

UNIVERSIDAD NACIONAL DE SAN MARTÍN

ESCUELA DE CIENCIA Y TECNOLOGÍA



UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

**Interfaz de Adquisición, Control y
Automatización con Tecnología ZIGBEE**

PROYECTO FINAL INTEGRADOR PARA OBTENER EL GRADO DE
INGENIERO ELECTRÓNICO

Autor

GASTON MACIEL

gastonmaciel2@gmail.com

Tutores

Ing. Sergio Caprile

scaprile@cika.com.ar

2020

Agradecimientos

Índice general

Agradecimientos	2
1. Introducción	12
1.1. Motivaciones y Objetivos del Proyecto	13
1.1.1. Motivaciones	13
1.2. Objetivos	14
2. Introducción a la comunicación inalámbrica	16
2.1. Introducción	16
2.2. Tipos de Comunicaciones	20
3. ZigBee	23
3.1. Introducción a ZigBee	23
3.2. Estándar ZigBee	25
3.2.1. Capa Física (PHY)	26
3.2.2. Capa de Control de Acceso al Medio (MAC)	40
3.2.3. Capa de Red (NWK)	48
3.2.4. Capa de aplicación (APL)	55
3.3. Comparativa ZigBee con otras tecnologías	64
4. Dispositivos XBee	66
4.1. Tipos	66
4.2. Características Generales XBee Serie 2	69
4.2.1. Modos de Operación	69
4.2.2. Comunicación con el Usuario	72

5. Modo API	75
5.1. Estructura de frame API	75
5.2. Comandos y Frames	77
5.2.1. Frame 0x17 (Envío de Comando Remoto)	79
5.2.2. Frame 0x97 (Respuesta de Comando AT Remoto)	81
5.2.3. Frame 0x92 (Recepción de Estados I/O de Dispositivo Remoto)	83
6. Consideraciones de Hardware y Software	87
6.0.1. Alimentación	88
6.0.2. Entradas Digitales	89
6.0.3. Entradas Analógicas	90
6.0.4. Salidas Digitales	91
6.0.5. Aplicación de Software	91
7. Desarrollo	93
7.1. Hardware	93
7.1.1. Regulación de Voltaje	95
7.1.2. Entradas Digitales	101
7.1.3. Entradas Analógicas	107
7.1.4. Circuito de Salidas Digitales	110
7.2. Software de Aplicación	115
7.2.1. Instrumentación Virtual	115
7.2.2. LabVIEW	116
7.2.3. Estructura de Recepción	125
7.2.4. Parsing	131
7.2.5. Generador de Comando	136
7.2.6. Casting	139
7.2.7. Detector de Alarma	141
7.2.8. Envío de Correo Electrónico	144
7.2.9. Pantalla Principal	147
7.3. Configuración de Red	156

8. Resultados	167
8.1. RSSI	167
8.2. Throughput	174
8.3. Throughput en Red con Múltiples Saltos	177
8.4. Consumo	179
8.5. Recepción de Frames	181
8.6. Envío de Comando Remoto	193
8.7. Envío de Alarma	195
8.8. Lecturas ADC 0-10V/4-20mA	200
8.9. Conclusiones	204
8.10. Posibles Trabajos Futuros	207
A. Gerenciamiento del Proyecto	211
A.1. Anteproyecto	211
A.2. Gestión de Riegos	214
A.3. Cash Flow	217
A.3.1. VAN y TIR	218
Bibliografía	219

