementado para el establecimiento de la conexión de capa de enlace de datos se presenta en la figura 3.49. Los paquetes LCP consisten en un código, identificación, longitud, a una lista de opciones de control mencionadas anteriormente, seguidos del estándar de 2 bytes destinados al PPP común.

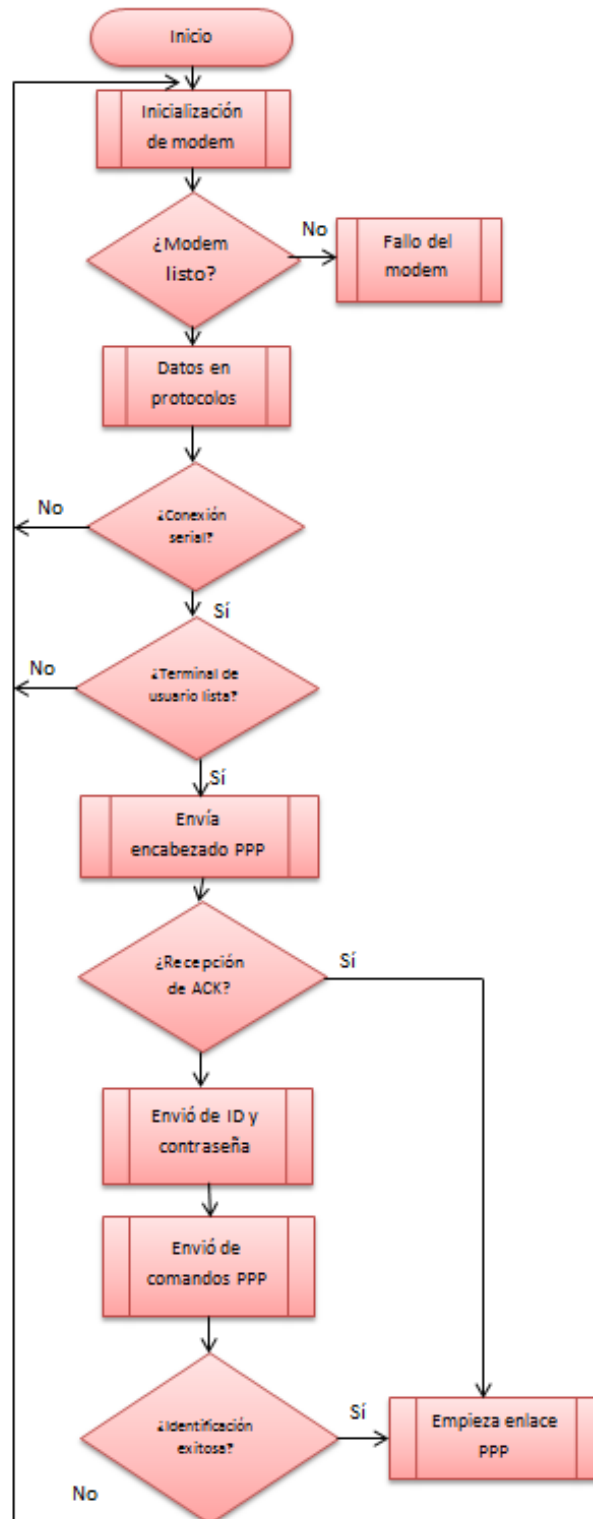


Figura 3.49 Diagrama de flujo para la simulación de PPP

En la figura 3.50 se muestra un simple paquete simulado para hacer ping en la red de datos del CAT que se simuló.

bit	0	8	16	24	31
4	Ver. 0100	IHL 0101	Tipo del servicio 0000 0000	Longitud del paquete 00000000 00011100	0000 : FF 03 00 21
8	Identificación 1000 1000 0001 0000		Flags 010	Desplazamiento de fragmentos 00000 00000000	0004 : 45 00 00 1C
12	Periodo de vida 0111 1111	Protocolo 0000 0001	Header Checksum 00110011 10100111		0008 : 88 10 40 00
16	Dirección de origen 11001101 11001000 00101101 01111100				000C : 7F 01 33 A7
20	Dirección de destino 11001111 10100001 01110101 01000011				0010 : CD C8 2D 7C
24	Tipo de ICMP 0000 1000	Código ICMP 0000 0000	ICMP Checksum 11110111 11111110		0014 : CF A1 75 43
28	Identificador de PING 00000000 00000001		Numero de secuencia PING 00000000 00000000		0018 : 08 00 F7 FE
					001C : 00 01 00 00
					0020 : 22 7C 7E

Figura 3.50 Paquete para PING simulado

El sistema se muestra enviando y recibiendo información de forma completa en la figura 3.51.

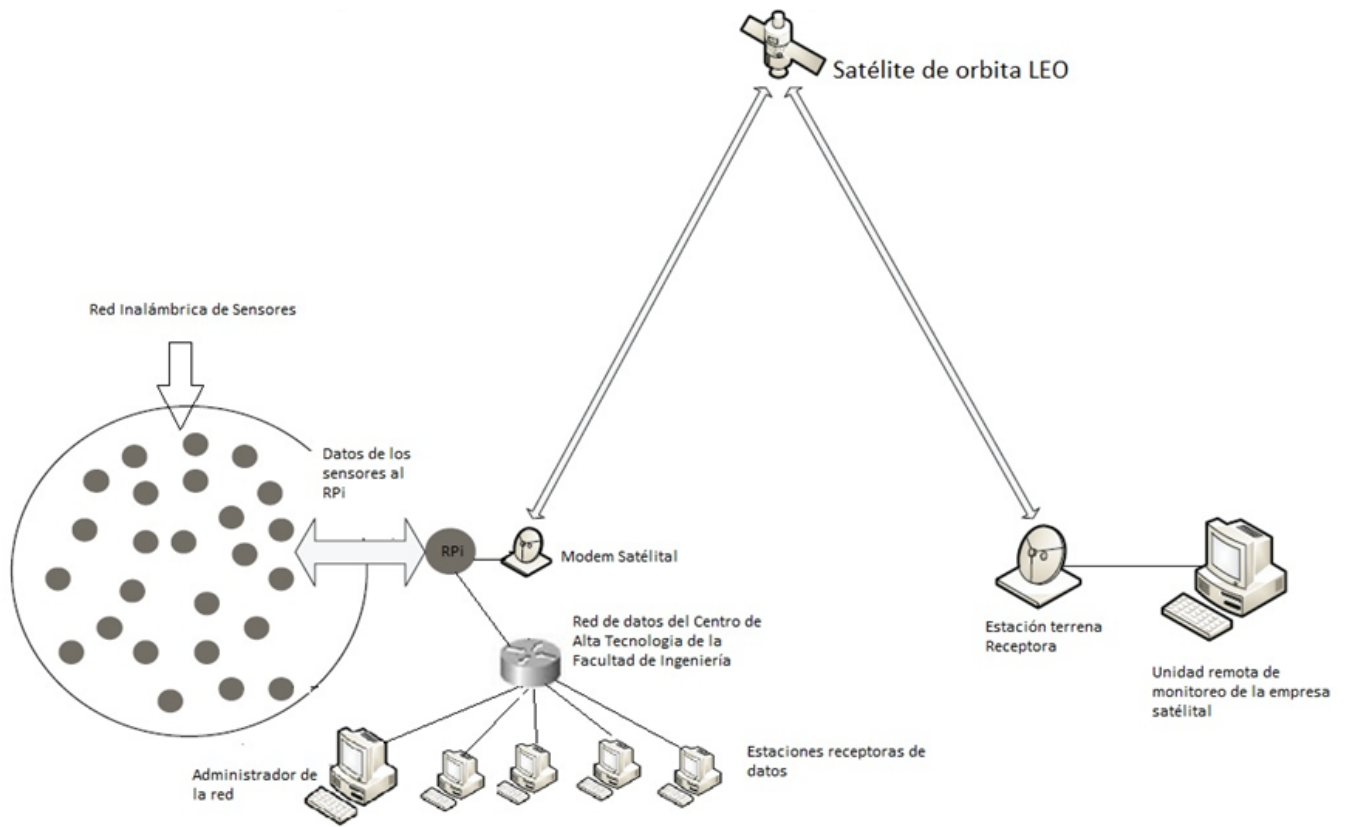


Figura 3.51 Sistema completo

Se observa las diferentes partes que conforman la comunicación del prototipo: Red inalámbrica de sensores a la izquierda, la red de datos del CAT en la parte inferior, el Raspberry Pi en el centro junto con el modem satelital y su antena

estableciendo una conexión con la constelación satelital representada en la parte derecha de la figura.

3.5.3. Subsistema de potencia

Las conexiones son importantes no solo por su interfaz, ya que también indican cuanta energía se está consumiendo. En la tabla 3.11 se hace un resumen en diferentes unidades de medición para representar la energía consumida por los dispositivos utilizados en el prototipo.

Tabla 3.11 Síntesis de consumo de energía en a) tensión y amperes y b) potencia.

Equipo	Requerimientos de tensión [V]	Requerimientos de corriente [mA]
Raspberry Pi	5	700-1000(dependiendo periféricos conectados)
Sistema Nodo sensor (x3)	5	50
Modem satelital	5.6-16 en CD (generalmente 12)	23.33
GPS	3.3	40
Arduino como emulación de modem	5	50

a)

Equipo	Requerimientos de potencia [W]
Raspberry Pi	5
Sistema Nodo sensor (x3)	5
Modem satelital	Inactivo: mínimo a 1.2 [mW], típico 2.4 [mW] y máximo a 3.6 [mW] Recepción: mínimo a 0.28 [W], típico 0.5 [W] y máximo a 2.4 [W] Transmisión: mínimo a 3.6 [W], típico 4.8 [W] y máximo a 5.4 [W]
GPS	3.3
Arduino como emulación de modem	5

b)

Mucha de la potencia y corriente consumida por los dispositivos dependerá de la forma del desempeño demandado por los dispositivos y de cuantos periféricos existan conectados a la computadora central.

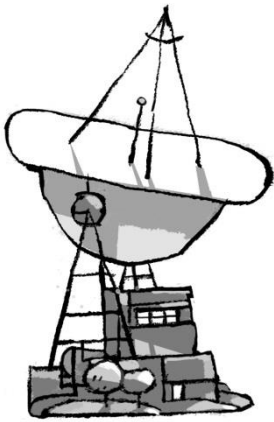
Capítulo 4. Implementación del sistema

If you've ever wondered where your dreams come from when you go to sleep at night, just look around. This is where they are made

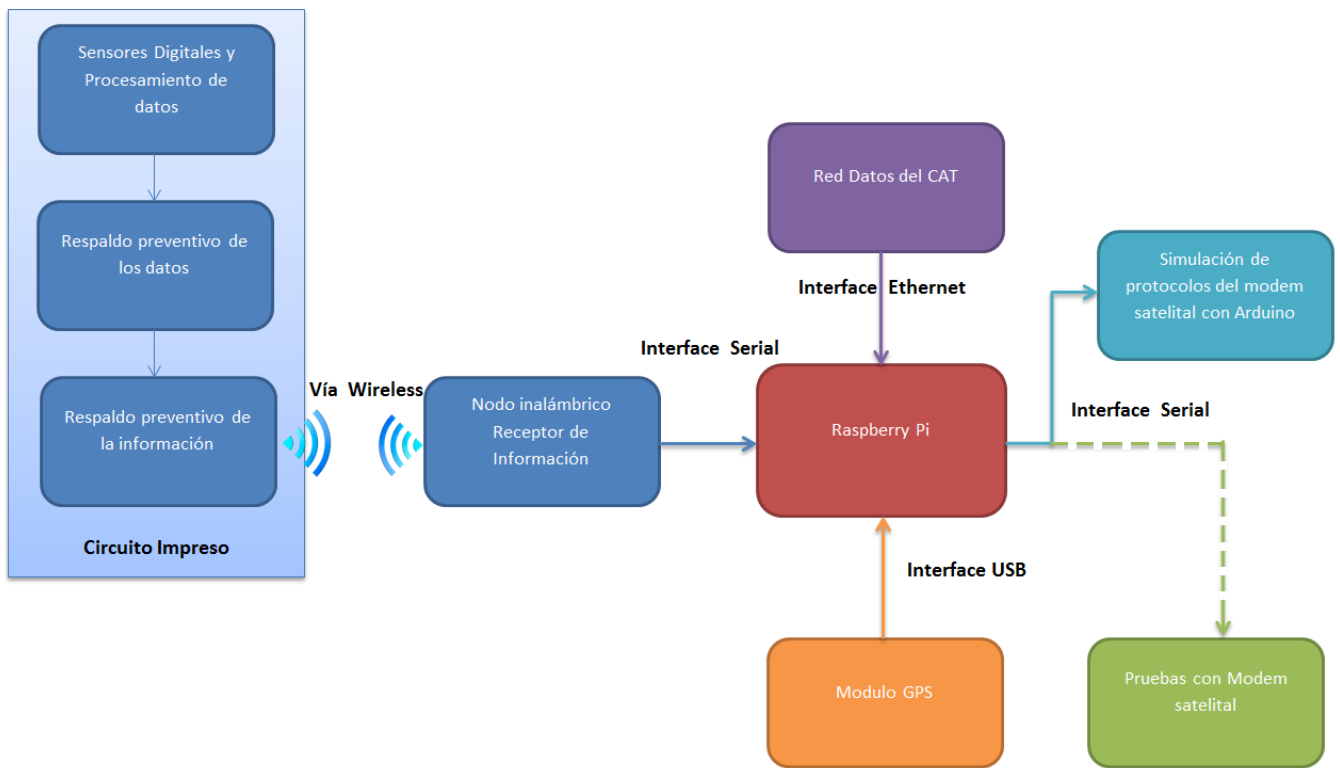
Brian Selznick

Cuanto más complicado, mejor; cuanto más imposible más bello

Paco Ignacio Taibo II



El diagrama de proceso mostrado en la figura 4.1 muestra en diferentes colores cada sistema del prototipo así como sus interfaces correspondientes.



a)



b)

Figura 4.1 a) Sistemas a probar. b) Protocolos de referencia

En la figura 4.1 b) están los diferentes protocolos de referencia utilizados para las conexiones del prototipo, tanto la parte del sistema de telemetría (red inalámbrica de sensores) como la del modem satelital se comparan con el modelo de referencia OSI.

En total se realizaron pruebas de 6 diferentes sistemas:

- Sistema de telemetría, incluye pruebas de la red inalámbrica de sensores y pruebas de sensores directamente conectados al Raspberry Pi.
- Red de datos del CAT. Se simuló dicha red y se probó su conectividad
- Modulo GPS. Se probó la tarjeta de desarrollo GPS
- Pruebas con el Arduino y su simulación de protocolo de capa dos
- Raspberry Pi soportando entrada y almacenamiento de información.
- Modem satelital conectado a computadora ordinaria a través de interface Serial-USB.

4.1. Pruebas individuales de los sistemas

Sensores

Las primeras pruebas se realizaron con sensores conectados directamente al Raspberry Pi, utilizando protocolos SPI e I2C para comprobar su velocidad, de los cuales se probó primero los sensores analógicos de temperatura TMP36 y posteriormente con los sensores digitales DS18B20 .

El propósito de las pruebas con sensores analógicos y digitales es comprobar la viabilidad de los protocolos de comunicación y si estos son efectivos en funcionamiento comparados con un protocolo serial.

Al realizar las pruebas con sensores analógicos se usó el circuito con sensores de temperatura analógicos junto con el integrado MCP3008, el cual es un convertidor analógico digital. Los sensores TMP36 trabajan siguiendo el principio de impulsos eléctricos, o sea analógicos y el Raspberry Pi para leer dichas tensiones necesita ayuda de un convertidor analógico-digital. También se cuenta con un registro de corrimiento en el integrado 74HC595, que permite trabajar con más de dos dispositivos maestros en SPI y así agregar sensores de redundancia para hacer una medición con más datos y por ende más precisa.

Otro propósito de la prueba es corroborar el máximo de sensores que soportan SPI e I2C. Una prueba destructiva. La tabla 4.1 muestra los diferentes tiempos y sensores usados en la prueba.

