



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA ELECTRÓNICA Y REDES DE COMUNICACIÓN

PROTOCOLO DE CAPA MAC PARA EL SISTEMA “MOBILE SMART NODE”, EN EL USO EFICIENTE DE ENERGÍA EN AMBIENTES DOMÉSTICOS.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN.

AUTORA: Valeria Susana Mafla Navarrete

DIRECTOR: Ing. Hernán Domínguez

Ibarra, Julio de 2011

DECLARACIÓN

Yo **VALERIA SUSANA MAFLA NAVARRETE** declaro bajo juramento que el trabajo aquí descrito es de mi autoría; y que este no ha sido previamente presentado para ningún grado o calificación profesional.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica Del Norte, según lo establecido por las leyes de propiedad intelectual, reglamentos y normatividad vigente de la Universidad Técnica Del Norte.

Valeria Susana Mafla Navarrete

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por la estudiante: **VALERIA SUSANA MAFLA NAVARRETE**, bajo mi supervisión.

Ing. Hernán Domínguez

DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Primeramente quiero agradecer a mi querida Universidad Técnica Norte, institución que me formó como parte de ella por cinco años de mi vida. A mi facultad, la Facultad de Ingeniería en Ciencias Aplicadas FICA y a la Carrera de Ingeniería en Electrónica y Redes de Comunicación. También quiero agradecer al Instituto de Postgrado y de manera muy especial a la Dra. Patricia Aguirre y al Dr. Mario Montenegro. A la Universidad Leuphana de Lüneburg, concretamente al Institut für Umweltkommunikation (INFU), especialmente al Señor Prof. Dr. Gerg Michelsen y al Señor Dr. Marco Rieckmann por su trabajo e interés de vincular relaciones con instituciones y estudiantes de Latinoamérica, por la oportunidad que me brindaron para realizar un semestre de intercambio estudiantil en la República Federal de Alemania, lugar donde nació la iniciativa de realizar el presente proyecto de grado. Quisiera también agradecer al Ing. Mauricio Domínguez por la dedicación y el apoyo amistoso en la realización de este proyecto. Agradezco a todos los profesores de la facultad que me han apoyado una y otra vez, en especial al Ing. Milton Gavilánez y al Ing. Jaime Michilena. También quiero agradecer a: David Narváez, María Fernanda Chiriboga, Oleg Sim y Dennis Bauch, quienes me han ayudado con las inquietudes técnicas sobre mi trabajo. Mil gracias a todos mis amigos y compañeros que a lo largo de mi carrera universitaria compartieron conmigo tantas vivencias, y de todo corazón y sin menos importancia a mis padres Susana y Oswaldo quienes siempre han estado a mi lado para brindarme su apoyo y amor incondicional.

DEDICATORIA

Quiero dedicar el presente proyecto con mucho cariño a mi familia, en especial a mis padres: Susana Navarrete y Luis Mafla, quienes me dieron la vida y han estado conmigo en cada momento para apoyarme, educarme y sobretodo amarme. También dedico el trabajo realizado al amor de mi vida y a todos mis amigos quienes siempre me han dado fuerza y valor para seguir adelante.

TABLA DE CONTENIDOS

PORTADA.....	i
DECLARACIÓN.....	ii
CERTIFICACIÓN.....	iii
AGRADECIMIENTOS.....	iv
DEDICATORIA.....	v
TABLA DE CONTENIDOS.....	vi
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS.....	xiii
RESUMEN.....	xiv
ABSTRACT.....	xv
1 INTRODUCCIÓN A LA INTELIGENCIA AMBIENTAL.....	1
1.1 GENERALIDADES.....	1
1.2 EL PROBLEMA DEL CAMBIO CLIMÁTICO	1
1.2.1 ¿QUÉ ES Y POR QUÉ SE PRODUCE EL CALENTAMIENTO GLOBAL?	
2	
1.2.2 GENERACIÓN Y USO DE ENERGÍA COMO PROBLEMA AMBIENTAL	3
1.3 DESARROLLO SUSTENTABLE	4
1.3.1 ELEMENTOS CLAVES DEL DESARROLLO SUSTENTABLE	5
1.3.1.1 Justicia.....	5
1.3.1.2 Orientación global	5
1.3.1.3 Barreras ecológicas	5
1.3.1.4 Participación ciudadana.....	5
1.3.2 DIMENSIONES DE LA SUSTENTABILIDAD	6
1.3.3 ESTRATEGIAS PARA LA IMPLEMENTACIÓN DEL DESARROLLO	
SUSTENTABLE	6
1.3.3.1 Estrategia de eficiencia	7

1.3.3.2	Estrategia de suficiencia	7
1.3.3.3	Estrategia de consistencia	7
1.3.3.4	Estrategia de formación	7
1.4	EFICIENCIA ENERGÉTICA COMO ESTRATEGIA CLAVE PARA UN DESARROLLO SUSTENTABLE	8
1.4.1	DEFINICIÓN DE LA EFICIENCIA ENERGÉTICA.....	9
1.4.2	PROGRAMA NACIONAL DE EFICIENCIA ENERGÉTICA EN EL ECUADOR.....	11
1.5	INTELIGENCIA AMBIENTAL, HERRAMIENTA TECNOLÓGICA PARA EL USO EFICIENTE DE ENERGÍA.....	12
1.5.1	EL CONCEPTO DE INTELIGENCIA AMBIENTAL.....	12
1.5.1.1	Computación ubicua	14
1.5.1.2	Comunicación ubicua.....	14
1.5.1.3	Interfaces de usuario inteligentes	14
1.5.2	PROPIEDADES DE LOS ENTORNOS CON INTELIGENCIA AMBIENTAL	15
1.5.3	VENTAJAS DEL USO DE LA INTELIGENCIA AMBIENTAL PARA LA EFICIENCIA ENERGÉTICA	16
2	FUNCIONAMIENTO DEL MOBILE SMART NODE (MSN).....	17
2.1	GENERALIDADES.....	18
2.2	REDES INALÁMBRICAS	18
2.2.1	REDES INALÁMBRICAS DE ÁREA PERSONAL (WPAN).....	19
2.2.1.1	ZigBee	20
2.3	EL ESTÁNDAR IEEE 802.15.4	21
2.3.1	TIPOS DE DISPOSITIVOS IEEE 802.15.4	22
2.3.2	TOPOLOGÍAS DE RED IEEE 802.15.4	22
2.3.3	ARQUITECTURA IEEE 802.15.4.....	23
2.3.3.1	La Capa Física.....	24
2.3.3.2	La capa MAC (Media Access Control o Control de Acceso al Medio)	

2.4	EL SISTEMA MOBILE SMART NODE (MSN).....	35
2.4.1	DISPOSITIVOS Y FUNCIONAMIENTO DEL SISTEMA MOBILE SMART NODE (MSN)	35
2.4.2	APLICACIONES DEL SISTEMA MSN EN EL USO EFICIENTE DE ENERGÍA.....	37
2.4.2.1	Control de iluminación	37
2.4.2.2	Control de calefacción y aire acondicionado.....	37
2.4.2.3	Administración de encendido y apagado de electrodomésticos y equipos eléctricos	38
2.4.2.4	Control de portones y puertas eléctricas.....	38
2.4.3	DESCRIPCIÓN TÉCNICA DEL HARDWARE DEL MOBILE SMART NODE (MSN).....	39
2.4.4	DESCRIPCIÓN DEL FUNCIONAMIENTO LÓGICO DEL MOBILE SMART NODE (MSN).....	41
2.4.4.1	Capa física IEEE 802.15.4.....	42
2.4.4.2	Capa MAC IEEE 802.15.4	42
2.4.4.3	Capa de conversión IP IEEE 802.15.4.....	44
2.4.4.4	Capa IP.....	45
2.4.4.5	Capa TCP	46
2.4.4.6	Demonio o daemon XML RCP.....	47
2.4.4.7	Capa de decisión semántica.....	48
2.4.5	PILA IP VERSÁTIL PARA SISTEMAS EMBEBIDOS (VISE, VERSATILE IP STACK FOR EMBEDDED SYSTEMS)	49
2.4.6	ANÁLISIS DEL PROGRAMA VISE	51
2.4.6.1	Programa Principal o Main.....	52
2.4.6.2	Reactor	53
2.4.6.3	Tareas del Reactor (Reactor Tasks).....	54
2.4.6.4	Ciclo infinito para recepción de solicitudes	56
3	PROTOCOLO DE CAPA MAC.....	57
3.1	GENERALIDADES.....	58

3.2 ACCESO MÚLTIPLE POR DETECCIÓN DE PORTADORA CON ANULACIÓN DE COLISIONES (CSMA-CA, CARRIER SENSE, MULTIPLE ACCESS, COLLISION AVOIDANCE)	58
3.2.1 PROBLEMAS DE CSMA-CA	60
3.3 ACCESO MÚLTIPLE CON PREVENCIÓN DE COLISIONES (MACA, MULTIPLE ACCES WITH COLLISION AVOIDANCE)	61
3.4 ACCESO MÚLTIPLE CON PREVENCIÓN DE COLISIONES PARA WIRELESS (MACAW, MULTIPLE ACCES WITH COLLISION AVOIDANCE FOR WIRELESS).....	63
3.5 FUNCIONES DE LA CAPA MAC DEL ESTÁNDAR IEEE 802.11	63
3.5.1 Función de Coordinación Puntual (PCF, Point Coordination Function)..	64
3.5.2 Función de Coordinación Distribuida (DCF, Distributed Coordination Function).....	64
3.5.3 El espacio inter-trama IFS (Interframe Space).....	64
3.5.4 Vector de Asignación de Red NAV (Network Allocation Vector)	66
3.5.5 Mecanismo de retracción o Backoff	66
3.6 CONTEXTUALIZACIÓN DEL PROBLEMA.....	67
3.7 ANÁLISIS DE SOLUCIÓN PARA EL DESARROLLO DEL PROTOCOLO DE CAPA MAC.....	68
3.7.1 SUBROUTINA 1 DEL DISPOSITIVO TRANSMISOR MSN.....	70
3.7.2 SUBROUTINA 2 DEL DISPOSITIVO TRANSMISOR MSN	72
3.7.3 SUBROUTINA DEL DISPOSITIVO RECEPTOR FESN	73
3.7.4 Cálculo del tiempo de una transmisión de datos para la implementación del contador NAV.....	75
3.7.4.1 Dimensionamiento de los parámetros.....	77
3.7.4.2 Cálculos	78
3.8 IMPLEMENTACIÓN DEL PROTOCOLO DE CAPA MAC.....	81
3.8.1 LOS ARCHIVOS mac_Protocol_RTS_CTS. c/h	84
3.8.1.1 Proceso para la programación de las subrutinas del protocolo de capa MAC	85

3.8.2 PROGRAMACIÓN DEL PROTOCOLO EN EL MICROCONTROLADOR LPC 2148 DEL SISTEMA MSN	86
4 PRUEBAS DE FUNCIONAMIENTO	90
4.1 GENERALIDADES.....	91
4.2 PRUEBAS DE FUNCIONAMIENTO DEL PROTOCOLO DE CAPA MAC EN EL SISTEMA MSN	91
4.2.1 CONDICIONES PARA ESTABLECER UNA COLISIÓN ENTRE DISPOSITIVOS INALÁMBRICOS	91
4.2.2 REALIZACIÓN DE LAS PRUEBAS DE FUNCIONAMIENTO.....	93
4.2.2.1 Programación de los dispositivos.....	93
4.2.2.2 Comunicación entre un ordenador y un dispositivo del sistema MSN 94	
4.2.3 RESULTADOS DE LAS PRUEBAS DE FUNCIONAMIENTO	97
5 CONCLUSIONES Y RECOMENDACIONES.....	102
5.1 CONCLUSIONES	103
5.2 RECOMENDACIONES	105
6 BIBLIOGRAFÍA	108
7 SIGLAS Y ABREVIACIONES.....	113

ANEXOS

ÍNDICE DE FIGURAS

CAPÍTULO 1

Figura 1.1: Consumo de combustibles para generación eléctrica en el Ecuador... 4	4
Figura 1.2: Dimensiones de la sustentabilidad..... 6	6

CAPÍTULO 2

Figura 2.1: Tipos de redes inalámbricas. 19	19
Figura 2.2: Topologías de estrella y peer to peer..... 23	23
Figura 2.3: Arquitectura de dispositivos LR-WPAN..... 24	24
Figura 2.4: Formato del PPDU 25	25
Figura 2.5: Unidad de Datos de Protocolos MAC (MPDU)..... 29	29
Figura 2.6: Formato de la trama Beacon..... 30	30
Figura 2.7: Estructura de las súper tramas..... 32	32
Figura 2.8: Formato de la Trama de Datos..... 32	32
Figura 2.9: Formato de la trama de Acuse de Recibo. 33	33
Figura 2.10: Formato de la Trama de Comandos MAC..... 34	34
Figura 2.11: Dispositivos del Sistema MSN..... 36	36
Figura 2.12: Funcionamiento del Sistema MSN. 38	38
Figura 2.13: Dispositivo Mobile Smart Node, vista frontal. 39	39
Figura 2.14: Dispositivo Mobile Smart Node, vista posterior. 40	40
Figura 2.15 : Plataforma de comunicación del sistema MSN. 41	41
Figura 2.16: Funcionamiento del protocolo XML- RCP. 48	48
Figura 2.17: Protocolos de Internet de la pila VISE..... 50	50
Figura 2.18: Estructura del proyecto VISE 52	52

CAPÍTULO 3

Figura 3.1: (a) Nodo A transmitiendo. (b) Nodo B transmitiendo..... 60	60
Figura 3.2: Protocolo MACA: (a) A enviando a B un RTS. (b) B respondiendo a A un CTS..... 62	62

Figura 3.3: Duración de los IFS de acuerdo a su prioridad	65
Figura 3.4: Uso del protocolo RTS/CTS, IFS y NAV en el estándar IEEE 802.11	66
Figura 3.5: Tramas intercambiadas entre Tx y Rx.....	70
Figura 3.6: Subrutina 1 del algoritmo para el transmisor MSN.....	71
Figura 3.7: Subrutina 2 del algoritmo para el transmisor MSN.....	72
Figura 3.8: Subrutina del algoritmo para el receptor FESN.....	74
Figura 3.9: Duración de una transmisión de datos	76
Figura 3.10: Interfaz del programa CrossWorks para ARM.....	82
Figura 3.11: Proyecto VISE	83
Figura 3.12: Ficheros del proyecto VISE	83
Figura 3.13: Ubicación de los archivos ejecutables mac_Protocol_RTS_CTS.....	85
Figura 3.14: Ciclo de desarrollo del archivo ejecutable mac_Protocol_RTS_CTS	86
Figura 3.15: Interfaz Macraigor Wiggler	87
Figura 3.16: Placa convertidor JTAG hembra de 20 pines a 10 pines	88
Figura 3.17: Montaje completo para la programación de las placas del sistema MSN.	88

CAPÍTULO 4

Figura 4.1: Subrutina de un MSN para transmitir constantemente.....	92
Figura 4.2: Proceso para la comunicación entre un computador y un dispositivo del sistema MSN	94
Figura 4.3: Montaje completo del sistema MSN.....	95
Figura 4.4: Descripción de la conexión del HyperTerminal	96
Figura 4.5: Selección del puerto COM para el HyperTerminal	96
Figura 4.6: Especificación de las propiedades de conexión del HyperTerminal...	97
Figura 4.7: Mensajes recibidos, MSN transmitiendo ininterrumpidamente hacia el FESN.....	98
Figura 4.8: Interrupción en la recepción de mensajes por colisión.....	99
Figura 4.9: Recepción de mensajes en el HyperTerminal después de implementar el protocolo de capa MAC en el sistema MSN	101

ÍNDICE DE TABLAS

CAPÍTULO 2

Tabla 2.1: Características del estándar IEEE 802.15.4.....	22
Tabla 2.2: Tipos de Comandos de Tramas MAC	34
Tabla 2.3: Parámetros de red.....	54
Tabla 2.4: Parámetros de los buffers TCP	54
Tabla 2.5: Parámetros para enrutamiento AODV	55

CAPÍTULO 3

Tabla 3.1: Valores de los diferentes parámetros para el cálculo del contador NAV	78
Tabla 3.2: Parámetros para el programa del protocolo de capa MAC.....	80

RESUMEN

El Instituto de Sistemas y Tecnologías Distribuidos y Autónomos, VauST de la Universidad Leuphana de Lüneburg de la República Federal de Alemania, ha creado el proyecto Mobile Smart Node (MSN ,Nodo Inteligente Móvil) como un sistema móvil inalámbrico, basado en el modelo TCP/IP, para la administración y uso eficiente de energía eléctrica en ambientes domésticos. Sin embargo, el sistema presenta un problema de comunicación originado por colisiones existentes entre los dispositivos inalámbricos que lo conforman. El presente proyecto plantea una propuesta de solución para eliminar dichas colisiones mediante un protocolo de capa MAC. Se realiza un análisis del protocolo y se plantea un diseño que posteriormente se programa en lenguaje C bajo la plataforma de CrossWorks, y finalmente se realizan pruebas de funcionamiento del protocolo planteado.

ABSTRACT

The Institute for Distributed and Autonomous Systems and Technologies, VauST, of the Leuphana University of Lüneburg in the Federal Republic of Germany, has created the project “Mobile Smart Node” (MSN) as a wireless mobile system, based on the TCP/IP model, for the administration and efficient use of electricity in domestic environments. However, the system has a communication problem caused by collisions between wireless devices that are part of the system. This project proposes a solution to eliminate such collisions by a MAC layer protocol. An analysis of the protocol is carried out and a design is proposed which subsequently is programmed in the language “C” using the CrossWorks platform, and finally a testing of the operation of the proposed protocol is performed.

1 INTRODUCCIÓN A LA INTELIGENCIA AMBIENTAL



1.1 GENERALIDADES

1.2 EL PROBLEMA DEL CAMBIO CLIMÁTICO

1.3 DESARROLLO SUSTENTABLE

1.4 EFICIENCIA ENERGÉTICA COMO ESTRATEGIA CLAVE PARA UN DESARROLLO SUSTENTABLE

1.5 LA INTELIGENCIA AMBIENTAL, HERRAMIENTA TECNOLÓGICA PARA EL USO EFICIENTE DE ENERGÍA

1.1 GENERALIDADES

En este primer capítulo se trata el problema del cambio climático; qué es y por qué se produce; se enfatiza en la generación y uso de energía como problema ambiental. Estos temas son el prelude para abordar el concepto de Desarrollo Sustentable que figura como una solución a los problemas planteados. Se habla sobre los elementos claves, las dimensiones y las estrategias para un Desarrollo Sustentable, para luego destacar a la eficiencia energética como una estrategia clave para el Desarrollo Sustentable. Se establece además el concepto de eficiencia energética y se mencionan los proyectos realizados en el Ecuador en cuanto a este tema. Para terminar el capítulo, se analiza el concepto de Inteligencia Ambiental como una herramienta tecnológica para el uso eficiente de energía, se analizan las propiedades de los entornos implementados con Inteligencia ambiental, y para finalizar se resaltan las ventajas de utilizar esta herramienta en el desarrollo de tecnologías para el uso eficiente de energía como es el caso del sistema Mobile Smart Node.

1.2 EL PROBLEMA DEL CAMBIO CLIMÁTICO

En la actualidad el cambio climático constituye uno de los problemas más graves de nuestra generación, no necesariamente por la importancia que se da, sino por las catástrofes que está causando y que en el futuro van a ser más graves. El cambio climático, tiene varias repercusiones para el ambiente y, obviamente, para los seres humanos en aspectos económicos, sociales y culturales. Actualmente ya se pueden observar los efectos producidos por la fuerza de la naturaleza a causa del cambio climático, por ejemplo en África existe una gran pérdida de biodiversidad y degradación de los suelos; los glaciares del polo sur y Groenlandia se están deshielando y a su vez ocasionando que el nivel del mar suba; y así entre otros ejemplos más. Las secuelas de estos efectos sin duda están ocasionando problemas no solo ambientales como se mencionó sino que van más allá y su trascendencia es de gran importancia para la vida de todos los seres vivos.

En el Ecuador, han comenzado a salir a la luz ciertos efectos del cambio climático. Los suelos desérticos tienen un avance en zonas como Manabí y el sur de Loja. El volcán Cotopaxi se está deshielando, en los últimos tres años el glaciar se redujo 20 metros¹. En 1997 la costa ecuatoriana tuvo que enfrentar uno de los peores inviernos conocido como el fenómeno del niño. Ciertas especies de ranas han desaparecido de los Andes ecuatorianos desde 1998, también existe gran preocupación por el calentamiento del agua de la Reserva Marina de Galápagos, lo que origina que disminuya la población de peces y por ende otros animales que se alimentan de estos. La situación del Ecuador es preocupante, sobre todo con alza de temperaturas ya registradas (+1°C en América Central y América del Sur en un siglo²). Los glaciares andinos se retiran y esta tendencia es similar en Ecuador, Bolivia, Perú y Colombia ya que la disponibilidad del agua destinada al consumo y a la hidroelectricidad está comprometida.

1.2.1 ¿QUÉ ES Y POR QUÉ SE PRODUCE EL CALENTAMIENTO GLOBAL?

El calentamiento global es el incremento a largo plazo en la temperatura promedio del planeta Tierra y se debe a la emisión de gases de efecto invernadero. La humedad, el dióxido de carbono, los óxidos de nitrógeno y el metano presentes en la atmósfera, retienen parte del calor que resulta de la radiación infrarroja que penetra a la Tierra por el efecto de la luz solar. Esto permite que la oscilación diaria de la temperatura no sea tan brusca. La combinación de estos gases atmosféricos actúa como un techo de vidrio semejante al de los invernaderos, impidiendo que escape el calor de la corteza terrestre, por este motivo se conoce a éste efecto como efecto invernadero. Gracias al efecto invernadero natural, se mantiene una temperatura promedio en la Tierra que permite la existencia de seres vivos. El problema actual radica en que se está depositando grandes cantidades de gases de efecto invernadero artificiales en la atmósfera. De acuerdo con el Panel Intergubernamental sobre Cambio Climático (IPCC,

¹ El universo (2007), Se acelera deshielo del Cotopaxi.
<http://www.eluniverso.com/2007/01/19/0001/12/BA1AB99262D14D9AA0CA1B4AAE37C1FA.html> (Octubre de 2009)

² IPCC. (2007). Intergovernmental Panel on Climate Change. En <http://www.ipcc.ch/> (Octubre de 2009)

Intergovernmental Panel on Climate Change), los tres gases que contribuyen al calentamiento global son el dióxido de carbono, el metano y el óxido nitroso. A pesar de que estos gases se encuentran en forma natural en la atmósfera, la rápida liberación de cantidades artificiales proviene de las actividades humanas. Son consecuencias de las centrales eléctricas y de la industria energética, de las fábricas, oficinas y hogares, del uso de materiales fósiles, deforestación y la agricultura. La actividad humana influye de forma inequívoca en el cambio climático y, a menos que se modifique, causará un aumento de temperaturas de hasta 6.4 °C antes de que finalice el siglo XXI como lo indica el IPCC.

1.2.2 GENERACIÓN Y USO DE ENERGÍA COMO PROBLEMA AMBIENTAL

El planeta está siendo contaminado por las constantes emisiones de gases de invernadero, particularmente el CO₂, debidas principalmente a la utilización de combustibles de origen fósil para generar energía. Se debe considerar además que el transporte y uso de la energía es una de las actividades del hombre con mayor repercusión negativa en el ambiente.

Los gases de efecto invernadero, son generados en su mayoría por la combustión de los derivados de petróleo y por la quema de carbón mineral en cantidades industriales. En el mundo entero se consume millones de barriles diarios de petróleo. En el Ecuador se usan derivados de petróleo para generar el 45% ³ de la energía eléctrica total, en la figura 1.1 se puede observar el consumo de combustibles en el Ecuador destinados para generación eléctrica en los últimos años.

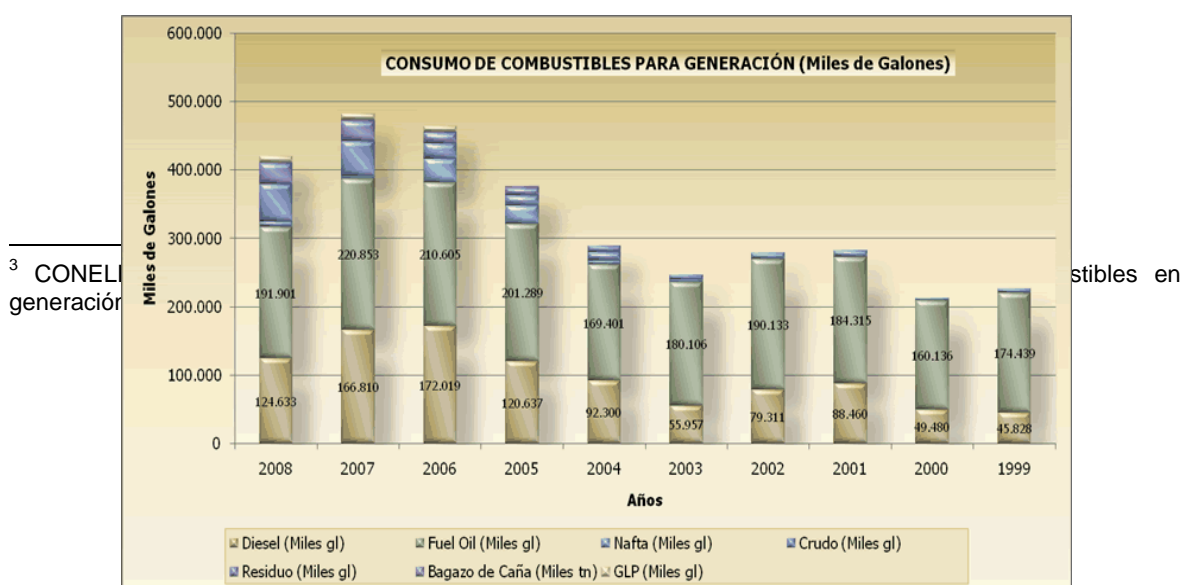


Figura 1.1: Consumo de combustibles para generación eléctrica en el Ecuador.

Fuente: CONELEC, <http://www.conelec.gov.ec/>

El carbón mineral se consume en todos los continentes en grandes cantidades, sobre todo para la fabricación de acero. Sin embargo, el mayor consumidor de carbón es la industria eléctrica. Australia genera el 84% de electricidad en centrales térmicas movidas por carbón. En China, Sudáfrica y Dinamarca las tres cuartas partes de electricidad provienen de carbón, Estados Unidos depende del carbón para generar la mitad de la producción de electricidad. El carbón mineral genera al año 2.000 millones de toneladas de gases letales que contaminan y deterioran el planeta⁴.

1.3 DESARROLLO SUSTENTABLE

El problema del cambio climático es un tema serio, una verdad que preocupa. Frente a esta realidad nacen ciertas preguntas: ¿qué medidas se pueden realizar para mitigar los efectos del cambio climático?, ¿qué puede hacer cada persona y todos los demás?, la respuesta posiblemente se encuentra en llevar a la práctica el concepto del desarrollo sustentable en todos los campos.

Según el informe Brundtland, la definición internacional general del desarrollo sustentable es: “El desarrollo sustentable es un desarrollo que satisface las

⁴ Naturaleza y Cultura Internacional. (2009). Calentamiento Global, Salvemos nuestro único planeta . Loja.

necesidades de las generaciones actuales sin comprometer la satisfacción de las necesidades de las generaciones futuras.”⁵

1.3.1 **ELEMENTOS CLAVES DEL DESARROLLO SUSTENTABLE** ^[6]

“Para lograr un desarrollo sustentable, cada persona debe concebir principios fundamentales, los cuales son: la justicia, la orientación global, barreras ecológicas y participación ciudadana”.

1.3.1.1 Justicia

Un desarrollo sustentable requiere de equidad tanto intrageneracional, es decir justicia distributiva entre el norte y el sur; entre ricos y pobres. Y justicia intergeneracional que significa el equilibrio entre las generaciones actuales y las generaciones futuras.

1.3.1.2 Orientación global

El análisis de los problemas de insustentabilidad y su solución requieren de una orientación global.

1.3.1.3 Barreras ecológicas

Todas nuestras actividades y bienes se basan en la naturaleza (“retinidad”). La capacidad de carga ecológica demarca los límites de nuestras actividades económicas y nuestro desarrollo social.

1.3.1.4 Participación ciudadana

Un desarrollo sustentable es un proceso social de aprendizaje, comunicación y transformación. Éste requiere de la participación de muchos actores para poder incluir sus ideas y visiones.

⁵ Informe Brundtland. (1987). UN Document. En: <http://www.un-documents.net/wced-ocf.htm> (octubre de 2008)

⁶ Michelsen, G., & Rieckmann, M. (2008). Programa de Maestría Internacional "Sustainable Development and Management" Introducción al Desarrollo Sustentable (Vol. II). Lüneburg

1.3.2 DIMENSIONES DE LA SUSTENTABILIDAD ^[6]

Al hablar de sustentabilidad existe una perspectiva común de que esta se logra a través de la integración de distintas dimensiones del desarrollo social. Existen diferentes puntos de vista sobre cuáles y qué tan importantes son estas dimensiones. El informe Brundtland menciona tres: ecología, aspectos sociales y economía. Además se discuten las dimensiones cultural, institucional y política. Según los autores Michelsen/Rieckmann quienes a su vez hacen referencia a Stoltenberg, la dimensión cultural es de gran importancia; y se elabora un modelo completo que se resume en la figura 1.2.

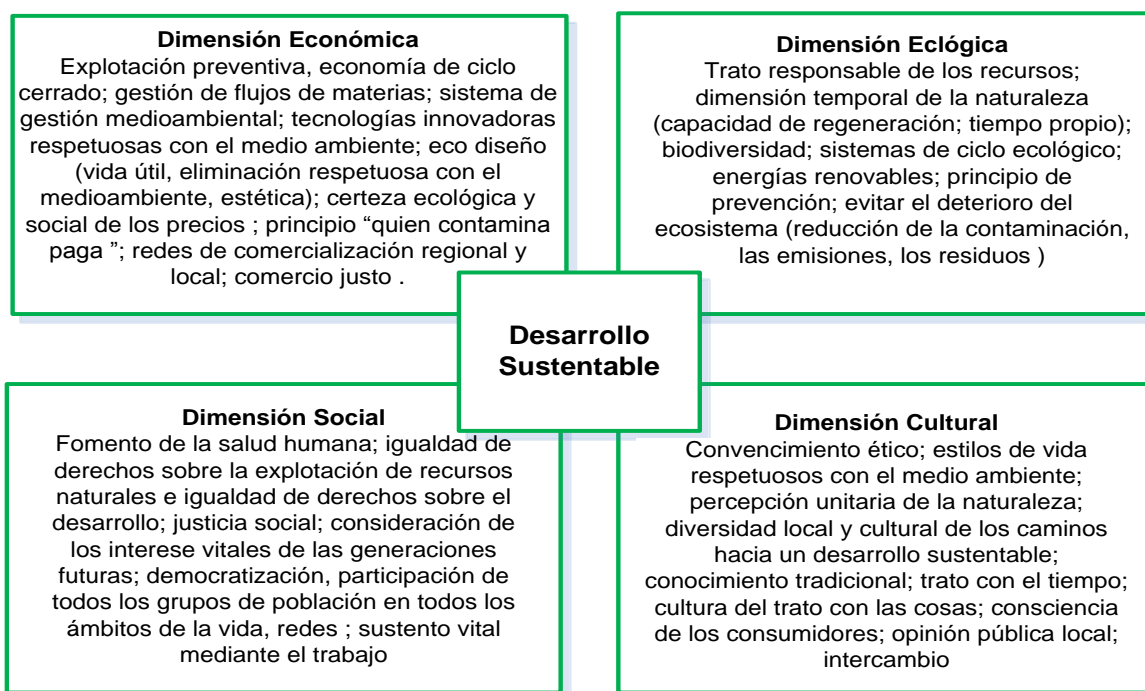


Figura 1.2: Dimensiones de la sustentabilidad.

Fuente: Michelsen, G., & Rieckmann, M. (2008). Programa de Maestría Internacional "Sustainable Development and Management" Introducción al Desarrollo Sustentable (Vol. II). Lüneburg, pág70.

1.3.3 ESTRATEGIAS PARA LA IMPLEMENTACIÓN DEL DESARROLLO SUSTENTABLE ^[6]

Existen muchas estrategias que se pueden aplicar para alcanzar un desarrollo sustentable. Sin embargo, se destacan cuatro estrategias que se complementan mutuamente, estas son: La estrategia de eficiencia, la estrategia de suficiencia, la estrategia de consistencia y la estrategia de formación.

1.3.3.1 Estrategia de eficiencia

Esta estrategia se aplica al consumo de energía y materias primas. El objetivo es reducir los consumos de energía y de materias primas a través de innovaciones de tecnología medioambiental, manteniendo o incluso elevando la calidad de los productos o los servicios. Esto conlleva a la realización de procesos eficientes, productos eficientes, aumentar la vida útil y la capacidad de reparación de los productos y el reciclaje.

1.3.3.2 Estrategia de suficiencia

Está encaminada a limitar las prácticas contaminantes para el medio ambiente y las que necesitan muchos recursos naturales, o sustituirlas con prácticas menos contaminantes, se trata de “vivir mejor en lugar de tener más”.

1.3.3.3 Estrategia de consistencia

Es una consistencia que adapta los flujos de materias y de energía de forma cualitativa y cuantitativa a la capacidad de regeneración de los ecosistemas. En lugar de reducir los daños medioambientales mediante un aprovechamiento más eficiente o suficiente, estos daños deberían ser evitados mediante formas de vida y de economía ecológicamente adecuadas.

1.3.3.4 Estrategia de formación

Es muy importante crear una conciencia de sustentabilidad en la sociedad y para esto se requiere de formación. La educación permite una sensibilización y cualificación para la participación responsable en la formación del desarrollo futuro.

1.4 EFICIENCIA ENERGÉTICA COMO ESTRATEGIA CLAVE PARA UN DESARROLLO SUSTENTABLE

Una de las alternativas para mitigar los efectos ambientales negativos por la generación de energía eléctrica, es el uso de energías renovables o energías limpias, debido a su diversidad, abundancia, capacidad de regeneración por medios naturales y principalmente porque no generan emisiones de gases de invernadero. Su uso estaría dentro de la estrategia de consistencia para la implementación del desarrollo sustentable. Sin embargo, esta predisposición inagotable de las energías renovables, es solamente un concepto virtual, ya que a pesar de que el Sol abastecerá estas fuentes de energía (radiación solar, viento, lluvia, etc.) durante los próximos cuatro mil millones de años⁷, no se puede disponer fácilmente de estas fuentes, puesto que la mayoría no siempre están presentes en la naturaleza en igual cantidad. Además a pesar de que la energía provenga de una fuente renovable, si esta se pierde o no se optimiza su uso, todo el proceso para su generación y distribución es en vano, ocasionando pérdidas energéticas, económicas y ambientales. Por este motivo es muy importante controlar y gestionar el uso de energía. Es aquí donde interviene otra estrategia del desarrollo sustentable que es la eficiencia. Para este objetivo se puede valer de muchos medios, y actualmente en muchos países existen políticas ambientales que impulsan el ahorro, la eficiencia energética, la utilización de equipos eficientes y reingenierías de los sistemas de distribución. Sin embargo, para la aplicación de estas políticas se necesita el desarrollo de tecnologías como herramientas esenciales para el uso eficiente y la administración de energía.

La situación ambiental actual requiere de acciones inmediatas que ayuden a preservar el ambiente para las generaciones futuras. La eficiencia energética es un factor clave para alcanzar un desarrollo sustentable dentro del campo energético. “La energía más económica es la que no se consume y la más costosa es la que no se tiene. Del uso racional de energía se garantiza el

⁷ Gutiérrez, Á. (2007). AGROENERGÉTICA: Ventajas e Inconvenientes de las Energías Renovables. En <http://agroenergetica.iespana.es/ventajas.html> (Octubre de 2009).

suficiente abastecimiento. De la eficiencia se obtienen beneficios para el desarrollo sustentable de los pueblos, con el fin de satisfacer necesidades indispensables de vivienda, alimentación, salud, educación, disminuir la pobreza y conservar el ambiente”⁸.

Actualmente en el Ecuador la nueva Constitución del 2008 reconoce el derecho de la población a vivir en un ambiente sano y ecológicamente equilibrado, que garantice la sostenibilidad y el Buen Vivir, Sumak Kawsay, además el Estado se compromete a promover la eficiencia energética, el desarrollo, uso de prácticas y tecnologías ambientalmente limpias y sanas. Esto demuestra conjuntamente con los nuevos proyectos sobre eficiencia energética que el Gobierno está impulsando a través del Ministerio de Electricidad y Energías Renovables el interés y la necesidad de implementar proyectos en esta área en el Ecuador. Sin embargo, la falta de oferta y el desconocimiento de nuevas tecnologías constituyen una gran barrera para implementar este tipo de sistemas en el país. Esto expresa la necesidad de introducir en el Ecuador tecnologías para el uso eficiente de energía a través de la investigación y la cooperación con otros países que ya han experimentado en este campo como es Alemania ya que según la Agencia Internacional de Energía, Alemania y Japón lideran en términos de eficiencia energética a nivel mundial.

1.4.1 DEFINICIÓN DE LA EFICIENCIA ENERGÉTICA

De acuerdo con el Ministerio de Electricidad y Energía Renovable del Ecuador se puede definir a la eficiencia energética de la siguiente forma:

“La eficiencia energética es la reducción del consumo energético sin disminuir la comodidad y calidad de vida de sus usuarios. Es el ahorro de energía, mediante la adopción de buenos hábitos para su uso, que representa una gran inversión económica para la sociedad, además de ser la forma más fácil de reducir la

⁸ OLADE. (2008). Organización Latinoamericana de Energía. <http://www.olade.org.ec/eficiencia.html> (octubre de 2009)

emisión de gases de invernadero a la atmósfera y asegurar las necesidades de este recurso para las futuras generaciones”⁹.

Refiriéndose a la definición de eficiencia energética Mentor Poveda dice:

“Es preciso tener presente que la eficiencia energética en su concepción más amplia pretende mantener el servicio que presta, reduciendo al mismo tiempo el consumo de energía. Es decir, se trata de reducir las pérdidas que se producen en toda transformación o proceso, incorporando mejores hábitos de uso y mejores tecnologías. Incluso es ir más allá de solo mantener los servicios que se obtienen de la energía y se demuestra, con múltiples ejemplos, que es posible reducir a la mitad el consumo duplicando los beneficios”¹⁰.

Establecida la definición de eficiencia energética, es importante su aplicación, por ejemplo en la industria sin duda es donde más se consume energía, su ahorro no implica sacrificios, exige un cambio de hábitos e incorporación de tecnologías eficientes, para mejorar los servicios usando menos energía. El éxito de eficiencia energética depende de la sociedad en general, no solo las grandes industrias son las que deben implementar proyectos en este campo, cada persona es parte de la solución, comenzando con acciones sencillas en el hogar como por ejemplo apagar luces y electrodomésticos en caso de no estar utilizándolos, e implementar nuevas tecnologías eficientes.