Índice de figuras

2.1. Espectro Electromagnético, figura extraída de [5].	17
2.2. Tecnologías asociadas al tipo y alcance de red, imagen extraída de [16].	21
3.1. Imagen oficial de ZigBee Alliance haciendo referencia a la gran variedad de campos de aplicación [4].	24
3.2. Capas de protocolo ZigBee[2].	25
3.3. Estructura de canales[12].	26
3.4. Trama de sincronismo nivel físico.	28
3.5. Modulación BPSK.	30
3.6. Diagrama de bloques simplificado, modulador BPSK.	31
3.7. Diagrama de bloques simplificado, demodulador BPSK.	31
3.8. Modulación QPSK.	34
3.9. Constelación de modulación QPSK.	35
3.10. Diagrama de bloques, modulador QPSK.	36
3.11. Diagrama de bloques, demodulador QPSK.	36
3.12. Etapas de modulación, transmisor y receptor[17].	38
3.13. DSSS en el Transmisor[17].	39
3.14. DSSS en el receptor[17].	39
3.15. Secuencia de transmisión con DSSS[17].	40
3.16. Barrera de espectro ensanchado bajo el nivel de ruido[17].	40
3.17. Frame para el envío de información.	41
3.18. Trama con más detalle.	42
3.19. El frame control incluido en el encabezado indica el tipo de trama de datos.	42

3.20. El frame control incluido en el encabezado indica el tipo de trama de comando.	43
3.21. El frame control incluido en el encabezado indica el tipo de trama ACK.	43
3.22. El frame control incluido en el encabezado indica el tipo de trama ACK.	43
3.23. Diagrama de flujo del algoritmo CSMA-CA[6].	45
3.24. Topología estrella.	50
3.25. Topología árbol.	50
3.26. Topología malla.	51
3.27. Trama NWK.	55
3.28. Trama APS.	60
3.29. Trama de datos APS.	61
3.30. Trama de comandos APS.	61
3.31. Trama de ACK de APS.	61
3.32. Relación entre tramas 802.15.4, NWK y APS.	62
3.33. Interacción entre dispositivos.	63
4.1. XBee Serie 1[1].	66
4.2. XBee Serie 2[1].	67
4.3. XBee PRO Serie 1 y Serie 2[1].	68
4.4. XBee PRO Serie 3[1].	68
4.5. Tipo de antenas.	69
4.6. Estructura de envío de Comando AT[11].	70
4.7. Comunicación Transparente entre módulos XBee.	72
4.8. Ejemplo de configuración para envío y recepción transparente[11].	73
4.9. Comunicación API entre módulos XBee[9].	74
5.1. Estructura de trama modo API[11].	75
6.1. Configuración típica del regulador LM2596. Imagen tomada de [14].	89
7.1. Adaptador mini USB Serial para módulos XBee.	94
7.2. Imagen representativa.	95
7.3. Circuito de regulación de voltaje.	96
7.4. Curva típica de corriente RMS vs tensión del capacitor.	97

7.5. Tabla de selección de capacitor de salida.	98
7.6. Valores de selección de inductor para LM2596-ADJ.	99
7.7. Tabla de selección de diodo.	100
7.8. Modulo Xbee.	101
7.9. Circuitos para entradas digitales.	102
7.10. Respuesta Schmitt Trigger.	104
7.11. Curvas CTR lineal y en saturación con respecto a la corriente de entrada.	105
7.12. Curvas de corriente de colector vs corriente de entrada en función de la temperatura.	106
7.13. Circuito de adquisición para señales analógicas con protecciones, amplificador Rail to Rail y divisor resistivo.	108
7.14. Circuito de salida a relé.	110
7.15. Diseño de Board en Eagle.	111
7.16. Capa Top.	112
7.17. Capa Bottom.	112
7.18. Capa Top.	113
7.19. Capa Bottom.	113
7.20. Logo LabVIEW creado por National Instruments[13].	117
7.21. Estructura de Software[18].	118
7.22. Pantalla de Panel Frontal.	119
7.23. Pantalla de Diagrama de Bloques.	120
7.24. Barra de herramientas principal del LabVIEW.	120
7.25. Controles del Panel Frontal.	122
7.26. Funciones del Diagrama de Bloques.	124
7.27. Funciones disponibles para el desarrollo con Queues.	127
7.28. Patrón Productor/Consumidor con Queues.	127
7.29. Patrón Productor.	128
7.30. Estructura While(Izquierda) y Case(Medio) y Estructura For(Derecha).	128
7.31. Funcion Bytes at Port(Izquierda),Función de comparación(Medio) y Función de lectura de puerto(Derecha).	129
7.32. Patrón Consumidor.	130