Tabla 4.1 Pruebas con sensores analógicos

Numero de sensores	Tiempo de tranmision de datos con protocolo:	
	I2C [ms]	SPI [ms]
1	308.87	301.261
2	310.759	305.496
3	310.432	303.524
4	309.432	306.005
5	312.145	302.115
6	313.193	305.897
7	312.749	306.201
8	313.752	307.018
9	318.335	309.894
10	319.063	311.027

Se observa una velocidad superior por parte del protocolo SPI, lo cual se debe a la utilización de varios cables (tres) para el control de información a diferencia de I2C que utiliza solo dos.

La parte de la red inalambrica de sensores se trabajo con tres nodos de red y sigue la estructura presentada en la figura 4.2.

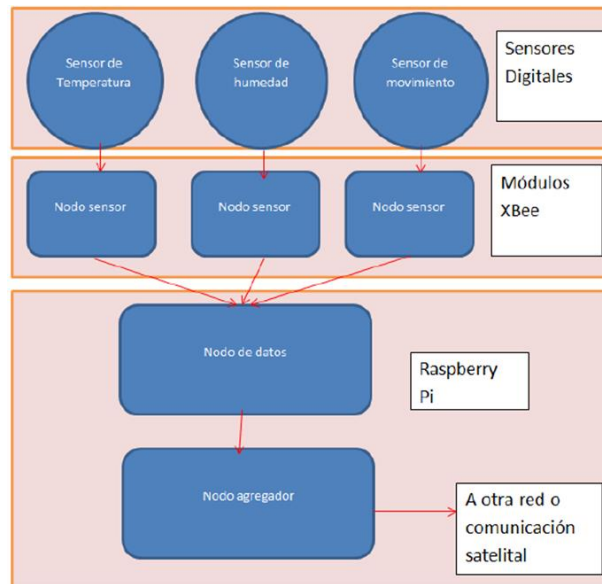


Figura 4.2 Estructura para la adquisición de datos en la red inalámbrica

Se utilizó tres tipos de sensores: temperatura, humedad y de movimiento. Cada uno está conectado a su respectivo microcontrolador, memoria de almacenamiento preconfigurada y módulo de transmisión inalámbrica.

La estructura comienza con los nodos sensores posicionados en el ambiente a sensor, dicho nodo incluye al microcontrolador y al módulo inalámbrico de

comunicación. La siguiente parte se compone del Raspberry Pi y tiene la función de nodo de datos y nodo agregador, lo que en conjunto se conoce como nodo coordinador dentro de ambientes de redes inalámbricas de sensores.

La memoria de almacenamiento tiene la función de guardar la información sin necesidad de que esta pase por un proceso sofisticado de tratamiento, se graba en bruto en un sistema de almacenamiento tipo EEPROM en un integrado 24LC512, trabajando con bus I2C para establecer comunicación con el microcontrolador. Existen una gran diversidad en cuanto a capacidad, se escogió una memoria de 512[Kb] debido a que las pruebas con los sensores analógicos arrojaron resultados donde la información total no pasaba de los 18[Kbits]. Ahora se tiene más que suficiente espacio para almacenar la información hasta por varios días.

El diseño del circuito de dos de los nodos sensores, de temperatura y humedad se muestra en la figura 4.3 y 4.4 respectivamente.

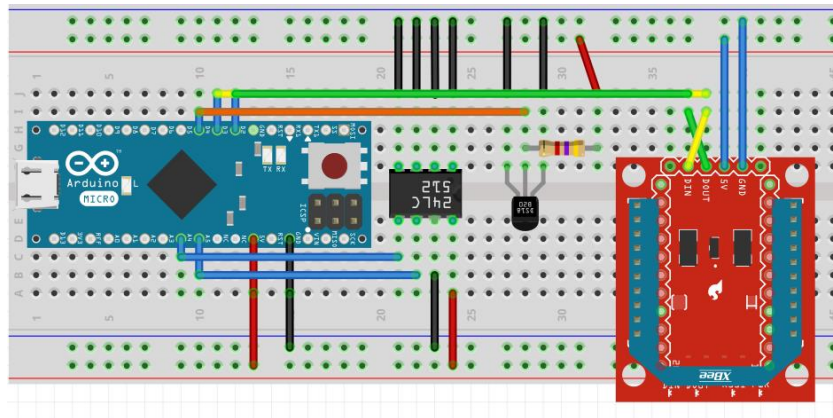


Figura 4.3 Prototipo del nodo sensor de temperatura

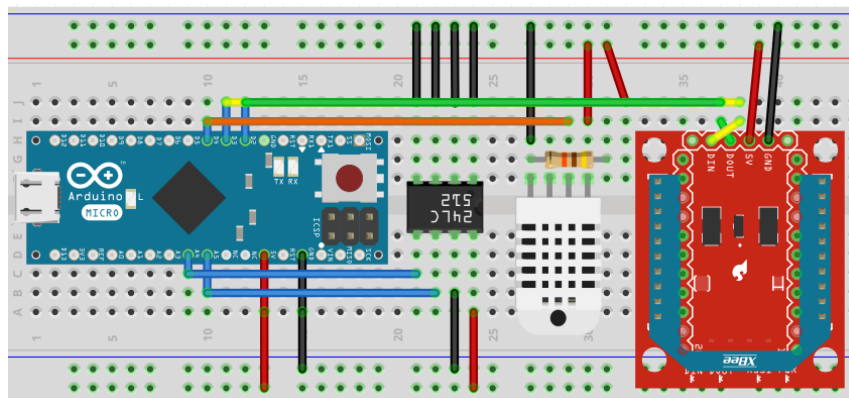


Figura 4.4 Prototipo del nodo sensor de temperatura

Por cuestiones de austeridad no se pudo homologar toda la red con el mismo microcontrolador. Para el tercer nodo se utilizó una tarjeta de desarrollo Arduino Uno, el mismo integrado EEPROM y un sensor de movimiento PIR en la figura 4.5.

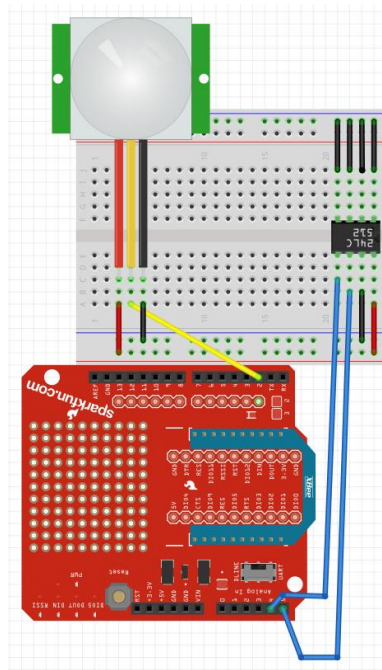


Figura 4.5 Prototipo del nodo sensor de movimiento

El diseño del PCB se muestra en la figura 4.6 y 4.7 para los sensores de temperatura y humedad respectivamente.

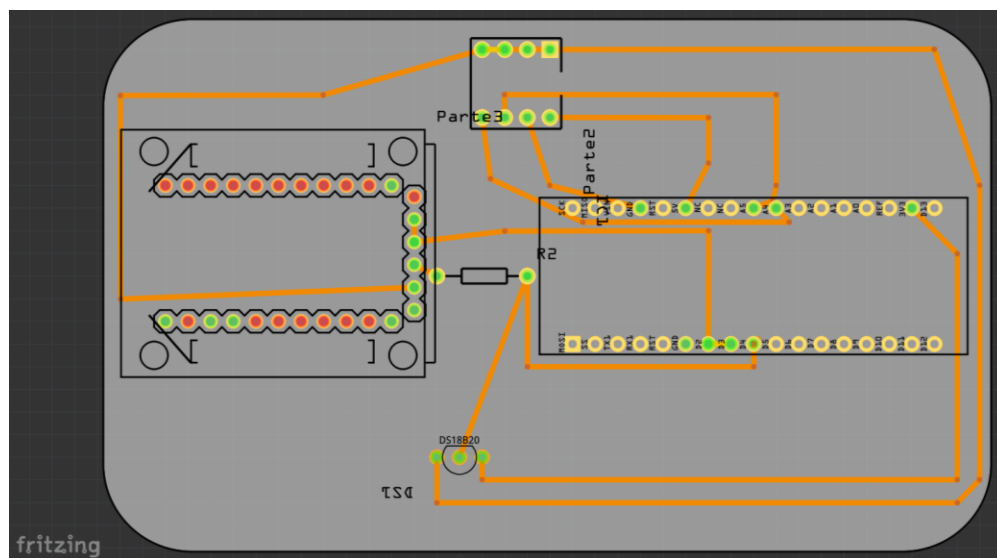


Figura 4.6 PCB de sensor de temperatura

Se realizaron las placas para los nodos de humedad y temperatura, con las cuales se realizaron pruebas en espacios abiertos y cerrados.

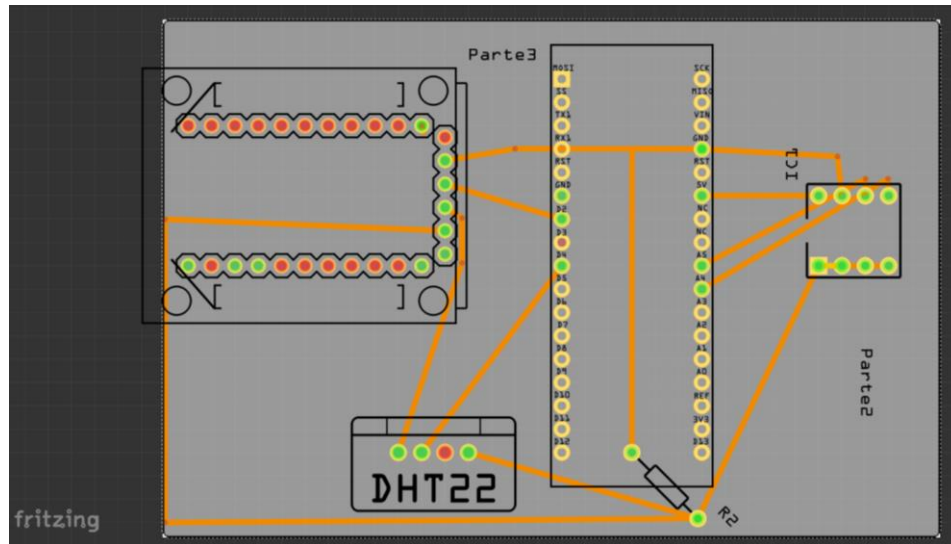


Figura 4.7 PCB de sensor de humedad

Se realizaron pruebas de los tres sensores trabajando a diferentes distancias en el exterior. La figura 4.8 se encuentran enviando información a distancia de algunos metros.



Figura 4.8 Prueba a 1 [m] (izquierda) y prueba a 3 [m] (derecha)

Las pruebas fueron exitosas a distancias de hasta 10 [m], lo cual se considera viable para formar parte de una red inalámbrica de sensores según Dargie, Walteneus & Poellabauer, Christian (2010).

Otro tipo de pruebas fueron realizadas en exteriores pero a una corta distancia entre los nodos. Dichas pruebas se observan en la figura 4.9.

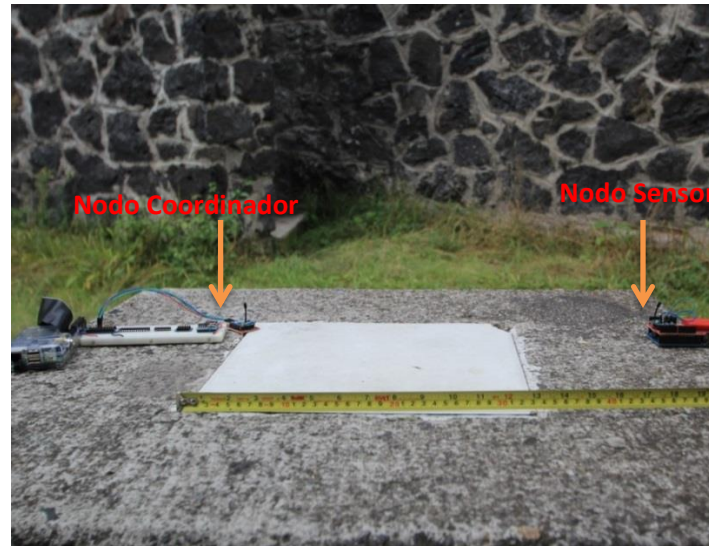


Figura 4.9 Prueba a 0.5 [m]

Las pruebas cortas distancia demostraron que el lapso para transmitir datos es prácticamente el mismo comparado con las pruebas a larga distancia (de más de 2[m]), con diferencias de milisegundos. Los tiempos de prueba fueron medidos con el Raspberry Pi.

Otro tipo de pruebas fueron realizadas en interiores y sin interferencias y obstáculos en la línea de vista de los sensores. Como se puede apreciar en dichas pruebas (figura 4.10) solo se han realizado con tres nodos sensor y el nodo coordinador.

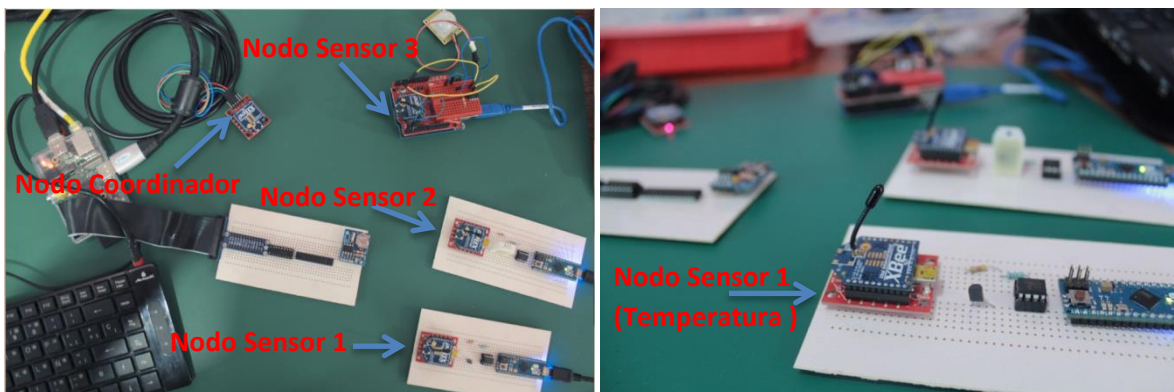


Figura 4.10 Red inalámbrica de sensores (izquierda) y nodo de temperatura (derecha)

En interiores no hubo problemas relacionados con algún tipo de interferencia electromagnética y las pruebas transcurrieron sin novedad alguna. La variante de pruebas en interiores con obstáculos en línea de vista se descartó por falta de un ambiente viciado de interferencias en donde realizarla.

Para el nodo coordinador (modulo receptor inalambrico) que esta conectado al Raspberry Pi se utilizo una interfaz serial-UART. Para llegar a este resultado se probo la comunciacion entre el modulo inalambrico mediante tres diferentes buses de informacion: SPI, I2C y UART. Una conexión mas detallada aparece en la figura 4.11.

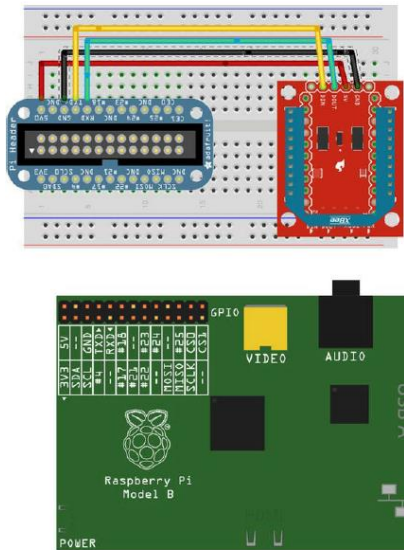


Figura 4.11 Conexiones del nodo coordinador

Red de datos

Como se menciona en el capitulo 3, se utilizo el diseño de red (figura 3.44) de datos del CAT simulado con ayuda de un compañero. La conexión con la red simulada y la computadora central se muestran en la figura 4.12.