Los proyectos sobre eficiencia energética requieren de largo plazo para ver sus frutos. En América Latina y el Caribe, existe un gran potencial de eficiencia energética que representaría un gran ahorro en recursos naturales y económicos, ya que se puede disminuir significativamente el consumo de combustibles destinados para generación de energía. Se estima que se puede ahorrar 156 mil millones de dólares en combustibles entre el 2003 y 2018, según la base del estudio de prospectiva energética de la Organización Latinoamericana de Energía OLADE. Éste estudio toma el ejemplo de los esfuerzos realizados por la Unión

⁹ Ministerio de Electricidad y Energía Renovable (2009).
http://www.mer.gov.ec/Meer/portal_meer/internaView.htm?code=764&template=meer.internas

¹⁰ Organización Latinoamericana de Energía (2009).
<http://www.olade.org.ec/documentos2/EFICIENCIA%20ENERG%C3%89TICA%20RECURSO%20NO%20APROVECHADO-Agosto-2007.pdf>

Europea en cuanto a eficiencia energética, siendo los que mejores resultados han tenido hasta la presente fecha, éstos se han canalizado a través de instituciones nacionales, constituidas con la responsabilidad específica de desarrollar la eficiencia energética en su país. Alrededor del 85% de países europeos cuentan con instituciones especializadas en el tema. Las instituciones promocionan la eficiencia actuando cerca de las empresas energéticas, coordinan las acciones gubernamentales en conservación de energía para mantener una sola línea de acción, lo cual es particularmente importante en el caso de las industrias donde se utilizan diversas formas de energía y además trabajan con la sociedad pues la integración de otros actores fuera del sector energético es clave en la implantación de programas orientados hacia el lado de la demanda.

1.4.2 PROGRAMA NACIONAL DE EFICIENCIA ENERGÉTICA EN EL ECUADOR

De acuerdo con la Subsecretaría de Energía Renovable y Eficiencia Energética del Ministerio de Electricidad y Energías Renovables del Ecuador, las principales ventajas del uso eficiente de energía para el país son: aplazar los requerimientos de financiamiento para la infraestructura energética; reducir la importación de electricidad y combustibles; y promover nuevas tecnologías y la modernización de los sectores de consumo. Las acciones que se plasman en este programa concretamente son las siguientes:

- Normalización y etiquetado.
- Incentivos tributarios para equipos eficientes.
- Segunda etapa del proyecto focos ahorradores.
- Proyecto inversión conjunta.
- Componente educativo.
- Eficiencia energética en edificios públicos.
- Proyectos de cooperación.
- Plan Nacional de Eficiencia Energética.

Durante Julio y octubre de 2008 en todo el Ecuador se realizó una sustitución masiva de 6 millones de focos incandescentes por ahorradores, siendo el proyecto más grande sobre eficiencia energética que se ha ejecutado en el Ecuador. Este proyecto le representa al país un ahorro aproximado de energía de 500.000 MWh al año, 250 MW de demanda en horas pico, y una reducción de 400.000 ton. de CO₂ al año.

Otro proyecto ejecutado en el país es el ahorro de energía en edificios públicos; el objetivo primordial de este proyecto, es diagnosticar los índices de consumo energético de los inmuebles e identificar las oportunidades de ahorro para una posterior implementación de sistemas de bajo consumo de energía. Además los trabajadores del sector público fueron capacitados sobre el uso eficiente de energía. El proyecto comenzó con 50 instituciones de carácter público y las instituciones que logren mejorar sus indicadores de eficiencia energética serán galardonadas con un Certificado de Excelencia y recibirán un manual de eficiencia energética con las pautas y procedimientos a seguir para un aprovechamiento sostenido y responsable de la energía nacional y de los recursos naturales que sirven para su generación.

1.5 INTELIGENCIA AMBIENTAL, HERRAMIENTA TECNOLÓGICA PARA EL USO EFICIENTE DE ENERGÍA

Existen varias formas para ahorrar energía, lo más sencillo posiblemente es mediante cambios en los comportamientos y en la consciencia de las personas, pero lograr estos cambios resulta en algunos casos difícil, además en ciertas situaciones se derrocha energía a pesar de las buenas acciones y voluntad de los individuos por fallas o deficiencias técnicas. Con ayuda de la tecnología, concretamente con la herramienta de la inteligencia ambiental, que se define en el siguiente tema, se puede eliminar ciertas dificultades en el ahorro de energía.

1.5.1 EL CONCEPTO DE INTELIGENCIA AMBIENTAL

El concepto de la Inteligencia Ambiental, se aplica a un entorno en el cual las personas interactúan de forma habitual con sistemas informáticos inteligentes que perciben a los individuos o a las situaciones del ambiente y actúan por sí mismos en consecuencia a dichas situaciones. Alejandro Sacristán explica el concepto de Inteligencia Ambiental como:

*“La Inteligencia Ambiental o Ambient Intelligence describe un entorno en el que las personas estarán envueltas y asistidas por inteligentes e intuitivos interfaces embebidos (incrustados internamente) en objetos cotidianos en comunicación entre sí, que conformarán un medioambiente electrónico que reconocerá y responderá a la presencia de los individuos inmersos en él de una forma invisible y anticipatorio”.*¹¹

Otras especialistas en el tema como: Noelia Carretero y Ana Belén Bermejo detallan más la concepción de Inteligencia Ambiental, ampliando las características propias de las tecnologías (adaptación y autonomía) basadas en el siguiente concepto:

*La Inteligencia Ambiental (Ambient Intelligence), también conocida como “entornos inteligentes”, consiste en la creación de espacios donde los usuarios interaccionen de forma natural y sin esfuerzo con los diferentes sistemas, gracias a que las tecnologías de computación y comunicación se convierten, en estos entornos, en invisibles para el usuario, al estar siempre presentes e integradas en los objetos cotidianos del mismo. De esta forma, es la propia tecnología la que se adapta a los individuos y a su contexto, actuando de forma autónoma, y facilitándoles la realización de sus tareas diarias y la comunicación entre ellos y con el entorno.*¹²

Los fundamentos de la Inteligencia Ambiental surgen a partir del concepto planteado en 1991 por Weiser, denominado “Computación Ubicua”¹³, que

¹¹Sacristán, A. (2009) La Inteligencia Ambiental aumentará nuestras capacidades cognitivas. http://www.tendencias21.net/La-Inteligencia-Ambiental-aumentara-nuestras-capacidades-cognitivas_a963.html

¹² Carretero, Noelia; Bermejo, Ana Belén (2009), de www.ceditec.etsit.upm.es/.../5-Resumen-Inteligencia-Ambiental.html

¹³ Weiser, M. (1991). *Weiser Mark*, de <http://www.ubiq.com/weiser/>(noviembre de 2009)

aprovecha los resultados obtenidos en el desarrollo de muchas otras áreas tecnológicas, entre las cuales caben destacar: las comunicaciones y dispositivos ubicuos, los interfaces multimodales de usuario, los sistemas de agentes artificiales, la domótica y la inteligencia artificial. Sin embargo, las tecnologías que realmente están siendo claves para el desarrollo de la Inteligencia Ambiental son principalmente tres: la computación ubicua, la comunicación ubicua y los interfaces de usuario inteligentes.

1.5.1.1 Computación ubicua

Es el acceso a gran cantidad de información y procesamiento de la misma, independientemente de la ubicación de los usuarios. Para este fin existe una gran cantidad de elementos de computación disponibles en un determinado entorno físico. Los elementos están empotrados o embebidos en utensilios, mobiliario y electrodomésticos comunes y comunicados por una red inalámbrica. El concepto de ubicuo se refiere en general a la presencia de una entidad en todas partes; pero en el área de computación adquiere la característica de ser, además, invisible. Este paradigma pretende brindar sistemas de cómputo inteligentes que se adapten al usuario, y cuyas interfaces permitan que éste realice un uso intuitivo de los sistemas.

1.5.1.2 Comunicación ubicua

Consiste en la comunicación entre los objetos y los usuarios, tiene como principal objetivo conseguir información en el momento y el lugar en que el usuario lo necesite.

1.5.1.3 Interfaces de usuario inteligentes

La visión principal de la Inteligencia Ambiental presenta al usuario rodeado de interfaces inteligentes e intuitivas, integradas en los objetos cotidianos de su entorno de forma transparente. Estas interfaces poseen capacidad para reconocer

la presencia de diferentes usuarios, y modificar su comportamiento en función de la identidad de dicho usuario, sus necesidades y las características del contexto o entorno donde se encuentren. Especial atención merecen aspectos como la facilidad de uso, el soporte eficiente de los servicios y la posibilidad de mantener interacciones naturales con las personas.

1.5.2 PROPIEDADES DE LOS ENTORNOS CON INTELIGENCIA AMBIENTAL

[12]

Los sistemas basados en este concepto pueden crearse bajo diversos escenarios conocidos como “entornos inteligentes ubicuos” o “entornos inteligentes”, en los que es posible lo siguiente:

- *Capturar las experiencias diarias:* mediante la monitorización y almacenamiento de toda la información asociada al entorno, referente al contexto, los usuarios y sus actividades.
- *Acceder a la información:* se refiere tanto a información del propio sistema, como a nueva información obtenida del exterior a través de Internet. El acceso debe ser posible desde cualquier lugar (ubicuo) y de una manera eficiente.
- *Soportar la comunicación y colaboración:* en el sistema se debe ofrecer capacidades de comunicación con el menor esfuerzo posible y en cualquier lugar. La implementación debe ser extensible a todos los escenarios con los que conviva el usuario, interconectando los sistemas de forma eficiente.
- *Desarrollar entornos sensibles al contexto:* Los entornos deben ser sensibles tanto a la información del entorno, como a la información del usuario. Para ello el sistema debe disponer de herramientas que capturen esa información, la procesen y, en función de los resultados de ese análisis, modifiquen su comportamiento.
- *Proporcionar nuevas formas de interacción hombre-máquina:* Mediante una amplia variedad de interfaces de usuario, el usuario interactúa con su entorno. Estas interfaces deben ser naturales, ubicuas y transparentes.

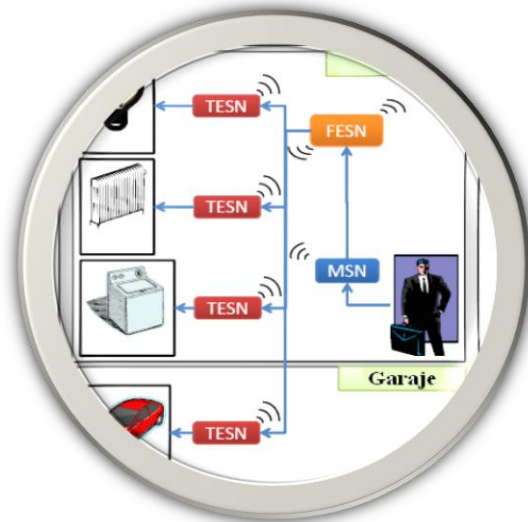
- *Ejercer de guía automática:* es decir, el sistema es capaz de detectar a los visitantes y en función de su perfil, facilitarles información y guiarles por ese entorno.
- *Facilitar el aprendizaje y entrenamiento* de diversas actividades.

1.5.3 VENTAJAS DEL USO DE LA INTELIGENCIA AMBIENTAL PARA LA EFICIENCIA ENERGÉTICA

La posibilidad que los propios entornos tales como hogares y edificios sean capaces de controlar los recursos que utilizan los seres humanos, para emplear tan sólo los que realmente son necesarios, es algo práctico para hacer a dichos entornos más sustentables. Gracias a la ayuda de la tecnología se puede ahorrar energía en un entorno doméstico. Dentro del hogar juegan un papel fundamental las redes y tecnologías inalámbricas, que se adaptan perfectamente a éste ambiente, al no haber comunicaciones a grandes distancias. Además resultan muy útiles al hacer que la comunicación entre agentes sea invisible para el usuario, pues no necesitan medios físicos que queden visibles al usuario.

Sin duda alguna, esta visión ha despertado un creciente interés por utilizar las tecnologías de la computación y electrónica en la construcción de sistemas que soporten las actividades de la vida diaria de forma más eficiente. Por ejemplo, actividades relacionadas con el control del hogar, la educación, el trabajo o la salud, etc...Es así que el instituto de Sistemas y Tecnologías Distribuidos y Autónomos, VauST (Verteilte autonome Systeme und Technologien,) de la Facultad de Ambiente y Tecnología de la Universidad Leuphana de Lüneburg de la República Federal de Alemania ha desarrollado el sistema Mobile Smart Node (MSN), con el objetivo de usar eficientemente la energía en ambientes domésticos, basándose en tecnología con Inteligencia Ambiental, ya que estos sistemas pueden mejorar la calidad de vida de los usuarios y dependiendo de su aplicación, contribuir haciendo más sustentables a los ambientes en donde sean utilizados, gracias a que ofrecen una mayor variedad de servicios de gran valor, además de proveer a los usuarios con nuevas formas de comunicaciones fáciles y eficientes para interactuar.

2 FUNCIONAMIENTO DEL MOBILE SMART NODE (MSN)



2.1 GENERALIDADES

2.2 REDES INALÁMBRICAS

2.3 EL ESTÁNDAR 802.15.4

2.4 EL SISTEMA MOBILE SMART NODE (MSN)

2.1 GENERALIDADES

El segundo capítulo del presente proyecto, comienza con una breve descripción de las redes inalámbricas, su clasificación y se centra en la tecnología inalámbrica de ZigBee, para luego abarcar el estándar IEEE 802.15.4 que define el nivel físico y el control de acceso al medio de las redes inalámbricas de área personal con tasas bajas de transmisión de datos (LR-WPAN, Low-Rate Wireless Personal Area network).

El siguiente subtema del capítulo describe el funcionamiento del sistema Mobile Smart Node, se detalla cómo trabaja, los dispositivos que lo conforman y se especifica tanto el Hardware como del Software que componen a este sistema, finalizando con un análisis de cómo está programado su Software.

2.2 REDES INALÁMBRICAS

En los últimos años la tecnología inalámbrica ha crecido significativamente, convirtiendo al mundo cada vez más móvil. Una red inalámbrica proporciona grandes beneficios a sus usuarios como: movilidad, rápida instalación, bajos costos de mantenimiento, escalabilidad, accesibilidad, entre otros.

Al igual que toda red, una red inalámbrica transmite datos a través de un medio. El medio en este tipo de redes es el espacio aire, un medio no guiado por donde se transmiten ondas electromagnéticas. Estas redes pueden ser clasificadas de acuerdo al área de cobertura en: redes inalámbricas de área personal (WPAN, Wireless Personal Area Network), redes inalámbricas de área local (WLAN, Wireless Local Area Network), redes inalámbricas de área metropolitana (WMAN, Wireless Metropolitan Area Network) y redes inalámbricas de área extendida (WWAN, Wireless Wide Area Network), como se observa en la figura 2.1.

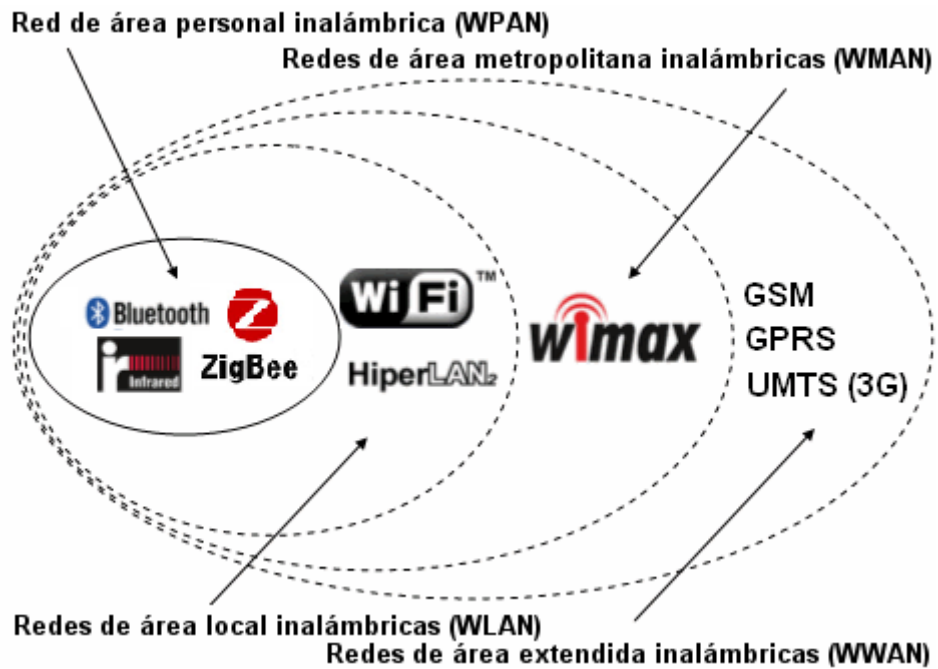


Figura 2.1: Tipos de redes inalámbricas.

Fuente: Kioskea, <http://es.kioskea.net/contents/wireless/wlintro.php3#> Kioskea.
Rediseñado por: Susana Mafla

2.2.1 REDES INALÁMBRICAS DE ÁREA PERSONAL (WPAN)

Las redes inalámbricas de área personal (WPAN, Wireless Personal Area Network) nacieron por la necesidad de conectar dispositivos periféricos personales (impresoras, teléfonos celulares, PDAs y electrodomésticos) sin conexión por cables para intercambiar información de forma eficiente, rápida y confiable. La principal característica de este tipo de redes es que su área de cobertura es de 10 metros a la redonda y que envuelve a una persona o a un dispositivo ya sea que esté o no en movimiento. A diferencia de las redes WLAN, una red WPAN implica pocas conexiones directas hacia el mundo exterior. Además para estas redes inalámbricas se han desarrollado protocolos simples y óptimos que hacen un uso eficiente de recursos como por ejemplo bajo consumo de energía ya que se trabaja con bajas tasas de transmisión de datos.

Para satisfacer las diferentes necesidades de comunicaciones dentro de un área personal la IEEE ha creado un grupo de trabajo el IEEE 802.15, el mismo que se

subdivide en cuatro grupos más: 802.15.1 (Bluetooth), 802.15.2(coexistencia entre WLAN Y WPAN), 802.15.3(Nuevo estándar de alta velocidad, mayores a 20 Mbit/s) y el 802.15.4 (Baja transmisión de datos). El grupo IEEE 802.15 establece tres tipos de WPANs. Las WPANs con un rango de velocidad elevada (802.15.3) diseñadas para aplicaciones multimedia con altos niveles de Calidad de Servicio (QoS, Quality of Service). Las WPANS de rango medio(802.15.1/Bluetooth) con QoS apropiado para aplicaciones de voz y la última clase son las LR-WPAN (802.15.4) de baja trasmisión de datos.

2.2.1.1 ZigBee

ZigBee es un estándar para comunicaciones inalámbricas desarrollado por la ZigBee Alliance¹⁴. Este estándar a su vez está basado en el estándar IEEE 802.15.4. Su principal objetivo son las aplicaciones que requieren comunicaciones inalámbricas seguras, a un costo muy bajo, con bajas tasas de trasmisión de datos y bajo consumo de energía.

La ZigBee Alliance es una comunidad internacional de más de 100 compañías como Motorola, Mitsubishi, Philips, Samsung, Honeywell, Siemens, entre otras; cuyo objetivo es habilitar redes inalámbricas con capacidades de control y monitoreo que sean confiables, de bajo consumo energético y de bajo costo, que funcione vía radio y de modo bidireccional; todo basado en un estándar público global que permita a cualquier fabricante crear productos que sean compatibles entre ellos. El nombre de ZigBee surgió de la idea de una colmena de abejas moviéndose alrededor de su panal y comunicándose entre ellas.

ZigBee funciona en la banda de frecuencia de 2,4 GHz y en 16 canales, y puede alcanzar una velocidad de transferencia de hasta 250 Kbps con un rango de cobertura de 10 a 75 metros. Las características propias de este estándar, de baja transmisión de datos, hacen que ZigBee no sea una competencia para otro tipo de redes como WiFi o Bluetooth porque sus aplicaciones son diferentes. Otra característica del estándar es la capacidad de operar en redes de gran densidad,

¹⁴ ZigBee Alliance. (2009). ZigBee Alliance, de: <http://www.zigbee.org/> (diciembre de 2009)

ya que entre más nodos existan dentro de una red, mayor número de rutas alternas existen. Esto gracias al protocolo de comunicación multi-salto, es decir, que se puede establecer comunicación entre dos nodos aún cuando estos se encuentren fuera del rango de transmisión, siempre y cuando existan otros nodos intermedios que los interconecten. Además cada red ZigBee tiene un identificador de red único, lo que permite que coexistan varias redes en un mismo canal de comunicación sin ningún problema. Teóricamente pueden existir hasta 16 000 redes diferentes en un mismo canal y cada red puede estar constituida por hasta 65 000 nodos, obviamente estos límites se ven truncados por algunas restricciones físicas (memoria disponible, ancho de banda, etc.)¹⁵.

La arquitectura de la pila de comunicación está formada por el estándar IEEE 802.15.4 que define la capa física y la capa MAC. La alianza ZigBee ha añadido las especificaciones de las capas red y aplicación para completar la pila o stack ZigBee y finalmente los fabricantes son los encargados de desarrollar de acuerdo a su mercado las aplicaciones.

2.3 EL ESTÁNDAR IEEE 802.15.4^[16]

En el año 2000 la alianza ZigBee y el grupo 15 de trabajo IEEE 802 se unieron para dar a conocer la necesidad de un nuevo estándar de redes inalámbricas de área personal con tasas bajas de transmisión de datos (LR-WPAN, Low-Rate Wireless Personal Area Network) y en diciembre del mismo año nace el estándar que se conoce como IEEE 802.15.4, conocido también como ZigBee que es su nombre comercial. La primera versión del estándar se publicó en el 2003 y luego en el 2006 se aprobó una revisión que mejoraba a la primera versión.

El IEEE 802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de las redes LR-WPAN. Estas redes se destinan para aplicaciones de un bajo consumo de energía y costos reducidos. Las LR-WPAN se aplican para una

¹⁵ Valverde, J. (2007). El Estándar Inalámbrico ZigBee. de <http://www.seccperu.org/files/ZigBee.pdf> (diciembre de 2009).

¹⁶ IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks.

baja transmisión de datos y baja QoS. Las principales características del estándar se resumen en la tabla 2.1.

PROPIEDAD	RANGO
Rango de transmisión de datos	868 MHz: 20Kb/s; 915 MHz: 40Kb/s; 2.4GHz:100 Kb/s o 250 Kb/s.
Alcance	10 - 75 m.
Latencia	Debajo de los 15 ms.
Canales	868 MHz: 3 canales 915 MHz: 30 canales. 2.4 GHz: 16 canales.
Bandas de frecuencia	868 MHz, 915 MHz y 2.4GHz.
Direccionamiento	Cortos de 8 o 64 bits IEEE
Canal de acceso	CSMA-CA y CSMA-CA ranurado
Temperatura	-40 °C a +85° C

Tabla 2.1: Características del estándar IEEE 802.15.4.

Fuente: Archundia, F. (2003). *Universidad de las Américas Puebla*. de Wireless Personal Area Network & Home Networking:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf.

2.3.1 TIPOS DE DISPOSITIVOS IEEE 802.15.4

El estándar IEEE 802.15.4 define dos tipos de dispositivos; el dispositivo de funciones completas (FFD, Full Function Device) y el dispositivo de funciones reducidas (RFD, Reduce Function Device). El FFD Puede funcionar como coordinador de una red de área personal (PAN), como coordinador sólo de su entorno o como un nodo normal, además los FFD puede comunicarse con otros FFD's y con RFD's. El RFD es un dispositivo muy sencillo con recursos y necesidades de comunicación muy limitadas, por ello, sólo puede comunicarse con otros FFD's y no puede ser un coordinador.

2.3.2 TOPOLOGÍAS DE RED IEEE 802.15.4

Dependiendo de los requerimientos de las aplicaciones de una red LR –WPAN IEEE 802.15.4, puede operar en dos tipos de topologías; estrella y peer to peer que se muestran en la figura 2.2. En la topología de estrella, existe un coordinador PAN y todos los dispositivos finales se comunican únicamente con este coordinador. En la topología peer to peer, también existe un coordinador PAN, pero en esta topología cada dispositivo puede comunicarse con otro dispositivo mientras estén en el mismo rango de comunicación.

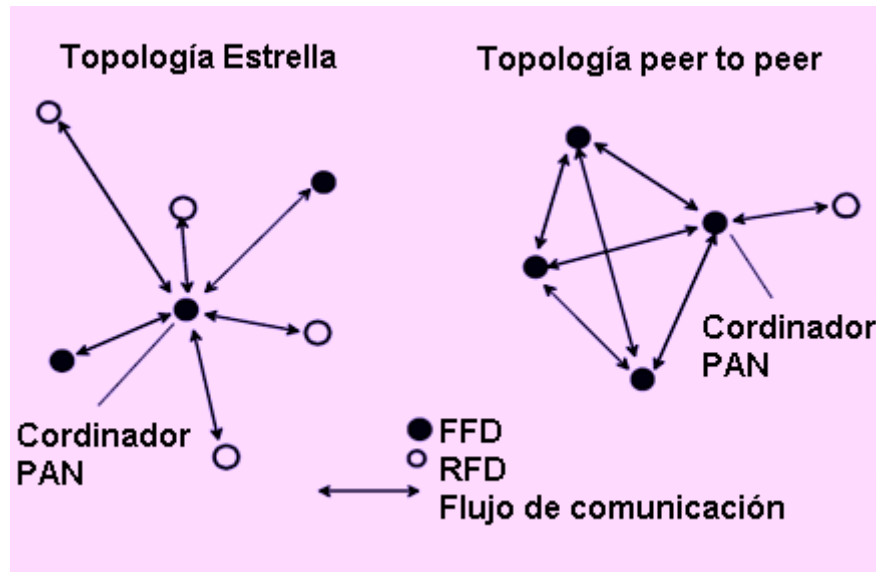


Figura 2.2: Topologías de estrella y peer to peer.

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate, pág 14.

Rediseñado por: Susana Mafla

2.3.3 ARQUITECTURA IEEE 802.15.4

La arquitectura del estándar IEEE 802.15.4 que se muestra en la figura 2.3; está definida en términos de capas; cada capa es responsable por una parte del estándar y ofrece servicios a la capa superior. Como se mencionó anteriormente el estándar IEEE 802.15.4 describe la capa física que provee la interfaz con el medio físico donde ocurre la comunicación y la capa MAC que proporciona el acceso al canal físico para todos los tipos de transferencia de información.

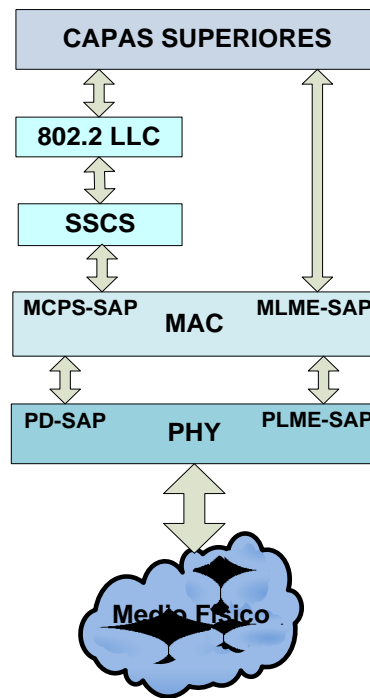


Figura 2.3: Arquitectura de dispositivos LR-WPAN

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY)

Rediseñado por: Susana Mafla

2.3.3.1 La Capa Física

La capa física provee dos servicios: el servicio de datos de capa física y el servicio de gestión de interfaz, con la entidad de gestión de la capa física (PLME, Physical Layer Management Entity) a través del punto de acceso al servicio (SAP, Service Access Point), conocido como el PLME-SAP. El servicio de datos permite la transmisión y recepción de las unidades de datos del protocolo de capa física (PPDUs, Physical Protocol Data Unit) a través del canal físico.

Las características de la capa física son la activación y desactivación del transceiver de radio, la selección de canales, la evaluación de canal dedicado (clear channel), y la transmisión así como la recepción de paquetes a través del medio físico.

- **Estructura del paquete de datos de la capa física**

La unidad de datos de protocolo de la capa física PDU (Physical Protocol Data Unit), establece al paquete como unidad. La PDU se encarga de encapsular todos los datos de los niveles superiores. Como se ve en la figura 2.4, se fundamenta en tres componentes: la cabecera de sincronización SHR (Synchronization Header), la cabecera de la capa física PHR (Physical Header) y la carga útil de la capa física (PHY payload).

		Octetos		
		1	Variable	
Preámbulo	SFD	Tamaño de trama (7 bits)	Reservado (1 bit)	PSDU
SHR		PHR		carga útil PHY

Figura 2.4: Formato del PDU

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate, pág 43.

Rediseñado por: Susana Mafla

Preámbulo.- Tiene un tamaño de 32 bits, está diseñado para la adquisición de símbolos, y en algunos casos se utiliza para ajustes bruscos en la frecuencia.

SFD.- Indica el final de la cabecera de sincronización y el inicio del paquete de datos.

Tamaño de trama.- Tiene un tamaño de 7 bits y especifica el número total de octetos que tiene el PSDU.

PSDU.- Este campo es de tamaño variable, va de 0 a 127 bytes y lleva los datos del paquete de capa física.

2.3.3.2 La capa MAC (Media Access Control o Control de Acceso al Medio)

La capa MAC define el procedimiento para acceder al medio físico empleado en una red para lograr el intercambio de datos. Esta capa además de encargarse del

control de acceso al medio, es responsable de transmitir los paquetes, validar las tramas que recibe, comprobar errores en la transmisión y confirmar la recepción de tramas al emisor. Otras funciones importantes que realiza la capa MAC son:

- Asociación/disociación
- Acuse de recibo (ACK)
- Mecanismos de acceso al canal
- Validación de trama
- Control de garantía de ranuras de tiempo (Slot Time)
- Control de guías (Beacon)
- Sondeo del canal (Scan)
- Empleo del mecanismo CSMA-CA para el acceso al canal
- Proporcionar un vínculo fiable entre dos entidades iguales MAC
- Control de flujo
- Gestión de batería

Las subcapas MAC proporcionan dos tipos de servicios hacia las capas superiores, que se acceden a través de dos puntos de acceso a servicios (SAPs). Los servicios de datos MAC se acceden por medio de la parte común de la subcapa MAC (MCPS-SAP, MAC Common Part Sublayer Service Access Point), y el manejo de servicios MAC, se accede por medio del punto de acceso a servicios de la entidad de gestión de la subcapa MAC (MLME-SAP, MAC Sublayer Management Entity Service Access Point). Estos dos servicios proporcionan una interface entre las subcapas de convergencia de servicios específicos (SSCS, Service-Specific Convergence Sublayer) y el Control de Enlace Lógico (LLC, Logical Link Control) con la capa física ¹⁷.

- **Control de acceso al medio**

El estándar IEEE 802.15.4 utiliza dos tipos de mecanismos de acceso al canal dependiendo de la configuración de red. Las redes PAN con beacon habilitado

¹⁷ Archundia, F. (2003). *Universidad de las Américas Puebla*. de Wireless Personal Area Network & Home Networking: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf (diciembre de 2009)

utilizan un mecanismo CSMA-CA ranurado y las redes PAN con beacon no habilitado un mecanismo CSMA-CA no ranurado.

En una red PAN con beacon habilitado el acceso al medio se controla en su totalidad por el coordinador PAN de la red mediante el uso de intervalos de tiempo de garantía o GTS's. En este mecanismo se utiliza una portadora dividida en intervalos de tiempo que permiten el acceso a los dispositivos durante un periodo libre de contención mediante CSMA-CA, cuando un dispositivo durante este periodo desea transmitir espera a que empiece la siguiente ranura de tiempo y luego determina si la ranura está ocupada por otra transmisión, si esto ocurre el dispositivo se extiende a un número aleatorio de ranuras o indica un error en la conexión.

Un coordinador de una red PAN con beacon habilitado puede opcionalmente limitar su tiempo de canal con una estructura de súper trama. Una súper trama está limitada por la transmisión de una trama beacon y puede tener una parte activa y una parte inactiva. El detalle de las súper tramas se analiza en un siguiente tema.

En una red PAN con beacon deshabilitado el acceso al canal se realiza de una manera distribuida mediante un algoritmo de backoff CSMA-CA. Cada vez que un dispositivo desea transmitir revisa si otro dispositivo se encuentra transmitiendo sobre el canal. Si es el caso espera un período de tiempo aleatorio hasta que el canal esté libre y pueda transmitir o se indica una falla de conexión después de varios intentos fallidos.

- **Espacio inter-trama (IFS, Interframe Space)**

La capa MAC del estándar IEEE 802.15.4 necesita una cantidad finita de tiempo para procesar los datos recibidos por la capa física. Para permitir esto, dos tramas sucesivas de transmisión de un dispositivo deben estar separadas por lo menos un período de tiempo IFS, si la primera transmisión requiere un acuse de recibo ACK, la separación entre la trama y el reconocimiento de la segunda transmisión es de al menos un período de IFS. La duración del período de IFS depende del

tamaño de la trama que se transmite. Las tramas de menor longitud van seguidas de un espacio inter-trama corto (SIFS, Short Interframe Spacing). Las tramas con longitudes mayores van seguidas de un período (LIFS, Long Interframe Spacing).

- **Primitivas**

En un modelo de comunicaciones basado en capas, cada capa utiliza los servicios de comunicaciones de la capa inmediatamente inferior y ofrece servicios a la capa inmediatamente superior. Un servicio se especifica de manera formal como un conjunto de primitivas (operaciones) disponibles para que un usuario acceda a un servicio. Los servicios se prestan a través de unos puntos de acceso al servicio (SAP, Service Access Point). Las primitivas indican al servicio que debe efectuar una acción o que notifique la acción que haya tomado la entidad similar.

El administrador de servicios MAC IEEE 802.15.4 tiene 26 primitivas. Estas primitivas pueden pertenecer a uno de los siguientes grupos genéricos:

- *Solicitud (Request)*: La primitiva de solicitud va desde un N-usuario a una N-capa para solicitar la inicialización de un servicio.
- *Indicación (Indication)*: La primitiva de indicación va desde una N-capa hacia un N-usuario para notificar al usuario del servicio sobre una acción o indicar que ha sido invocado un procedimiento por un N-usuario del servicio y para suministrar los parámetros asociados.
- *Respuesta (Response)*: La primitiva de respuesta va desde un N-usuario hacia una N-capa para completar el procedimiento anteriormente invocado por una primitiva de indicación.
- *Confirmación (Confirm)*: la primitiva de confirmación va desde una N-capa hacia un N-usuario para transmitir los resultados de las solicitudes de servicio anteriores.

- **Estructura de tramas MAC**

La estructura general de trama MAC del estándar IEEE 802.15.4 es muy flexible, simple y se ajusta a las necesidades de las diferentes aplicaciones. La unidad de datos de protocolos MAC (MPDU) que se muestra en la figura 2.5, se compone del encabezado MAC (MHR, MAC Header), la unidad de servicio de datos MAC (MSDU, MAC Service Data Unit) y del pie de trama MAC (MFR, MAC Footer).

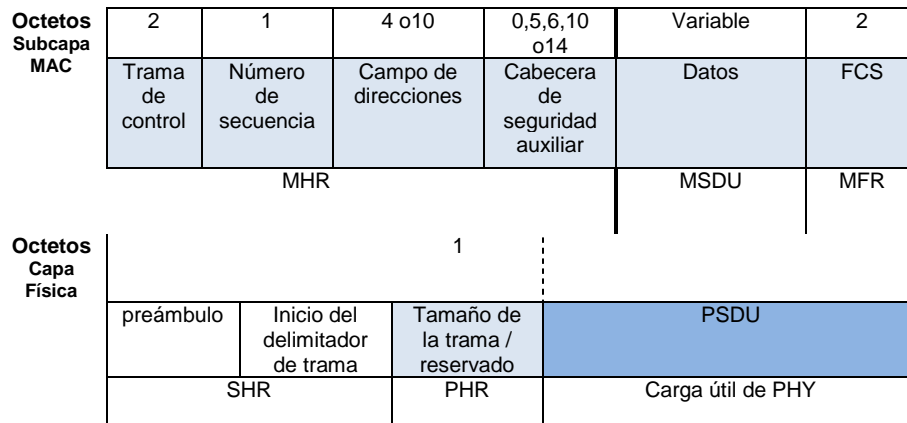


Figura 2.5: Unidad de Datos de Protocolos MAC (MPDU)

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY).

Rediseñando por : Susana Mafla.

El MHR consta de un control de trama (frame control) que indica el tipo de trama, el número de secuencia de datos (data sequence number), que verifica la integridad de la trama MAC, el campo de direcciones (address field) que es un campo de direccionamiento que contiene las direcciones de origen y destino de la trama. Además existe una cabecera de seguridad auxiliar que se puede o no activar y que contiene información de cómo está protegida la trama. El MSDU o Payload contiene la carga útil o datos y depende del tipo de trama. Sin embargo, la trama completa de MAC no debe de exceder los 127 bytes. El MFR o pie de la trama es un campo de chequeo de secuencia de trama (FCS, Frame Check Sequence), sirve para verificar la integridad de las tramas MAC utilizando un código de redundancia cíclica (CRC, Cyclic Redundancy Check) de 16 bits.

El estándar define cuatro tipos de tramas: trama beacon, trama de datos, trama de acuse de recibo y trama de comandos MAC.

- **Trama beacon (beacon frame)**

La trama beacon es habilitada por los dispositivos de funciones completas FFD's para localizar dispositivos de funciones reducidas RFD's y unirlos a la red. Los dispositivos RFD pueden activarse solamente cuando es transmitida una señal de guía o beacon, escuchan su dirección y si no la escuchan vuelven al estado inactivo, ahorrando así energía. La estructura de este tipo de trama se observa en la figura 2.6.

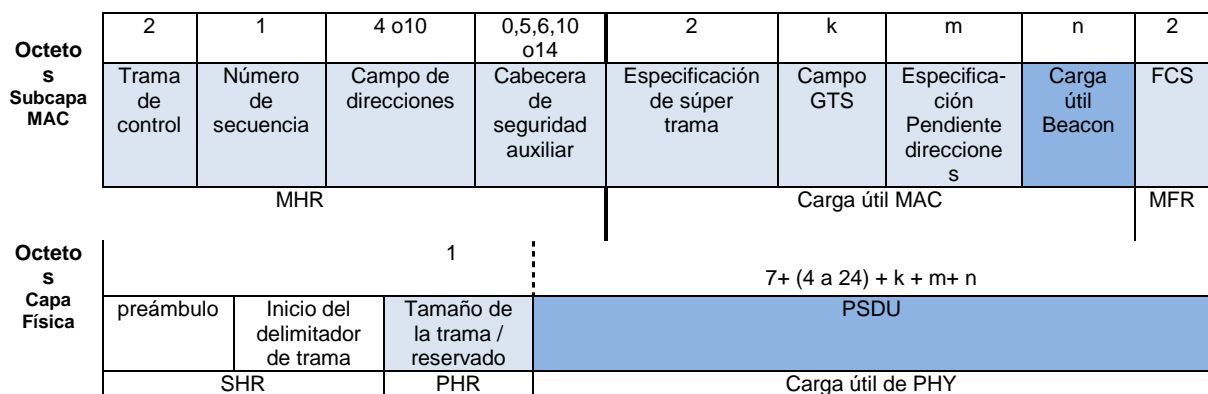


Figura 2.6: Formato de la trama Beacon

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY).

Rediseñando por : Susana Mafla.

En una trama beacon, el campo de direcciones contiene la fuente del Identificador de Red de Área Personal (PAN ID) y la fuente de direcciones de los dispositivos. La carga útil MAC de una trama beacon consta de los siguientes cuatro campos:

- *Especificación de la Súper trama (Superframe Specification):* Contiene los parámetros que especifican la estructura de la súper trama.
- *Especificación Pendiente de Direcciones (Pending Address Specification):* Contiene el número y tipo de direcciones especificadas en el campo de lista de direcciones.

- *Intervalos de Tiempo de Garantía (GTS, Guaranteed Time Slot):* Lleva los intervalos de tiempo asignados por el coordinador PAN.
- *Carga útil Beacon (Beacon Payload):* Es un campo opcional que contiene datos para todos los dispositivos participantes en la red (broadcast).

