



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DESARROLLO DE UN BUS PROPIETARIO, PARA IMPLEMENTAR UNA RED EN CONFIGURACIÓN MAESTRO - MULTIESCLAVO

ING. PABLO ANDRÉS FLORES SIGÜENZA

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN INDUSTRIAL

Riobamba – Ecuador

Junio 2017

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, titulado “Desarrollo de un bus propietario, para implementar una red en configuración maestro - multiesclavo”, de responsabilidad del Ingeniero Pablo Andrés Flores Sigüenza, ha sido prolijamente revisado y se autoriza su presentación.

Dr. Fredy Proaño Ortiz PhD.

PRESIDENTE

FIRMA

Ing. Ismael Minchala Ávila PhD.

DIRECTOR DE TESIS

FIRMA

Ing. Javier Gavilanes Carrión M.Sc.

MIEMBRO DEL TRIBUNAL

FIRMA

Ing. Daniel Haro Mendoza M.Sc.

MIEMBRO DEL TRIBUNAL

FIRMA

Riobamba, junio de 2017

DERECHOS INTELECTUALES

Yo, Pablo Andrés Flores Sigüenza declaro que soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo y que el patrimonio intelectual generado por la misma pertenece a la Escuela Superior Politécnica de Chimborazo.

PABLO ANDRÉS FLORES SIGÜENZA

N^o Cédula: 060378106-3

DECLARACIÓN DE AUTENTICIDAD

Yo, Pablo Andrés Flores Sigüenza, declaro que el presente proyecto de investigación, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

PABLO ANDRÉS FLORES SIGÜENZA

N^o Cédula: 060378106-3

DEDICATORIA

A mi esposa Angélica por su apoyo incondicional en todos estos años que llevamos juntos

AGRADECIMIENTO

A mi esposa Angélica, mis padres, mi tutor, a Remigio y a Ernesto, quienes contribuyeron a que éste proyecto sea posible.

CONTENIDO

ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XIII
RESUMEN	XIV
SUMMARY	XV
1 CAPÍTULO I	- 1 -
1 INTRODUCCIÓN Y CONSIDERACIONES GENERALES	- 1 -
1.1 Introducción:	- 1 -
1.2 Problema de Investigación:	- 2 -
1.2.1 Planteamiento del Problema:	- 2 -
1.2.2 Formulación del Problema:	- 4 -
1.2.3 Sistematización del Problema:	- 5 -
1.3 Justificación de la Investigación:	- 5 -
1.4 Objetivos de la Investigación:	- 6 -
1.4.1 Objetivo General:	- 6 -
1.4.2 Objetivos Específicos:	- 6 -
1.5 Hipótesis:	- 7 -
2 CAPÍTULO II	- 8 -
2 MARCO DE REFERENCIA	- 8 -
2.1 Modelo OSI:	- 9 -
2.2 Protocolo MODBUS:	- 10 -
2.3 Transmisión por Radio Frecuencia:	- 10 -
2.3.1 Modulación Digital:	- 11 -
2.4 Domótica:	- 12 -

2.5	Protocolo X-10:	- 13 -
2.6	Protocolo ZigBee:	- 16 -
2.6.1	Capa Física (PHY):	- 17 -
2.6.2	Capa de Acceso al Medio (MAC):	- 17 -
2.6.3	Capa de Red (NWK):	- 19 -
2.6.4	Capa de Aplicación (APL):	- 20 -
2.7	Protocolo LonWorks:	- 21 -
2.7.1	Direccionamiento y Medio de Transporte:	- 22 -
2.7.2	Mensaje y Trama de Datos:	- 23 -
3	CAPÍTULO III	- 25 -
3	DISEÑO DE INVESTIGACIÓN	- 25 -
3.1	Metodología	- 25 -
3.1.1	Tipo y Diseño de Investigación:	- 25 -
3.1.2	Métodos de Investigación:.....	- 25 -
3.1.3	Enfoque de Investigación:	- 25 -
3.1.4	Alcance de la Investigación:.....	- 26 -
3.1.5	Población de Estudio:	- 26 -
3.1.6	Unidad de Análisis:	- 26 -
3.1.7	Selección de la Muestra:.....	- 26 -
3.1.8	Tamaño de la Muestra:	- 27 -
3.1.9	Técnicas de Recolección de Datos Primarios y Secundarios:	- 27 -
3.1.10	Instrumentos de Recolección de Datos Primarios y Secundarios:	- 27 -
3.1.11	Instrumentos Para Procesar Datos Recopilados:.....	- 27 -
3.2	Diseño y Desarrollo del Bus Propietario	- 27 -
3.2.1	Diseño del Bus Propietario:.....	- 27 -
3.2.2	Capa Física:	- 28 -
3.2.2.1	Características del transreceptor nRF24L01	- 29 -
3.2.2.2	Protocolo de banda base embebida nRF24L01	- 29 -
3.2.2.3	Velocidad de datos en el aire del nRF24L01	- 29 -
3.2.2.4	Canal de Frecuencia RF del nRF24L01	- 30 -
3.2.2.5	Diagrama de Bloques del nRF24L01.....	- 30 -
3.2.3	Capa Enlace:.....	- 31 -

3.2.3.1	Intercambio de Información Modo Unicast	- 31 -
3.2.3.2	Intercambio de Información Modo Broadcast	- 32 -
3.2.3.3	Trama del Mensaje	- 32 -
3.2.3.4	Direccionamiento	- 33 -
3.2.3.5	Diagrama de Estado del Maestro	- 34 -
3.2.3.6	Diagrama de Estado del Esclavo	- 37 -
3.2.3.7	Diagrama de tiempo de la comunicación maestro – esclavo.....	- 39 -
3.2.3.8	Transmisión de datos por RF.....	- 40 -
3.2.3.9	Funciones	- 40 -
3.2.3.10	Función Leer Registros.....	- 41 -
3.2.3.11	Función Escribir Registros	- 42 -
3.2.3.12	Función Eliminar Registros	- 44 -
3.2.3.13	Chequeo de Redundancia Cíclica (CRC).....	- 45 -
3.2.3.14	Algoritmo de cálculo del CRC	- 46 -
3.3	Desarrollo del Software.....	- 49 -
3.3.1	Diagrama de Flujo Microcontrolador Arduino Maestro	- 50 -
3.3.2	Diagrama de Flujo Microcontrolador Arduino Esclavo CSE.....	- 56 -
3.3.3	Diagrama de Flujo Microcontrolador Arduino Esclavo CSS.....	- 59 -
3.3.4	Diagrama de Flujo Teclado de Celular	- 62 -
3.4	Implementación del Hardware	- 64 -
3.4.1	Arquitectura de la Red.....	- 64 -
3.4.2	Funcionamiento de la Red.....	- 66 -
3.4.3	Elementos de la Red:	- 67 -
3.4.3.1	Maestro.....	- 67 -
3.4.3.2	Esclavo CSE.....	- 69 -
3.4.3.3	Esclavo CSS	- 71 -
3.4.3.4	Teclado Celular	- 74 -
3.4.4	Aplicación Domótica:.....	- 75 -
4	CAPÍTULO IV	- 81 -
4	ANÁLISIS DE RESULTADOS Y DISCUSIÓN	- 81 -
4.1	Pruebas de Comunicación:	- 82 -
4.1.1	Pruebas: Función Leer	- 82 -
4.1.2	Pruebas: Función Escribir.....	- 83 -

4.1.3	Pruebas: Función Eliminar	- 84 -
4.1.4	Análisis de Resultados:.....	- 85 -
4.2	Análisis de las Tramas de Datos y Velocidad de Comunicaciones:.....	- 86 -
4.2.1	Función Leer	- 87 -
4.2.1.1	Tiempo Real de Solicitud, Procesamiento y Respuesta	- 87 -
4.2.1.2	Tiempos Teóricos de Solicitud y Respuesta	- 88 -
4.2.2	Función Escribir	- 90 -
4.2.2.1	Tiempo Real de Solicitud, Procesamiento y Respuesta	- 90 -
4.2.2.2	Tiempos Teóricos de Solicitud y Respuesta	- 91 -
4.2.3	Función Eliminar:.....	- 93 -
4.2.3.1	Tiempo Real de Solicitud, Procesamiento y Respuesta	- 93 -
4.2.3.2	Tiempos Teóricos de Solicitud y Respuesta	- 94 -
4.2.4	Tiempo Real de Activación de Carga	- 95 -
4.3	Velocidad de Comunicaciones en Protocolos Domóticos Comerciales:	- 96 -
4.3.1	Protocolo X-10.....	- 96 -
4.3.2	Protocolo ZigBee.....	- 97 -
4.3.3	Protocolo LonWorks	- 100 -
4.4	Análisis Comparativo de Protocolos:	- 100 -
5	CAPÍTULO V.....	- 104 -
5	PROPUESTA DE TRABAJOS FUTUROS	- 104 -
	CONCLUSIONES.....	- 106 -
	RECOMENDACIONES.....	- 108 -
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE FIGURAS

<i>Figura 1-1: Formato de la trama LonWorks</i>	- 3 -
<i>Figura 1-2: Formato de la trama ZigBee</i>	- 4 -
<i>Figura 2-1: Capas del Modelo OSI</i>	- 9 -
<i>Figura 2-2: Pilares Fundamentales de la Domótica</i>	- 13 -
<i>Figura 2-3: Envío de un “1” y un “0” binario</i>	- 14 -
<i>Figura 2-4: Ciclos para transmisión completa en X-10</i>	- 15 -
<i>Figura 2-5: Capas de 802.15.4 y ZigBee</i>	- 16 -
<i>Figura 2-6: Estructura de una Supertrama</i>	- 17 -
<i>Figura 2-7: Estándar 802.15.4. Trama de comando</i>	- 18 -
<i>Figura 2-8: Estándar 802.15.4. Trama de baliza</i>	- 18 -
<i>Figura 2-9: Estándar 802.15.4. Trama de dato</i>	- 19 -
<i>Figura 2-10: Estándar 802.15.4. Trama de confirmación</i>	- 19 -
<i>Figura 2-11: Topologías de Red en ZigBee</i>	- 20 -
<i>Figura 3-1: Capas del Bus Propietario</i>	- 28 -
<i>Figura 3-2: Transreceptor nRF24L01</i>	- 28 -
<i>Figura 3-3: Diagrama de Bloques del Transreceptor nRF24L01</i>	- 30 -
<i>Figura 3-4: Trama de Datos del Protocolo Desarrollado</i>	- 31 -
<i>Figura 3-5: Modo Unicast</i>	- 32 -
<i>Figura 3-6: Modo Broadcast</i>	- 32 -
<i>Figura 3-7: Trama del Mensaje</i>	- 33 -
<i>Figura 3-8: Direccionamiento</i>	- 33 -
<i>Figura 3-9: Bloques de Estado</i>	- 34 -
<i>Figura 3-10: Diagrama de Estado del Maestro</i>	- 35 -
<i>Figura 3-11: Diagrama de Estado del Esclavo</i>	- 38 -
<i>Figura 3-12: Diagrama de tiempo comunicación maestro - esclavo</i>	- 39 -
<i>Figura 3-13: Formato de cada byte de la trama del mensaje</i>	- 40 -
<i>Figura 3-14: Solicitud del maestro (Función Leer)</i>	- 41 -
<i>Figura 3-15: Respuesta del esclavo (Función Leer)</i>	- 41 -
<i>Figura 3-16: Ejemplo Función Leer</i>	- 42 -
<i>Figura 3-17: Solicitud del maestro (Función Escribir)</i>	- 42 -
<i>Figura 3-18: Respuesta del esclavo (Función Escribir)</i>	- 43 -
<i>Figura 3-19: Ejemplo Función Escribir</i>	- 43 -
<i>Figura 3-20: Solicitud del maestro (Función Eliminar)</i>	- 44 -
<i>Figura 3-21: Respuesta del esclavo (Función Eliminar)</i>	- 44 -

<i>Figura 3-22: Ejemplo Función Eliminar.....</i>	<i>- 45 -</i>
<i>Figura 3-23: Diagrama de Flujo Algoritmo de Cálculo CRC.....</i>	<i>- 47 -</i>
<i>Figura 3-24: Arquitectura de la Red.....</i>	<i>- 64 -</i>
<i>Figura 3-25: Flujo de Información Red Centralizada.....</i>	<i>- 65 -</i>
<i>Figura 3-26: Procedimiento ejecución de tareas.....</i>	<i>- 66 -</i>
<i>Figura 3-27: Maestro.....</i>	<i>- 67 -</i>
<i>Figura 3-28: Diagrama de Bloques del Maestro</i>	<i>- 68 -</i>
<i>Figura 3-29: Esclavo CSE (Contactos Secos de Entrada)</i>	<i>- 69 -</i>
<i>Figura 3-30: Diagrama de Bloques del Esclavo CSE.....</i>	<i>- 70 -</i>
<i>Figura 3-31: Esclavo CSS (Contactos Secos de Salida).....</i>	<i>- 71 -</i>
<i>Figura 3-32: Diagrama de Bloques del Esclavo CSS.....</i>	<i>- 73 -</i>
<i>Figura 3-33: Teclado Celular - Maestro.....</i>	<i>- 74 -</i>
<i>Figura 3-34: Aplicación Domótica</i>	<i>- 75 -</i>
<i>Figura 3-35: Instalaciones de Luminarias del Área Exterior-Aplicación Domótica.....</i>	<i>- 77 -</i>
<i>Figura 3-36: Instalaciones Eléctricas Planta Baja-Aplicación Domótica</i>	<i>- 78 -</i>
<i>Figura 3-37: Instalaciones Eléctricas Buhardilla-Aplicación Domótica</i>	<i>- 79 -</i>
<i>Figura 4-1: Laboratorio de Pruebas.....</i>	<i>- 81 -</i>
<i>Figura 4-2: Tramas de datos para la función Leer: (a) Bytes enviados en la solicitud del maestro y en la respuesta del esclavo; (b) Tiempo total de la transacción</i>	<i>- 82 -</i>
<i>Figura 4-3: Tramas de datos para la función Escribir: (a) Bytes enviados en la solicitud del maestro y en la respuesta del esclavo; (b) Tiempo total de la transacción</i>	<i>- 83 -</i>
<i>Figura 4-4: Tramas de datos para la función Eliminar: (a) Bytes enviados en la solicitud del maestro y en la respuesta del esclavo; (b) Tiempo total de la transacción</i>	<i>- 84 -</i>
<i>Figura 4-5: Tiempo Real Función Leer: (a) Solicitud del maestro; (b) Procesamiento; (c) Respuesta del Esclavo</i>	<i>- 87 -</i>
<i>Figura 4-6: Tiempo Real Función Escribir: (a) Solicitud del maestro; (b) Procesamiento; (c) Respuesta del Esclavo</i>	<i>- 91 -</i>
<i>Figura 4-7: Tiempo Real Función Eliminar: (a) Solicitud del maestro; (b) Procesamiento; (c) Respuesta del Esclavo</i>	<i>- 93 -</i>

ÍNDICE DE TABLAS

Tabla 2-1: Bandas de Frecuencia	- 11 -
Tabla 2-2: Elementos de una trama del protocolo X-10	- 14 -
Tabla 2-3: Complemento del código X-10	- 15 -
Tabla 2-4: Tipos y Características de Canales en Lonworks	- 23 -
3-1: Especificaciones Técnicas Maestro.....	- 69 -
Tabla 3-2: Especificaciones Técnicas Generales del Esclavo CSE	- 71 -
Tabla 3-3: Especificaciones Técnicas Generales de las Partes del Esclavo CSS	- 73 -
Tabla 3-4: Botones Teclado Celular – Cargas Controladas	- 75 -
Tabla 3-5: Simbología de los Accesorios de la Aplicación Domótica	- 80 -
Tabla 3-6: Resumen del Comando de las Cargas de la Aplicación Domótica	- 80 -
Tabla 4-1: Bytes en la Trama de Solicitud de Lectura del Maestro	- 88 -
Tabla 4-2: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Lectura Cuando no hay Acciones Detectadas.....	- 89 -
Tabla 4-3: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Lectura Cuando hay Acciones Detectadas (activar)	- 89 -
Tabla 4-4: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Lectura Cuando hay Acciones Detectadas (desactivar).....	- 90 -
Tabla 4-5: Bytes en la Trama de Solicitud del Maestro Para Escribir	- 92 -
Tabla 4-6: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Escribir.....	- 92 -
Tabla 4-7: Bytes en la Trama de Solicitud del Maestro Para Eliminar	- 94 -
Tabla 4-8: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Eliminar	- 95 -
Tabla 4-9: Mediciones Tiempo de Activación de Carga	- 96 -
Tabla 4-10: Bytes en la Trama de Solicitud de Configuración de Salidas Digitales del XBee Remoto.....	- 98 -
Tabla 4-11: Bytes Trama de Respuesta de Comando Remoto XBee.....	- 99 -
Tabla 4-12: Bytes Trama de Configuración XBee Remoto	- 99 -
Tabla 4-13: Bytes Trama Típica de LonWorks	- 100 -
Tabla 4-14: Tabla Resumen Comparación de Protocolos.....	- 101 -

RESUMEN

El objetivo de la investigación fue desarrollar un bus propietario, basado en el protocolo Modbus, sobre radio frecuencia, para implementar una red en configuración maestro –multiesclavo que permita implementar aplicaciones de domótica. Se inició con un análisis de las características principales de los protocolos domóticos más utilizados a nivel mundial; lo cual permitió establecer las bases del diseño del bus propietario. Una vez definida la comunicación y realizada la programación de los elementos maestro y esclavo se estructuró una red de microcontroladores, misma que posteriormente se implementó en una aplicación domótica con la finalidad de verificar la fiabilidad del bus propietario, los resultados obtenidos a través de toma de datos y mediciones, evidenció el correcto funcionamiento del bus propietario y las ventajas conseguidas en comparación con otros protocolos, en velocidad y confiabilidad de transmisión de datos. Como conclusión el protocolo propietario es versátil, es decir puede ser utilizado con cualquier tipo de microcontrolador o medio de transmisión, acoplándose a las características y necesidades de cada uno de ellos; por lo que se recomienda desarrollar el protocolo de comunicación del bus propietario en microcontroladores con mejores características para obtener mayores prestaciones, un control más ampliado de todas las acciones y sobre todo aprovechar al máximo todos los beneficios del protocolo.

Palabras clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <DOMÓTICA>, <BUS DE COMUNICACIÓN>, <PROTOCOLO MAESTRO – MULTIESCLAVO>, <RED DE MICROCONTROLADORES>, <REDES INDUSTRIALES>

SUMMARY

The objective of this research is to develop a proprietary bus, based on the Modbus protocol, on radio frequency, to implement a network in master – multi slave configuration, which allows the implementation of home automation applications. It begins with an analysis of the main characteristics of home automations protocols most used worldwide; which allows to establish the bases of the design of the proprietary bus. Once the communication was defined and the programming of the master and slave elements, a network of microcontrollers was structured, which is subsequently implemented in a home automation application with the propose of verifying the reliability of the proprietary bus. The results obtained through data collection and measurements, showed the correct operation of the proprietary bus and advantages achieved compared to other protocols, in speed and reliability of data transmission. In conclusion, the proprietary protocol is versatile, that is to say it can be used with any type of microcontroller or way of transmission, adapting to the characteristics and needs of each one of them. So it is recommended to develop the communication protocol of the proprietary bus in microcontrollers with better characteristics to obtain greater benefits, a more extended control of all the actions and especially to take full advantage of all the benefits of the protocol.

Key words: <TECHNOLGY AND ENGINEERING SCIENCES>, <HOME AUTOMATION>, <COMMUNICATION BUS>, <MASTER – MULTI SLAVE PROTOCOL>, <NETWORK OF MICROCONTROLLERS>, <INDUSTRIAL NETWORKS>

CAPITULO I

1 INTRODUCCIÓN Y CONSIDERACIONES GENERALES

1.1 Introducción:

El presente proyecto de titulación abarca el desarrollo de un bus propietario, basado en el protocolo Modbus, sobre radio frecuencia, para implementar una red en configuración maestro – multiesclavo, específico para aplicaciones domóticas.

En el sector residencial las instalaciones domóticas crecen exponencialmente, muestra de ello son los múltiples protocolos de comunicación diseñados y empleados en la actualidad, siendo los más comunes: X-10, LonWorks y ZigBee; estos protocolos presentan en su trama de datos múltiples campos, funciones y sub-funciones con la finalidad de estandarizar y cubrir todas las necesidades de la automatización de viviendas, subutilizando la mayoría de características de sus tramas de datos.

El bus propietario utiliza un protocolo maestro-multiesclavo, es decir que el bus de comunicación de datos está conformado por un único dispositivo maestro y uno o varios dispositivos esclavos. Ha sido desarrollado para las comunicaciones de una red de microcontroladores Arduino, la misma que comanda cargas residenciales de control ON/OFF, en una aplicación domótica.

Además, la trama de datos del protocolo propietario es compacta, pues prioriza el tamaño de la misma, al considerar únicamente tres campos: dirección, datos y control de errores, lo cual permite aumentar la velocidad de transmisión, la fiabilidad y la seguridad de la información, con respecto a los protocolos existentes.

El diseño del Bus propietario está basado en el protocolo MODBUS sobre línea serial, utiliza la capa física y la capa de enlace de datos del modelo OSI, el medio de transmisión es RF (radio frecuencia), y los modos de transmisión utilizados son: *unicast* y *broadcast*.

Complementariamente, la arquitectura de la red es centralizada, su elemento central es el maestro, el cual realiza el arbitraje de la red, recepta las diferentes solicitudes, procesa las mismas y envía las respectivas órdenes a los esclavos. Quienes recogen las solicitudes son los elementos de entrada y quienes ejecutan las órdenes son los elementos de salida. El número máximo de esclavos que pueden conectarse a la red es de 250 y la velocidad de comunicación en el medio es de 2Mbps, pudiendo ser esta velocidad mayor o menor acorde sean los requerimientos de diseño o de hardware.

1.2 Problema de Investigación:

1.2.1 Planteamiento del Problema:

Las redes de comunicación industrial buscan transmitir grandes volúmenes de información a gran velocidad y de forma confiable. Por lo que la demanda de mejores características para los medios de transmisión es mayor. Esto es particularmente cierto para las redes industriales de comunicación, en donde las condiciones distan mucho de ser ideales debido a las posibles interferencias de máquinas eléctricas.

Algunos de los protocolos de buses de campo aterrizaron el concepto de las comunicaciones industriales en el ámbito residencial. Las tendencias actuales en el tema del confort, seguridad, telecomunicaciones y ahorro energético aplicadas a las viviendas, edificios, oficinas y centros comerciales, han propiciado el desarrollo de nuevas áreas del conocimiento y concepción de sistemas como la Domótica e Inmótica (Zamora, Santa, J., & Gómez, A., 2010).

Según Herrera (2005), en la actualidad existen protocolos específicos para las comunicaciones residenciales, entre los que se destacan:

- X-10
- LonWorks
- ZigBee
- EIB
- JINI
- C-Bus
- KNX

Las comunicaciones en domótica se orientan a ofrecer y administrar un conjunto de aplicaciones y servicios. Las redes de domótica no necesariamente son compatibles entre sí, respecto a medios de comunicación y protocolos (Moreaux, Sartor, F., & Vernier, F., 2012). Los protocolos a su vez presentan en su trama de datos múltiples campos, funciones y sub-funciones con la finalidad de estandarizar y cubrir todas las necesidades de la automatización de viviendas y edificios (Barrera, 2011). Por tanto, la mayoría de características de la trama de datos están subutilizadas, provocando el envío de información que no es necesaria o indispensable en la comunicación. Por ejemplo, el protocolo LonWorks maneja cinco campos: sincronización, control, dirección, datos y control de errores (Junestrand, 2005); según se muestra en la Figura 1-1:

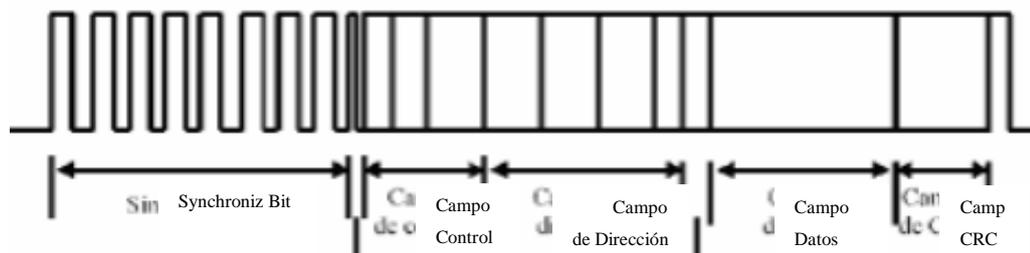


Figura 1-1: Formato de la trama LonWorks

Fuente: (Junestrand, 2005)

Por otro lado, el protocolo ZigBee posee ocho campos: secuencia de preámbulo, delimitación de inicio de trama, longitud de trama, control de trama, número de secuencia de datos, dirección, carga de datos y control de errores, según se observa en la Figura 1-2.

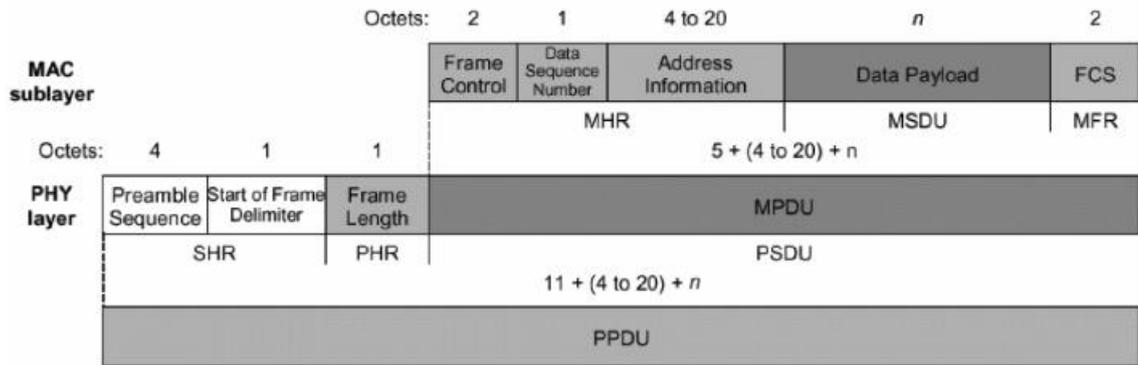


Figura 1-2: Formato de la trama ZigBee

Fuente: (Longares, 2013)

El presente proyecto pretende aportar una alternativa de comunicación específica para microcontroladores y aplicaciones domóticas, en la cual se prioriza la disminución del tamaño de la trama considerando únicamente tres campos: dirección, datos y control de errores, que permite aumentar la velocidad de transmisión eliminando campos como: control, secuencia de preámbulo, delimitación de inicio de trama, longitud de trama, control de trama, número de secuencia de datos; considerando los dos protocolos tomados como referencia.

Existen algunos artículos científicos dedicados a esta problemática, uno de ellos es (Barrera, Londoño, N., Carvajal, J., & Fonseca, A., 2012) en donde se presenta la evaluación y selección de alternativas encaminadas a diseñar un sistema domótico de bajo costo para aplicaciones didácticas, el énfasis está en la selección del protocolo de comunicación, detallando sus características y concluyendo que es posible desarrollar sistemas domóticos con tecnología propia, minimizando costos y optimizando recursos.

1.2.2 *Formulación del Problema:*

¿El desarrollo de un bus propietario, basado en el protocolo Modbus, sobre radio frecuencia, aporta una alternativa de comunicación específica para aplicaciones domóticas a través de la especificación de una trama de datos compacta y una mayor velocidad de transmisión con respecto a los protocolos existentes?

1.2.3 Sistematización del Problema:

- ¿Cuáles son los protocolos de comunicación más utilizados en domótica?
- ¿Cuáles son las principales características de los protocolos de comunicación que se emplean en domótica?
- ¿Qué alternativa de solución puede darse para mejorar la comunicación entre dispositivos interconectados en una aplicación domótica?
- ¿Cómo se puede aumentar la velocidad de transmisión en una red de datos?
- ¿Se puede considerar la construcción de una aplicación domótica para implementar el bus desarrollado?
- ¿Cuál es la mejor forma de verificar la fiabilidad del bus propietario?

1.3 Justificación de la Investigación:

En la actualidad los escenarios donde las personas interactúan son de tipo inteligente, lo que quiere decir que están dotados de un sinnúmero de funciones para su bienestar, obteniendo escenarios vivos, donde el hombre es el principal actor (Herrera, 2005). El deseo del hombre de mejorar su vivienda incorporando a ella todos los elementos que le faciliten o hagan más agradable su interacción, responde a una necesidad humana básica, relacionada no solo con sus condicionantes fisiológicas, sino también con otras características más complejas de relación con el grupo en el que se integra.

Hoy en día la globalización ha conducido a la sociedad al uso de las TIC's. Por esta razón, es necesario utilizar un medio de comunicaciones en el que se plantee la forma en la cual se transmite la información, brindando rapidez y seguridad de los datos. Para ello, es menester diseñar y desarrollar protocolos que permitan satisfacer las necesidades mencionadas.

Según Romero, Vásquez & Castro (2007) las instalaciones domóticas crecen exponencialmente en el mundo, muestra de ello son los múltiples protocolos de comunicación diseñados y empleados en la actualidad. El desarrollo de un bus propietario, objeto de este trabajo, pretende aportar una alternativa de comunicación específica para aplicaciones domóticas implementada con microcontroladores Arduino, en la cual se va a priorizar la disminución del tamaño de la

trama -permitiendo aumentar la velocidad de transmisión de los datos- y la seguridad de la información.

El proyecto tiene amplia justificación en base al objetivo número 10 del Plan Nacional del Buen Vivir que menciona: “Impulsar la transformación de la matriz productiva”; y en una de sus partes señala: “la transformación de la matriz productiva supone una interacción con la frontera científico-técnica, en la que se producen cambios estructurales que direccionan las formas tradicionales del proceso y la estructura productiva actual”. (Semplades, 2013 - 2017)

1.4 Objetivos de la Investigación:

1.4.1 Objetivo General:

Desarrollar un bus propietario, basado en el protocolo Modbus, sobre radio frecuencia, para implementar una red en configuración maestro –multiesclavo que permita implementar aplicaciones de domótica.

1.4.2 Objetivos Específicos:

- Desarrollar un bus propietario basado en el protocolo Modbus, considerando aspectos prioritarios como son la velocidad de comunicaciones y la confiabilidad de los datos.
- Determinar la técnica de modulación a emplearse en la transmisión por radio frecuencia, acorde a las especificaciones de diseño de la aplicación domótica.
- Construir una aplicación domótica utilizando 5 microcontroladores Arduino en configuración maestro multi-esclavo, para comandar cargas residenciales con control ON/OFF, implementando el bus propietario desarrollado.
- Desarrollar cuatro interfaces de contactos secos de entrada, para controlar la aplicación domótica de manera local.
- Desarrollar una interfaz Android para controlar la aplicación domótica con un celular Smart, mediante comunicación Bluetooth.

1.5 Hipótesis:

El desarrollo de un bus propietario, basado en el protocolo MODBUS sobre radio frecuencia, permitió que la velocidad con la que se transmiten sus tramas de datos sea superior en comparación con los protocolos domóticos más utilizados a nivel mundial.

CAPITULO II

2 MARCO DE REFERENCIA

Las comunicaciones industriales son el área de la tecnología que estudia la transmisión de información entre circuitos y sistemas electrónicos utilizados para llevar a cabo tareas de control y gestión del ciclo de vida de los productos industriales.

En la década de 1980, las comunicaciones industriales comenzaron a realizarse mediante comunicaciones digitales punto a punto para, posteriormente, evolucionar hacia la aplicación de redes multipunto.

Actualmente la mayor parte de procesos industriales son controlados a través de autómatas programables y un número reducido utiliza reguladores industriales y computadores de procesos. Las redes de comunicaciones industriales conocidas como buses de campo, permiten la interacción entre sensores, actuadores y equipos de control, otorgando ventajas marcadas en comparación a los sistemas de cableado tradicional, tales como: visualización y supervisión del proceso productivo, toma de datos rápida y efectiva, mejoramiento del rendimiento del proceso, ahorro en el cableado, instalación más sencilla y menor mantenimiento.

Este capítulo presenta de manera general fundamentos teóricos que forman parte del sustento para la elaboración del diseño del Bus Propietario.

2.1 Modelo OSI:

Los diversos sistemas digitales existentes, con sus múltiples funciones pueden integrarse en una red siempre y cuando estén contruidos bajo un cierto principio de normalización; en vista de ésta necesidad la ISO (Organización Internacional de Normalización) creó la norma marco internacional más general, denominada Modelo OSI (Open Systems Interconnection Model), la misma que fue desarrollada para abarcar desde las redes locales hasta las grandes redes de paquetes conmutados.

Las redes por lo general no deben ser muy rígidas, ya que ésta característica las convierte en obsoletas y más bien deben garantizar la mínima compatibilidad entre productos antiguos y los más evolucionados, integrando además productos que parecían muy distintos, como pueden ser, un autómatas, un enlace telefónico, un sistema de transmisión de imágenes, una PC de diseño CAD, etc.

Para obtener una transferencia exitosa de datos en una red, las funciones a cumplir se estructuran en capas, ubicándose las funciones más elementales en las capas inferiores, y posteriormente cada capa le brinda servicio a su capa superior y recibe servicios de su capa inferior.