Figura 4.12 a) Conexión Ethernet entre red LAN y el Raspberry Pi (izquierda) transmisión de información entre ambos (derecha)

Se probo la interface del cable LAN Ethernet de la computadora central hacia la simulacion como se indica en la figura 4.12

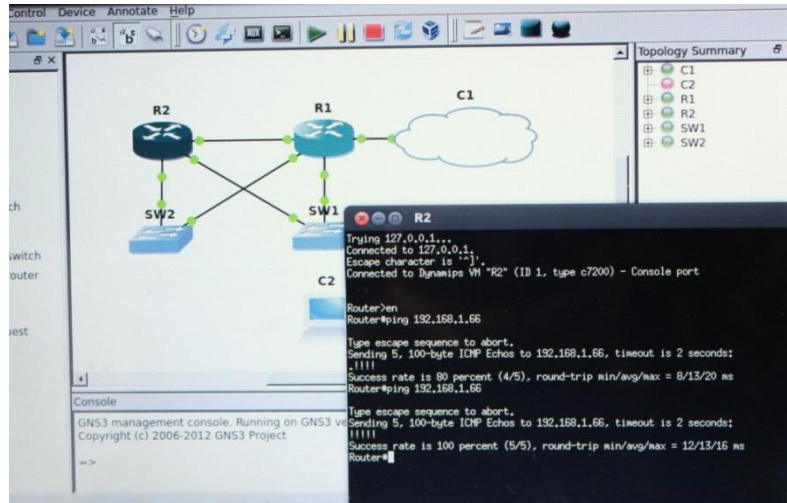


Figura 4.13 Prueba de PING para la red de datos del CAT

La computadora central con la dirección IP 192.168.1.66 se conectó a la red de datos del CAT. Se realiza primero una prueba con la interface de loopback (127.0.0.1) para probar la conectividad con el host local, después desde la red de datos del CAT se realizó una prueba de PING hacia la computadora central. Dicha prueba resulto exitosa por lo que el destino es alcanzado y se comprueba la conectividad y el transmitir los datos.

Modulo GPS

Rajkumar, Sankaranarayanan y Sundari(2013) mostraron algunos de los beneficios de tener un sistema GPS en tiempo real como apoyo para el control y monitoreo, esta vez del sistema como tal. Por ello se justifica la utilizacion de este dispositivo adicional al prototipo. Se probó un modulo GPS como indica la figura 4.14, usando una interface USB, se instaló el IDE Arduino para poder visualizar los datos.

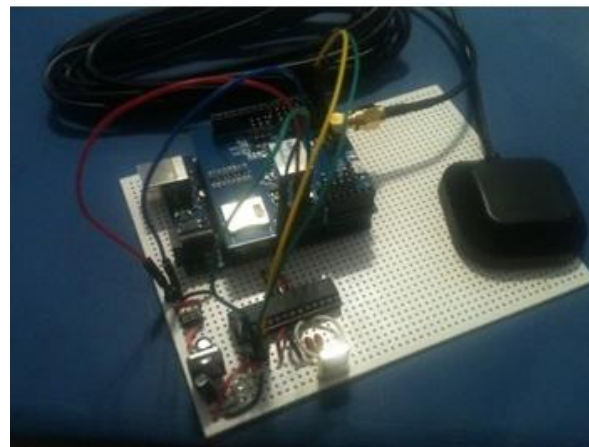


Figura 4.14 Modulo GPS⁶

⁶ Colaboración del Ingeniero Roberto

Simulación del Modem satelital

Para cargar la trama de paquetes en el Arduino y enviarla al Rapberry Pi se uso la conexión presentada en la figura 4.1.

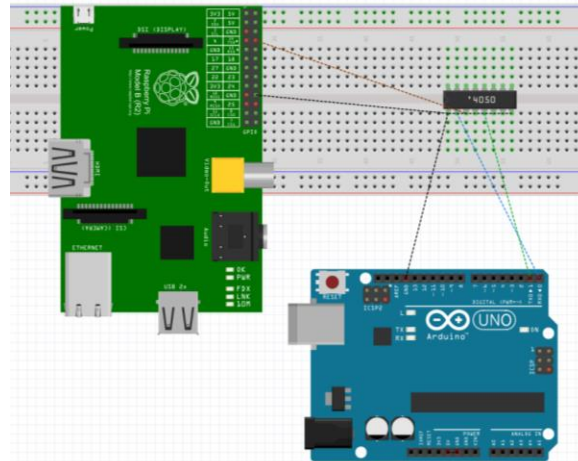


Figura 4.15 Diagrama de conexión entre Arduino y computadora central

La interfaz utilizada es mediante los puertos UART. Se probó el bus I2C pero se encontraron limitantes al tener que agregar otro microcontrolador como esclavo para su correcto funcionamiento.

Computadora central

Uno de los principales problemas con el Rapsberry Pi es la falta de sincronización en tiempo con la red o con el modem si estos llegaran a fallar. Se tienen los datos pero no hay una referencia exacta de cuando fueron tomados. Para ello también se instaló un reloj de tiempo real en la computadora central. El circuito. Para pruebas con los sensores analógicos se utilizó el integrado MCP23017, para mayor facilidad, todos ellos junto fueron instalados en un solo circuito impreso como se observa en la figura 4.16.

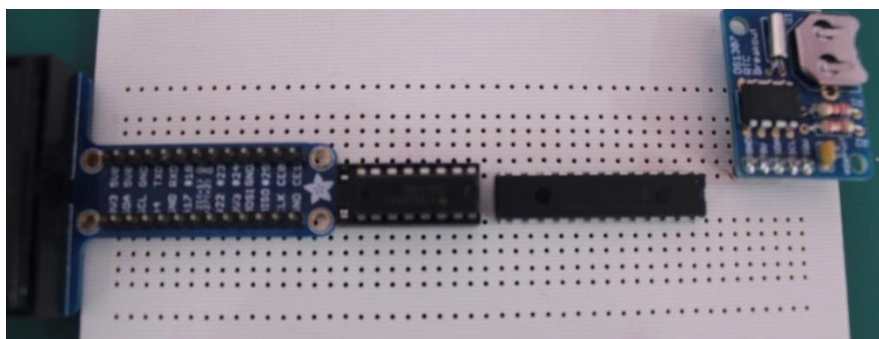


Figura 4.16 Circuito usado por la computadora central

Todos los datos censados por la red inalámbrica se guardan en la base de datos con la fecha y hora exacta de la prueba.

Modem satelital

Se tuvo la oportunidad de trabajar con el modem satelital y realizar pruebas de conexión entre una computadora y el modem con el fin de comprender mejor su funcionamiento. En la figura 4. 17 se ilustra el procedimiento para una conexión exitosa del modem y una computadora.

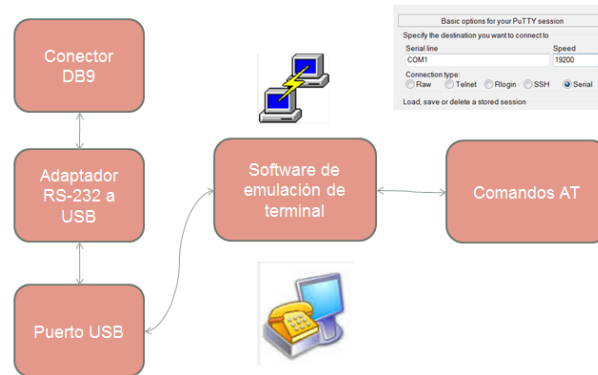


Figura 4.17 Diagrama de proceso para comunicación satelital

El proceso de trabajo del modem consiste en desde una interfaz DB9 transmitir los datos a un adaptador RS-232 hasta el puerto USB de la computadora. Los datos están codificados mediante comandos AT y se ingresan mediante un software de emulación de comandos de terminal.

Se realizaron varias pruebas con el modem explicando su estructura (figura 4.18) lo que permitió conocer detalles de su funcionamiento y así poder proponer este prototipo.

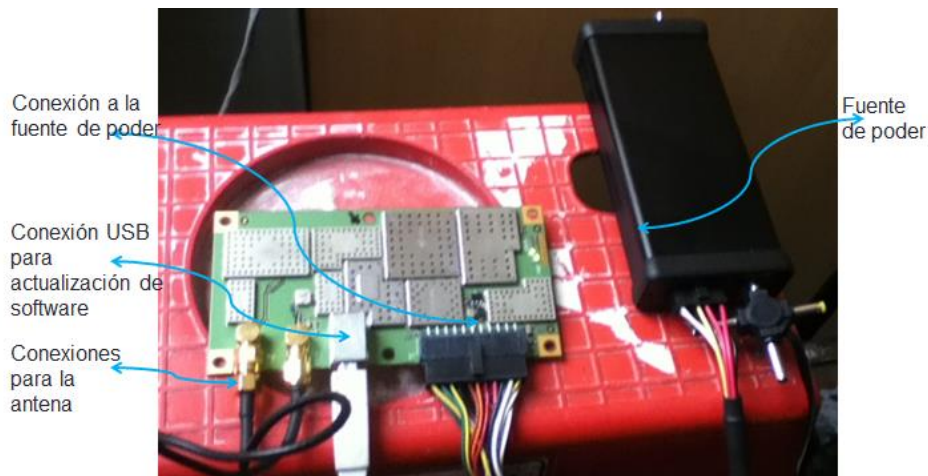


Figura 4.18 Modem satelital

Las múltiples pruebas con el modem consistieron en establecer llamadas y enviar SMS con otros teléfonos satelitales. También se efectuaron pruebas para comprobar el ancho de banda del canal de comunicación.

4.2. Integración de los sistemas

Los sistemas fueron probados todos en conjunto como se observa en la figura 4.19.

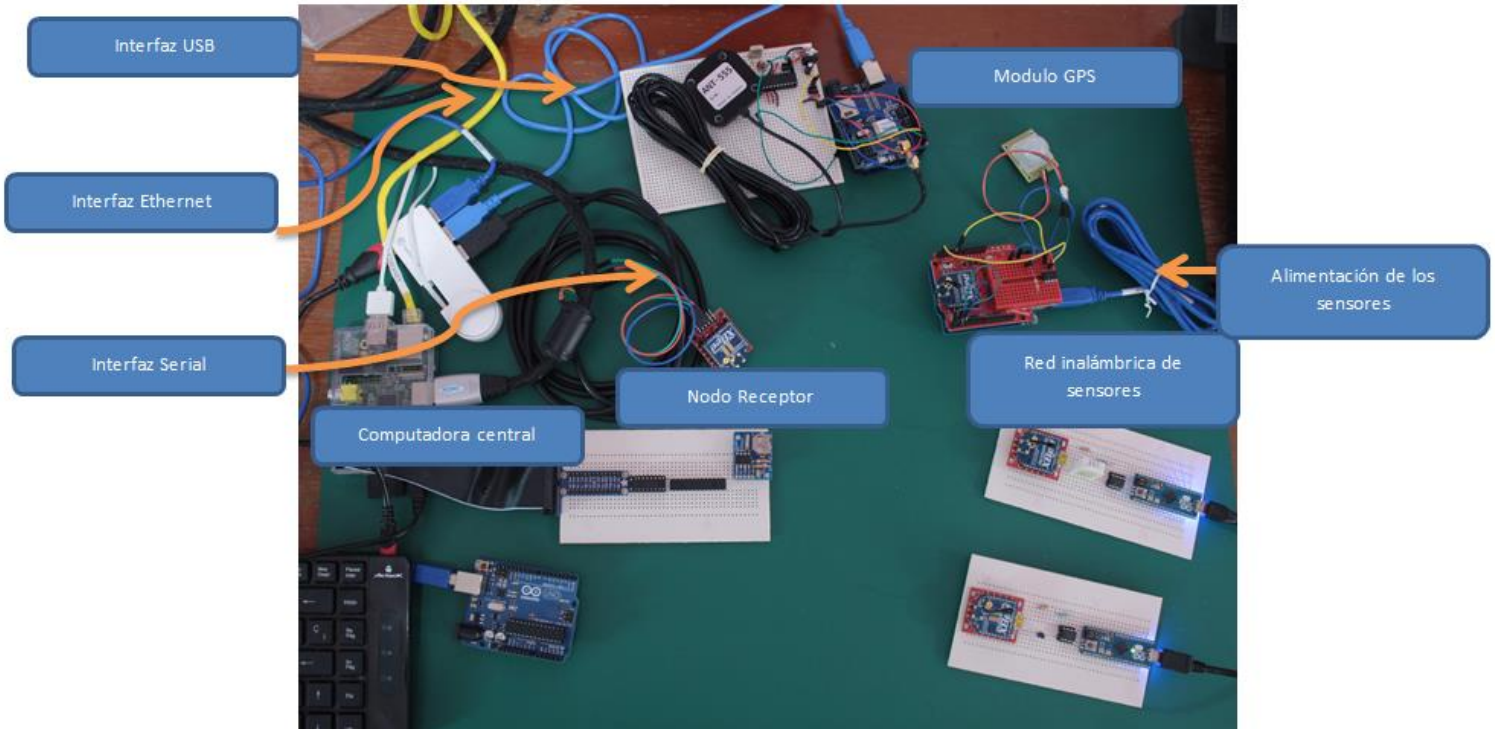


Figura 4.19 Prototipo de pruebas completo

Se realizaron pruebas donde todos los subsistemas se encontraban en operación.

4.3. Validación del prototipo

I2C muestra un resultado de velocidad de transmisión de datos lento en comparación con SPI y UART. Se utilizó también el modulo inalámbrico con protocolo I2C con resultados catastróficos. La lentitud y desconexiones fueron abundantes todo el tiempo y ni siquiera se permitió realizar una solo prueba fructífera. Esto se debe a que la utilización del módulo XBee aún no está completamente adaptada al protocolo I2C. Las bibliotecas que permiten programas una comunicación con ambos paradigmas no son lo suficientemente robustas para soportar las tramas de códigos API que propone este proyecto.

Con el software X-CTU (figura 4.20) se puede revisar la correcta recepción de las tramas provenientes de la red inalámbrica de sensores.

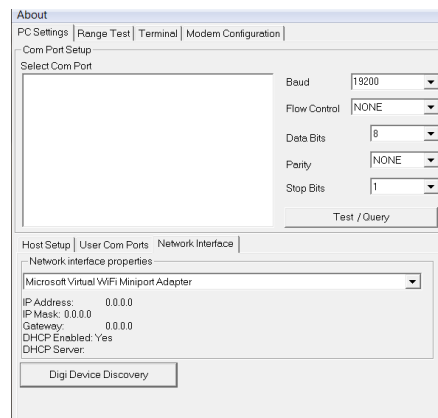


Figura 4. 20 Software X-CTU

Para este caso la figura 4.21 muestra el paquete de información enviado.



Figura 4. 21 Trama API enviada

En la figura 4.22 la recepción de la trama indicada en rojo.