Esta estructura de la trama beacon contiene las denominadas súper tramas o en inglés Superframes. Algunas aplicaciones requieren anchos de banda dedicados para lograr grandes estados latentes, para un bajo consumo de energía. Para lograr dichos estados latentes el estándar IEEE 802.15.4 puede operar este modo opcional de súper tramas, en donde el coordinador de red, transmite súper tramas de guía llamadas tramas beacon en intervalos definidos. Estos intervalos pueden ser tan cortos como 15 ms o tan largos como 245 s. El tiempo entre cada uno de ellos se divide en 16 intervalos de tiempo o "time slots", que son independientes de la duración de cada súper trama. Un dispositivo puede transmitir cuando sea durante un intervalo de tiempo, pero debe terminar su transmisión antes de la siguiente trama beacon. El periodo de acceso de contención en los intervalos de tiempo es una contención de base; sin embargo el coordinador de PAN puede asignar intervalos de tiempo a un solo dispositivo que requiera un determinado ancho de banda permanentemente o transmisiones de baja latencia. Estos intervalos de tiempo asignados son llamados intervalos de tiempo de garantía (GTS, Guaranteed Time Slot). El tamaño del periodo libre de contención puede variar dependiendo de la demanda de los demás dispositivos asociados a la red; cuando el GTS se utiliza, todos los dispositivos deben completar todas sus transacciones de contención base antes de que el periodo

libre de contención comience¹⁸. En la figura 2.7 se ilustra una súper trama.

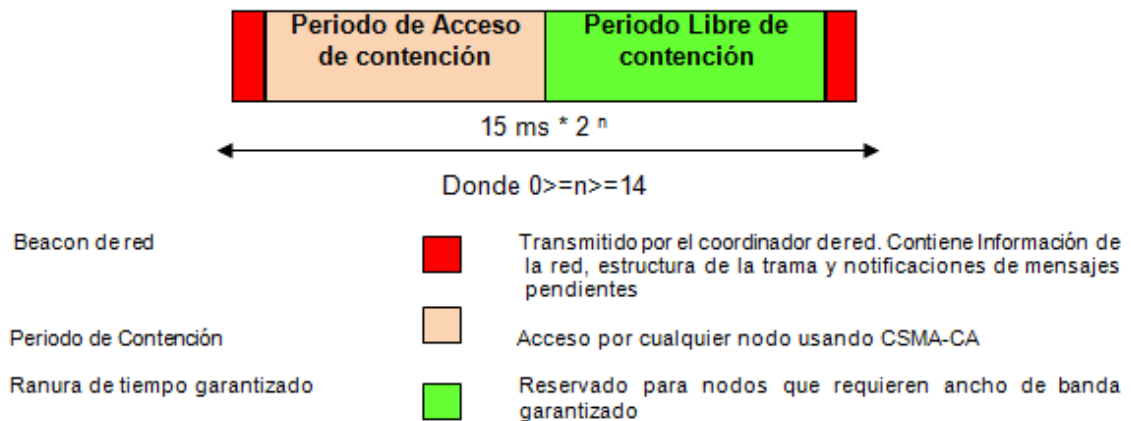


Figura 2.7: Estructura de las súper tramas.

Fuente: Mayné, J. (2007). Buenos Aires Robotics Society, de IEEE 802.15.4 y ZigBee: http://www.bairesrobotics.com.ar/data/ieee_zigbee_silica.pdf (enero de 2010).

Rediseñado por: Susana Mafla

- **Trama de datos (data frame)**

La trama de datos se visualiza en la Figura 2.8; esta trama es usada por la capa MAC para transmitir todos los datos.

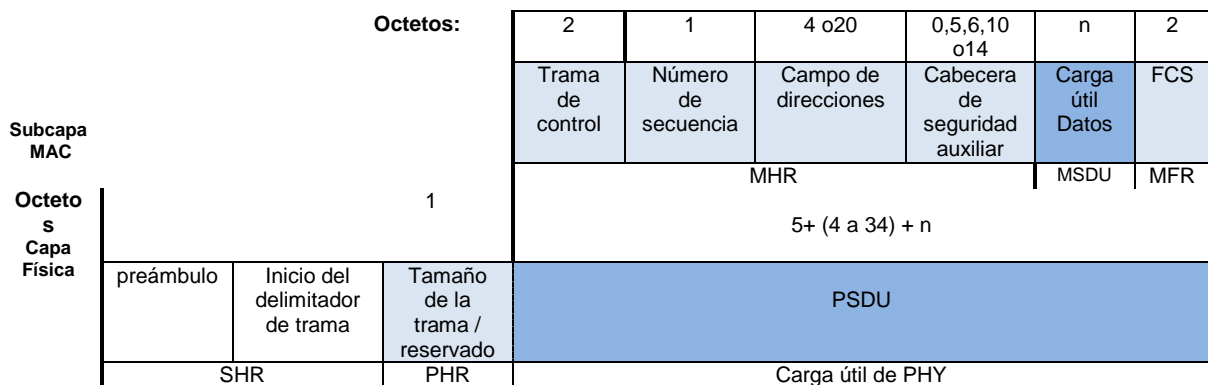


Figura 2.8: Formato de la Trama de Datos.

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks

Rediseñado por: Susana Mafla

¹⁸ Mayné, J. (2007). Buenos Aires Robotics Society, de IEEE 802.15.4 y ZigBee: http://www.bairesrobotics.com.ar/data/ieee_zigbee_silica.pdf (enero de 2010),

- **Trama de acuse de recibo ACK (acknowledgment frame)**

La trama de acuse de recibo se observa en la figura 2.9, esta trama es enviada por el receptor hacia el emisor, informándole de que el paquete fue recibido sin error. Este corto paquete aprovecha un tiempo de silencio inmediatamente después de la transmisión del paquete de datos para evitar posibles colisiones.

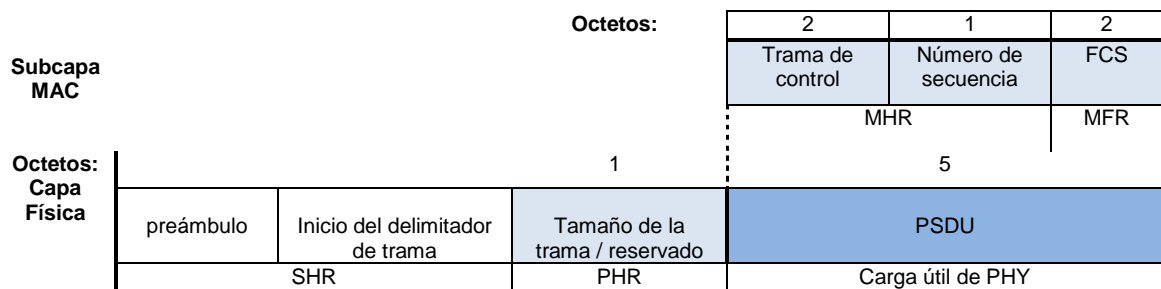


Figura 2.9: Formato de la trama de Acuse de Recibo.

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate, pág 22.

Rediseñado por: Susana Mafla .

- **Trama de comandos MAC (MAC command frame)**

La trama de comandos MAC es un mecanismo para el control o configuración a distancia de dispositivos. Permite que un coordinador PAN pueda configurar a los dispositivos individualmente sin importar lo grande que sea la red.

La carga útil de la MAC (MAC payload) que se puede observar en la figura 2.10, tiene dos campos: el tipo de comando MAC (MAC command type) y la carga útil del comando MAC (MAC command payload). El contenido de este último campo depende de los valores del tipo de comando que se visualizan en la tabla 2.2.

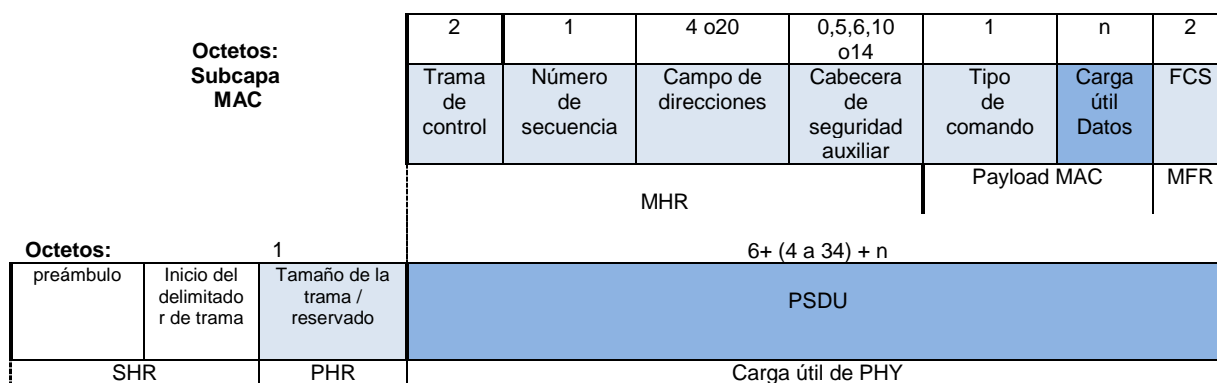


Figura 2.10: Formato de la Trama de Comandos MAC.

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY)
Rediseñado por: Susana Mafla.

Tipo de Comando	Carga útil del comando MAC
1	Solicitud de Asociación
2	Respuesta de Asociación
3	Notificación de Disociación
4	Petición de Datos
5	Conflicto en la notificación del identificador de red de PAN ID
6	Notificación de Huérfano
7	Petición de Trama Beacon
8	Realineación de Coordinador
9 - 255	Reservado

Tabla 2.2: Tipos de Comandos de Tramas MAC

Fuente: IEEE. (2006). Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate.
Rediseñado por: Susana Mafla.

2.4 EL SISTEMA MOBILE SMART NODE (MSN)

El Mobile Smart Node (MSN) es un sistema móvil de comunicación, desarrollado en el contexto de plataformas de inteligencia ambiental, cuyos campos de aplicación son la administración, optimización y uso eficiente de energía; y también la asistencia de personas en entornos domésticos. Este sistema es desarrollado como un proyecto del instituto de Sistemas y Tecnologías Distribuidos y Autónomos, VauST (Verteilte autonome Systeme und Technologien) de la Facultad de Ambiente y Tecnología de la Universidad Leuphana de Lüneburg de la República Federal de Alemania.

Básicamente el sistema está formado por una red de dispositivos inalámbricos, los cuales en conjunto analizan las condiciones ambientales, recopilan información, y basados en la tecnología de Inteligencia Ambiental, que se discutió en el capítulo anterior, toman decisiones para administrar eficientemente el uso de energía en ambientes domésticos. Estos dispositivos son capaces de decidir por sí solos el funcionamiento o no de aparatos eléctricos o la realización de alguna actividad que requiera del consumo de energía, si se ajustan a las condiciones de eficiencia energética. De esta manera, los desarrolladores del sistema pretenden contribuir a disminuir el impacto ambiental producido por la generación de energía que se pierde por ineficiencia y falta de administración energética en ambientes domésticos.

2.4.1 DISPOSITIVOS Y FUNCIONAMIENTO DEL SISTEMA MOBILE SMART NODE (MSN)^[19]

El sistema Mobile Smart Node (MSN), cuya traducción al español es “Nodo Inteligente Móvil”, está formado por tres tipos de nodos o dispositivos: el Mobile Smart Node (MSN), el Full Embedded Smart Node (FESN), en español “Nodo Inteligente Embebido Completo”, y Thin Embedded Smart Node (TESN), en español “Nodo Inteligente Embebido Ligero”. Todos los dispositivos del sistema se

¹⁹ VauST. (2008). User Manual for the Mobile Smart Node . Lüneburg.

componen de la misma plataforma de hardware, la única diferencia entre ellos es a nivel de software, ya que su programación es diferente según el tipo de dispositivo. El MSN es manejado por el usuario, el FESN actúa como una interfaz entre las preferencias del usuario almacenadas en el MSN y el TESN que controla los dispositivos eléctricos como se muestra en la figura 2.11.

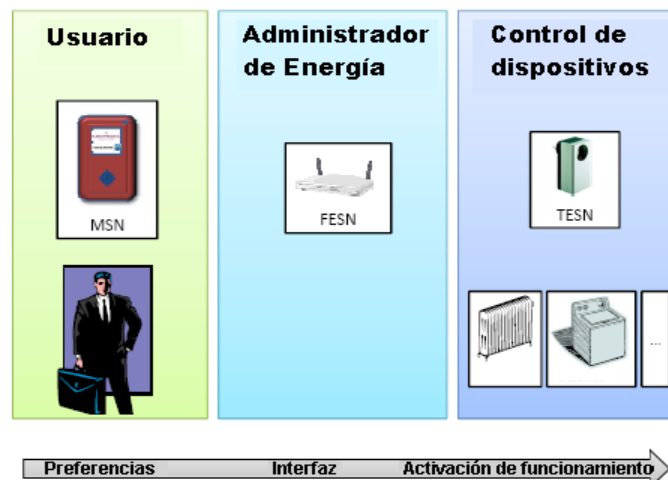


Figura 2.11: Dispositivos del Sistema MSN.

Fuente: VauST. (2008). User Manual for the Mobile Smart Node. Lüneburg.

El usuario del sistema a través del Mobile Smart Node (MSN) envía un mensaje, donde se guardan sus preferencias o información a cerca de la configuración del sistema. Este primer dispositivo, proporciona la interfaz entre el usuario y la red de Inteligencia Ambiental, representa la comunicación hombre-máquina. El segundo dispositivo el Full Embedded Smart Node (FESN) recibe el mensaje enviado por el (MSN) y aquí se decide si estas preferencias entran en el entorno de Inteligencia Ambiental. Ésta plataforma decide entre los deseos del usuario y lo que es mejor para el ambiente. Éste dispositivo también es el responsable del control y administración de toda la red. Una red del sistema MSN tiene una topología tipo estrella, formada por un FESN que desempeña las funciones de coordinador PAN o dispositivo FFD y alberga un máximo de 8 dispositivos MSN que actúan como dispositivos RFD en la red. Finalmente el control de los aparatos eléctricos conectados a la red se ejecuta a través del Thin Embedded Smart Node (TESN), que mediante un relé controla los aparatos eléctricos.

2.4.2 APLICACIONES DEL SISTEMA MSN EN EL USO EFICIENTE DE ENERGÍA

Como se describe anteriormente el sistema Mobile Smart Node se aplica específicamente en la administración, optimización y uso eficiente de energía eléctrica que es utilizada para el funcionamiento de los diferentes equipos, dispositivos y electrodomésticos que se encuentran en un ambiente doméstico. Las aplicaciones del sistema MSN son:

- Control de iluminación.
- Control de calefacción y aire acondicionado.
- Administración de encendido y apagado de electrodomésticos y equipos eléctricos.
- Control de portones y puertas eléctricas.

2.4.2.1 Control de iluminación

Para el control de iluminación de un hogar el sistema MSN utiliza relés y sensores de luz como dispositivos TESN. El usuario a través del MSN envía un mensaje al FESN para controlar la iluminación de su hogar y el FESN con ayuda de los sensores censa el nivel de iluminación y compara los deseos del usuario para activar o desactivar la iluminación optimizando el uso de energía.

2.4.2.2 Control de calefacción y aire acondicionado

De igual manera el control de calefacción y aire acondicionado se realiza utilizando sensores de temperatura, sensores de movimiento y relés como dispositivos TESN. Para que el usuario regule la temperatura de su hogar primeramente se censa la temperatura del ambiente a través de los sensores de temperatura y el FESN analiza si el resultado está en el rango normal, si no lo está, con ayuda los sensores de movimiento se determina el si las puertas y las ventanas están cerradas y luego de este proceso se ejecuta el encendido, regulación o apagado de los equipos de calefacción y aire acondicionado.

2.4.2.3 Administración de encendido y apagado de electrodomésticos y equipos eléctricos

También con el sistema MSN se puede controlar el encendido y apagado de los diferentes equipos que utilizan energía eléctrica para su funcionamiento, a través de relés que prenden o apagan dichos dispositivos según los deseos del usuario. Adicionalmente si se determina que ciertos equipos están encendidos y nadie los está utilizando se procede con la desactivación de los mismos para no derrochar energía eléctrica.

2.4.2.4 Control de portones y puertas eléctricas

El sistema MSN además es capaz de controlar los portones y puertas eléctricas de un hogar, con el mismo principio de funcionamiento del manejo de los dispositivos eléctricos.

En la figura 2.12 se esquematiza el funcionamiento y las aplicaciones de los diferentes dispositivos del sistema MSN.

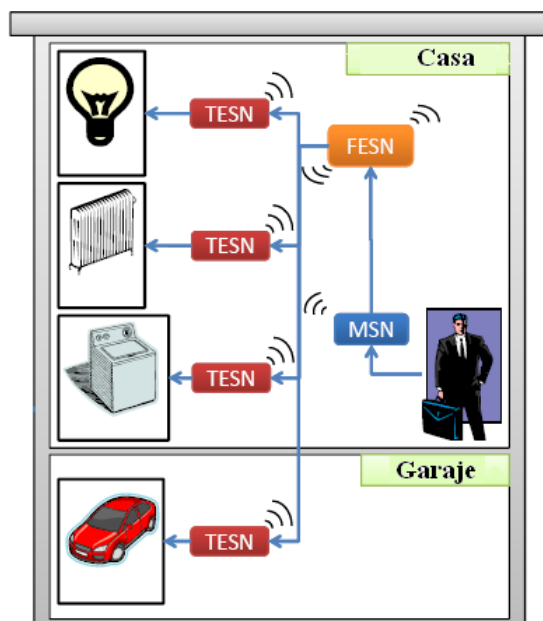


Figura 2.12: Funcionamiento del Sistema MSN.

Fuente: VauST. (2008). User Manual for the Mobile Smart Node. Lüneburg.
Rediseñado por: Susana Mafla.

2.4.3 DESCRIPCIÓN TÉCNICA DEL HARDWARE DEL MOBILE SMART NODE (MSN)

Como se ha mencionado el sistema Mobile Smart Node (MSN) está formado por tres tipos de dispositivos que se diferencian únicamente a nivel lógico entre sí. A nivel de Hardware el dispositivo físico del sistema es una sola placa base, la misma que está construida como un sistema embebido o integrado. Esto significa que la placa está diseñada para realizar un conjunto específico de tareas u operaciones, es única en su diseño y se encuentra formada por varios circuitos integrados que son una parte integral del sistema en el que se encuentran.

La placa base está compuesta principalmente por el microcontrolador LPC 2148 ARM7, de 32 bits del fabricante NXP, las características técnicas de este microcontrolador se encuentran en el [Anexo 1](#). La información se maneja por una palanca de 5 posiciones (5-way-joystick) y los datos se visualizan por un pantalla de diodos orgánicos de emisión de luz (OLED, Organic Light-Emitting Diode). Además, posee un módulo ZigBee de radio con antena para generación de las señales de comunicación, adaptadores para tarjeta SD y USB mini. La ilustración de los componentes de la placa se observan en las figuras 2.13 y 2.14, el listado de todos los elementos que forman la placa se detalla en el [Anexo 2](#)

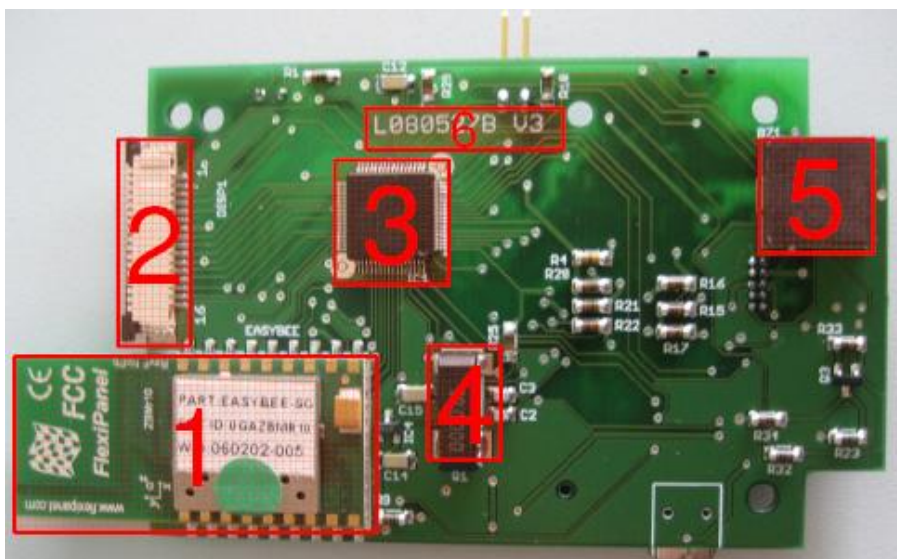


Figura 2.13: Dispositivo Mobile Smart Node, vista frontal.

Fuente: VauST. (2008). User Manual for the Mobile Smart Node. Lüneburg.

Acorde con la figura 2.13, el recuadro rojo marcado con el número 1 corresponde al Módulo ZigBee o Transceiver, el número 2 es el adaptador para display, el número 3 es el microcontrolador ARM, el número 4 es el cuarzo de 12 MHz, el número 5, ubicado en el lado derecho de la placa es el buzzer y finalmente el recuadro rojo con el número 6 indica el número de identificación del dispositivo.

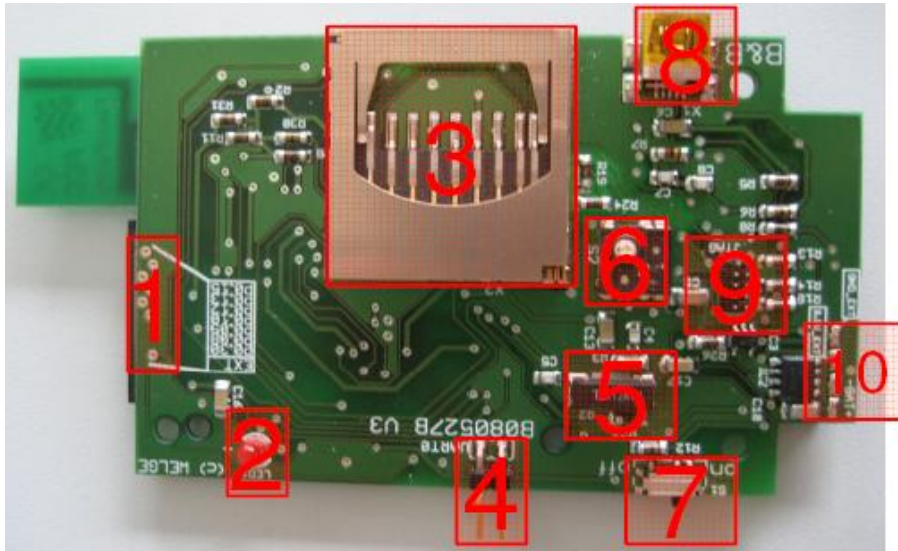


Figura 2.14: Dispositivo Mobile Smart Node, vista posterior.

Fuente: VauST. (2008). User Manual for the Mobile Smart Node. Lüneburg.

De igual manera en la figura 2.14, el recuadro rojo con el número 1 indica la posición de los pines externos, el número 2, ubicado en la parte inferior del lado izquierdo representa al led de carga, el número 3 es el slot para tarjeta SD, el número 4 indica dónde está el dispositivo para la comunicación UART 0, el número 5 es el cuarzo de 32Khz., el número 6 es el joystick, el número 7 es el switch para encender o apagar al dispositivo, el número 8 es un mini adaptador para interfaz USB, mientras que el número 9 es otro mini adaptador para interfaz JTAG (Joint Test Action Group) y finalmente el número 10 indica los pines utilizados para voltaje externo.

2.4.4 DESCRIPCIÓN DEL FUNCIONAMIENTO LÓGICO DEL MOBILE SMART NODE (MSN)

Técnicamente la plataforma de comunicación general del sistema Mobile Smart Node es la misma para todos los tres tipos de dispositivos y se indica en la figura 2.15.

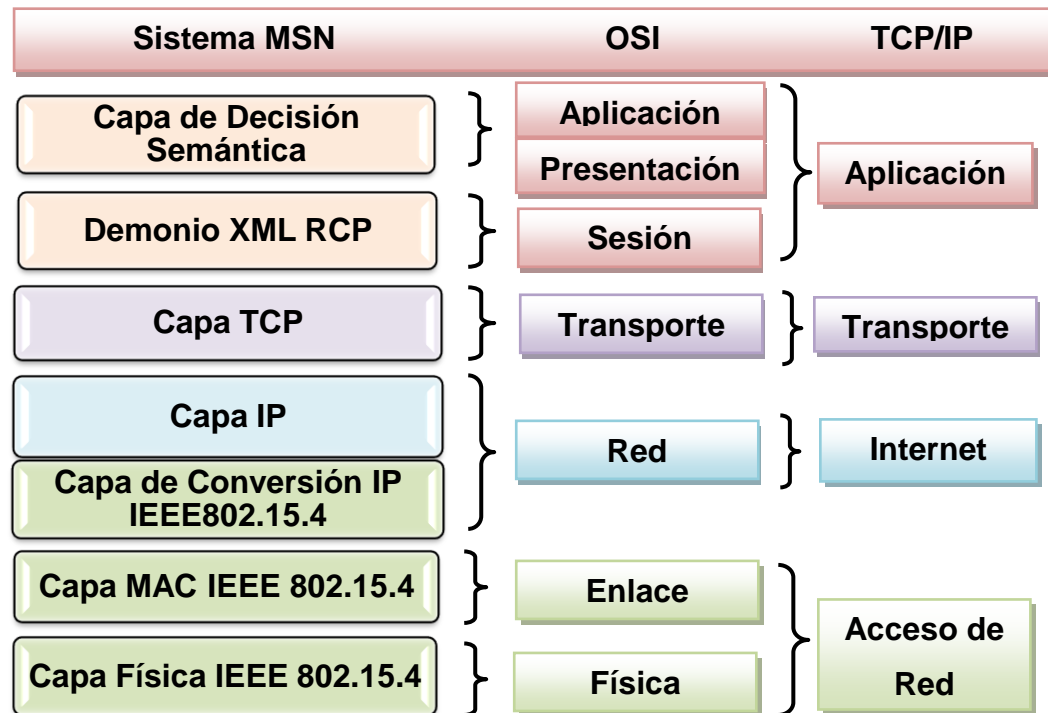


Figura 2.15 : Plataforma de comunicación del sistema MSN.

Fuente: VauST Welge, R. (2006). Embedded Internet, Embedded Middleware. Lüneburg.
Rediseñado por: Susana Mafla.

La arquitectura de comunicación del sistema MSN está basada en el modelo TCP/IP. Ésta arquitectura toma como referencia para la capa de acceso a la red el estándar IEEE 802.15.4, luego se encuentra la capa de internet o IP, seguida de la capa de transporte o TCP y la capa final de aplicación está compuesta por dos subcapas: la capa del demonio XML RCP y la capa de decisión semántica. A continuación se detallan las funciones de cada capa y subcapa del sistema.

2.4.4.1 Capa física IEEE 802.15.4

La capa define y reglamenta todas las características físicas - mecánicas y eléctricas que debe cumplir el sistema para operar. Como es el nivel más bajo, se encarga de las conexiones físicas del sistema hacia la red, es el responsable del manejo de la señal inalámbrica, del tipo de antena y la forma en que se transmite los bits de información. Además se ocupa de las características eléctricas de los diferentes componentes y las interfaces de comunicación.

Algunos componentes que forman parte de la capa física del sistema ya fueron nombrados anteriormente en la descripción del hardware del dispositivo. Sin embargo, el componente más representativo en cuanto a comunicación para la capa física del sistema MSN, es el Transceiver de ZigBee RF CC2420²⁰, diseñado para aplicaciones inalámbricas de baja potencia y bajo voltaje. Trabaja en la banda de licencia libre a nivel mundial de 2.4GHz, incluye un modem de banda base que utiliza Espectro Expandido en Secuencia Directa (DSSS, Direct Sequence Spread Spectrum) con ganancia de propagación de 9 dB y una tasa de transmisión de datos de 250 Kbps, usa modulación O-QPSK, consume poca corriente (RX: 18,8 mA, TX: 17,4 mA) y trabaja a baja tensión (2,1 - 3,6 V), más información en el [Anexo 3](#).

2.4.4.2 Capa MAC IEEE 802.15.4

La capa MAC tiene que coordinar el acceso a un canal compartido de radio y utilizar su capa física para detectar la portadora. La capa MAC comprueba una forma de evitar colisiones, como condición para permitir el acceso al medio. Esta capa tiene dos principales propósitos que son: el encapsulamiento de datos y el control de acceso al medio.

²⁰ Chipcon Products from Texas Instruments. (2006). CC2420 2.4GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Oslo.

- **Encapsulamiento de datos**

Se refiere al ensamblado de las tramas antes de su transmisión, y la detección de errores en la trama durante la recepción de la misma. Este proceso es responsable de las funciones de direccionamiento y del chequeo de errores. El encapsulado es realizado por la estación emisora, es el acto de agregar información, direcciones y bytes para el control de errores, al comienzo y al final de la unidad de datos transmitidos. La información añadida es necesaria para realizar las siguientes tareas:

- Sincronizar la estación receptora con la señal.
- Indicar el comienzo y el fin de la trama.
- Identificar las direcciones tanto de la estación emisora como la receptora.
- Detectar errores en la transmisión.

El desencapsulado a su vez es realizado por la estación receptora. Cuando es recibida una trama, la estación receptora es responsable de realizar las siguientes tareas:

- Reconocer la dirección de destino y determinar si coincide con su propia dirección.
- Realizar la verificación de errores.
- Remover la información de control que fue añadida por la función de encapsulado de datos en la estación emisora.

- **Control de acceso al medio**

EL control de acceso al medio constituye la inicialización de la transmisión de la trama y la recuperación de transmisiones fallidas. En la estación emisora, la función de administración de acceso al medio es responsable de determinar si el canal de comunicación se encuentra disponible; si el canal se encuentra disponible puede iniciarse la transmisión de datos. Adicionalmente, la función de administración es responsable de determinar que acción deberá tomarse en caso de detectarse una colisión y cuándo intentará retransmitir. En la estación

receptora la función de administración de acceso al medio es responsable de realizar las comprobaciones de validación en la trama antes de pasarla a la función de desencapsulado. Sin embargo, en el sistema MSN no se encuentra desarrollado un mecanismo para determinar si el canal está disponible, ocasionando que se produzcan colisiones que la capa MAC del sistema tampoco sabe resolver ya que el mecanismo CSMA-CA definido por el estándar IEEE 802.15.4 para el acceso al canal no está implementado en el sistema MSN.

2.4.4.3 Capa de conversión IP IEEE 802.15.4

La capa de conversión IP tiene dos trabajos: La definición y asignación de IPs; y la asociación de IPs con direcciones MAC.

- **Definición y asignación de IPs**

Cada dispositivo necesita una dirección IP para funcionar dentro del sistema MSN. Para asignar direcciones IP en el sistema se ha desarrollado un protocolo sencillo basado en el Protocolo Configuración Dinámica de Servidor, DHCP (Dynamic Host Configuration Protocol), éste protocolo sirve para distribuir direcciones IP en una red en forma dinámica, es decir, sin intervención particular. Para su funcionamiento se necesita de un servidor DHCP que asigne las direcciones IP a los equipos que forman la red, en el sistema MSN este servidor se encuentra en el FESN, aquí se encuentra un rango determinado de direcciones IP las cuales son asignadas a los MSN y a los TESN apenas estos son inicializados dentro del área de la red.

- **Asociación de IPs con direcciones MAC**

Otra función de esta capa es asociar una dirección física proporcionada por la capa MAC IEEE 802.15.4 con una dirección lógica o IP que es utilizada por las capas superiores.

Para la realización de este proceso de asociación la capa de conversión IP utiliza el Protocolo de Resolución de Direcciones (ARP, Address Resolution Protocol) ²¹. Este protocolo se encarga de enlazar direcciones IP de alto nivel con direcciones físicas de bajo nivel. El Software de enlace de direcciones marca una frontera entre las capas superiores del protocolo de software, que solo usan direcciones IP, y las capas inferiores del software controlador de dispositivos, que solo utilizan direcciones MAC.

El protocolo ARP sondea a los dispositivos de la red del sistema MSN para averiguar sus direcciones físicas y luego crea una tabla de búsqueda entre las direcciones lógicas y físicas en una memoria caché. Cuando un dispositivo debe comunicarse con otro, consulta la tabla de búsqueda. Si la dirección requerida no se encuentra en la tabla, el protocolo ARP envía una solicitud a la red. Todos los dispositivos en la red comparan esta dirección lógica con la suya. Si alguno de ellos se identifica con esta dirección, el dispositivo responderá al ARP, que almacenará el par de direcciones en la tabla de búsqueda y a continuación podrá establecerse la comunicación.

2.4.4.4 Capa IP

Este nivel define el enrutamiento y el envío de paquetes entre la red, además establece, mantiene y termina las conexiones. Este nivel controla la congestión de los paquetes de información en la red inalámbrica. Aquí se realiza el direccionamiento lógico y la determinación de la ruta de los datos hasta su receptor final IP.

Los dispositivos del sistema MSN forman una red móvil inalámbrica Ad-Hoc (MANET), donde se utiliza el protocolo de Vector Distancia, RFC3561 (AODV, Ad hoc On-Demand Distance Vector) ²² como protocolo de enrutamiento. Este protocolo entra en acción cuando se va a enviar un paquete a una dirección IP hacia la cual no se tiene una ruta conocida lo cual es frecuente en una red Ad –

²¹ Comer, D., & Stevens, D. (2000). Interconectividad de Redes con TCP/IP. Prentice Hall.

²² Peterson, L., & Davie, B. (2007). COMPUTER NETWORKS. Amsterdam

Hoc normal. AODV hace un broadcast hasta encontrar el dispositivo con la dirección IP requerida; pasando por los nodos intermedios que se encuentran entre los equipos que desean comunicarse. El algoritmo del protocolo calcula la dirección y la distancia hasta cualquier enlace en la red y elige el más corto para establecer la ruta.

2.4.4.5 Capa TCP

Este nivel actúa como un puente entre los niveles inferiores totalmente orientados a las comunicaciones y los niveles superiores totalmente orientados al procesamiento de la información. Además, garantiza una entrega confiable de la información entre los dispositivos. Asegura que la llegada de datos del nivel de la capa IP encuentren las características de transmisión y calidad de servicio requerido por el siguiente nivel (Sesión). Algunas de las funciones de este nivel son:

- Define cómo direccionar la localidad física de los dispositivos de la red.
- Asigna una dirección única de transporte a cada usuario.
- Define una posible multicanalización, es decir que puede soportar múltiples conexiones.
- Define la manera de habilitar y deshabilitar las conexiones entre los nodos.
- Los protocolos usados por esta capa son el TCP (Transmission Control Protocol) y UDP (User Datagram Protocol).

En resumen la tarea de esta capa es la integridad de datos de extremo a extremo, o sea que se encarga el flujo de datos del transmisor al receptor verificando la integridad de los mismos por medio de algoritmos de detección y corrección de errores.

2.4.4.6 Demonio o daemon XML RCP

Este nivel permite que los dispositivos del sistema Mobile Smart Node (MSN, FESN y TESN) establezcan sesiones entre ellos, especialmente permite que los dispositivos del sistema se comuniquen entre ellos bajo el paradigma de cliente servidor para que puedan transmitir datos. El paradigma cliente servidor consiste en que la comunicación es iniciada por uno de los programas que se denomina cliente, el segundo programa que espera que el otro inicie la comunicación se denomina servidor y atiende a las peticiones del cliente. Para el caso específico del presente sistema el Full Embedded Smart Node (FESN) es el dispositivo que hace las funciones de servidor, el Mobile Smart Node (MSN) y el Thin Embedded Smart Node (TESN) son los clientes. Estos dos últimos por ejemplo solicitan al Full Embedded Smart Node (FESN) la asignación de una dirección IP al iniciar una sesión.

Los programas de los tres dispositivos pueden intercambiar flujos de datos de manera fiable y ordenada, ya que estos cuentan con una dirección IP, un protocolo de transporte (UDP) y un número de puerto. El Demonio de llamada de procedimiento remoto XML (XML-RCP XML, Remote Procedure Call), que está en el dispositivo servidor y en el dispositivo cliente, sirve para que los programas (cliente, servidor) lean y escriban la información. Esta información será luego transmitida por las diferentes capas de la pila de comunicación.

El demonio XML RCP es un proceso no interactivo, no dispone de una "interfaz" directa con el usuario, ya sea gráfica o textual y se ejecuta en forma continua. La plataforma de comunicación del MSN utiliza el protocolo XML RCP que es un protocolo de llamada a procedimiento remoto utilizado por el lenguaje XML para codificar los datos por el Protocolo de Transferencia de Hipertexto (HTTP, HyperText Transfer Protocol) como protocolo de transmisión de paquetes, como se puede observar en la figura 2.16.

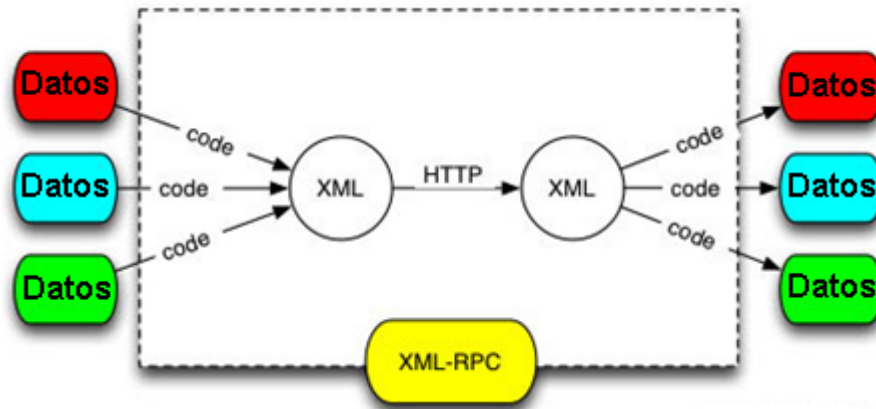


Figura 2.16: Funcionamiento del protocolo XML- RCP.

Fuente: UserLand Software. (2004). *XML-RCP.com*. De: com: <http://www.xmlrpc.com/> (diciembre de 2009)

Una llamada a procedimiento remoto (RCP)²³ consiste en un protocolo que permite a un software o programa ejecutar código en otra máquina remota sin preocuparse por la comunicación, por lo regular es bastante utilizado en el paradigma cliente y servidor. Siendo el cliente el que inicia el proceso solicitando al servidor que ejecute cierto procedimiento o función y enviando éste de vuelta el resultado de dicha operación al cliente.

XML-RPC²⁴, es un protocolo de llamada a procedimiento remoto que funciona a través de Internet para codificar datos. Un mensaje XML-RPC es una petición HTTP-POST, el cuerpo de la petición está en XML y el procedimiento que se ejecuta en el servidor y el valor que devuelve también está en formato XML. Los parámetros de procedimiento pueden ser escalas, números, series, fechas, etc., y también pueden ser complejos registros y estructuras de la lista.

2.4.4.7 Capa de decisión semántica

Finalmente este último nivel representa a la capa de aplicación del sistema Mobile Smart Node, es aquí donde se encuentran programadas todas las condiciones de uso eficiente de energía y las acciones que el usuario puede llevar a cabo con el

²³ Stervinou, J. (2004). *XML-RCP.com*. De: <http://www.xmlrpc.com/>(diciembre de 2009).