7.33. De izquierda a derecha, Unión de Bytes, Función de Suma, Función de Comparación Igual, Función de Resta, Función de Formación de Cluster(Bundle).	130
7.34. Ícono del VI Receptor con sus entradas y salidas.	131
7.35. Elemento Virtual de Análisis y Guardado.	132
7.36. De izquierda a derecha, Función de Extracción por Índice, Conversion a Binario, Conversion a Cluster, Extracción de elementos del Cluster, Indicador.	132
7.37. De izquierda a derecha, Función de Extracción por Índice, Unión de Bytes, Conversión a Double, Indicador.	133
7.38. Variables Globales utilizadas en el proyecto.	134
7.39. Configuración de VI para escritura.	135
7.40. A la izquierda se observa la Herramienta de combinación de señales y a la derecha el VI para crear archivos EXcel.	136
7.41. Ícono virtual con puertos de entrada.	136
7.42. Instrumento Virtual para generar estructura del frame de comando.	137
7.43. De izquierda a derecha, Convertir Número a Variable de 2 Bytes, Suma de los Elementos de un Array, Conversión de String a Byte, Construir Array, Suma Múltiple, Función de Formación de Cluster(Bundle) y la Función de Resta.	138
7.44. Ícono de instrumento virtual para la creación de frames con sus puertos de entrada y salida.	139
7.45. Instrumento virtual para realizas el casting.	140
7.46. De izquierda a derecha, Función de Casteo, Función de Conversión a Byte, Constructor de Array y Lectura de Tamaño de Array.	140
7.47. Ícono del Parsing con puertos correspondientes.	141
7.48. Estructura para la detección de magnitud fuera de rango.	141
7.49. Icono correspondiente al detector, con sus tres entradas y dos salidas.	144
7.50. Estructura para el envío automático de correo electrónico.	145

7.51. De izquierda a derecha, Función de Concatenación de Strings, Constructor de Array, Configuración SMTP, Configuración de Destino, Herramienta de Texto Adjunto, Función de Envío de Correo, Cierre de Herramientas de Correo.	145
7.52. Ícono para el envío de correo electrónico.	146
7.53. Diagrama de bloques principal del sistema.	148
7.54. Configuración de puerto serie.	150
7.55. Inicialización de Queues para el uso de datos.	151
7.56. Ingreso de datos a analizar.	151
7.57. Valores de lectura analógicos.	152
7.58. Valores de lectura digitales.	152
7.59. Panel Frontal de variables.	153
7.60. Proceso de envío de correo por alarma.	154
7.61. Ingreso de destinatario.	155
7.62. Proceso de envío de comando.	155
7.63. Firmware API a utilizar en el proyecto para coordinador, router y end device.	159
7.64. Comunicación Coordinador y End Device.	161
7.65. Comunicación Coordinador y Múltiples End Devices.	162
7.66. Comunicación Coordinador y Múltiples End Devices.	163
7.67. Red Mesh, comunicación Coordinador, Múltiples Routers y End Device.	164
8.1. Imagen representativa de las zonas de Fresnel [19] para el caso n=1 (primer zona).	169
8.2. Gráfico del modelo de reflexiones en el suelo	170
8.3. Resultados RSSI en función de la distancia para los distintos modelos propuestos.	172
8.4. Resultados RSSI en función de la distancia y potencia de transmisión.	174
8.5. Resultados medición de throughput en función de la velocidad de transmisión y tamaño del paquete.	176
8.6. Resultados medición de throughput en función de configuración de red y tamaño del paquete.	178
8.7. Oscilograma para el estudio de corriente consumida.	180