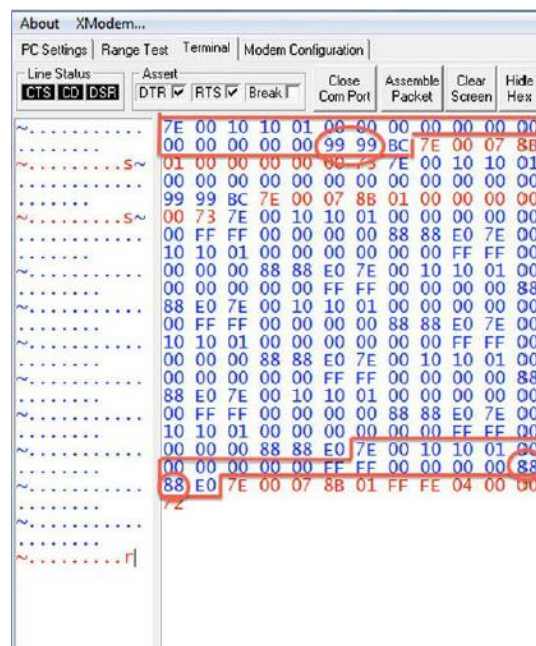


Figura 4. 22 Trama de datos recibida

Para la interfaz final para la interconectividad entre la computadora central y la red inalámbrica de sensores se tiene la serial–UART. En la figura 4.23 se observa una trama ya procesada, del lado izquierdo la temperatura en grados Celsius y del derecho en Fahrenheit.

```
pi@raspberrypi ~ $
(23.437, 74.1866)
(23.437, 74.1866)
(23.437, 74.1866)
(23.437, 74.1866)
(23.437, 74.1866)
(25.5, 77.9)
(27.25, 81.05)
(28.312, 82.9616)
(29.0, 84.2)
(29.437, 84.9866)
(29.75, 85.55)
(29.687, 85.4366)
(29.0, 84.2)
(28.25, 82.85)
```

Figura 4.23 Trama de datos ya procesada con la información accesible

Al ejecutar el script “Ejecutar.py” en la computadora central desplegará la información que está contenida en la base de datos tal y como llegó a la computadora central (toda la trama sin procesar). La figura 4.24 presenta esta orden. Los resultados fueron guardados en un archivo .CSV o archivo dividido en comas. Se optó por este formato por dos razones: es el más utilizado por compañías para manejar cantidades medianas de datos y por el espacio que requiere en memoria el cual es mucho más bajo que formatos como el .xls, además de que es un formato de bastante utilizado en los open source de Linux.

```
-----RTU slot-----[0]
[5]
RX from RTU 3 [03sd0xx0,0,0,0,46.97,19.82,0,0,21.31,] (37 bytes)
RX from RTU 3 [03sda*90.00,0.00,0.00,0.00,0.00,0.00,0.00,39.39,0.00,0.00,0,0,0,0,0,0,0,0,] (78 bytes)
-----MTU slot----- FRAME START -----[0]
[0]
TX Broadcasting Frame Sync [*0-F0xx01,01,03,a1d3a9d9]
-----RTU slot-----[0]
[1]
RX from RTU 1 [01sd0xx0,0,1,0,15.53,-50.00,0,0,0.00,] (37 bytes)
RX from RTU 1 [01sda*928.03,149.41,0.00,0.00,242.05,150.00,150.00,21.97,9568110.00,0.00,0,0,0,0,0,0,0,0,] (93 bytes)
-----MTU slot-----[0]
[2]
nothing for MTU this slot
-----RTU slot-----[0]
[3]
RX from RTU 2 [02sd0xx0,0,0,0,14.94,20.80,0,0,22.38,] (37 bytes)
RX from RTU 2 [02sda*90.00,0.00,0.00,0.00,0.00,0.00,0.00,2654127.00,0.00,0,0,0,0,0,0,0,0,] (83 bytes)
-----MTU slot-----[0]
[4]
nothing for MTU this slot
-----RTU slot-----[0]
[5]
RX from RTU 3 [03sd0xx0,0,0,0,33.11,19.82,0,0,21.31,] (37 bytes)
RX from RTU 3 [03sda*90.00,0.00,0.00,0.00,0.00,0.00,0.00,45.45,0.00,0.00,0,0,0,0,0,0,0,0,] (78 bytes)
-----MTU slot----- FRAME START -----[0]
[0]
TX Broadcasting Frame Sync [*0-F0xx01,01,03,a1d3a9d9]
-----RTU slot-----[0]
[1]
RX from RTU 1 [01sd0xx0,0,1,0,17.58,-50.00,0,0,0.00,] (37 bytes)
RX from RTU 1 [01sda*932.91,149.41,0.00,0.00,242.05,150.00,150.00,17.09,9961320.00,0.00,0,0,0,0,0,0,0,0,] (93 bytes)
-----MTU slot-----[0]
[2]
nothing for MTU this slot
-----RTU slot-----[0]
[3]
RX from RTU 2 [02sd0xx0,0,0,0,14.84,20.31,0,0,22.38,] (37 bytes)
RX from RTU 2 [02sda*90.00,0.00,0.00,0.00,0.00,0.00,0.00,2850729.00,0.00,0,0,0,0,0,0,0,0,] (83 bytes)
```

Figura 4.24 Tramas de información recibidas de los nodos

Con dicha información adquirida, queda demostrada el propósito de esta pequeña red de sensores. La tabla 4.2 resume las condiciones en las que fue probada la red inalámbrica de sensores para su validación. Ahora se puede decir que esta red inalámbrica de sensores cumple firmemente con su propósito.

Tabla 4.2 Resumen de pruebas sobre la red inalámbrica de sensores

Distancia [m]	Interior	Exterior
0-1	Excelente	Excelente
1-2	Excelente	Excelente
2-3	Buena	Excelente
3-5	Buena	Excelente
5-10	Buena	Buena
10-20	Regular	No disponible
20-30	No disponible	No disponible

Primero se asignó una dirección IP estática (192.168.1.66) a la computadora central, luego se utilizó la herramienta PING del protocolo IP que permite asegurar conexión entre dos terminales, como se constata en la figura 4.25.

```
File Edit Tabs Help
From 192.168.1.66 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.1.3 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4012ms
pipe 3
pi@raspberrypi ~ $ ping 192.168.3.254
PING 192.168.3.254 (192.168.3.254) 56(84) bytes of data:
64 bytes from 192.168.3.254: icmp_req=1 ttl=255 time=4.41 ms
64 bytes from 192.168.3.254: icmp_req=2 ttl=255 time=2.79 ms
64 bytes from 192.168.3.254: icmp_req=3 ttl=255 time=3.63 ms
64 bytes from 192.168.3.254: icmp_req=4 ttl=255 time=5.15 ms
64 bytes from 192.168.3.254: icmp_req=5 ttl=255 time=2.96 ms
64 bytes from 192.168.3.254: icmp_req=6 ttl=255 time=3.61 ms
64 bytes from 192.168.3.254: icmp_req=7 ttl=255 time=5.06 ms
64 bytes from 192.168.3.254: icmp_req=8 ttl=255 time=14.3 ms
64 bytes from 192.168.3.254: icmp_req=9 ttl=255 time=4.58 ms
64 bytes from 192.168.3.254: icmp_req=10 ttl=255 time=6.64 ms
64 bytes from 192.168.3.254: icmp_req=11 ttl=255 time=6.92 ms
64 bytes from 192.168.3.254: icmp_req=12 ttl=255 time=7.36 ms
64 bytes from 192.168.3.254: icmp_req=13 ttl=255 time=5.32 ms
64 bytes from 192.168.3.254: icmp_req=14 ttl=255 time=4.64 ms
64 bytes from 192.168.3.254: icmp_req=15 ttl=255 time=2.18 ms
64 bytes from 192.168.3.254: icmp_req=16 ttl=255 time=5.35 ms
64 bytes from 192.168.3.254: icmp_req=17 ttl=255 time=3.21 ms
```

Figura 4.25 Prueba de PING exitosa

De esta manera se alcanza el destino a la LAN del CAT y se comprueba que los paquetes ICMP son recibidos, transmitidos y leídos por la computadora central y por.

Según Wan, Ye y Song (2014) la mejor manera de brindar seguridad a la información proveniente de una red satelital, es fortificando el paquete de IP. En este caso se justifica esta recomendación y se implanta una VPN en la interfaz del

Raspberry Pi a la red al modem (en esta ocasión simulado con Arduino). En la figura 4.26 se muestra la VPN para la IP asignada en nuestra LAN.

```
GNU nano 2.2.6 File: /etc/network/interfaces
#auto lo

iface lo inet loopback
iface eth0 inet dhcp
pre-up /etc/firewall-openvpn-rules.sh
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Figura 4. 26 Activación de VPN para LAN

Los datos del GPS se pueden visualizar en la figura 4.27. Se observa la fecha y la hora en la que se realizaron las mediciones así como la latitud y la longitud donde ocurrió el evento de medición.

```
Modulo GPS/GSM
Iniciando el modulo
-----
Evento      Fecha      Hora      Latitud    Longitud
1, 5/31/2013, 13:18, 19.32737, -99.18302
2, 5/31/2013, 13:18, 19.32738, -99.18300
3, 5/31/2013, 13:18, 19.32738, -99.18302
4, 5/31/2013, 13:18, 19.32737, -99.18305
5, 5/31/2013, 13:18, 19.32738, -99.18306
6, 5/31/2013, 13:18, 19.32740, -99.18306
7, 5/31/2013, 13:20, 19.32732, -99.18306
8, 5/31/2013, 13:20, 19.32716, -99.18319
9, 5/31/2013, 13:20, 19.32712, -99.18318
10, 5/31/2013, 13:20, 19.32703, -99.18325
11, 5/31/2013, 13:20, 19.32699, -99.18326
```

Figura 4. 27 Datos recibidos por el GPS

Para este módulo se realizaron varias pruebas de ubicación, una de ellas en la Facultad de Ingeniería (figura 4.28) y otra en el Centro de Alta Tecnología Campus Juriquilla (figura 4.28).



Figura 4. 28 Ubicación en el D.F

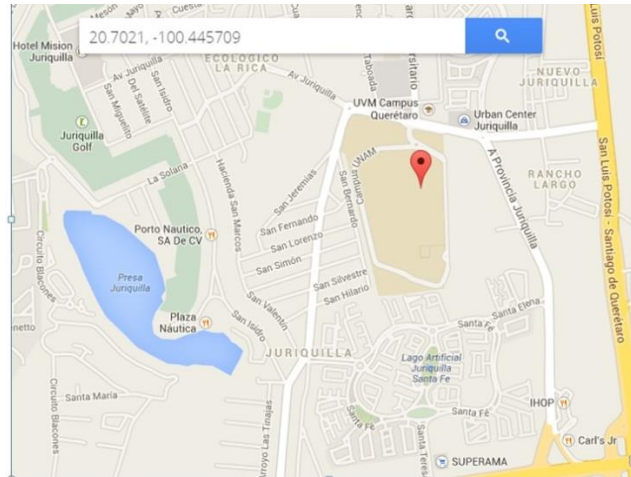


Figura 4.29 Ubicación en Juriquilla

En la simulación del Arduino como modem satelital, se programa la trama PPP y paquete IP para poder comunicar información vía serial entre Arduino y la computadora central. La trama y el paquete de datos enviados desde la computadora central al Arduino se verifica con un analizador de protocolos, en él se corrobora que todos los paquetes han sido transmitidos y recibidos de forma correcta (figura 4.33) los paquetes ICMP programados en el Raspberry Pi. Por lo tanto queda demostrado que con este tipo de tramas es posible enviar exitosamente información más compleja y se recibirá exitosamente.

No.	Time	Source	Destination	Protocol	Length	Info
90	56.47573600	192.168.1.66	192.168.3.254	ICMP	98	Echo (ping) request id=0x0b87, seq=30/7
91	56.49686300	192.168.3.254	192.168.1.66	ICMP	98	Echo (ping) reply id=0x0b87, seq=30/7
92	57.47682200	192.168.1.66	192.168.3.254	ICMP	98	Echo (ping) request id=0x0b87, seq=31/7
93	57.48029100	192.168.3.254	192.168.1.66	ICMP	98	Echo (ping) reply id=0x0b87, seq=31/7
94	58.47831100	192.168.1.66	192.168.3.254	ICMP	98	Echo (ping) request id=0x0b87, seq=32/8
95	58.48192600	192.168.3.254	192.168.1.66	ICMP	98	Echo (ping) reply id=0x0b87, seq=32/8
96	59.47991300	192.168.1.66	192.168.3.254	ICMP	98	Echo (ping) request id=0x0b87, seq=33/8
97	59.48443400	192.168.3.254	192.168.1.66	ICMP	98	Echo (ping) reply id=0x0b87, seq=33/8
98	60.48139400	192.168.1.66	192.168.3.254	ICMP	98	Echo (ping) request id=0x0b87, seq=34/8
99	60.48404900	192.168.3.254	192.168.1.66	ICMP	98	Echo (ping) reply id=0x0b87, seq=34/8
100	61.48298700	192.168.1.66	192.168.3.254	ICMP	98	Echo (ping) request id=0x0b87, seq=35/8
101	61.48828400	192.168.3.254	192.168.1.66	ICMP	98	Echo (ping) reply id=0x0b87, seq=35/8

```

Frame 69: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: Raspberr_8e:3e:21 (b8:27:eb:8e:3e:21), Dst: ca:00:0b:6e:00:1c (ca:00:0b:6e:00:1c)
Internet Protocol Version 4, Src: 192.168.1.66 (192.168.1.66), Dst: 192.168.3.254 (192.168.3.254)
Internet Control Message Protocol