²⁴ UserLand Software. (2004). *XML-RCP.com*. De: <http://www.xmlrpc.com/> (diciembre de 2009).

sistema. Es el nivel de interfaz con el usuario y en términos de programación es el programa construido en lenguaje de alto nivel (C) para que los usuarios interactúen con todo el sistema. La capa de aplicación es responsable por lo siguiente:

- Identificar y establecer la disponibilidad de los socios de la comunicación deseada.
- Sincronizar las aplicaciones.
- Establecer acuerdos con respecto a los procedimientos para la recuperación de errores.
- Controlar la integridad de los datos.

A diferencia de las otras capas, por ser la más alta, no proporciona ningún servicio a otro nivel, lo cual es característico de las capas de aplicación, sí en cambio proporciona servicios de red a las aplicaciones del usuario, lo que además significa proporcionar el acceso a los servicios de las demás capas. También ésta capa define el protocolo para intercambiar datos entre aplicaciones, en este caso el protocolo utilizado es el HTTP.

2.4.5 PILA IP VERSÁTIL PARA SISTEMAS EMBEBIDOS (VISE, VERSATILE IP STACK FOR EMBEDDED SYSTEMS) ^[25]

En la actualidad el desarrollo de sistemas embebidos que forman redes Ad Hoc ha crecido considerablemente, es así que el instituto VauST de la Universidad Leuphana de Lüneburg, bajo la dirección del Prof. Dr. Ralph Welge ha creado una arquitectura de pila IP versátil para sistemas embebidos VISE, orientada para su uso en el sistema MSN. Éste software puede considerarse como un modelo para aplicaciones de servidores web embebidos, además está específicamente diseñado para las necesidades de L3-NET que significa baja potencia, bajo costo y baja velocidad de datos (Low Power, Low Cost, Low Datarate).

²⁵ Welge, R. (2006). VISE pila IP versátil para sistemas embebidos, Lüneburg,

Acorde con el modelo TCP/IP, el programa VISE define la capa de acceso; es decir la capa física y la capa MAC en base al estándar IEEE 802.15.4, la subcapa LLC se define en base a el estándar IEEE 802.2. Luego en la capa de internet se manejan los protocolos IP, ARP e ICMP, la capa de transporte maneja los protocolos TCP y UDP. Finalmente en la capa de aplicación se utiliza el protocolo HTTP a través del puerto 80 para aplicaciones de servidor web y para otras aplicaciones como servidor DHCP se utiliza el puerto 67 y para el cliente DHCP el puerto 68. En la figura 2.17 se muestra un gráfico con los protocolos de internet utilizados por la pila VISE y se hace una relación con el modelo TCP/IP y el modelo OSI.

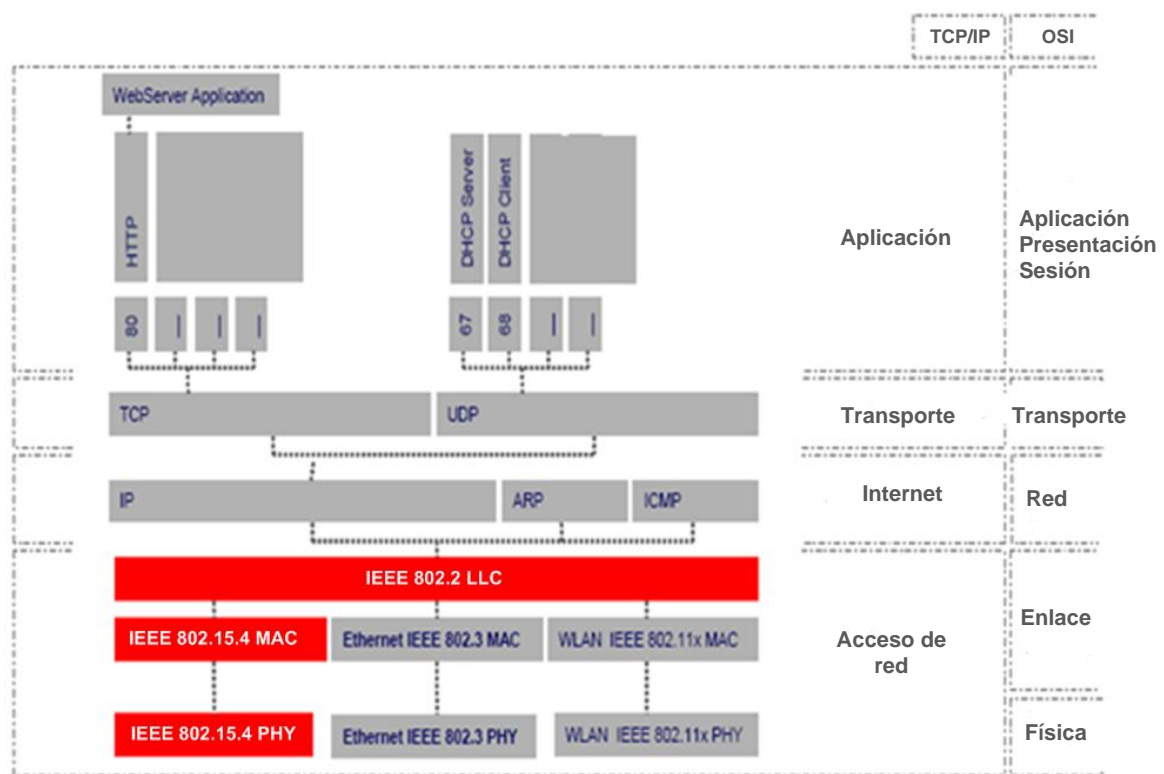


Figura 2.17: Protocolos de Internet de la pila VISE.

Fuente: Welge, R. (2006). Embedded Internet, Embedded Middleware. Lüneburg.
Rediseñado por: Susana Mafla.

Adicionalmente se encuentra un resumen de los protocolos manejados por VISE en el [Anexo 4](#).

El programa VISE es un software complejo desarrollado bajo la plataforma de CrossWorks de Rowley Associates²⁶; y programado en lenguaje C, la imagen completa del proyecto se puede ver en el [Anexo 5](#).

2.4.6 ANÁLISIS DEL PROGRAMA VISE

Para comprender la estructura lógica del programa VISE desarrollado por el Instituto VauST, se resume a continuación los componentes principales que integran el conjunto de funciones que estructuran la pila IP versátil para sistemas embebidos VISE.

Como se viene mencionando VISE es el software desarrollado en el programa CrossWorks en el cual se ha programado toda la estructura lógica de una pila IP para el sistema MSN. El código del programa VISE que se analiza a continuación se adjunta en formato digital.

El programa VISE se encuentra estructurado por un programa principal o Main, luego se inicializa un sistema operativo que se denomina Reactor, se inicializan las tareas principales que ejecuta el sistema operativo y finalmente el reactor se mantiene en un ciclo infinito para recibir las solicitudes que realizan las diferentes aplicaciones. En la figura 2.18 se observa un diagrama con la estructura del programa VISE.

²⁶ Rowley Associates (2010), CrossStudio Tutorial, de:
http://www.rowleydownload.co.uk/documentation/arm_2_0/index.htm (diciembre de 2010)

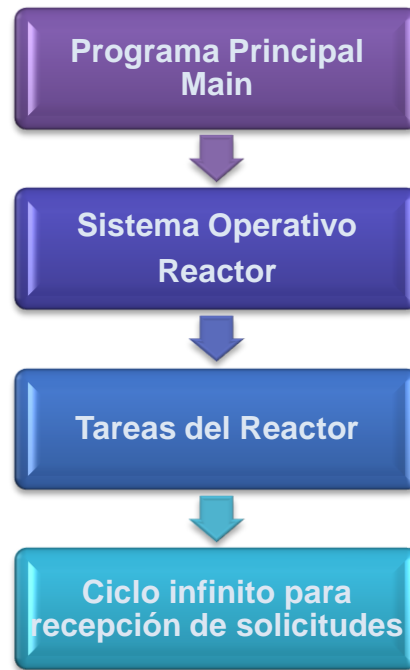


Figura 2.18: Estructura del proyecto VISE

Diseñado por: Susana Mafla

A continuación se detalla el funcionamiento de cada componente principal del programa VISE.

2.4.6.1 Programa Principal o Main

Como en todo programa el proyecto VISE empieza ejecutando la primera línea de la función main que contiene el algoritmo o módulo principal del programa, el main del proyecto VISE se denomina *systemStart.c*.

Para comenzar en el main se incluyen los archivos de cabecera o *Header file .h*, que contienen las declaraciones de variables y funciones para ser compartidas con otros archivos. Los archivos incluidos en el main *systemStart.c* inicializan la configuración de la comunicación UART, las interrupciones, los relojes y las definiciones de los dispositivos del sistema MSN como FESN o MSN a elección del programador.

Continuando en el main se desactiva todas las solicitudes de interrupción IRQ y se setea todos los registros. Además se establece la velocidad de transmisión y se inicializa la comunicación UART, una vez inicializada esta comunicación se envía a imprimir un mensaje que informa la frecuencia de reloj del CPU, la frecuencia del reloj periférico y la velocidad de transmisión.

Finalmente en el main se llama a la función *startReactor ()*; que inicializa el Reactor.

2.4.6.2 Reactor

El reactor es el núcleo o kernel del proyecto VISE, es el programa que actúa de sistema operativo para la familia NXP LPC 2000 de microcontroladores ARM7. El reactor es el responsable de facilitar a las distintas aplicaciones acceso seguro al hardware del sistema embebido, gestionar los distintos programas o tareas y administrar el hardware (memoria, periféricos de entrada/salida, etc.)

Cuando se ejecuta el reactor primeramente se inicializan los puertos del microcontrolador, luego conforme a las características definidas en el main se indica el tamaño de la memoria disponible que es de 11648 bytes, el número y tamaño de los bloques de memoria (1456 bloques de 8 bytes) y la cantidad de memoria que usa los mapas de bits (182 bytes).

Luego se confirma que la comunicación UART está activada y se setea el callback para recibir tramas de datos a través de la comunicación UART. Continuando se inicializan los timers del reactor y los timers para los contadores definiendo al contador máximo con el valor de 10 y al contador máximo rápido con el valor de 5. Se verifica que el sistema no esté en interrupción y se mide la latencia con respecto a la velocidad de la consola.

Finalmente se envía mensajes informando que se ha completado la inicialización del reactor y que se va a continuar con la inicialización de las tareas del reactor llamando a la función *reactor_start_tasks ()*;

2.4.6.3 Tareas del Reactor (Reactor Tasks)

Las tareas del reactor son las funciones que se encargan de estructurar e inicializar las capas de la pila IP y definir los protocolos que se utilizan en ellas.

En la función *reactor_start_tasks()*; primeramente se verifica el tipo de microcontrolador utilizado, para el caso del sistema MSN es el LPC2148 y luego se inicializa el administrador de buffers, donde se establece los parámetros de la red detallados en la tabla 2.3.

Parámetro	Valor
Número de buffers	20
Número de redes	20
Tamaño máximo de trama	630 bytes

Tabla 2.3: Parámetros de red

Diseñado por: Susana Mafla

Luego se inicializa la capa IP y el manejo de los buffers utilizados por esta capa, se establece el número de buffers que se puede mapear en 5 y se inicializa la capa de transporte. La siguiente tarea que se ejecuta inicializa el protocolo TCP que permite el mapeo de buffers TCP y se define los parámetros que se encuentran en la tabla 2.4.

Parámetro	Valor
Número máximo de buffers que se puede escuchar	4
Tamaño máximo de la ventana	100
Ciclos de terminación del tiempo	5

Tabla 2.4: Parámetros de los buffers TCP

Diseñado por: Susana Mafla

Continuando con las tareas del reactor se define el acceso al protocolo http por el puerto 80, si este puerto no está habilitado se envía un mensaje informando su estado y se activan las funciones para transmitir mensajes bajo el protocolo http. Se continúa inicializando el Joystick de la placa del dispositivo y luego se crea las tablas de distancias de rutas para la ejecución del protocolo de enrutamiento AODV que se establece bajo los parámetros que se encuentran en la tabla 2.5.

Parámetros	Valor
Máxima distancia de rutas	2^{32}
Tamaño de la tabla	10
Tamaño de memoria caché para broadcast	10
Tiempo de espera para activar la ruta	3000
Tiempo de espera para ruta propia	2* (Tiempo para activar la ruta)
Siguiente salto desconocido	65534

Tabla 2.5: Parámetros para enrutamiento AODV

Diseñado por: Susana Mafla

Continuando en con las tareas del reactor se inicializa la capa de puente entre la capa IP y la capa MAC, de acuerdo al tipo de dispositivo se asigna una dirección IP y una máscara de red. Luego se ejecuta la función *SAP_CC2420_open ()*;

- **Manejo del transceiver Easybee CC2420**

La función *SAP_CC2420_open ()*; inicializa el transceiver Easybee CC2420 donde primeramente se inicializa el bus de interfaz de periféricos serie SPI (Serial Peripheral Interface) que se utiliza para la comunicación entre el transceiver y el microcontrolador, se configura el control de registros y luego se configura los pines para operar en modo máster, como la comunicación SPI es sincrónica se espera al oscilador para sincronizarse, se apaga la seguridad y se establece el

método de acceso FIFOP (First-In-First-Out stack with Priorities). Para finalizar se establece las direcciones de las IRQ FIFO, escogiendo pines y flancos.

En seguida se comprueba estado de conexión y si el transmisor está activado para poder recibir datos.

Se hace una comparación para identificar al tipo de dispositivo ya sea MSN o FENS. Si el dispositivo está definido como MSN se envía un mensaje al reactor para indicar que el dispositivo es un cliente dentro del sistema y por el contrario si el dispositivo está definido como FESN se envía un mensaje al reactor donde se informa que el dispositivo es el servidor dentro del sistema.

Se finalizan las tareas del reactor y se imprime un mensaje informando que todas las tareas se han inicializado exitosamente y que se empieza a ejecutar el bucle de mensajes.

2.4.6.4 Ciclo infinito para recepción de solicitudes

Para finalizar se llama a la función *reactor_run ()*; su tarea es crear un ciclo infinito para recibir los mensajes que le llegan al reactor. Este ciclo infinito chequea la cola de mensajes que en realidad son solicitudes que llegan al reactor y luego son entregadas al puerto correcto del reactor para ser atendidas.

3 PROTOCOLO DE CAPA MAC



3.1 GENERALIDADES

3.2 PROTOCOLO DE ACCESO MÚLTIPLE POR DETECCIÓN DE PORTADORA CON ANULACIÓN DE COLISIONES (CSMA-CA)

3.3 ACCESO MÚLTIPLE CON DETECCIÓN DE COLISIONES (MACA)

3.4 ACCESO MÚLTIPLE CON DETECCIÓN DE COLISIONES PARA WIRELESS (MACAW)

3.5 FUNCIONES DE LA CAPA MAC DEL ESTÁNDAR IEEE 802.11

3.6 CONTEXTUALIZACIÓN DEL PROBLEMA

3.7 ANÁLISIS DE SOLUCIÓN PARA EL DESARROLLO DEL PROTOCOLO DE CAPA MAC

3.8 IMPLEMENTACIÓN DEL PROTOCOLO DE CAPA MAC

3.1 GENERALIDADES

Hasta ahora se ha analizado los conceptos teóricos necesarios para llegar a este punto en donde se desarrolla un protocolo de capa MAC. Éste protocolo propuesto posteriormente será implementado en el sistema Mobile Smart Node para eliminar problemas de colisiones.

Primeramente se menciona el Protocolo de Acceso Múltiple por Detección de Portadora con Anulación de Colisiones (CSMA-CA, Carrier Sense Multiple Access-Collision Avoidance), se detalla tanto su funcionamiento como los problemas que presenta, a raíz de estos problemas se analiza el Protocolo de Acceso Múltiple con Prevención de Colisiones (MACA, Multiple Acces With Collision Avoidance) que nació con el objetivo de mejorar a CSMA-CA, a su vez se describe el protocolo de Acceso Múltiple con Detección de Colisiones para Wireless (MACAW, Multiple Acces With Collision Avoidance for Wireless), que también añade mejoras al protocolo MACA y además se analiza el funcionamiento de la capa MAC del estándar IEEE802.11, a pesar que el sistema está basado en el estándar IEEE 802.15.4 se obtiene conceptos claves del funcionamiento de la capa MAC del estándar IEEE 802.11 para ser aplicados en el diseño del protocolo.

Sobre la base del análisis teórico de los estándares, se desarrolla la propuesta de solución para el protocolo de capa MAC del sistema Mobile Smart Node en el uso eficiente de energía en ambientes domésticos. Se describe cada parte del algoritmo del protocolo planteado usando diagramas de flujo y se puntualiza el proceso de implementación de la propuesta, es decir cómo se realiza la programación.

3.2 ACCESO MÚLTIPLE POR DETECCIÓN DE PORTADORA CON ANULACIÓN DE COLISIONES (CSMA-CA, CARRIER SENSE, MULTIPLE ACCESS, COLLISION AVOIDANCE) ^[27] ^[16]

Una de las funciones de la capa MAC es el acceso al medio libre de colisiones entre las estaciones que pugnan por transmitir, ya sea en un medio guiado o no

guiado. Para esto existen diferentes protocolos cuyos algoritmos están diseñados para evitar que se produzcan colisiones.

En redes inalámbricas por sus características, no se puede transmitir y escuchar a la vez, esto constituye un factor para que se ocasionen colisiones entre los paquetes de datos y para que ciertos protocolos de acceso al medio no sean factibles. En la mayoría de redes inalámbricas se utiliza el protocolo CSMA-CA, en donde cada estación anuncia opcionalmente su intención de transmitir antes de hacerlo reduciendo la probabilidad de colisiones en el canal.

El estándar IEEE 802.15.4 LR-WPAN utiliza dos tipos de mecanismos de acceso al canal, dependiendo de la configuración de red. Las PAN's con Non-beacon habilitado utilizan un mecanismo de acceso CSMA-CA no ranurado. Cada vez que un dispositivo desea transmitir tramas de datos o de comandos MAC, espera un período de tiempo aleatorio. Si el canal se encuentra libre, tras el periodo de tiempo, el dispositivo transmite sus datos. Si el canal se encuentra ocupado, el dispositivo espera un nuevo período de tiempo aleatorio antes de intentar acceder al canal nuevamente, esto con la finalidad de evitar colisiones. Cuando el dispositivo receptor recibe la trama, envía una señal de reconocimiento (ACK) para asegurar que la trama ha sido transmitida y recibida correctamente. Las tramas ACK se envían sin utilizar un mecanismo CSMA-CA.

Las PAN's con Beacon habilitado utilizan un mecanismo de acceso al canal CSMA CA ranurado, donde las ranuras de retroceso están alineadas con el inicio de la transmisión de la trama Beacon. Las ranuras de retroceso de todos los dispositivos dentro de la red PAN se alinean con el coordinador PAN. Cada vez que un dispositivo desea transmitir tramas de datos, se localiza el límite de la ranura de retroceso próxima y se espera un número aleatorio de ranuras de retroceso. Si el canal está ocupado, después de este retroceso aleatorio, el dispositivo espera otro número aleatorio de ranuras de retroceso antes de tratar de acceder al canal nuevamente. Si el canal está libre, el dispositivo comienza a transmitir en la siguiente ranura de retroceso. Las tramas Beacon y ACK se envían sin utilizar un mecanismo CSMA-CA.

La red PAN del sistema MSN utiliza una configuración de red con Nonbeacon habilitado por lo cual se debería utilizar un mecanismo CSMA-CA no ranurado. Sin embargo, este mecanismo no ha sido implementado aun dentro de la capa MAC de la pila del sistema MSN. A pesar de que el mecanismo fuera implementado en el sistema, CSMA-CA no anularía totalmente las colisiones ya que se presentan dos problemas particulares que se analizan a continuación: los nodos ocultos y los nodos expuestos.

3.2.1 PROBLEMAS DE CSMA-CA ^[27]

Para analizar estos dos problemas se considera la Figura 3.1, en la que se ilustran cuatro nodos inalámbricos: A, B, C y D. El alcance de radio es tal que A y B están en el mismo alcance y potencialmente pueden interferirse. C también puede interferir tanto a B como D, pero no al nodo A.

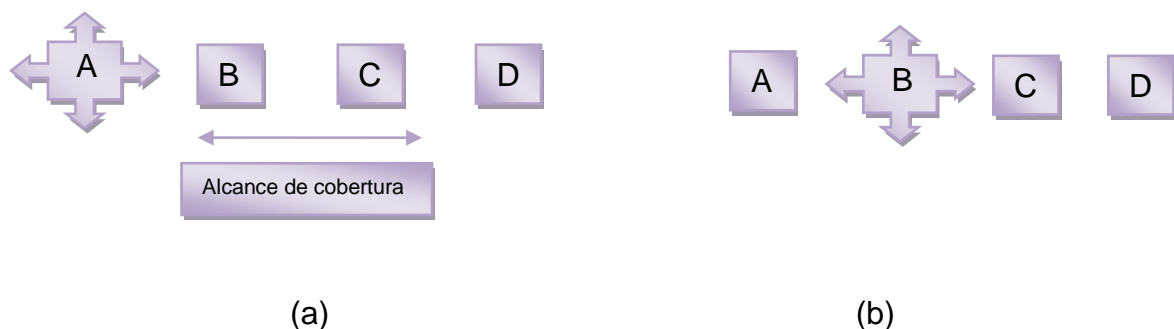


Figura 3.1: (a) Nodo A transmitiendo. (b) Nodo B transmitiendo.

Fuente: Tanenbaum, A. (2003). Redes de Computadoras. Amsterdam.
Rediseñado por: Susana Mafla.

Cuando A está transmitiendo a B, como se muestra en la figura 3.1 (a). Si C detecta el medio, no podrá escuchar a A porque está fuera de su alcance de cobertura, y por lo tanto deducirá falsamente que puede transmitir. Si C comienza a transmitir, interferirá en B, eliminando la comunicación con A. El problema cuando un nodo cree que el canal está libre, pero en realidad está ocupado por

²⁷ Tanenbaum, Andrew (2003). Redes de Computadoras. Amsterdam

otro nodo al que no oye por no estar en su alcance de cobertura, se denomina *Nodo oculto*.

Ahora considerando una situación inversa: B transmitiendo hacia A, como se muestra en la figura 3.1 (b). Si C detecta el medio, escuchará una transmisión y concluirá equivocadamente que no puede enviar datos a D. Cuando un nodo cree que el canal está ocupado, pero en realidad está libre pues el nodo al que oye no interferiría en su transmisión, se denomina *Nodo expuesto*.

El origen de estos dos problemas radica en que la estación inalámbrica transmisora pretende saber si existe o no actividad alrededor de la estación inalámbrica receptora, pero CSMA sencillamente indica si hay o no actividad alrededor de la estación que detecta la portadora. El protocolo que surgió para solucionar este problema se conoce como Acceso Múltiple con Prevención de Colisiones MACA.

3.3 ACCESO MÚLTIPLE CON PREVENCIÓN DE COLISIONES (MACA, MULTIPLE ACCES WITH COLLISION AVOIDANCE) ^[27]

MACA es un protocolo que se basa en estimular al receptor a enviar una trama corta, de manera que las estaciones cercanas puedan detectar esta transmisión y eviten ellas mismas transmitir durante la siguiente trama de datos para evitar posibles colisiones.

Para la explicación del funcionamiento del protocolo MACA se considera la figura 3.2 (a). La estación A antes de transmitir, envía una trama de solicitud de envío (RTS, Request To Send) a B. Dicha trama tiene una longitud de 30 bytes y su función es indicar la longitud del paquete de datos que se va a enviar. Luego como se observa en la figura 3.2 (b), la estación B contesta con una trama libre para envío (CTS, Clear To Send). La trama CTS contiene la longitud del paquete de datos copiada de la trama RTS. Al recibir A esta trama de CTS, empieza con la transmisión.

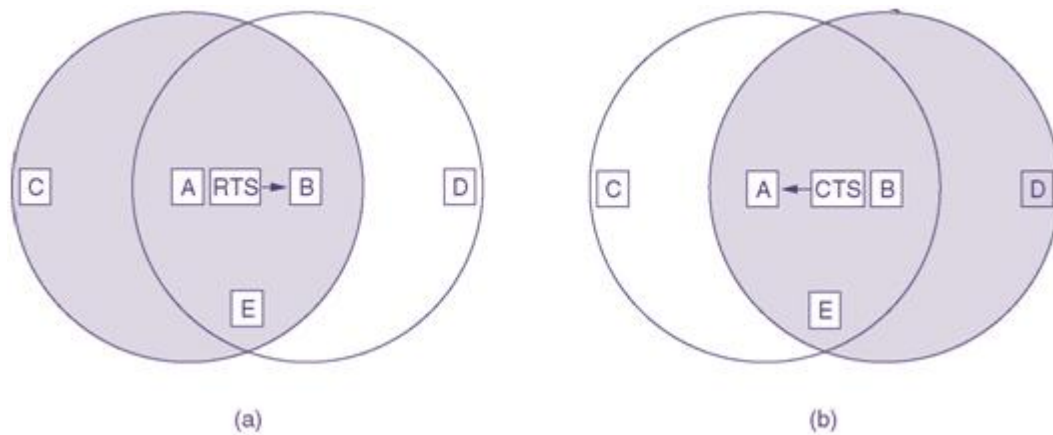


Figura 3.2: Protocolo MACA: (a) A enviando a B un RTS. (b) B respondiendo a A un CTS.

Fuente: Tanenbaum, A. (2003). Redes de Computadoras. Amsterdam.
Rediseñado por: Susana Mafla

Ante esto, el resto de estaciones actuarán de tal forma que, si “escuchan” un RTS evidentemente están cerca de A y deben permanecer en silencio durante el periodo de tiempo necesario para que el CTS se transmita de regreso a A. Si “escuchan” un CTS, evidentemente están cerca de B y deben permanecer en silencio y esperar el tiempo necesario para que se transmita la longitud del paquete de datos indicada en el CTS.

C escucha el RTS de A pero no el CTS de B. Mientras no interfiera con el CTS, puede transmitir. D, escucha el CTS de B, eso le indica que debe esperar hasta el momento en que termine la transmisión de la trama del paquete de datos. E, permanece en silencio todo el tiempo porque escucha tanto al RTS y al CTS. Sin embargo, aun pueden ocurrir colisiones; en caso de que dos estaciones envíen tramas RTS a la misma estación y al mismo tiempo, las tramas colisionarán y se perderán.

3.4 ACCESO MÚLTIPLE CON PREVENCIÓN DE COLISIONES PARA WIRELESS (MACAW, MULTIPLE ACCES WITH COLLISION AVOIDANCE FOR WIRELESS) ^[27]

En 1994 Bharghavan y otros autores mejoraron el desempeño del protocolo MACA para redes wireless, notaron que, sin acuse de recibo (ACK) de la capa de enlace de datos, las tramas no eran retransmitidas sino hasta que la capa de transporte notaba su ausencia, mucho después. Esto se resolvió introduciendo una trama ACK tras cada trama de datos recibida exitosamente. También observaron que CSMA puede utilizarse para evitar que dos estaciones cercanas transmitan un RTS hacia un mismo destino al mismo tiempo, por lo que se agregó la detección de portadora. Además, decidieron ejecutar el algoritmo de retroceso por separado para cada corriente de datos (par origen-destino), en lugar de para cada estación. El algoritmo de retroceso sirve para que al ocurrir una colisión la estación espere un tiempo aleatorio y reintente la transmisión. En el protocolo MACAW adicionalmente se agregaron mecanismos para que las estaciones intercambien información sobre congestiónamiento.

3.5 FUNCIONES DE LA CAPA MAC DEL ESTÁNDAR IEEE 802.11 ^[28]

La capa MAC del estándar IEEE 802.11 se encarga de proporcionar un servicio de datos fiable a los protocolos de capas superiores y permitir un acceso equitativo al medio inalámbrico. Aunque este estándar se define para mayores tasas y velocidades de transmisión, y por ende para diferentes aplicaciones que el estándar IEEE 802.15.4, es necesario analizar el funcionamiento de la capa MAC del estándar IEEE 802.11 para obtener conceptos que se pueden utilizar en el diseño del protocolo de capa MAC propuesto para el sistema MSN.

El estándar IEEE 802.11 define dos modos de funcionamiento de la capa MAC: Función de Coordinación Puntual PCF (Point Coordination Function) y Función de Coordinación Distribuida (DCF, Distributed Coordination Function).

²⁸ Pozo, Nelson (2009). Estudio y diseño de una red lan inalámbrica con QoS, para voz y datos en el CIGMYP, empleando los estándares IEEE802.11 g/e:
<http://bibdigital.epn.edu.ec/bitstream/15000/1316/1/CD-2019.pdf> (marzo 2001)

3.5.1 Función de Coordinación Puntual (PCF, Point Coordination Function)

Bajo esta función se proporciona un periodo libre de contención, durante el cual una estación base controla el uso del medio para eliminar la contienda entre las estaciones por el acceso al canal y evitar colisiones. La estación base sondea a las demás estaciones, preguntándoles si tienen algo para enviar.

3.5.2 Función de Coordinación Distribuida (DCF, Distributed Coordination Function)

Es el método de acceso básico para transmitir asíncronamente con el mejor esfuerzo. En éste método no se usa ningún tipo de estación base, el medio de transmisión opera en modo de contención, sin asignar ninguna prioridad a las estaciones que contienden por el canal de transmisión.

DCF está basado en el protocolo CSMA-CA descrito anteriormente. La detección de portadora involucra controlar que el canal esté libre antes de transmitir para evitar colisiones. Sin embargo, cuando el canal se libera inmediatamente las estaciones intentan transmitir, existiendo la probabilidad de que las estaciones intenten tomar el canal simultáneamente y haya colisiones. Para evitar este problema se aleatoriza los tiempos en los cuales las estaciones contienden por el canal.

3.5.3 El espacio inter-trama IFS (Interframe Space)

Es el tiempo de espera en el que todas las estaciones deben aguardar después de que una transmisión ha sido completada. La duración de este tiempo depende del tipo de trama que la estación está por transmitir.

El espacio inter-trama permite dar prioridad a los paquetes de información, clasificándolos en cuatro espacios de tiempo diferentes: SIFS, PIFS, DIFS, EIFS. Estos periodos de tiempo varían en su duración de acuerdo a su prioridad como se observa en la figura 3.3.

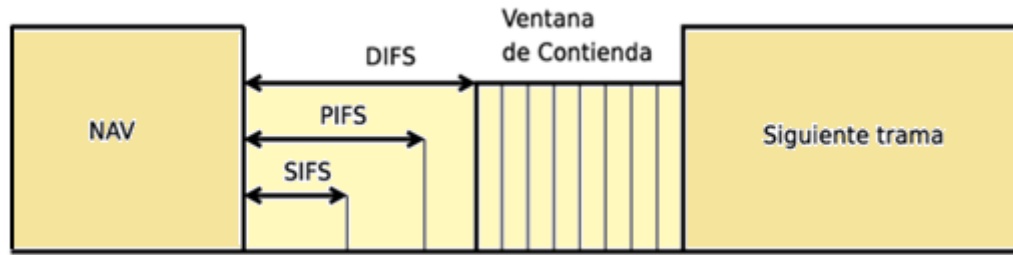


Figura 3.3: Duración de los IFS de acuerdo a su prioridad

Fuente: Universidad Rey Juan Carlos, Departamento de Sistemas Telemáticos y Computación (2010). De: <http://gsyc.es/~mortuno/rii/02-802.11.pdf> (marzo de 2011)

- *Espacio inter-trama corto (SIFS, Short Interframe Spacing).*- Se utiliza para transmitir tramas de alta prioridad, como ACK y CTS entre las estaciones sin la intervención de la estación base.
- *Espacio inter-trama PCF (PIFS, PCF Interframe Spacing).*- Es el tiempo de espera que se utiliza por la estación base para tomar el control del medio, cuando se trabaja bajo la Función de Coordinación Puntual PCF.
- *Espacio inter-trama DCF (DIFS, DCF Interframe Spacing).*- Es el periodo de espera para que las estaciones que trabajan bajo la función de Coordinación Distribuida DCF puedan hacer uso del medio, se usa para tramas que no tienen alta prioridad por ejemplo RTS.
- *Espacio inter-trama extendido (EIFS, Extended Interframe Spacing).*- Se utiliza exclusivamente cuando se producen errores en la transmisión de las tramas, para reportar los errores a la capa MAC.

Para eliminar el problema de nodos escondidos y nodos expuestos, la capa MAC del estándar IEEE 802.11 utiliza el protocolo RTS/CTS en conjunto con los espacios inter- tramas. Adicionalmente se informa al resto de nodos que el medio está siendo utilizado por un lapso establecido de tiempo, esto se realiza emitiendo un paquete conocido como Vector de Asignación de Red NAV (Network Allocation Vector). En la figura 3.4 se observa el uso completo del protocolo RTS/CTS, IFS y NAV.

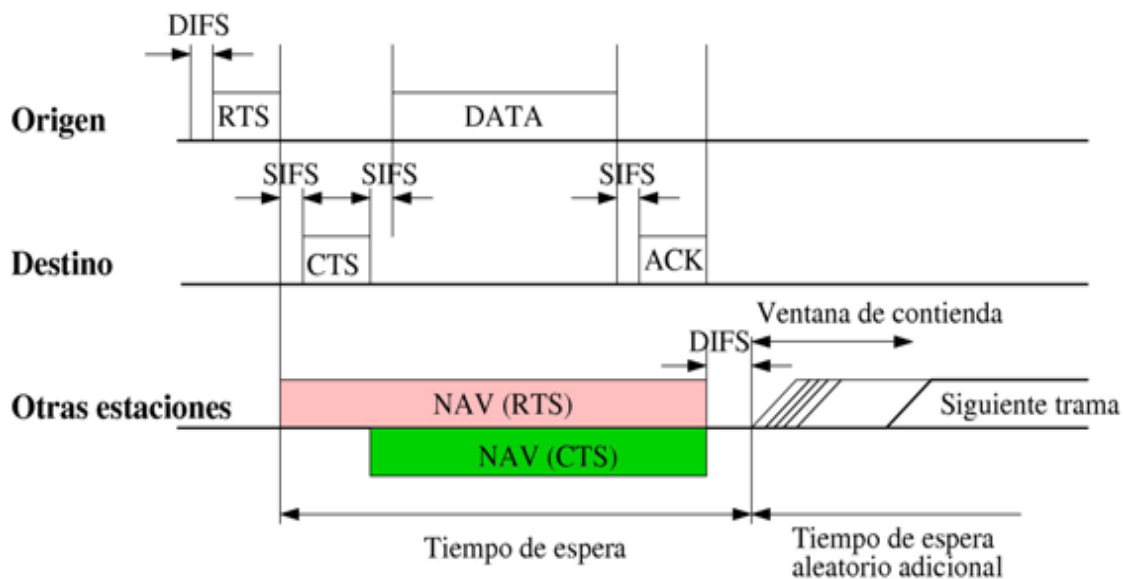


Figura 3.4: Uso del protocolo RTS/CTS, IFS y NAV en el estándar IEEE 802.11

Fuente: Universidad Rey Juan Carlos, Departamento de Sistemas Telemáticos y Computación (2010). De: <http://gsync.es/~mortuno/rii/02-802.11.pdf> (marzo de 2011)

3.5.4 Vector de Asignación de Red NAV (Network Allocation Vector) ^[29]

NAV es un contador que indica el tiempo en el cual el medio está ocupado. Cuando el NAV es diferente de 0, el medio está ocupado; cuando es 0, el medio está libre. Utilizando el NAV, las estaciones se aseguran de que las transmisiones no serán interrumpidas ya que el resto de estaciones no accederán al medio hasta que la transmisión se complete.

3.5.5 Mecanismo de retracción o Backoff ^[30]

El mecanismo de retracción o Backoff se utiliza para evitar que se produzcan colisiones cuando dos estaciones se encuentran esperando que el medio esté libre y transmitan simultáneamente.

²⁹ Suárez, Belén Revert (2009) Implementación del Protocolo WDS para Redes 802.11. <http://upcommons.upc.edu/pfc/bitstream/2099.1/3554/2/41794-2.pdf> (marzo de 2011)

³⁰ Pérez, Israel Oropeza (2006) Análisis de Rendimiento de IEEE 802.11 b <http://148.206.53.231/UAMI13093.pdf> (marzo 2011)

En el procedimiento de backoff se censa el canal para ser ocupado. La estación espera hasta que el canal se libere y determina un tiempo de retracción aleatorio. Los tiempos de los periodos de tiempo son fijados con relación a los Time Slots que son usados también para definir los intervalos IFS. Las estaciones decrementan sus contadores de retracción hasta que el canal sea ocupado o hasta que los temporizadores expiren. Si los contadores no han expirado, las estaciones detienen sus contadores. Cuando el temporizador finalmente ha expirado, la estación envía su trama. En un principio cada estación fija su temporizador en rango de 0 a 7 Time Slots, si hay colisión las estaciones nuevamente reinician sus contadores en un rango de 0 a 15. El periodo libre después de un DIFS, es conocido como la ventana de contención.

3.6 CONTEXTUALIZACIÓN DEL PROBLEMA

El MSN es un sistema embebido o integrado, esto significa que está diseñado para realizar un conjunto específico de tareas, es único en su diseño y se encuentra formado por varios circuitos integrados. Por este motivo los circuitos que conforman al MSN se utilizan para realizar funciones específicas que en conjunto logran realizar las tareas del sistema.

Es así que el transceiver RF de ZigBee, el CC2420 de Texas Instruments basado en el estándar IEEE 802.15.4, que se utiliza dentro del sistema embebido MSN, únicamente realiza las funciones de antena para la recepción y transmisión de datos. Todas las funciones que se encuentran programadas en el transceiver no se utilizan y las especificaciones del estándar IEEE 802.15.4 se definen en la programación del microcontrolador ARM LPC 2148 de NXP, que es el cerebro del sistema embebido.

El estándar IEEE 802.15.4 establece como mecanismo de acceso al canal el protocolo CSMA-CA, pero éste no ha sido implementado en el sistema, ocasionando que se produzcan colisiones entre los dispositivos inalámbricos. Además CSMA-CA presenta dos problemas específicos: los nodos escondidos y los nodos expuestos que provocan la ocurrencia de colisiones. Frente a estos dos problemas existen protocolos desarrollados para su solución como es el protocolo

MACA y el protocolo MACAW, ambos protocolos utilizan el intercambio de tramas RTS y CTS para alertar a toda la red inalámbrica antes de iniciar una transmisión de datos y reducir la posibilidad de colisiones. Sin embargo, existe otra probabilidad de colisión producida por transmisiones simultáneas de tramas RTS que son emitidas por los dispositivos que se encuentran esperando por acceder al medio.

3.7 ANÁLISIS DE SOLUCIÓN PARA EL DESARROLLO DEL PROTOCOLO DE CAPA MAC

La solución que se propone en el presente trabajo ante el problema de colisión entre los dispositivos del sistema MSN, se basa en los conceptos de los protocolos y estándares previamente estudiados, utilizando lógicamente las partes que se aplican de acuerdo a las necesidades del sistema MSN.

El protocolo MACAW puede considerarse como un modelo de referencia en el desarrollo del nuevo protocolo a implementar en el sistema MSN. MACAW logra solucionar los problemas de CSMA, eliminando los nodos escondidos y los nodos expuestos, mejora el desempeño de MACA; modificando sus funciones acorde al sistema MSN y eliminando funciones redundantes del protocolo se puede conseguir un protocolo específico y eficiente que elimine las colisiones en el sistema MSN.

Además en referencia a la función de coordinación distribuida de la capa MAC del estándar IEEE 802.11 se plantea emplear un vector de asignación de red NAV como temporizador para que informe a las demás estaciones que no participan en la comunicación que el canal está ocupado por un determinado tiempo, ya que en el estándar IEEE 802.15.4 no se aplica este concepto.