8.8. Oscilograma para el estudio de corriente consumida.	181
8.9. Recolección de variables y guardado.	182
8.10. Celdas de Excel correspondientes al BackUp.	183
8.11. Resultados obtenidos.	184
8.12. Parsing con estructura Case adicional.	186
8.13. Celdas de inicio y fin de guardado correspondiente al dispositivo A. .	187
8.14. Celdas de inicio y fin de guardado correspondiente al dispositivo B. .	187
8.15. Resultados obtenidos con dos End Devices.	188
8.16. Resultados obtenidos con dos End Devices.	189
8.17. Inicio y Fin de guardado correspondiente al dispositivo A.	190
8.18. Inicio y Fin de guardado correspondiente al dispositivo B.	191
8.19. Inicio y Fin de guardado correspondiente al dispositivo C.	191
8.20. Resultados obtenidos con tres End Devices.	192
8.21. Panel de configuración para envío de comando remoto.	194
8.22. Resultados obtenidos a partir de la recepción luego de enviar el comando.	195
8.23. Resultados obtenidos a partir de la recepción luego de enviar el comando.	197
8.24. Mails recibidos por alarma en la variable ADC2.	197
8.25. Mails recibidos por alarma en la variable ADC0.	198
8.26. Mails recibidos por alarma en la variable ADC1.	198
8.27. Mails recibidos por alarma en la variable ADC2.	198
8.28. Mails recibidos por alarma ADC0 Y ADC1.	199
8.29. Mails recibidos por alarma ADC0 Y ADC2.	199
8.30. Mails recibidos por alarma ADC1 Y ADC2.	199
8.31. Mails recibidos por alarma ADC0, ADC1 Y ADC2.	200
8.32. Hardware final.	206
8.33. Hardware final.	207
A.1. Flujo de caja proyectado.	217
A.2. Cálculo de VAN y TIR con una tasa mínima de 10%.	218

Capítulo 1

Introducción

El avance tecnológico ha crecido de manera exponencial a lo largo de estos años. Se ha buscado el desarrollo de nuevas tecnologías que permitan realizar distintos controles y gestionar sistemas de forma inalámbrica. Actualmente existen muchas disponibles para hacer esto posible, algunas con mayor impacto que otras debido a características propias, bajo costo, fácil configuración, bajo consumo y rápida compatibilidad con otros receptores y emisores de radiofrecuencia.

De forma paralela los sensores van acompañando la evolución tecnológica, la variedad presente en el mercado proporciona al usuario la capacidad de sensar todo tipo de variables físicas y señales de control creando un gran abanico de posibilidades para el desarrollo de diferentes aplicaciones, hemos llegado al punto en el cual hoy en día prácticamente vivimos rodeados de sensores. En los últimos años, mi trabajo me ha dado la posibilidad de trabajar en distintos sectores industriales involucradas en la distribución de energía eléctrica y gas natural. La telemetría se volvió un factor fundamental en las industrias gracias al control que brinda en tiempo real incluso en los lugares más remotos, logrando mejoras preventivas y datos mucho más precisos en la medición de variables a través de sensores de distinto tipo. Estas empresas utilizan diferentes tipos de PLC's y sofisticados Softwares para el monitoreo general mediante pantallas que interactúan con el usuario de un alto valor económico, por supuesto que disponen del caudal financiero necesario para ello, pero si se carece de ese nivel económico y se requiere la telemetría para la adquisición de datos de algún sistema ¿Qué podemos hacer?

Este proyecto se orienta a realizar una *Interfaz de Adquisición, Control y Automatización* inalámbrica de un costo mucho menor y de características similares siguiendo tres simples pasos:

- *adquisición*
- *envío de datos*
- *recepción para el usuario*

Mediante el uso de módulos de tecnología **ZigBee** se realiza la adquisición y envío de datos remotos, destacando la lectura de sensores con salida de corriente, salida de tensión, sensores digitales y activación remota de actuadores.

Una vez adquirida la información se mostrara al usuario utilizando un programa de computadora para realizar el correspondiente monitoreo. Se agregaran algunas características propias de los sistemas como envío de alarmas, envío de correo electrónico, gráficos y guardado de variables.