0000 ca 00 0b 6e 00 1c b8 27 eb 8e 3e 21 08 00 45 00 ...n...>...E.
0010 00 54 f9 4f 40 00 40 01 ba c8 c0 a8 01 42 c0 a8 ...T.0@.@...B..
0020 03 fe 08 00 68 8a 0b 87 00 15 4c 87 73 53 cb fb ...h...L.s...
0030 0d 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 ...


```

Figura 4.30 Recepción de los paquetes exitosa

Los paquetes ICMP son el principal elemento de la utilidad PING. Se observa como todos los request fueron atendidos con su correspondiente reply. Con dicha prueba se establece una conexión de capa uno, capa dos y capa tres del modelo OSI, las referentes a capa física, enlace de datos y de red respectivamente. Por lo que se tiene una transmisión entre los nodos de una red.

En la figura 4.31 se puede ver con detalle el tamaño del paquete enviado (98[bytes]) que corresponde a los datos paramétricos guardados en un formato conveniente para su procesamiento. También cuenta la dirección de IP de la computadora central identificada en el mensaje como Raspberr_Be.

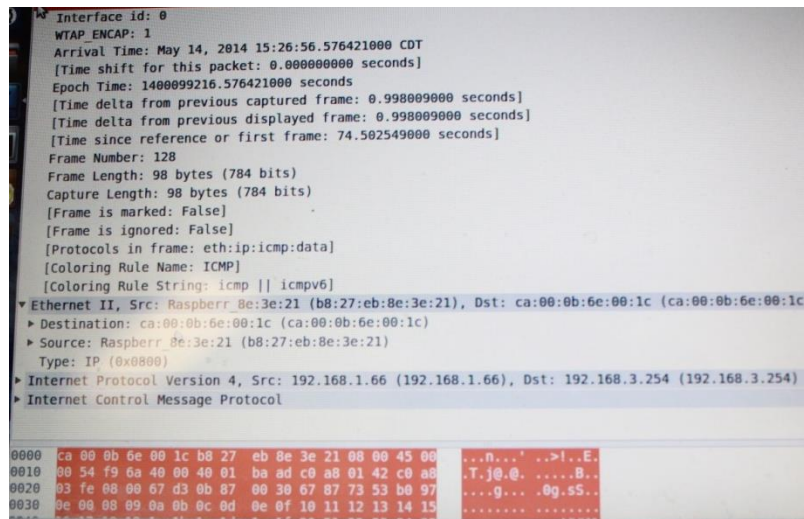


Figura 4.31 Recepción de protocolos de internet y de capa de enlace

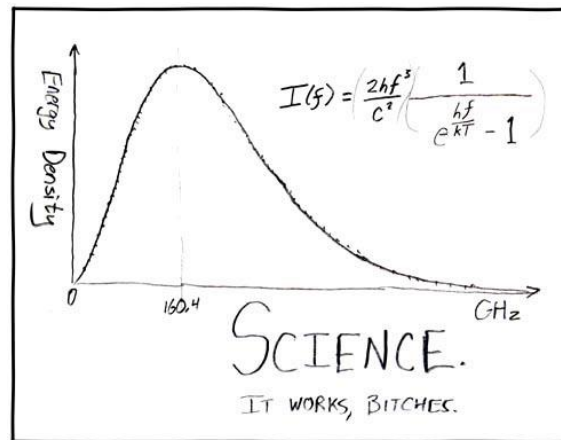
Otra cuestión importante es que se recibieron los protocolos IP y ICMP, los cuales fueron simulados también en el Raspberry Pi. IP solo se programa las cabeceras y campos importantes para enviar datos. Los campos concernientes a protocolos de capa cuatro (TCO, UDP) o cuestiones de QoS quedan fueran de los alcances de este proyecto.

Con ayuda de un software ya instalado en la computadora central, en combinación con la base de datos programada, se graficó ciertos valores de los sensores, se muestra más a detalle en la figura 4.32.



Figura 4.32 Graficas en computadora central de los datos adquiridos

Se realizaron pruebas con diferentes programas con la posibilidad de graficar pero el presente radico en su facilidad de uso y sencillez en comparación a otros softwares más sofisticados (MatLab, Mapple). Es posible ver las gráficas sobre un servidor web solo accediendo remotamente a la dirección de respaldo (:8080) de la computadora central, lo cual brinda facilidad de acceso.



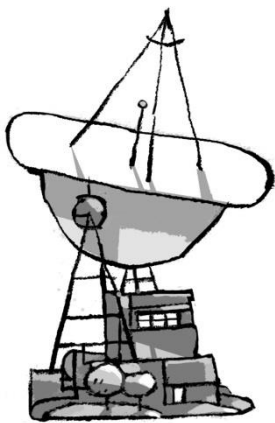
Capítulo 5. Resultados y conclusiones

*En literatura no hay temas buenos ni temas malos,
hay tan sólo temas bien o mal tratados*

Julio Cortázar

*The true measure of a man is this: how quickly can
he respond to the needs of others and how much of
himself he can give*

Philip K. Dick



El capítulo de resultados corresponde a la última fase de la metodología Vee, en la cual después de integrar los componentes y subsistemas propuestos, se valida y se verifica el prototipo. En caso contrario se replantearía la propuesta con nuevos componentes u otros requerimientos. En este capítulo se recapitulan las pruebas individuales de la parte de validación y permite tener los argumentos necesarios para construir un esquema que sea capaz de verificar el prototipo y la tesis.

Las primeras pruebas con sensores conectados directa y de manera alámbrica permitieron conocer las limitaciones de los buses de datos ISP e I2C. La desventaja de ISP es el excesivo desgaste de cables para tener varios esclavos. El I2C es un protocolo muy compacto para transmitir datos pero aún es muy lento. Las pruebas realizadas con este protocolo tardaron más que con SPI. I2C aún tiene de su lado la portabilidad, característica de gran ayuda, sin embargo, se seleccionó una interface serial como canal para que corra la información de la red inalámbrica de sensores por su.

Las ventajas que brinda el Raspberry Pi fueron evidentes: una pequeña computadora independiente que corre con un sistema operativo de Linux (código abierto), puede ser reprogramada si es necesario, extensa memoria de trabajo (RAM de 256 [Mb]) que la mayoría de los microcontroladores (no más de 56 [Mb]) no pose, capacidad de agregar memoria expandible para soportar bases de datos más elaboradas, incluir varios componentes electrónicos como sensores y circuitos integrados usando salidas y entradas digitales así como los protocolos I2C o SPI actualmente utilizados por la mayoría de sensores comerciales en el mercado.

El no tener un conversor Analogo-Digital hace que la operación de conversión se tenga que hacer con circuitos integrados, lo que aumenta el tamaño neto del prototipo, por lo que quedo descartado los sensores analógicos para usarse en el prototipo.

Al terminar de realizar las pruebas con los sensores se pudo comprobar el prototipo de sistema de adquisición de datos, conmutación y multiplex ion (sistema de telemetría), también se demostró que los medios (protocolo serial), información (tramas de información API) y materiales (sensores digitales) son adecuados para transmitir los datos paramétricos (temperatura, humedad) a un solo dispositivo (Raspberry Pi).

Se utilizó la computadora central como un nodo dentro de una red inalámbrica de sensores. Al poder enviar tramas de datos paramétricos desde un ambiente a censar y ser recibidas por el Raspberry Pi demostró su funcionalidad como nodo sensor y al mismo tiempo como computadora principal (nodo

agregador) que sea capaz de controlar una variedad de subsistemas. La principal ventaja es el poder de procesamiento que tiene, así como la capacidad de memoria, las diversas opciones de conectividad que ofrece, el uso variado de puertos USB, pero como desventaja principal está el gran consumo de energía que utiliza para funcionar con varios periféricos conectados (hasta 1[A]), lo cual para ciertas industrias como la aeroespacial es una cantidad de uso de corriente excesiva.

El Raspberry Pi tiene grandes ventajas en comunicación y mediante un módulo ZigBee puede comunicarse a distancia, capacidad que lo hace adecuado para la construcción de nodos de sensores inalámbricos y nodos de sensores web que forman parte de una red inalámbrica de sensores. El uso de ahorro de energía para dispositivos inalámbricos es mayor comparado con muchos RTU de sistemas SCADA usados comúnmente en la industria, que llegan hasta tener diferencias de potencial de 10 [V] para ser energizados. La velocidad con la son adquiridos los datos paramétricos del módulo inalámbrico varía en lapsos de milisegundos comparados la de medios alámbricos probados (SPI y 12C). Al probar el sistema de telemetría se demostró que el uso de una WSN como reemplazo de los sistema SCADA alámbricos es conveniente en términos de portabilidad, consumo de energía y velocidad de adquisidor de datos.

Se logró usar funcional y satisfactoriamente las plataformas de software libre que usan lenguajes de alto nivel para programar y tener al mismo tiempo una solución sencilla al problema. Por ello todo el proyecto se realizó utilizando el paradigma FLOSS en la parte de programación y la de COTS en la parte de hardware.

La prueba exitosa de la herramienta PING entre el Raspberry Pi y la red de datos del CAT proporciona el sistema de red conmutada por paquetes de manera exitosa. Mediante dicha conexión red LAN se tiene acceso remoto al sensor a través de internet, lo cual aumenta de manera considerable el número de usuarios con posibilidades de usar esta tecnología.

La experiencia de trabajo con el modem satelital (anterior a este proyecto) estableció las bases para el sistema de comunicación vía satelital en prototipo que se presenta en esta tesis y cuyas pruebas de recepción de paquetes de información se traducen en un correcto funcionamiento de protocolos IP y PPP.

Al hacer pruebas con el modem satelital se constató que era muy frágil conexión de la línea de vista entre el satélite y la antena receptora, a pesar de llamarse servicio 24/7, este llega a fallar, a veces con 30 [min] sin una conexión estable. También se debe tomar en cuenta el retardo en la propagación de la señal

que para satélites en órbita baja terrestre es de 3.3 [ms], esto es una desventaja comparado con el retardo de un cable Ethernet de tan solo 3[μ s].

El modem satelital puede proporcionar un bajo ancho de banda, con lo cual apenas es posible tener un acceso a internet. El ancho de banda puede mejorarse al pagar un servicio más costoso a la empresa Globalstar. En dicho servicio tanto la velocidad de transmisión como de ancho de banda aumentara. Por el momento se constata la capacidad del modem de poder ser usado como un punto de acceso de baja velocidad al poder enviar tramas de datos de un nodo LAN (Raspberry Pi) al nodo WAN (modem satelital.)

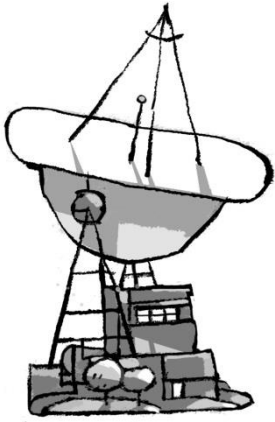
Mediante la utilización del Raspberry Pi se solucionaron problemas tanto de espacio como económicos, por el ahorro de recursos y las posibilidades de innovación que esto representa, al final se obtuvo un prototipo general económico, que resuelve el problema de comunicación por varias vías y la obtención de datos de manera eficaz, así mismo se solucionaron problemas relacionados al sistema de potencia del proyecto, esto con la finalidad de brindar mayor movilidad. Se comprobó la utilidad del sistema al brindar una manera robusta de procesar información de sensores y ya se planea implantarlo en algún ambiente que lo requiera, como en un CubeSat.

Teniendo las pruebas de diferentes subsistemas como son:

- Sistema de telemetría
- Computadora central
- Modem satelital
- Conexión de Red de datos del CAT
- Conexión a modulo GPS

Se pudo comprobar el objetivo principal de la tesis debido a que los datos adquiridos por el sistema de telemetría se enviaron a hacia un red LAN y también por el sistema PPP que simula al modem satelital. Después de estas pruebas es posible comprobar la hipótesis.

Capítulo 6. Trabajo a futuro



Así es exactamente como me gusta tomar decisiones, cuando no hay nada que perder, y todo que ganar

Etgar Keret

... God has a master plan and I guess I am in his demand

Thomas DeLonge



Después de haberse comprobada la hipótesis y de conocer las posibilidades del sistema. El prototipo puede seguir creciendo en diferentes maneras.

Se desarrollara en un proyecto de maestría un prototipo similar al presentado en esta tesis. Se mudara de una plataforma Raspberry Pi a algún sistema embebido, los sistemas ARM Cortex 9 brindan capacidades en uso de energía (amperes y potencia) que la computadora actual no tiene, por lo que se perfilan para ser un reemplazo.

La falta de un ambiente gráfico específicamente programado y diseñado para este proyecto (HMI o GUI) no estuvo dentro de las posibilidades, faltar tiempo y conocimientos más profundo de programación. La automatización y la falta de dinamismo también se presentaron. Es cierto que se pueden recabar datos y recibirlos en la computadora central de manera automática, pero para des encapsularlos y mostrarlos en la base de datos hace falta que un usuario lo realice. Lo mismo pasa con el empaquetamiento de datos hacia el modem o a la LAN. El automatizar esos campos significa un robustecimiento del prototipo.

Por otro lado, las redes inalámbricas de sensores están reemplazando a los sistemas SCADA rápidamente. El prototipo propuesto no solo es funcional. Si no también barato y puede competir en calidad con sistemas de medición que ya se encuentran en el mercado. También es posible escalar en el área inalámbrica; las redes neuronales son la nueva tendencia y las WSN ya las están adoptando por su fiabilidad y practicidad. Sea cualquiera de los casos, adaptar para la industria espacial o adaptar para la industria comercial, es necesario primero que nada, realizar pruebas de compatibilidad electromagnética, obviamente hacer pruebas para ambas industrias es un proceso distinto y todo un vasto campo de estudio.

Elon Musk anuncio un sistema de internet por satélite a inicios de 2015, con el objetivo de llevar el internet a cada rincón del mundo, sobre todo a los lugares más escaso y remotos, esto con la premisa de que el internet ya es derecho humano. Dichos satélites son de bajo costo y consumo ¿Por qué no hacer eso? La principal ventaja del modem que se probó es brindar un servicio de internet de baja velocidad mediante una constelación satelital, se puede adaptar el servicio para llevar el internet a sitios de difícil acceso para las comunicaciones alámbricas y así, entrar en la vanguardia tecnológica que tanto exige la actualidad.

Finalmente solo queda por trabajar aún más en un nuevo prototipo, Escalarlo y mejorarlo.



Apéndices

Código

```
// *****CODIGO REALIZADO POR HAZAELMIGUEL MOCTEZUMA FLORE*****//
#include <stdlib.h>
#include <string.h>
#include <math.h> int IP = 0x0021;
#include <SoftwareSerial.h>
int IPCP = 0x8021;
int REQ = 1;
int ACK = 2;
int NAK = 3;
int REJ = 4;
int TERM = 5;
int CCP = 0x80FD;
int LCP = 0xC021;
int PAP = 0xC023;
int Rx = 46;
int Tx = 46;
int rx, tx_in, tx_fin, c;
int checksum1, checksum2;

char dir1, dir2, dir3, dir4;
char tx1[Rx+1];
char rx1[Tx+1];
char num;

int TIME;
int TIME_SET(a) TIME=a;

static void interrupt (void) {
  if (cic) {
    cic = 100;
    TIME++;
  }
}

int calc( int c) {
  char i;
  c &= 0xFF;
  for (i=0;i<8;i++) {
    if (c&1) {
      c /= 2;
      c ^= 0x8408;
    } else c /= 2;
  }
  return c;
}

void add(char c) {
  checksum2 = calc(c^checksum2) ^ (checksum2/256);
}

tx1[tx_in] = c;
tx_in++;
}

void ensamblar(int paquete, char req, char num*final_header) {
  int length;
  tx_in = 1;
  tx1[0] = ' ';
  checksum2 = 0xFFFF;
}
```




```
add(0xFF);
tx1.add(3);
tx1.add(paquete/256);
tx1.add(paquete&255);
if (paquete==P) {
tx1.add(0x45);

tx1.add((*final_header)+12);
tx1.add(0x88);
tx1.add(0x10);
tx1.add(0x40);
tx1.add(127);
tx1.add(num);
length = 0x45+0x88+0x40+127+dir1+dir3+final_header[1]+final_header[3];
paquete = *final_header + 12 + 0x10 + num + dir2 + dir4 + final_header[2] + final_header[4];

paquete += length/256;
length = (length&255) + paquete/256;
paquete = (paquete&255) + length/256;
length += paquete/256;
paquete.add(dir1);
paquete.add(dir2);
paquete.add(dir3);
paquete.add(dir4);
length = *final_header - 4;
final_header++;
} else {
final_header.add(req);
final_header.add(num);
final_header.add(0);
length = *final_header - 3;
}
while (length) {
length--;
add(*final_header);
final_header++;
}
length = ~checksum2;
checksum2.add(length&255);
checksum2.add(length/256);
tx_fin=tx_in;
tx_in = 0;
}

char opciones(int opc){
int size;
ind1 = 8,
ind2 = 8;
char pas = 3;
size = rx1[7]+4;
if (size>Rx) size=Rx;
while (ind1<size) {
if (rx1[ind1]==3 && rx1[ind1+2]!=0x80 && rx1[2]==0xc2)
pas&=0xfd;
if (!(1<<(rx1[ind1]-1))&opc)
pas=0;
ind1 += rx1[ind1+1];
}
if (!(pas&2)) {
if (pas&1) {
opc=0xfffb;
}
for (ind1=8; ind1<size;) {
if (!(1<<(rx1[ind1]-1))&opc) {
```



```
for (pas=rx1[ind1+1]; ind1<size && pas; ind1++) {
rx1[ind2]=rx1[ind1];
ind2++;
pas--;
} else {
ind1+=rx1[ind1+1];
}
}
rx1[7] = ind2-4;
pas=0;
if (opc==0xfffb) pas=1;
}
return pas;
}

char send-wait(const char *send, const char *wait, int timeout) {
dir2=dir3=0;
for (TIME_SET(0); TIME<timeout; ) {
if (!dir1) PORTB&=0xFB;
else if (dir1==1) {
if (TIME&4) PORTB&=0xFB;
else PORTB|=4;
} else PORTB|=4;
if (serial_rx_ready()) {
PORTB|=1;
dir4 = serial_get();
if (serial_error()) serial_fix();
if (wait[dir2]==dir4) dir2++;
else dir2=0;
PORTB&=0xFE;
if (!wait[dir2]) return 1;
} else if (send[dir3] && (serial_tx_ready())) {
if (send[dir3]=='|') {
if (TIME>100) {
TIME_SET(0);
dir3++;
}
} else {
PORTB|=2;
TIME_SET(0);
serial_send(send[dir3]);
dir3++;
}
PORTB&=0xFD;
if (!send[dir3] && !(*wait))
return 1;
}
}
return 0;
}

void flash(void) {
for (TIME_SET(0);) {
if (TIME&8) PORTB|=0x07;
else PORTB&=0xF8;
if (TIME>3000) PORTB&=0xF7;
}
}

void pulse(unsigned char data) {
for(num=0;num<9;) {
if (TIME<100) PORTB&=0xFB;
else if (num<8) PORTB|=4;
if (TIME>200 || (!(data&0x80) && TIME>120)) {
TIME_SET(0);
num++;
data<<=1;
}
```



```
}
}
}
void main(void) {
signed int c;
int paquete = 0;
char estado = 0;
char expand = 0;
opc=0x85;
TIME_SET(0);
while (TIME<25);
for(num=1;;num++) {
if (num==10)
dir1=0;
dir1=1;
}

for (TIME_SET(0);) {
if (TIME>7000 || num>20);
if (serial_rx_ready()) {
PORTB ^=1;
c = serial_get();
if (serial_error()) serial_fix();
if (c == 0x7E) {
if (rx && (checksum1==0xF0B8))
paquete = rx1[2]*256 + rx1[3];
rx = 0;
checksum1 = 0xFFFF;
} else if (c == 0x7D) {
expand |= 1;
} else {
if (expand&1) {
c ^= 0x20;
expand &= 0xFE;
}
if (rx==0 && c!=0xff) rx1[rx++] = 0xff;
if (rx==1 && c!=3) rx1[rx++] = 3;
if (rx==2 && (c&1)) rx1[rx++] = 0;
rx1[rx++] = c;
if (rx>Rx) rx = Rx;
checksum1 = calc(c^checksum1) ^ (checksum1/256);
}
} else if (tx_fin && (serial_tx_ready())) { /

c = tx1[tx_in];
if (tx_in==tx_fin) {
tx_fin=0;
c='~';
} else if (expand&2) {
c^=0x20;
expand &= 0xFD;
tx_in++;
} else if (c<0x20 || c==0x7D || c==0x7E) {
expand |= 2;
c = 0x7D;
} else {
if (!tx_in) c='~';
tx_in++;
}
serial_send(c);
}
if (paquete == LCP) {
switch (rx1[4]) {
case REQ:
estado &= 0xfd;
if (c=opciones(0x00c6)) {
```