Para eliminar las transmisiones simultáneas se propone utilizar el concepto de los espacios inter-tramas definidos en las funciones del estándar IEEE 802.11 conjuntamente combinados con tiempos aleatorios. A pesar que el estándar IEEE 802.15.4 sí establece tiempos inter-tramas, éstos únicamente se utilizan por la capa MAC para tener tiempo al procesar los datos recibidos de la capa física y no

para dar prioridad a los diferentes tipos de tramas. Como el estándar IEEE 802.15.4 trabaja con el mecanismo CSMA-CA no se utiliza el intercambio de tramas RTS y CTS, las cuales deben tener diferentes prioridades definidas por los espacios inter-tramas. Por el contrario en el estándar IEEE 802.11 sí se utiliza el intercambio de tramas RTS y CTS con sus respectivas prioridades, por esta razón se hace referencia a ciertos conceptos del estándar IEEE 802.11 para el desarrollo del protocolo de capa MAC propuesto.

La propuesta del protocolo de capa MAC para solucionar el problema de colisión entre dispositivos del sistema MSN, se basa en un algoritmo destinado a esperar un tiempo aleatorio, en base a un espacio inter-trama, antes de enviar una trama RTS cuando una estación desee transmitir; ésta trama RTS contiene el tamaño del paquete de datos que se quiere enviar.

La estación receptora al recibir la trama RTS contesta con una trama CTS, la trama CTS contiene el tamaño del paquete de datos copiado de la trama RTS y se difunde en toda el área de alcance de la estación receptora, informando así a las estaciones cercanas que está presta a recibir datos y que este proceso demorará determinado tiempo. El tiempo se almacena en un temporizador NAV en cada estación y mientras el temporizador no sea igual a cero las estaciones no podrán transmitir evitando así que se produzcan colisiones.

Luego de recibir la trama CTS, la estación transmisora comienza con la transmisión, cuando la estación receptora recibe todos los datos, ésta envía una trama ACK para confirmar el éxito de la transmisión. La estación transmisora a su vez espera determinado tiempo para recibir la trama ACK, en caso de no recibir la trama reanuda el proceso, esto con la finalidad de corregir errores si la trama ACK se pierde.

Un resumen de las tramas intercambiadas entre las estaciones transmisoras MSN y la estación receptora FESN se muestra en la figura 3.5.

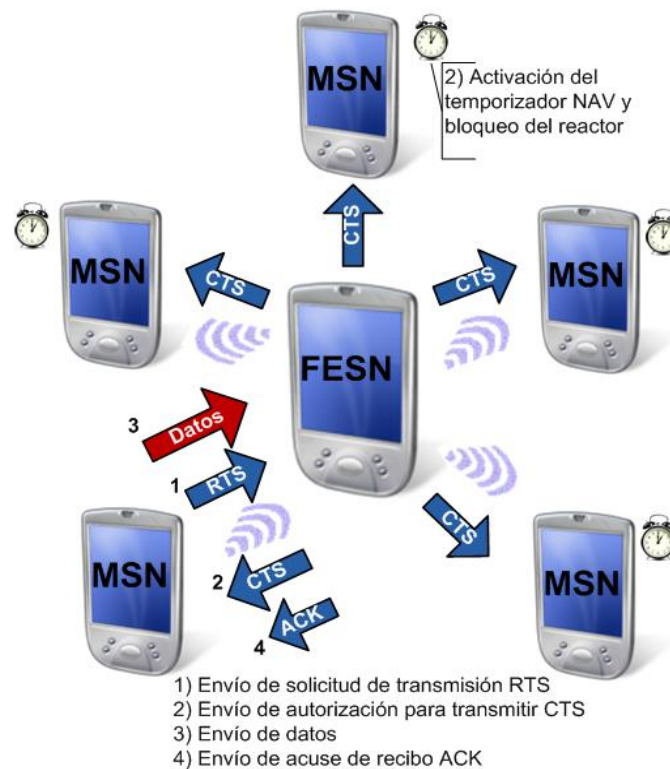


Figura 3.5: Tramas intercambiadas entre MSN y FESN.

Diseñado por: Susana Mafla

Si, a pesar de estas precauciones ocurre una colisión, se aplica el algoritmo de retroceso exponencial para que después de un determinado tiempo se reintente o se aborte la transmisión. No se utiliza el protocolo para intercambiar datos sobre congestión porque significa procesar más información, consumir más energía y demorar la comunicación entre los dispositivos inalámbricos. Detalladamente el algoritmo planteado se esquematiza en los siguientes diagramas de flujo que representan las tres subrutinas en las cuales está dividido el algoritmo.

3.7.1 SUBROUTINA 1 DEL DISPOSITIVO TRANSMISOR MSN

En la figura 3.6 se encuentra la primera subrutina del algoritmo propuesto para el protocolo de capa MAC. Esta primera subrutina tiene el objetivo de analizar las condiciones iniciales y preparar al dispositivo transmisor para que se inicie el proceso de transmisión.

La subrutina inicia con una interrogante: ¿desea transmitir?, si la respuesta es no, se mantiene inactivo y permanece en un bucle, si es sí, se pregunta: ¿detecta actividad cercana?, si la respuesta es no, espera un tiempo aleatorio definido por un espacio inter-trama DIFS, antes de comenzar con la transmisión; el tiempo aleatorio sirve para evitar que otros dispositivos comiencen a transmitir al mismo tiempo hacia la misma estación.

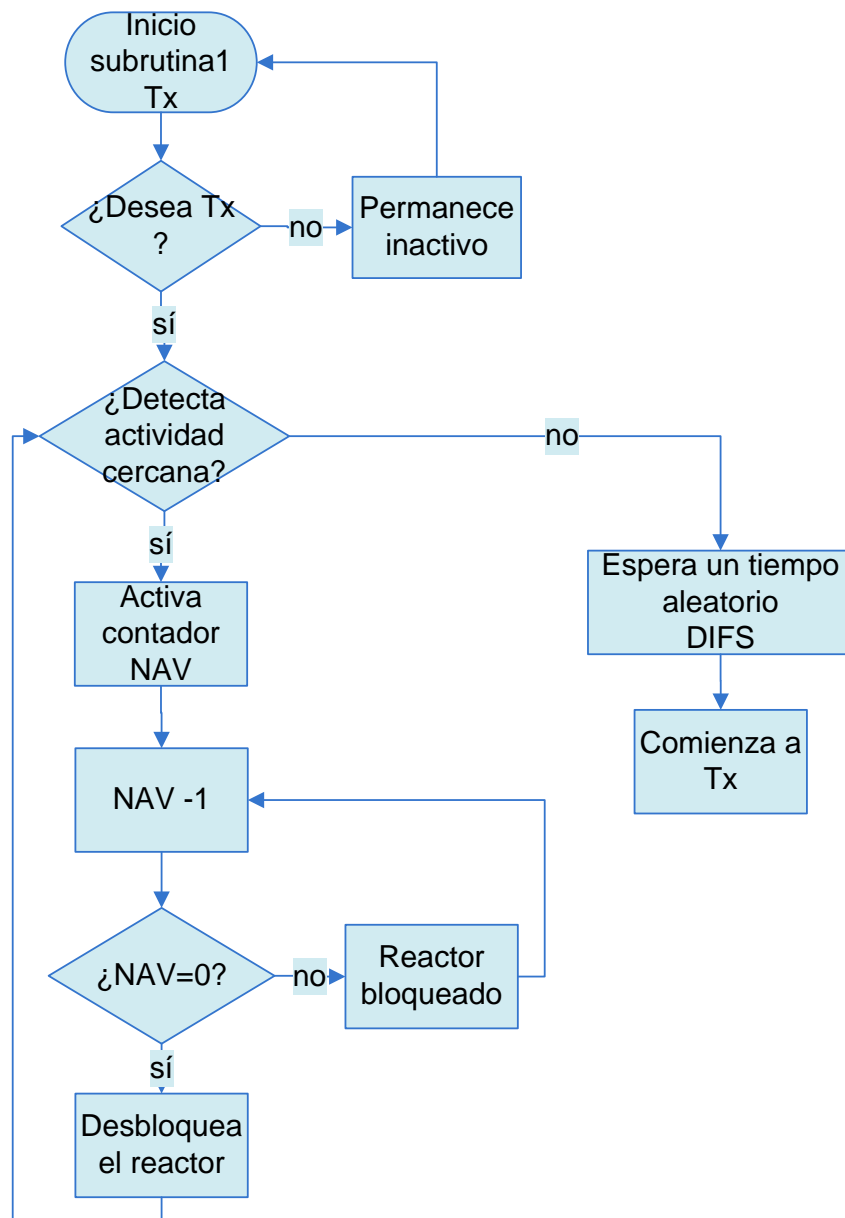


Figura 3.6: Subrutina 1 del algoritmo para el transmisor MSN.

Diseñado por: Susana Mafla

Si ha detectado actividad cercana se activa un contador NAV que se decremента una unidad y luego se comprueba si el contador es igual a 0, si no lo es se bloquea el reactor y se mantiene en un ciclo hasta que el contador NAV sea 0, se desbloquea el reactor y se vuelve a sondear si el canal está libre. El análisis completo de los cálculos realizados para determinar los valores del contador NAV se encuentra en el subtema 3.7.4

3.7.2 SUBROUTINA 2 DEL DISPOSITIVO TRANSMISOR MSN

La segunda subrutina para el dispositivo transmisor se visualiza en la figura 3.7.

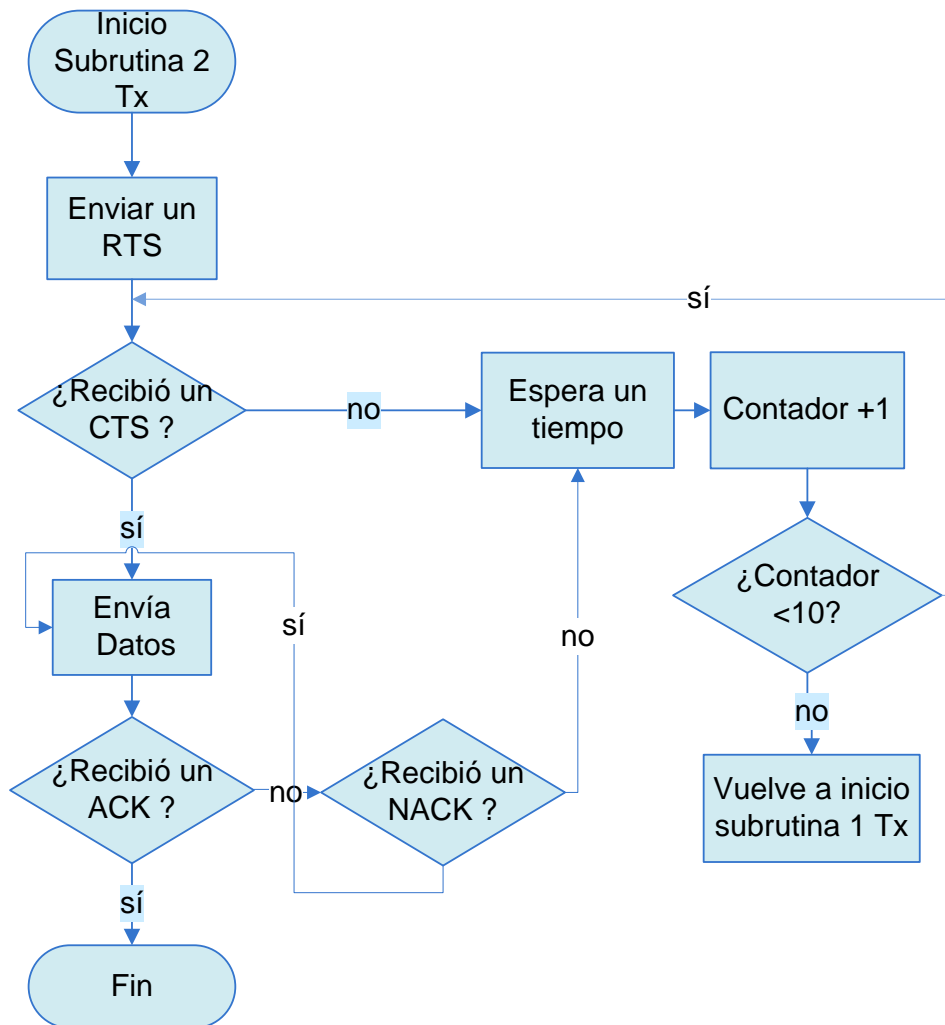


Figura 3.7: Subrutina 2 del algoritmo para el transmisor MSN.

Diseñado por: Susana Mafla

Ésta subrutina comienza transmitiendo una trama RTS hacia la estación receptora. Enviada la trama, espera el mensaje de confirmación, es decir una trama CTS, haciéndose la interrogante: ¿Recibió un CTS?, si la respuesta es no, espera un tiempo, incrementa un contador en una unidad y luego compara si el contador es menor a 10, si lo es vuelve a interrogarse si ha recibido un CTS, caso contrario se encera el contador y se vuelve al inicio de la subrutina 1 del transmisor. Todo esto con la finalidad de que exista un límite de intentos y el dispositivo transmisor no se quede permanentemente en un bucle que transmita RTS.

Si recibió una trama CTS, la estación transmisora envía los datos. Una vez enviados los datos, la estación transmisora espera un acuse de recibo o ACK, de igual manera haciendo una interrogante: ¿Recibió un ACK?, si la respuesta es no, existen dos opciones. En la primera se confirma si se recibió un NACK, esto significa que los datos llegaron incompletos al receptor y se vuelve a enviar únicamente los datos. En la segunda opción si no se recibió un NACK, entonces definitivamente el ACK se perdió o los datos no llegaron a su destino, entonces se espera un tiempo aleatorio, se utiliza nuevamente el contador con el proceso descrito anteriormente y se vuelve a reenviar los datos pero desde el inicio de la subrutina del dispositivo transmisor, para evitar la duplicación de datos el receptor es el responsable de analizar la información y desechar los datos duplicados.

Si la estación transmisora recibe un ACK, significa que los datos se recibieron exitosamente y se finaliza el proceso.

3.7.3 SUBRUTINA DEL DISPOSITIVO RECEPTOR FESN

Las dos subrutinas anteriores se desarrollan para el dispositivo transmisor, ahora en el lado del dispositivo receptor se ejecuta la subrutina representada por el diagrama de flujo que se observa en la figura 3.8.

El dispositivo receptor puede detectar dos tipos de tramas; tramas RTS o tramas CTS. La subrutina del dispositivo receptor inicia comparando cuál de estos dos tipos de tramas ha recibido o sencillamente no ha recibido ninguna trama. Para

esto se hace la interrogante: ¿recibió un RTS?, si la respuesta es no, nuevamente se pregunta ¿recibió un CTS?; si la respuesta a esta última interrogante es no, vuelve al inicio y se mantiene en un bucle a la espera de recibir alguna trama.

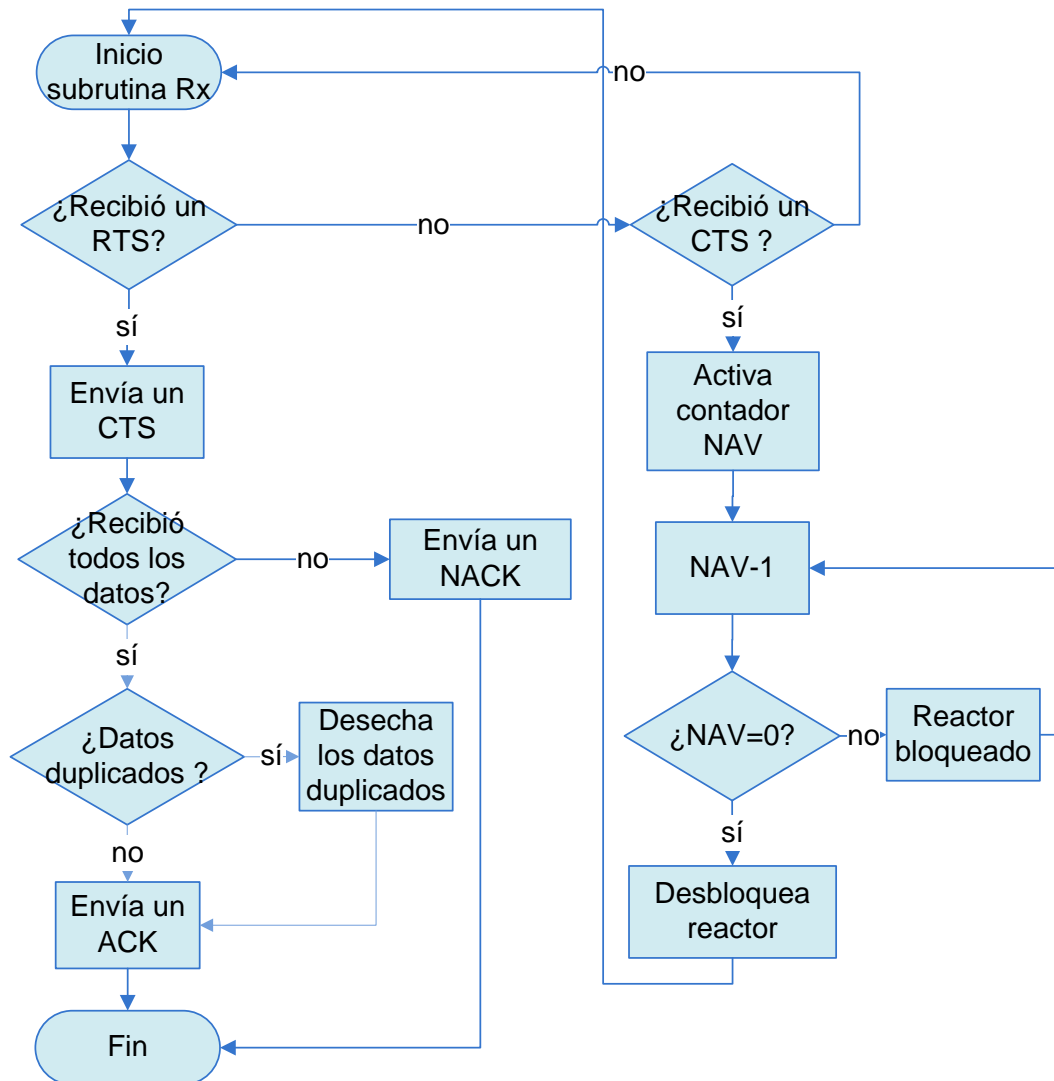


Figura 3.8: Subrutina del algoritmo para el receptor FESN.

Diseñado por: Susana Mafla

Si ha recibido un CTS, sin haber enviado un RTS significa que el dispositivo receptor es un MSN, que está en el área de cobertura de un FESN que a su vez está presto a recibir información y por lo tanto el MSN debe esperar el tiempo establecido en la trama CTS y no interferir en la comunicación en curso, para esto

se activa un contador NAV que se va decrementando hasta ser igual a cero y luego se desbloquea el sistema operativo o reactor del dispositivo MSN.

En el caso de que el dispositivo ha recibido una trama RTS significa que dicho dispositivo receptor es un FESN y envía un CTS, luego el receptor, espera por los datos que serán enviados, cuando estos hayan llegado, se hace una nueva interrogante: ¿Recibió todos los datos?. En la trama RTS se contempla el número de datos que serán enviados y si este número no coincide con el que se han recibido, se envía un NACK al dispositivo transmisor. En el caso afirmativo de que los datos han llegado completos se realiza otra interrogante ¿Están duplicados los datos?, con finalidad de evitar la duplicación de información, si es el caso se desecha estos datos y se envía un ACK al transmisor, si los datos no están duplicados se envía un ACK y se finaliza la subrutina del receptor.

Dado el caso en el que los datos no han llegado completos o con errores al receptor, se podría aplicar algún método de detección de errores o algún código de redundancia cíclica para detectar exactamente en dónde se ha producido el error y devolver una respuesta al transmisor, para que éste solucione el problema y envíe nuevamente los datos sin tener que volver a realizar todos los pasos anteriores, lógicamente esto se puede desarrollar a futuro para mejorar la transmisión, analizando su factibilidad ya que no son muchos los datos que se transmiten y mientras más robusto es el software del proyecto VISE se disminuye la velocidad de transmisión, se incluyen retrasos en la comunicación y se aumenta el consumo de energía por procesamiento de información.

3.7.4 Cálculo del tiempo de una transmisión de datos para la implementación del contador NAV

Analizando las características del enlace inalámbrico de comunicación entre los dispositivos del sistema MSN, a continuación se explica el procedimiento utilizado para el cálculo de la duración de una transmisión de datos. Ésta información es necesaria para la generación del contador NAV, el cual es responsable de mantener bloqueados a los dispositivos MSN que no intervienen en una

comunicación durante el tiempo que ésta transcurre para evitar colisiones dentro del algoritmo que se propone.

Para calcular el tiempo de una transmisión se considera dos tiempos, el tiempo de transmisión que es el tiempo que se demora el transmisor en transmitir los datos al medio de comunicación, que equivale a transmitir un MPDU (Unidad de Datos de Protocolo MAC), y el tiempo de propagación que es el tiempo en el cual los datos se propagan por el medio de comunicación hasta llegar al receptor. Además de esta consideración se debe tomar en cuenta que el transmisor envía los datos y cuando estos son recibidos por el receptor, éste último envía un mensaje de acuse de recibo o ACK como se observa en la figura 3.9.

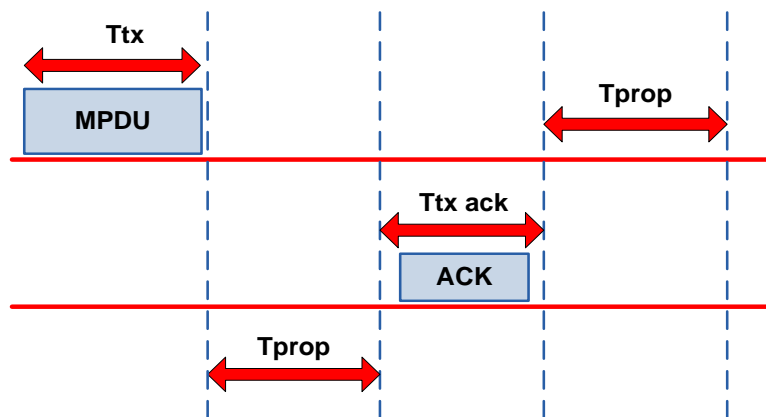


Figura 3.9: Duración de una transmisión de datos

Diseñado por: Susana Mafla

Acorde a las consideraciones anteriores el tiempo de una transmisión de datos entre dos dispositivos del sistema MSN es igual a la suma de los tiempos de transmisión más los tiempos de propagación.

$$\textit{Tiempo de transmisión de datos} = Ttx + Tpro + Ttx\ ack + Tprop$$

Donde el tiempo de transmisión Ttx está dado por la fórmula:

$$Ttx = \frac{\textit{tamaño de la trama (bits)}}{\textit{Ancho de banda (bits/s)}}$$

El ancho de banda se define por la fórmula:

$$AB = \frac{Vtx}{2}$$

Donde AB es el ancho de banda y Vtx es la velocidad de transmisión.

Continuando el tiempo de propagación de una onda electromagnética está dado por la siguiente fórmula:

$$T_{pro} = \frac{\lambda}{C}$$

Donde λ es la longitud de onda de la señal electromagnética, se expresa en metros y representa la longitud de separación entre el transmisor y el receptor. C es la velocidad de propagación de la onda electromagnética y representa la constante de la velocidad de propagación de las ondas electromagnéticas en el aire y su valor es de 2.997×10^8 m/s.³¹

3.7.4.1 Dimensionamiento de los parámetros.

Dentro de la implementación misma del contador NAV, se toma en cuenta las condiciones máximas que soporta el sistema MSN. Como éste sistema describe una red inalámbrica, para cuestiones de dimensionamiento se toma la distancia máxima que pueden estar separados dos dispositivos del sistema MSN que es de 10 m, además los dispositivos transmiten a una velocidad de transmisión de 250 kbits/s y el máximo tamaño de una trama o MPDU en el sistema MSN es de 630 Bytes. Basándose en este dimensionamiento a continuación en la tabla 3.1 se resume los valores de los parámetros utilizados para calcular el tiempo de transmisión de los datos e implementar el contador NAV.

Parámetro	Símbolo	Valor
Tamaño máximo de trama	MPDU	630 (Bytes)

³¹ Universidad de los Andes (2010). Velocidad de la Luz. <http://www.ciencias.ula.ve/index.htm> (junio de 2011)

Tamaño de la trama ACK	ACK	5 (Bytes)
Tamaños de tramas RTS y CTS	RTS/CTS	57 (Bytes)
Distancia máxima de separación entre dispositivos MSN	λ	10 (m)
Velocidad de Transmisión de los dispositivos MSN	Vtx	250 (Kbits/s)
Velocidad de propagación de la luz en el aire	C	2.997×10^8 (m/s)

Tabla 3.1: Valores de los diferentes parámetros para el cálculo del contador NAV

Diseñado por: Susana Mafla

3.7.4.2 Cálculos

Primeramente el dato del tamaño de trama para el cálculo del tiempo de transmisión en el programa se obtiene del campo length de la trama y se transforma a bits. Sin embargo, para este cálculo se toma el tamaño máximo de una trama en el sistema MSN que es 630 bytes.

Utilizando las fórmulas ya descritas se calcula el ancho de banda en base a la velocidad de transmisión de los dispositivos de 250 (kbits/s).

$$AB = \frac{Vtx}{2}$$

$$AB = \frac{250 \text{ (kbits/s)}}{2}$$

$$AB = 125 \text{ (kbits/s)}$$

$$Ttx \text{ MPDU} = \frac{\text{tamaño de la trama (bits)}}{\text{Ancho de banda (bits/s)}}$$

$$Ttx \text{ MPDU} = \frac{630 \times 8 \text{ (bits)}}{125\,000 \text{ (bits/s)}}$$

$$Ttx \text{ MPDU} = 0.04032 \text{ s}$$

$$T_{tx MPDU} = 40.32 \text{ ms}$$

Para calcular el tiempo de transmisión de una trama de acuse de recibo ACK el tamaño de esta trama es de 5 bytes.

$$T_{tx ACK} = \frac{\text{tamaño de la trama (bits)}}{\text{Ancho de banda (bits/s)}}$$

$$T_{tx ACK} = \frac{5 \times 8 \text{ (bits)}}{125\,000 \text{ (bits/s)}}$$

$$T_{tx ACK} = 0.32 \text{ ms}$$

Ahora para calcular el tiempo de propagación se considera la máxima longitud de separación entre el transmisor y el receptor λ que es 10 m y la velocidad de propagación de una onda electromagnética en el aire de $2,997 \times 10^8$ m/s.

$$T_{pro} = \frac{\lambda}{c}$$

$$T_{pro} = \frac{10 \text{ (m)}}{2,997 \times 10^8 \text{ (m/s)}}$$

$$T_{pro} = 3.3366 \times 10^8 \text{ (s)}$$

$$T_{pro} = 0.033366 \text{ (us)}$$

$$T_{pro} = 0.000033366 \text{ (ms)}$$

Entonces el tiempo de transmisión de los datos para las condiciones máximas, en el caso que los dispositivos se encuentren separados 10m y que se transmita una trama de datos con el máximo tamaño de 630 bytes es:

$$\text{Tiempo de transmisión de datos} = T_{txMPDU} + T_{pro} + T_{txACK} + T_{prop}$$

$$T_{tx datos} = 40.32 \text{ (ms)} + 0.000033366 \text{ (ms)} + 0.32 \text{ (ms)} + 0.000033366 \text{ (ms)}$$

$$T_{tx datos} = 40.64 \text{ (ms)}$$

40.64 (ms) es el tiempo que dura una transmisión de datos entre dos dispositivos del sistema MSN para las condiciones máximas y además representa el valor del contador NAV.

En referencia a los cálculos realizados los parámetros para la implementación del contador NAV en el programa se resumen en la tabla 3.2

Nombre del parámetro	Valor del parámetro	Nombre de la variable en el programa
Tamaño de la trama	campo length de la trama (Bytes)	size_mpdu
Ancho de banda	125 000 (bits/s)	Definido como constante
Tiempo de transmisión de ACK	320 (us)	Tx_ack
Tiempo de propagación	0.033366 (us)	T_propagation

Tabla 3.2: Parámetros para el programa del protocolo de capa MAC

Diseñado por: Susana Mafla

- **Tiempo de transmisión de tramas RTS y CTS**

Además de los tiempos descritos anteriormente en el protocolo propuesto se incluye el intercambio de tramas RTS y CTS entre los dispositivos del sistema MSN, se calcula el tiempo de transmisión de estas tramas para obtener un dato del tiempo de retardo en la comunicación introducido con la implementación del protocolo.

- **Trama RTS**

Tamaño de trama RTS: 57 bytes

$$T_{tx\ RTS} = \frac{\text{tamaño de la trama (bits)}}{\text{Ancho de banda (bits/s)}}$$

$$T_{tx\ RTS} = \frac{57 \times 8 \text{ (bits)}}{125\ 000 \text{ (bits/s)}}$$

$$T_{tx\ RTS} = 3.64 \text{ (ms)}$$

$$T_{pro} = 0.000033366 \text{ (ms)}$$

- **Trama CTS**

Tamaño de trama CTS: 57 bytes

$$T_{tx\ CTS} = \frac{\text{tamaño de la trama (bits)}}{\text{Ancho de banda (bits/s)}}$$

$$T_{tx\ CTS} = \frac{57 \times 8 \text{ (bits)}}{125\ 000 \text{ (bits/s)}}$$

$$T_{tx\ CTS} = 3.64 \text{ ms}$$

$$T_{pro} = 0.000033366 \text{ (ms)}$$

El tiempo de retardo incluido por la transmisión de tramas RTS y CTS es igual a:

$$T_{tx\ intercambio\ de\ tramas\ RTS\ y\ CTS} = T_{tx\ RTS} + T_{pro} + T_{tx\ CTS} + T_{prop}$$

$$T_{tx\ RTS_{CTS}} = 3.64 \text{ (ms)} + 0.000033366 \text{ (ms)} + 3.64 \text{ (ms)} + 0.000033366 \text{ (ms)}$$

$$T_{tx\ RTS_{CTS}} = 7.28 \text{ (ms)}$$

3.8 IMPLEMENTACIÓN DEL PROTOCOLO DE CAPA MAC

Una vez analizada la solución propuesta para el protocolo de capa MAC, a continuación se describe el proceso realizado para la implementación de dicha solución. El protocolo de capa MAC se desarrolló bajo la plataforma de CrossWorks para ARM, versión 2.0, de la compañía Rowley Associates. Éste software es un sistema completo para el desarrollo, edición, simulación, depuración, compilación y ensamblado de código en lenguaje C/C++ para

microcontroladores ARM. En la figura 3.10 se observa la interfaz gráfica del programa CrossWorks.

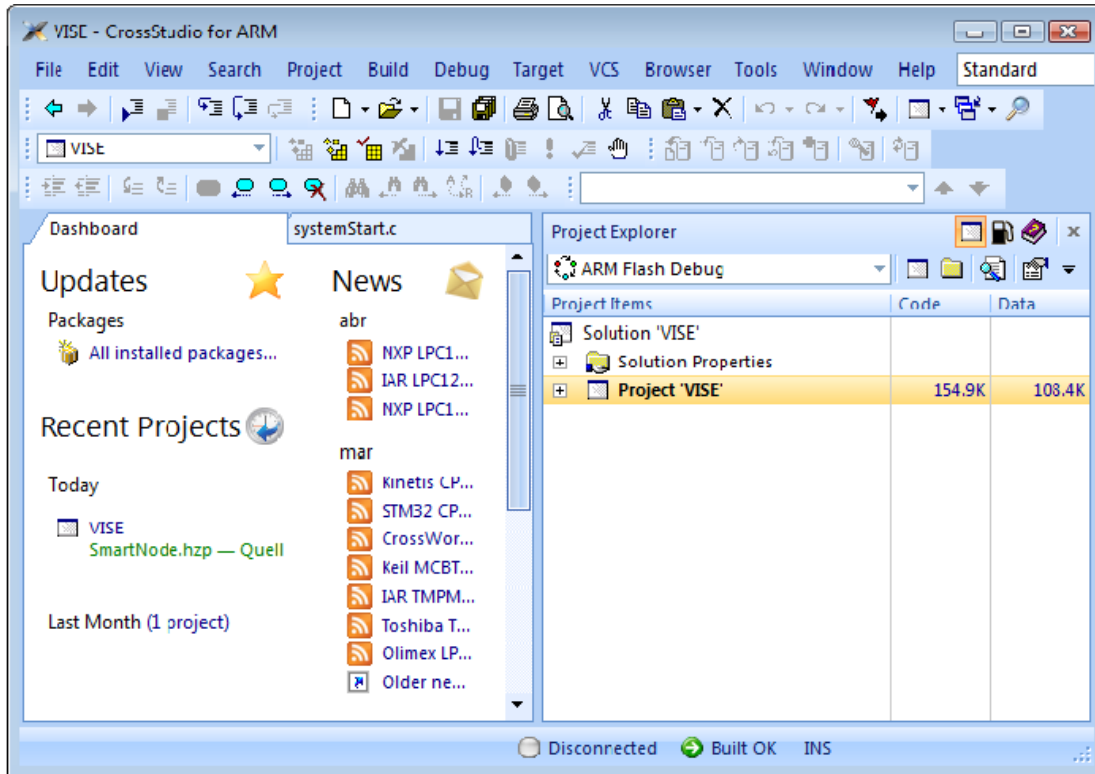


Figura 3.10: Interfaz del programa CrossWorks para ARM.

Tomado por: Susana Mafla

Como se describió en el tema 2.4.4, el proyecto VISE es el software, desarrollado en el programa CrossWorks, en el cual se ha programado toda la estructura lógica para el sistema MSN. El proyecto VISE se encuentra estructurado por 3 ficheros: *Project Properties* que contiene las propiedades del proyecto, *System Files* donde se guardan las librerías de configuración de los dispositivos utilizados y finalmente *Source Files* que contiene toda la estructura lógica desarrollada. En la figura 3.11 se visualiza el proyecto completo.

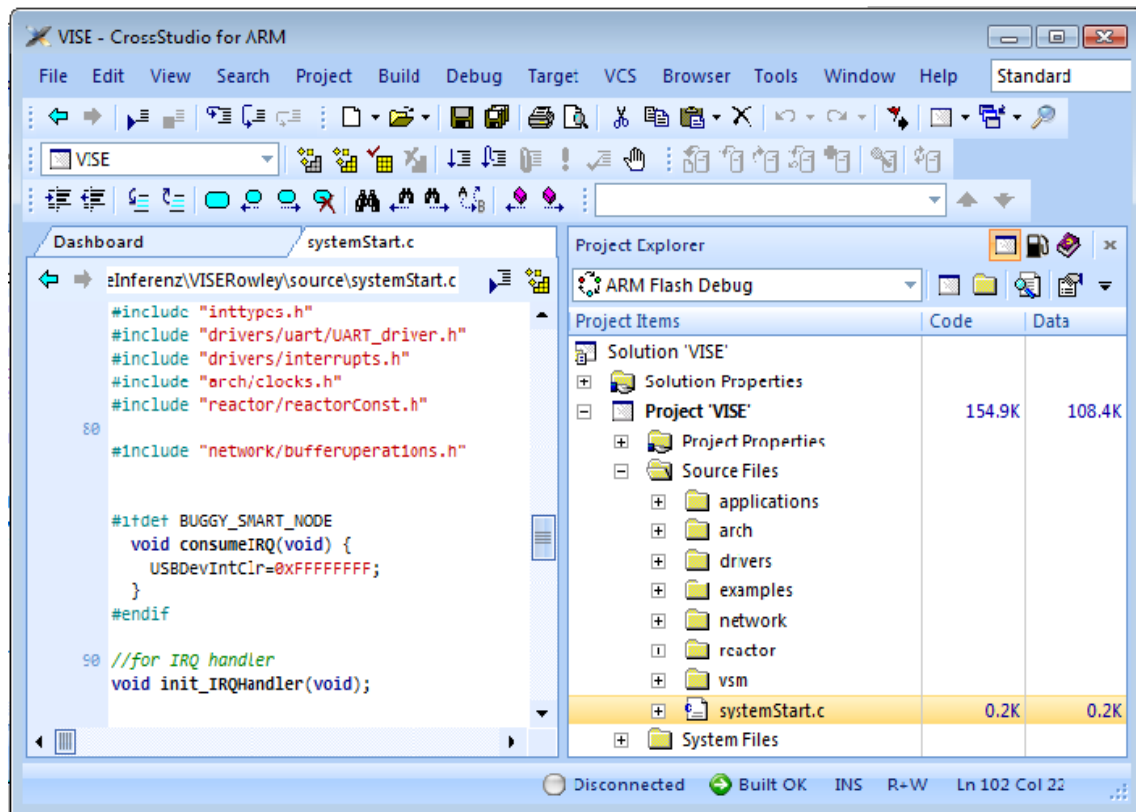


Figura 3.11: Proyecto VISE

Tomado por: Susana Mafla

Con la herramienta Project Explorer del programa CrossWorks se puede visualizar los componentes del proyecto. El fichero *Source files* tiene 6 ficheros en los cuales a su vez se guardan los archivos ejecutables en donde se encuentran programadas las funciones lógicas del sistema.

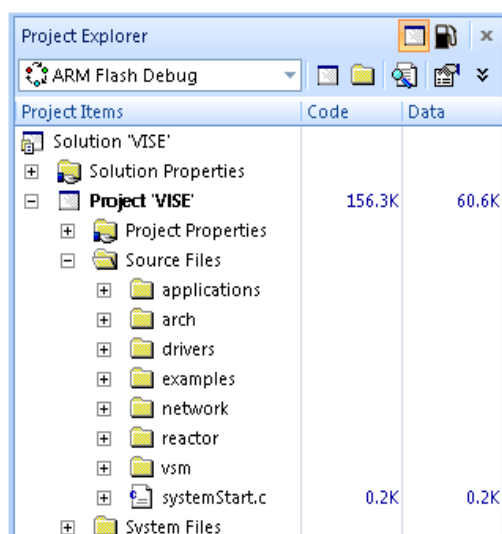


Figura 3.12: Ficheros del proyecto VISE

Tomado por: Susana Mafla

En la figura 3.12 se amplía el campo de la herramienta Project Explorer donde se observa los ficheros que contiene *Source Files* y los cuales se detallan a continuación:

- *Applications.*- Contiene la estructura de las aplicaciones que puede realizar VISE.
- *Arch.*- Este archivo tiene las subrutinas para crear los tiempos de retrasos y los relojes utilizados por el programa. Aquí también se encuentran programadas las interrupciones.
- *Drivers.*- Contiene la configuración de todos los controladores del Hardware de la placa utilizada.
- *Examples.*-Aquí se encuentran programadas aplicaciones de menor importancia como ejemplos y pruebas.
- *Network.*- Contiene toda la estructura necesaria basada en el protocolo TCP/IP para que los dispositivos del sistema MSN puedan transmitir datos dentro de la red inalámbrica establecida.
- *Reactor.*- Contiene la configuración del núcleo del sistema o kernel y también la configuración del temporizador del sistema.
- *Vsm.*- Virtual State Machine (Estado de Máquina Virtual), contiene la programación para crear un emulador simple para ejecutar un estado de máquina virtual usando un sistema de mensajería.
- *Fichero SystemStart.c.*- Es el main o programa principal de proyecto.

3.8.1 LOS ARCHIVOS *mac_Protocol_RTS_CTS.c/h*

Luego de un breve análisis de la estructura del programa VISE a continuación se describe el desarrollo del código fuente para el protocolo de capa MAC propuesto, su implementación en el proyecto VISE y en los dispositivos del sistema MSN.