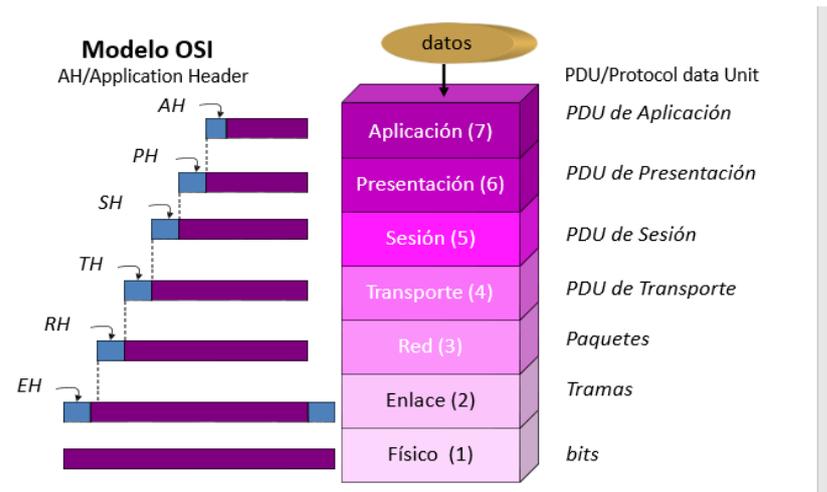


Figura 2-1: Capas del Modelo OSI

Fuente: <http://es.slideshare.net/LarryRuiz/modelo-osi-y-tcp-15503078>

2.2 Protocolo MODBUS:

La compañía Gould Modicon (ahora AEG Schneider Automation) en el año 1984 desarrolló el Protocolo MODBUS para sistemas de control y supervisión de procesos, se caracteriza por su simplicidad y especificación abierta, en su estructura establece como los mensajes se intercambian en forma ordenada y permite la detección de errores; todo esto ha contribuido a que sea ampliamente utilizado por diferentes fabricantes y dispositivos tales como: PLC, HMI, RTU, Drivers, sensores y actuadores remotos.

MODBUS no incluye en su diseño todas las capas del estándar de red, es decir los siete niveles, sino más bien se lo conoce como un protocolo de enlace (nivel OSI 2), puede, por tanto, implementarse con diversos tipos de conexión física y cada fabricante suele suministrar un software de aplicación propio, que permite parametrizar sus productos, no obstante, se suele hablar de MODBUS como un estándar de bus de campo.

2.3 Transmisión por Radio Frecuencia:

La Radio Frecuencia se conoce como la tecnología que usa ondas aéreas electromagnéticas para comunicar información desde un punto a otro; siendo portadoras de radio porque desempeñan la función de entregar energía al receptor. El sistema de transferencia de datos es bidireccional, los cuales son sobrepuestos sobre la señal de radio para que se puedan extraer de manera precisa por el receptor. Éstas ondas de radio frecuencia forman parte de un espectro electromagnético no perceptibles por el ser humano.

Uno de los métodos utilizados para la transmisión de la información es el de la modulación, el cual es un sistema que parte de dos ondas:

- Onda portadora: es la onda sobre la cual se debe trabajar y alterar para que se pueda transportar la información deseada, siendo además la encargada de fijar la frecuencia de transmisión.
- Onda moduladora: es la onda a transmitir (voz, música, datos, etc.).

Entonces la modulación se basa en alterar de una forma específica la onda portadora en función de la onda moduladora, adquiriendo como resultado final la onda modulada, misma que posteriormente será radiada, basándose en parámetros como la amplitud y la frecuencia.

Cabe resaltar que internacionalmente se ha dividido todo el espectro de frecuencia en las denominadas bandas de frecuencia. Esto se realizó con el objetivo de poder delimitar el acceso de los usuarios a estas bandas, de manera general existen las siguientes:

Tabla 2-1: Bandas de Frecuencia

Denominación	Siglas	Margen de Frecuencias
Frecuencias muy bajas	VLF	3 – 30 KHz
Frecuencias bajas	LF	30 – 300 KHz
Frecuencias medias	MF	300 – 3000 KHz
Frecuencias altas	HF	3 – 30 MHz
Frecuencias muy altas	VHF	30 – 300 MHz
Frecuencias ultra altas	UHF	300 – 3000 MHz
Frecuencias super altas	SHF	3 – 30 GHz
Frecuencias extra altas	EHF	30 – 300 GHz

Fuente: <https://www.taringa.net/post/ciencia-educacion/17255781/La-transmision-por-radio.html>

2.3.1 Modulación Digital:

La modulación digital a través de la modulación de la onda portadora permite enviar datos digitales, los mismos que generalmente están conformados por cadenas de ceros y unos, dejando a un lado las formas de ondas analógicas; esto no impide que los datos puedan representar información analógica (tal como la voz, audio, video), pasando previamente por una etapa de conversión de analógico a digital (ADC, por sus siglas en inglés), es decir tienen que representarse por medio de un valor definido dentro de un rango previamente especificado.

Los moduladores digitales pueden ser de los siguientes tipos:

- por cambio de fase (PSK).
- por cambio de frecuencia (FSK).
- por cambio de amplitud (ASK).
- modulación de amplitud en cuadratura (QAM).

A más de los descritos anteriormente, se ha desarrollado un tipo de modulación especial, que es de mucha utilidad, la misma se la conoce como: modulación por desplazamiento de frecuencia gausiana (GFSK), la cual es una técnica que realiza transiciones suaves entre cada símbolo digital, evitando de este modo el cambio abrupto de frecuencia durante la transmisión, que a diferencia de FSK, esta no realiza transiciones de frecuencia de un modo suave, llegando a presentarse algunas transiciones de frecuencia de 180°.

La modulación GFSK está constituida por un banco de filtros Gaussianos donde los pulsos digitales (pulsos cuadrados que representan los unos y ceros) son suavizados antes de entrar a la etapa de modulación, es decir, los pulsos utilizados tienen una forma de onda más parecida a una onda senoidal, pero sin perder su periodo. Este sistema presenta de forma general ventajas como la reducción de la potencia de la banda lateral y la interferencia con canales adyacentes, pero genera un incremento de la interferencia intersímbolo.

2.4 Domótica:

La palabra domótica es el resultado de la unión de la palabra “domo” y el sufijo “tica”; la palabra “domo” etimológicamente proviene del latín domus que significa casa, y el sufijo “tica” proviene de la palabra automática, aunque algunos textos también diferencian entre “tic” de tecnologías de la información y de la comunicación y “a” de automatización.

Una definición más técnica del concepto es: "conjunto de servicios de la vivienda garantizados por sistemas que realizan varias funciones, los cuales pueden estar conectados entre sí y a redes interiores y exteriores de comunicación. Gracias a ello se obtiene un notable ahorro de energía, una eficaz gestión técnica de la vivienda, una buena comunicación con el exterior y un alto nivel de seguridad".

En resumen, se podría decir que los cuatro pilares fundamentales de la domótica son:

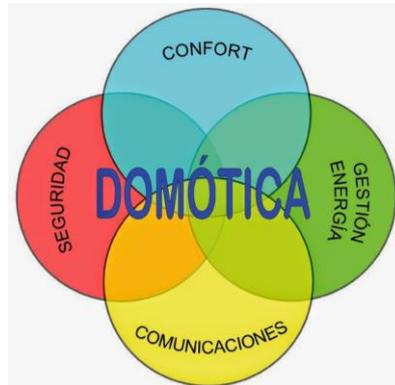


Figura 2-2: Pilares Fundamentales de la Domótica

Fuente: <http://es.slideshare.net/xendika2/hobetuz-domotica-1>

2.5 Protocolo X-10:

X-10 es uno de los protocolos más antiguos que se están usando en aplicaciones domóticas (Allen, 2005). El objetivo primordial era el de aprovechar las líneas eléctricas de baja tensión para transmitir datos, obteniendo un bajo costo de transmisión pero con una baja velocidad (60 bps en EEUU y 50 bps en Europa). El formato de codificación X-10 es un estándar de transmisión por corrientes portadoras (Power Line Communication.) que, como tal, permite que todos los equipos que lo utilizan sean compatibles entre sí (Held, 2006). La técnica de corrientes portadoras se basa en el aprovechamiento doble de las instalaciones eléctricas ya existentes, ya que conducen energía e información.

La forma de transmitir las señales se basa en una sincronización con el punto de cruce en cero de la corriente eléctrica (de 50Hz a 60Hz), tratando siempre de transmitir lo más cerca posible al punto cero, el margen aceptable de variación es de 200 microsegundos desde el cruce por cero. El mensaje está compuesto tanto por unos y ceros binarios, los cuales se los representa de la siguiente manera: al "1" binario por un pulso de 120 KHz durante 1 ms, en el cruce por cero de la señal de red, y al "0" binario por la ausencia de ese pulso de 120 KHz.

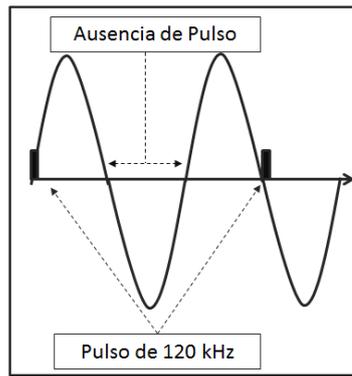


Figura 2-3: Envío de un “1” y un “0” binario.

Fuente: (Chan, y otros, 2011)

La estructura de un mensaje en X-10 se basa en un código de inicio 1110, seguido por código de casa y por un código de control, éste último puede ser una dirección de unidad o un comando. El Protocolo X-10 puede utilizar hasta 16 unidades por cada dirección de casa, dando un total de 256 unidades. En la Tabla 2-2 se puede apreciar tanto las direcciones con los códigos de casa, así como las direcciones de unidad con sus respectivos códigos de control, y por último se listan los comandos con los cuales se pueden controlar las 16 unidades que maneja el protocolo X-10.

Tabla 2-2: Elementos de una trama del protocolo X-10

Dirección y códigos de casa		Direcciones de unidad y códigos control					Comandos y códigos de control									
Dirección de casa	Códigos de casa				Dirección de unidad	Códigos de control					Comandos	Códigos de control				
	H1	H2	H4	H8		D1	D2	D4	D8	D16		D1	D2	D3	D4	D5
A	0	1	1	0	1	0	1	1	0	0	Encender todas las unidades	0	0	0	0	1
B	1	1	1	0	2	1	1	1	0	0	Apagar todas las unidades	0	0	0	1	1
C	0	0	1	0	3	0	0	1	0	0	Encender	0	0	1	0	1
D	1	0	1	0	4	1	0	1	0	0	Apagar	0	0	1	1	1
E	0	0	0	1	5	0	0	0	1	0	Atenuar la intensidad	0	1	0	0	1
F	1	0	0	1	7	0	1	0	1	0	Aumentar la intensidad	0	1	0	1	1
G	0	1	0	1	8	1	1	0	1	0	Apagar todas las luces	0	1	1	0	1
H	1	1	0	1	9	0	1	1	1	0	Código extendido	0	1	1	1	1
I	0	1	1	1	10	1	1	1	1	0	Petición de saludo	1	0	0	0	1
J	1	1	1	1	11	0	0	1	1	0	Aceptación de saludo	1	0	0	1	1
K	0	0	1	1	12	1	0	1	1	0	Atenuación preestablecida	1	0	1	X	1
L	1	0	1	1	13	0	0	0	0	0	Datos extendidos	1	1	0	0	1
M	0	0	0	0	14	1	0	0	0	0	Estado = On	1	1	0	1	1
N	1	0	0	0	15	0	1	0	0	0	Estado =Off	1	1	1	0	1
O	0	1	0	0	16	1	1	0	0	0	Petición de Estado	1	1	1	1	1
P	1	1	0	0												

Fuente: (Chan, y otros, 2011)

Una red de dispositivos X-10 se construye asignando a cada dispositivo una identificación de un código de 9 bits, donde los 4 primeros bits hacen referencia al “código de casa” y los siguientes 5 bits a la “dirección de unidad” o “código de control”. El “código de inicio” representado por el número binario 1110, requiere 2 ciclos completos de la corriente eléctrica para poder ser enviado ya que cada dígito binario ocupa medio ciclo. Los bits del “código de casa”, “dirección de unidad” y “código de control”, se transmiten en *forma de complemento* en los ciclos de la corriente alterna; es decir, para enviar el 0 se envía un 0 en el primer medio ciclo y un 1 en el segundo medio ciclo, para enviar el 1, se envía un 1 en el primer medio ciclo y un 0 en el segundo medio ciclo. Cabe resaltar que el “código de inicio” es el único que no se envía en complemento.

Tabla 2-3: Complemento del código X-10

	Código de inicio	Código de casa				Código de control				
Código X-10	1110	0	1	1	0	0	1	1	0	0
Complemento	1110	01	10	10	01	01	10	10	01	01

Fuente: (Chan, y otros, 2011)

La trama de datos X-10 está dividida en dos paquetes de datos los cuales deben enviarse un mínimo dos veces para garantizar que los datos sean correctos; para poder enviar el segundo paquete de datos es necesario esperar tres ciclos de corriente alterna sin transmisión, equivalente a seis cruces por cero, según se observa en la Figura 2-4; los dos paquetes difieren en que el código de control del primero se refiere a una dirección de unidad mientras que en el segundo se refiere a un comando.

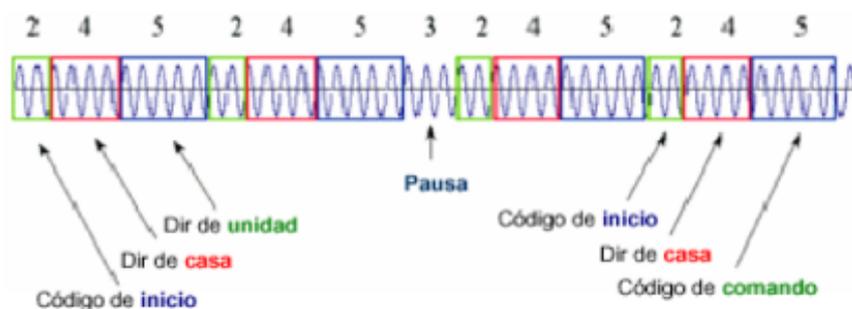


Figura 2-4: Ciclos para transmisión completa en X-10

Fuente: (Yunaningsih, 2009)

Uno de los inconvenientes más significativos del protocolo X-10 es su susceptibilidad al ruido externo; que incluso, en ambientes con alto grado de contaminación electrónica es prácticamente

imposible el correcto funcionamiento del sistema. Fuentes de ruido como: taladros, licuadoras, máquinas de soldar, prensas, etc., contaminan la red y ocasionan la pérdida de datos en la transmisión.

2.6 Protocolo ZigBee:

ZigBee es una especificación de red de malla, conformada por un conjunto de protocolos de alto nivel de comunicación inalámbrica, basados en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN); su mayor utilización se da en aplicaciones que necesitan una baja tasa de envío de datos, maximización de la vida útil de sus baterías y un alto nivel de seguridad en la comunicación; opera en las bandas de 868 MHz, 915 MHz y 2.4 GHz y puede transferir datos hasta 250Kbps.

El protocolo ZigBee se basa en el modelo OSI, no utiliza las 7 capas del estándar de red, sino solamente 4 capas como se puede observar en la Figura 2-4, su objetivo es el de simplificar la arquitectura para consolidar una red simple, de bajo consumo y de baja transmisión. La capa física (PHY) y la capa de acceso al medio (MAC) están definidas por el Standard IEEE 802.15.4; mientras que las capas de red (NWK) y de aplicación (APL) se definen en ZigBee. La forma de conectar las capas de la red es por medio de un SAP (Service Access Point). Un SAP es un lugar por donde una capa superior requiere un servicio a una capa inferior. (Dignani, 2011).

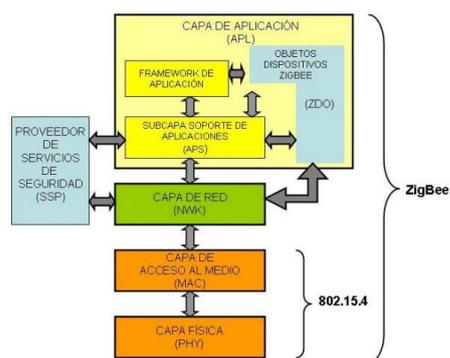


Figura 2-5: Capas de 802.15.4 y ZigBee

Fuente: (Dignani, 2011)

2.6.1 Capa Física (PHY):

La capa Física como se había mencionado anteriormente corresponde al Standar IEEE 802.15.4; define aspectos básicos como la potencia del transmisor y la sensibilidad del receptor; además de las funciones y la relación con la capa MAC.

2.6.2 Capa de Acceso al Medio (MAC):

La capa MAC provee una interfaz entre la capa física y la de red, en esta capa se crean tramas especiales, llamadas tramas de baliza, las cuales permiten en la red, disponer de ranuras de tiempo garantizados (GTS), y de esta forma utilizar las supertramas, cuya estructura es la siguiente:

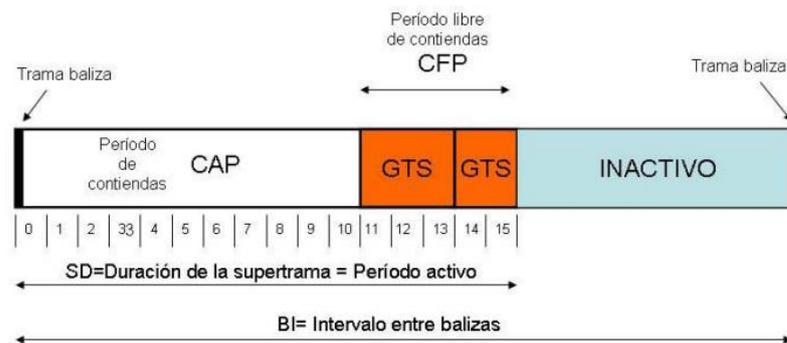


Figura 2-6: Estructura de una Supertrama

Fuente: (Dignani, 2011)

Los tres periodos de una supertrama son: período de acceso en contienda (CAP), período libre de contiendas (CFP) y período inactivo. En el periodo CAP los dispositivos no tienen la garantía de poder usar el canal en el momento deseado ya que están en competencia con otros dispositivos, por tal razón los nodos que desean transmitir durante éste periodo deben usar CSMA-CA para acceder a un canal que está disponible igualmente para todos los dispositivos. En el período CFP, los dispositivos no necesitan competir usando CSMA-CA ya que pueden tener garantizado una ranura de tiempo (time slot). El periodo de inactividad se lo conoce también como nodo en modo “dormir”.

El formato de la trama MAC se caracteriza por que presenta 4 tipos de tramas: de baliza, de dato, de comando y de confirmación (ack); a su vez todas las tramas están compuestas por 3 partes: encabezado (MHR), carga útil (payload) y un pie (MFR). El encabezado contiene información sobre el tipo de trama, campos de direcciones y banderas de control. La carga útil tiene un largo variable y contiene comandos o datos ó nada (cero bytes). El MFR contiene una secuencia de chequeo (FCS) para verificar los datos basada en el clásico polinomio cíclico redundante (CRC) (Sosa, 2011). Las figuras 2-6, 2-7, 2-8 y 2-9 muestran los diferentes formatos que puede tener la trama MAC (IEEE.802.15.4, 2006)

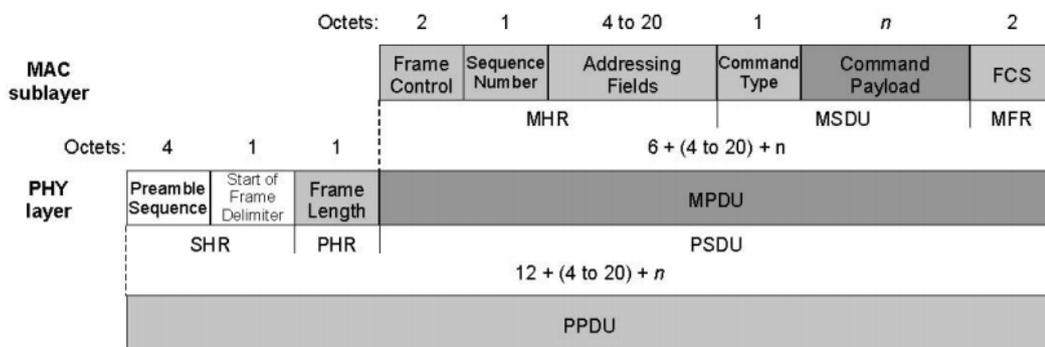


Figura 2-7: Estándar 802.15.4. Trama de comando

Fuente: (IEEE.802.15.4, 2006)

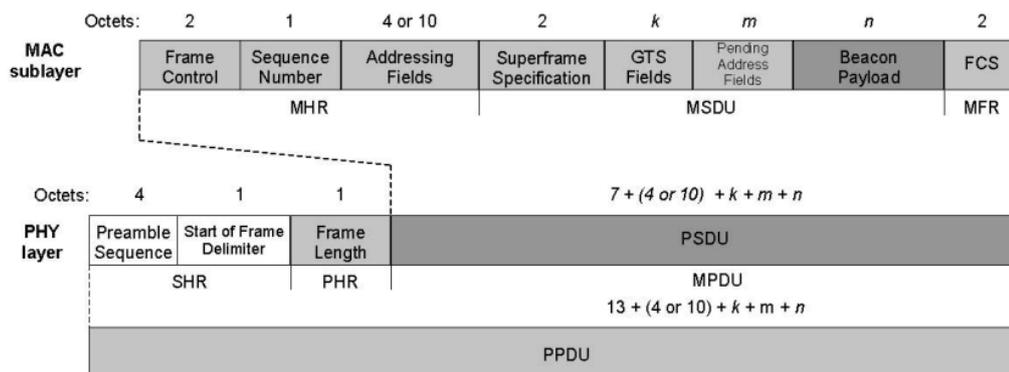


Figura 2-8: Estándar 802.15.4. Trama de baliza

Fuente: (IEEE.802.15.4, 2006)

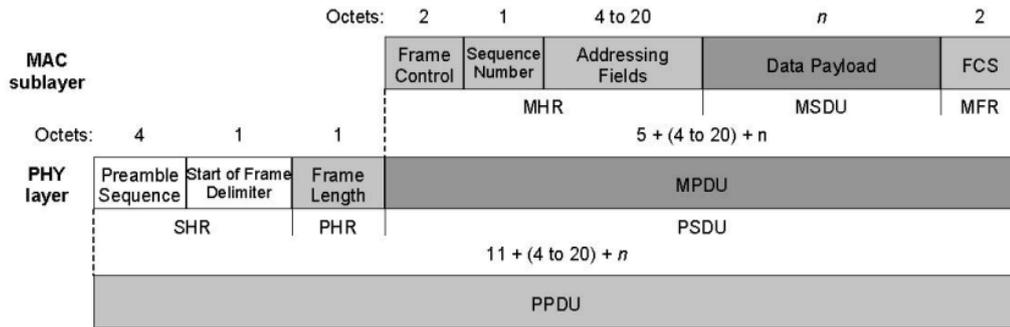


Figura 2-9: Estándar 802.15.4. Trama de dato

Fuente: (IEEE.802.15.4, 2006)

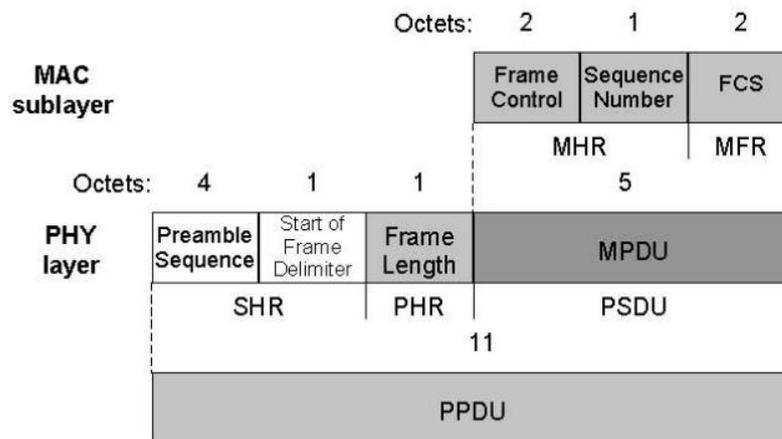


Figura 2-10: Estándar 802.15.4. Trama de confirmación

Fuente: (IEEE.802.15.4, 2006)

2.6.3 Capa de Red (NWK):

La capa de red proporciona herramientas para el armado y manejo de redes, mediante una interfaz simple que permite relacionarla con las aplicaciones de los usuarios. Los dispositivos o nodos con los que cuenta son tres: coordinador, ruteador y dispositivo final.

El coordinador se encarga de incluir/excluir nodos a/de la red, inicia la red, configura los parámetros de la red y asigna las direcciones de la red. Es obligatoria la presencia de un solo nodo coordinador y actúa como nodo raíz en la topología árbol. Los enrutadores son dispositivos opcionales en la red, tienen la capacidad de extenderla para aumentar la confiabilidad con la

creación de rutas adicionales de datos. Los dispositivos finales trabajan en conjunto con los coordinadores y enrutadores, se optimizan para consumir muy poca potencia.

El protocolo ZigBee en la transferencia de datos usa las topologías de IEEE 802.15.4 y agrega las topologías de árbol, estrella y de malla.

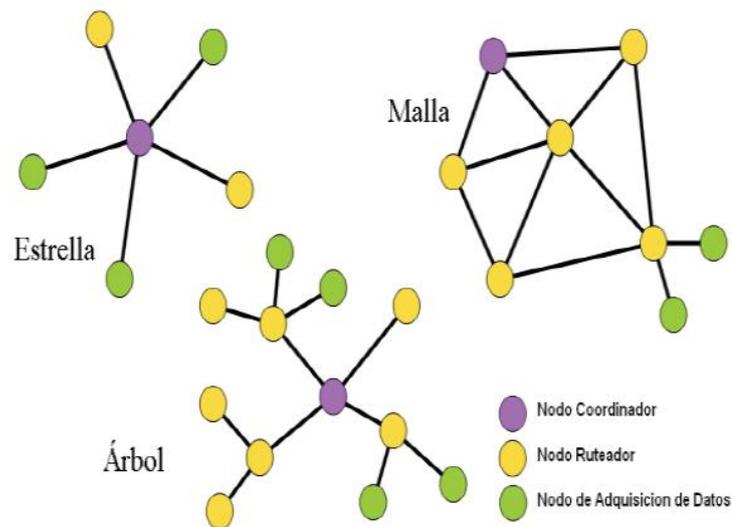


Figura 2-11: Topologías de Red en ZigBee

Fuente: (Cázares, y otros, 2011)

2.6.4 Capa de Aplicación (APL):

Constituido por las subcapas: APS (Application Support) y la ZDO (ZigBee Device Object), su función es la de mantener las tablas para los enlaces (binding), lo cual consiste en balancear o adaptar dos dispositivos entre ellos, basándose en los servicios y necesidades. Cada subcapa se puede explicar de la siguiente manera:

- APS trata de descubrir a otros dispositivos que están operando en su mismo espacio operativo.
- ZDO: Define el rol de un dispositivo dentro de la red.

2.7 Protocolo LonWorks:

El sistema *Lonworks* fue creado por la empresa americana *Echelon* en el año 1988, se presentó como una plataforma universal para implementar casi cualquier sistema de control, existen más de 4000 fabricantes de dispositivos que usan esta tecnología para desarrollar sistemas distribuidos empleados en diversos ámbitos, como: hogares, empresas, edificios inteligentes, sistemas de transporte, etc., todas las herramientas de software y hardware necesarias para el desarrollo de sistemas LonWorks son ofrecidas por la empresa Echelon. Debido al coste de los dispositivos, el protocolo no ha tenido una implantación masiva en los hogares, ya que existen tecnologías más baratas.

La esencia de las redes Lonworks se sustentó en los siguientes conceptos: 1) independientemente de la aplicación final, todos los sistemas de control tienen una estructura idéntica; 2) los sistemas de control distribuidos presentan ventajas en cuanto a la potencia, flexibilidad y posibilidad de amplitud en comparación con los sistemas de control centralizado; y 3) las empresas cuando instalan redes distribuidas en lugar de redes centralizadas, ahorran dinero a largo plazo.

El protocolo de comunicaciones se denomina *LonTalk*, el cual implementa las siete capas del modelo OSI y posee una arquitectura abierta a cualquier fabricante, el sistema de control se basa en una agrupación de nodos que trabajan de forma independiente interconectados entre sí, los mismos que se basan en un microcontrolador llamado Neuro Chip, el cual está constituido por tres microprocesadores en uno, dos están programados para ejecutar el protocolo de comunicaciones y uno está dedicado a ejecutar el control del nodo (aplicación);

La comunicación se realiza mediante intercambio de paquetes de datos, cada dispositivo dispone un identificador único y analiza todos los paquetes que le van llegando para determinar si corresponde con su dirección. La comunicación es fiable con CSMA, y se realiza mediante el envío de telegramas que contienen la dirección del destino, información para el routing, datos del control así como los datos de la aplicación del usuario y un checksum como código detector de errores; cada receptor tiene un transceptor para conectarse físicamente a la red.

La plataforma Lonworks es independiente de los medios de comunicación, puede funcionar sobre RS-485 opto-aislado, acoplado a un cable coaxial o de pares trenzados con un transformador, sobre corrientes portadoras, fibra óptica e incluso radio

2.7.1 Direccionamiento y Medio de Transporte:

El sistema Lonworks soporta varios tipos de direccionamiento:

- Dirección física: cada dispositivo tiene en su estructura un único identificador de 48 bits conocido como Neuron ID, nunca cambia durante toda su vida útil.
- Dirección del dispositivo: el momento de instalar los dispositivos en una red, se les asigna una dirección particular que consta de tres partes: identificación de dominio, identificación de subred e identificación de nodo.
- Dirección de grupo: se considera a un grupo como la colección lógica de dispositivos dentro de un dominio, y es utilizada para reducir la cantidad de paquetes necesarios para el acceso a varios dispositivos.
- Dirección broadcast: reconoce a todos los dispositivos en un dominio o una subred, si el mensaje interesa a todos, se convierte en el método más eficiente.

LonTalk independientemente del medio de transmisión utilizado permite a los dispositivos comunicarse a través de cualquier medio físico de transporte (canal), usando los transceivers adecuados; cada canal tiene limitaciones físicas de distancia, velocidad, número de dispositivos máximos conectables, etc.; tal como se puede observar en la Figura 2-4:

Tabla 2-4: Tipos y Características de Canales en Lonworks

Tipo Canal	Medio	Velocidad	Máx. Distancia	Máx. N° Nodos	Observaciones
TP/FT-10	Par trenzado Topología libre o bus (*)	78 kbps	500 m. Topología Libre 2700 m. Topología Bus	64	Es el más usado por la simplicidad de su bus como sistema de conexión entre nodos en un edificio.
TP/LPT-10	Par Trenzado Link Power Topología libre o bus	78 kbps	500 m. Topología Libre 2200 m. Topología Bus	128	Menos usado, permite llevar alimentación al nodo (limitada) para alimentar su electrónica.
TP/XF-1250	Par trenzado	1,25 Mbps	130 m.	64	Usado como 'troncal' para unir varios canales
TP/XF-78	Par trenzado	78 kbps	1400 m.	64	Primer tipo de bus Lon, tiende a desaparecer.
PL-20	Línea potencia	5,4 kbps Banda C 3,6 kbps Banda A	Depende del entorno	---	Se usa el tendido eléctrico como bus, no precisa tirada bus, usado en SmartCities.
IP-10	Lonworks sobre IP	10, 100 Mbps 1Gb	Determinado por red IP	---	Usado para conectar instalaciones entre sí o para acceso remoto desde Internet.
FO-20	Fibra óptica	1,25 Mbps	30 kilómetros	---	Para unir instalaciones a larga distancia.

Fuente: (Lonworks, 2012)

2.7.2 Mensaje y Trama de Datos:

Para el correcto envío y recepción de paquetes el protocolo contempla distintos tipos de mensajes:

- Reconocimiento: el emisor espera la respuesta de todos los dispositivos individualmente, si no se produce después de un tiempo de espera determinado, el emisor reintenta la transmisión.
- Repetición del mensaje: el mensaje es reenviado múltiples veces a un dispositivo o grupo, muy útil en el caso de reconocimiento de un grupo de dispositivos.
- Mensaje con No reconocimiento: no se espera reconocimiento y disminuye notablemente la carga de la red.
- Servicio de Autenticación: el receptor puede determinar si el emisor estaba autorizado para el envío del mensaje.

En cuanto a la trama de datos, LonWorks maneja cinco campos: sincronización, control, dirección, datos y control de errores; según se pudo observar en la figura 1-1 del capítulo I: *Formato de la trama LonWorks*.

Para mejorar la eficiencia de la comunicación entre los nodos, se ha creado una lista estándar de datos “variables”, llamados SNVTs (Estándar Network Variable Type), los cuales se encuentran

en la programación de los nodos e indican un formato para el intercambio de la información, por ejemplo: SNVT_temp, hace referencia a la temperatura en grados Celsius.

CAPITULO III

3 DISEÑO DE INVESTIGACIÓN

3.1 Metodología

3.1.1 Tipo y Diseño de Investigación:

La metodología utilizada en el presente proyecto es Experimental, ya que abarca el diseño y desarrollo de un bus propietario y la implementación de una aplicación domótica, basado en pruebas de funcionamiento.

Por otra parte el diseño de investigación es transversal, debido a que se analiza el desempeño del protocolo propietario en un corto periodo de tiempo.

3.1.2 Métodos de Investigación:

El método utilizado es deductivo ya que se analizan los protocolos comerciales más utilizados en domótica, y se desarrolla un protocolo propietario alternativo que permita mejorar la velocidad en las comunicaciones.

3.1.3 Enfoque de Investigación:

El enfoque de investigación es cuantitativo, en vista a que se realizan mediciones numéricas de la velocidad de las transacciones de datos dentro de una red de microcontroladores maestro – multiesclavo, que utilizan radio frecuencia como medio de comunicación.

3.1.4 Alcance de la Investigación:

El alcance de la investigación es descriptivo, ya que se especifican las propiedades importantes del protocolo propietario en base a resultados (mediciones) de pruebas y análisis experimentales realizados.

3.1.5 Población de Estudio:

La población de estudio se centra en el sector residencial; sin embargo, las pruebas de funcionamiento del bus propietario se desarrollan en una aplicación domótica, para lo cual se construye una maqueta a escala de una vivienda.