```
if (c>1) {
c = ACK;
if (estado<3) estado |= 2;
} else {
rx1[10]=0xc0;
c = NAK;
}
} else {
c = REJ;
}
TIME_SET(0);
ensamblar(LCP,c,rx1[5],rx1+7);
break;
case ACK:
if (rx1[5]!=num) break;
if (estado<3) estado |= 1;
break;
case NAK:
estado &= 0xfe;
break;
case REJ:
estado &= 0xfe;
break;
case TERM:
break;
}
if (estado==3) estado = 4;
} else if (paquete == PAP) {
switch (rx1[4]) {
case REQ:
break;
case ACK:
estado = 5;
break;
case NAK:
break;
}
} else if (paquete == IPCP) {
switch (rx1[4]) {
case REQ:
if (opciones(0x0004)) {
c = ACK;
estado = 6;
} else {
c = REJ;
}
ensamblar(IPCP,c,rx1[5],rx1+7);
break;
case ACK:
if (rx1[5]==num) {
estado = 7;
pulse(dir1);
pulse(dir2);
pulse(dir3);
pulse(dir4);
PORTB|=4;
TIME_SET(5800);
}
break;
case NAK:
dir1 = rx1[10];
dir2 = rx1[11];
dir3 = rx1[12];
dir4 = rx1[13];
ensamblar(IPCP,REQ,rx1[5],rx1+7);
break;
}
```



```
case REJ:
break;
case TERM:
break;
}
} else if (paquete == IP) {
if (estado<7 || (rx1[19]==dir4 && rx1[18]==dir3 &&
rx1[17]==dir2 && rx1[16]==dir1)) {

} else if (rx1[13]==1) {
if (rx1[24]==8) {
rx1[20]=rx1[16];

rx1[21]=rx1[17];
rx1[22]=rx1[18];
rx1[23]=rx1[19];
rx1[19]=16;
rx1[24]=0;
paquete = rx1[28]+rx1[30];
rx1[26] = paquete&255;
rx1[27] = paquete/256;
paquete = rx1[27]+rx1[29]+rx1[31];
rx1[27] = paquete&255;
paquete = paquete/256 + rx1[26];
rx1[26] = paquete&255;
rx1[27] += paquete/256;
rx1[26] = ~rx1[26];
rx1[27] = ~rx1[27];
ensamblar(IP,0,1,rx1+19);
} else if (rx1[24]==0) {
if ((rx1[28]|rx1[30]|rx1[31])+rx1[29]==1)
PORTB&=0xF7;
}
}
} else if (paquete == CCP) {
switch (rx1[4]) {
case REQ:
c = REJ;
if (opciones(0x0004)) c = ACK;
ensamblar(CCP,c,rx1[5],rx1+7);
}
} else if (paquete) {
} else if (!tx_fin && (estado==0 || estado==2) && TIME>100) {
num++;
TIME_SET(0);
ensamblar(LCP,REQ,num,"\\x0E\\x02\\x06\\x00\\x0A\\x00\\x07\\x02\\x08\\x02");

} else if (!tx_fin && estado == 4 && TIME>100) {

TIME_SET(0);
num++;

ensamblar(PAP,REQ,num,"\\x14\\x06\\xuserid\\x08password");
} else if (!tx_fin && estado == 6 && TIME>100) {
num++;
TIME_SET(0);
ensamblar(IPCP,REQ,num,"\\xA\\x3\\x6\\x0\\x0\\x0");
} else if (!tx_fin && estado == 7 && TIME>3000) {
TIME_SET(0);
num++;
ensamblar(IP,0,1,"\\x10\\xCF\\xA1\\x75\\x43\\x8\\x0\\xF7\\xFE\\x0\\x1\\x0\\x0");
}
paquete = 0;
}
}
import pprint
```



```
import serial
import xbee
SERIAL_PORT = '/dev/ttyAMA0'
BAUD_RATE = 9600

ser_port = serial.Serial(SERIAL_PORT, BAUD_RATE)

xbee1 = xbee.zigbee.ZigBee(ser_port)

while True:
    try:
        data_samples = xbee1.wait_read_frame()
        address = data_samples['source_addr_long']
        samples = data_samples['samples'][0]

        temp_c = ((samples['adc-3'] * 1200.0 / 1024.0) - 500.0) / 10.0

        temp_f = ((temp_c * 9.0) / 5.0) + 32.0

        print(" Temperatura: {} c, "
              "{1} f".format(temp_c, temp_f))
        volts = (float(samples['adc-7']) * (1200.0 / 1024.0)) / 1000.0
    except KeyboardInterrupt:
        break
    ser_port.close()
    #include <XBee.h>
    #include <SoftwareSerial.h>
    uint8_t recv = 2;
    uint8_t trans = 3;
    SoftwareSerial soft_serial(recv, trans);
    XBee xbee = XBee();
    ZBRxIoSampleResponse ioSample = ZBRxIoSampleResponse();
    void setup() {
        Serial.begin(9600);
        while (!Serial);
        start
        soft_serial.begin(9600);
        xbee.setSerial(soft_serial);
    }
    void get_address(ZBRxIoSampleResponse *ioSample) {
        Serial.print(ioSample->getRemoteAddress64().getMsb(), HEX);
        Serial.print(ioSample->getRemoteAddress64().getLsb(), HEX);
        Serial.println("");
    }

    void get_temperature(ZBRxIoSampleResponse *ioSample) {
        float adc_data = ioSample->getAnalog(3);
        Serial.print("Temperatura: ");
        float temperaturaC = ((adc_data * 1200.0 / 1024.0) - 500.0) / 10.0;
        Serial.print(temperaturaC);
        Serial.print("c, ");
        float temperaturaF = ((temperaturaC * 9.0)/5.0) + 32.0;
        Serial.print(temperaturaF);
        Serial.println("f");
    }

    void get_supply_voltage() {
        Serial.print("Supply voltage is ");
        int ref = xbee.getResponse().getFrameData()[17] << 8;
        ref += xbee.getResponse().getFrameData()[18];
        float volts = (float(ref) * float(1200.0 / 1024.0))/1000.0;
    }
    void loop() {
        xbee.readPacket();
```



```

if (xbee.getResponse().isAvailable()) {

if (xbee.getResponse().getApild() == ZB_IO_SAMPLE_RESPONSE) {

xbee.getResponse().getZBRxIoSampleResponse(ioSample);