1.1. Motivaciones y Objetivos del Proyecto

1.1.1. Motivaciones

La motivación principal de este proyecto está en la posibilidad de crear un sistema propio de adquisición, control y automatización, realizar una integración entre las diferentes etapas que la componen. Este trabajo no está pensado solo para poder obtener un título de grado, sino para poder utilizar esta herramienta en mi ámbito laboral y personal.

Una de las principales razones para la elección del tema fue desarrollar algo dentro del entorno de electrónica industrial. Desde mis primeros empleos hasta hoy en día mis decisiones me han llevado al lugar correcto afirmando cada vez más mi deseo de querer trabajar en esta rama de la electrónica, estar en este campo durante estos años me ha impulsado no solo a terminar la carrera sino también a elegir un proyecto final acorde al tema que me apasiona.

1.2. Objetivos

El objetivo principal de este proyecto es integrar una solución telemétrica con adquisición de datos vía sensores y posibilidad de salida para actuadores. Dentro de la gama de sensores se espera poder adquirir datos de:

- *sensores de lazo de corriente 4-20 mA*
- *sensores analógicos 0 10 VDC máximo*
- *sensores digitales de 24 VDC*

Y se pretende un control de actuadores de *12VDC o 220VAC*.

La interfaz será diseñada para un propósito general y no para un tipo específico de sistema, dando la posibilidad de utilizarse en diferentes escenarios.

Se pretende aprender, investigar y adquirir experiencia en la programación gráfica para poder crear cualquier tipo de sistema a partir de este momento. Se creará un entorno universal, que luego podrá modificarse fácilmente para adaptarlo a las necesidades requeridas.

Debe ser escalable, es decir poder admitir más de un punto de envío para discriminar información enviada desde diferentes equipos al nodo de recepción.

Aumentar el conocimiento en cuanto al diseño profesional de Hardware, analizando distintas posibilidades y características deseadas, teniendo en cuenta qué componentes están disponibles en el mercado local y los costos que requieren.

Reforzar y aumentar los conocimientos en la herramienta de diseño electrónico para crear circuitos impresos, esquemático y board.

Se aplicarán los conceptos básicos de la electrónica a la hora de diseñar una aplicación:

- *selección y diseño de circuitos*
- *diseño de circuito impreso*
- *diseño de hardware*
- *diseño de software*

El proyecto debe servir para el ámbito profesional y como una herramienta didáctica para aquel que quiera aprender los conceptos que aquí se tratan y crear sus propios sistemas, que lo aprendido aquí quede para siempre como una herramienta más a utilizar.

Esta será la primer version de este proyecto, como principal objetivo debe cumplir con los puntos antes mencionados y luego se irá mejorando acorde a los primeros resultados obtenidos aquí.

Capítulo 2

Introducción a la comunicación inalámbrica

2.1. Introducción

Actualmente nos encontramos inmersos en la que se denomina revolución tecnológica de las comunicaciones inalámbricas, similar a las predecesoras como la electricidad, la televisión, el ordenador, etc. Esta se refiere al intercambio de información entre dispositivos mediante el espectro electromagnético, sin necesidad de utilizar conexiones físicas entre emisor y receptor. Muchas veces la posibilidad de generar cableados es limitada, por lo que este tipo de conexión nos ofrece la conectividad dentro de una zona geográfica específica y da la ventaja de conservar la estructura del lugar sin tener que realizar cambios significativos en la zona. El hecho de que el punto de entrada a una red no esté ligado a una ubicación fija y que el medio de transmisión ya este preparado favorece su expansión a comparación de otras tecnologías.