3.1.6 Unidad de Análisis:

La unidad de análisis consiste en una red de microcontroladores ATMEGA 2560 y ATMEGA 328P en configuración maestro – multiesclavo que implementan módulos nRF24L01 2.4GHz para la transmisión/recepción de datos.

3.1.7 Selección de la Muestra:

La selección de la muestra se realizará de manera no probabilística. Los datos corresponden a experimentos realizados en una red de microcontroladores ATMEGA 2560 y ATMEGA 328P en configuración maestro – multiesclavo que implementan módulos nRF24L01 2.4GHz para la transmisión/recepción de datos, los cuales manejan los sistemas implementados en una vivienda a escala. Todos los resultados obtenidos durante el periodo de pruebas serán utilizados para el análisis.

3.1.8 *Tamaño de la Muestra:*

El tamaño de la muestra está confinado al experimento realizado en la vivienda a escala utilizado.

3.1.9 *Técnicas de Recolección de Datos Primarios y Secundarios:*

Los datos de velocidad del protocolo son recolectados en una red de microcontroladores ATMEGA 2560 y ATMEGA 328P, en configuración maestro – multiesclavo que implementan módulos nRF24L01 2.4GHz para la transmisión/recepción de datos.

Los datos de las velocidades de los protocolos más utilizados en la domótica son recolectados de la información publicada en otros trabajos de investigación.

3.1.10 *Instrumentos de Recolección de Datos Primarios y Secundarios:*

Los datos de velocidad del protocolo propietario son recolectados con un osciloscopio de dos – canales: 100MHz TEKTRONIX TDS 2012B.

3.1.11 *Instrumentos Para Procesar Datos Recopilados:*

Se utiliza la herramienta informática Excel para ingresar los resultados de las pruebas realizadas y procesar la información.

3.2 *Diseño y Desarrollo del Bus Propietario*

3.2.1 *Diseño del Bus Propietario:*

El diseño del Bus propietario está basado en el protocolo MODBUS sobre línea serial, utilizando las capas 1 y 2 del modelo OSI, es decir la capa física y la de enlace de datos, respectivamente, según se representa en la Figura 3-1. Ha sido desarrollado para las comunicaciones de la red de microcontroladores Arduino, la misma que comanda cargas residenciales con control ON/OFF, en una aplicación domótica.

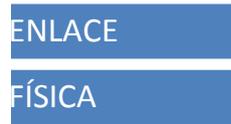


Figura 3-1: Capas del Bus Propietario

Realizado por: Flores, Pablo. 2017

El Bus propietario desarrollado utiliza un protocolo maestro-multiesclavo, es decir que el bus de comunicación de datos está conformado por un único dispositivo maestro y uno o varios dispositivos esclavos, en éste caso cuatro.

3.2.2 *Capa Física:*

El medio de transmisión, es decir la capa física, es RF (radio frecuencia), para lo cual se ha empleado el transreceptor nRF24L01 2.4GHz



Figura 3-2: Transreceptor nRF24L01

Fuente: <http://www.dx.com/es/p/nrf24l01-2-4ghz-wireless-transceiver-module->

3.2.2.1 Características del transreceptor nRF24L01

El nRF24L01 es un único chip transreceptor de 2.4GHz, con una ingeniería de protocolo de banda base embebida (Enhanced ShockBurst), adecuado para aplicaciones inalámbricas de muy baja potencia, opera en una banda de frecuencia que va desde 2,400 a 2,4835 GHz. Los módulos usan una modulación GFSK para la transferencia / recepción de datos, además poseen parámetros configurables tales como: canal de frecuencia, potencia de salida y velocidad de datos en el aire. (nRF24L01-Datasheet).

La potencia de salida del transmisor puede ser de -18, -12, -6 y 0 dBm, alcanzando su máxima potencia con una carga de $15\Omega + j88\Omega$ en la antena de salida; en el presente protocolo se ha utilizado la potencia más baja ya que se trabaja con una antena impresa en la tarjeta de circuito, que brinda un alcance en línea de vista de 15 mts.

3.2.2.2 Protocolo de banda base embebida nRF24L01

El protocolo Enhanced Shock Burst es una capa de enlace de datos basada en paquetes que incluye el ensamblaje automático de paquetes y la sincronización, reconocimiento automático y retransmisiones de paquetes, permite la implementación de la comunicación de potencia ultra baja y alto rendimiento con microcontroladores de bajo costo; permite además mejoras significativas en la eficiencia energética para sistemas bidireccionales y unidireccionales, sin añadir complejidad en el lado del controlador host (nRF24L01-Datasheet).

El protocolo soporta varios modos de operación manual hasta avanzados protocolos de operación automátatas. En su interior la programación: primero en entrar - primero en salir (FIFOs), asegura un continuo flujo de datos entre el receptor y el microcontrolador.

3.2.2.3 Velocidad de datos en el aire del nRF24L01

La velocidad de datos de aire es la velocidad de señalización modulada que el nRF24L01 utiliza al transmitir y recibir datos, ésta puede ser 250Kbps, 1Mbps y 2Mbps. Utilizando una velocidad

de datos inferior, da una mejor sensibilidad del receptor; pero, la alta velocidad de datos de aire da un menor consumo de corriente promedio y una menor probabilidad de colisiones en el aire. (nRF24L01-Datasheet)

La más alta velocidad de datos en el aire combinada con dos modos de ahorro de energía, convierten al nRF24L01 en un transceptor adecuado para diseños de energía ultra baja. La velocidad utilizada en el bus propietario es de 2Mbps

3.2.2.4 Canal de Frecuencia RF del nRF24L01

El canal de frecuencia RF, determina el centro del canal utilizado por el nRF24L01. El canal ocupa un ancho de banda inferior a 1MHz a 250Kbps y 1Mbps, y un ancho de banda inferior a 2MHz a 2Mbps. El transceptor puede operar a frecuencias de 2,400 a 2,4835 GHz; y viene seteado a 1MHz.

3.2.2.5 Diagrama de Bloques del nRF24L01

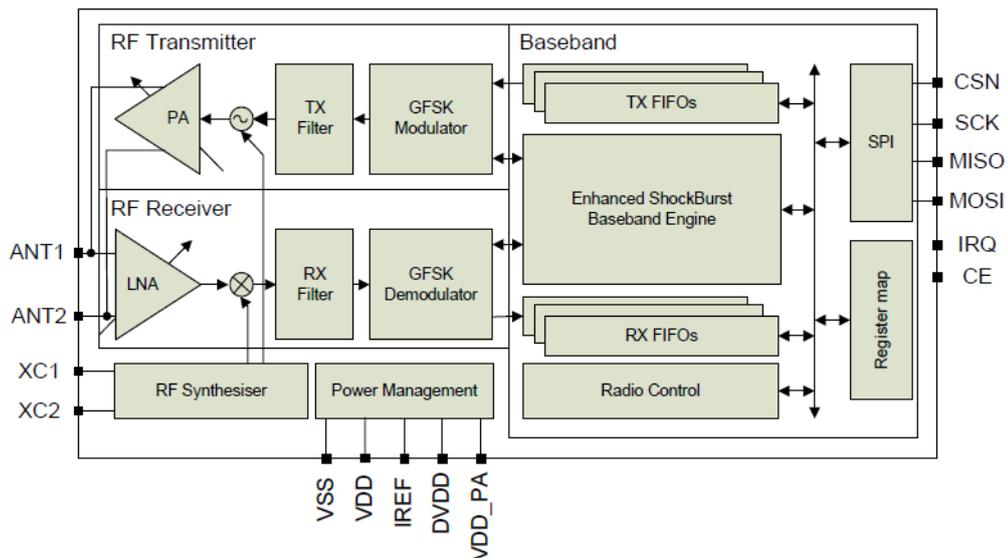


Figura 3-3: Diagrama de Bloques del Transceptor nRF24L01

Fuente: (nRF24L01-Datasheet)

3.2.3 Capa Enlace:

En cuanto a la capa de enlace, los modos de transmisión que se han utilizado son: *unicast* y *broadcast*. El protocolo desarrollado por su parte, trabaja de forma general con las siguientes características:

- El maestro siempre inicia la comunicación.
- La información no es transmitida por los esclavos, sin antes recibir una solicitud del maestro.
- No existe una comunicación entre los esclavos.
- El maestro puede iniciar solamente un proceso de intercambio de información al mismo tiempo.



Figura 3-4: Trama de Datos del Protocolo Desarrollado

Realizado por: Flores, Pablo. 2017

3.2.3.1 Intercambio de Información Modo Unicast

Una transacción del protocolo en el modo *unicast*, está conformado por dos mensajes: el primero consiste en una solicitud (request) del maestro y el segundo hace referencia a una respuesta (reply) del esclavo. En éste tipo de intercambio de información cada esclavo debe tener una dirección única, para que pueda ser direccionado de forma independiente con relación a los otros esclavos.

En este sentido, el maestro envía una solicitud a un único esclavo, el cual recibe y procesa dicha solicitud, para posteriormente devolver un mensaje de respuesta al maestro.

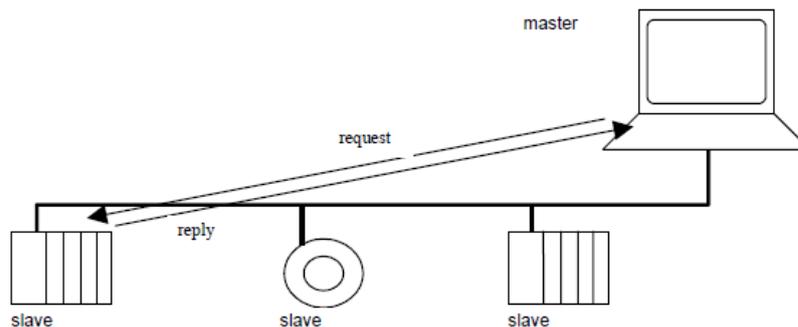


Figura 3-5: Modo Unicast

Fuente: (Modbus-Organization, 2006)

3.2.3.2 Intercambio de Información Modo Broadcast

Una transacción del protocolo en el modo *broadcast*, está conformado de un único mensaje, el cual consiste en una solicitud (request) del maestro. Para poder identificar una solicitud broadcast, se ha reservado la dirección “254”.

En este sentido, la solicitud enviada por el maestro está dirigida a todos los esclavos, quienes reciben y procesan dicha solicitud, para posteriormente pasar al estado de reposo (idle), no tienen que emitir ningún mensaje de respuesta al maestro.

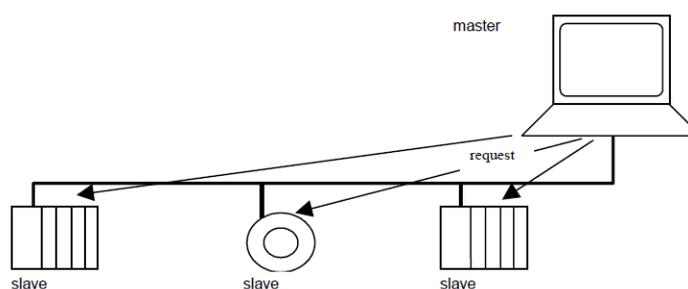


Figura 3-6: Modo Broadcast

Fuente: (Modbus-Organization, 2006)

3.2.3.3 Trama del Mensaje

La trama del mensaje del Protocolo está constituida por los siguientes campos:

- Dirección: se estructura solamente con la dirección del esclavo.
- Función: el código ingresado revela la acción a ser realizada por el esclavo
- Datos: abarca la información del mensaje
- CRC: consiste en la verificación de errores, para lo cual se realiza una comprobación de redundancia cíclica a todo el contenido del mensaje, a través de un algoritmo.

Dirección	Función	Datos	CRC	
			CRCL	CRCH
1 byte	1 byte	(N-4) bytes	1 byte	1 byte

N: número de bytes de la trama del mensaje
 CRCL: byte de orden inferior del campo CRC
 CRCH: byte de orden superior del campo CRC

Figura 3-7: Trama del Mensaje

Realizado por: Flores, Pablo. 2017

Cabe resaltar que el campo “Dirección” posee en su estructura solamente la dirección del esclavo citado; es decir que, el momento de realizar una transacción maestro-esclavo, el maestro colocará la dirección del esclavo en el campo correspondiente a la dirección del mensaje de solicitud (request); y, cuando el esclavo responda, éste debe de igual manera ubicar su propia dirección en el campo dirección del mensaje de respuesta (reply), todo esto con el objetivo de que el maestro pueda reconocer si está respondiendo el esclavo correspondiente o un esclavo inesperado.

3.2.3.4 Direccionamiento

En el protocolo desarrollado constan 256 direcciones posibles, ya que se utiliza un byte en el campo “Dirección” según la trama del mensaje descrita en el apartado anterior, las direcciones son distribuidas de la siguiente manera:

Dirección Broadcast	Direcciones Válidas de los Esclavos	Direcciones Reservadas	Dirección Inválida
254	1 - 250	0, 251 - 253	255

Figura 3-8: Direccionamiento

Realizado por: Flores, Pablo. 2017

Cabe resaltar que para la dirección *broadcast* se ha reservado la dirección “254”, mientras que la dirección “255” siempre se utilizará para indicar una dirección inválida. En el bus propietario los esclavos deben tener una dirección única y específica; en cambio el maestro no posee ninguna dirección. Adicionalmente se reservan 3 direcciones para futuras aplicaciones, estas son de la 251 a la 253.

3.2.3.5 Diagrama de Estado del Maestro

Los diagramas de estado muestran el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación en respuesta a eventos, junto con sus respuestas y acciones. También ilustran qué eventos pueden cambiar el estado de los objetos en cuestión.

En el desarrollo del protocolo del bus propietario, los diagramas de estado permiten explicar el comportamiento del maestro y de los esclavos, estos diagramas fueron elaborados en concordancia con el estándar UML, el cual señala lo siguiente: un sistema que se encuentra en un estado A y es sometido a un evento de activación, el sistema cambia a un estado B, sólo si una condición de protección es verdadera. Seguidamente, una acción es realizada.

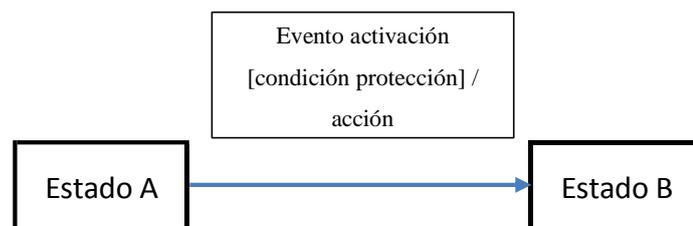


Figura 3-9: Bloques de Estado

Realizado por: Flores, Pablo. 2017

En la siguiente figura se puede observar el comportamiento del dispositivo maestro:

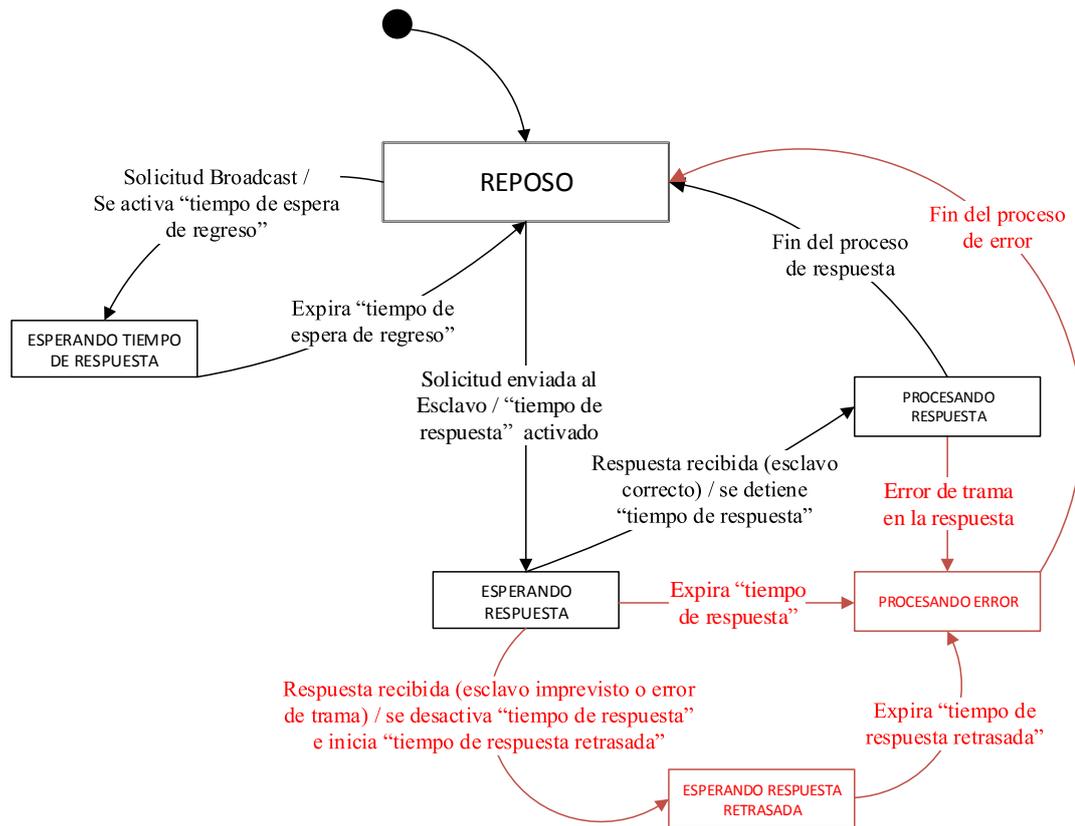


Figura 3-10: Diagrama de Estado del Maestro

Realizado por: Flores, Pablo. 2017

Del diagrama de estado del maestro, podemos decir que:

- El estado “reposo” hace referencia al estado inicial del maestro después del encendido, e indica que el dispositivo no tiene ninguna solicitud pendiente.
- Cuando el maestro se encuentra en el estado “reposo”, solamente ahí se puede enviar una solicitud, lo cual ocasiona que el dispositivo en primera instancia cambie dicho estado y posteriormente le impide enviar otra solicitud al mismo tiempo.
- En cuanto a las solicitudes *unicast*:
- ✓ Al enviarse una solicitud *unicast* a uno de los esclavos, el maestro realiza dos acciones: la primera es cambiar su estado de “reposo” al estado “esperando respuesta”; y la segunda es activar el tiempo “tiempo de respuesta”.

- ✓ La funcionalidad del “tiempo de respuesta” es para evitar o prevenir que el maestro permanezca por un periodo indefinido de tiempo en el estado “esperando respuesta”; el valor que otorgamos a este tiempo debe ser suficiente para que los esclavos procesen la solicitud y devuelvan la respuesta, el valor también dependerá del tipo de aplicación que se esté ejecutando.

- ✓ Para establecer un control y estar seguros de la validez de la respuesta enviada al maestro, antes de iniciar el procesamiento de los datos y en el momento de recibir la respuesta, ésta es chequeada o revisada por el maestro, byte por byte.

- ✓ Cuando la respuesta contiene un error en la trama del byte recibido, tal como: dato inválido, función, overrun, overflow, etc.; o en su defecto, la respuesta fue recibida por un esclavo imprevisto, el maestro procede a desactivar el tiempo “tiempo de respuesta” e inmediatamente activa el tiempo “tiempo de respuesta retrasada”, cambiando además su estado, de “esperando respuesta” a “esperando respuesta retrasada”.

- ✓ El tiempo “tiempo de respuesta retrasada” otorga al esclavo un tiempo adicional para que pueda terminar de enviar su respuesta y de esa forma regrese al estado “reposo”; si la respuesta no ha sido enviada completamente una vez que se haya cumplido este tiempo, se genera un error en la comunicación. El valor determinado para este tiempo depende de igual forma de la aplicación.

- ✓ Cuando las respuestas son recibidas de forma íntegra del esclavo o esclavos esperados, el maestro procede a desactivar el tiempo “tiempo de respuesta” y además cambia su estado, de “esperando respuesta” a “procesando respuesta”.

- ✓ Si se detecta un error en la trama del mensaje recibido el momento de procesar la respuesta, se genera un error; de lo contrario, el maestro finaliza con éxito el procesamiento de la respuesta y retorna al estado “reposo”.

- ✓ Un error también puede generarse cuando una respuesta no ha sido recibida durante el tiempo “tiempo de respuesta”.

- ✓ Al generarse un error por cualquiera de las circunstancias descritas en los puntos anteriores, el maestro cambia su estado actual, al estado “procesando error”. Este estado permite que el error sea revisado y tratado para posteriormente permitir al maestro regresar a su estado de reposo, el cual reenvía la solicitud al esclavo, si es que así procede. Se programa el número máximo de reintentos para el envío de solicitudes, dependiendo del tipo de aplicación.

- En cuanto a las solicitudes Broadcast:
 - ✓ El momento de enviar una solicitud broadcast, los esclavos no envían ninguna respuesta y el maestro activa el tiempo “tiempo de espera de regreso”, cambiando además su estado de “reposo” a “esperando tiempo de espera”.

 - ✓ El tiempo “tiempo de espera” permite a los esclavos terminar de procesar la solicitud actual y regresar al estado “reposo”, el momento que finaliza este tiempo, el maestro regresa al estado “reposo”, su valor es definido dependiendo de la aplicación.

3.2.3.6 Diagrama de Estado del Esclavo

En la siguiente figura se puede observar el comportamiento del dispositivo esclavo:

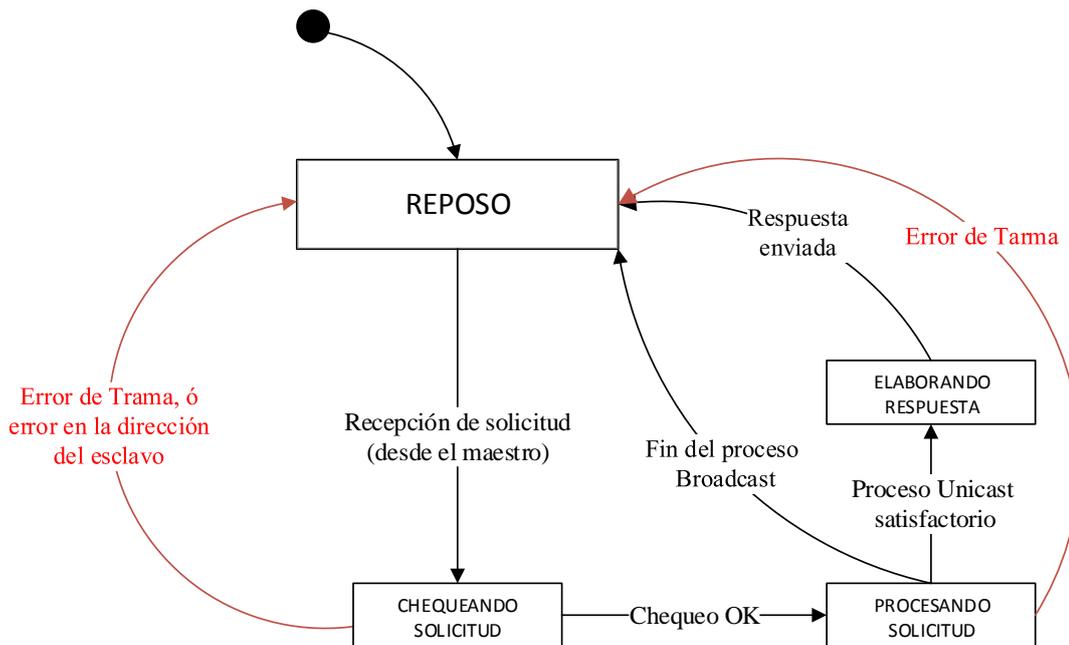


Figura 3-11: Diagrama de Estado del Esclavo

Realizado por: Flores, Pablo. 2017

Del diagrama de estado del esclavo, podemos decir que:

- El estado “reposo” hace referencia al estado inicial del esclavo, posterior al encendido, e indica que el dispositivo no tiene ninguna solicitud pendiente.
- Cuando el maestro envía una solicitud, el momento de estar siendo recibida dicha solicitud, el esclavo revisa o chequea, byte por byte, antes de iniciar el procesamiento de los datos, en el caso de existir algún error en la trama del byte tal como: dato inválido, función, overrun, overflow, etc.; o en su defecto el maestro no se dirigió a este esclavo; éste último no devuelve ninguna respuesta al maestro y retorna al estado “reposo”.
- Cuando una solicitud del maestro es recibida de forma completa por el esclavo, éste cambia su estado, de “chequeando solicitud” a “procesando solicitud” y por último ejecuta las acciones solicitadas por el maestro. En el caso de existir un error en la trama del mensaje recibido, el esclavo no devuelve ninguna respuesta al maestro y simplemente regresa al estado “reposo”.

- Si se trata de una solicitud *unicast* procesada de forma satisfactoria, el esclavo cambia de estado, de “procesando solicitud” a “elaborando respuesta”; estado en el cual el esclavo elabora una respuesta para enviarla al maestro, antes de regresar al estado “reposo”.
- Si se trata de una solicitud *broadcast* procesada de forma satisfactoria, el esclavo no devuelve ninguna respuesta al maestro y solamente regresa al estado “reposo”.

3.2.3.7 Diagrama de tiempo de la comunicación maestro – esclavo

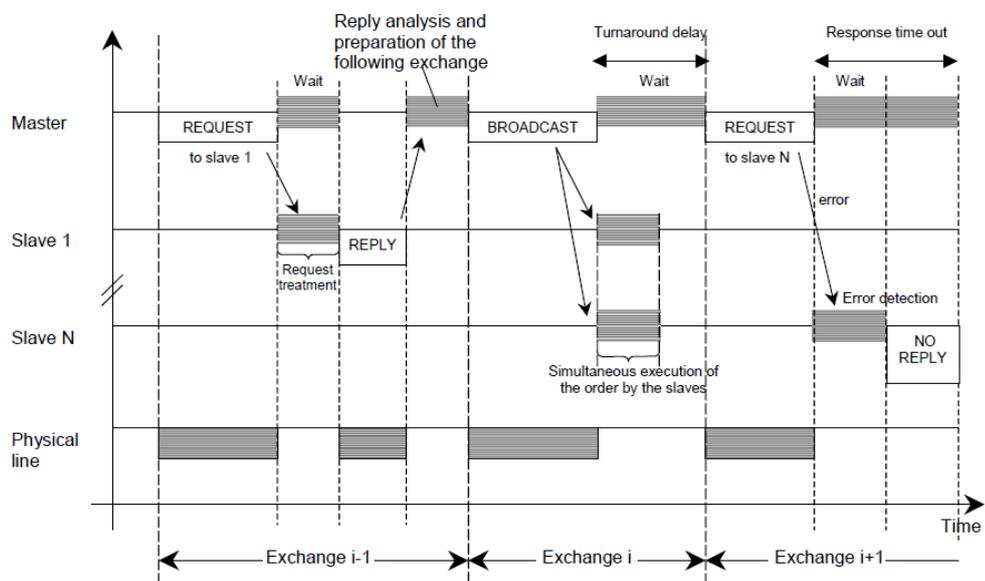


Figura 3-12: Diagrama de tiempo comunicación maestro - esclavo

Fuente: (Modbus-Organization, 2006)

En la figura 3-12 se observa el diagrama de tiempo de una comunicación maestro – esclavo, en la cual destacan tres escenarios típicos y las siguientes características:

- Características de la comunicación, tales como: longitud de la trama y el rendimiento, son las que determinan la duración de las fases: request, reply y broadcast.
- El tiempo que necesita un esclavo para procesar una solicitud, determina la duración de las fases: wait y treatment.

3.2.3.8 Transmisión de datos por RF

Para la transmisión de datos en el protocolo desarrollado se ha utilizado un bus RF, el cual es apto para realizar una conexión de dos tipos: punto – punto (maestro – esclavo) y punto – multipunto (maestro – multiesclavos), dicha conexión dependerá de los requerimientos de la aplicación.

El bus RF utilizado en las pruebas del protocolo, maneja transceivers nRF24L01, cuya conexión con el microcontrolador ATMEL del arduino está realizada con el estándar SPI (Interfaz Serial Periférica).

El formato de cada byte de la trama del mensaje es el siguiente:

Conexión	Sistema de Codificación	Bits por byte	
Maestro - Multiesclavo	8 Bits	8 Bits	Datos

Figura 3-13: Formato de cada byte de la trama del mensaje

Realizado por: Flores, Pablo. 2017

La figura 3-13 muestra el sistema de codificación para la conexión maestro – multiesclavo, considerando 8 Bits para los datos.

3.2.3.9 Funciones

El protocolo desarrollado implementa funciones específicas para realizar las transacciones maestro – esclavo, a través de RF. Las funciones implementadas son tres:

- Leer Registros
- Escribir Registros
- Eliminar Registros

3.2.3.10 Función Leer Registros

Esta función es utilizada en el protocolo desarrollado para leer los datos almacenados en la memoria del dispositivo esclavo, bloque a bloque. Su estructura se describe a continuación:

- Solicitud del maestro: se define el código de función, dirección del registro de inicio y el número de bytes a ser leídos.

Campo		Tamaño	Valor
Función	Código	1 Byte	0x64
Datos	Dirección del registro de inicio	1 Byte	
	Número de bytes	1 Byte	N

Figura 3-14: Solicitud del maestro (Función Leer)

Realizado por: Flores, Pablo. 2017

- Respuesta del esclavo: se define el código de función, dirección de registro de inicio, número de bytes a enviarse y el valor de cada byte.

Campo		Tamaño	Valor
Función	Código	1 Byte	0x64
Datos	Dirección del registro de inicio	1 Byte	
	Número de bytes	1 Byte	N
	Valor del byte	N x 1 Byte	

Figura 3-15: Respuesta del esclavo (Función Leer)

Realizado por: Flores, Pablo. 2017

Cabe resaltar que los campos de “Dirección de registro de inicio” y “Número de bytes”, tienen un tamaño de 1 byte, pero dependiendo de las necesidades de una aplicación, el tamaño puede incrementarse a 2 bytes o más.

Para aclarar la función de “Leer Registros”, se muestra un ejemplo a continuación:

SOLICITUD MAESTRO			RESPUESTA ESCLAVO		
Campo		Valor	Campo		Valor
Función	Código	0x64	Función	Código	0x64
Datos	Dirección del registro de inicio	0X0A	Datos	Dirección del registro de inicio	0X0A
	Número de bytes	0X04		Número de bytes	0X04
				Valor del byte (10)	0X42
				Valor del byte (11)	0X48
				Valor del byte (12)	0X24
				Valor del byte (13)	0X56

Figura 3-16: Ejemplo Función Leer

Realizado por: Flores, Pablo. 2017

Se puede resumir el ejemplo anterior de la siguiente manera: el maestro solicita la lectura de 4 (0x04) bytes a partir de la dirección 10 (0x0A); por consecuencia el esclavo responde primero devolviendo la dirección desde la cual empezó a leer los registros, inmediatamente el número de bytes a enviarse y por último los valores de los bytes ubicados en las direcciones 10, 11 12 y 13.

3.2.3.11 Función Escribir Registros

Esta función es utilizada en el protocolo desarrollado para escribir los registros almacenados en la memoria del dispositivo esclavo, bloque a bloque. Su estructura se describe a continuación:

- Solicitud del maestro: se define el código de función, dirección del registro de inicio, número de bytes que se van a escribir y el valor de cada byte.

Campo		Tamaño	Valor
Función	Código	1 Byte	0x65
Datos	Dirección del registro de inicio	1 Byte	
	Número de bytes	1 Byte	N
	Valor del byte	N x 1 Byte	N

Figura 3-17: Solicitud del maestro (Función Escribir)

Realizado por: Flores, Pablo. 2017

- Respuesta del esclavo: se define el código de función, dirección de registro de inicio y el número de bytes que se han sido escritos.

Campo		Tamaño	Valor
Función	Código	1 Byte	0x65
Datos	Dirección del registro de inicio	1 Byte	
	Número de bytes	1 Byte	N

Figura 3-18: Respuesta del esclavo (Función Escribir)

Realizado por: Flores, Pablo. 2017

Cabe resaltar que al igual que en la función “Leer Registros”, los campos de “Dirección de registro de inicio” y “Número de bytes”, tienen un tamaño de 1 byte, pero dependiendo de las necesidades de una aplicación, el tamaño puede incrementarse a 2 bytes o más.

Para aclarar la función “Escribir Registros”, se muestra un ejemplo a continuación:

SOLICITUD MAESTRO			RESPUESTA ESCLAVO		
Campo		Valor	Campo		Valor
Función	Código	0x65	Función	Código	0x65
Datos	Dirección del registro de inicio	0X04	Datos	Dirección del registro de inicio	0X04
	Número de bytes	0X05		Número de bytes	0X05
Datos	Valor del byte (04)	0X39			
	Valor del byte (05)	0X0B			
	Valor del byte (06)	0X24			
	Valor del byte (07)	0XA2			
	Valor del byte (08)	0XB1			

Figura 3-19: Ejemplo Función Escribir

Realizado por: Flores, Pablo. 2017

Se puede resumir el ejemplo anterior de la siguiente manera: el maestro envía la solicitud de escribir 5 (0x05) bytes a partir de la dirección 4 (0x04); por consecuencia los bytes 0x39, 0x0B, 0x24, 0xA2 y 0xB1 deberán ser escritos en el esclavo, en las direcciones 04, 05, 06, 07 y 08, respectivamente. Posterior a la solicitud, el esclavo responde los siguientes campos: dirección del

registro de inicio, es decir desde donde empezó a escribir los registros y también indica el número total de bytes escritos.