get_address(&ioSample);
get_temperature(&ioSample);
get_supply_voltage();
}
else {
Serial.print(xbee.getResponse().getApild(), HEX);
}
} else if (xbee.getResponse().isError()) {
Serial.print("Error");
Serial.println(xbee.getResponse().getErrorCode());
}
}

```

Documentos de control de Interfaz

Tipo de Protocolo	Interfaz inicial	Interfaz final	Descripción	Otra información
Capa de Red OSI				
<i>IP</i>	<i>Modem Satelital</i>	<i>Raspberry Pi</i>	<i>Genera paquetes de IP para enviar</i>	<i>Dirección IP depende del proveedor de servicio satelital</i>
<i>IP</i>	<i>Red de datos del CAT</i>	<i>Raspberry Pi</i>	<i>Genera paquetes de IP para enviar</i>	<i>Dirección IP depende de administrador de Red del CAT</i>
<i>ZigBee</i>	<i>Sistema de telemetría</i>	<i>Raspberry Pi</i>	<i>Genera paquetes API</i>	<i>Protocolo embebido en modulo inalámbrico</i>
Capa de enlace OSI				
<i>PPP</i>	<i>Modem Satelital</i>	<i>Raspberry Pi</i>	<i>Genera tramas de datos sin dirección MAC</i>	<i>Solo un punto de destino</i>
<i>IEEE 802.3</i>	<i>Red de datos del CAT</i>	<i>Raspberry Pi</i>	<i>Genera tramas con dirección MAC</i>	<i>Protocolo automático</i>
<i>IEEE 802.15.4</i>	<i>Sistema de telemetría</i>	<i>Raspberry Pi</i>	<i>Establece tramas para datos paramétricos</i>	<i>Protocolo embebido en modulo inalámbrico</i>
Capa física OSI				
<i>Serial 232</i>	<i>Modem Satelital</i>	<i>Raspberry Pi</i>	<i>Enlace WAN</i>	<i>Utiliza puertos mini UART del Raspberry Pi</i>
<i>Ethernet</i>	<i>Red de datos del CAT</i>	<i>Raspberry Pi</i>	<i>Enlace LAN</i>	<i>Acceso a red con privilegios de administrador</i>
<i>Serial 232</i>	<i>Sistema de telemetría</i>	<i>Raspberry Pi</i>	<i>Enlace WLAN</i>	<i>Adquisición de datos paramétricos para enviar</i>
<i>USB</i>	<i>Modulo GPS</i>	<i>Raspberry Pi</i>	<i>Enlace a constelación de satélites GPS</i>	<i>Proporcionar datos específicos de ubicación</i>

Hojas de datos

74HC595; 74HCT595

8-bit serial-in, serial or parallel-out shift register with output latches; 3-state

Rev. 7 — 26 January 2015

Product data sheet

1. General description

The 74HC595; 74HCT595 are high-speed Si-gate CMOS devices and are pin compatible with Low-power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard No. 7A.

The 74HC595; 74HCT595 are 8-stage serial shift registers with a storage register and 3-state outputs. The registers have separate clocks.

Data is shifted on the positive-going transitions of the shift register clock input (SHCP). The data in each register is transferred to the storage register on a positive-going transition of the storage register clock input (STCP). If both clocks are connected together, the shift register will always be one clock pulse ahead of the storage register.

The shift register has a serial input (DS) and a serial standard output (Q7S) for cascading. It is also provided with asynchronous reset (active LOW) for all 8 shift register stages. The storage register has 8 parallel 3-state bus driver outputs. Data in the storage register appears at the output whenever the output enable input (OE) is LOW.


2. Features and benefits

- 8-bit serial input
- 8-bit serial or parallel output
- Storage register with 3-state outputs
- Shift register with direct clear
- 100 MHz (typical) shift out frequency
- ESD protection:
 - ◆ HBM JESD22-A114F exceeds 2000 V
 - ◆ MM JESD22-A115-A exceeds 200 V
- Multiple package options
- Specified from $-40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$ and from $-40\text{ }^{\circ}\text{C}$ to $+125\text{ }^{\circ}\text{C}$

3. Applications

- Serial-to-parallel data conversion
- Remote control holding register





MICROCHIP 24AA512/24LC512/24FC512

512K I²C™ Serial EEPROM

Device Selection Table

Part Number	Vcc Range	Max. Clock Frequency	Temp. Ranges
24AA512	1.7-5.5V	400 kHz ⁽¹⁾	I
24LC512	2.5-5.5V	400 kHz	I, E
24FC512	1.7-5.5V	1 MHz ⁽²⁾	I

Note 1: 100 kHz for Vcc < 2.5V
Note 2: 400 kHz for Vcc < 2.5V

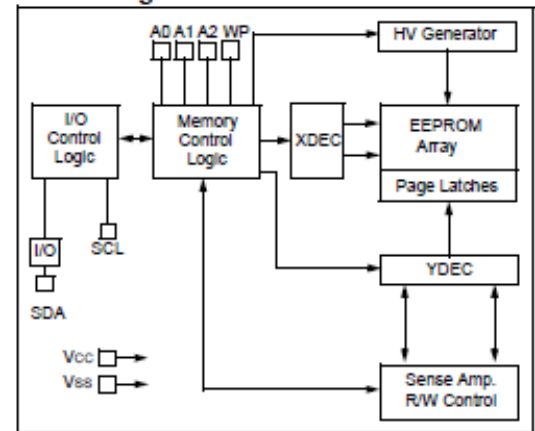
Features:

- Single Supply with Operation down to 1.7V for 24AA512 and 24FC512 Devices, 2.5V for 24LC512 Devices
- Low-Power CMOS Technology:
 - Active current 400 uA, typical
 - Standby current 100 nA, typical
- 2-Wire Serial Interface, I²C™ Compatible
- Cascadable for up to Eight Devices
- Schmitt Trigger Inputs for Noise Suppression
- Output Slope Control to Eliminate Ground Bounce
- 100 kHz and 400 kHz Clock Compatibility
- Page Write Time 5 ms max.
- Self-Timed Erase/Write Cycle
- 128-Byte Page Write Buffer
- Hardware Write-Protect
- ESD Protection >4000V
- More than 1 Million Erase/Write Cycles
- Data Retention > 200 years
- Packages Include 8-lead PDIP, SOIJ, SOIC, TSSOP, DFN, Chip Scale and 14-lead TSSOP
- Pb-Free and RoHS Compliant
- Temperature Ranges:
 - Industrial (I): -40°C to +85°C
 - Automotive (E): -40°C to +125°C

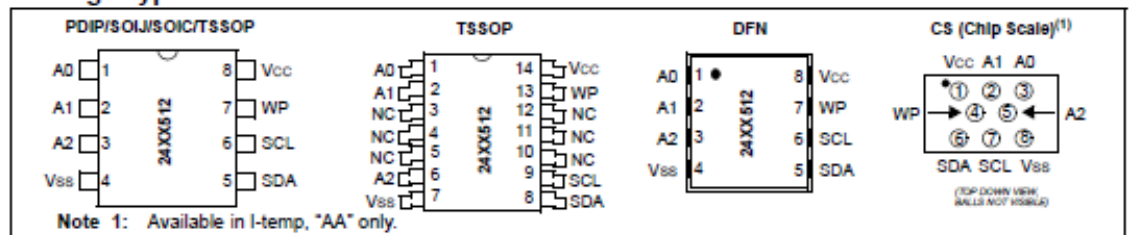
Description:

The Microchip Technology Inc. 24AA512/24LC512/24FC512 (24XX512*) is a 64K x 8 (512 Kbit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.7V to 5.5V). It has been developed for advanced, low-power applications such as personal communications and data acquisition. This device also has a page write capability of up to 128 bytes of data. This device is capable of both random and sequential reads up to the 512K boundary. Functional address lines allow up to eight devices on the same bus, for up to 4 Mbit address space. This device is available in the standard 8-pin plastic DIP, SOIJ, SOIC, TSSOP, DFN, and 14-lead TSSOP packages. The 24AA512 is also available in the 8-lead Chip Scale package.

Block Diagram



Package Type



* 24XX512 is used in this document as a generic part number for the 24AA512/24LC512/24FC512 devices.

MICROCHIP MCP23017/MCP23S17

16-Bit I/O Expander with Serial Interface

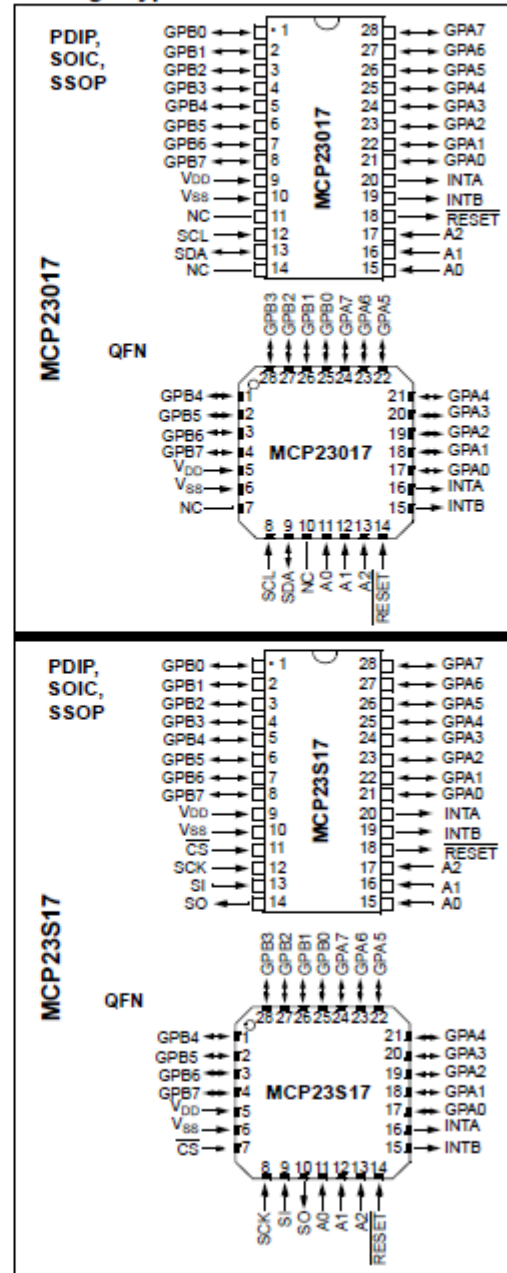
Features

- 16-bit remote bidirectional I/O port
 - I/O pins default to input
- High-speed I²C™ interface (MCP23017)
 - 100 kHz
 - 400 kHz
 - 1.7 MHz
- High-speed SPI interface (MCP23S17)
 - 10 MHz (max.)
- Three hardware address pins to allow up to eight devices on the bus
- Configurable interrupt output pins
 - Configurable as active-high, active-low or open-drain
- INTA and INTB can be configured to operate independently or together
- Configurable interrupt source
 - Interrupt-on-change from configured register defaults or pin changes
- Polarity Inversion register to configure the polarity of the input port data
- External Reset input
- Low standby current: 1 µA (max.)
- Operating voltage:
 - 1.8V to 5.5V @ -40°C to +85°C
 - 2.7V to 5.5V @ -40°C to +85°C
 - 4.5V to 5.5V @ -40°C to +125°C

Packages

- 28-pin PDIP (300 mil)
- 28-pin SOIC (300 mil)
- 28-pin SSOP
- 28-pin QFN

Package Types



Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

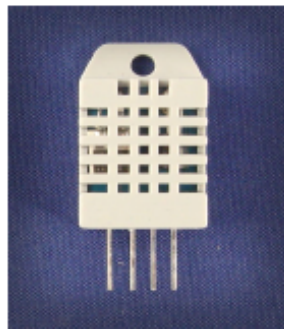
<http://www.aosong.com>

Email: thomasliu198518@yahoo.com.cn sales@aosong.com

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China

Digital-output relative humidity & temperature sensor/module

AM2303



Capacitive-type humidity and temperature module/sensor

1. Feature & Application:

- * Full range temperature compensated
- * Relative humidity and temperature measurement
- * Calibrated digital signal
- * Outstanding long-term stability
- * Extra components not needed
- * Long transmission distance
- * Low power consumption
- * 4 pins packaged and fully interchangeable

2. Description:

AM2303 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable AM2303 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	AM2303
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer humidity capacitor & DS18B20 for detecting temperature
Measuring range	humidity 0-100%RH; temperature -40~125Celsius



DS18B20 Programmable Resolution 1-Wire Digital Thermometer

DESCRIPTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to $+125^{\circ}\text{C}$ and is accurate to $\pm 0.5^{\circ}\text{C}$ over the range of -10°C to $+85^{\circ}\text{C}$. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

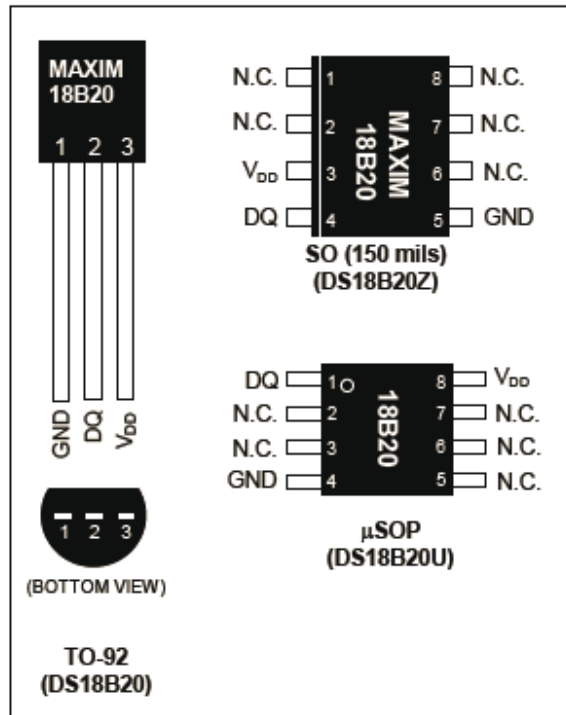
Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to $+125^{\circ}\text{C}$ (-67°F to $+257^{\circ}\text{F}$)
- $\pm 0.5^{\circ}\text{C}$ Accuracy from -10°C to $+85^{\circ}\text{C}$
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)

- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin μSOP , and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS



1-Wire is a registered trademark of Maxim Integrated Products, Inc.

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

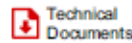
REV: 042208



Product Folder



Sample & Buy



Technical Documents



Tools & Software



Support & Community



MAX232, MAX232I

SLLS047M – FEBRUARY 1989 – REVISED NOVEMBER 2014

MAX232x Dual EIA-232 Drivers/Receivers

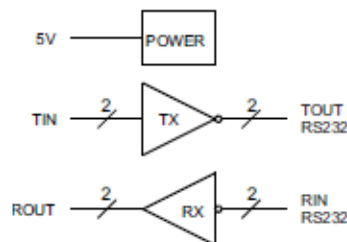
1 Features

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates up to 120 kbit/s
- Two Drivers and Two Receivers
- \pm 30-V Input Levels
- Low Supply Current: 8 mA Typical
- ESD Protection Exceeds JESD 22
 - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202 Device

2 Applications

- TIA/EIA-232-F
- Battery-Powered Systems
- Terminals
- Modems
- Computers

4 Simplified Schematic



3 Description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept \pm 30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels.

Device Information⁽¹⁾

ORDER NUMBER	PACKAGE (PIN)	BODY SIZE
MAX232x	SOIC (16)	9.90 mm \times 3.91 mm
	SOIC (16)	10.30 mm \times 7.50 mm
	PDIP (16)	19.30 mm \times 6.35 mm
	SOP (16)	10.3 mm \times 5.30 mm

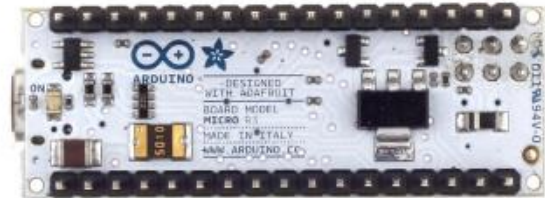
(1) For all available packages, see the orderable addendum at the end of the datasheet.



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.



Arduino Micro



The Arduino Micro is a microcontroller board based on the ATmega32u4. It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a micro USB cable to get started.

The Micro is similar to the Arduino Leonardo in that the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Micro to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port. It also has other implications for the behavior of the board; these are detailed on the getting started page.

The Arduino Micro has been co-designed with Adafruit.

Technical specification

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (recommended)	7–12V
Input Voltage (limits)	6–20V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz



MCP3004/3008

2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface

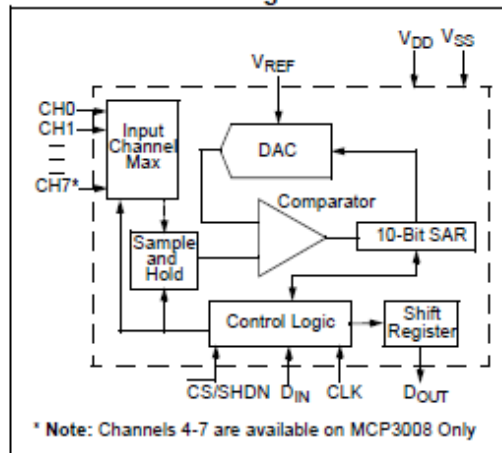
Features

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- 4 (MCP3004) or 8 (MCP3008) input channels
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200 ksp/s max. sampling rate at $V_{DD} = 5V$
- 75 ksp/s max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
- 5 nA typical standby current, 2 μA max.
- 500 μA max. active current at 5V
- Industrial temp range: -40°C to +85°C
- Available in PDIP, SOIC and TSSOP packages

Applications

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

Functional Block Diagram

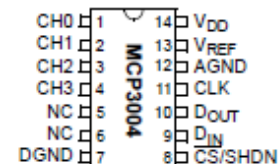


Description

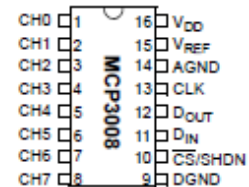
The Microchip Technology Inc. MCP3004/3008 devices are successive approximation 10-bit Analog-to-Digital (A/D) converters with on-board sample and hold circuitry. The MCP3004 is programmable to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are specified at ± 1 LSB. Communication with the devices is accomplished using a simple serial interface compatible with the SPI protocol. The devices are capable of conversion rates of up to 200 ksp/s. The MCP3004/3008 devices operate over a broad voltage range (2.7V - 5.5V). Low-current design permits operation with typical standby currents of only 5 nA and typical active currents of 320 μA . The MCP3004 is offered in 14-pin PDIP, 150 mil SOIC and TSSOP packages, while the MCP3008 is offered in 16-pin PDIP and SOIC packages.

Package Types

PDIP, SOIC, TSSOP



PDIP, SOIC



XBee®/XBee-PRO® ZB RF Modules

ZigBee RF Modules by Digi International

Models: XBEE2, XBEEPRO2, PRO S2B

Hardware: S2 and S2B

Firmware Versions:

- 20xx - Coordinator - AT/Transparent Operation
- 21xx - Coordinator - API Operation
- 22xx - Router - AT/Transparent Operation
- 23xx - Router - API Operation
- 28xx - End Device - AT/Transparent Operation
- 29xx - End Device - API Operation



Digi International Inc.
11001 Bren Road East
Minnetonka, MN 55343
877 912-3444 or 952 912-3444
<http://www.digi.com>

90000976_K
March 2012

Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

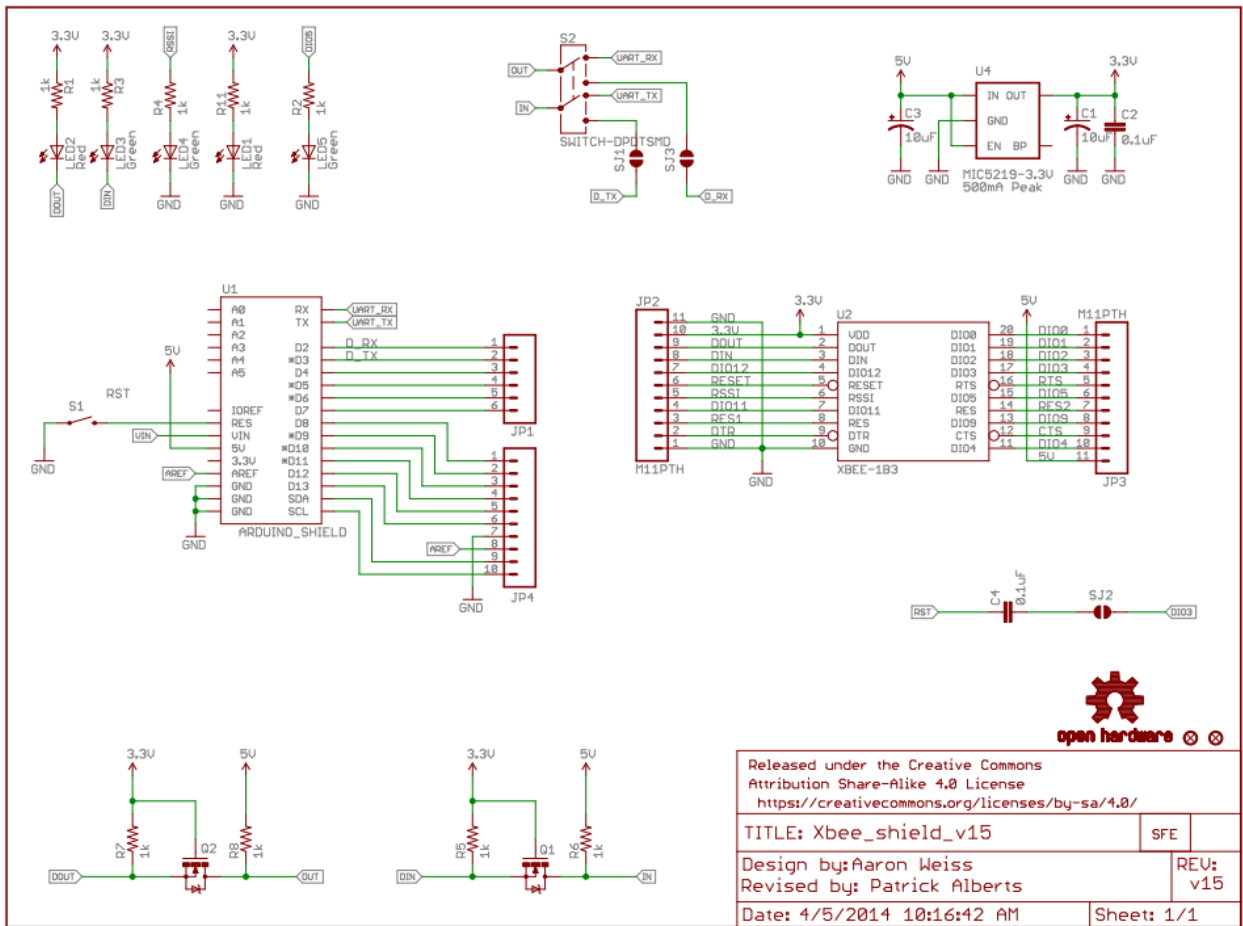
Page 7



radiospares

RADIONICS







Publicaciones



IV FORO DE DIVULGACIÓN CIENTÍFICA Y TECNOLÓGICA
"INTERFACULTADES 2014"

Implementación del prototipo del sistema de adquisición de datos para plataforma satelital

C. Hazael Miguel Moctezuma Flores^a

^aCentro de Alta Tecnología de la Facultad de Ingeniería, Universidad Nacional Autónoma de México, Distrito Federal, México. mighaza@comunidad.unam.mx

RESUMEN

Se presenta un prototipo experimental que tiene la función de monitorear parámetros físicos (temperatura, humedad, etc) así mismo tiene la capacidad de enviar los datos recabados por varios métodos: vía satelital, inalámbricamente o usando una red cableada Ethernet.

1. INTRODUCCIÓN

Los sistemas de supervisión, control y adquisición de datos han evolucionado vertiginosamente en los últimos años. Han pasado de ser sistemas estáticos que no tienen mucha interacción (más que la estrictamente necesaria) con el usuario, a ser sistemas complejos y dinámicos, ya no solo interactúan con los usuarios, también lo hacen con otros dispositivos inteligentes, la tecnología M2M (Machine to Machine) está presente en este ámbito. Los antiguos sistemas de adquisición de datos están siendo reemplazados por redes inalámbricas de sensores (Wireless Sensors Network) que brindan un mayor rango de alcance, modos de configuración amigable, seguridad más robusta, y con mayores opciones de escalabilidad, como por ejemplo usar un enfoque de redes neuronales artificiales para una administración eficiente de la red.

Las aplicaciones que utilizan SCADA's, generalmente se tiene como HMI (Human Machine Interface) una computadora para procesar los datos y una red cableada Ethernet para poder enviar la información recabada, ya sea en una base de datos o por envío en tiempo real. Esto es bastante útil cuando se cuenta con dicha posibilidad, la ventaja que ofrece es de velocidad en transmisión full-duplex, el estándar en Ethernet es de 100 [Mb/s]. Sin embargo, existen muchos casos donde el objetivo a monitorear se encuentra en zonas de difícil acceso, donde es imposible tener dicha conexión, por ejemplo: en monitoreo de bombas y reservas acuíferas, estaciones meteorológicas alejadas, en áreas de administración de flotillas. Una solución viable a este problema es la utilización de un modem satelital que brinde transmisión de datos y sea fácilmente adaptable a cualquier tipo de sistema de adquisición de datos o de telemetría.