En la vida cotidiana estamos rodeados de ondas electromagnéticas, las cuales pueden clasificarse según el campo de aplicación:

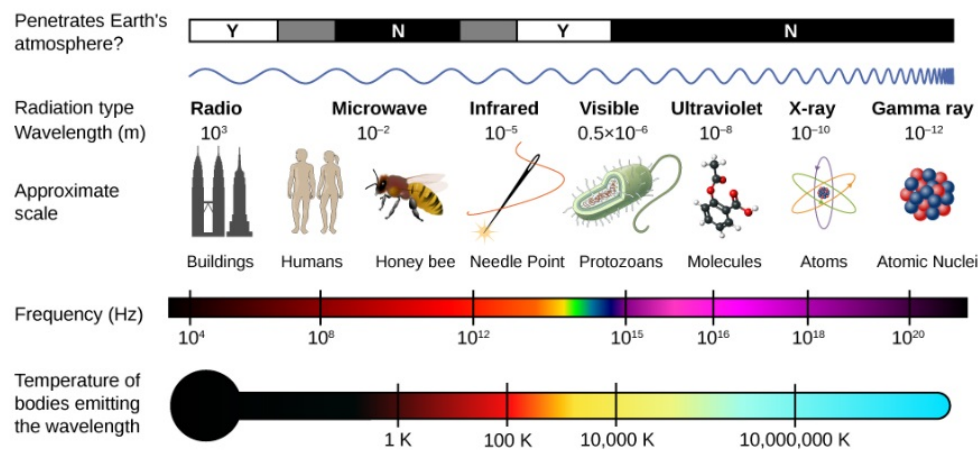


Figura 2.1: Espectro Electromagnético, figura extraida de [5].

En la Figura 2.1 se encuentran los distintos niveles en la escala de frecuencias, dentro de este amplio rango se encuentran aplicaciones típicas como la radio AM, FM, telefonía móvil, frecuencias audibles, aplicaciones digitales, colores y otras más complejas como las que utilizan rayos gamma. Si bien los campos son variados todas tienen como protagonista principal la utilización de ondas electromagnéticas. Estas no viajan simplemente de un punto físico hacia otro más lejano, dentro del medio de propagación al encontrarse frente a obstáculos se producen distintos fenómenos como los descritos a continuación:

- *Absorción*: a medida que la onda se aleja del punto donde fue emitida su intensidad disminuye debido a la disipación que sufre en el medio donde se propaga, reduciendo la energía considerablemente. La absorción que ocurre en un medio determinado puede variar según la frecuencia de la onda, esto quiere decir que para ciertas frecuencias las ondas se verán afectadas en mayor o menor medida por el medio. Para una onda sonora al encontrarse frente a una superficie, parte de su energía es reflejada y parte es absorbida por la superficie, la absorción dependerá del material. Este fenómeno es aprovechado por ejemplo en salas acústicas donde se busca que las paredes absorban los sonidos y no provoquen eco.
- *Reflexión*: cuando una onda viajera se encuentra con un medio reflectante, este

actúa como un espejo. La onda es reflejada en un ángulo perpendicular a la superficie. Este fenómeno es utilizado en transmisiones de onda corta, logrando un gran alcance ya que la onda rebota a distintas alturas (que dependen de la frecuencia) de la ionósfera logrando grandes distancias e incluso dando vuelta al planeta o en algo tan típico como un microondas, los que tienen ventana para ver el interior tienen una rejilla sobre el vidrio, esta tiene una longitud menor a la longitud de onda que emite el magnetrón por lo tanto son reflejadas hacia dentro calentando el contenido mientras que las ondas visibles pasan y nos permiten ver el interior.

- *Refracción:* la refracción ocurre cuando la onda incidente pasa por un medio no homogéneo provocando un cambio en su velocidad y dirección. Los medios de propagación llevan asociados los índices de refracción, este índice es el cociente entre la velocidad de la luz en el vacío y la velocidad de la onda en el medio, cuanto mayor sea la diferencia de índice entre los medios habrá una mayor refracción. Este fenómeno se aprecia al observar como los objetos sumergidos en agua sufren alteraciones en su aspecto.
- *Difracción:* cuando la onda incidente se encuentra con objetos cuyo tamaño es similar a su propia longitud de onda, este obstáculo ahora se convierte en un foco emisor. Su estudio se realiza a través del *Principio de Huygens*, donde cada frente de onda se comporta como una fuente de onditas secundarias pudiéndose obtener un nuevo frente de onda mediante las envolventes de estas ondas secundarias. Se observa el fenómeno al hablar con una persona con un obstáculo en medio, o en un cuarto oscuro cuando la luz entra por una pequeña rendija se puede ver un rayo de luz que ilumina parcialmente la zona.
- *Polarización:* esta es una característica de las ondas transversales cuyo campo eléctrico y magnético están orientados físicamente de forma perpendicular y estrechamente enlazados. La polarización hace referencia a la dirección del campo eléctrico, en la vertical lo hace perpendicularmente a la superficie terrestre, en la polarización horizontal oscila en un plano horizontal a la superficie, mientras que en la circular el campo gira alrededor de su eje de propagación. Esto es útil por ejemplo en telecomunicaciones, las antenas deben adecuarse a las posiciones correspondientes a la polarización de las ondas para hacer mas