3.2.3.12 Función Eliminar Registros

Esta función es utilizada en el protocolo desarrollado para eliminar los registros almacenados en la memoria del dispositivo esclavo, bloque a bloque. Su estructura se describe a continuación:

- Solicitud del maestro: se define el código de función, dirección del registro de inicio y el número de bytes que se desean eliminar

Campo		Tamaño	Valor
Función	Código	1 Byte	0x66
Datos	Dirección del registro de inicio	1 Byte	
	Número de bytes	1 Byte	N

Figura 3-20: Solicitud del maestro (Función Eliminar)

Realizado por: Flores, Pablo. 2017

- Respuesta del esclavo: se define el código de función, dirección de registro de inicio y el número de bytes que se han eliminado.

Campo		Tamaño	Valor
Función	Código	1 Byte	0x66
Datos	Dirección del registro de inicio	1 Byte	
	Número de bytes	1 Byte	N

Figura 3-21: Respuesta del esclavo (Función Eliminar)

Realizado por: Flores, Pablo. 2017

Cabe resaltar que al igual que en la función “Leer Registros” y en la función “Escribir Registros”, los campos de “Dirección de registro de inicio” y “Número de bytes”, tienen un tamaño de 1 byte,

pero dependiendo de las necesidades de una aplicación, el tamaño puede incrementarse a 2 bytes o más.

Para aclarar la función “*Eliminar Registros*”, se muestra un ejemplo a continuación:

SOLICITUD MAESTRO			RESPUESTA ESCLAVO		
Campo		Valor	Campo		Valor
Función	Código	0x65	Función	Código	0x65
Datos	Dirección del registro de inicio	0X01	Datos	Dirección del registro de inicio	0X01
	Número de bytes	0X06		Número de bytes	0X06

Figura 3-22: Ejemplo Función Eliminar

Realizado por: Flores, Pablo. 2017

Se puede resumir el ejemplo anterior de la siguiente manera: el maestro envía la solicitud de eliminar 6 (0x05) bytes a partir de la dirección 01 (0x01). Posterior a la solicitud, el esclavo responde con los siguientes campos: dirección del registro de inicio, es decir desde donde empezó a eliminar los registros y el número total de bytes eliminados.

3.2.3.13 *Chequeo de Redundancia Cíclica (CRC)*

El protocolo desarrollado como se ha podido observar incluye en la trama del mensaje un campo de chequeo de errores, el mismo que se utiliza para validar la información de un mensaje cuando es recibido por cualquier dispositivo; esto se puede conseguir gracias a que la totalidad del mensaje es sometido a un algoritmo de comprobación de redundancia cíclica.

El campo CRC tiene un total 16 bits, se encuentra agregado al final del mensaje, implementado como dos bytes de 8 bits. Se tiene un byte de orden inferior (CRCL) el cual es agregado en primer lugar, y a continuación un byte de orden superior (CRCH), quien se convierte en el último byte del mensaje, a ser enviado.

Los dispositivos que transmiten el mensaje, calculan el valor del CRC y lo añaden al mensaje, previo a ser enviado; de igual forma los dispositivos receptores, el momento de recibir el mensaje, calculan nuevamente el valor del CRC, para poder compararlo con el valor recogido en el campo CRC. Si los dos valores son iguales, se valida la información e inmediatamente se procesa; pero si los dos valores no son iguales, se concluye que la información tiene errores y por ende no puede ser procesada.

3.2.3.14 *Algoritmo de cálculo del CRC*

Para calcular el valor del CRC se utiliza los 8 bits de cada byte, y se considera el contenido del mensaje entero. El diagrama de flujo del algoritmo de cálculo se muestra a continuación:

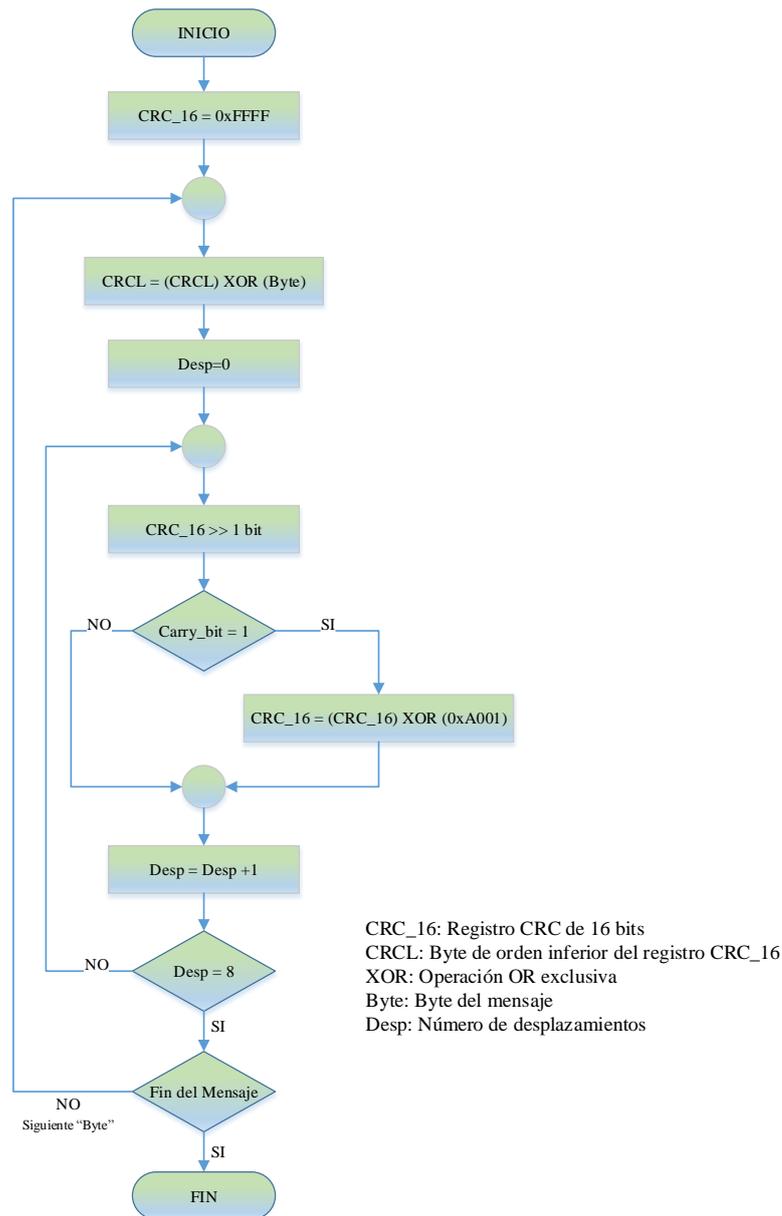


Figura 3-23: Diagrama de Flujo Algoritmo de Cálculo CRC

Realizado por: Flores, Pablo. 2017

Según lo que se puede observar en el diagrama de flujo, podemos describir los pasos del algoritmo de la siguiente forma:

- Se inicializa el registro CRC de 16 bits (CRC_16), con el valor 0xFFFF (hex); posteriormente los bits se organizan, de izquierda a derecha, del bit más significativo (MSB) al bit menos significativo (LSB).
- Realizar una operación OR exclusiva (XOR) entre el byte CRCL (byte de orden inferior del registro CRC_16) y el primer byte del mensaje. El resultado debe cargarse en CRCL.

- c) Desplazar el registro CRC_16 un bit hacia la derecha, es decir hacia el bit menos significativo, y extraer el bit desplazado (Carry_bit). El bit más significativo se rellena con cero.

- d) Analizar el bit desplazado, es decir el Carry_bit:
Si Carry_bit = 0, repetir el paso “c”, el cual consiste en generar otro desplazamiento.
Si Carry_bit = 1, realizar una operación XOR entre el registro CRC_16 y el valor constante 0xA001 (hex). El resultado debe cargarse en el registro CRC_16.

- e) Repetir los pasos “c” y “d” hasta tener un total de 8 desplazamientos y de esta forma procesar por completo el byte.

- f) Para los siguientes bytes del mensaje se deben repetir los pasos “b”, “c”, “d” y “e”, hasta procesar de forma completa todos los bytes.

- g) El valor final calculado del registro CRC_16 corresponde al valor del CRC a enviarse en la trama.

3.3 Desarrollo del Software

El desarrollo del software hace referencia a los algoritmos que han sido elaborados para la programación de la red de microcontroladores Arduinos (maestro - esclavos), mismos que utilizan el protocolo propietario desarrollado para las comunicaciones.

Para describir y explicar los algoritmos de cada uno de los elementos que conforman la red se utilizan diagramas de flujo cuya secuencia lógica radica en una sucesión de procesos que abarcan una o varias líneas de códigos, cabe resaltar que algunos subprocesos no se los considera directamente en los diagramas sino más bien están implícitos, todo cuanto para simplificar de alguna forma el flujo grama y optimizar su comprensión.

Los diagramas de flujo que se han elaborado son en base a los siguientes elementos:

- Microcontrolador arduino maestro
- Microcontrolador arduino esclavo CSE (Contactos Secos de Entrada)
- Microcontrolador arduino esclavo CSS (Contactos Secos de Salida)

Adicionalmente se desarrolló el diagrama de flujo para representar el algoritmo utilizado en el funcionamiento de en un teclado celular, al cual se lo considera como un teclado local del microcontrolador arduino maestro.

Todos los diagramas de flujo poseen variables y constantes de carácter global y local, aquellas que son globales son enlistadas y declaradas al inicio de cada diagrama por el hecho de ser visibles y utilizadas durante todo el algoritmo; por otro lado las variables y constantes locales, se declaran en el interior de cada uno de los diferentes procesos en las que son utilizadas.

Cuando se da la comunicación entre los diferentes elementos de la red, tanto los mensajes que se desean enviar o recibir, como la transmisión y recepción de la trama de mensajes, byte por byte,

son elaborados y procesados, según corresponda el caso, en el interior de la rutina principal (main).

Las tareas que se generan en la red, se procesan utilizando criterios de prioridad mediante banderas y pilas de solicitudes (Stacks), lo cual permite que el flujo normal del algoritmo no se detenga. El procesamiento de las solicitudes se lleva a cabo en la rutina principal (main) y su almacenamiento obedece al orden FIFO (primero en entrar, primero en salir).

3.3.1 Diagrama de Flujo Microcontrolador Arduino Maestro

El diagrama de flujo que se observa a continuación, describe el algoritmo desarrollado para el funcionamiento del maestro, empleando la siguiente nomenclatura:

COM: Comunicación

CSE: Contacto Seco de Entrada

CSS: Contacto Seco de Salida

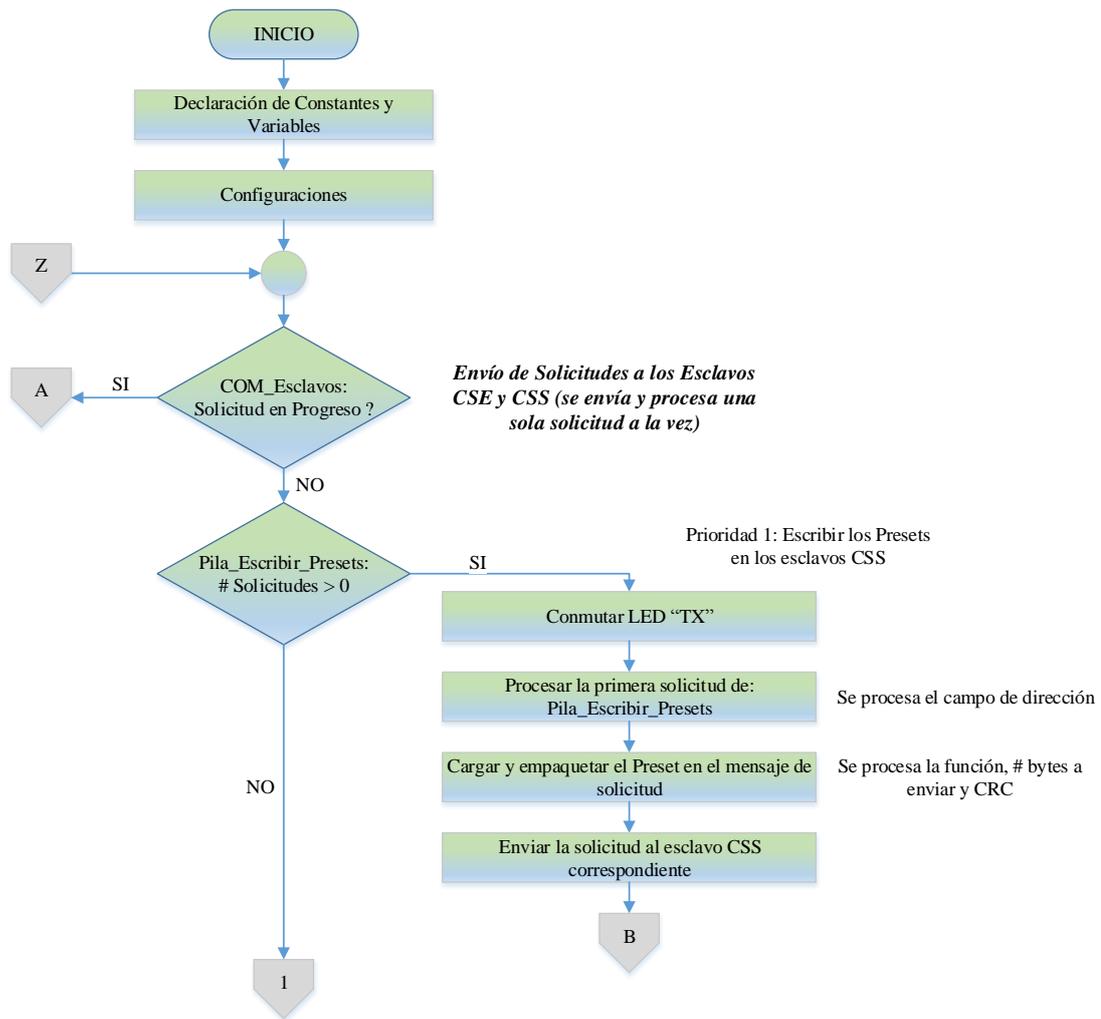
CRC: Chequeo de Redundancia Cíclica

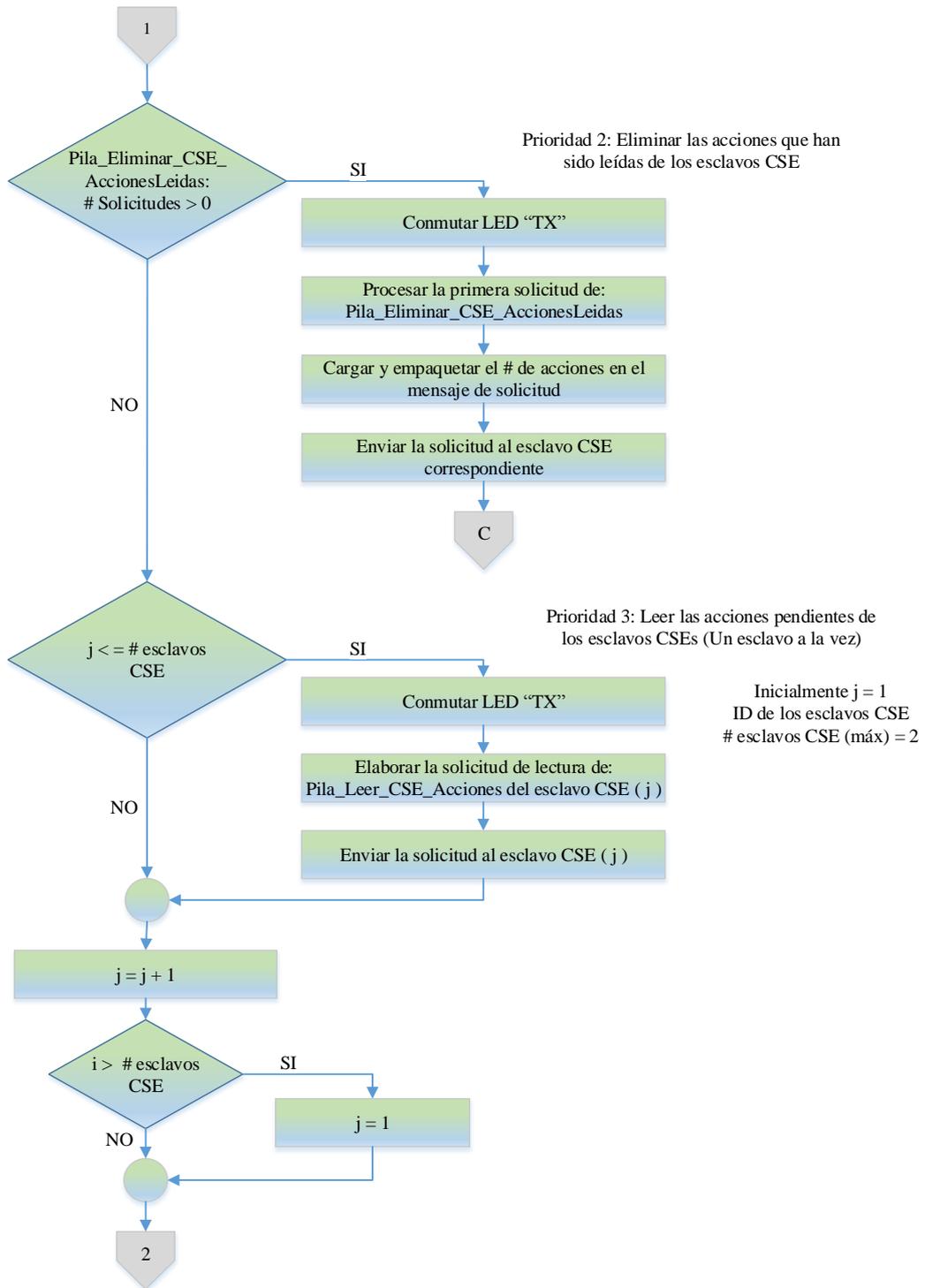
LED: Diodo Emisor de Luz

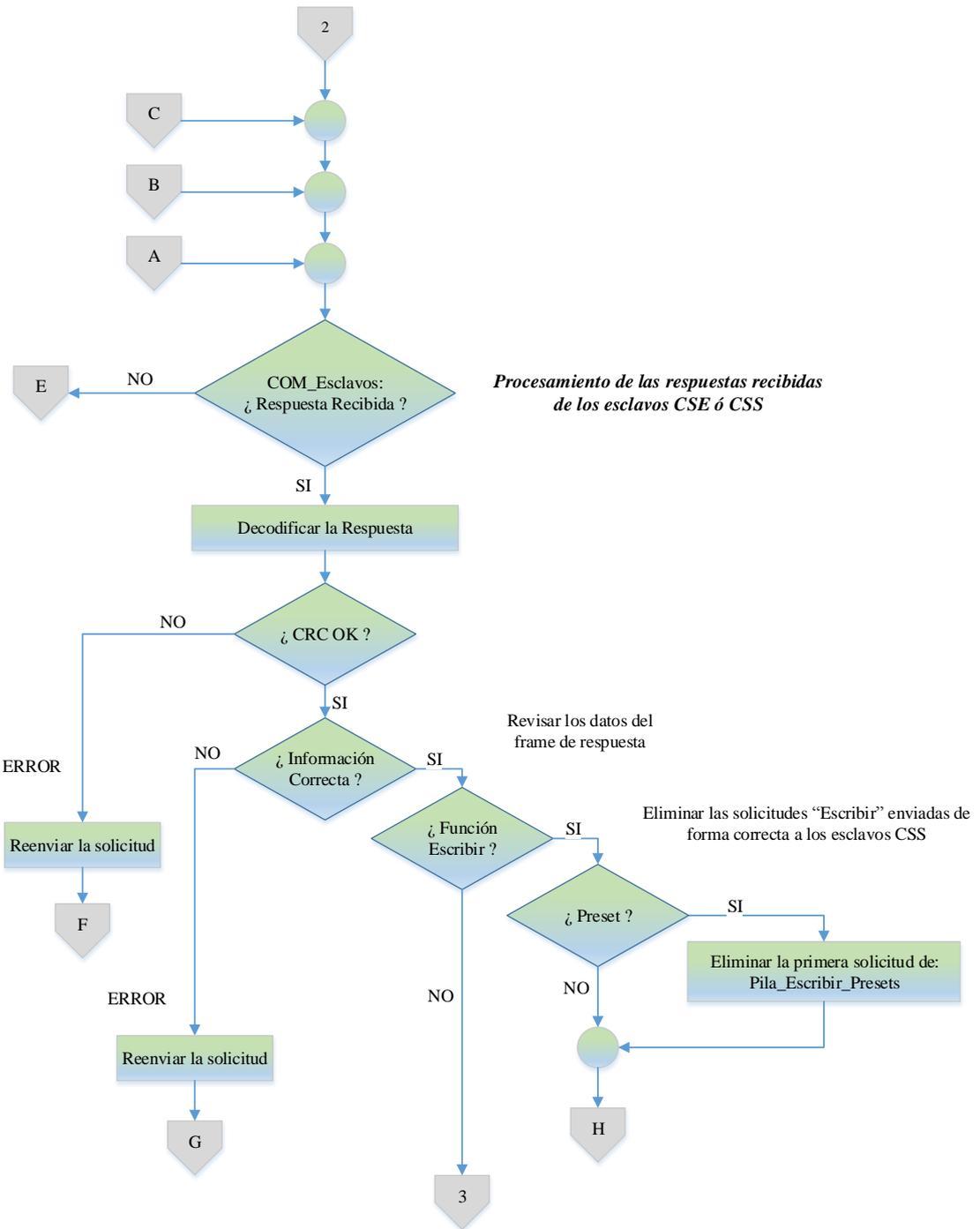
TX: Transmisión

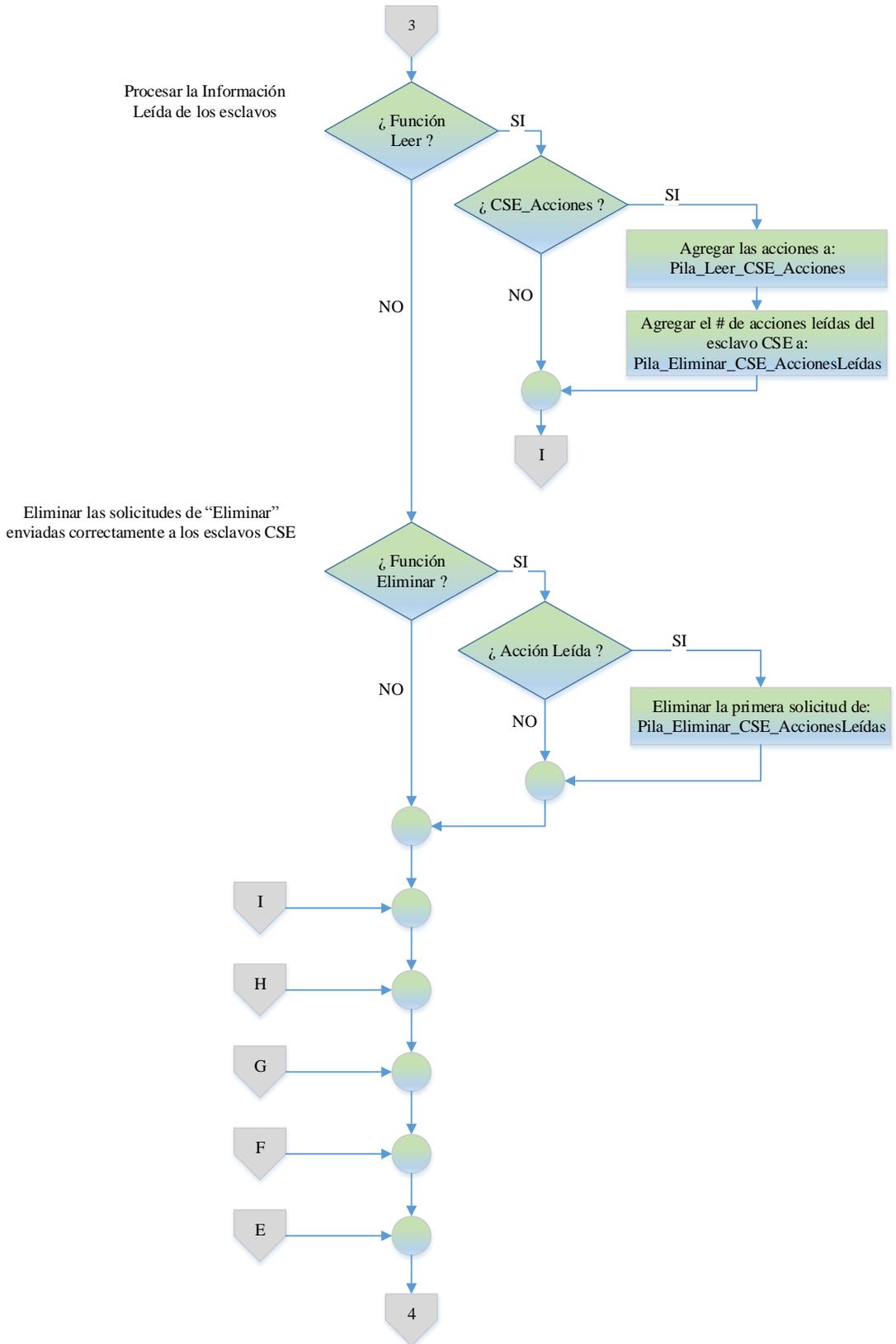
i, j: Variables

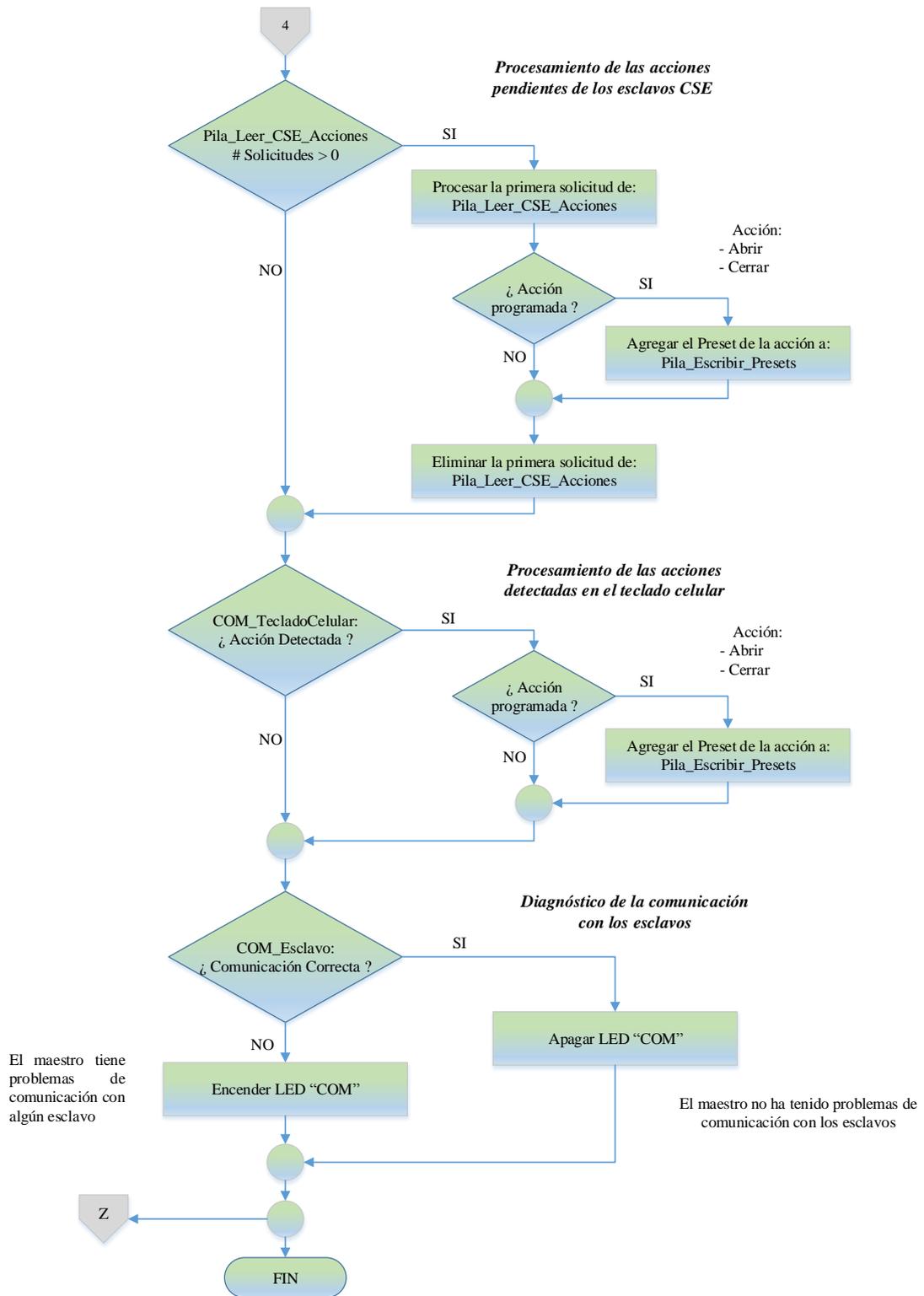
#: Número de











3.3.2 Diagrama de Flujo Microcontrolador Arduino Esclavo CSE

El diagrama de flujo que se observa a continuación, describe el algoritmo desarrollado para el funcionamiento del microcontrolador arduino esclavo CSE (Contactos Secos de Entrada), empleando la siguiente nomenclatura:

COM: Comunicación

CSE: Contacto Seco de Entrada

ID: Identificar

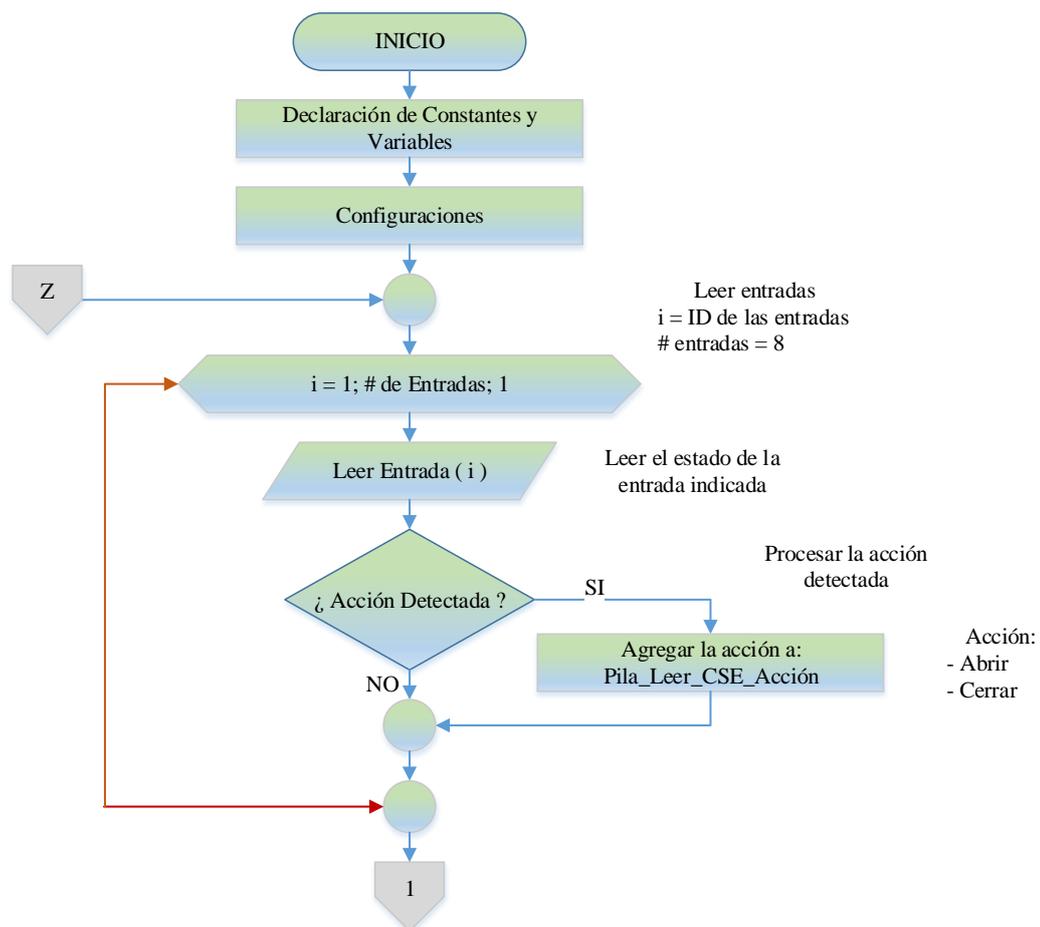
LED: Diodo Emisor de Luz

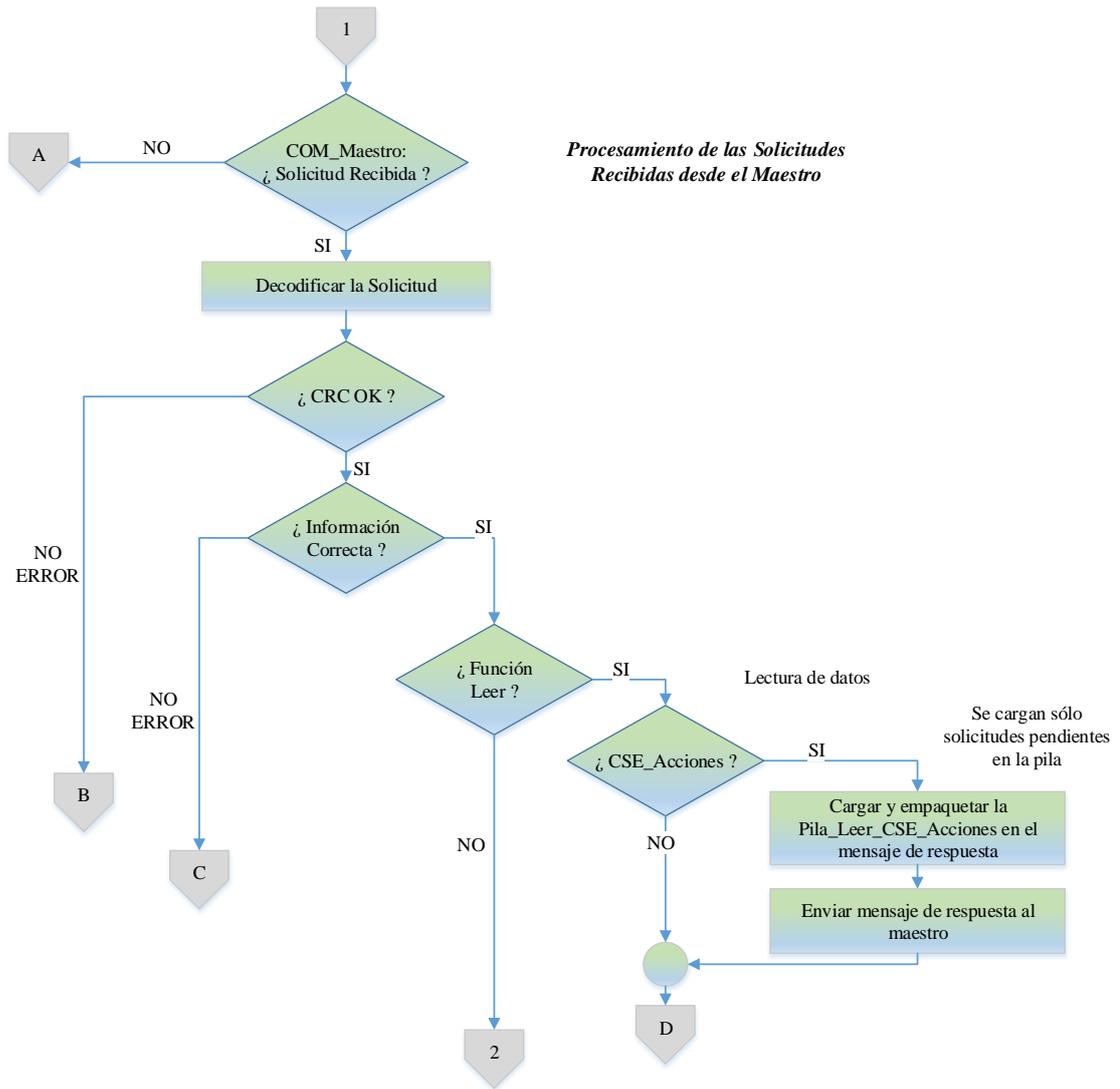
RF: Radio Frecuencia

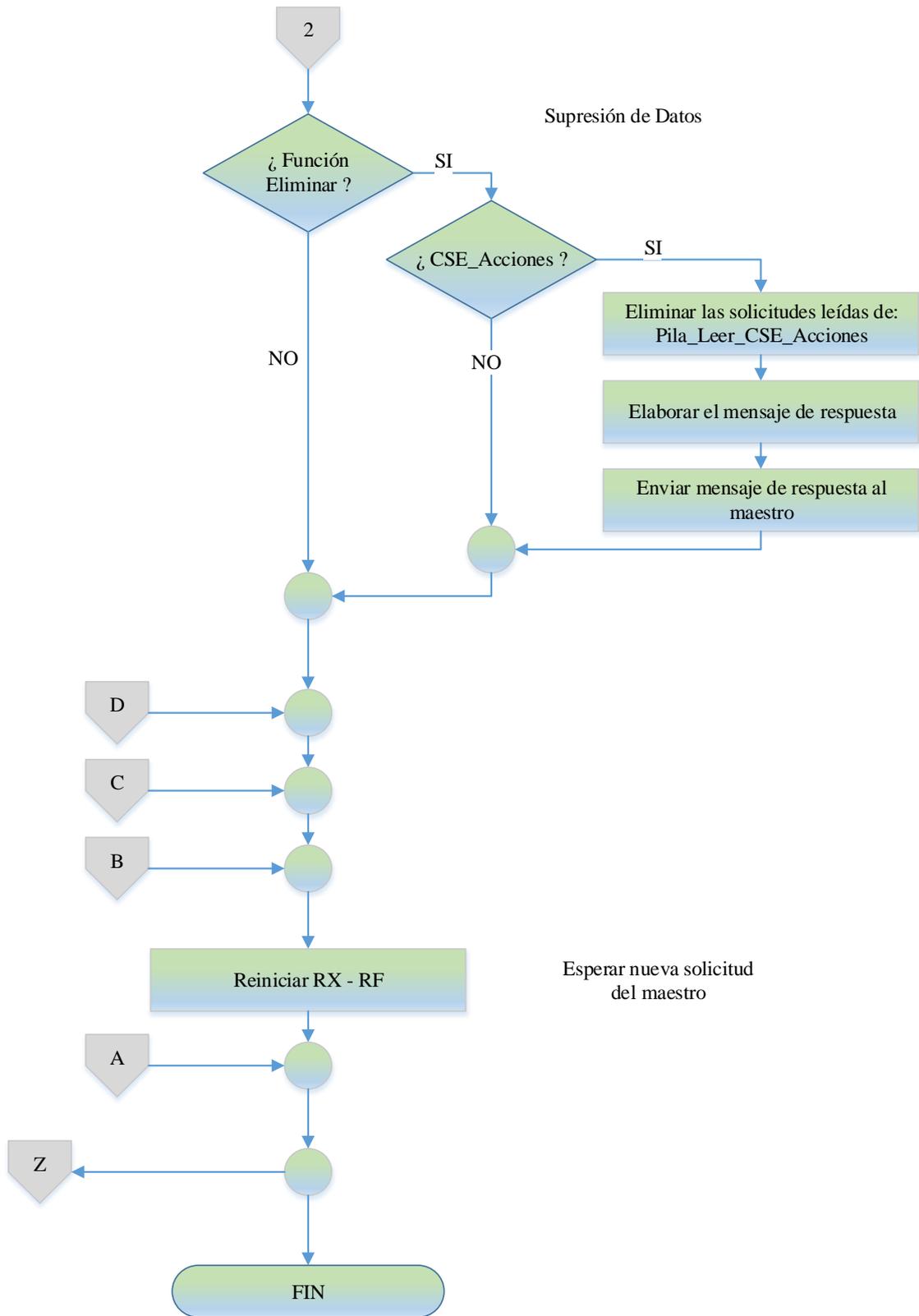
RX: Recepción

i: Variable

#: Número de







3.3.3 Diagrama de Flujo Microcontrolador Arduino Esclavo CSS

El diagrama de flujo que se observa a continuación, describe el algoritmo desarrollado para el funcionamiento del microcontrolador arduino esclavo CSS (Contactos Secos de Salida), empleando la siguiente nomenclatura:

COM: Comunicación

CSS: Contacto Seco de Salida

EstadoA: Estado Actual

EstadoF: Estado Futuro

ID: Identificar

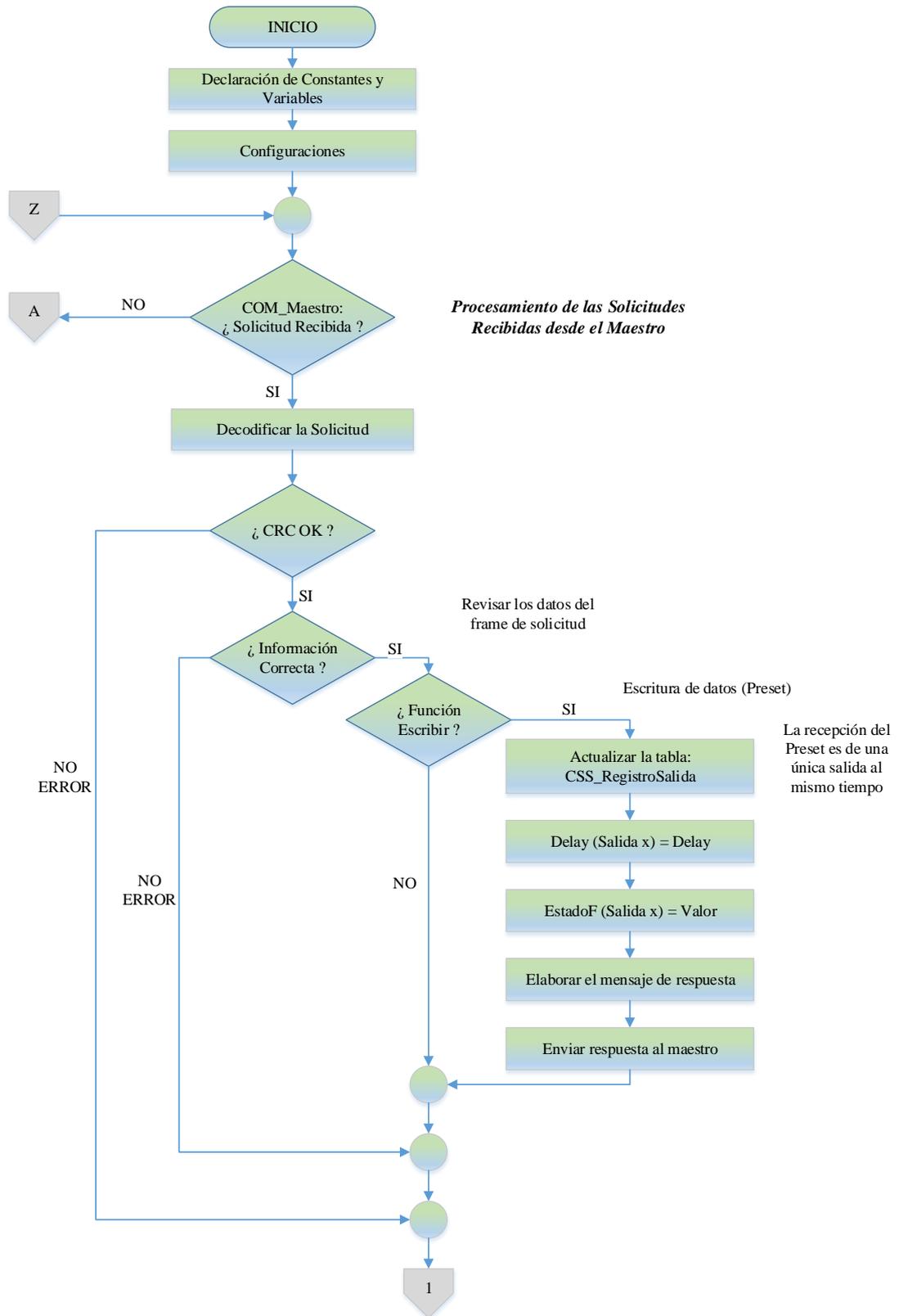
LED: Diodo Emisor de Luz

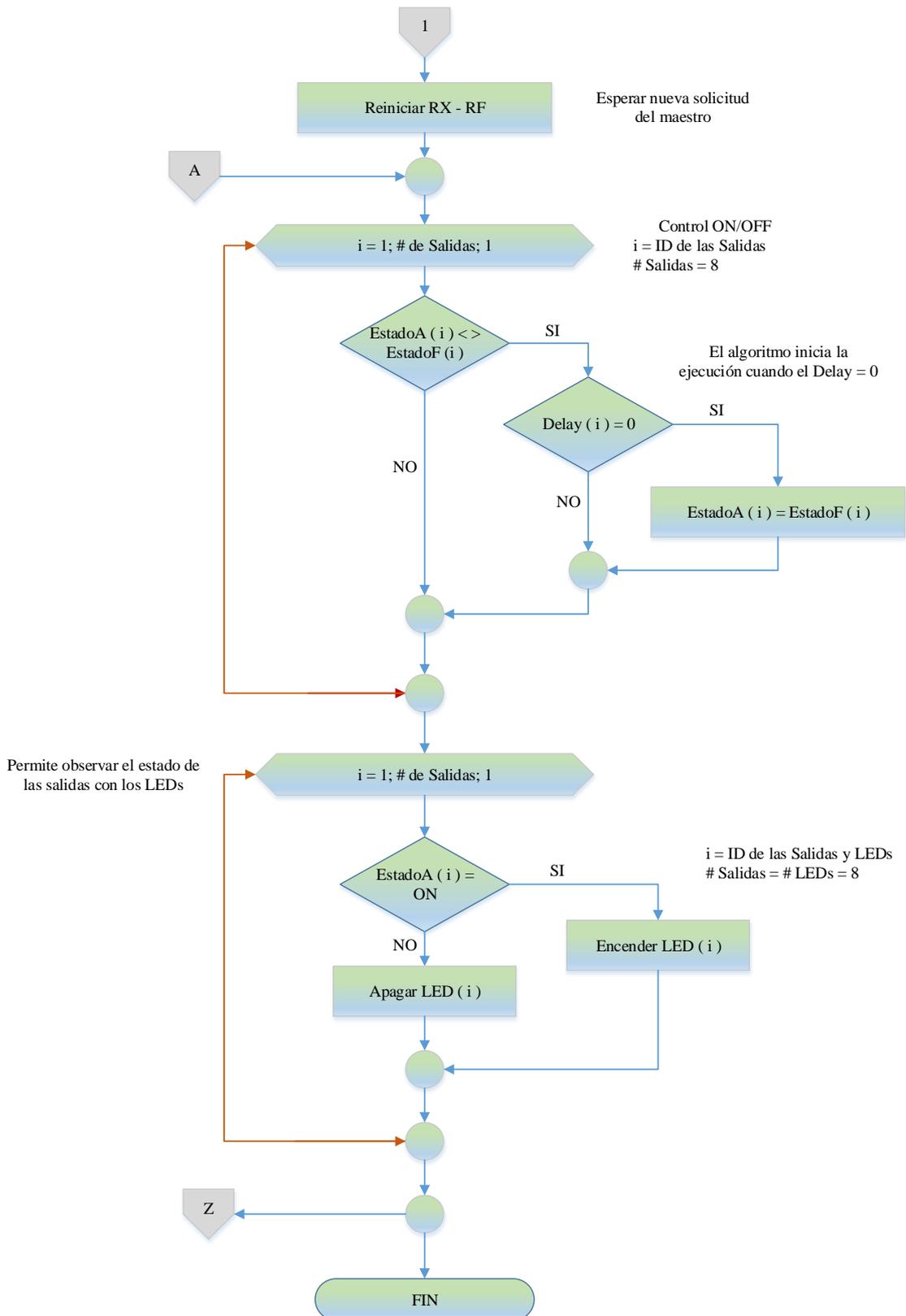
RF: Radio Frecuencia

RX: Recepción

i: Variable

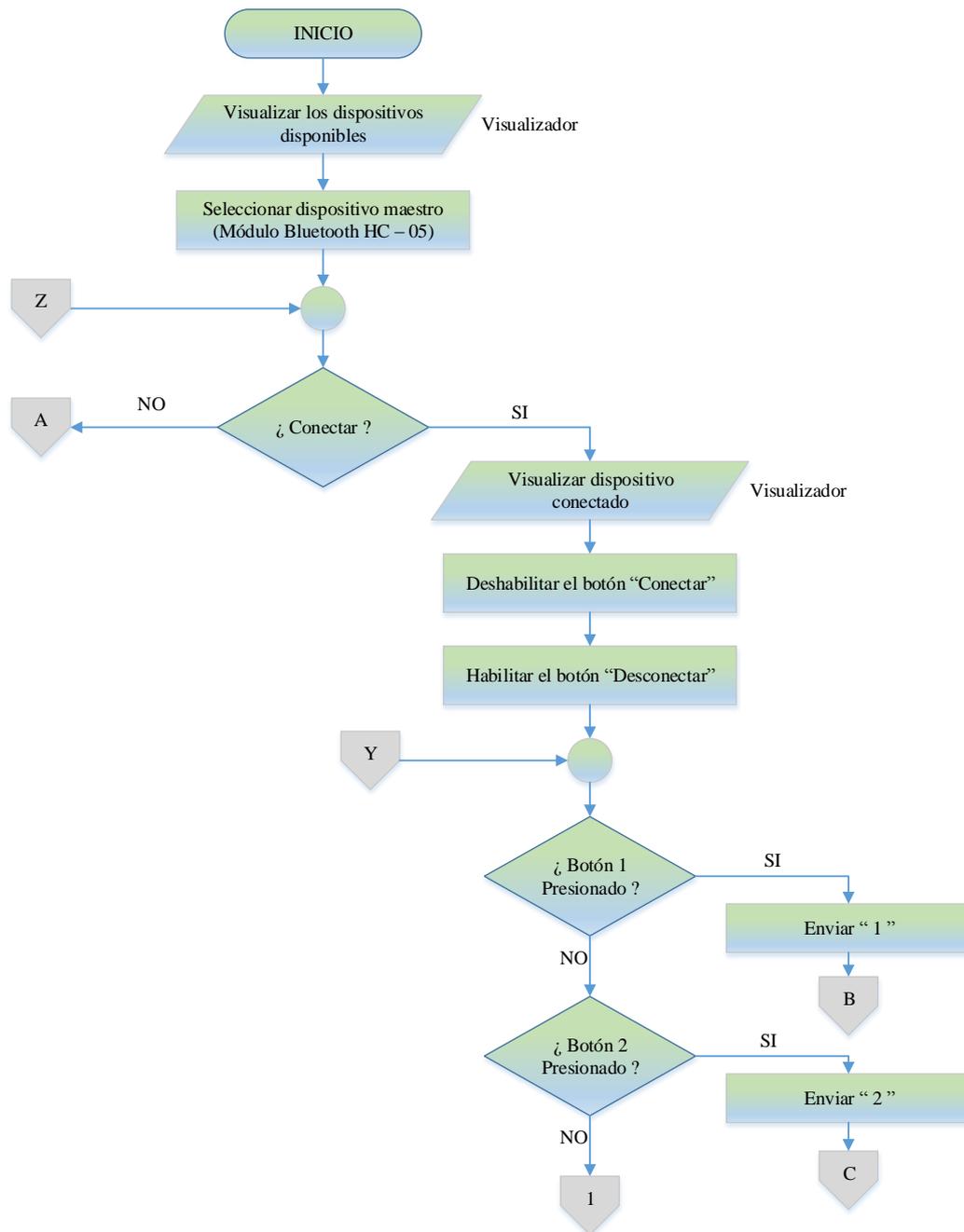
#: Número de

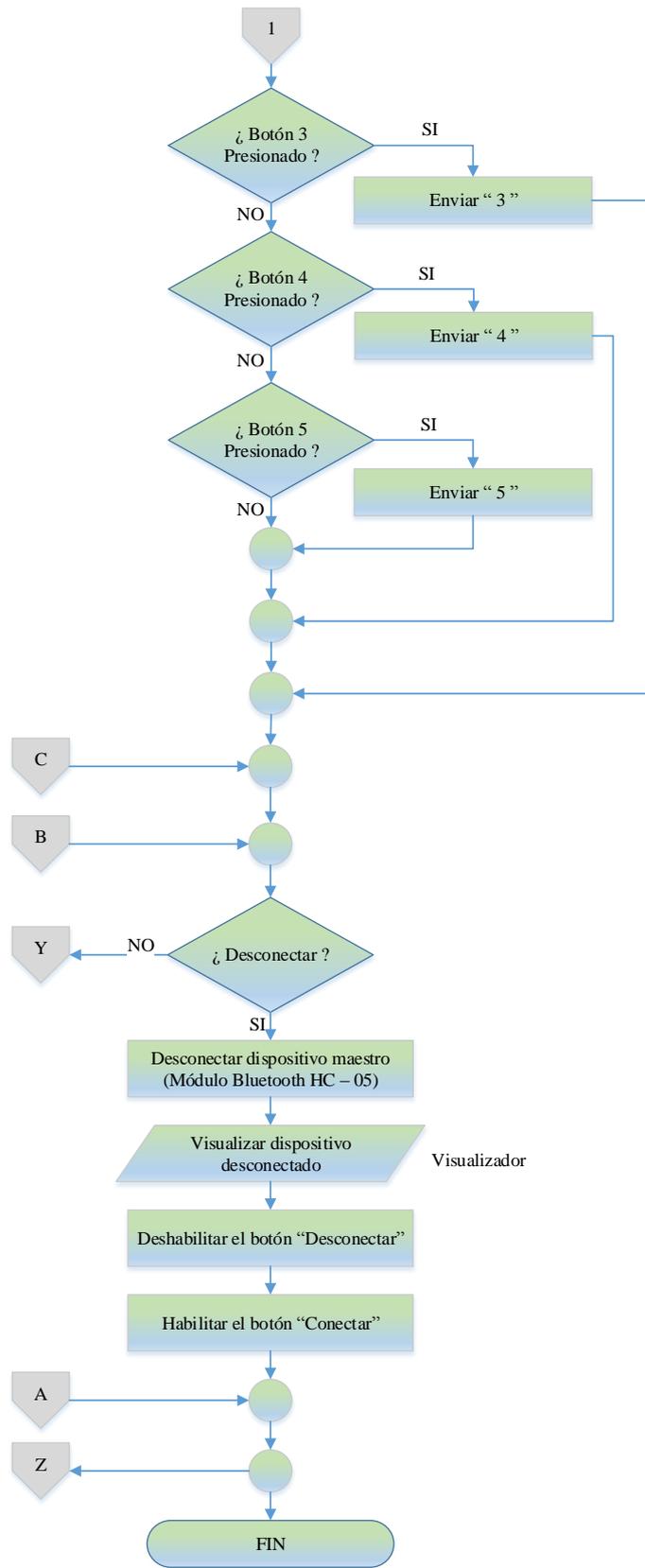




3.3.4 Diagrama de Flujo Teclado de Celular

El diagrama de flujo que se observa a continuación, describe el algoritmo desarrollado para el funcionamiento del teclado celular, al cual se lo considera como un teclado local del microcontrolador arduino maestro:





3.4 Implementación del Hardware

La implementación del hardware hace referencia a la arquitectura de la red, su funcionamiento y las características de cada uno de sus elementos conformantes. Después de la descripción detallada de lo antes mencionado se presenta una aplicación domótica en la cual se implementa el bus propietario desarrollado.

3.4.1 Arquitectura de la Red

La arquitectura de la red elaborada es centralizada, resumiéndose en una configuración maestro – esclavo, en la cual se tiene un único maestro considerado como dispositivo central y seis esclavos que son los elementos de entrada y salida. Según se puede observar en la siguiente figura:

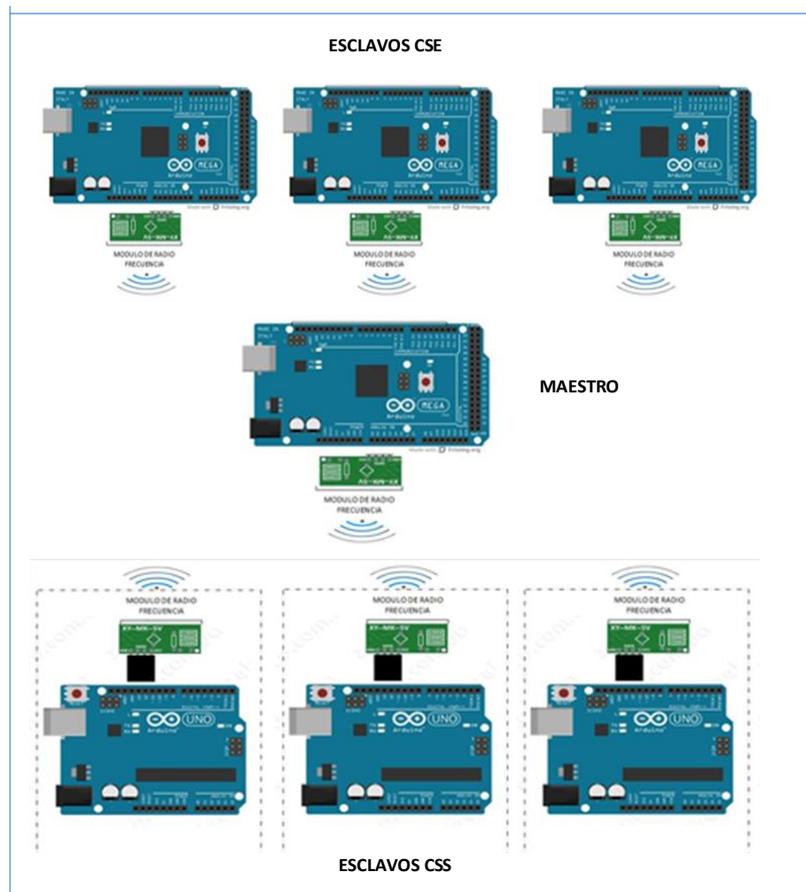


Figura 3-24: Arquitectura de la Red

Realizado por: Flores, Pablo. 2017

En la arquitectura implementada, el maestro está configurado de tal manera que recepte las diferentes solicitudes, procese las mismas y envíe las respectivas órdenes a los esclavos. Por otro lado quienes reciben las solicitudes son los elementos de entrada y quienes ejecutan las órdenes son los elementos de salida; es decir estos dos dispositivos tienen la función de interactuar con el exterior.

Entre el esclavo CSE (Contactos Secos de Entrada), el maestro y el esclavo CSS (Contactos Secos de Salida) existe un flujo de información bidireccional, tal como se aprecia en la figura 3-25:

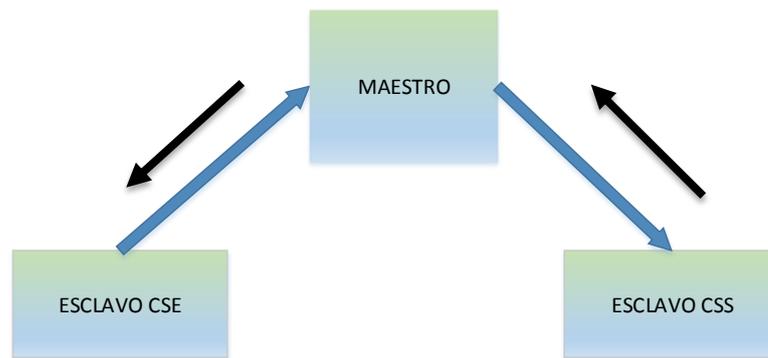


Figura 3-25: Flujo de Información Red Centralizada

Realizado por: Flores, Pablo. 2017

La información de color azul que circula de izquierda a derecha, es decir, desde el esclavo CSE hacia el maestro y posteriormente hacia el esclavo CSS es la que permite procesar y realizar las solicitudes generadas por los usuarios.

Por otro lado la información de color negro que circula de derecha a izquierda realizada por los pares: esclavo CSS – maestro y maestro – esclavo CSE; es la responsable de lo siguiente:

- En el par de comunicación maestro – esclavo CSE, el maestro mediante un algoritmo programado realiza un monitoreo de la red, preguntando si los esclavos CSE tienen solicitudes pendientes.
- En el par de comunicación maestro – esclavo CSS, el maestro gracias a una retroalimentación puede conocer si las órdenes enviadas fueron o no ejecutadas.

3.4.2 Funcionamiento de la Red

En la red desarrollada como lo hemos visto en los apartados anteriores existen tres elementos: maestro, esclavo CSE y esclavo CSS. La interacción entre los tres elementos permite llevar a cabo las tareas encomendadas por los usuarios.

Las tareas se definen como una actividad específica a ser realizada, por ejemplo, la apertura automática de una puerta, el encendido de luces, etc.

Para que todos los elementos de una red trabajen de forma coordinada se ha delimitado funciones específicas para cada uno de ellos:

- La función principal del esclavo CSE es la de activar una tarea de la red, la misma que puede ser realizada directamente por un usuario al presionar un interruptor o puede ser automática como la activación de un sensor de movimiento
- Las funciones principales del maestro son: monitorizar a los esclavos CSE, procesar las tareas activadas por los mismos y ordenar a los esclavos CSS las acciones que deben ser realizadas.
- La función principal del esclavo CSS se enmarca específicamente en ejecutar la tarea según las órdenes recibidas.

EL procedimiento que se da para ejecutar una tarea cuando se presiona un interruptor se observa en la figura 3-26:



Figura 3-26: Procedimiento ejecución de tareas

Realizado por: Flores, Pablo. 2017

Los pasos del procedimiento de ejecución son los siguientes:

- a) Al presionarse un interruptor, el esclavo CSE almacena la acción detectada en una pila de lectura y espera a que el maestro le solicite enviar las acciones detectadas.
- b) EL maestro procesa la solicitud y envía la orden al esclavo CSS
- c) El esclavo CSS cumple la orden y responde al maestro comunicando el cumplimiento de la misma.
- d) El maestro finaliza el proceso enviando un mensaje al esclavo CSE, indicando el cumplimiento de la tarea; para que de esta forma elimine las acciones detectadas de la pila de lectura.

Cabe resaltar que en el caso de existir una interrupción en la comunicación, ya sea por ruido, pérdida de información o cualquier otra, el proceso se repetirá en algunas ocasiones, lo cual garantiza una comunicación eficiente y confiable.

3.4.3 Elementos de la Red:

En éste apartado se da a conocer todos y cada uno de los elementos que intervienen en la red, describiendo algunas características básicas y un resumen de las especificaciones técnicas de cada uno de ellos.

3.4.3.1 Maestro

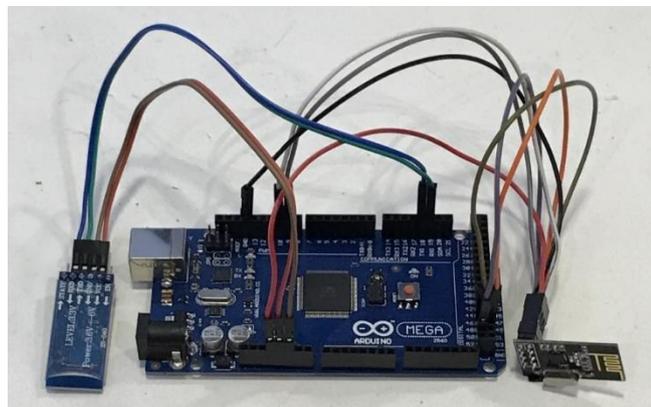


Figura 3-27: Maestro

Realizado por: Flores, Pablo. 2017

El maestro es el elemento central de la red, realizando el arbitraje de la misma. Controla a los dispositivos esclavos CSE y esclavos CSS, con los cuales se comunica a través de radio frecuencia. El número máximo de esclavos que pueden conectarse a la red es de 250 y la velocidad de comunicación se realiza a 2Mbps, pudiendo ser esta velocidad mayor o menor acorde sean los requerimientos de diseño o de hardware.

El envío/recepción de tramas de datos desde (o hacia) el dispositivo maestro se establece a través de una comunicación SPI (Interfaz Serial Periférica) entre el microcontrolador ATMEGA2560 y el módulo RF nRF24L01.

El maestro posee además un módulo Bluetooth, que sirve para conectarse a un teclado celular, el cual funciona como un teclado local que permite comandar acciones en la aplicación domótica desarrollada.

La estructura interna del maestro se muestra en el siguiente diagrama de bloques:

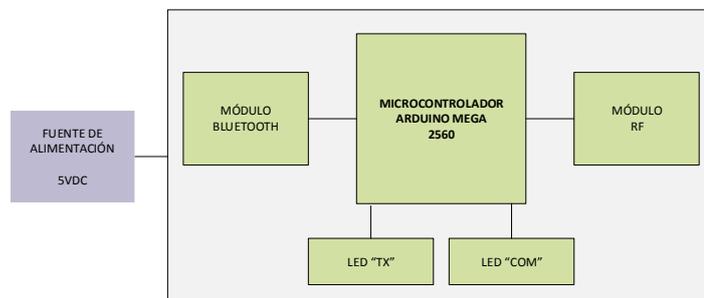


Figura 3-28: Diagrama de Bloques del Maestro

Realizado por: Flores, Pablo. 2017

Como se puede observar en el diagrama de bloques de la Figura 3-28, las partes del maestro son:

- Fuente de 5 V. de Corriente Continua
- Microcontrolador Arduino Mega: ATMEGA 2560
- Módulo RF: transreceptor nRF24L01 2.4GHz
- Módulo Bluetooth: HC-05
- Led "TX" que indica la transmisión de datos
- Led "COM" que indica problemas en las comunicaciones

Las especificaciones técnicas generales de las partes del maestro se exponen en la siguiente tabla:

3-1: Especificaciones Técnicas Maestro

Parte	Parámetro	Valor / Descripción
Fuente de Alimentación	Voltaje de salida	5 VDC
	Corriente de salida	2 A.
Microcontrolador Arduino MEGA	Numeración	ATMEGA 2560
	Voltaje de entrada	5 VDC
	Flash	256 KB
	RAM	8 KB
Módulo RF	Transreceptor	nRF24L01
	Voltaje de entrada	3.3 VDC
	Banda de Frecuencia	2.4 GHz
Módulo Bluetooth	Numeración	HC-05
	Voltaje de entrada	5 VDC
	Banda de Frecuencia	2.4 GHz
Led "TX"	Diagnóstico	Verde. Indica transmisión de datos
Led "COM"	Diagnóstico	Rojo. Indica problemas comunicación

Realizado por: Flores, Pablo. 2017

3.4.3.2 Esclavo CSE

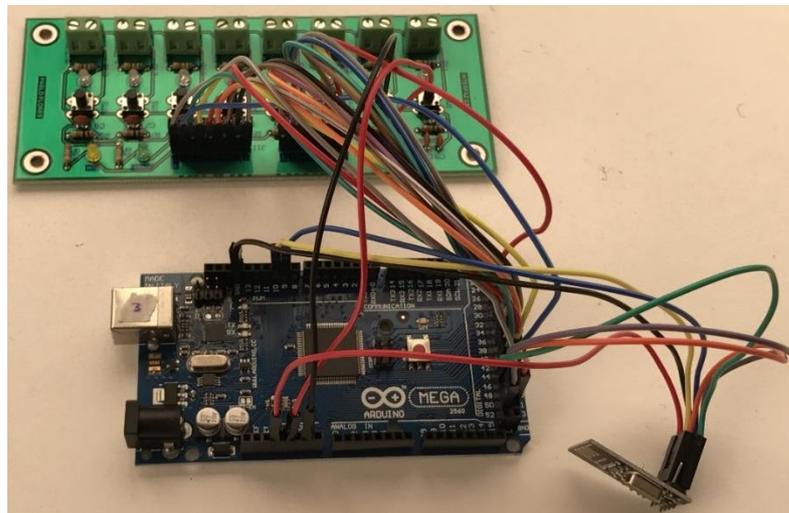


Figura 3-29: Esclavo CSE (Contactos Secos de Entrada)

Realizado por: Flores, Pablo. 2017

El esclavo CSE es el encargado de manejar todas las entradas de la red, cada esclavo contiene ocho entradas a través de una tarjeta de contactos secos (entradas de cierre de contacto), integrando de esta forma dispositivos externos a la red.