El raspberry pi es una tarjeta de desarrollo que cuenta con la adaptabilidad para integrar más dispositivos periféricos, provee un lenguaje propio para su programación, interfaz gráfica amigable y un procesador con capacidad superior a los microcontroladores. Con este tipo de equipo se ahorra espacio y recursos, ya que cuenta con las especificaciones necesarias para funcionar como una computadora de bajo costo, y que se puede adaptar específicamente a las necesidades de proyectos de electrónica.

2. TEORÍA

A. Esquema general del sistema

Se desarrolló un prototipo que tiene como computadora central al Raspberry Pi y con la finalidad de procesar y enviar los datos recabados de manera satelital o por una red Ethernet. El esquema general se presenta en la figura 1.

Bibliografía

- [1] Amarawardhana, Chakshika. Dayananada, Sharma. Porawagama, Harshana. (2009). Case Study of WSN as a Replacement for SCADA. *Fourth International Conference on Industrial and Information Systems*. 11 (2), 49-54.
- [2] Antunes, Sandy (2013). *DIY Instruments for Amateur Space*. California: O'Reilly. 120.
- [3] Antunes, Sandy (2013). *DIY Satellite Platforms*. California: O'Reilly.
- [4] Antunes, Sandy (2013). *Surviving Orbit the DIY Way*. California: O'Reilly. 90.
- [5] Aslam, Muhammad & Rea, Susan. (2012). Service Provisioning For TheWSN Cloud. *IEEE Fifth International Conference on Cloud Computing*. 11 (9), 962-969.
- [6] Axelson, Jan (2007). *Serial Port Complete* . Madison: IPG. 400.
- [7] Banerjee, Soham. Sethia, Divyashikha. Mittal Tanuj & Arora Ujjwal. (2013). Secure Sensor Node with Raspberry Pi. *IMPACT*. 7 (5), 26-30
- [8] Benatia, Mohamed. Louis, Anne. Baudry, David. Mazari, Belahcene. (2014). WSN's Modeling For a Smart Building Application. *ENERGYCON*. 36 (14), 821-827.
- [9] Bonganay, Altir Chirstian D. Magno, Josef C. Marcellana, Adrian G. Marvin, John E & Perez, Noel G. (2014). Automated Electric Meter Reading and Monitoring System using ZigBee-Integrated Raspberry Pi Single Board Computer via Modbus. *IEEE*. 1 (1), 1-6.
- [10] Botta, Allesio & Pescapé, Antonio. (2013). New generation satellite broadband Internet services: should ADSL and 3G worry? *The 5th IEEE International Traffic Monitoring and Analysis Workshop*. 13 (5), 3279-3284.
- [11] Botta, Allesio & Pescapé, Antonio. (2014). On the Performance of New Generation Satellite Broadband Internet Services. *IEEE Communications Magazine*. 14 (4), 202-209.
- [12] Chaklander, Sayem. Alam, Junaed. Islam, Monirul & Sabbir, Ali. (2013). Bridging Digital Divide: 'Village Wireless LAN', A Low Cost Network Infrastructure Solution for Digital Communication, Information Dissemination & Education in Rural Bangladesh. *2nd International Conference on Advances in Electrical Engineering*. 15 (14), 277-281.
- [13] Chaklander, Sayem. Alam, Junaed. Islam, Monirul & Sabbir, Ali. (2013). Bridging Digital Divide: 'Village Wireless LAN', A Low Cost Network Infrastructure Solution for Digital Communication, Information Dissemination & Education in Rural Bangladesh. *2nd International Conference on Advances in Electrical Engineering*. 15 (14), 277-281.
- [14] Chandra, Abel. Lee, Yeonwoo & Park, Hyeok. (2013). Review on Sensor Cloud and its Integration with Arduino based Sensor Network. *IEEE*. 14 (2), 20-23.



- [15] Chu, Hongyu. Xie, Zhijiang. Shao, Yanhua & Liu, Qin. (2010). Design and Implement of WSN Based on Bluetooth and Embedded System. *International Conference on Computer Application and System Modeling*. 5 (2), 641-643.
- [16] Clausen, Horst D. Hilmar, Linder. & Collini-Nocker, Bernade. (1999). Internet over Direct Broadcast Satel I ites. *IEEE Communications Magazine*. 16 (3), 148-151.
- [17] Costanzo, Alfio. (2013). An Arduino Based System Provided with GPS/GPRS Shield for Real Time Monitoring of Traffic Flows. *IEEE*. 1 (1), 1-5.
- [18] Dargie, Walteneagus & Poellabauer, Christian (2010). *Fundamentals of Wireless Sensor Networks*. United Kingdom: Wiley. 331.
- [19] Faludi, Robert (2011). *Building Wireless Sensor Networks*. California: O'Reilly. 321.
- [20] González, Raúl (2011). *Python para todos*. España: Creative Commons. 160.
- [21] Hendrickson, Richard. (1998). GLOBALSTAR FOR THE MILITARY. *IEEE*. 6 (5), 808-813.
- [22] Horak Ray (2007). *Telecommunications and data communications Handbook*. New Jersey: Wiley. 832.
- [23] Hrykiewicz, S & Koch, B.. (2002). SATELLITE COMMUNICATIONS NETWORKING CAPABILITIES REFERENCE MODEL. *IEEE*. 25 (2), 208-212. Dua
- [24] Hyberston, Duane W. *Model-oriented Systems Engineering Science: A Unifying Framework for Traditional and Complex Systems*. CRC Press. 320.
- [25] Iqbal, M. Shamoan, Shahana & Bhatti, Mohsin. (2013). Future of Satellite Broadband Internet Services and Comparison with Terrestrial Access Methods e.g. DSL and Cable modem. *IEEE*. 13 (9), 59-96.
- [26] Israel, David J. (2005). Space Network IP Services (SNIS): An Architecture for Supporting Low Earth Orbiting IP Satellite Missions. *IEEE*. 7 (5), 900-903.
- [27] Jamalipour, Abbas & Tung Tracy. (2001). The Role of Satellites in Global IT: Trends and Implications. *IEEE Personal Communications*. 16 (1), 5-11.
- [28] Jamalipour, Abbas. Katayama Masaaki. Yamazato, Takaya & Ogawa Akira. (1995). Performance of an Integrated VoiceData System in Nonuniform Traffic Low Earth-Orbit Satellite Communication Systems. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*. 13 (2), 465-473.
- [29] Jayanthiladevi, A. Suma, S & Lalitha, T. (2013). Challenges and Authentication in Wireless Sensor Networks. *IEEE*. 13 (4), 123-129
- [30] Karray, M & BenSaleh, M. (2014). A Review on Wireless Sensor Node Architectures. *IEEE*. 15 (7), 321-326.
- [31] Kochlan, Michal & Hodon, Michal. (2014). WSN for Traffic Monitoring using Raspberry Pi Board. *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*. 2 (1), 1023-1026.
- [32] Kodali, Ravi & Sarma, Narasimba. (2013). Test bed for Wireless sensor networks using XMesh networking protocol. *IEEE*. 28 (12), 1506-1511.



- [33] Larson, Wiley J & Wertz James R. (1999). *Space Mission Analysis and Design*. 3rd ed. California: Microcosm Press. 976.
- [34] Li, Jo-Ping. Chang, Shing & Chou Pai. (2014). Demonstration Abstract: Enabling WSN Nodes to Send Data to Smartphones by Blinking LEDs. *IEEE*. 26 (14), 343-344.
- [35] Ma, Yanpeng. Wang, Xiaofeng. Zhao, Baokang & Su, Junshu. (2013). A Location Management Algorithm for LEO Satellite Networks. *IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. 164 (10), 1362-1366.
- [36] Mancuso, Vincenzo. Bianco Giuseppe. BLeafari, Nicola & Bimbacher, Ulla. (2005). Switched Ethernet Networking over LEO Satellite. *IEEE*. 10 (5), 883-886.
- [37] Mansor, Hasmah. Shukor, Abdul. Meskam, Siti & Rusli, Nur. (2013). Body Temperature Measurement for Remote Health Monitoring System. *IEEE International Conference on Smart Instrumentation, Measurement and Applications*. 21 (6), 63-65.
- [38] Maral, Gérard. Bousquet, Michel (2002). *Satellite Communications Systems*. 5th Ed. United Kingdom: Wiley. 743.
- [39] Michalak, Slawomir. (2014). Raspberry Pi as a Measurement System Control Unit. *IEEE*. 1 (1), 1-4.
- [40] Milenkovic, Aleksandar M. Markovic, Ivica M. Jankovic, Dragan & Rajkovic, Petar. (2013). Using of Raspberry Pi for Data Acquisition from Biochemical Analyzers. *TELSIKS*. 17 (7), 389-392.
- [41] Monk, Simon (2014). *Raspberry Pi Cookbook*. California: O'Reilly. 410.
- [42] Nogueira, Mark. (2008). The Benefits of Low-Earth Orbiting Satellite Technology for the International Community: Can the Potential be Realized. *Indiana Journal of Global Legal Studies*. 5 (2), 739-760.
- [43] Poulakis, Marison. Vassaki, Stavrouka & Panagopoulos, Athanasios. (2013). Satellite-Based Wireless Sensor Networks: Radio Communication Link Design. *IEEE*. 40 (13), 2620-2624.
- [44] Pratt, Stephen R. Raines, Richard A. Fossa, Carl E & Temple, Michael A.. (1999). AN OPERATIONAL AND PERFORMANCE OVERVIEW OF THE IRIDIUM Low EARTH ORBIT SATELLITE SYSTEM. *IEEE Communications Surveys*. 1 (1), 1-9.
- [45] Raines, Richard A. & Davis Nathaniel J. (1995). PERSONAL COMMUNICATIONS VIA LOW EARTH ORBIT SATELLITE COMMUNICATION NETWORKS. *IEEE*. 38 (2), 1229-1233.
- [46] Rajkumar, Immanuel. Sankaranarayanan, PE & Sundari, G. (2013). GPS and Ethernet based Real Time Train Tracking System. *IEEE Communications Surveys*. 13 (8), 282-286.
- [47] Rappaport, Theodore S. (2002). *Wireless Communications. Principles and practica*. 2nd ed. United Kingdom: Prentice Hall. 689.



- [48] Raskovic, Dejan & Milenkovic, Aleksandar . (2009). Embedded Web Server for Wireless Sensor Networks. *41st Southeastern Symposium on System Theory*. 9 (2), 19-23.
- [49] Roddy Dennis (2006). *Satellite Communications*. 5th ed. New York: McGraw-Hill. 656.
- [50] Sandhoo, GP & Hernández M. José (2014). *Spacecraft communications systems desing*.
- [51] Sabastian, Tonny A. Guarddin Gladhi & Rheza, Abi. (2012). Aeronautical Telecommunication Network Protocol Tunnel Prototype Over IP Based Infrastructure. *ICACISIS*. 15 (7), 83-88.
- [52] Smith, Dan & Hendrickson Rich. (1995). MISSION CONTROL FOR THE 48-SATELLITE GLOBALSTAR CONSTELLATION. *IEEE*. 5 (25), 828-832.
- [53] Sun, Zhili (2005). *Satellite Networking. Principles and Protocols*. United Kingdom: Wiley. 377.
- [54] Takei, Jun & Murai, Jun. (2003). Satellite Communication in the Internet: Its history and the Technology. *IEEE*. 1 (1), 1-5.
- [55] Tanenbaum Andrew S & Wetherall David J. (2011). *Computer Networks*. 5th ed. Boston, Massachusetts: Prentice Hall. 901.
- [56] Tardif, Pierre. (1994). TCP/IP PERFORMANCE OVER SATELLITE. *Canadian Crown Copyright*. 11 (3), 435-439.
- [57] The International Telecommunication Union (2002). *Handbook On Satellite Communications*. Usa: Wiley. 1112.
- [58] Tian, Yu. Gao, Bo & Tong, Ling. (2008). DESIGN AND APPLICATION OF WIRELESS SENSOR NETWORKS FOR GROUND VERIFICATION OF REMOTE SENSING. *IEEE Communications Surveys*. 11 (5), 381-383.
- [59] Tropea, M & Fazio, P. (2013). Evaluation of TCP Versions over GEO Satellite Links. *IEEE*. 1 (1), 1-5.
- [60] Tso, Fung Po. White, David R. Jouet, Simon. Singer, Jeremy & Pezaros Dimitrios. (2013). The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures. *IEEE 33rd International Conference on Distributed Computing Systems Workshops*. 7 (5), 108-112.
- [61] Tsumoda, Hiroshi. Ohta, Kohei. Kato, Nei & Nemoto Yoshiaki. (2004). Supporting IP/LEO Satellite Networks by Handover-Independent IP Mobility Management. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*,. 22 (2), 300-308.
- [62] Verma, S. Pillai, P & Hu, F. (2013). Performance Analysis of Data Aggregation and Security in WSN-Satellite Integrated Networks. *IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications: Mobile and Wireless Networks*. 13 (4), 3297-3301.
- [63] Verma, Suraj & Pillai, Prashant. (2013). Performance Evaluation of Alternative Network Architectures for Sensor-Satellite Integrated Networks. *International Conference on Advanced Information Networking and Applications Workshops*. 13 (10), 120-125.



- [64] Vujovic, Vladimir & Maskimovic, Mirjana. (2014). Raspberry Pi as a Wireless Sensor Node: Performances and Constraints. *MIPRO*. 40 (25), 1013-1018.
- [65] Wan, Peng. Ye, Jianshe & Song, Shijie. (2014). Study on the Telecommunication Technology Based on the Distributed Satellite Constellation Networks. *IEEE*. 1 (1), 1-8.
- [66] Wang, Ershen. Zhang, Shufang & Zhang, Zhixian. (2013). Research and Implement of PPP and TCP/IP Protocol based on GPRS. *IEEE*. 11 (3), 52-56.
- [67] Wu, William S. (2010). Internet Satellite Challenges. *Second International Conference on Evolving Internet*. 3 (10), 31-35.
- [68] Zafar, Abu. Atiquazzaman, Mohammed & Ivancic William. (2011). Network Mobility in satellite networks: architecture and the protocol. *INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS*. 10 (5), 179-197.
- [69] Adafruit <https://learn.adafruit.com/> [Acceso: Octubre 2014]
- [70] Arduino <http://forum.arduino.cc/> [Acceso Octubre 2014]
- [71] Raspberry PI <http://www.raspberrypi.org/forums/> [Acceso Octubre 2014]
- [72] MIT Open Course Ware <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/> [Noviembre 2014]
- [73] Repositorio digital Universidad Jaume <http://repositori.uji.es/xmlui/> [Acceso Noviembre 2014]
- [74] Secure Computer Systems, Stanford University <http://www.scs.stanford.edu> [Acceso noviembre 2014]
- [75] Wikipedia, Sensor. <http://en.wikipedia.org/wiki/Sensor> [Julio 2014]
- [76] Wikepedia Surrey Satellite Technology. http://en.wikipedia.org/wiki/Surrey_Satellite_Technology [Enero 2015]
- [77] Goggle Nears Major Investment in SpaceX to Bolster Satellites <https://www.theinformation.com/Google-Nears-Major-Investment-in-SpaceX-to-Bolster-Satellites> [Febrero 2015]
- [78] Revealed: Elon Musk's Plan to Build a Spacce Internet. <http://www.bloomberg.com/news/articles/2015-01-17/elon-musk-and-spacex-plan-a-space-internet#p2> [Febrero 2015]
- [79] Electrónica <http://jeremyblythe.blogspot.mx/p/raspberry-pi-page.html> [Septiembre 2014]
- [80] RFC 1661 <https://www.ietf.org/rfc/rfc1661.txt> [Octubre 2014]
- [81] RFC 2687 <http://tools.ietf.org/html/rfc2687> [Octubre 2014]
- [82] RFC 791 <http://tools.ietf.org/html/rfc791>[Octubre 2014]
- [83] RFC 1122 <http://tools.ietf.org/html/rfc1122>[Octubre 2014]
- [84] RFC 1968 <http://tools.ietf.org/html/rfc1968>[Octubre 2014]
- [85] RFC 2460 <http://tools.ietf.org/html/rfc2460> [Octubre 2014]
- [86] Imágenes <http://xkcd.com/> [Septiembre 2014]
- [87] Ilustración en contraportada por Patricio Betteo

Epilogo

Siempre me ha parecido muy anticlimático el final tan abrutado que brinda leer las referencias como última instancia en cualquier texto profesional. Por ello aprovecho este espacio para abordar ciertas cuestiones importantes que no encontraron lugar en el desarrollo de la presente tesis.

Un agradecimiento especial a la empresa Globalstar por todas las facilidades prestadas durante las pruebas del modem satelital. A los compañeros del CAT y de la Facultad de Ingeniería. Ya tenemos a X y Y, ya viene Z y aunque no se menciona, sé que vendrá un agradecimiento futuro en alguna parte.

Y así hay que quedar expectantes al próximo encuentro en algún texto de mismo tipo que el presente, porque como dice el filósofo contemporáneo barcelonés, Santi Balmes, que no es necesario tener principios, lo importante es

LUCERO
ATOLERO