eficaz la comunicación, un ejemplo mas cotidiano esta en los lentes para el sol, estos utilizan un filtro vertical que bloquea las ondas que vienen en todas direcciones provenientes del sol y solo deja pasar las que tienen dirección vertical, produciendo una imagen ligeramente mas oscura donde los objetos pueden distinguirse perfectamente.

- *Interferencias*: el fenómeno ocurre cuando dos o mas ondas se superponen en un mismo punto en el espacio, generando una nueva onda con mayor, menor o igual amplitud. Ocurre por la propagación de ondas en distintas trayectorias que llegan a una antena receptora por distintos caminos y tiempos. También puede utilizarse en beneficio, auriculares de una gran calidad toman sonidos del ambiente (frecuencias no deseadas) en ambas bobinas magnéticas, desfasan una de las dos señales 180 grados provocando la anulación del ruido, por supuesto esto es un simple ejemplo que conlleva procesos mucho mas delicados que los mencionados aquí como ejemplo.

Actualmente existen acuerdos internacionales para organizar el uso del espectro electromagnético debido a que es limitado y como pudo verse hay una gran variedad de servicios que lo utilizan.

Particularmente las ondas de radio son ideales cuando se habla de conexiones de tipo inalámbrica. La problemática que conlleva es que requieren de mucha potencia de transmisión en comparación de las conexiones que utilizan cables.

El motivo es que al emitir la señal de radio hacia el aire, gran parte de la energía irradiada se atenúa. Por lo tanto hay variables fundamentales que intervienen a la hora de realizar un desarrollo con este tipo de comunicaciones.

Cuando se realiza una transmisión, se entrega energía de RF (radiofrecuencia) a una carga (antena), generalmente expresada en dBm aunque también se puede encontrar en watts o miliwatts, esta señal emitida se ve afectada por los fenómenos descritos anteriormente.

El campo electromagnético tiene comportamientos distintos alrededor de la antena en función de la distancia, en campo se trabaja normalmente lejos de la antena cuando el frente de onda se ha establecido, esa zona es conocida como *campo lejano*, también existe una zona donde el campo no se ha establecido y no se comporta de la misma manera conocido como *campo cercano*. La frontera entre ambos está relacionado con

la apertura de la antena (D) y la longitud de onda (λ):

$$\frac{2D^2}{\lambda} \quad (2.1)$$

Suponiendo una antena isotrópica, es decir idealmente irradia energía en todas las direcciones por igual, la pérdida de energía desde emisor a receptor puede explicarse con la ecuación de transmisión en espacio libre.

$$\frac{P_r}{P_t} = G_r G_t \left(\frac{\lambda}{4\pi d} \right)^2 \quad (2.2)$$

La ecuación anterior nos dice que la relación entre la potencia transmitida y la recibida depende de la ganancia de ambas antenas, del cuadrado de la longitud de onda y de la inversa del cuadrado de la distancia entre ambas antenas. La ecuación de relación entre la potencia transmitida P_t y la recibida P_r depende de las ganancias de las antenas G_r y G_t , del cuadrado de la longitud de onda (λ) y como se ha mencionado, del cuadrado de la distancia. Al aumentarse este parámetro, la relación disminuirá. El término cuadrático $\left(\frac{\lambda}{4\pi d}\right)^2$ se utiliza para expresar el concepto de atenuación [6].