Dentro del proyecto *VISE*, en el fichero *Source Files*, en el fichero *network* y a su vez dentro del fichero *mac802154*, se creó el archivo ejecutable del protocolo de

capa MAC denominado *mac_Protocol_RTS_CTS.c*, junto con su respectivo archivo de cabecera llamado *mac_Protocol_RTS_CTS.h*, en la figura 3.13 se encuentra una captura de pantalla donde se visualiza la ubicación de los archivos.

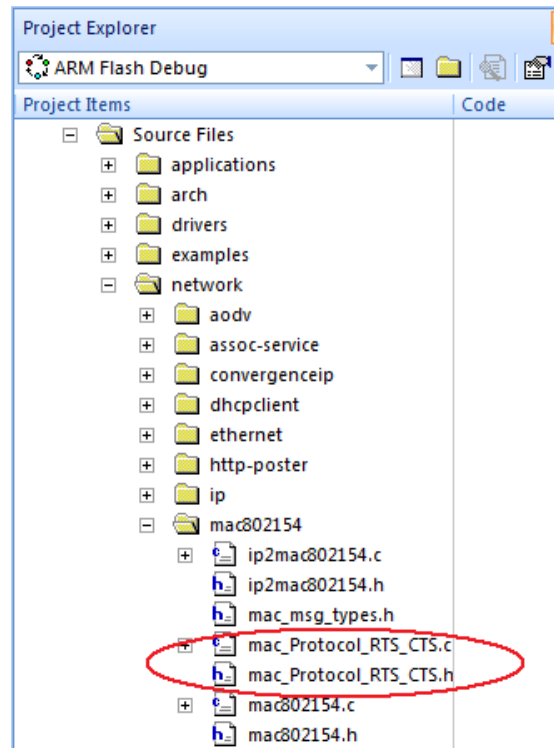


Figura 3.13: Ubicación de los archivos ejecutables *mac_Protocol_RTS_CTS*

Tomado por: Susana Mafla

Luego se realizó una copia de todo el proyecto VISE para programar por separado las subrutinas del MSN y del FESN y así evitar confusiones y reducir el tamaño del código en cada archivo.

3.8.1.1 Proceso para la programación de las subrutinas del protocolo de capa MAC

Para implementar las subrutinas del protocolo de capa MAC propuesto en los archivos ejecutables *mac_Protocol_RTS_CTS* se sigue un proceso de programación que consiste primeramente en la codificación del programa, que implica la escritura real del programa en lenguaje C. Una vez escrito el programa, se realiza la verificación de escritorio del mismo, luego se compila el programa, el compilador procesa el programa escrito en lenguaje de alto nivel para generar un

archivo objeto. Seguidamente se ejecuta el programa donde se combina el archivo objeto generado con las librerías necesarias creando un archivo ejecutable el cual está listo para ser depurado. La depuración proporciona una vista de lo que sucede en el programa y así detectar posibles errores que luego de un análisis se corrigen. Este proceso viene a ser un ciclo repetitivo que se muestra en la figura 3. 14.

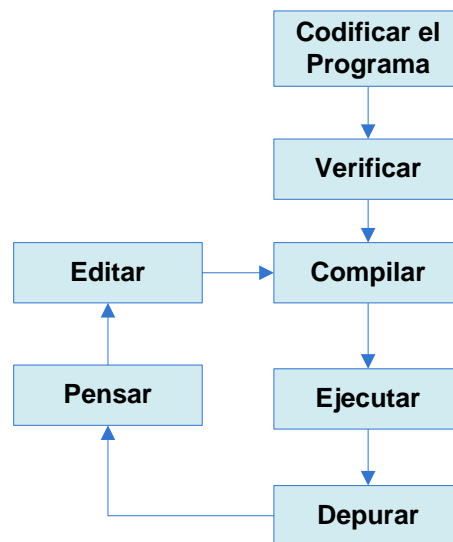


Figura 3.14: Ciclo de desarrollo del archivo ejecutable *mac_Protocol_RTS_CTS*

Diseñado por: Susana Mafla

Con el proceso anterior se programó los archivos *mac_Protocol_RTS_CTS* tanto para el MSN y el FESN. El archivo *mac_Protocol_RTS_CTS.c* dependiendo del tipo de dispositivo, contiene el código fuente de las subrutinas del algoritmo que se encuentra detallado en los temas del 3.7.1, al 3.7.3 del presente capítulo y el archivo *mac_Protocol_RTS_CTS.h* es el archivo de cabecera. El código completo de estos archivos para cada tipo de dispositivo se encuentra en el [Anexo 6](#).

3.8.2 PROGRAMACIÓN DEL PROTOCOLO EN EL MICROCONTROLADOR LPC 2148 DEL SISTEMA MSN

Una vez que se creó el código del protocolo de capa MAC dentro de los archivos *mac_Protocol_RTS_CTS.c/h* en el proyecto VISE, para programar los cambios en los dispositivos del MSN primeramente se establece una conexión física entre el ordenador en donde se ejecuta el programa CrossWorks que contiene el código

fuente y la placa de los dispositivos MSN. CrossWorks es capaz de descargar y grabar código en la memoria de los microcontroladores a través de interfaces JTAG (Joint Test Action Group), éste tipo de interfaz es comúnmente utilizada para programar microcontroladores basados en sistemas embebidos, ya que a través de ella se puede acceder directamente a todos los chips de un sistema embebido. Para grabar el código del protocolo de capa MAC en la memoria del microcontrolador LPC2148 del sistema MSN, se utiliza la interfaz Macraigor Wiggler que es compatible con CrossWorks. En la figura 3.15 se muestra dicha interfaz.



Figura 3.15: Interfaz Macraigor Wiggler

Tomado por: Susana Mafla

El Macraigor Wiggler es una interfaz que se usa en el diseño, depuración y programación de microprocesadores y microcontroladores basados en sistemas embebidos. Un extremo de la interfaz se conecta al puerto paralelo de un host PC y el otro extremo se conecta a un puerto JTAG del sistema de destino. Los controladores de esta interfaz están incluidos en el programa CrossWorks.

La placa del sistema MSN tiene un conector JTAG macho de 10 pines, mientras que la interfaz Macraigor Wiggler tiene un conector JTAG hembra de 20 pines. Para solucionar esta incompatibilidad se usa una placa adicional que convierte la salida del conector JTAG Macraigor Wiggler de 20 pines hembra, a un conector JTAG de 10 pines hembra (conexión de los pines, [Anexo 7](#)). En la figura 3.16 se visualiza la placa antes mencionada.



Figura 3.16: Placa convertidor JTAG hembra de 20 pines a 10 pines

Tomado por: Susana Mafla

El montaje completo con la interfaz y el conector utilizados para programar el microcontrolador de las placas de los dispositivos MSN, desde el programa CrossWorks se muestra en la figura 3.17

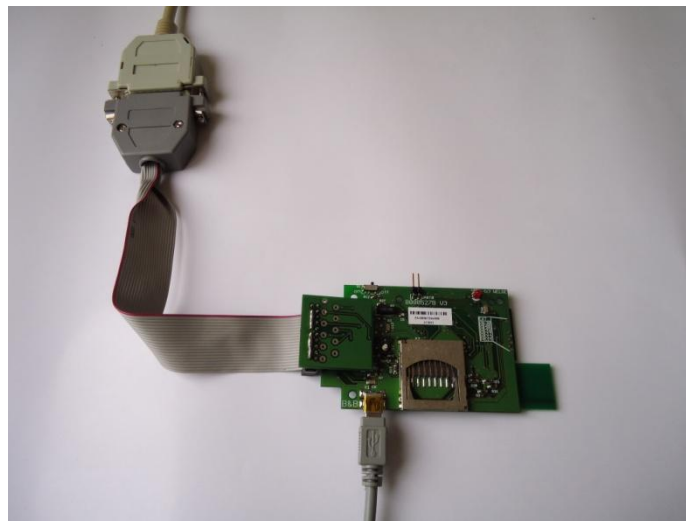


Figura 3.17: Montaje completo para la programación de las placas del sistema MSN.

Tomado por: Susana Mafla

Una vez que se establece la conexión física entre la placa del MSN y el ordenador en donde se ejecuta el programa CrossWorks con el proyecto VISE, para programar el microcontrolador LPC2148 se necesita primeramente definir al

dispositivo como MSN o FESN en el código del proyecto, para esto se especifica el tipo dentro de la definición del dispositivo que se encuentra en la carpeta Reactor, en el archivo ejecutable *reactorConst.h*. Para definir al dispositivo como MSN se activa la instrucción *#define RF154PEER* mientras que para definirlo como FESN se activa *#define FESN*, tomando en cuenta que las dos instrucciones no deben activarse a la vez ya que se produce un error.

Luego de definir al dispositivo se compila el programa con la función Ctrl+ F7, se ejecuta con F7, se depura para comprobar que no existe ningún error y finalmente se graba el programa en el microcontrolador con la función F5.

El proceso anterior se repite con cada uno de los dispositivos del sistema MSN, asignando siempre el rol del dispositivo en la red como MSN o FESN.

4 PRUEBAS DE FUNCIONAMIENTO



4.1 GENERALIDADES

4.2 PRUEBAS DE FUNCIONAMIENTO DEL PROTOCOLO DE CAPA MAC EN EL SISTEMA MSN

4.1 GENERALIDADES

El objetivo del cuarto capítulo del presente proyecto es evaluar y comprobar el funcionamiento del protocolo que se desarrolló e implementó en el capítulo anterior. Evaluar es una herramienta muy útil para detectar posibles fallas y que estas sean eliminadas, examinar el comportamiento del protocolo bajo condiciones específicas, mejorar el trabajo realizado o simplemente para corroborar el buen funcionamiento del protocolo. Es así que la evaluación del protocolo de capa MAC para el sistema MSN se realiza mediante la comprobación de su funcionamiento en el mismo sistema MSN con los dispositivos que describen una red inalámbrica. De los resultados de la evaluación surgen algunas conclusiones para la redacción del capítulo final.

4.2 PRUEBAS DE FUNCIONAMIENTO DEL PROTOCOLO DE CAPA MAC EN EL SISTEMA MSN

En las pruebas de funcionamiento del protocolo se propone crear una colisión intencional entre los dispositivos del sistema MSN que describen una red inalámbrica y así verificar si el protocolo soluciona el conflicto ocasionado intencionalmente.

Para ocasionar una colisión entre los dispositivos inalámbricos se desarrolla un algoritmo que establece dicha condición. Al provocarse una colisión entre los dispositivos inalámbricos del sistema MSN, entra en funcionamiento el protocolo de capa MAC implementado y de acuerdo a su desempeño se valora los resultados obtenidos.

4.2.1 CONDICIONES PARA ESTABLECER UNA COLISIÓN ENTRE DISPOSITIVOS INALÁMBRICOS

Para crear la situación de colisión se utiliza tres dispositivos inalámbricos: un FESN y dos MSN, el objetivo es mantener a uno de los dos MSN transmitiendo ininterrumpidamente hacia el FESN, mediante la creación de un nuevo programa

basado en un algoritmo para transmitir constantemente. Cuando el segundo MSN transmite hacia el FESN se produce una colisión.

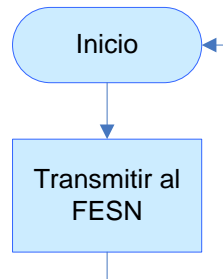


Figura 4.1: Subrutina de un MSN para transmitir constantemente.

Diseñado por: Susana Mafla

La figura 4.1 muestra el diagrama de flujo del algoritmo que utiliza uno de los dos dispositivos MSN para transmitir constantemente datagramas UDP hacia el FESN.

En la implementación del algoritmo, se creó dos archivos: `udp_continuos_tx.c` y `udp_continuos_tx.h` dentro del fichero Examples del proyecto VISE. En el [Anexo 8](#) se encuentra el código fuente de los archivos ejecutables con sus respectivos comentarios. El archivo `udp_continuos_tx.h` es el archivo de cabecera que contiene las librerías utilizadas, la declaración de ciertas variables y llama a las funciones que están definidas en el archivo ejecutable `udp_continuos_tx.c`. Mientras tanto el archivo ejecutable `udp_continuos_tx.c` contiene la definición de las funciones utilizadas para la transmisión constante de datagramas UDP. En primer lugar se solicita la asignación de un puerto y de un temporizador a través de la función `void initudp_tx ()`. Luego se encuentra la función `deactiveudp_continuos_tx ()` que sirve para desactivar las funciones ejecutadas por `udp_continuos_tx.c`. Continuando se define la función `udp_tx_register_process ()` que sirve para registrar al puerto y a la señal de identificación con la que se trabajará en `udp_continuos_tx.c`, también se tiene la función `udp_tx_deregister_process ()` que realiza el proceso contrario, eliminar el registro de un puerto. Finalmente se encuentra la función

udp_tx_reactor_port () que es la responsable de transmitir el mensaje deseado utilizando las funciones del reactor.

4.2.2 REALIZACIÓN DE LAS PRUEBAS DE FUNCIONAMIENTO

En la ejecución de las pruebas de funcionamiento como se describe anteriormente se utiliza tres dispositivos los cuales se programan cada uno por separado con el software del proyecto VISE, para que cumplan las funciones de un FESN y de dos MSN.

Para visualizar los mensajes que se transmiten entre los dispositivos del sistema MSN se utiliza la comunicación UART de los dispositivos, ya que a través de ella se puede comunicar con un puerto COM de un ordenador y a su vez utilizar el HyperTerminal de Windows para observar los mensajes.

4.2.2.1 Programación de los dispositivos

En la primera parte de la prueba se graba uno de los MSN con el algoritmo descrito en el tema 4.2.1, para que el dispositivo esté constantemente enviando datagramas UDP hacia el FESN. El otro MSN junto con el FESN se programan con el software normal que les corresponde para desempeñar sus funciones en el sistema. Programados cada uno de los dispositivos se pone en funcionamiento al sistema, en primera instancia el MSN transmite mensajes ininterrumpidamente al FESN y luego cuando se presiona el joystick del segundo MSN se envía un mensaje al FESN para que se produzca una colisión.

En la segunda etapa de la prueba los tres dispositivos del sistema MSN se programan con el protocolo de capa MAC y se repite el proceso anterior para evaluar cómo responde el sistema con el protocolo implementado y verificar que no se produzca más una colisión.

4.2.2.2 Comunicación entre un ordenador y un dispositivo del sistema MSN

Para establecer una comunicación serial entre un computador utilizando el HyperTerminal de Windows y un dispositivo del sistema MSN, se hace que los datos recibidos por el dispositivo se retransmitan a su vez por comunicación UART hacia un conversor, en este caso el Ft232BL, que convierte la señal de comunicación UART a USB, a su vez en el computador se instala un controlador que crea un puerto COM virtual a partir del puerto USB, con el puerto COM virtual se establece un comunicación serial a través del HyperTerminal de Windows. En la figura 4.2 se observa un diagrama que explica el proceso para establecer la comunicación entre un ordenador y un dispositivo del sistema MSN.

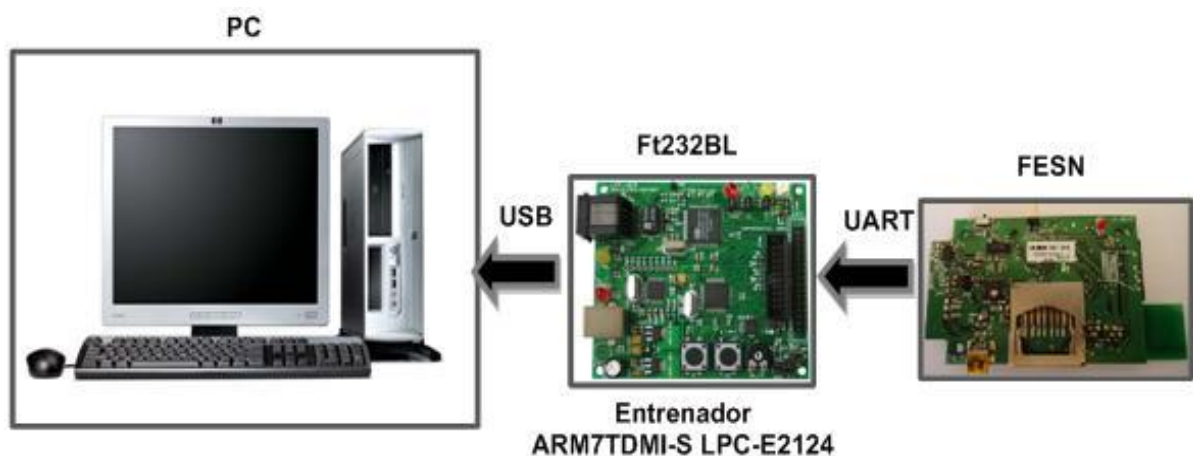


Figura 4.2: Proceso para la comunicación entre un computador y un dispositivo del sistema MSN

Diseñado por: Susana Mafla

El Ft232BL como se menciona es una interfaz para convertir una señal serial UART a USB, sus características están detalladas en el [Anexo 9](#) . Para la realización de las pruebas de funcionamiento se utiliza el chip Ft232BL que se encuentra en la placa del entrenador para microcontroladores ARM7TDMI-S LPC-E2124, en el [Anexo 10](#) se presentan las características de éste entrenador. En la figura 4.3 se observa una foto del sistema MSN con tres dispositivos: el nodo máster o FESN y dos nodos esclavos o MSN, más el entrenador LPC-E2124 donde se localiza el conversor Ft232BL.

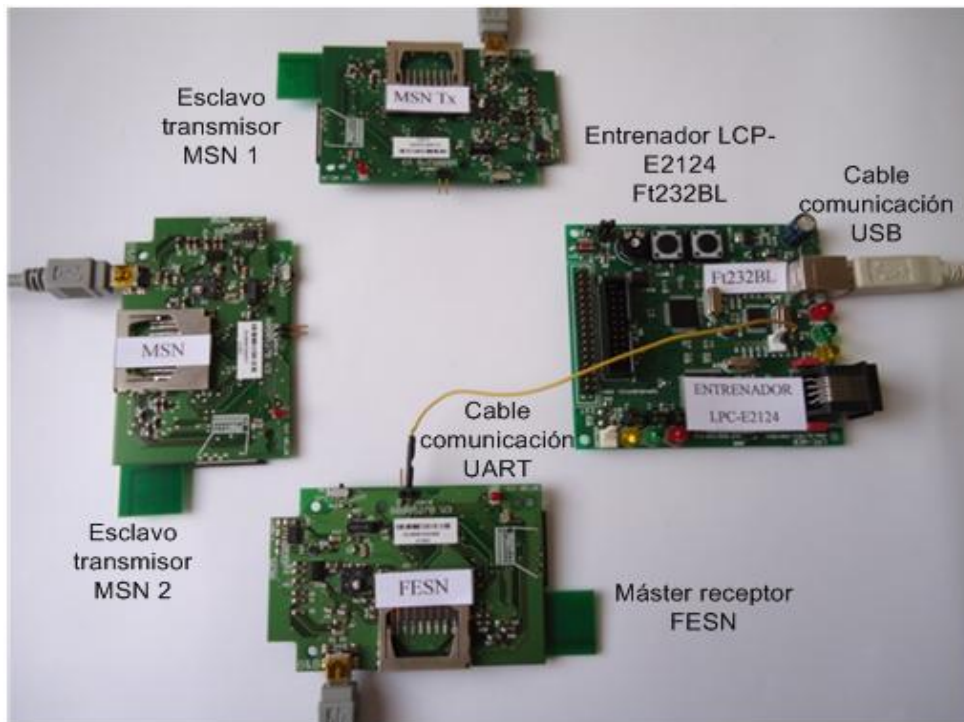


Figura 4.3: Montaje completo del sistema MSN.

Diseñado por: Susana Mafla

- **Configuración y utilización del HyperTerminal de Windows**

Una vez realizadas las conexiones físicas entre los dispositivos necesarios y el ordenador, para visualizar los mensajes transmitidos entre los dispositivos del sistema MSN, el HyperTerminal de Windows tiene que ser configurado con ciertos parámetros que se describen a continuación:

1. Acceder al programa HyperTerminal y en la ventana emergente "Descripción de la conexión" se asigna un nombre y un ícono para la conexión.



Figura 4.4: Descripción de la conexión del HyperTerminal

Tomado por: Susana Mafla

2. En la ventana emergente “Conectar a” se selecciona el puerto COM asignado por el computador luego de conectar el entrenador LPC-E2124.



Figura 4.5: Selección del puerto COM para el HyperTerminal

Tomado por: Susana Mafla

3. Luego en la siguiente ventana “Propiedades de COM” se especifica las propiedades de la conexión.

- Bits por segundo: 57600 (valor que se define en el Main del proyecto VISE, en el archivo ejecutable systemStart.c)
- Bits de datos: 8
- Paridad: Ninguna
- Bits de parada: 1
- Control de flujo: Hardware

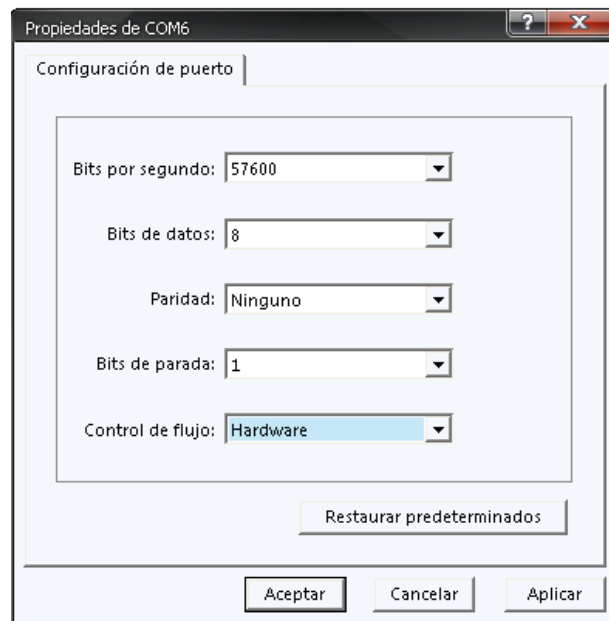


Figura 4.6: Especificación de las propiedades de conexión del HyperTerminal

Tomado por: Susana Mafla

Cuando aparece la ventana de sesión de HyperTerminal se enciende el dispositivo FESN y se empieza a recibir los mensajes.

4.2.3 RESULTADOS DE LAS PRUEBAS DE FUNCIONAMIENTO

Como se esperaba en la primera etapa de la prueba, el MSN se mantiene transmitiendo hacia el FESN sin ningún problema. El HyperTerminal recibe los mensajes de la aplicación *udp_continuos_tx*, que define las funciones necesarias para transmitir constantemente datagramas UDP hacia el FESN. En la figura 4.7 se observan los mensajes recibidos.

```

Sistema_MSN - HyperTerminal
File Edit View Call Transfer Help
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Mensaje: recibido "UP" desde 0xAC100201
Check if ip in slot 0 is still alive
Mensaje: recibido "UP" desde 0xAC100201
TESN 0 still alive
Mensaje: recibido "UP" desde 0xAC100201
Check if ip in slot 0 is still alive
Disconnected Auto detect 57600 8-N-1 SCROLL CAPS NUJI

```

Figura 4.7: Mensajes recibidos, MSN transmitiendo ininterrumpidamente hacia el FESN

Tomado por: Susana Mafla

El mensaje que se recibe continuamente es *“Mensaje: recibido “UP” desde 0xAC100201”*, además se especifica la dirección IP del dispositivo MSN desde donde se transmiten los mensajes. Cada cierto tiempo el reactor de la pila VISE comprueba si los dispositivos MSN que están asociados a la red se encuentran activos y se recibe el mensaje *“Check if IP in slot 0 is still alive”* que indica este proceso de comprobación y luego se recibe un mensaje que informa el estado de los dispositivos MSN *“TESN 0 still alive”*. Continuando con la prueba, cuando el segundo MSN envía un mensaje se produce una colisión y el FESN tiene un conflicto, dejando de transmitir datos hacia el HyperTerminal como se visualiza en la figura 4.8.

```

Mensaje: recibido "UP" desde 0xAC100203
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Next hop for destination 172.16.2.2unknown
AODV: Send RREQ
Send RREQ for 172.16.2.2on behalf of 172.16.1.20 from 172.16.1.20.
insert into bacast cache
bcast send restore keep
Dumping local message to 9: No free message slots available.
Reactor message slot status
Available slots: 32
Currently used slots:
Slot 0 : Message 999 to IP2MAC802154
Slot 1 : Message 999 to IP2MAC802154
Slot 2 : Message 9999 to IP2MAC802154
Slot 3 : Message 999 to IP2MAC802154
Slot 4 : Message 1 to TCP_HLPR
Slot 5 : Message 999 to MAC_802154
Slot 6 : Message 3 to MAC_802154

```

Figura 4.8: Interrupción en la recepción de mensajes por colisión

Tomado por: Susana Mafla

Cuando el segundo MSN envía un mensaje corta la comunicación en curso entre el primer MSN y el FESN, en este momento el reactor de la pila VISE del FESN comprueba si los dispositivos MSN siguen activos, al no recibir ninguna señal se informa que el siguiente salto para el destino 172.16.2.2 es desconocido y el protocolo de ruteo vector distancia En-demanda ad hoc (AODV) envía una solicitud de ruta o RREQs (Route Requests) a la dirección IP del dispositivo que dejó de transmitir en nombre y desde la dirección IP del FESN.

Next hop for destination 172.16.2.2 unknown

AODV: Send RREQ

Send RREQ for 172.16.2.2 on behalf of 172.16.1.20 from 172.16.1.20.

Luego se inserta este suceso en la memoria caché de broadcast bacast y se envía un mensaje para informar que se ha enviado a restaurar y guardar la información.

insert into bacast cache

bcast send restore keep

Como se produce una colisión el sistema falla y el reactor que es el sistema operativo del FESN envía un mensaje informando que se han desbordado los slots para mensajes locales, que vienen a ser las solicitudes que se pueden hacer al reactor y se comunica que no hay slots libres disponibles, se informa sobre el estado de los slots de mensajes del reactor, el número de slots disponibles y el uso actual de cada slot. El reactor del dispositivo FESN se queda bloqueado en este proceso y no responde a ninguna otra solicitud.

Dumping local message to 9: No free message slots available

Reactor message slot status

Available slots: 32

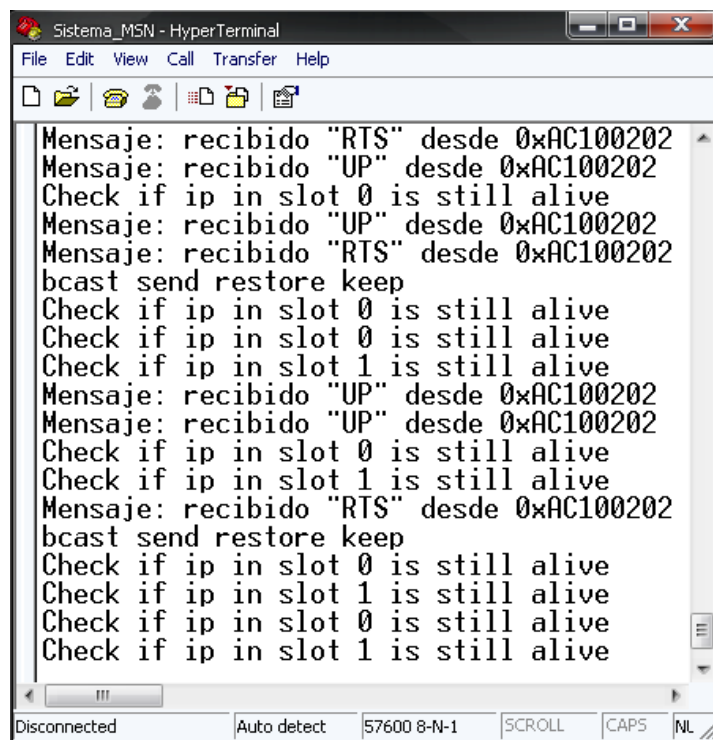
Currently used slots:

Slot 0: Message 999 to IP2MAC802154

Slot 1: Message 999 to IP2MAC802154

.....

Posteriormente se programa a los dispositivos con el protocolo de capa MAC propuesto y se repite el proceso anterior, pero esta vez, al momento que el segundo MSN envía un mensaje al FESN, el reactor del FESN continúa funcionando y se sigue recibiendo los mensajes que el primer MSN transmite durante todo el tiempo. Demostrando que con la implementación del protocolo de capa MAC propuesto, no se produce ninguna colisión y la comunicación en curso entre el primer MSN y el FESN no se interrumpe. En la figura 4.9 se observa los datos que recibe el HyperTerminal al momento en el cual el segundo MSN envía un mensaje al FESN.



```
Sistema_MSN - HyperTerminal
File Edit View Call Transfer Help
Mensaje: recibido "RTS" desde 0xAC100202
Mensaje: recibido "UP" desde 0xAC100202
Check if ip in slot 0 is still alive
Mensaje: recibido "UP" desde 0xAC100202
Mensaje: recibido "RTS" desde 0xAC100202
bcast send restore keep
Check if ip in slot 0 is still alive
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Mensaje: recibido "UP" desde 0xAC100202
Mensaje: recibido "UP" desde 0xAC100202
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Mensaje: recibido "RTS" desde 0xAC100202
bcast send restore keep
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Disconnect Auto detect 57600 8-N-1 SCROLL CAPS NL
```

Figura 4.9: Recepción de mensajes en el HyperTerminal después de implementar el protocolo de capa MAC en el sistema MSN

Tomado por: Susana Mafla

Adicionalmente se explican los pasos del procedimiento para la realización de las pruebas de funcionamiento en el [Anexo11](#).

5 CONCLUSIONES Y RECOMENDACIONES



5.1 CONCLUSIONES

5.2 RECOMENDACIONES

5.1 CONCLUSIONES

- El cambio climático está causando problemas no solo ambientales, sino también económicos, sociales y culturales; que en el futuro serán más graves, si los seres humanos no actuamos desde ya. Cada persona, debe cambiar su mentalidad y actuar con orientación hacia el concepto de un desarrollo sustentable. Este concepto es tan amplio e inclusivo que permite que desde cada perspectiva y disciplina se pueda contribuir para alcanzarlo, es así que la tecnología se perfila como una herramienta muy útil y quienes se relacionan con ella no están excluidos de colaborar.
- La eficiencia energética es una estrategia clave para disminuir los impactos negativos producidos por la generación, transporte y uso de energía eléctrica. Para lograr una eficiencia energética se pueden llevar a cabo cambios en la mentalidad de las personas, mejoras en los dispositivos eléctricos, etc. Pero siempre existirán ciertas fallas que con ayuda de la inteligencia ambiental por ejemplo pueden ser evitadas, como es el caso del sistema Mobile Smart Node.
- La función del sistema Mobile Smart Node es la administración y uso eficiente de energía eléctrica en ambientes domésticos por medio de un sistema embebido, basado en el concepto de inteligencia ambiental y que maneja una pila de comunicación bajo el protocolo TCP/IP. La capa de acceso se define por el estándar IEEE 802.15.4 que permite un bajo consumo de energía y costos reducidos ya que la aplicación no requiere altas tasas de transmisión de datos y ni alta QoS.
- Se diseñó e implementó un protocolo de capa MAC para el sistema MSN, dicho protocolo fue desarrollado en base al protocolo MACAW, ya que MACAW soluciona los problemas de CSMA, mejora el desempeño del

protocolo MACA y con los ajustes realizados es capaz de eliminar las colisiones entre los dispositivos inalámbricos del sistema MSN.

- El protocolo de capa MAC desarrollado consiste en un algoritmo que permite el intercambio de tramas RTS, CTS y ACK entre los dispositivos MSN y FESN, para informar a toda la red inalámbrica cuando existe la intención de comenzar una comunicación, alertando a los otros dispositivos para que estos no interfieran, eliminando así los problemas de nodo escondido y nodo expuesto que dan lugar a que se produzcan colisiones.
- Para evitar que los dispositivos MSN transmitan simultáneamente hacia un FESN y se produzcan colisiones se incluyó un tiempo de espera aleatorio antes de transmitir un RTS. Éste tiempo de espera está definido por un espacio inter-trama IFS.
- Para implementar el protocolo de capa MAC en el sistema MSN se dividió al algoritmo en tres subrutinas, dos para el dispositivo transmisor o MSN y una para el dispositivo receptor o FESN. Estas subrutinas fueron desarrolladas en el programa de CrossWorks en lenguaje C.
- En la ejecución del presente proyecto se trabajó con el programa CrossWorks para la programación de código en lenguaje C, probar con nuevas herramientas informáticas aporta al desarrollo de conocimiento y a la investigación.
- Si ocurren colisiones entre tramas del tipo RTS/CTS la pérdida de ancho de banda es menor que en las colisiones entre tramas de datos. Sin embargo, las tramas RTS/CTS introducen retardos en la transmisión.

- Al producirse una colisión entre los dispositivos del sistema MSN, las comunicaciones en curso se interrumpen y el sistema operativo o reactor del FESN comprueba si los dispositivos MSN asociados a la red WPAN están activos, en este proceso se desbordan los slots de memoria para mensajes locales y el reactor del FESN se bloquea y no responde a ninguna solicitud.
- Las pruebas de funcionamiento del protocolo de capa MAC se llevaron a cabo utilizando tres dispositivos inalámbricos; un FESN y dos MSN. En estas pruebas se demostró que el protocolo evita que se produzcan colisiones entre los dispositivos del sistema MSN.

5.2 RECOMENDACIONES

- Alcanzar un desarrollo sustentable no es un tema aislado que compete únicamente a profesionales relacionados a las ciencias ambientales. Todos podemos contribuir, es así que los profesionales de áreas técnicas deberían ampliar más sus perspectivas e involucrarse en dar solución a problemas globales como el cambio climático.
- En el Ecuador y en todo el mundo se está impulsando el uso eficiente de energía, quienes están vinculados a la tecnología y a áreas técnicas deberían desarrollar nuevas tecnologías que mejoren procesos de generación, transporte, distribución y uso de la energía, para además de contribuir con el avance científico y tecnológico colaborar en la disminución de los efectos negativos producidos por el uso de energía.

- En el desarrollo del diseño del protocolo de capa MAC, surgió la idea de incluir dentro del algoritmo un mecanismo de detección y corrección de errores en caso de que las tramas lleguen incompletas o con errores. Se recomienda que se analice la factibilidad de desarrollar esta función sobre el protocolo de capa MAC para que se implemente a futuro.
- Un protocolo para el acceso al medio libre de colisiones como el desarrollado en el proyecto, también puede ser capaz de intercambiar datos sobre el congestionamiento de la red. En este caso en particular se recomienda no incluir esta función, ya que significa mayor procesamiento de información y volvería lenta la comunicación entre los dispositivos del sistema MSN.
- El programa CrossWorks es una herramienta muy completa para la programación de microcontroladores ARM, AVR, MSP430, y MAXQ. Éste software debería ser estudiado y utilizado por los estudiantes de la Escuela de Electrónica de la FICA para el desarrollo de proyectos con microcontroladores de los tipos mencionados.
- Se recomienda al Instituto VauST de la Universidad Leuphana de Lüneburg implementar el protocolo desarrollado en el presente proyecto para eliminar el problema de colisiones del sistema MSN y así este sistema de administración y uso eficiente de energía eléctrica pueda ser llevado a la aplicación práctica en ambientes domésticos.
- Durante el desarrollo del protocolo se detectó que el tiempo asignado de 7ms para la transmisión de datagramas UDP entre los dispositivos del sistema MSN es muy corto y provoca que el microcontrolador en ciertas ocasiones no responda, así como también que el transceiver del dispositivo receptor se desborde por la cantidad de datagramas que llegan a la vez.

Por este motivo se recomienda aumentar el tiempo para la transmisión de datagramas UDP.

6 BIBLIOGRAFÍA

1. **El universo.** Se acelera deshielo del Cotopaxi. *El universo*. [En línea] 19 de enero de 2007. [Citado el: 10 de octubre de 2008.]
<http://www.eluniverso.com/2007/01/19/0001/12/BA1AB99262D14D9AA0CA1B4AAE37C1FA.html>.
2. **IPCC.** Intergovernmental Panel on Climate Change. [En línea] 2007. [Citado el: 10 de noviembre de 2009.] <http://www.ipcc.ch/>.
3. **CONELC.** Consejo Nacional de Electrificación. [En línea] 2008. [Citado el: 10 de noviembre de 2009.] <http://www.conelec.gov.ec/>.
4. **Naturaleza y Cultura Internacional.** Calentamiento Global. *Calentamiento Global, Salvemos nuestro único planeta*. Loja : Imprenta COSMOS, 2009.
5. **Informe Brundtland.** UN Documents. [En línea] 1987. [Citado el: 15 de octubre de 2008.] <http://www.un-documents.net/wced-ocf.htm>.
6. **Michelsen, Gerd y Rieckmann, Marco.** *Programa de Maestría Internacional "Sustainable Development and Management" Introducción al Desarrollo Sustentable*. Lüneburg : VAS, 2008. Vol. II.
7. **Gutiérrez, Álvaro.** AGROENERGÉTICA. *VENTAJAS E INCONVENIENTES DE LAS ENERGÍAS RENOVABLES*. [En línea] 2007. [Citado el: 20 de octubre de 2009.] <http://agroenergetica.iespana.es/ventajas.html>.
8. **OLADE.** Organización Latinoamericana de Energía. [En línea] 2008. [Citado el: 12 de octubre de 2009.] <http://www.olade.org.ec/eficiencia.html>.
9. **MEER.** Ministerio de Electricidad y Energía Renovable. [En línea] 2008. [Citado el: 12 de octubre de 2009.]
http://www.mer.gov.ec/Meer/porta_l_meer/internaView.htm?code=764&template=meer.internas.
10. **Poveda, Mentor.** Organización Latinoamericana de Energía. *Organización Latinoamericana de Energía*. [En línea] agosto de 2007. [Citado el: 12 de noviembre de 2009.]

<http://www.olade.org.ec/documentos2/EFICIENCIA%20ENERG%C3%89TICA%20RECURSO%20NO%20APROVECHADO-Agosto-2007.pdf>.

11. **Sacristán, A.** La Inteligencia Ambiental aumentará nuestras capacidades cognitivas. *Revista electrónica Megatendencias*. [En línea] 2006. [Citado el: 18 de noviembre de 2009.] http://www.tendencias21.net/La-Inteligencia-Ambiental-aumentara-nuestras-capacidades-cognitivas_a963.html.

12. **Carretero, Noelia y Bermejo, Ana Belén.** Centro de Difusión de Tecnologías, Universidad Politécnica de Madrid. [En línea] 2005. [Citado el: 13 de noviembre de 2009.] www.ceditec.etsit.upm.es/.../5-Resumen-Inteligencia-Ambiental.html.

13. **Weiser, Mark.** Weiser Mark. [En línea] 1991. [Citado el: 24 de noviembre de 2009.] <http://www.ubiq.com/weiser/>.

14. **ZigBee Alliance.** ZigBee Alliance. [En línea] 2009. [Citado el: 16 de diciembre de 2009.] <http://www.zigbee.org/>.

15. **Valverde, Jorge.** El Estándar Inalámbrico ZigBee. [En línea] 2007. [Citado el: 20 de diciembre de 2009.] <http://www.seccperu.org/files/ZigBee.pdf>.

16. **IEEE.** *Standard for Information technology Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate*. New York : s.n., 2006.

17. **Archundia, Fransisco.** Universidad de las Américas Puebla. *Wireless Personal Area Network & Home Networking*. [En línea] diciembre de 2003. [Citado el: 17 de diciembre de 2009.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf.

18. **Mayné, Jordi.** Buenos Aires Robotics Society. *IEEE 802.15.4 y ZigBee*. [En línea] 2007. [Citado el: 12 de enero de 2010.] http://www.bairesrobotics.com.ar/data/ieee_zigbee_silica.pdf.