El elemento principal del esclavo CSE es el microcontrolador “arduino Mega” ya que administra todas las acciones realizadas en el esclavo, una de sus principales funciones es la de procesar las acciones detectadas en las entradas, almacenarlas de forma ordenada y enviarlas al maestro a través de radio frecuencia. La velocidad de comunicación se realiza a 2Mbps, pudiendo ser esta velocidad mayor o menor acorde sean los requerimientos de diseño o de hardware.

Un elemento adicional del esclavo CSE son los LEDs implementados y asociados a cada una de las entradas, los cuales cumplen el rol de indicadores.

Cabe mencionar que el envío/recepción de tramas de datos desde (o hacia) el dispositivo esclavo CSE se establece a través de una comunicación SPI (Interfaz Serial Periférica) entre el microcontrolador ATMEGA2560 y el módulo RF nRF24L01.

La estructura interna del esclavo CSE se muestra en el siguiente diagrama de bloques:

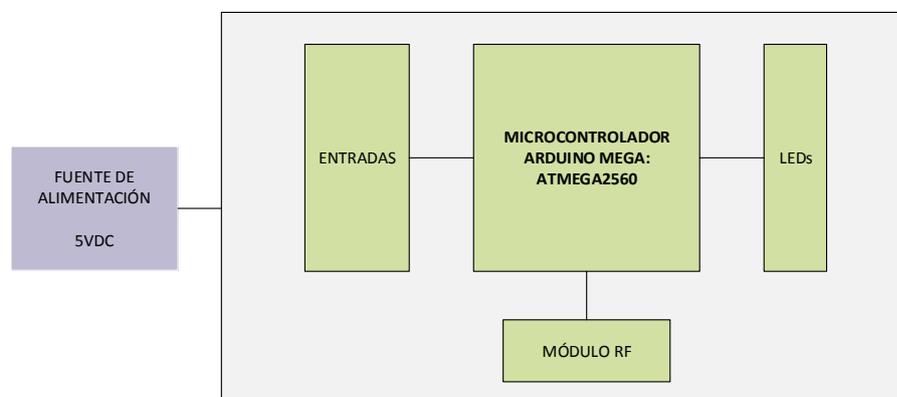


Figura 3-30: Diagrama de Bloques del Esclavo CSE

Realizado por: Flores, Pablo. 2017

Como se puede observar en el diagrama de bloques de la Figura 3-30, las partes del esclavo CSE son:

- Fuente de 5 V. de Corriente Continua
- Microcontrolador Arduino Mega: ATMEGA2560
- Módulo RF: transreceptor nRF24L01 2.4GHz

- Tarjeta de Contactos Secos de
- Entrada (8 entradas de cierre de contacto)
- LEDs Indicadores, 8

Las especificaciones técnicas generales de las partes del esclavo CSE se exponen en la siguiente tabla:

Tabla 3-2: Especificaciones Técnicas Generales del Esclavo CSE

Parte	Parámetro	Valor / Descripción
Fuente de Alimentación	Voltaje de salida	5 VDC
	Corriente de salida	2 A.
Microcontrolador Arduino MEGA	Numeración	ATMEGA 2560
	Voltaje de entrada	5 VDC
	Flash	256 KB
	RAM	8 KB
Módulo RF	Transreceptor	nRF24L01
	Voltaje de entrada	3.3 VDC
	Banda de Frecuencia	2.4 GHz
Tarjeta de Contactos Secos de Entrada	# Entradas	8
	Tipo de entradas	Cierre de Contacto
LEDs	Indicador	8 LEDs, uno junto a cada entrada

Realizado por: Flores, Pablo. 2017

3.4.3.3 Esclavo CSS

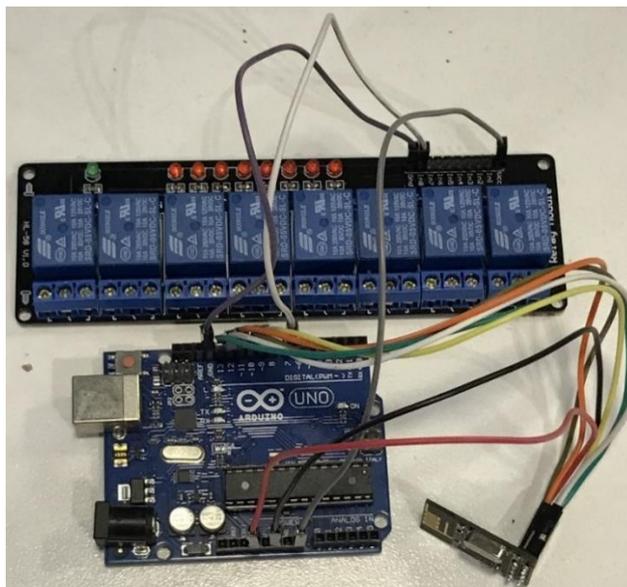


Figura 3-31: Esclavo CSS (Contactos Secos de Salida)

Realizado por: Flores, Pablo. 2017

El esclavo CSS es el encargado de manejar todas las salidas de la red, cada esclavo contiene ocho salidas a través de una tarjeta de contactos secos (salidas de cierre de contacto), lo cual permite integrar a la red cargas o dispositivos externos con un control a través de relés, por ejemplo: puerta eléctrica, jacuzzi, calefacción, etc. El esclavo CSS mediante la tarjeta de salidas permite además controlar todas las cargas luminosas de la aplicación domótica, mediante control on/off.

El elemento principal del esclavo CSS es el microcontrolador “arduino Uno” ya que administra de igual forma que en el esclavo CSE, todas las acciones realizadas en el mismo, siendo su principal función, la de procesar las solicitudes enviadas por el maestro a través de la elaboración de una tabla de registro de salidas, un análisis del “delay” y una comparación entre el estado actual y el estado futuro de cada una de las salidas.

La comunicación con el maestro se realiza mediante radio frecuencia, a una velocidad de 2Mbps, la cual puede ser mayor o menor acorde sean los requerimientos de diseño o de hardware.

Cada una de las salidas en el esclavo CSE están acompañadas por un LED, los cuales se han implementado para cumplir el rol de indicadores de la ejecución de una acción, es decir que, el LED se enciende cuando el relé de la tarjeta de salida se activa, y cuando este se desactiva, el LED se apaga inmediatamente.

Cabe mencionar que el envío/recepción de tramas de datos desde (o hacia) el dispositivo esclavo CSS se establece a través de una comunicación SPI (Interfaz Serial Periférica) entre el microcontrolador ATMEGA328P y el módulo RF nRF24L01.

La estructura interna del esclavo CSS se muestra en el siguiente diagrama de bloques:

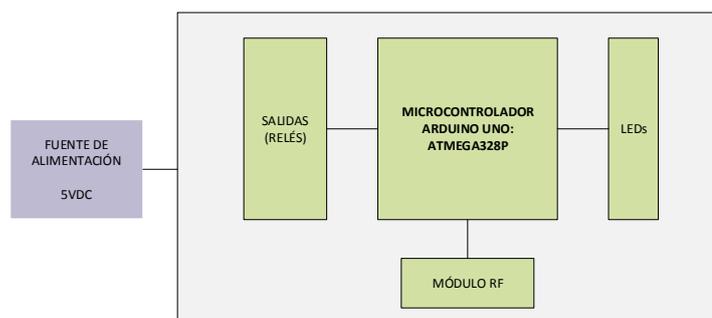


Figura 3-32: Diagrama de Bloques del Esclavo CSS

Realizado por: Flores, Pablo. 2017

Como se puede observar en el diagrama de bloques de la Figura 3-32, las partes del esclavo CSS son:

- Fuente de 5 V. de Corriente Continua
- Microcontrolador Arduino Uno: ATMEGA328P
- Módulo RF: transreceptor nRF24L01 2.4GHz
- Tarjeta de Contactos Secos de Salida (8 salidas de cierre de contacto, Relés)
- LEDs Indicadores, 8

Las especificaciones técnicas generales de las partes del esclavo CSE se exponen en la siguiente tabla:

Tabla 3-3: Especificaciones Técnicas Generales de las Partes del Esclavo CSS

Parte	Parámetro	Valor / Descripción
Fuente de Alimentación	Voltaje de salida	5 VDC
	Corriente de salida	2 A.
Microcontrolador Arduino UNO	Numeración	ATMEGA 328P
	Voltaje de entrada	5 VDC
	Flash	32 KB
	RAM	2 KB
Módulo RF	Transreceptor	nRF24L01
	Voltaje de entrada	3.3 VDC
	Banda de Frecuencia	2.4 GHz
Tarjeta de Contactos Secos de Salida	# Salidas	8
	Tipo de salidas	Cierre de Contacto
LEDs	Indicador	8 LEDs, uno junto a cada salida

Realizado por: Flores, Pablo. 2017

3.4.3.4 Teclado Celular

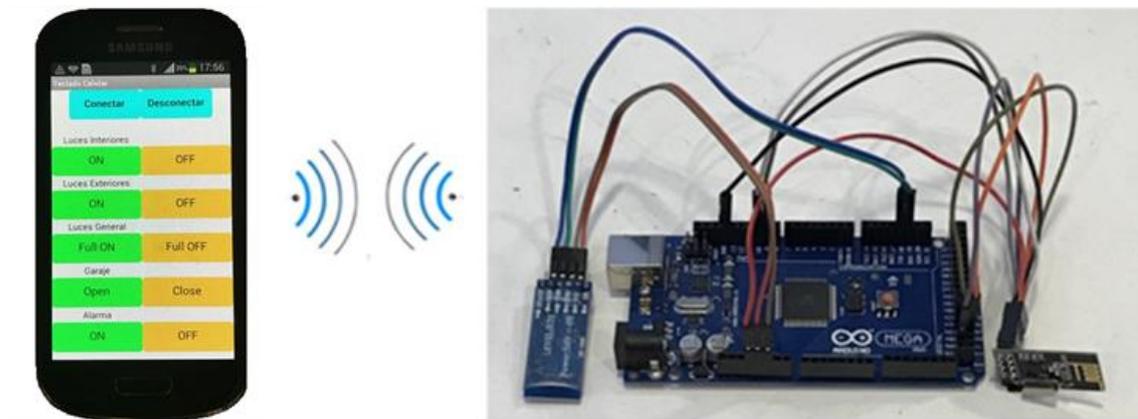


Figura 3-33: Teclado Celular - Maestro

Realizado por: Flores, Pablo. 2017

El maestro tiene un teclado celular que, para efectos de la aplicación, es considerado como un teclado local; es decir, en la comunicación bluetooth que se establece entre dicho teclado y el maestro no se utiliza el protocolo propietario desarrollado, únicamente se envían comandos en un solo sentido desde el teclado hacia el maestro.

Por lo tanto el maestro continuamente monitorea si se ha presionado algún botón del teclado celular, una vez que se detecte alguna acción, el maestro procesa la misma, verificando si se trata de alguna acción programada para agregar el correspondiente Preset de la acción a la Pila de Escritura de Presets, finalmente se envía la solicitud a los esclavos de salida.

La función principal del teclado celular consiste en mejorar y ampliar la aplicación domótica desarrollada, en la cual se ha implementado el bus propietario elaborado, teniendo de esta forma un medio de control adicional de las cargas residencial ON/OFF

El teclado celular está conformado por 10 botones que comandan diferentes cargas de la aplicación domótica, las mismas que se detallan en la siguiente tabla:

Tabla 3-4: Botones Teclado Celular – Cargas Controladas

# de Entrada	Descripción	# Esclavo CSS	Acción
Botón 1	Encender Luces Interiores	5, 6, 7	0
Botón 2	Apagar Luces Interiores	5, 6, 7	1
Botón 3	Encender Luces Exteriores	5	1
Botón 4	Apagar Luces Exteriores	5	0
Botón 5	Encender Luces Generales	5, 6, 7	0
Botón 6	Apagar Luces Generales	5, 6, 7	1
Botón 7	Abrir Garaje	7	1
Botón 8	Cerrar Garaje	7	0
Botón 9	Activar Alarma	7	1
Botón 10	Desactivar Alarma	7	0

Realizado por: Flores, Pablo. 2017

3.4.4 Aplicación Domótica:



Figura 3-34: Aplicación Domótica

Realizado por: Flores, Pablo. 2017

La aplicación domótica construida comanda cargas residenciales con control ON/OFF a través de una tabla fija programada, implementando el bus propietario desarrollado, tiene por entradas a interruptores, sensores de movimientos y contactos magnéticos; mientras que las salidas son cargas luminosas, zumbadores, un servomotor y un ventilador.

La función principal de la aplicación domótica es la de validar la fiabilidad del bus propietario desarrollado, comprobando la estructura de la red y el protocolo de comunicación utilizado entre el maestro y sus esclavos, a través de la ejecución inmediata de todas las solicitudes, las cuales son generadas en el esclavo CSE, procesadas en el maestro y realizadas por los esclavos CSS.

Las acciones que se pueden observar en la aplicación domótica son las siguientes:

- Todas las luces de la vivienda, exteriores y espacios verdes pueden controlarse individualmente.
- Existen senderos de luz, se enciende un conjunto de luces para guiar el trayecto a un lugar determinado.
- Mediante una sola señal se pueden encender o apagar todas las luces de la vivienda.
- Existen ambientes, es decir mediante una sola señal se encienden luces de la vivienda de áreas cercanas
- Al activarse un sensor de movimiento, se encienden las luces que se encuentran alrededor del mismo.
- Al activarse un sensor magnético ubicado en las puertas de ingreso de la casa, se activa la alarma de aviso y la luz del pasillo.
- Las cargas instaladas en la vivienda representan a cualquier carga de control ON/OFF, como: audio, video, calefacción, chimenea, jacuzzi, acceso, cortinas, riego, piletas, etc.
- La puerta del garaje se abre automáticamente y se detiene gracias a un final de carrera, funciona de igual forma para el cierre.
- A través del teclado celular se comandan algunas cargas de la aplicación.

Las figuras 3-35, 3-36 y 3-37 hacen referencia a los planos de la aplicación domótica, resaltando la ubicación de todas las cargas y sensores:

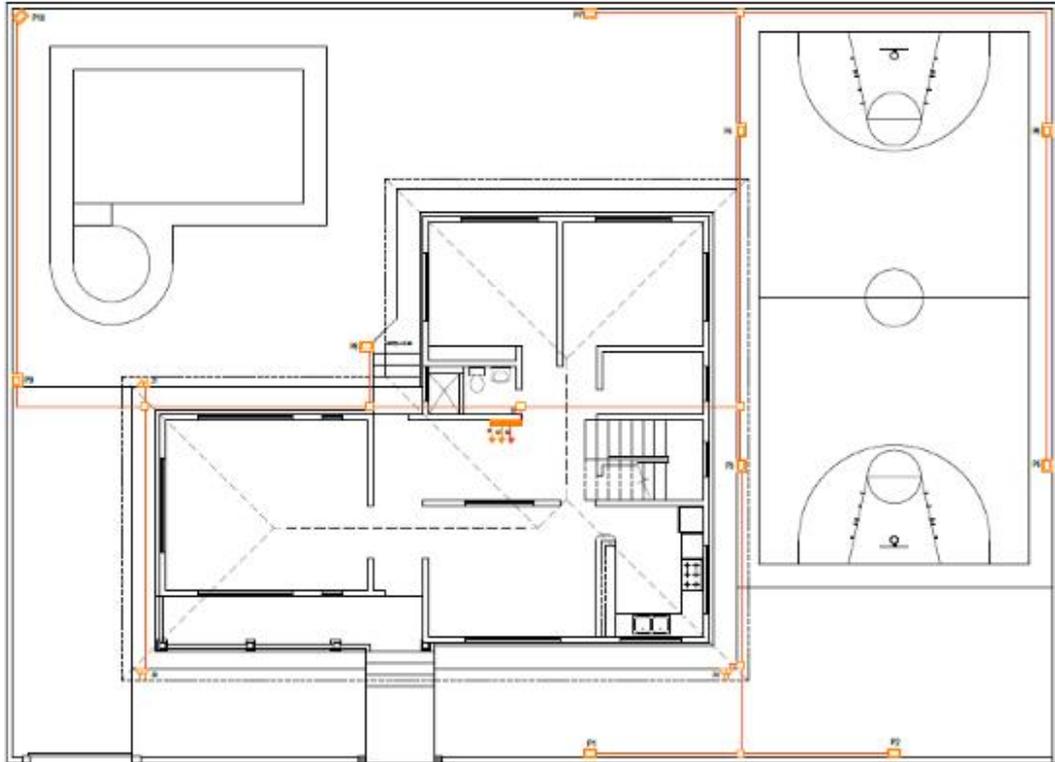


Figura 3-35: Instalaciones de Luminarias del Área Exterior-Aplicación Domótica

Realizado por: Flores, Pablo. 2017

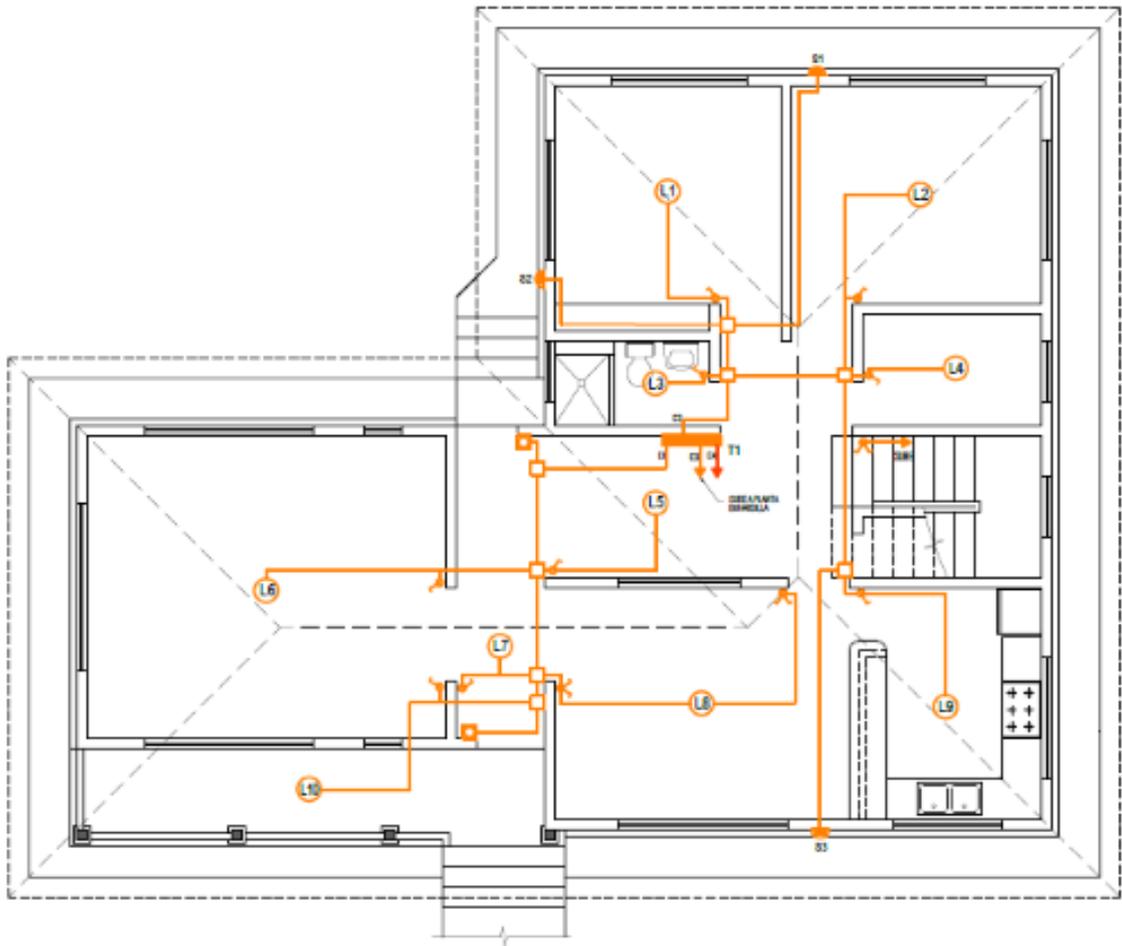


Figura 3-36: Instalaciones Eléctricas Planta Baja-Aplicación Domótica

Realizado por: Flores, Pablo. 2017

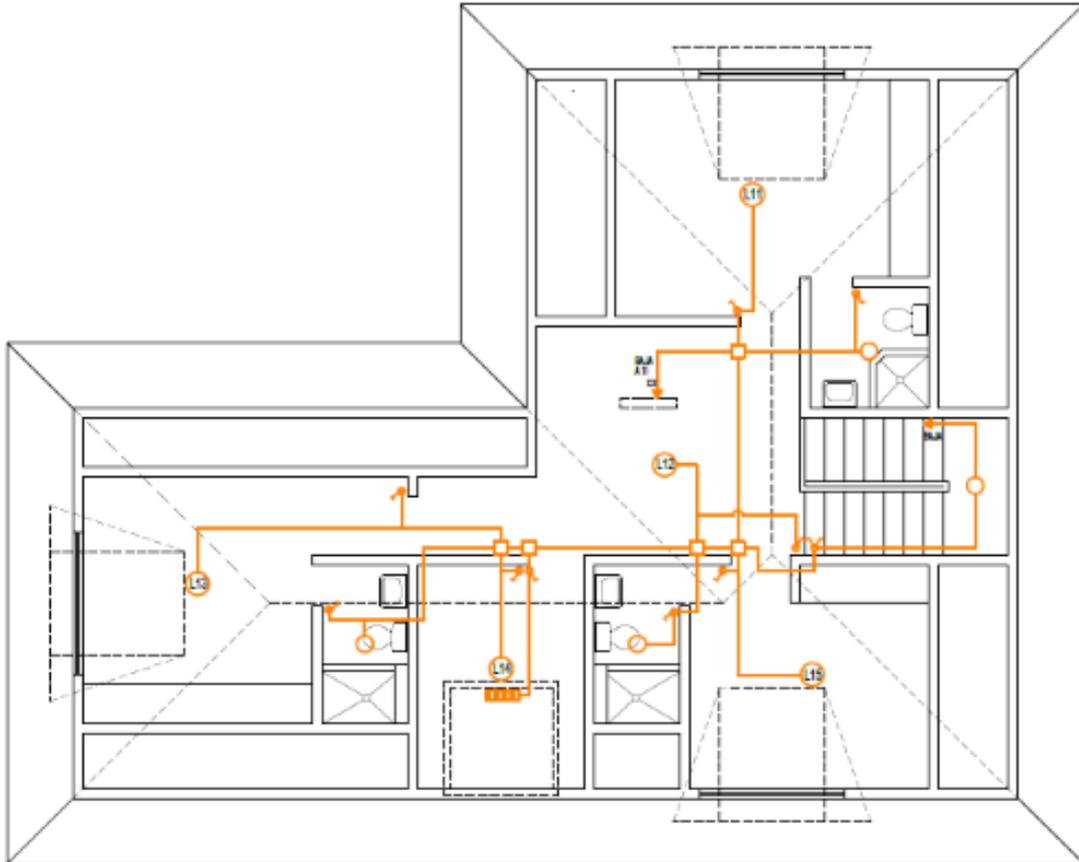


Figura 3-37: Instalaciones Eléctricas Buhardilla-Aplicación Domótica

Realizado por: Flores, Pablo. 2017

La tabla 3-5 indica la simbología utilizada en los planos para las luminarias, postes, sensores y cargas de la aplicación domótica. Mientras que la tabla 3-6 es un resumen del comando de las cargas de la aplicación.

Tabla 3-5: Simbología de los Accesorios de la Aplicación Domótica

CUADRO DE LUM INARIAS			
PLANTA BAJA		PLANTA BUHARDILLA	
L1	ESTUDIO	L11	DORMITORIO 1
L2	GIMNASIO	L12	BESTIBULO
L3	BAÑO	L13	DORMITORIO 2
L4	LAVANDERIA	L14	DORMITORIO VISITAS
L5	VESTIBULO INTERIOR	L15	DORMITORIO 3
L6	SALA	P1,... P10	ILUMINACION EXTERIOR
L7	VESTIBULO ACCESO	Z1, Z2, Z3	ZUMBADORES EXTERIORES
L8	COMEDOR		
L9	COCINA		
L10	PORTAL		

Realizado por: Flores, Pablo. 2017

Tabla 3-6: Resumen del Comando de las Cargas de la Aplicación Domótica

# Esclavo CSE	# Entrada	Activadores	# Esclavo CSS	# Relé	Cargas	Área
CSE 1	1, 2	S. Movimiento 1, Interruptor 1	5	1	L10, P1, P2	Frente de la casa
	3	Interruptor 2	5	2	P3, P4, P5, P6	Cancha
	4,5,6	S. Movimiento 2 y 3, Interruptor 3	5	3	P7, P8, P9	Posterior de la casa
	7	Interruptor 4	5	4	P10	Poste Piscina
	7	Interruptor 5	5	5	L16, L17	Iluminación Piscina
	8	Interruptor 6	5	6	L1	Estudio
CSE 2	1	Interruptor 7	5	7	L2	Gimnasio
	2	Interruptor 8	5	8	L3	Baño planta baja
	3	Interruptor 9	6	1	L4	Lavandería
	4	Interruptor 10	6	2	L5	Sala
	5	S. Magnético 1	6	3	L6	Vestíbulo
	5	S. Magnético 1	7	4	Z1, Z2, Z3	Alarma Casa
	6	S. Magnético 2	6	4	L7	Recibidor
	6	S. Magnético 2	7	4	Z1, Z2, Z3	Alarma Casa
	7	Interruptor 11	6	5	L8, L9	Cocina - Comedor
	8	Interruptor 12	6	6	L11	Dormitorio 1
CSE 3	1	Interruptor 13	6	7	L12	Estar Buhardilla
	2	Interruptor 14	6	8	L13	Dormitorio 2
	3	Interruptor 15	7	1	L14	Dormitorio visitas
	4	Interruptor 16	7	2	L15	Dormitorio 3
	5	Interruptor 17	7	3	V1	Ventilador
	8	Interruptor 20	5, 6, 7		LUCES	Luces General

Realizado por: Flores, Pablo. 2017

CAPITULO IV

4 ANÁLISIS DE RESULTADOS Y DISCUSIÓN

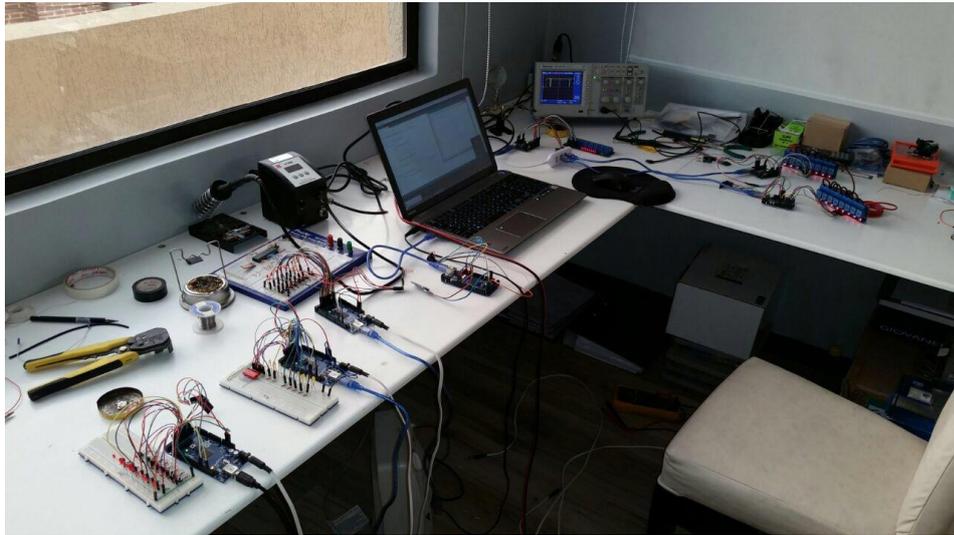


Figura 4-1: Laboratorio de Pruebas

Realizado por: Flores, Pablo. 2017

El presente capítulo comprende las pruebas de funcionamiento del bus propietario, así como el análisis de los resultados obtenidos y la discusión de los mismos.

Las pruebas de funcionamiento se realizaron sobre todos y cada uno de los elementos que intervienen en la red, la cual está conformada por un elemento central, que es el maestro, y seis elementos esclavos divididos en 3 esclavos CSE y 3 esclavos CSS; la red comanda cargas residenciales de control ON/OFF en la aplicación domótica elaborada, a través de interruptores, sensores y el teclado celular.

4.1 Pruebas de Comunicación:

Para las pruebas de comunicación entre el maestro y los esclavos se utilizó los siguientes instrumentos:

- Osciloscopio two-channels 100MHz TEKTRONIX TDS 2012B
- Puntas de prueba pasivas 1X/10X 200MHz TEKTRONIX P2220

4.1.1 Pruebas: Función Leer

Se han tomado capturas del osciloscopio en las que se muestra de forma simultánea la solicitud enviada por el maestro y la respuesta del esclavo en relación a la función leer, evidenciando el número de Bytes enviados y el tiempo total de la transacción.

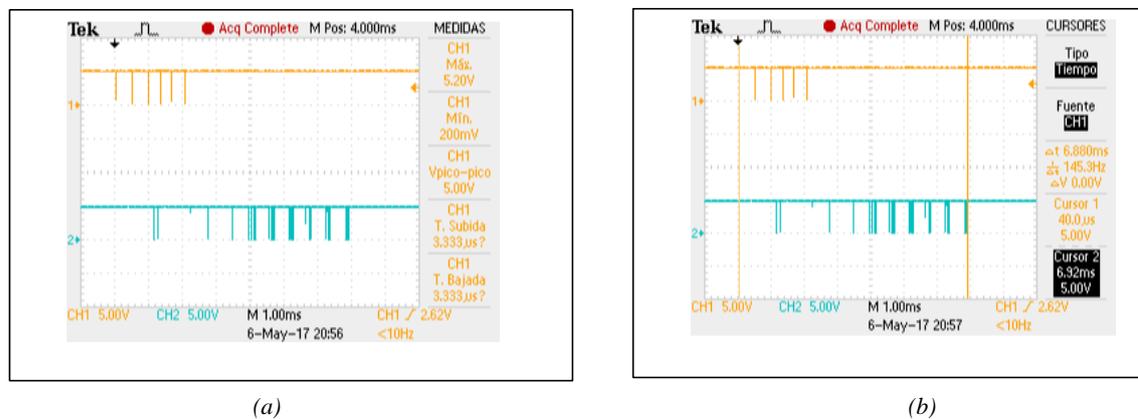


Figura 4-2: Tramas de datos para la función Leer: (a) Bytes enviados en la solicitud del maestro y en la respuesta del esclavo; (b) Tiempo total de la transacción

Realizado por: Flores, Pablo. 2017

En la figura 4-4 (a), se puede observar que la trama enviada por el maestro para una solicitud de lectura es de 6 Bytes, mientras que la respuesta del esclavo -cuando no tiene acciones almacenadas en la pila de lectura- es de igual forma de 6 Bytes. Los Bytes previos a la respuesta del esclavo no son considerados como parte de la trama de datos de la transacción ya que corresponden a la secuencia de inicio del modo recepción (para mayor información refiérase al Anexo B).

En la figura 4-4 (b), se indica el tiempo total de la transacción, el cual es de 6,88 ms y hace referencia al tiempo de envío de la solicitud realizada por el maestro, y el tiempo de su procesamiento y respuesta realizado por el esclavo.

4.1.2 Pruebas: Función Escribir

De igual forma se han tomado capturas del osciloscopio en las que se muestra de forma simultánea la solicitud enviada por el maestro y la respuesta del esclavo en relación a la función leer, evidenciando el número de Bytes enviados y el tiempo total de la transacción.

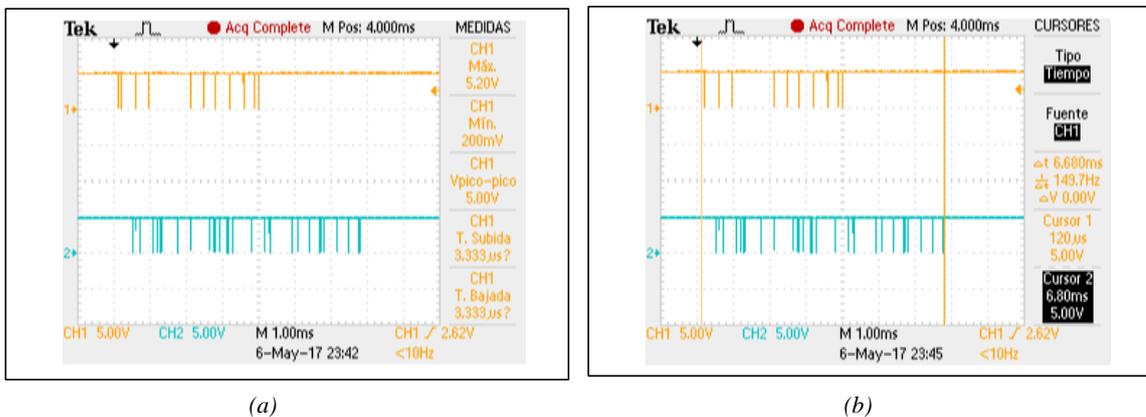


Figura 4-3: Tramas de datos para la función Escribir: (a) Bytes enviados en la solicitud del maestro y en la respuesta del esclavo; (b) Tiempo total de la transacción

Realizado por: Flores, Pablo. 2017

En la figura 4-5 (a), se puede observar que la trama enviada por el maestro para una solicitud de escritura es de 11 Bytes, mientras que la respuesta del esclavo es de 6 Bytes. Los Bytes previos a la respuesta del esclavo no son considerados como parte de la trama de datos de la transacción ya que corresponden a la secuencia de inicio del modo recepción (para mayor información refiérase al Anexo B).

En la figura 4-5 (b), se indica el tiempo total de transacción de 6,68 ms, y hace referencia al tiempo de envío de la solicitud realizada por el maestro, y el tiempo de su procesamiento y respuesta realizado por el esclavo.

4.1.4 Análisis de Resultados:

- El bus propietario funciona de forma correcta, según las especificaciones de diseño predefinidas.
- La comunicación entre el maestro y sus esclavos se realiza sin inconvenientes, utilizándose para cada transacción de datos el protocolo propietario desarrollado.
- La comunicación entre los tres dispositivos de entrada, el maestro y los tres dispositivos de salida fue exitosa, ya que las instrucciones fueron ejecutadas de manera inmediata (tiempo real según percepción humana).
- El LED “TX” del maestro, se enciende mientras existe transferencia de datos entre el maestro y sus esclavos.
- El LED “COM” se enciende cuando existe un fallo en la comunicación.
- Los datos enviados en la comunicación son enviados y recibidos en su totalidad.
- El maestro se encuentra en constante monitoreo del estado de todas las entradas de los esclavos CSE.
- Cuando una entrada de los esclavos CSE es cerrada o abierta, la acción se almacena en una pila de lectura y espera la solicitud de lectura del maestro para mandar la información, es decir el esclavo CSE no puede iniciar la comunicación.
- El maestro una vez que recibe la información de las acciones detectadas, revisa su programación interna para procesar los datos y comparar los mismos con las tablas de los registros Presets, posteriormente envía a ejecutar dichas acciones a los esclavos CSS.
- Cuando los esclavos CSS reciben una solicitud del maestro para modificar el estado de una de sus salidas, ésta se lleva a cabo conforme a los parámetros del Preset procesado.
- Una vez ejecutada satisfactoriamente la acción enviada por el maestro, el esclavo CSS responde con una trama de datos de 6 Bytes como se evidencia en la figura 4-5.
- Posterior a la notificación de la ejecución de las acciones, el maestro nuevamente se comunica con el esclavo CSE, para indicarle que borre las acciones detectadas que se encuentra almacenadas en la pila.
- El proceso de comunicación finaliza con la respuesta del esclavo CSE, quien indica que la las acciones almacenadas en su pila ya han sido eliminadas.
- En el caso de que las acciones detectadas no fueran ejecutadas por el esclavo CSS, éstas se mantienen almacenadas en la pila de lectura del esclavo CSE hasta que se restablezca la comunicación y se ejecuten las mismas, garantizando el cumplimiento de todas las acciones.

- Cuando por algún motivo no responden los esclavos, se pierde la comunicación, o existe información incorrecta, el maestro ha sido programado para que reintente la comunicación con los esclavos hasta por 5 ocasiones.
- El algoritmo de comprobación de redundancia cíclica incluido en la trama de cada mensaje, valida correctamente la información de los mensajes recibidos por cualquier dispositivo.
- Los LEDs ubicados junto a cada salida de los esclavos CSS, se encienden cuando el relé correspondiente a dicha salida se activa.
- Las cargas de control ON/OFF instaladas en la aplicación domótica responden de forma inmediata según la entrada activada.
- Los interruptores, sensores de movimiento y sensores magnéticos activan los relés predefinidos en las tablas de programación fija.
- Todos los botones del teclado celular funcionan sin inconvenientes, activando los relés y cargas correspondientes según su tabla de programación fija.

4.2 Análisis de las Tramas de Datos y Velocidad de Comunicaciones:

Las pruebas realizadas en la red estructurada han podido determinar y confirmar el número de bytes, campos y valores que son enviados en cada trama de datos de cada función del protocolo propietario, es por ello que en éste apartado se exponen ejemplos de tramas de datos enviadas tanto por el maestro y los esclavos para cumplir instrucciones específicas, dependiendo de la función ejecutada.

De igual forma se procede a calcular el tiempo teórico necesario para transmitir cada trama de datos considerando una velocidad de comunicación en el medio de 2Mbps, que es la velocidad de comunicación programada en los transreceptores nRF24L01 2.4GHz. En todos los casos el tiempo de Byte será el siguiente:

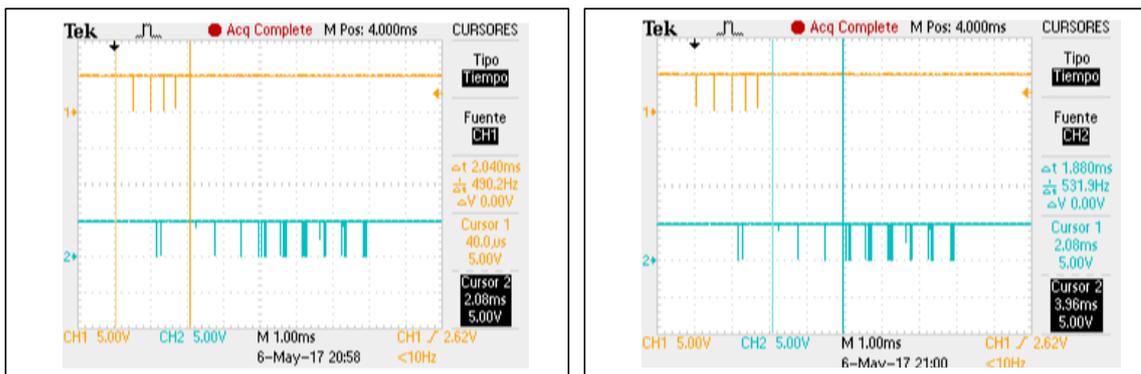
$$\textit{tiempo de Byte} = \frac{1s}{2000000bits} \times 8bits = 0,000004 s = 4\mu s$$

Cabe resaltar que los tiempos reales en los que se envía una trama o se ejecuta una transacción, varían en relación a los tiempos teóricos, debido al tiempo de procesamiento de los microcontrolador Arduinos utilizados (UNO y MEGA).

Este tiempo de procesamiento se refiere al tiempo que tarda el microcontrolador en ejecutar las líneas de código del programa alojado en su memoria flash; en este sentido para enviar una solicitud, los datos son cargados Byte por Byte en el buffer de transmisión del puerto serial del microcontrolador, y una vez que los datos se encuentran en este buffer la velocidad a la que se transmiten los datos es de 2 Mbps; sin embargo antes de ser cargados dichos Bytes en el buffer referido, el microcontrolador debe ejecutar un proceso interno, el cual mediante pruebas se ha podido determinar que es mucho mayor al tiempo de envío de los datos.

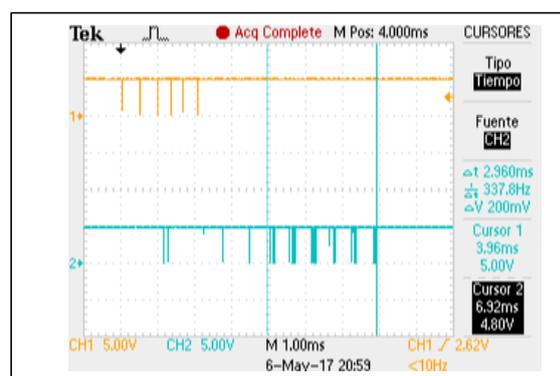
4.2.1 Función Leer

4.2.1.1 Tiempo Real de Solicitud, Procesamiento y Respuesta



(a)

(b)



(c)

Figura 4-5: Tiempo Real Función Leer: (a) Solicitud del maestro; (b) Procesamiento; (c) Respuesta del Esclavo

Realizado por: Flores, Pablo. 2017

La figura 4-5 (a) indica que el tiempo de solicitud es de 2,040 ms; la figura 4-5 (b) visualiza un tiempo de 1,880 ms del procesamiento y la figura 4-5 (c) evidencia un tiempo de respuesta de 2,960 ms.

4.2.1.2 Tiempos Teóricos de Solicitud y Respuesta

- a) El maestro solicita leer las acciones del esclavo CSE:

1 100 0 0 64 7

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-1: Bytes en la Trama de Solicitud de Lectura del Maestro

Campo	Valor	# Bytes
Dirección	1	1
Función (leer)	100	1
Dirección de inicio	0	1
Número de Bytes a leer	0 (lee todo)	1
CRCL	64	1
CRCH	7	1
Total de Bytes		6

Realizado por: Flores, Pablo. 2017

$$\text{tiempo de trama} = 4us \times 6 \text{ bytes} = 24us = 0,024ms$$

- a) El esclavo CSE responde no tener acciones detectadas:

1 100 0 0 64 7

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-2: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Lectura Cuando no hay Acciones Detectadas

Campo	Valor	# Bytes
Dirección	1	1
Función (leer)	100	1
Dirección de inicio	0	1
Número de Bytes	0 (no hay nada)	1
CRCL	64	1
CRCH	7	1
Total de Bytes		6

Realizado por: Flores, Pablo. 2017

$$tiempo\ de\ trama = 4us \times 6\ bytes = 24us = 0,024ms$$

- b) El esclavo CSE responde tener una acción detectada para activar un relé:

1 100 0 2 1 1 17 146

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-3: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Lectura Cuando hay Acciones Detectadas (activar)

Campo	Valor	# Bytes
Dirección	1	1
Función (leer)	100	1
Dirección de inicio	0	1
Número de Bytes	2	1
Número de entrada activada	1	1
Acción	1 (cerrado)	1
CRCL	17	1
CRCH	146	1
Total de Bytes		8

Realizado por: Flores, Pablo. 2017

$$tiempo\ de\ trama = 4us \times 8\ bytes = 32us = 0,032ms$$

- c) El esclavo CSE responde tener una acción detectada para desactivar un relé:

1	100	0	2	1	2	81	147
---	-----	---	---	---	---	----	-----

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-4: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Lectura Cuando hay Acciones Detectadas (desactivar)

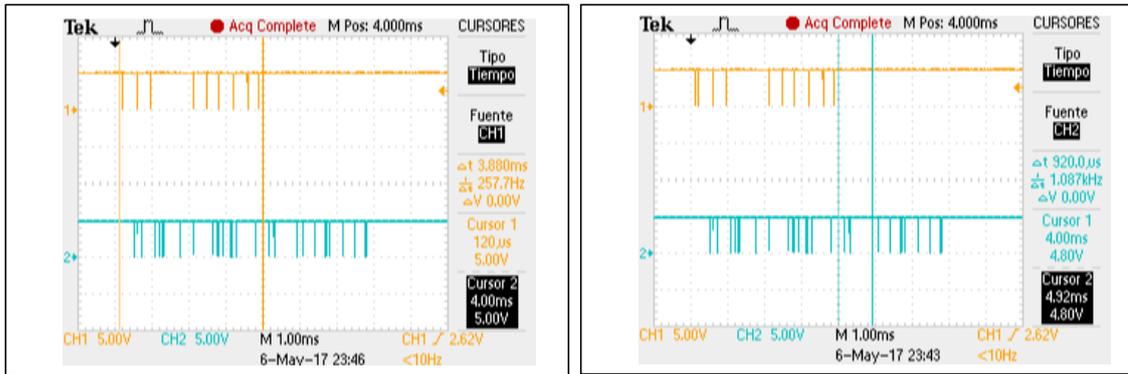
Campo	Valor	# Bytes
Dirección	1	1
Función (leer)	100	1
Dirección de inicio	0	1
Número de Bytes	2	1
Número de entrada activada	1	1
Acción	2 (abierto)	1
CRCL	81	1
CRCH	147	1
Total de Bytes		8

Realizado por: Flores, Pablo. 2017

$$tiempo\ de\ trama = 4us \times 8\ bytes = 32us = 0,032ms$$

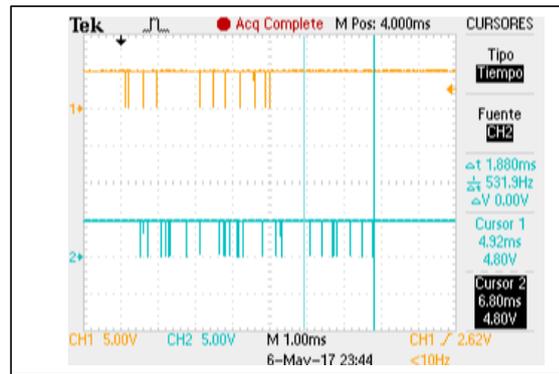
4.2.2 Función Escribir

4.2.2.1 Tiempo Real de Solicitud, Procesamiento y Respuesta



(a)

(b)



(c)

Figura 4-6: Tiempo Real Función Escribir: (a) Solicitud del maestro; (b) Procesamiento; (c) Respuesta del Esclavo

Realizado por: Flores, Pablo. 2017

La figura 4-6 (a) indica que el tiempo de solicitud es de 3,880 ms; la figura 4-6 (b) visualiza un tiempo de 920 us del procesamiento y la figura 4-6 (c) evidencia un tiempo de respuesta de 1,880 ms.

4.2.2.2 Tiempos Teóricos de Solicitud y Respuesta

a) El maestro solicita escribir los Presets en el esclavo CSS:

```
5 101 0 5 0 0 0 0 0 50 34
```

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-5: Bytes en la Trama de Solicitud del Maestro Para Escribir

Campo	Valor	# Bytes
Dirección	5	1
Función (escribir)	101	1
Dirección de inicio	0 (depende del # salida)	1
Número de Bytes a escribir	5	1
Retardo_H	0	1
Retardo_L	0	1
Reservado_H	0	1
Reservado_L	0	1
Acción	0 (encender)	1
CRCL	50	1
CRCH	34	1
Total de Bytes		11

Realizado por: Flores, Pablo. 2017

$$tiempo\ de\ trama = 4us \times 11\ bytes = 44us = 0,044ms$$

- b) El esclavo CSS responde la acción ejecutada:

5	101	0	5	208	244
---	-----	---	---	-----	-----

Tabla 4-6: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Escribir

Campo	Valor	# Bytes
Dirección	5	1
Función (escribir)	101	1
Dirección de inicio	0	1
Número de Bytes a escribir	5	1
CRCL	208	1
CRCH	244	1
Total de Bytes		6

Realizado por: Flores, Pablo. 2017

$$\text{tiempo de trama} = 4\mu\text{s} \times 6 \text{ bytes} = 24\mu\text{s} = 0,024\text{ms}$$

4.2.3 Función Eliminar:

4.2.3.1 Tiempo Real de Solicitud, Procesamiento y Respuesta

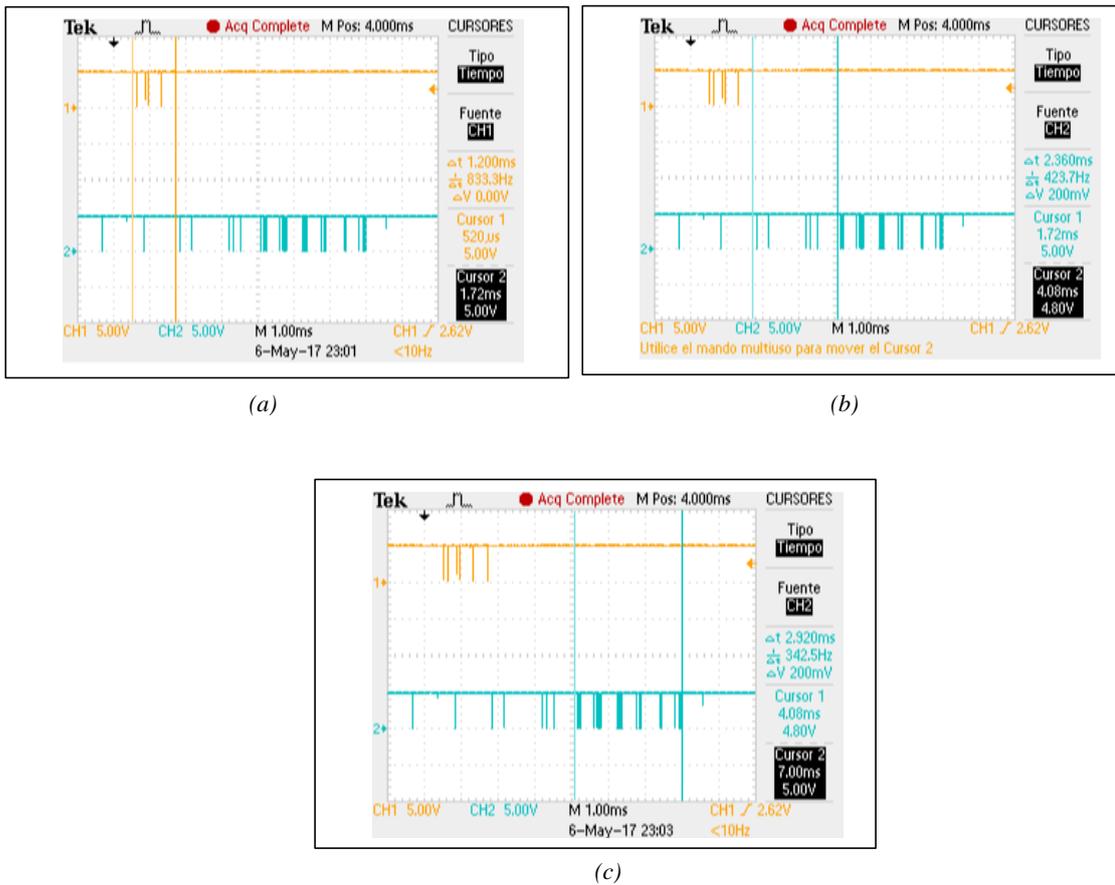


Figura 4-7: Tiempo Real Función Eliminar: (a) Solicitud del maestro; (b) Procesamiento; (c) Respuesta del Esclavo

Realizado por: Flores, Pablo. 2017

La figura 4-7 (a) indica que el tiempo de solicitud es de 1,200 ms; la figura 4-7 (b) visualiza un tiempo de 2360 ms del procesamiento y la figura 4-7 (c) evidencia un tiempo de respuesta de 2,920 ms.

4.2.3.2 *Tiempos Teóricos de Solicitud y Respuesta*

- a) El maestro solicita eliminar las acciones que fueron detectadas en el esclavo CSE y ya fueron ejecutadas por el esclavo CSS:

1 102 0 2 96 6

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-7: Bytes en la Trama de Solicitud del Maestro Para Eliminar

Campo	Valor	# Bytes
Dirección	1	1
Función (eliminar)	102	1
Dirección de inicio	0	1
Número de Bytes a eliminar	2	1
CRCL	96	1
CRCH	6	1
Total de Bytes		6

Realizado por: Flores, Pablo. 2017

$$tiempo\ de\ trama = 4us \times 6\ bytes = 24us = 0,024ms$$

- b) El esclavo CSE responde haber cumplido la solicitud de eliminar dichas acciones:

1 102 0 2 96 6

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-8: Bytes en la Trama de Respuesta del Esclavo a la Solicitud de Eliminar

Campo	Valor	# Bytes
Dirección	1	1
Función (eliminar)	102	1
Dirección de inicio	0	1
Número de Bytes a eliminar	2	1
CRCL	96	1
CRCH	6	1
Total de Bytes		6

Realizado por: Flores, Pablo. 2017

$$\mathbf{tiempo\ de\ trama = 4us \times 6\ bytes = 24us = 0,024ms}$$

4.2.4 Tiempo Real de Activación de Carga

En el bus propietario se considera tiempo real de activación de carga, a la suma de los tiempos correspondientes a los siguientes mensajes:

- Maestro: Solicitud de lectura del estado las entradas
- Esclavo CSE: Respuesta con la información de las acciones detectadas
- Maestro: Solicitud de escribir los presets de las acciones detectadas
- Esclavo CSS: Respuesta indicando que ya fueron ejecutadas las acciones solicitadas

Adicional al tiempo necesario para transmitir éstas tramas de mensajes, se considera los tiempos de procesamiento de los microcontroladores utilizados.

La tabla 4-9 recoge información de los tiempos obtenidos a través de mediciones para la activación de cargas:

Tabla 4-9: Mediciones Tiempo de Activación de Carga

# de Medición	Tiempo Real de Activación de Carga
1	13,56 ms
2	13,55 ms
3	13,56 ms
4	13,55 ms
5	13,57 ms
6	13,56 ms
7	13,56 ms
8	13,57 ms
9	13,55 ms
10	13,56 ms

Realizado por: Flores, Pablo. 2017

Una vez que se han obtenido los datos correspondientes a diez mediciones realizadas mediante el osciloscopio, se ha podido determinar que el tiempo promedio para la activación de una carga, es de 13,559 ms. En cuanto el tiempo teórico es de 0,124ms.

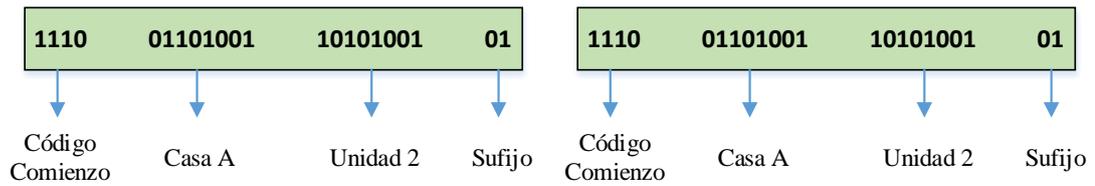
4.3 Velocidad de Comunicaciones en Protocolos Domóticos Comerciales:

En el presente apartado se calcula el tiempo de transmisión de algunas tramas de datos utilizadas por protocolos domóticos comerciales, para enviar ya sea una solicitud o una respuesta con relación a instrucciones específicas. Para el cálculo se especifica de igual forma que en el protocolo propietario una velocidad de comunicación en el medio de 2Mbps.

4.3.1 Protocolo X-10

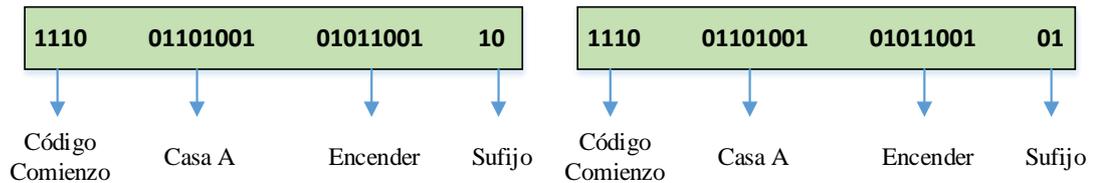
Según (Chan, y otros, 2011) para encender un módulo X-10 configurado con el código A, unidad 2, el transmisor deberá enviar las siguientes cadenas de datos por la línea de red:

a) Se envía la dirección dos veces:



b) Se espera tres ciclos de red (seis pasos por cero)

c) Se envía el comando dos veces



d) Se espera tres ciclos de red antes de enviar el próximo bloque

Para poder transmitir estas cadenas de datos se necesitan 47 ciclos de la señal de la red. A una frecuencia de 60 Hz, se puede determinar el tiempo de transmisión con los siguientes cálculos:

$$t = \frac{1}{f} = \frac{1}{60\text{Hz}} = 0,0166667\text{s} = 16,67\text{ms}$$

$$\text{Tiempo para activar un relé} = 16,67\text{ms} \times 47 \text{ ciclos} = 783,49\text{ms}$$

4.3.2 Protocolo ZigBee

Según (Rodríguez, 2014) para activar un actuador conectado a un módulo Xbee remoto, se envían las siguientes tramas de datos en el orden indicado:

a) El módulo XBee coordinador envía un paquete de datos al módulo XBee remoto para que cambie el estado de la salida digital que corresponda:

7E 00 10 17 01 00 00 00 00 00 00 00 00 00 01 02 44 35 05 66

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-10: Bytes en la Trama de Solicitud de Configuración de Salidas Digitales del XBee Remoto

Campo	Valor	# Bytes
Comienzo de trama	0 x 7E	1
Longitud del Frame Data	0 x 0010	2
Identificador de comando remoto	0 x 17	1
Identificador de trama	0 x 01	1
Dirección de 64 bits	0 x 00	8
Dirección de 16 bits Xbee remoto	0 x 0001	2
Opción de ejecución de comando	0 x 02	1
Comando ATD5	0 x 4435	2
Parámetro a fijarse el comando	0 x 05	1
Checksum de la trama API	0 x 66	1
Total de Bytes		20

Realizado por: Flores, Pablo. 2017

$$tiempo\ de\ Byte = \frac{1s}{2000000bits} \times 8bits = 0,000004\ s = 4us$$

$$tiempo\ de\ trama = 4us \times 20\ bytes = 80us = 0,08ms$$

- b) Si todo es correcto el módulo XBee remoto responde al módulo XBee coordinador una trama de Respuesta de Comando Remoto:

7E 00 0F 97 01 00 13 A2 00 40 AA 5E A0 00 01 44 35 00 50

El significado de los Bytes se indica en la siguiente tabla:

Tabla 4-11: Bytes Trama de Respuesta de Comando Remoto XBee

Campo	Valor	# Bytes
Comienzo de trama	0 x 7E	1
Longitud del Frame Data	0 x 000F	2
Identificador de respuesta de comando remoto	0 x 97	1
Identificador de trama	0 x 01	1
Dirección de 64 bits (hardware)	0 x 0013A20040 AA5EA0	8
Dirección de 16 bits Xbee remoto	0 x 0001	2
Comando ATD5	0 x 4435	2
Estado de ejecución correcta del comando	0 x 00	1
Checksum de la trama API	0 x 50	1
Total de Bytes		19

Realizado por: Flores, Pablo. 2017

$$\text{tiempo de trama} = 4\mu s \times 19 \text{ bytes} = 76\mu s = 0,076\text{ms}$$

- c) Por último el módulo XBee coordinador envía una trama al módulo XBee remoto para indicar que guarde en su memoria la nueva configuración:

7E 00 0F 17 01 00 00 00 00 00 00 00 00 00 01 00 57 52 3D

Tabla 4-12: Bytes Trama de Configuración XBee Remoto

Campo	Valor	# Bytes
Comienzo de trama	0 x 7E	1
Longitud del Frame Data	0 x 000F	2
Identificador de comando remoto	0 x 17	1
Identificador de trama	0 x 01	1
Dirección de 64 bits (hardware)	0 x 000000000000 00000	8
Dirección de 16 bits Xbee remoto	0 x 0001	2
Sin opción de comando especial	0 x 00	1
Comando ATWR	0 x 5752	2
Checksum de la trama API	0 x 3D	1
Total de Bytes		19

Realizado por: Flores, Pablo. 2017

$$\text{tiempo de trama} = 4\mu s \times 19 \text{ bytes} = 76\mu s = 0,076ms$$

4.3.3 Protocolo LonWorks

Según (Muñoz, 2015), una de las tramas típicas de datos utilizadas para activar una carga empleando el protocolo Lonworks, está conformada por los campos que se detallan en la siguiente tabla:

Tabla 4-13: Bytes Trama Típica de LonWorks

Campo	# Bytes
Encabezado	1
Formato de dirección – Longitud del dominio	1
Dirección de la fuente	2
Dirección destino	1
Tipo de servicio ID, número de transacción	1
Selector de variables de red	2
Datos	2
CRC	2
Total de Bytes	12

Realizado por: Flores, Pablo. 2017

$$\text{tiempo de Byte} = \frac{1s}{2000000bits} \times 8bits = 0,000004 s = 4\mu s$$

$$\text{tiempo de trama} = 4\mu s \times 12 \text{ bytes} = 48\mu s = 0,048ms$$

4.4 Análisis Comparativo de Protocolos:

Una vez que se han realizado las pruebas de comunicación, se han analizado las tramas de datos y velocidad de comunicación del protocolo propietario y de los protocolos domóticos comerciales, se pueden realizar comparaciones y establecer diferencias entre los mismos, con el objetivo de demostrar la funcionalidad y beneficios del bus propietario desarrollado en éste

proyecto. Cabe indicar que para las comparaciones se utilizan los tiempos teóricos calculados en todos los protocolos, en las mismas condiciones de Hardware (exceptuando el protocolo X-10).

La siguiente tabla indica un resumen de las características analizadas tanto del protocolo propietario como de los protocolos domóticos comerciales:

Tabla 4-14: Tabla Resumen Comparación de Protocolos

Campo	Protocolo Propietario	Protocolo X-10	Protocolo ZigBee	Protocolo Lonworks
Tiempo de bit	0,5 us	-	0,5 us	0,5 us
Tiempo de ciclo	-	16.666,7 us	-	-
Tiempo de Byte	4 us	66.666,8 us	4 us	4 us
# Bytes de una solicitud de lectura	6	-	20	12
Tiempo de trama para una solicitud de lectura	24 us	-	80 us	48 us
Tiempo de trama para una solicitud de escritura	0,044 ms	366,74 ms	0,08 ms	0,048 ms
Tiempo total para activar una carga, desde la lectura de las entradas	0,124 ms	783,49 ms	0,232 ms	0,144 ms

Realizado por: Flores, Pablo. 2017

Como se ha podido constatar en las pruebas realizadas el bus propietario presenta ventajas claramente perceptibles en comparación con los protocolos domóticos estudiados.

Las ventajas mencionadas, son las siguientes:

- El bus propietario en el protocolo de comunicación utiliza tramas de datos más compactas en comparación con los protocolos domóticos estudiados, tal es el caso que para activar un relé, la trama enviada por el maestro maneja únicamente 11 Bytes, de los cuales 6 Bytes son fijos y hacen referencia al empaquetamiento (dirección esclavo, función, dirección de inicio a escribir, número de Bytes a escribir y el CRC), además de 5 Bytes internos para los datos (tiempo de retardo, reserva y valor que indica la acción de encender o apagar).
- Al tener una trama compacta y con menos campos, influye directamente en el tiempo de transmisión de los paquetes de datos, lo cual se pudo constatar en la tabla 4-13, en la cual se observa que el protocolo propietario posee tiempos menores tanto para dar una sola instrucción como en el tiempo total que se necesita para activar una carga, es decir desde su lectura, procesamiento y ejecución. Dando de esta forma cumplimiento a la hipótesis planteada en un inicio, ya que el bus propietario desarrollado transmite sus tramas de datos a una velocidad mayor, en comparación con los protocolos domóticos más utilizados a nivel mundial.
- La trama utilizada en el protocolo propietario para encender o apagar un relé se puede optimizar aún más, eliminando los dos Bytes de Retardo y los dos Bytes de Reserva.
- El protocolo del bus propietario ha sido programado para que el maestro reciba una respuesta de confirmación de todas las instrucciones o solicitudes enviadas, lo que no ocurre en X-10, pues el protocolo no tiene respuesta o confirmación tanto de la recepción como de la ejecución de las solicitudes enviadas por el maestro, por lo que debe enviarse por duplicado las tramas para asegurar su recepción, generándose problemas graves cuando se presenta ruido, ya que al no llegar la instrucción, la acción no se ejecuta.
- ZigBee y Lonworks como se pudo constatar en el apartado 4.3 utilizan direcciones largas, varias funciones, subfunciones, datos de campo y otros tipos de identificadores de campo, generando entonces más información para ejecutar las instrucciones; esto se debe a que los protocolos son estándar y tratan de que su alcance cubra todo tipo de aplicación domótica, ya sea para viviendas, edificios o condominios, lo cual les obliga a dejar Bytes libres en sus tramas, generando tramas largas y ociosas.

Por otro lado, las posibles desventajas que puede tener el protocolo propietario en comparación con los protocolos domóticos estudiados son las siguientes:

- El bus propietario desarrollado es exclusivo para cargas de control ON/OFF, lo cual secciona el control que se puede tener en una vivienda, pero a la vez otorga la posibilidad de seguir investigando y desarrollando un sistema que englobe todas las necesidades de domótica.
- El protocolo de comunicación utilizado en el bus propietario utiliza una tabla fija de programación y no una interfaz en la cual se pueda programar y/o modificar las cargas o escenarios que se desean activar según las necesidades particulares de cada vivienda.
- La aplicación domótica es susceptible a ruido debido a que se utiliza el medio de transmisión de radio frecuencia, por lo que hay que tener un especial cuidado en cuanto a realizar las pruebas en lugares que no estén cercanos a equipos WI-FI, antenas celulares, y otras fuentes generadoras de ruido.
- El protocolo de comunicación desarrollado utiliza solamente tres funciones con las cuales se ejecutan las transacciones, por lo que se pueden implementar funciones específicas que permitan realizar acciones que se encuentren ya programadas en la memoria del elemento esclavo.

CAPITULO V

5 PROPUESTA DE TRABAJOS FUTUROS

Como continuación de este proyecto y al igual que en la mayoría de proyectos de investigación, existe diversas líneas de investigación que quedan abiertas y en las que es posible seguir trabajando, pudiendo entonces elaborar estudios que permitan mejorar las características del bus propietario desarrollado.

En esta tesis, los esfuerzos de investigación se concentraron en el desarrollo del protocolo de comunicación del bus propietario, lo cual permitió obtener una alternativa de comunicación específica para aplicaciones domóticas a través de la especificación de una trama de datos compacta y una mayor velocidad de transmisión con respecto a los protocolos existentes. Sin embargo el protocolo propietario brinda las herramientas necesarias para seguir investigando y mejorar de esa forma las características del mismo.

Una de las opciones que pueden tomarse para retomar este proyecto consiste en crear funciones específicas en el protocolo, las cuales permitan realizar instrucciones preestablecidas en la programación, generando así tramas de datos más complejos pero aún más compactos que las actuales. Cabe resaltar que el protocolo propietario trabaja solamente con tres funciones: Leer, Escribir y Borrar, por lo que existe la amplia posibilidad de desarrollar funciones específicas, como por ejemplo:

- Función “broadcast - específica” para encender únicamente las cargas luminosas de la vivienda, más no todas las cargas instaladas de control On/OFF.
- Función “reset” para reiniciar los microcontroladores cuando uno de ellos por algún motivo no pueda sincronizarse.
- Función para encender ciertos dispositivos en una vivienda.