19. **VauST.** *User Manual for the Mobile Smart Node* . Lüneburg, Alemania : s.n., 20 de octubre de 2008.

20. **Chipcon Products from Texas Instruments.** *CC2420 2.4GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver.* Oslo, Noruega : s.n., 2006.
21. **Comer, Douglas y Stevens, David.** *Interconectividad de Redes con TCP/IP.* s.l. : Prentice Hall, 2000.
22. **Peterson, Larry y Davie, Bruce.** *COMPUTER NETWORKS.* Amsterdam : Morgan Kaufmann Publishers, 2007. 0-12-370548-7.
23. **Stervinou, JY.** XML-RCP.com. [En línea] 2004. [Citado el: 8 de diciembre de 2009.] <http://www.xmlrpc.com/>.
24. **UserLand Software.** XML-RCP.com. [En línea] 2004. [Citado el: diciembre de 8 de 2009.] <http://www.xmlrpc.com/>.
25. **Welge, Ralph.** *Embedded Internet, Embedded Middleware.* Lüneburg, Alemania : s.n., 2006.
26. **Rowley Associates.** CrossStudio Tutorial. *CrossWorks for ARM version 2.0.* [En línea] 2004. [Citado el: 20 de 04 de 2010.] http://www.rowleydownload.co.uk/documentation/arm_2_0/index.htm.
27. **Tanenbaum, Andrew.** *Redes de Computadoras.* Amsterdam : Prentice Hall, 2003.
28. **Pozo, Nelson.** Estudio y diseño de una red lan inalámbrica con QoS, para voz y datos en el CIGMYP, empleando los estándares IEEE802.11 g/e. [En línea] febrero de 2009. [Citado el: 30 de marzo de 2011.] <http://bibdigital.epn.edu.ec/bitstream/15000/1316/1/CD-2019.pdf>.
29. **Suárez, Belén Revert.** Implementación del Protocolo WDS para Redes 802.11. [En línea] <http://upcommons.upc.edu/pfc/bitstream/2099.1/3554/2/41794-2.pdf>.
30. **Pérez, Israel Oropeza.** Análisis de Rendimiento de IEEE 802.11 b. [En línea] 2006. [Citado el: 30 de marzo de 2011.] <http://148.206.53.231/UAMI13093.pdf>.
31. **Facultad de Ciencias Universidad de los Andes.** Velocidad de la Luz. [En línea] [Citado el: 15 de 06 de 2011.] <http://www.ciencias.ula.ve/index.htm>.

32. **Constituyente, Asamblea.** *Constitución del Ecuador.* Montecristi : Publicación Ofical, 2008.
33. **IEA.** International Energy Agency. *International Energy Agency.* [En línea] 2008. [Citado el: 10 de enero de 2009.] <http://www.iea.org/>.
34. **Weizsäcker, E., Lovins A. & Lovins H.** *Informe al Club de Roma.* s.l. : Galaxia Gutenberg, 1997.
35. **Kioskea.** Kioskea.net. *El protocolo ARP.* [En línea] 16 de octubre de 2008. [Citado el: 8 de diciembre de 2009.] <http://es.kioskea.net/contents/internet/arp.php3>.
36. **Valle, Luis.** Universidad de las Américas Puebla, Tesis digitales. *WPAN.* [En línea] Mayo de 2005. [Citado el: 17 de diciembre de 2009.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/valle_i_lf/capitulo2.pdf.
37. **Cantero, Mirna y Oliveira, María Reina.** www.jeuazarru.com. [En línea] 5 de diciembre de 2004. [Citado el: 12 de febrero de 2010.] <http://www.jeuazarru.com/docs/802.11n.pdf>.
38. **Vasilakos, Athanasios y Pedrycz, Witold.** *Ambient Intelligence, Wireless Networking, and Ubiquitous Computing.* Norwood : Artech House, 2006.
39. **Gast, Matthew.** *Redes Wireless 802.11.* Madrid : O'Reilly, 2006.
40. **Stallings, William.** *Comunicaciones y Redes de Computadora.* Madrid : Pretince Hall, 2007.
41. **Schildt, Sebastián.** *Reactor VSM/VISE.* Lüneburg, Alemania : s.n., 2008.
42. **IEEE 802.15 .** IEEE 802.15 Working Group for WPAN. [En línea] 2009. [Citado el: 16 de diciembre de 2009.] <http://www.ieee802.org/15/>.
43. **Kioskea.** Kioskea. *Redes Inalámbricas.* [En línea] 16 de octubre de 2008. [Citado el: 27 de octubre de 2009.] <http://es.kioskea.net/contents/wireless/wlintro.php3#>.
44. **Roth, Jörg.** *Mobile Computing.* Heidelberg : dpunkt.verlag, 2004. 3-89864-165-1.

45. **Staugaard, Andrew.** *Técnicas Estructuradas y Orientadas a objetos, una Introducción Utilizando C++*. 1998 : Prentice Hall. 970-17-0130-5.
46. **OMNeT ++ Community.** OMNeT ++. [En línea] 2001-2009. [Citado el: 2 de julio de 2010.] <http://www.omnetpp.org>.
47. **C., Andrew y Jr., Staugaard.** *Técnicas estructuradas y orientadas a objetos, una introducción utilizando C++*.
48. **Cohon, James y Davidson, Jack.** *Programación y Diseño en C++*. Madrid : Mac Graw Hill, 2000. 84-481-2682-3.
49. **Future Technology Devices International Ltd.** Future Technology Devices . *FTDI Chip*. [En línea] 2011. [Citado el: 14 de 02 de 2011.] <http://www.ftdichip.com>.
50. **Philips Semiconductors.** *LPC214x User Manual*. s.l. : Philips Electronics, 2005.
51. **OLIMEX.** LPC-E2124 BOARD WITH ETHERNET INTERFACE FOR LPC2124 ARM7TDMI-S MICROCONTROLLER. [En línea] [Citado el: 28 de 03 de 2011.] <http://www.olimex.com/dev/lpc-e2124.html>.
52. **Future Technology Devices International Ltd.** Future Technology Devices International. [En línea] [Citado el: 28 de 03 de 2011.] <http://www.ftdichip.com/Products/ICs/FT232BM.htm>.

7 SIGLAS Y ABREVIACIONES

ACK	Acuse de Recibo (Acknowledgment)
AODV	Vector de Distancia de En-demanda ad hoc (Ad hoc On-Demand Distance Vector)
ARP	Protocolo de Resolución de Direcciones (Address Resolution Protocol)
CONELEC	Consejo Nacional de Electrificación
CRC	Código de Redundancia Cíclica (Cyclic Redundancy Check)
CSMA-CA	Acceso Múltiple por Detección de Portadora con Anulación de Colisiones (Carrier Sense, Multiple Access, Collision Avoidance)
CTS	Libre para Envío (Clear To Send)
DCF	Función de Coordinación Distribuida (Distributed Coordination Function)
DHCP	Protocolo Configuración Dinámica de Servidor (Dynamic Host Configuration Protocol)
DIFS	Espacio inter- trama DCF (DCF Interframe Spacing)
DNS	Sistema de Nombre de Dominio (Domain Name System)
DSSS	Espectro Expandido en Secuencia Directa (Direct Sequence Spread Spectrum)
EIFS	Espacio inter- trama extendido (Extended Interframe Spacing)
FCS	Chequeo de Secuencia de Trama (Frame Check Sequence)
FESN	Nodo Inteligente Embebido Completo Full Embedded Smart Node
FFD	Dispositivo de Funciones Completas (Full Function Device)
GTS	Intervalos de Tiempo de Garantía (Guaranteed Time Slot)
HTTP	Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol)
IEA	Agencia Internacional de Energía (International Energy Agency)

IEEE	Instituto de Ingenieros Electricistas y Electrónicos (Institute of Electrical and Electronics Engineers)
IFS	Espacio inter trama (Interframe Space)
IP	Protocolo de internet (Internet Protocol)
IPCC	Panel Intergubernamental sobre Cambio Climático (Intergovernmental Panel on Climate Change)
IRQ	Pedidos de Interrupción (Interrupt Request)
ISR	Rutina de servicio de interrupciones (Interrupt Service Routine)
JTAG	Joint Test Action Group
LLC	Control de Enlace Lógico (Logical Link Control)
LR-WPAN	Redes Inalámbricas de Área Personal con Tasas Bajas de Transmisión de Datos (low-rate wireless personal area network)
MAC	Control de Acceso al Medio (Media Access Control)
MACA	Acceso Múltiple con Prevención de Colisiones (Multiple Access With Collision Avoidance)
MACAW	Acceso Múltiple con Prevención de Colisiones para Wireless (Multiple Access with Collision Avoidance for Wireless)
MCPS-SAP	Parte Común de la Sub Capa MAC (MAC Common Part Sublayer Service Access Point)
MEER	Ministerio de Electricidad y Energía Renovable
MFR	Pie de trama MAC (MAC footer)
MHR	Encabezado MAC (MAC Header)
MLME-SAP	Punto de Acceso a Servicios de la Entidad de Gestión de la Subcapa Capa MAC (MAC sublayer management entity service access point)
MSDU	Unidad de Servicio de Datos MAC (MAC service data unit)
MSN	Nodo inteligente Móvil (Mobile Smart Node)
NACK	No Acuse de recibo (No Acknowledgment)
NAV	Vector de Asignación de Red (Network Allocation Vector)
NXP	Next eXPerience semiconductors
OLADE	Organización Latinoamericana de Energía

OLED	Organic Light Emitting Diode
PAN	Red de Área Personal (Personal Area Network)
PCF	Función de Coordinación Puntual (Point Coordination Function)
PDA	Asistente Digital Personal (Personal Digital Assistant)
PDU	Protocolo de Unidad de Datos Protocol Data Unit
PHR	Cabecera de la Capa Física (Physical Header)
PHY	Capa física (Physical Layer)
PIFS	Espacio inter- trama PCF (PCF Interframe Spacing)
PPDU	Unidad de Datos de Protocolo de La Capa Física (Physical Protocol Data Unit)
QoS	Calidad de Servicio (Quality of Service)
RCP	Llamada a Procedimiento Remoto ()
RFD	Dispositivo de Funciones Reducidas (Reduce Function Device).
RTS	Solicitud de Envío (Request To Send)
SAP	Servicio de Punto de Acceso (Service Access Point)
SHR	Cabecera de Sincronización (Synchronization Header)
SIFS	Espacio inter-trama corto (Short Interframe Spacing)
SSCS	Sub Capa de Convergencia de Servicios Específicos (Service-Specific Convergence Sublayer)
TCP	Protocolo de Control de Transmisión (Transmission Control Protocol)
TESN	Nodo inteligente embebido ligero (Thin Embedded Smart Node)
UART	Transmisor-Receptor Asíncrono Universal (Universal Asynchronous Receiver-Transmitter)
UDP	Protocolo de Datagrama de Usuario (User Datagram Protocol)
USB	Bus Universal en Serie (Universal Serial Bus)
VauST	Instituto de Sistemas y Tecnologías Distribuidos y Autónomos (Verteilte autonome Systeme und Technologien)

VIC	Vector Controlador de Interrupciones (Vector Interrupt Controller)
WISE	Pila IP Versátil para Sistemas Embebidos (Versatile IP Stack for Embedded Systems)
WLAN	Red Inalámbrica de Área Local (Wireless Local Area Network)
WMAN	Red Inalámbrica de Área Metropolitana (Wireless Metropolitan Area Networks)
WPAN	Red Inalámbrica de Área Personal (Wireless Personal Area Networks)
WWAN	Red Inalámbrica de Área Extensa Wireless (Wide Area Network)
XML	Lenguaje de marcado extensible (Extensible Markup Language)
XML-RPC	Llamada de procedimiento remoto XML (XML-Remote Procedure Call)

ANEXOS

Anexo 1

Características del microcontrolador LPC2148

<ul style="list-style-type: none"> • Microcontrolador ARM7TDMI-S de 16/32 bits en un paquete pequeño LQFP64. • De 8 a 40 Kb de memoria RAM estática en el chip y de 32 a 512 Kb de memoria flash de programa. • Ancho de 128 bits Interfaz / acelerador que permite la operación de alta velocidad a 60 MHz • Programación (ISP/IAP) a través del software gestor de arranque en el chip. Borrado completo del chip en 400 ms y programación de 256 bytes de 1 ms. • USB 2.0 de alta velocidad del controlador de dispositivos con 2 KB de memoria RAM. Además ofrece 8 Kb de RAM en el chip de acceso a USB por DMA. • Dos canales conversores A / D de 10 bits que proporcionan un total de 14 entradas analógicas, con tiempos de conversión de 2.44 us por canal. • Dos temporizadores de 32 bits, unidad PWM de seis salidas y watchdog. 	<ul style="list-style-type: none"> • Múltiples interfaces serie, incluyendo dos UART (16C550) y dos Fast I2C-bus (400 kbit / s), SPI y SSP con amortiguación variable. • Control de vector de interrupciones con direcciones de vector y prioridades configurables. • De 5 V a 45 V en pines E/ S en un paquete pequeño LQFP64. • Oscilador integrado en el chip que funciona con un cristal externo en el rango de 1 MHz a 30 MHz y con un oscilador externo de hasta 50 MHz • Modos de ahorro de energía que incluyen inactivo y Apagado. Activación / desactivación a criterio de las funciones periféricas, así como la ampliación del reloj periférico para optimización de la energía. • Procesador de despertador desde el modo de energía hacia abajo a través de interrupción externa, USB, Brown-Out Detector (DBO) y reloj de tiempo real (RTC). • Suministro de energía con circuitos de Power-On Reset (POR) y DBO. • Rango de voltaje de operación de 3,0 V a 3,6 V (3,3 V \pm 10%).
--	---

Tabla de los sectores de memoria flash en los microcontroladores de la familia LPC 2000

Sector Number	Sector Size [kB]	Address Range	LPC2141 (32kB)	LPC2142 (64kB)	LPC2144 (128kB)	LPC2146 (256kB)	LPC2148 (512kB)
0	4	0X0000 0000 - 0X0000 0FFF	+	+	+	+	+
1	4	0X0000 1000 - 0X0000 1FFF	+	+	+	+	+
2	4	0X0000 2000 - 0X0000 2FFF	+	+	+	+	+
3	4	0X0000 3000 - 0X0000 3FFF	+	+	+	+	+
4	4	0X0000 4000 - 0X0000 4FFF	+	+	+	+	+
5	4	0X0000 5000 - 0X0000 5FFF	+	+	+	+	+
6	4	0X0000 6000 - 0X0000 6FFF	+	+	+	+	+
7	4	0X0000 7000 - 0X0000 7FFF	+	+	+	+	+
8	32	0x0000 8000 - 0X0000 FFFF		+	+	+	+
9	32	0x0001 0000 - 0X0001 7FFF			+	+	+
10 (0x0A)	32	0x0001 8000 - 0X0001 FFFF			+	+	+
11 (0x0B)	32	0x0002 0000 - 0X0002 7FFF				+	+
12 (0x0C)	32	0x0002 8000 - 0X0002 FFFF				+	+
13 (0x0D)	32	0x0003 0000 - 0X0003 7FFF				+	+
14 (0X0E)	32	0x0003 8000 - 0X0003 FFFF				+	+
15 (0x0F)	32	0x0004 0000 - 0X0004 7FFF					+
16 (0x10)	32	0x0004 8000 - 0X0004 FFFF					+
17 (0x11)	32	0x0005 0000 - 0X0005 7FFF					+
18 (0x12)	32	0x0005 8000 - 0X0005 FFFF					+
19 (0x13)	32	0x0006 0000 - 0X0006 7FFF					+
20 (0x14)	32	0x0006 8000 - 0X0006 FFFF					+
21 (0x15)	32	0x0007 0000 - 0X0007 7FFF					+
22 (0x16)	4	0x0007 8000 - 0X0007 8FFF					+
23 (0x17)	4	0x0007 9000 - 0X0007 9FFF					+
24 (0x18)	4	0x0007 A000 - 0X0007 AFFF					+
25 (0x19)	4	0x0007 B000 - 0X0007 BFFF					+
26 (0x1A)	4	0x0007 C000 - 0X0007 CFFF					+

Fuente: Philips Electronics (2005). LPC214x User Manual. Página 296

Anexo 2

Lista de componentes de la placa del sistema MSN

Cantidad	Descripción	Distribuidor
1	Microcontrolador LPC2148FBD64	NXP
1	MAX1811	Maxim
1	TPS73633DBVTG4	Texas Instruments
1	MCP130T-reset-component	Microchip
1	Quarz 12 MHz-LF A158D	Rakon
1	Quarz 32,768 KHz-LF A103S	Rakon
1	LED 3mm low-current, rojo	Kingbright
1	Plug connecter 10-pins 1,27mm	TYCO
1	Mini-switch	C&K Components
1	5-way joystick-TPA511G	C&K Components
1	mini-USB-slot	Molex
1	MMC/SD-card-slot	Multicomp
1	FFC-plug-in connector, 16-pins	Tyco
1	Modulo de radio con antena EASYBEE	RF Solutions
1	buzzer-PKLCS1212E4001-R1	Murata
1	BC807-40	-
2	Capacitor cerámico 18pF	AVX
4	Capacitor cerámico 39pF	AVX
6	Capacitor cerámico 100nF	Phycomp
1	Capacitor cerámico 1_ F	Phycomp
2	Capacitor cerámico 4,7_ F	Phycomp
1	Resistencia 100kΩ	Vishay
21	Resistencia 10kΩ	Vishay
1	Resistencia 56kΩ	Vishay
1	Resistencia 24kΩ	Vishay

Cantidad	Descripción	Distribuidor
1	Resistencia 330 Ω	Vishay
2	Resistencia 1,5k Ω	Vishay
2	Resistencia 33 Ω	Vishay
1	Resistencia 2k Ω	Vishay
1	Resistencia 1k Ω	Vishay
3	Resistencia 10 Ω	Vishay
1	plug connector 2-pins, 2.54mm	Molex

Anexo 3

Características del CC2420



<ul style="list-style-type: none"> • Transceiver RF de 2400–2483.5 MHz • Transceiver Direct Sequence Spread Spectrum (DSSS) • Velocidad de datos de 250 Kbps, 2 MChip/s de velocidad de chip • Modulación O-QPSK • Muy bajo consumo de corriente (RX: 18.8 mA, TX: 17.4 mA) • Alta sensibilidad (-95 dBm) • Alto rechazo de canal adyacente (30/45 dB) • Alto rechazo de canal alternativo (53/54 dB) • On-chip VCO, LNA y PA • Bajo suministro de voltaje(2.1 – 3.6 V) con regulador de voltaje en el chip • Potencia de salida programable • I/Q bajo –IF receptor flexible • Conversión directa I/Q • Transmisión y recepción separada FIFOs • Transmisión de datos FIFO de 128 byte • Recepción de datos FIFO de 128 byte • Muy pocos componentes externos • No necesita filtros externos • Interfaz de configuración fácil • Interfaz SPI de 4 hilos • Reloj de serie de hasta 10 MHz • Soporte de hardware MAC 802.15.4 	<ul style="list-style-type: none"> • cálculo de CRC-16 y control sobre la carga útil de las tramas MAC • Evaluación de Clear Channel • Detección de energía /digital RSSI • Enlace de indicación de calidad • Seguridad MAC completamente automática (CTR, CBC-MAC, CCM) • Operaciones automatizadas de seguridad para recibir y transmitir FIFOs • Modo de encriptación /des encriptación CTR • Autenticación CBC-MAC • Encriptación / des encriptación CCM y autenticación • Cifrado independiente AES • Software fácil de usar para generar los datos de configuración CC2420 • Tamaño pequeño, paquete QLP-48, de 7 x 7 mm • Cumple con la norma EN 300 328, EN 300
---	--

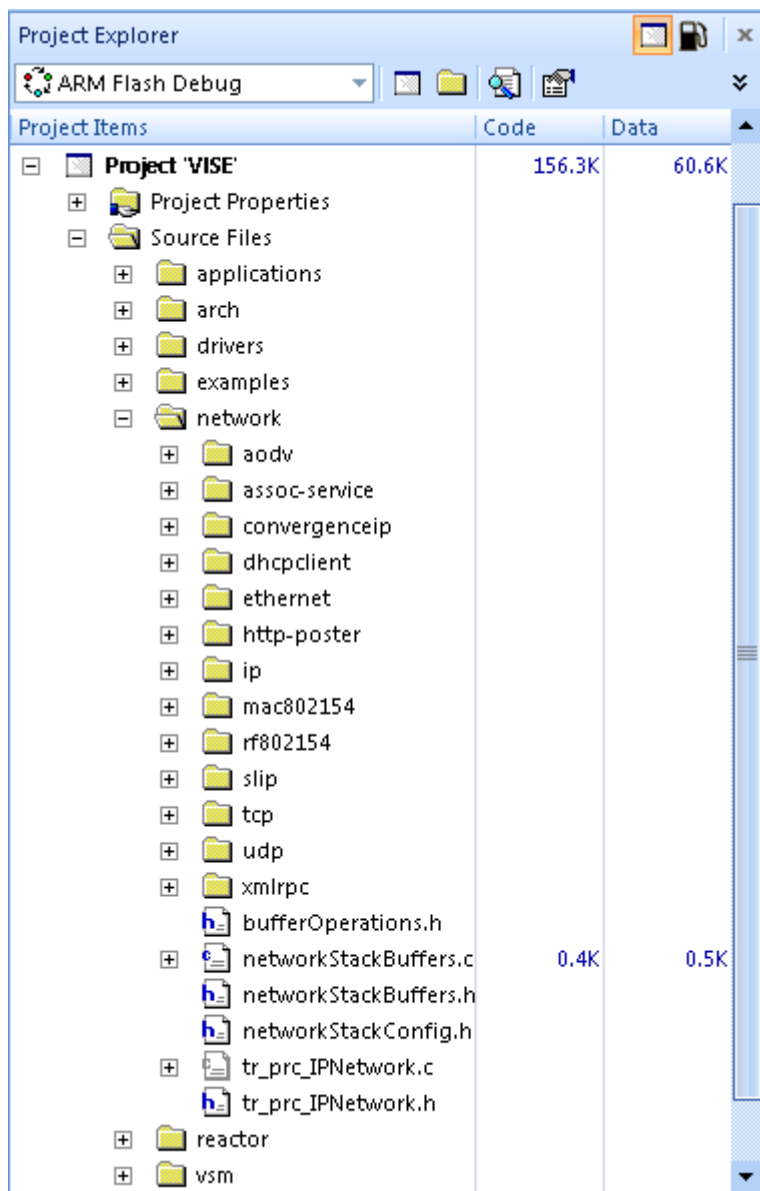
Anexo 4

Protocolos de internet usados por la pila VISE

Nombre	Versión	Descripción
IEEE 802.15.4		Capa física y MAC construidas bajo el Chipcon CC2420
Ethernet driver	1.00	compatible con el cristal CS8900A
ARP	1.01	ARP compatible para IPv4, para las plataformas de recursos limitados ARP se puede utilizar en modo pasivo.
IP	1.01	Compatible con IPv4. La implementación actual no soporta fragmentación IP
ICMP	1.00	ICMP soporta IPv4 ICMP
UDP	1.01	Compatible con el sistema, se encuentran activados diferentes puertos UDP que pueden escuchar un número arbitrario de procesos.
TCP	1.00	Una primera aplicación compacta TCP es parte de VISE. Se encuentran activados los puertos TCP. La retransmisión de TCP es apoyada por medio de la capa de aplicación.
HTTP	1.00	Un pequeño servicio HTTP es parte de VISE. El servicio se presta para las distintas páginas HTTP.
DHCP	0.9	Cliente DHCP en el puerto 68
DHCP	0.9	Servidor DHCP en el puerto 69

Anexo 5

Visualización con la herramienta Project Explorer de todos los ficheros y archivos que forman el proyecto VISE



Anexo 6

Código fuente del archivo de cabecera **mac_Protocol_RTS_CTS.h** para el dispositivo **MSN**.

```

/**
 * @author Susana Mafla
 */
#ifndef __mac_Protocol_RTS_CTS_h__ //Se define el archivo de cabecera mac_Protocol_RTS_CTS
    // para poder compartir variables y funciones con otras aplicaciones

#define __mac_Protocol_RTS_CTS_h__ //dentro del programa

#include <targets/lpc2148.h> //se incluye la librería targets/lpc2148.h que contiene las
    //configuraciones específicas para el microcontrolador IPC2148

#include"D:\Susy\Quellcode_SmartNodeInferenz\VISERowley\source\reactor\reactor.h"
    //se incluye el archivo de cabecera para reactor.h donde se inicializan
    //las funciones del reactor

#include"D:\Susy\Quellcode_SmartNodeInferenz\VISERowley\source\network\udp\SAP_udp.h"
    //se incluye el archivo de cabecera para para definir el protocolo UDP

#include"D:\Susy\Quellcode_SmartNodeInferenz\VISERowley\source\network\ip\SAP_ip.h"
    //se incluye el archivo de cabecera para para definir la capa IP

#include"D:\Susy\Quellcode_SmartNodeInferenz\VISERowley\source\network\mac802154\ip2mac802154.
h"
    //se incluye el archivo de cabecera para para definir el paso de capa IP a capa MAC

void mac_Protocol_init(); //se llama a la función mac_Protocol_init
    //para asignar y registrar un puerto para la aplicación

void mac_Protocol_udpCallback(BUFFER_DESCRIPTOR *desc, uint32 source_ip);
    //se llama a la función mac_Protocol_udpCallback
    //para visualizar en pantalla los datos recibidos

void mac_Protocol_reactor_port(reactorMessage *msg); //se llama a la función
mac_Protocol_reactor_port para
    //transmitir información a través del puerto y
    //del buffer de transmisión asignados

#endif //fin del archivo de cabecera mac_Protocol_RTS_CTS

```



```

if(ip != 0x00000000) mac_Protocol_remoteip = ip;    //si el parámetro IP es diferente de 0 se copia su
// valor a la variable mac_Protocol_remoteip

mac_Protocol_port = reactor_register_messageport("mac_Protocol", *mac_Protocol_reactor_port);

                //a la variable mac_Protocol_port se le asigna un puerto

                //que es designado por la función reactor_register_messageport

if(mac_Protocol_port == 0)    //si el resultado de la función reactor_register_messageport es 0
{
puts("mac_Protocol: No reactor port free!"); //se imprime un mensaje informando que no existe un

                //puerto libre para ser asignado

return;
}

udp_tx_register_process(mac_Protocol_port, mac_Protocol_JOY_ID);

                //se llama a la función udp_tx_register_process

                //que registra el puerto con su señal de identificación

if (SAP_udp_register_listener(mac_Protocol_UDPPORT, mac_Protocol_udpCallback) == -1)
{
                //si el resultado de la función

                //SAP_udp_register_listener que registra a

                //la estación receptora es igual a -1

printf("Can't register callback for UDP port %i\n", mac_Protocol_UDPPORT);

                //se imprime un mensaje indicando que

                //no se puede registrar la devolución de llamada

                //para el puerto que contiene la variable

                //mac_Protocol_UDPPORT

}

}

void mac_Protocol_udpCallback(BUFFER_DESCRIPTOR *desc, uint32 source_ip)

                //se define a la función mac_Protocol_udpCallback

                //que muestra en pantalla los datos recibidos

                //por el dispositivo.

{

```

```

int bandera; //se declara una variable bandera para guardar
//en ella los datos recibidos

uint16 size_mpdu; // se declara la variable size_mpdu para guardar el
// tamaño de una trama

int nav ; //se declara la variable nav para utilizarla
//en el temporizador NAV (Vector de Asignación de Red)

int Tx_ack = 320; // Tiempo de transmisión de una trama ack en us, valor
// definido en la tabla 3.2, página 81 del presente documento

int T_propagation = 0.0667; // Tiempo de propagación total (tiempo de propagación de
// los datos + el tiempo de propagación del ack), valor
// definido en la tabla 3.2, página 81 del presente documento

int contador=0; // se declara la variable contador

printf("Mensaje UDP: recibido \"%s\" desde 0x%X\n", desc->data, source_ip);
//se imprime un mensaje que indica los datos recibidos
//y la dirección del transmisor

char *ver; //se declara el puntero ver tipo char

bandera = desc->data; //a bandera se le asigna el contenido de data que es
//apuntado por el puntero desc

ver=strchr(bandera,'CTS'); //al puntero ver se le asigna el resultado de la instrucción
//strchr que busca dentro de la variable bandera la cadena
// CTS, si la encuentra devuelve en el puntero ver su contenido

if(ver >= 1073756000) //se comprueba si ver es igual o mayor que 1073756000
//que representa en números enteros el valor ASCII de la cadena CTS
//Esto con la finalidad de comprobar si se ha recibido un CTS

{
if (envio_rts== 1) // si además de recibir un CTS, el dispositivo envió una
// solicitud RTS

{
BUFFER_DESCRIPTOR *reply; //se declara un puntero reply de tipo BUFFER_DESCRIPTOR

```



```

reply = SAP_udp_request_TXBuffer(sizeof("DATOS")); //al puntero se le asigna el valor de la función
if(reply == NULL)                                //SAP_udp_request_TXBuffer, donde se asigna un buffer
                                                    //para la transmisión y si el resultado de la función es 0
{
    return;                                       // el programa retorna
}
memcpy(reply->data, "DATOS", sizeof("DATOS"));

                                                    //caso contrario en la posición de data apuntada por
                                                    //el puntero reply se copia "DATOS" con el tamaño
                                                    //que tiene "DATOS"

    SAP_udp_request_transmit(reply->id, mac_Protocol_remoteip, mac_Protocol_UDPPORT,
mac_Protocol_UDPPORT);

                                                    //Con la función SAP_udp_request_transmit se transmite
                                                    //"DATOS" la función tiene como parámetros la
                                                    //identificación del buffer de transmisión, la IP de destino,
                                                    //un puerto de origen y un puerto de destino
for (int16 sec=0; sec<200; sec++) // con un ciclo for y la función delay_usec
    // se crea un retraso de 4000 us para que el buffer
    // de transmisión no se desborde. Este tiempo representa
    // el tamaño intertramas DIFS y su valor se determinó
    // en base a pruebas empíricas
{
    delay_usec(200);
}
ver= -1;                                         //Se cambia el valor de la variable ver para resetearla
bandera = -1;                                   //Se cambia el valor de la variable bandera para resetearlo
envio_rts= 0;                                   //se cambia el valor de la variable envio_rts para resetearlo
}
else                                             // si se ha recibido un CTS pero el dispositivo no ha enviado
                                                // una solicitud de envío RTS Se activa el temporizador NAV

```

```

{
    size_mpdu = desc->length;    // Se copia en la variable size_mpdu el valor del tamaño de
                                // la trama recibida, utilizando el puntero desc

    size_mpdu = ((size_mpdu*8*1000)/125); // El tiempo de transmisión de una trama es igual a
                                // tamaño de trama/ancho de banda. Al tamaño de la trama se lo
                                // multiplica por 8 para convertir bytes en bits y se divide
                                // para 125000 (bits/s) que es el ancho de banda del canal,
                                // finalmente se multiplica el resultado por 1000000 para obtener
                                // el resultado en us

    nav = size_mpdu + Tx_ack + T_propagation; // el tiempo total de transmisión de datos es igual a
                                // el tiempo de transmisión del mpdu + tiempo de transmisión
                                // del ack + tiempo de propagación total

    reactor_lock();             // se bloquea al reactor para que el dispositivo no pueda
                                // realizar una transmisión y provocar una colisión

    delay_usec(nav);           // se crea un retardo igual al tiempo de transmisión de
                                // un mpdu

    reactor_unlock();          // Transcurrido el tiempo se desbloquea al reactor

    nav = 0;                   // se borra el contenido de nav
}
}

else                            // si no se ha recibido un CTS
{
    delay_usec(100);           // se espera 100 us

    for (contador; contador<10; contador++); // y se realiza un bucle para intentar hasta 10 veces
                                // la retransmisión de los datos

    { mac_Protocol_udpCallback(desc, source_ip); // se vuelve al inicio de la transmisión
        }

    contador = 0;              // se encera nuevamente el contador
}

```

```

if(ver == 'NACK')                                // si se ha recibido un NACK se reenvía los datos
{
    BUFFER_DESCRIPTOR *reply;                    //se declara un puntero reply de tipo BUFFER_DESCRIPTOR
    reply = SAP_udp_request_TXBuffer(sizeof("DATOS")); //al puntero se le asigna el valor de la función
    if(reply == NULL)                            //SAP_udp_request_TXBuffer, donde se asigna un buffer
                                                //para la transmisión y si el resultado de la función es 0
        { return;                               // el programa retorna
          }
    memcpy(reply->data, "DATOS", sizeof("DATOS"));
                                                //caso contrario en la posición de data apuntada por
                                                //el puntero reply se copia "DATOS" con el tamaño
                                                //que tiene "DATOS"
    SAP_udp_request_transmit(reply->id, mac_Protocol_remoteip, mac_Protocol_UDPPORT,
    mac_Protocol_UDPPORT);
                                                //Con la función SAP_udp_request_transmit se transmite
                                                //"DATOS". la función tiene como parámetros la
                                                //identificación del buffer de transmisión, la IP de destino,
                                                //un puerto de origen y un puerto de destino
    for (int16 sec=0; sec<200; sec++) // con un ciclo for y la función delay_usec
                                                // se crea un retazo de 4000 micro-segundos
        {
            delay_usec(200);
        }
}
else                                             // si no llega un NACK
{
    delay_usec(100);                            // se espera 100 us
    for (contador; contador<10; contador++); // y se realiza un bucle para intentar hasta 10 veces
                                                // la retrasmisión de los datos
    {

```

```

        mac_Protocol_udpCallback(desc, source_ip);

        // Vuelve al inicio de la transmisión hasta que llegue un ACK o NACK
    }

    contador = 0;
}

}

void mac_Protocol_reactor_port(reactorMessage *msg) //se define la función mac_Protocol_reactor_port
{
    BUFFER_DESCRIPTOR *reply;    //se declara un puntero reply de tipo BUFFER_DESCRIPTOR
    int delay = 0;                //éste último es el PDU (Protocol Data Unit) del sistema MSN
    switch(msg->messageld)        //se analiza el valor de messageld apuntado por el puntero msg
    {
        case mac_Protocol_JOY_ID:    //si el valor de messageld es igual a mac_Protocol_JOY_ID
            switch( *((char*)msg->payload) ) //se analiza el contenido del Payload apuntado por msg
            {
                case 'U':                // si el contenido es 'U'
                    envio_rts= 1;        // se modifica el valor de la variable envio_rts, para comprobar
                                            // que el dispositivo envió un RTS

                    reply = SAP_udp_request_TXBuffer(sizeof("RTS")); //a reply se le asigna el valor que retorna la
                                            //función SAP_udp_request_TXBuffer que hace
                                            //una solicitud al buffer de transmisión

                    if(reply == NULL)      //si el resultado de la función es 0
                    {
                        return;           //el programa retorna
                    }

                    memcpy(reply->data, "RTS", sizeof("RTS")); //si es diferente de 0 a data apuntado por reply
                                            //se copia "RTS", con el tamaño de "RTS"

                    break;

                case 'S':                // si el contenido es 'S'

```

```

reply = SAP_udp_request_TXBuffer(sizeof("SELECT")); //a reply se le asigna el valor que retorna la
//función SAP_udp_request_TXBuffer que hace
//una solicitud al buffer de transmisión
if(reply == NULL) //si el resultado de la función es 0
{
return; //el programa retorna
}
memcpy(reply->data, "SELECT", sizeof("SELECT")); //si es diferente de 0 a data apuntado por reply
//se copia "SELECT", con el tamaño de "SELECT"
break;
}
delay=rand(); // se asigna a delay un número aleatorio generado
// con la instrucción rand()
delay_usec(delay); // con la función delay_usec se crea un retraso en
// us definido por el número aleatorio para evitar
// transmisiones simultáneas
SAP_udp_request_transmit(reply->id, mac_Protocol_remoteip, mac_Protocol_UDPPORT,
mac_Protocol_UDPPORT);
// se hace una solicitud para transmitir con la
// función SAP_udp_request_transmit
delay =0; // se cambia el contenido de la variable delay
// para borrar el número aleatorio generado
break;

default:
puts("mac_Protocol: Unknown message id."); // si no se reconoce el mensaje que se recibió
// se imprime un mensaje indicando esto

return;
break;

```

}

}

Código fuente del archivo de cabecera **mac_Protocol_RTS_CTS.h** para el dispositivo **FESN**.

```

/**
 * @author Susana Mafla
 */

#ifndef __mac_Protocol_RST_CTS_h__
    //se define el archivo de cabecera mac_Protocol_RTS_CTS para compartir
#define __mac_Protocol_RST_CTS_h__
    //variables y funciones con otras aplicaciones dentro del programa
#include <targets/lpc2148.h> //se incluye la librería targets/lpc2148.h que contiene las
    //configuraciones específicas para el microcontrolador IPC2148

#include"C:\Documents and Settings\pc\Mis
documentos\UTN\Quellcode_SmartNodeInferenz\VISERowley\source\reactor\reactor.h"
    //se incluye el archivo de cabecera reactor.h donde se inicializan las funciones del reactor

#include"C:\Documents and Settings\pc\Mis
documentos\UTN\Quellcode_SmartNodeInferenz\VISERowley\source\network\udp\SAP_udp.h"
    //se incluye el archivo de cabecera SAP_udp.h para definir el protocolo UDP

#include"C:\Documents and Settings\pc\Mis
documentos\UTN\Quellcode_SmartNodeInferenz\VISERowley\source\network\ip\SAP_ip.h"
    //se incluye el archivo de cabecera para SAP_ip.h para definir la capa IP

#include"C:\Documents and Settings\pc\Mis
documentos\UTN\Quellcode_SmartNodeInferenz\VISERowley\source\network\mac802154\SAP_ip2mac80
2154.h" //se incluye el archivo de cabecera para SAP_ip2mac802154.h para definir el
    //paso de capa IP a capa MAC

void mac_Protocol_init();//se llama a la función mac_Protocol_init para asignar y registrar un puerto
    //para la aplicación

void mac_Protocol_udpCallback(BUFFER_DESCRIPTOR *desc, uint32 source_ip);
    //se llama a la función mac_Protocol_udpCallback para visualizar en pantalla
    //los datos recibidos

void mac_Protocol_reactor_port(reactorMessage *msg);
    //se llama a la función mac_Protocol_reactor_port para transmitir información
    // a través del puerto y del buffer de transmisión asignados

#endif // fin del archivo de cabecera mac_Protocol_RTS_CTS

```

Código fuente del archivo ejecutable **mac_Protocol_RTS_CTS.c** para el dispositivo **FESN**.

```

/**
 * @author Susana Mafla
 */
#include "mac_Protocol_RST_CTS.h"

        // incluye el archivo de cabecera mac_Protocol_RTS_CTS.h para su uso

#include "C:\Documents and Settings\pc\Mis
documentos\UTN\Quellcode_SmartNodeInferenz\VISERowley\source\arch\delay.h"

        // incluye el archivo de cabecera delay.h que contiene
        // los retrasos de tiempo

int mac_Protocol_port; //se declara la variable mac_Protocol_port para guardar
        //el puerto asignado a la aplicación por el reactor

int delay =0; //se declara una variable delay para los retrasos

int before_ip=0; // se declara la variable before_ip

int before_data = 0; // se declara la variable before data

#define mac_Protocol_JOY_ID 4711 //se define a la variable mac_Protocol_JOY_ID con el valor de 4711
        //para identificar al Joystick

#define mac_Protocol_UDPPORT 3000
        // se define a la variable mac_Protocol_UDPPORT con el valor de 3000
        //para utilizar el sector número 3 de la memoria flash del uC, valor de la
        //tabla del anexo 1, pág. 120 del presente documento

uint32 mac_Protocol_remoteip = 0x00000000; //se declara la variable mac_Protocol_remoteip y se
inicializa con 0

void mac_Protocol_init(uint32 ip) //se define la función mac_Protocol_init con un parámetro IP
        //de tipo entero de 32 bits. En esta función se asigna y registra un puerto para la aplicación
{
    if(ip != 0x00000000) mac_Protocol_remoteip = ip; //si el parámetro IP es diferente de 0 se copia
        //su valor a la variable mac_Protocol_remoteip

    mac_Protocol_port = reactor_register_messageport("mac_Protocol", *mac_Protocol_reactor_port);
        //a la variable mac_Protocol_port se le asigna un puerto
        //que es designado por la función reactor_register_messageport