Éstas funciones específicas son un ejemplo del amplio campo de investigación que se tiene, siendo su finalidad la de mejorar la versatilidad del protocolo propietario, tener tramas de datos más compactas y de esa forma mejorar incluso más, los tiempos de envío de transacciones de datos.

CONCLUSIONES

Las pruebas realizadas en el bus propietario, han evidenciado el funcionamiento idóneo de cada una de las partes que lo conforman, dando cumplimiento a los objetivos y alcances planteados en el proyecto.

La hipótesis planteada en el anteproyecto ha sido demostrada a través de medios estadísticos, los cuales indican que el bus propietario desarrollado transmite sus tramas de datos a una velocidad mayor, en comparación con los protocolos domóticos más utilizados a nivel mundial.

La red implementada por un maestro, tres esclavos CSE y tres esclavos CSS tiene una comunicación exitosa, en la cual todas las instrucciones recibidas se ejecutan en tiempo real según la percepción humana, lo cual indica que la comunicación no sufre altibajos ni pérdida de información.

Las funciones que implementa el protocolo desarrollado son únicamente tres: leer registros, escribir registros y eliminar registros; solo con éstas funciones se realizan múltiples instrucciones entre el maestro y sus esclavos.

La velocidad de transmisión no depende del protocolo de comunicación desarrollado en el bus propietario sino más bien del hardware, en este caso del transreceptor nRF24L01 2.4GHz el cual se puede programar para trabajar a tres velocidades: 250000 bps, 1Mbps y 2Mbps. Por otro lado, el tiempo necesario para enviar o recibir una trama de datos, si depende exclusivamente del protocolo propietario, ya que al trabajar con menor cantidad de Bytes, ocasiona que el tiempo de envío y recepción de las instrucciones disminuya.

El protocolo de comunicación del bus propietario no ha sido implementado en la comunicación Bluetooth entre el maestro y el teclado celular porque es una conexión punto – punto, y el protocolo es una conexión maestro – multiesclavo y no está dentro del alcance del proyecto; se considera entonces a la aplicación del celular como un teclado local conectado directamente al maestro.

El protocolo propietario deshabilitó funciones que vienen predefinidas por hardware en los transreceptores nRF24L01 2.4GHz, tales como: bit de reconocimiento ACK (acknowledgment) y control de errores CRC para tener una comunicación más limpia; pero lo que si se mantiene son las secuencias de Bytes generadas en el inicio y finalización del modo recepción.

Pese a que en un inicio el bus propietario iba a implementarse sobre una red configurada por 5 dispositivos, es decir un maestro y cuatro esclavos, las necesidades de la aplicación domótica han permitido extender la red a seis esclavos, de los cuales, tres esclavos manejan las entradas (CSE) y tres esclavos manejan las salidas (CSS), ratificando y comprobando que el software del protocolo permite incrementar dispositivos sin restricciones, hasta llegar a un máximo de 250 elementos.

El protocolo propietario es versátil, es decir puede ser utilizado con cualquier tipo de microcontrolador o medio de transmisión, acoplándose a las características y necesidades de cada uno de ellos.

RECOMENDACIONES

No utilizar los medios de comunicación RF en lugares cercanos a equipos WI-FI, antenas celulares, y otras fuentes generadoras de ruido, ya que pueden interrumpir los canales utilizados y generar datos erróneos, tramas incompletas e incluso la pérdida de la comunicación, impidiendo de esta forma la ejecución de las solicitudes.

Desarrollar el protocolo de comunicación del bus propietario en microcontroladores con mejores características para obtener mayores prestaciones, un control más ampliado de todas las acciones y sobre todo aprovechar al máximo todos los beneficios del protocolo.

Utilizar una fuente independiente para alimentar las tarjetas de relés, ya que al ser conectadas directamente a los microcontroladores arduinos, éstos no soportan la cantidad de amperaje necesario, teniendo el riesgo de quemarse, es por eso que lo más idóneo sería utilizar tarjetas de relés, separadas en dos áreas gracias a un octo-acoplador, en las cuales la parte de control se conecta al arduino y la parte del actuador se maneja con una fuente independiente de 5Vdc.

Alimentar los transreceptores nRF24L01 2.4GHz al voltaje nominal para que el rango de alcance de la transmisión de envío y recepción de datos no disminuya y garantizar que el mismo sea el especificado por el fabricante.

Enviar las transacciones del protocolo en el modo *broadcast*, dos o tres veces para reducir la probabilidad de errores, garantizar que llegue a todos los esclavos y de esta forma pueda ser ejecutada.

BIBLIOGRAFÍA

Allen, S. (2005). *Easy X10 Projects for creating a Smart Home* . *Indy - Tech Publishing*.

Barrera, M. (2011). Domótica e Inmótica: visión general y actualidad. *Revista AIE UdeA*, 13-19.

Barrera, M., Londoño, N., Carvajal, J., & Fonseca, A. (2012). Análisis y Diseño de un Prototipo de Bajo Costo. *Revista Facultad Ingeniería Universidad Antioquía N° 63*, 117-128.

Cázares, G., López, C., López, A., Morales, J., Montoya, A., Castillo, H., & Fonseca, J. (2011). Diseño de un prototipo didáctico para la implementación de redes de sensores inalámbricos basados en el protocolo ZigBee. *Ra Ximhai*, 315-324.

Chan, F., Núñez, K., Torres, E., Ortegón, J., Aguilar, O., & Toral, H. (2011). Análisis de sensibilidad al ruido en un sistema de comunicación sobre línea eléctrica para aplicaciones en Domótica empleando el protocolo X-10. *Ingeniería*, vol. 15, núm. 3, 147 - 156.

Dignani, J. (2011). *Análisis del Protocolo ZigBee*. Ciudad de la Plata: Facultad de Informática, Universidad Nacional de la Plata.

Held, G. (2006). *Understanding Broadband over Power Line*. *Auerbach Publications*.

Herrera, L. (2005). Viviendas Inteligentes (Domótica). *Revista Ingeniería e Investigación* (58), 47-53.

IEEE.802.15.4. (Septiembre de 2006). Wireless Medium Access Control and Physical Layer Specifications for Low –Rate Wireless Personal Area Networks.

Junestrand, S. (2005). *Domótica y Hogar Digital*. Thomson Paraninfo. España.

Longares, J. (2013). Introducción a ZigBee y las Redes de Sensores Inalámbricas. *Arte en 8 Bits*.

Lonworks. (2012). Protocolo Comunicaciones Lonworks. Automatización de Edificios.

Modbus-Organization. (2006). MODBUS Over Serial Line, Specification and Implementation Guide, V1.02 ed.

Moreaux, P., Sartor, F., & Vernier, F. (2012). An Effective Approach for Home Services Management. *20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, (págs. 44-51).

Muñoz, F. (2015). DT Lonworks. Universidad Politécnica de Madrid.

nRF24L01-Datasheet. (s.f.). nRF24L01 + Single Chip 2.4GHz Transceiver. Preliminary Product Specification.

Rodríguez, M. (Octubre de 2014). Sistema de Control Remoto para Aplicaciones Domóticas a través de Internet. Madrid: Universidad Autónoma de Madrid.

Romero, C., Vásquez, F., & Castro, C. (2007). *Domótica e Inmótica. Viviendas y Edificios Inteligentes. 2da Edición*. México: Alfaomega Grupo Editor.

Semplades. (2013 - 2017). *Plan Nacional Para el Buen Vivir*. Quito.

Sosa, E. (Febrero de 2011). Contribuciones al establecimiento de una red global de sensores inalámbricos interconectados. Tesis Doctoral UNLP.

The Modbus Organization. (2006). *MODBUS Over Serial Line, Specification and Implementation Guide* (V1.02 ed.).

Yunaningsih, R. (2009). X10 Protocol Man Machine Interface Implementation using Labview. *Journal of Applied Science*.

Zamora, M., Santa, J., & Gómez, A. (2010). An integral and networked home automation solution for indoor ambient intelligence. *Pervasive Computing IEEE*, 66-77.

ANEXOS

ANEXO A: TABLAS DE PROGRAMACIÓN

REGISTRO PRESETS:

Tabla A-1: Tabla de Registro Presets

REGISTROS PRESETS									
DIRECCIÓN DE MEMORIA	PRESET		CSS_ID	# SALIDA	RETARDO		RESERVADO		VALOR
	ID	REGISTRO			H	L	H	L	
0	001	01	1	1	0	0	0	0	0
7	002	01	1	1	0	0	0	0	1
14	003	01	1	2	0	0	0	0	0
21	004	01	1	2	0	0	0	0	1
28	005	01	1	3	0	0	0	0	0
35	006	01	1	3	0	0	0	0	1
42	007	01	1	4	0	0	0	0	1
49		02	1	5	0	0	0	0	0
56	008	01	1	4	0	0	0	0	0
63		02	1	5	0	0	0	0	1
70	009	01	1	6	0	0	0	0	0
77	010	01	1	6	0	0	0	0	1
84	011	01	1	7	0	0	0	0	0
91	012	01	1	7	0	0	0	0	1
98	013	01	1	8	0	0	0	0	0
105	014	01	1	8	0	0	0	0	1
112	015	01	2	1	0	0	0	0	0
119	016	01	2	1	0	0	0	0	1
126	017	01	2	2	0	0	0	0	0
133	018	01	2	2	0	0	0	0	1
140	019	01	2	3	0	0	0	0	0
147		02	3	4	0	0	0	0	0
154	020	01	2	3	0	0	0	0	1
161		02	3	4	0	0	0	0	1
168	021	01	2	4	0	0	0	0	0
175		02	3	4	0	0	0	0	0
182	022	01	2	4	0	0	0	0	1
189		02	3	4	0	0	0	0	1
196	023	01	2	5	0	0	0	0	0
...

PRESET CACHÉ:

Tabla A-2: Tabla Preset Caché

PRESET CACHÉ						
PRESET ID	# REGISTROS	PRESET_ID		DIRECCIÓN		
		DEC	HEX	PRESET_ID_HI	PRESET_ID_H	PRESET_ID_L
001	01	0	000000	0	0	0
002	01	07	000007	0	0	7
003	01	14	00000E	0	0	14
004	01	21	000015	0	0	21
005	01	28	00001C	0	0	28
006	01	35	000023	0	0	35
007	02	42	00002A	0	0	42
008	02	56	000038	0	0	56
009	01	70	000046	0	0	70
010	01	77	00004D	0	0	77
011	01	84	000054	0	0	84
012	01	91	00005B	0	0	91
013	01	98	000062	0	0	98
014	01	105	000069	0	0	105
015	01	112	000070	0	0	112
016	01	119	000077	0	0	119
017	01	126	00007E	0	0	126
018	01	133	000085	0	0	133
019	02	140	00008C	0	0	140
020	02	154	00009A	0	0	154
021	02	168	0000A8	0	0	168
022	02	182	0000B6	0	0	182
023	01	196	0000C4	0	0	196
024	01	203	0000CB	0	0	203
025	01	210	0000D2	0	0	210
026	01	217	0000D9	0	0	217
027	01	224	0000E0	0	0	224
028	01	231	0000E7	0	0	231
029	01	238	0000EE	0	0	238
030	01	245	0000F5	0	0	245
031	01	252	0000FC	0	0	252
032	01	259	000103	0	1	3
033	01	266	00010A	0	1	10
034	01	273	000111	0	1	17
035	01	280	000118	0	1	24
036	01	287	00011F	0	1	31
037	02	294	000126	0	1	38
038	02	308	000134	0	1	52
039	13	322	000142	0	1	66
040	13	413	00019D	0	1	157
041	04	504	0001F8	0	1	248
042	04	532	000214	0	2	20
043	17	560	000230	0	2	48
044	17	679	0002A7	0	2	167
045	01	798	00031E	0	3	30
046	01	805	000325	0	3	37

ENTRADAS PRESET CSE:

Tabla A-3: Tabla de Entradas Preset CSE

ENTRADAS PRESETS CSE											
CSE		PRESIÓN TIPO	CAMPO								
ID	ENTRADA		CERRADO			ABIERTO			DOBLE CLIC		
		COMANDO	PRESET_ID_H	PRESET_ID_L	COMANDO	PRESET_ID_H	PRESET_ID_L	COMANDO	PRESET_ID_H	PRESET_ID_L	
1	1	3	1	0	1	1	0	2	1	0	0
1	2	3	1	0	1	1	0	2	1	0	0
1	3	3	1	0	3	1	0	4	1	0	0
1	4	3	1	0	5	1	0	6	1	0	0
1	5	3	1	0	5	1	0	6	1	0	0
1	6	3	1	0	5	1	0	6	1	0	0
1	7	3	1	0	7	1	0	8	1	0	0
1	8	3	1	0	9	1	0	10	1	0	0
2	1	3	1	0	11	1	0	12	1	0	0
2	2	3	1	0	13	1	0	14	1	0	0
2	3	3	1	0	15	1	0	16	1	0	0
2	4	3	1	0	17	1	0	18	1	0	0
2	5	3	1	0	20	1	0	19	1	0	0
2	6	3	1	0	22	1	0	21	1	0	0
2	7	3	1	0	23	1	0	24	1	0	0
2	8	3	1	0	25	1	0	26	1	0	0
3	1	3	1	0	27	1	0	28	1	0	0
3	2	3	1	0	29	1	0	30	1	0	0
3	3	3	1	0	31	1	0	32	1	0	0
3	4	3	1	0	33	1	0	34	1	0	0
3	5	3	1	0	35	1	0	36	1	0	0
3	6	3	1	0	0	1	0	0	1	0	0
3	7	3	1	0	0	1	0	0	1	0	0
3	8	3	1	0	41	1	0	42	1	0	0

PRESIÓN	
TIPO	VALOR
Toggle	1
Single	2
Dual	3
Advanced Toggle	4

Realizado por: Flores, Pablo. 2017

ANEXO B: SECUENCIAS DE INICIO Y FIN DEL MODO RECEPCIÓN

Dentro de las comunicaciones realizadas a través de Radio Frecuencia, para detectar la llegada de datos en el buffer de recepción es necesario configurar previamente el módulo nRF24L01 en modo recepción, para lo cual debe generarse una secuencia periódica de inicio con la finalidad de monitorear la presencia de datos en el buffer indicado; y una vez que se haya finalizado de recibir dichos datos, el módulo nRF24L01 debe ser configurado en modo transmisión para poder enviar una respuesta, debiendo generarse una secuencia de finalización del modo recepción. La secuencia de inicio del modo recepción, necesarias para las transacciones de datos, se muestra en la figura B-1.

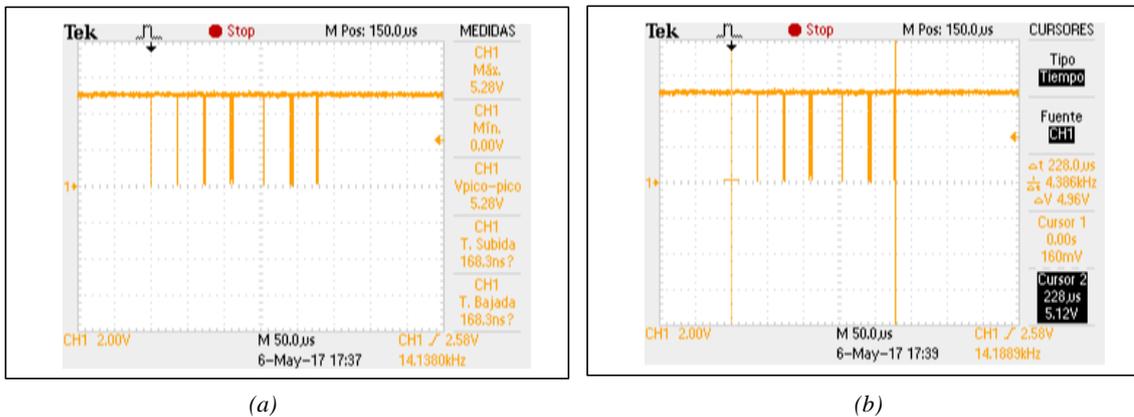
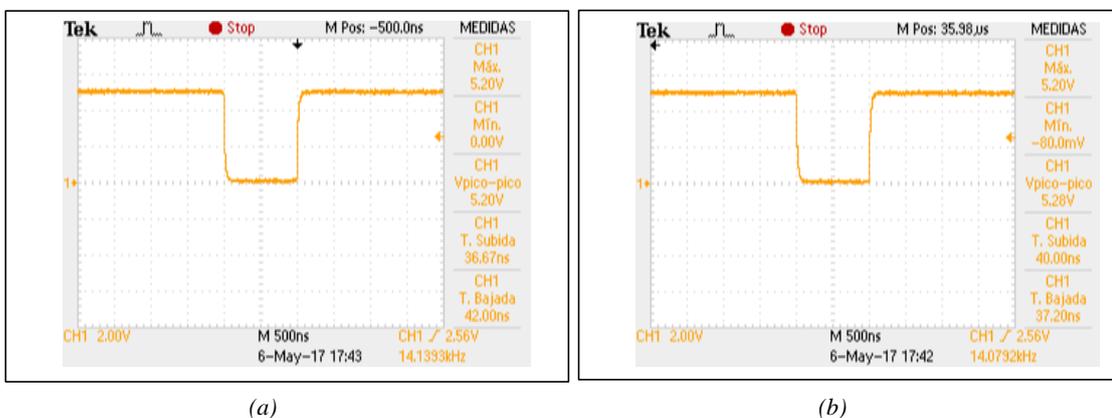
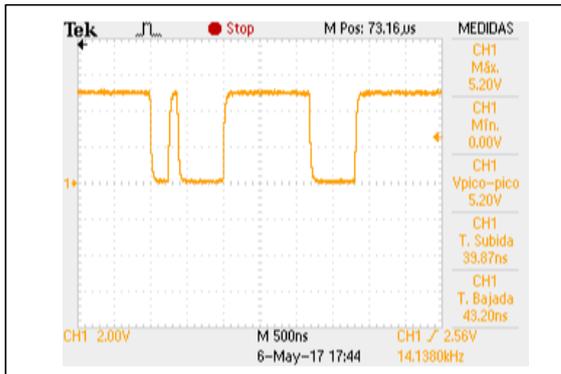


Figura B-1: Secuencia de Inicio Modo Recepción: (a) Bytes; (b) Tiempo de los Bytes

Realizado por: Flores, Pablo. 2017

Cada Byte de la secuencia de inicio del modo recepción se observa en las figura B-2.

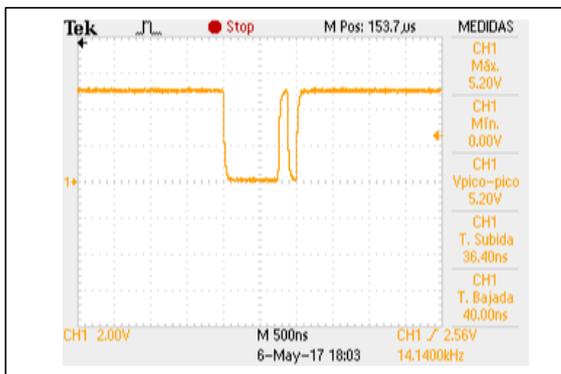




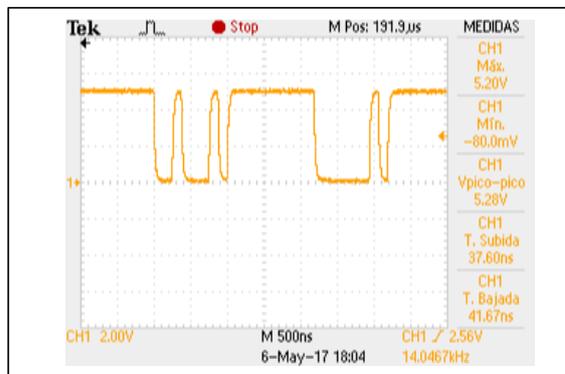
(c)



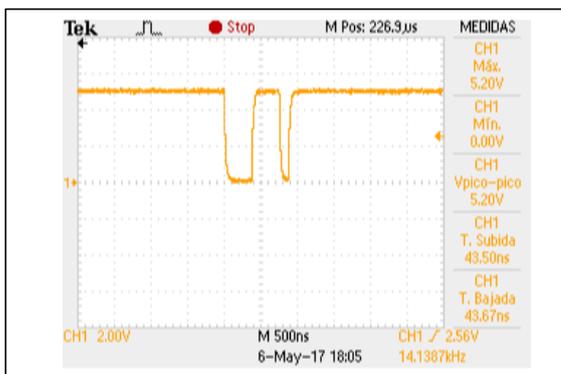
(d)



(e)



(f)



(g)

Figura B-2: Bytes Secuencia de Inicio Modo Recepci3n: (a) Byte Uno; (b) Byte Dos; (c) Byte Tres; (d) Byte Cuatro; (e) Byte Cinco; (f) Byte Seis; (g) Byte Siete.

Realizado por: Flores, Pablo. 2017

La secuencia de finalización del modo recepción, necesarias para las transacciones de datos, se muestra en la figura B-3.

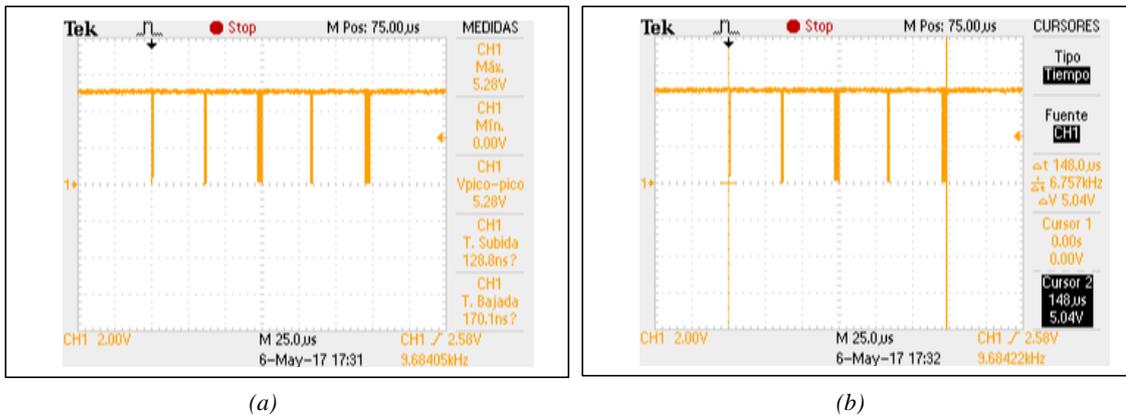
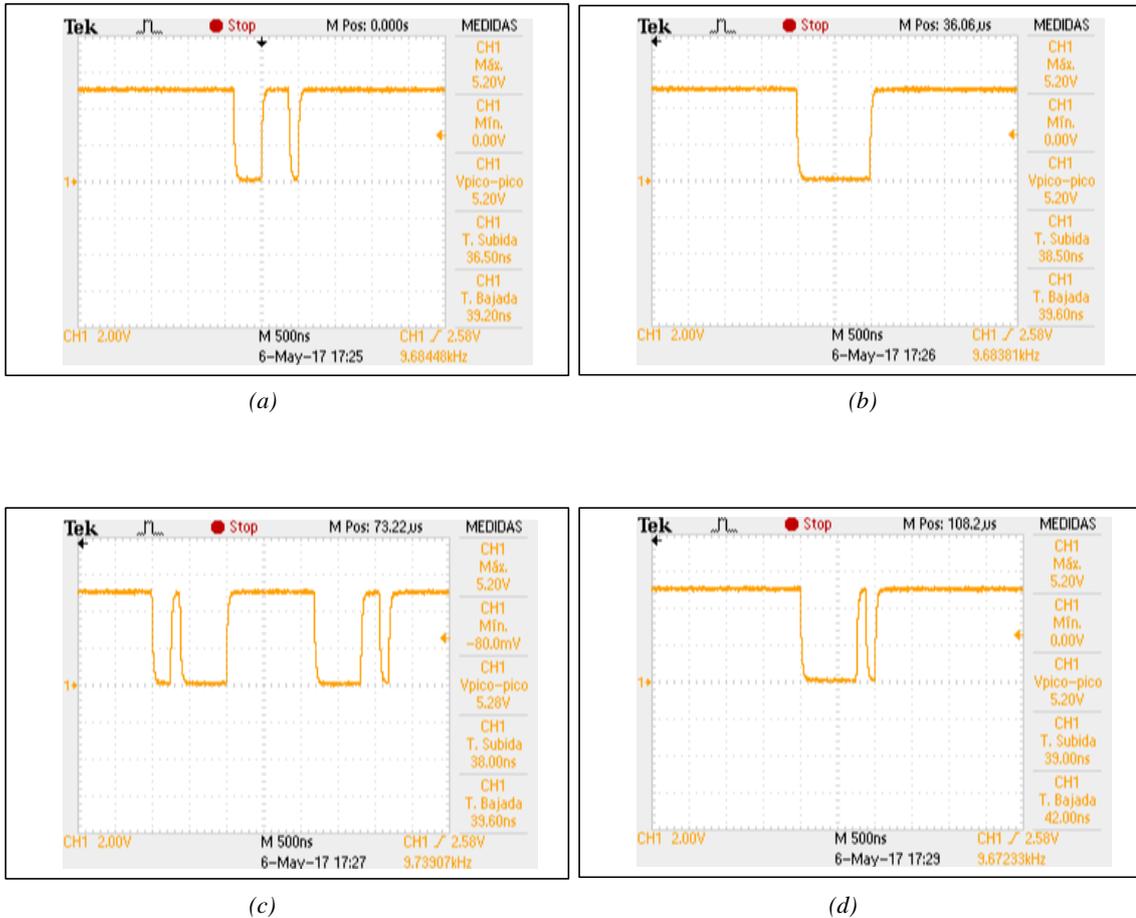
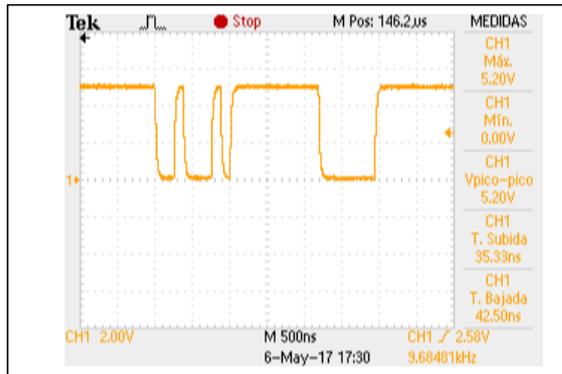


Figura B-3: Secuencia Finalización Modo Recepción: (a) Bytes; (b) Tiempo de los Bytes

Realizado por: Flores, Pablo. 2017

Cada Byte de la secuencia de finalización del modo recepción se observa en las figura 9-4.





(e)

Figura B-4: Bytes Secuencia de Finalización Modo Recepción: (a) Byte Uno; (b) Byte Dos; (c) Byte Tres; (d) Byte Cuatro; (e) Byte Cinco.

Realizado por: Flores, Pablo. 2017

ANEXO C: PLANOS APLICACIÓN DOMÓTICA

FACHADA:



FACHADA FRONTAL



FACHADA LATERAL IZQUIERDA



FACHADA POSTERIOR



FACHADA LATERAL DERECHA

Figura C-1: Fachada de la Aplicación Domótica

Realizado por: Flores, Pablo. 2017

PLANO GENERAL:



Figura C-2: Plano General de la Aplicación Domótica

Realizado por: Flores, Pablo. 2017

PLANO PLANTA BAJA:

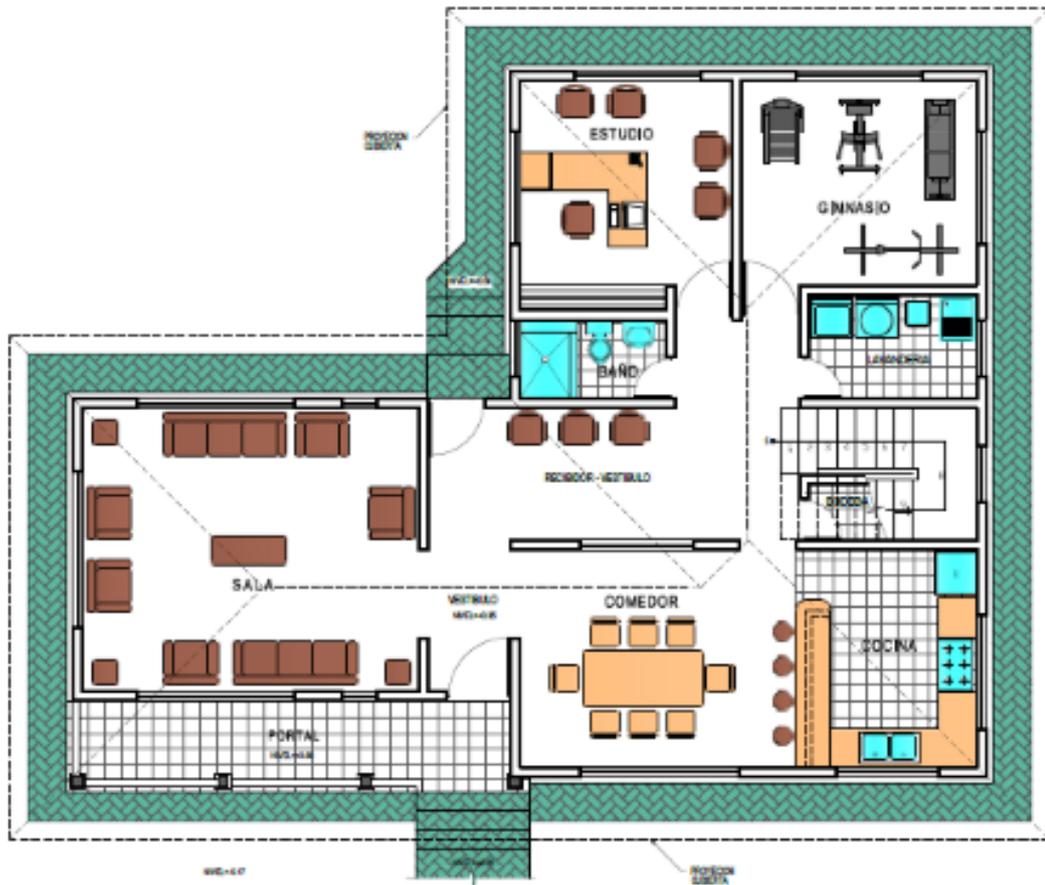


Figura C-3: Plano Planta Baja de la Aplicación Domótica

Realizado por: Flores, Pablo. 2017

PLANO BUHARDILLA:

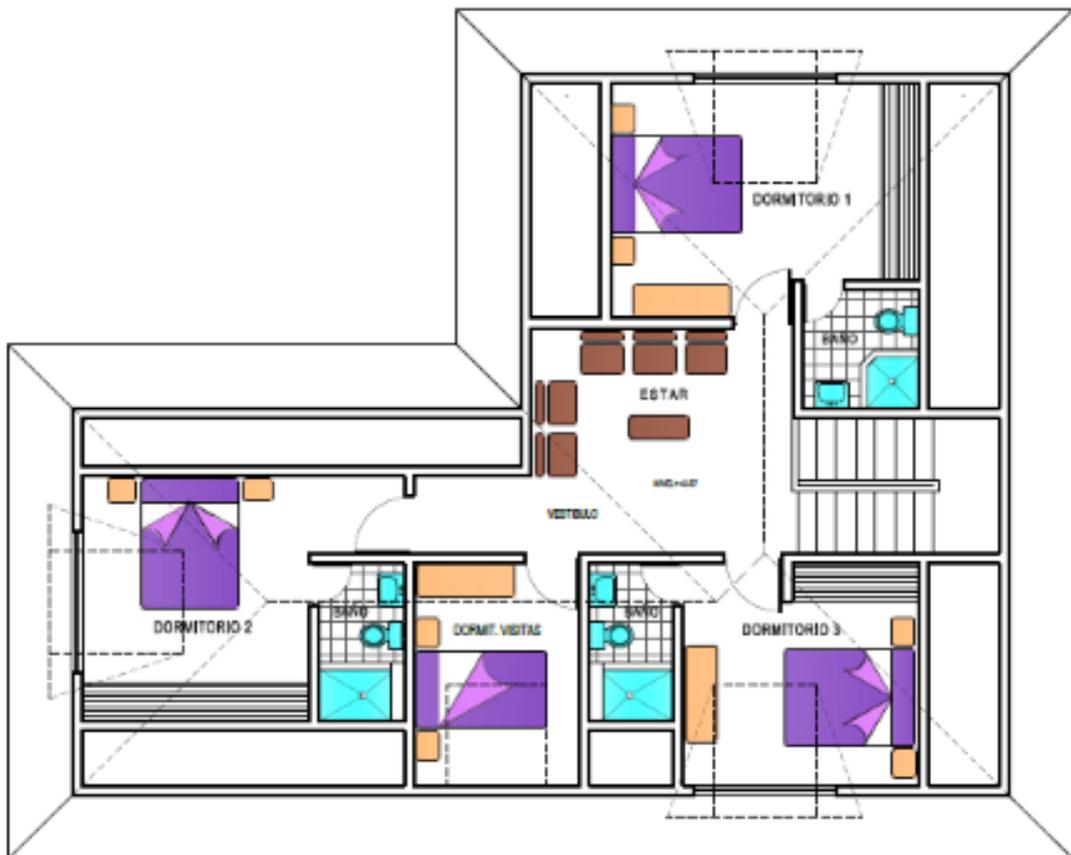


Figura C-4: Plano Planta Buhardilla de la Aplicación Domótica

Realizado por: Flores, Pablo. 2017