```



```

if(mac_Protocol_port == 0)    //si el resultado de la función reactor_register_messageport es 0
{
puts("mac_Protocol: No reactor port free!");

                                //se imprime un mensaje informando que no existe un puerto
                                //libre para ser asignado

return;
}

joystick_register_process(mac_Protocol_port, mac_Protocol_JOY_ID);

                                //se llama a la función udp_tx_register_process que registra
                                //el puerto con su señal de identificación

if (SAP_udp_register_listener(mac_Protocol_UDPPORT, mac_Protocol_udpCallback) == -1) {
                                //si el resultado de la función SAP_udp_register_listener
                                //que registra a la estación receptora es igual a -1

printf("Can't register callback for UDP port %i\n", mac_Protocol_UDPPORT);

                                //se imprime un mensaje indicando que no se puede registrar
                                //la devolución de llamada para el puerto mac_Protocol_UDPPORT
}
}

void mac_Protocol_udpCallback(BUFFER_DESCRIPTOR *desc, uint32 source_ip)
{
                                //se define a la función mac_Protocol_udpCallback que
                                //imprime los datos recibidos

int bandera;                    //se declara una variable bandera para
                                //guardar en ella los datos recibidos

int ack= 0;                    //se declara una variable ack para identificar a
                                //quien enviar un ack

int size_confirm;              //se declara una variable para confirmar el
                                //tamaño de los datos enviados

uint8 next_ip = mac_Protocol_remoteip; //se declara la variable next_ip y se le asigna
                                //la ip del dispositivo que está transmitiendo

int next_data =0;              //se declara la variable next_data

char *ver;                    //se declara el puntero ver de tipo char

```

```

bandera = desc->data;           //a la variable bandera se le asigna el contenido de data
                                //que es apuntado por el puntero desc
ver=strchr(bandera,'RTS');     //al puntero ver se le asigna el resultado de la instrucción
                                //strchr que busca dentro de la variable bandera la cadena
                                //RTS, si la encuentra devuelve en el puntero ver su contenido
size_confirm = desc -> length; //a la variable size_confirm se le asigna el contenido de
                                //de el campo length, apuntado por el puntero desc
size = size_confirm;          // al puntero size se le copia el valor de size_confirm
next_data = bandera;          // a la variable next_data se le asigna los datos recibidos
if(before_ip == next_ip && before_ip == next_data)
                                //se compara si la ip anterior y la ip actual del dispositivo
                                //transmisor son iguales y si los datos recibidos son igual a los datos
                                // recibidos anteriormente para evitar la duplicación de datos
{
    printf("Duplicate data"); //si se han duplicado los datos se imprime un mensaje informándolo
}
else                             //caso contrario
{
    before_ip = next_ip;          // se actualiza el valor de la anterior ip
    before_data = next_data;      // se actualiza el valor de los datos anteriores
}
printf("Mensaje: recibido \"%s\" desde 0x%X\n", desc->data, source_ip);
                                //se imprime un mensaje que indica los datos recibidos
if(ver >= 1073756000 && ack==0) //si ver es igual o mayor que 1073756000 que
                                //representa en números enteros la cadena RTS y ack=0
{
    BUFFER_DESCRIPTOR *reply;     //se declara un puntero reply de tipo BUFFER_DESCRIPTOR
    reply = SAP_udp_request_TXBuffer(sizeof("CTS"));
                                //al puntero reply se le asigna el valor de la función
    if(reply == NULL)             //SAP_udp_request_TXBuffer, donde se asigna un buffer
                                //para la transmisión y si el resultado de la función es 0

```

```

{
    return;           //el programa retorna
}

memcpy(reply->data, "CTS", sizeof("CTS")); //caso contrario en la posición de data apuntada por
                                           //reply se copia "CTS" con el tamaño que tiene "CTS"

    SAP_udp_request_transmit(reply->id, 0xFFFFFFFF, mac_Protocol_UDPPORT,
mac_Protocol_UDPPORT);

        //Con la función SAP_udp_request_transmit se transmite "CTS",
        //como broadcast hacia toda la red.

        //la función tiene como parámetros la identificación del buffer,
        //la IP de destino que en este caso es 255 la dirección de broadcast,
        //un puerto de origen y un puerto de destino

for (int16 sec=0; sec<200; sec++) // con un ciclo for y la función delay_usec

        // se crea un retazo de 4000 micro-segundos, para que el
        // buffer de transmisión no se desborde. Este tiempo representa
        // el tamaño intertramas DIFS y su valor se determinó
        // en base a pruebas empíricas

{
    delay_usec(200);
}

ver= -1;           //Se cambia el valor de la variable ver para resetearla
bandera = -1;     //Se cambia el valor de la variable bandera para resetearlo
ack = 1;         //Se cambia el valor de la variable ack para resetearlo
}

if (size = size_confirm) //Si recibió los datos del tamaño completo de
                        //los datos

{
    BUFFER_DESCRIPTOR *reply;

    reply = SAP_udp_request_TXBuffer(sizeof("ACK"));

        // se envía un ACK con el mismo procedimiento con el que se envió el CTS

    if(reply == NULL)

    {

```

```

    return;
}

memcpy(reply->data, "ACK", sizeof("ACK"));

SAP_udp_request_transmit(reply->id, mac_Protocol_remoteip, mac_Protocol_UDPPORT,
mac_Protocol_UDPPORT);

    // El ACK se envía al dispositivo que envió los datos

    for (int16 sec=0; sec<2000; sec++ ) // con un ciclo for y la función delay_usec
    {
        // se crea un retazo de 4000 micro-segundos, para que el
        // buffer de transmisión no se desborde. Este tiempo representa
        // el tamaño intertramas DIFS y su valor se determinó
        // en base a pruebas empíricas

        delay_usec(200);
    }

bandera = -1;    //Se cambia el valor de la variable ver para reseterarla
ver = -1;       //Se cambia el valor de la variable bandera para reseterarlo
ack = 0;        //Se cambia el valor de la variable ack para reseterarlo
}
}
else            // caso contrario si no se recibió los datos completos
{
BUFFER_DESCRIPTOR *reply;

reply = SAP_udp_request_TXBuffer(sizeof("NACK"));

    // se envía un NACK con el mismo procedimiento con el que se envió el CTS

    if(reply == NULL)
    {
        return;
    }

    memcpy(reply->data, "NACK", sizeof("NACK"));

    SAP_udp_request_transmit(reply->id, mac_Protocol_remoteip, mac_Protocol_UDPPORT,
mac_Protocol_UDPPORT);

    // El NACK se envía al dispositivo que envió los datos y llegaron

```

```

// incompletos o con un error

for (int16 sec=0; sec<2000; sec++) // con un ciclo for y la función delay_usec
{
    // se crea un retazo de 4000 micro-segundos
    delay_usec(200);
}
size = 0;
}
}

void mac_Protocol_reactor_port(reactorMessage *msg) //se define la función mac_Protocol_reactor_port
{
    BUFFER_DESCRIPTOR *reply; //se declara un puntero reply de tipo BUFFER_DESCRIPTOR
                                //éste último es el PDU (Protocol Data Unit) del sistema MSN
    switch(msg->messageId) //se analiza el valor de messageId apuntado por el puntero msg
    {
        case mac_Protocol_JOY_ID: //si el valor de messageId es igual al valor de mac_Protocol_JOY_ID
            switch( *((char*)msg->payload) ) //se analiza el contenido del Payload apuntado por msg
            {
                case 'U': // si el contenido es 'U'
                    reply = SAP_udp_request_TXBuffer(sizeof("RTS"));
                                //a reply se le asigna el valor que retorna la función
                                //SAP_udp_request_TXBuffer que hace una solicitud
                                //al buffer de transmisión
                    if(reply == NULL) //si el resultado de la función es 0
                    {
                        return; //el programa retorna
                    }
                    memcpy(reply->data, "RTS", sizeof("RTS")); //si es diferente de 0 a data apuntado por reply
                                //se copia "RTS", con el tamaño de "RTS"
                    break;
                case 'S': // si el contenido es 'S'
                    reply = SAP_udp_request_TXBuffer(sizeof("SELECT"));

```

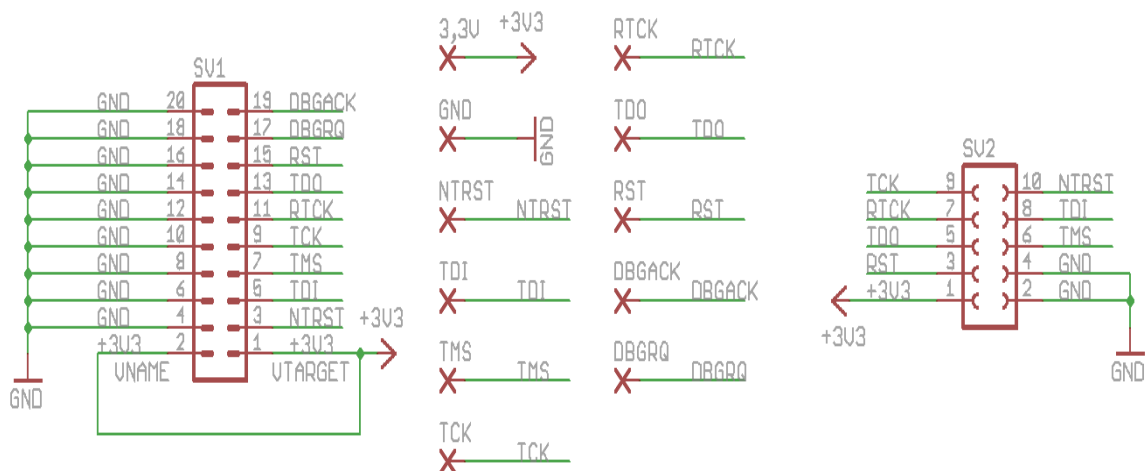
```

//a reply se le asigna el valor que retorna la función
//SAP_udp_request_TXBuffer que hace una solicitud al
//buffer de transmisión
if(reply == NULL) //si el resultado de la función es 0
{
return; //el programa retorna
}
memcpy(reply->data, "SELECT", sizeof("SELECT")); //si es diferente de 0 a data apuntado por reply
//se copia "SELECT", con el tamaño de "SELECT"
break;
}
SAP_udp_request_transmit(reply->id, mac_Protocol_remoteip, mac_Protocol_UDPPORT,
mac_Protocol_UDPPORT);
break; // se hace una solicitud para transmitir
default:
puts("mac_Protocol: Unknown message id.");
// si no se se reconoce el mensaje que se recibió se imprime un mensaje indicando esto
return;
break;
}
}

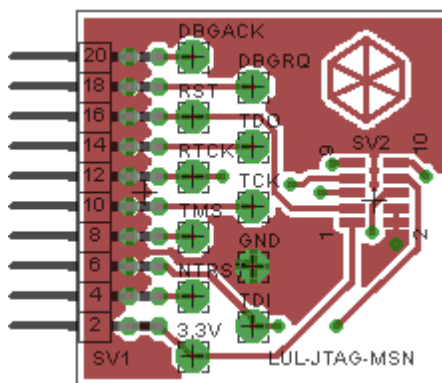
```

Anexo 7

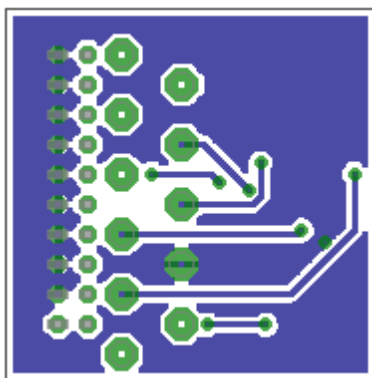
Esquema placa convertidor JTAG hembra de 20 pines a 10 pines



Placa convertidor JTAG hembra de 20 pines a 10 pines parte frontal



Placa convertidor JTAG hembra de 20 pines a 10 pines parte posterior



Anexo 8

Código fuente del archivo de cabecera **udp_continuos_tx.h** para el dispositivo **MSN**.

```

/**
 * @author Susana Mafla
 */
#ifndef __udp_continuos_tx_h__
    //definición del archivo ejecutable (archivo de cabecera .h) udp_continuos_tx_h
#define __udp_continuos_tx_h__ //definición del archivo ejecutable udp_continuos_tx_h
#include <targets/lpc2148.h>
    // incluye la librería targets/lpc2148.h con las configuraciones para el uC lpc2148
#include "D:\Susy\Quellcode_SmartNodeInferenz\VISERowley\source\reactor\reactor.h"
    // incluye el archivo ejecutable reactor.h donde se
    // inicializan las funciones del reactor

#define UP_PIN    23

#define JOYSTICK_UP    (!(IO1PIN & (1<<UP_PIN)))

    PORT_IDENTIFIER udp_tx_port; // declara la variable udp_tx_port de tipo PORT_IDENTIFIER (entero de 16
bytes)

int udp_tx_timer_id;                //declara una variable udp_tx_timer_id de tipo entero

void initudp_tx();                  //llamada la función initudp_tx()

int udp_tx_register_process(PORT_IDENTIFIER port, int SigID);
    //llamada a la función udp_tx_register_process

void udp_tx_deregister_process(PORT_IDENTIFIER port);
    //llamada a la función udp_tx_deregister_process

void udp_tx_reactor_port(reactorMessage *msg);
    //llamada a la función udp_tx_reactor_port

#endif

```


Código fuente del archivo ejecutable **udp_continuos_tx.c** para el dispositivo **MSN**.

```

/**
 * @author Susana Mafla
 */

#include "udp_continuous_tx.h" // incluye el archivo de cabecera udp_continuous_tx.h para su uso
const char up= 'U';          // declara la constante up de tipo carácter y la inicializa con el valor de u
PORT_IDENTIFIER extern_port = -1;

        // declara la variable extern_port de tipo PORT_IDENTIFIER y se inicializa con -1
PORT_IDENTIFIER extern_SigID = -1;

        // declara la variable extern_SigID de tipo PORT_IDENTIFIER y se inicializa con -1
void initudp_tx()           // define la función initudp_tx() que inicializa un puerto udp_tx
{
    udp_tx_port = reactor_register_messageport("udp_tx", *udp_tx_reactor_port);

        // asigna a la variable udp_tx_port el valor que devuelve la función
        // reactor_register_messageport la cual busca un puerto libre y devuelve
        // la posición en memoria de dicho puerto

    printf("initudp_tx\n"); //imprime el mensaje initudp_tx para indicar que se ha inicializado el puerto
    udp_tx

    if(udp_tx_port == -1)    //si el resultado de la función reactor_register_messageport es -1

        //significa que no existe un puerto libre para ser asignado

    {

        printf("udp_tx: No Reactor port free!\n");// imprime un mensaje informando que no existe un puerto
        libre

        return;

    }

    udp_tx_timer_id = reactor_set_timer( ((300* TIMER_SECONDS)/1000)+1, 99, udp_tx_port,
    SINGLE_SHOT_TIMER);

        //a la variable udp_tx_timer_id se le asigna el valor que retorna la función
        // reactor_set_timer que realiza un sondeo cada 700 ms

    if(udp_tx_timer_id == -1) //sí udp_tx_timer_id es -1 significa que los timers están ocupados

    {

```



```

}
else // sí extern_port no es igual a -1
{
    return -1; // la función retorna -1
}
}

void udp_tx_deregister_process(PORT_IDENTIFIER port)
// se declara a la función udp_tx_deregister_process
// que tiene como parámetro a la variable port
{
    if(extern_port == port) // si la variable extern_port es igual a port
    {
        extern_port = -1; // a extern_port se le asigna -1
    }
    else // caso contrario se imprime un mensaje
    {
        printf("udp_tx: Could not deregister port %i! Port was not registered.\n", port);
    }

    // indicando que no se pudo dar de baja al puerto
    // ya que el puerto no se registró
}

void udp_tx_reactor_port(reactorMessage *msg) // se declara a la función udp_tx_reactor_port
{
    static int up_state_old = 0; // se declara a la variable static int up_state_old

    udp_tx_timer_id = reactor_set_timer( ((300 * TIMER_SECONDS)/1000)+1, 99, udp_tx_port,
    SINGLE_SHOT_TIMER);

    // a la variable udp_tx_timer_id se le asigna el valor
    // que devuelve la función reactor_set_timer, ésta función
    // realiza un sondeo cada 700 ms del estado del puerto

    if(extern_port != -1) // sí extern_port no es igual a -1
    if(JOYSTICK_UP && !up_state_old) // y el estado del joystick ha cambiado
    {

```

```
reactor_send_localmessage(udp_tx_port, extern_port, extern_SigID, (char*)&up, sizeof(char));
}

// se llama a la función reactor_send_localmessage con los parámetros:
// puerto de origen, puerto de destino, dirección de mensaje,
// puntero payload y tamaño del payload
// para transmitir continuamente el carácter de la variable UP
}
```

Anexo 9

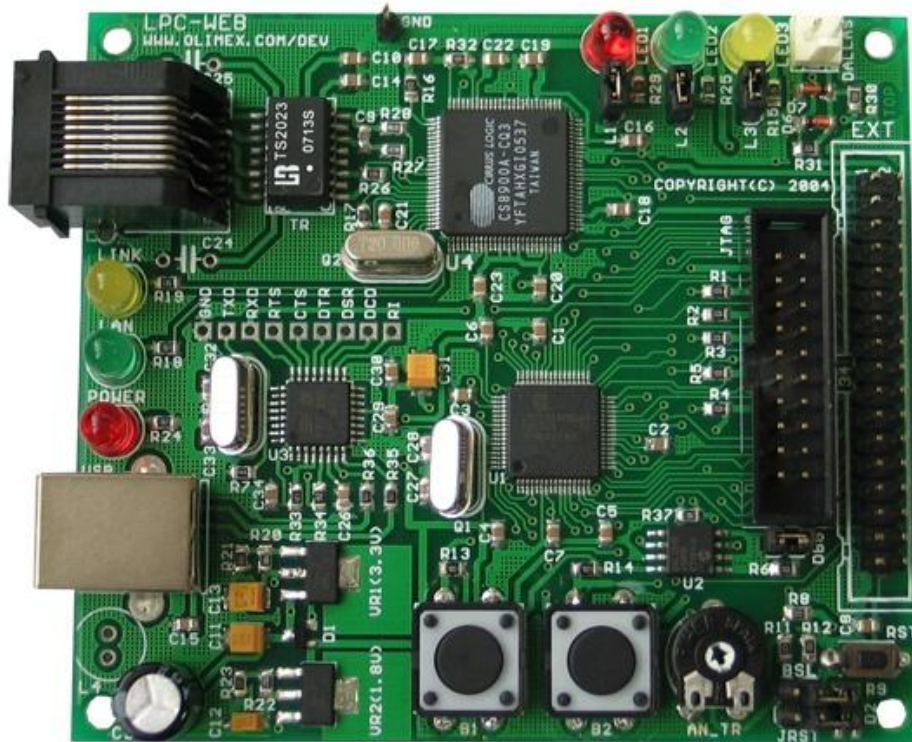
Características del Ft232BL



- | | |
|--|---|
| <ul style="list-style-type: none"> • USB <=> UART • Handshaking completo y señales de interfaz de módem • Interfaz UART compatible con 8.7 bits de datos, 1 / 2 bits de parada y Paridad n Velocidad de transferencia de 300 baudios => 3M baudios (TTL) • Velocidad de transferencia de 300 baudios => 1M baudios (RS232). • Velocidad de transferencia de 300 baudios => 3M baudios (RS422/RS485). • Buffer de RX de 384 bytes y buffer de Tx 128 bytes para transferencia alta de datos. • Tiempo de espera ajustable para buffer de TX. • Auto control de búfer transmisión para RS485. | <ul style="list-style-type: none"> • Soporte USB para suspender / reanudar a través de los pines SLEEP y RI. • Soporte para dispositivos de alta potencia, alimentación a través del pin PWREN. • Convertidor de nivel de UART y señales de control para interconectar lógica a 5V y 3.3V. • Regulador de 3.3V para USB de E / S. • Circuito integrado de Power-on-reset. • Reloj multiplicador PLL de 6MHz - 48Mhz. • Operación de prestación única de 4.35V a 5.25V. • Compatible con USB 1.1 y USB 2.0 EEPROM programable a través de USB. |
|--|---|

Anexo 10

Placa del entrenador LPC-E2124 para microcontroladores ARM7TDMI-S LPC2124



Características del entrenador LPC-E2124 para microcontroladores ARM7TDMI-S LPC2124

- | | |
|---|--|
| <ul style="list-style-type: none"> • 256K bytes de memoria de programa Flash, 16K bytes de RAM, RTC, 4x 10 bits ADC 2,44 us, 2x UART, I2C, SPI. • 2 contadores de tiempo de 32 bits, CCR 7x, 6x PWM, WDT. • Entradas y salidas tolerantes hasta 5V y 60MHz operación. • Interfaz Ethernet CS8900. • 2 pulsadores de uso general. • 1 potenciómetro conectado a la | <ul style="list-style-type: none"> • 2 reguladores de voltaje de 1,8 V y 3,3 V con un máximo de 800mA de corriente. • Requiere una sola fuente de alimentación de 5 V DC. • 3 LEDs de estado LAN. • Interfaz de USB a RS232. • Circuito RESET con control externo de ISP Philips a través de USB/RS232. • Botón RESET. |
|---|--|

<p>entrada analógica 0.</p> <ul style="list-style-type: none">• Interfaz Dallas i-button.• EEPROM 24LC515 para almacenamiento web externo.• Conector JTAG estándar para programación / depuración con ARM JTAG.	<ul style="list-style-type: none">• Puente DBG para habilitar la interfaz JTAG.• Puente JRST para activar / desactivar el control RESET externo vía RS232. <p>Cristal de 14.7456 MHz</p> <ul style="list-style-type: none">• Dimensiones de 8x9 cm.
---	--

Anexo 11

Resumen del procedimiento para la realización de las pruebas de funcionamiento

Para comprobar el funcionamiento de protocolo de capa MAC propuesto se realizan los siguientes pasos:

1. Definición de los dispositivos

Primeramente se programan los tres dispositivos para definirlos como: dos MSN y un FESN.

1.1 Conexión Física

Para definir a un dispositivo inicialmente se establece una conexión física a través de la interfaz Macraigor Wiggler entre el ordenador en donde se ejecuta el programa CrossWorks que contiene el código fuente y la placa del dispositivo MSN. La figura Anexo 12.1 muestra la conexión entre la placa del dispositivo MSN y la interfaz Macraigor Wiggler.

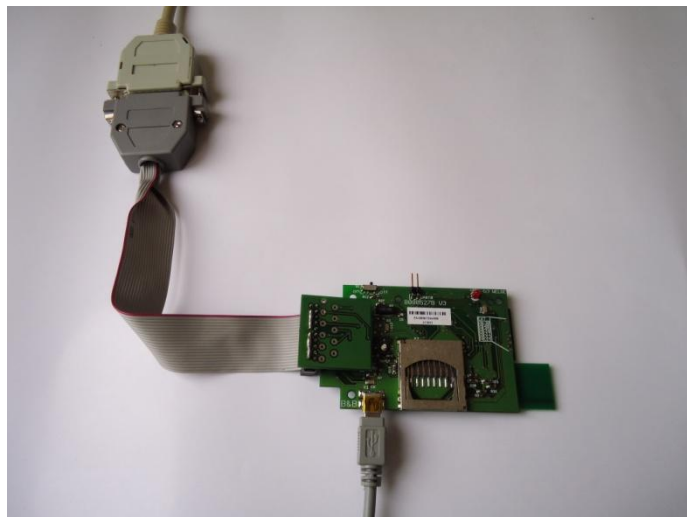


Figura Anexo 12.1: Montaje completo para la programación de las placas del sistema MSN.

1.2 Definición del dispositivo en el código del programa

Realizada la conexión el siguiente paso es establecer al dispositivo como MSN en el código del proyecto, para esto se ejecuta el programa CrossWorks y se abre el proyecto VISE. Con la ayuda de la herramienta Project Explorer del programa CrossWorks dentro del fichero Reactor se localiza y se abre el archivo ejecutable *reactorConst.h*, como se muestra en la figura Anexo 12.2.

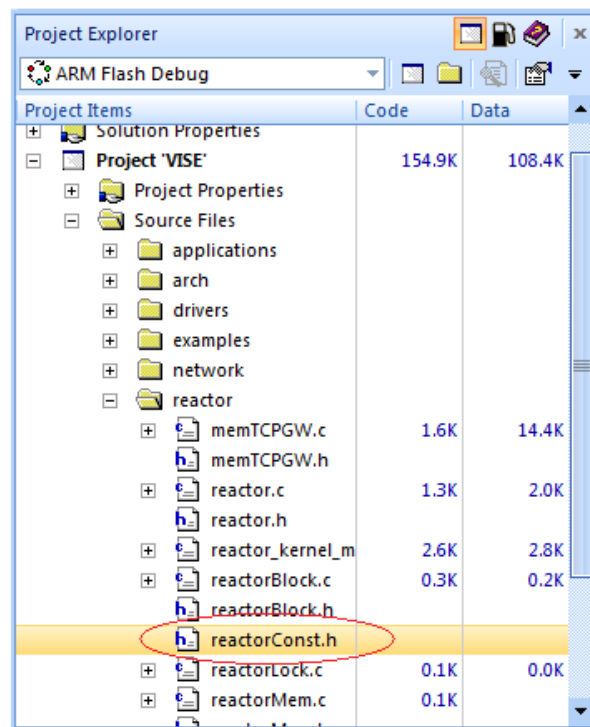
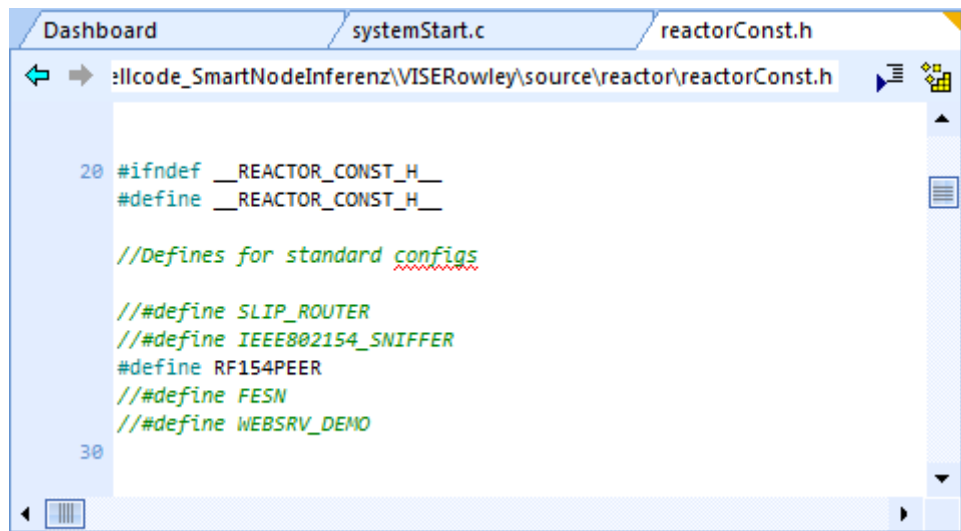


Figura Anexo 12.2: Ubicación de *reactorConst.h*,

Dentro del archivo ejecutable para definir al dispositivo como MSN se activa la instrucción `#define RF154PEER` en la línea número 27 del código, mientras que para definir al dispositivo como FESN se activa la instrucción `#define FESN` en la línea de código número 28.



```

Dashboard | systemStart.c | reactorConst.h
:llcode_SmartNodeInferenz\VISERowley\source\reactor\reactorConst.h

20 #ifndef __REACTOR_CONST_H__
   #define __REACTOR_CONST_H__

   //Defines for standard configs

   //define SLIP_ROUTER
   //define IEEE802154_SNIFFER
   #define RF154PEER
   //define FESN
   //define WEBSRV_DEMO

30

```

Figura Anexo 12.3: Definición del dispositivo como MSN

Se debe tomar en cuenta que las dos instrucciones no deben activarse a la vez porque se produce un error, ya que las funciones que efectúa el FESN dentro de la red inalámbrica es de coordinador PAN y el MSN está definido para trabajar como esclavo.

1.3 Compilar y ejecutar el programa

Luego de definir al dispositivo se compila el programa con la función Ctrl+ F7, se ejecuta con F7 y se depura para comprobar que no existe ningún error. También se puede utilizar la barra ejecutar o Build ubicada en la parte superior del programa CrossWorks

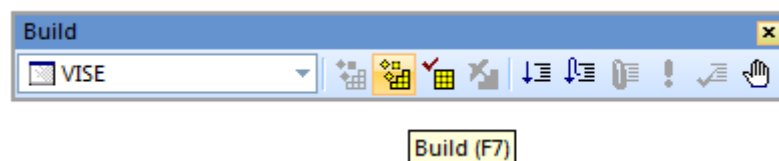


Figura Anexo12.4: Barra Build

1.4 Grabar el programa en la memoria del microcontrolador

Finalmente se graba el programa en la memoria del microcontrolador de la placa del sistema MSN con la función F5 o el ícono Start Debugging de la barra Build.

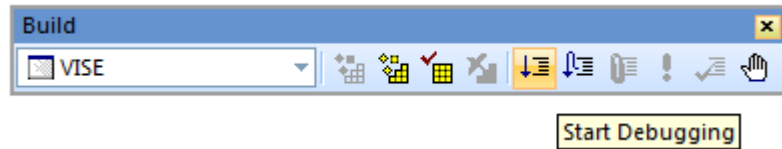


Figura Anexo 12.5: Barra Build

Todo el proceso anterior se repite para cada uno de los dispositivos del sistema MSN, asignando siempre el rol del dispositivo en la red como MSN o FESN.

2. Comunicación entre un ordenador y un dispositivo del sistema MSN

Para establecer una comunicación serial entre un computador utilizando el HyperTerminal de Windows y un dispositivo del sistema MSN, se hace que los datos recibidos por el dispositivo se retransmitan a su vez por comunicación UART hacia un conversor, en este caso el Ft232BL que se encuentra en la placa del entrenador para microcontroladores ARM7TDMI-S LPC-E2124. El Ft232BL convierte la señal de comunicación UART a USB, a su vez en el computador se instala un controlador que crea un puerto COM virtual a partir del puerto USB, con el puerto COM virtual se establece una comunicación serial a través del HyperTerminal de Windows. En la figura Anexo 12.6 se observa un diagrama que resume el proceso descrito anteriormente.

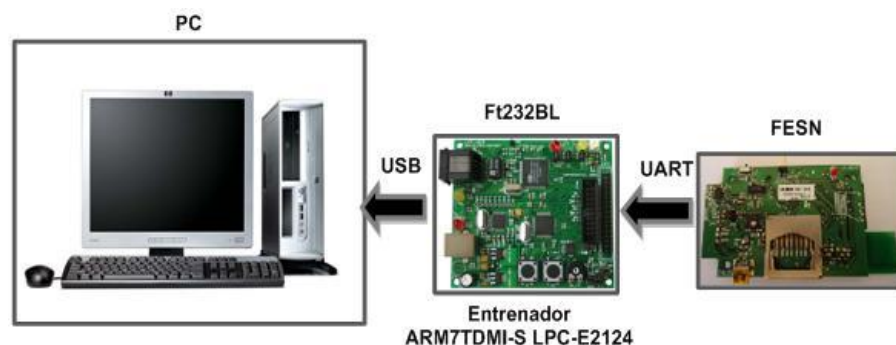


Figura Anexo 12.6: Proceso para la comunicación entre un computador y un dispositivo del sistema MSN

En la figura Anexo 12.7 se observa una foto del sistema MSN con tres dispositivos: el nodo máster o FESN y dos nodos esclavos o MSN, más el entrenador LPC-E2124 donde se localiza el conversor Ft232BL.

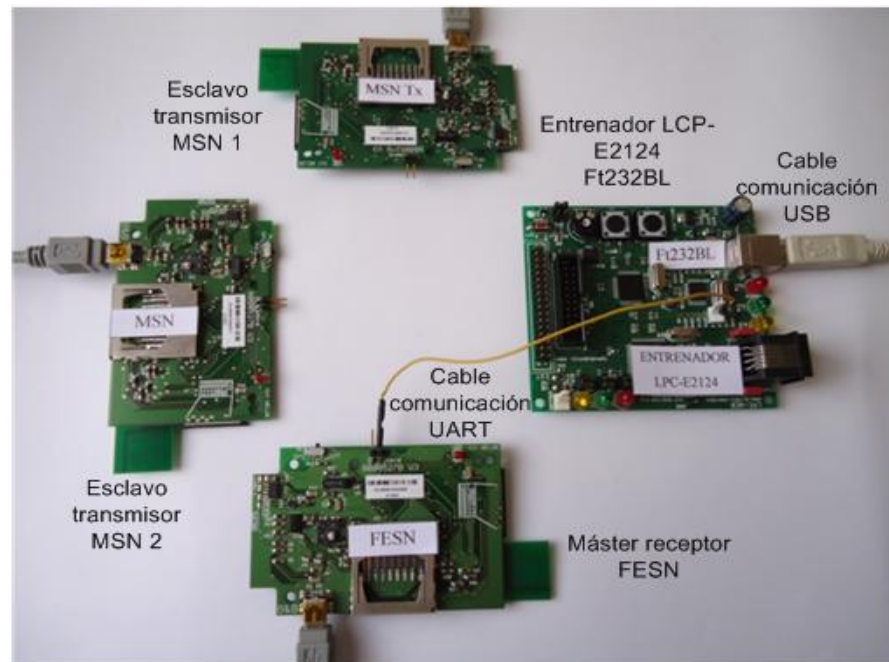


Figura Anexo 12. 7: Montaje completo del sistema MSN.

3. Configuración y utilización del HyperTerminal de Windows

Para visualizar los mensajes transmitidos entre los dispositivos del sistema MSN utilizando el HyperTerminal de Windows, éste tiene que ser configurado con ciertos parámetros que se describen a continuación:

3.1 Designación de nombre e ícono

Acceder al programa HyperTerminal y en la ventana emergente “Descripción de la conexión” se asigna un nombre y un ícono.



Figura Anexo 12.8.: Descripción de la conexión del HyperTerminal

3.2 Elección del puerto COM

En la ventana e “Conectar a” se selecciona el puerto COM asignado por el computador luego de conectar el entrenador LPC-E2124.



Figura Anexo 12.9: Selección del puerto COM para el HyperTerminal

3.3 Especificación de propiedades de conexión

Luego en la siguiente ventana “Propiedades de COM” se especifica las propiedades de la conexión, en este caso se escoge los siguientes valores:

- Bits por segundo: 57600 (valor que se define en el Main del proyecto VISE, en el archivo ejecutable systemStart.c)
- Bits de datos: 8
- Paridad: Ninguna
- Bits de parada: 1
- Control de flujo: Hardware

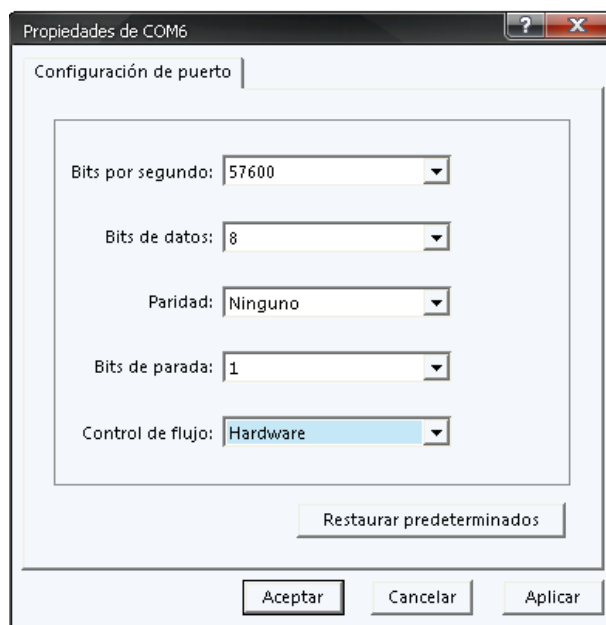
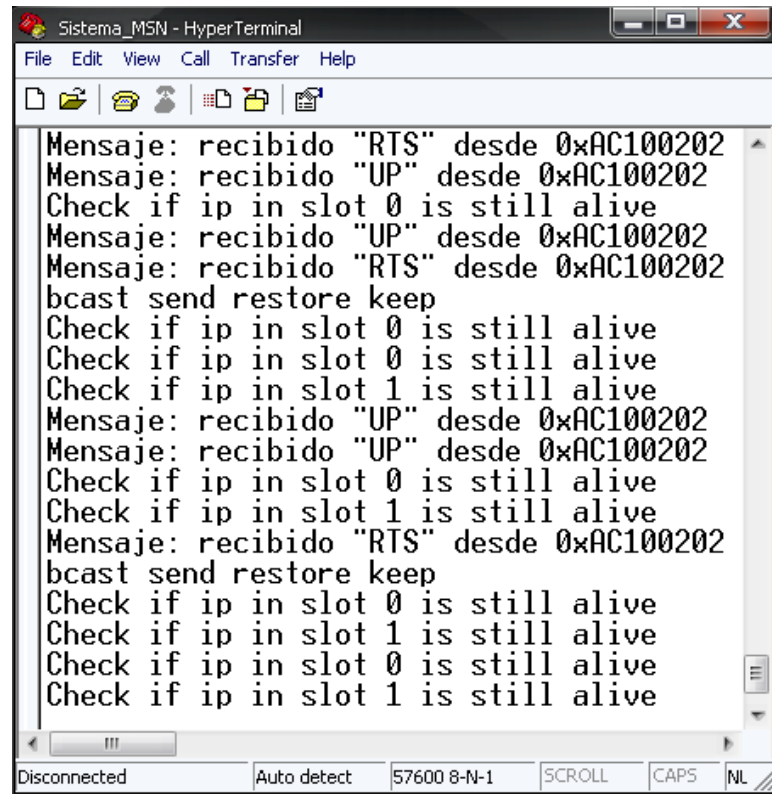


Figura Anexo 12.10: Especificación de las propiedades de conexión del HyperTerminal

Cuando aparece la ventana de sesión de HyperTerminal se enciende el dispositivo FESN y se empieza a recibir los mensajes, como se observa en la figura Anexo 12. 11.



```
Sistema_MSN - HyperTerminal
File Edit View Call Transfer Help
Mensaje: recibido "RTS" desde 0xAC100202
Mensaje: recibido "UP" desde 0xAC100202
Check if ip in slot 0 is still alive
Mensaje: recibido "UP" desde 0xAC100202
Mensaje: recibido "RTS" desde 0xAC100202
bcast send restore keep
Check if ip in slot 0 is still alive
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Mensaje: recibido "UP" desde 0xAC100202
Mensaje: recibido "UP" desde 0xAC100202
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Mensaje: recibido "RTS" desde 0xAC100202
bcast send restore keep
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Check if ip in slot 0 is still alive
Check if ip in slot 1 is still alive
Disconnected Auto detect 57600 8-N-1 SCROLL CAPS NL
```

Figura Anexo 12.11: Recepción de mensajes en el HyperTerminal