Sabemos que las antenas son componentes pasivos, por lo tanto no generan una ganancia como estamos acostumbrados en electrónica, sino que se refiere a la capacidad de concentrar la energía irradiada en un área del espacio.

Las redes ZigBee están diseñadas para que los dispositivos que utilizan esta tecnología no requieren baterías de alta capacidad para realizar transmisiones a larga distancia, sino que lo hacen utilizando "repetidores" ubicados a distancias más cortas para llegar a destino. Esto brinda la posibilidad de crear enlaces de larga distancia agregando nodos de repetición sin la necesidad de utilizar potencias tan elevadas.

2.2. Tipos de Comunicaciones

Dentro del mundo de la conectividad inalámbrica se pueden desglosar distintos tipos de redes, cada una de ellas conlleva particularidades diferentes de acuerdo al área donde el usuario se conecta a la red y al alcance que abarca (Figura 2.2): WPAN, WLAN, WWAN, WMAN.

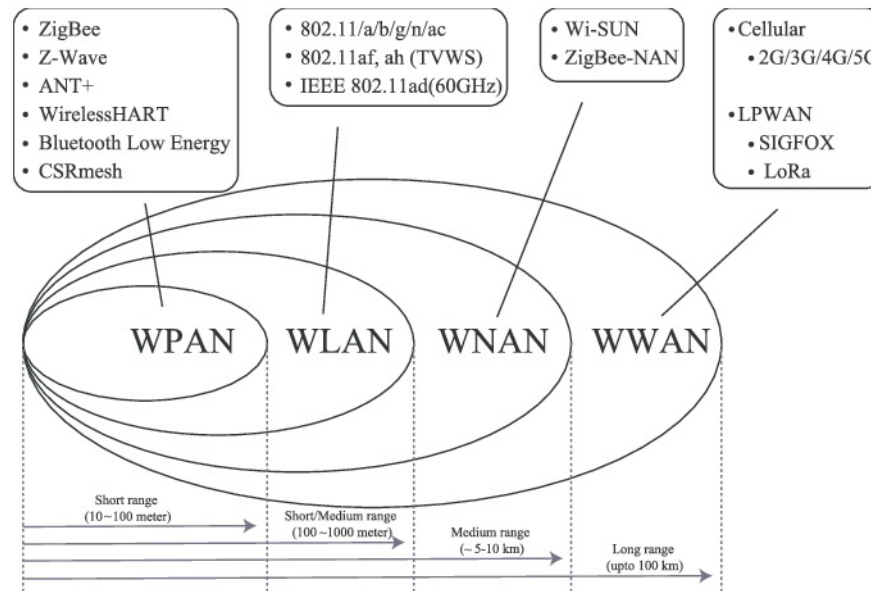


Figura 2.2: Tecnologías asociadas al tipo y alcance de red, imagen extraída de [16].

A continuación se mencionarán brevemente cada una de ellas:

- *Redes de Área Personal (WPAN):* Son redes privadas para ser utilizadas cerca del punto de acceso ya que funcionan a unos pocos metros y como su nombre lo indica suelen ser para uso personal. Dentro de esta categoría se encuentran tecnologías como Bluetooth y ZigBee.
- *Redes de Área Local (WLAN):* probablemente el tipo de red más conocida. Son privadas, generalmente ubicadas en edificios y lugares dentro de un área de cobertura de pocos kilómetros. Generalmente utilizadas en estaciones de trabajo para compartir información y recursos.
- *Redes de Área Extendida (WMAN):* aquí el área geográfica que abarca la conexión es muy extendida, por esta razón son utilizadas principalmente por teléfonos móviles, además permite que muchos usuarios estén conectados al mismo tiempo.
- *Redes de Cobertura Metropolitana (WWAN):* este tipo de red tiene una cobertura de distribución para un área geográfica alrededor de una ciudad, es decir, tiene un rango de alcance de decenas de kilómetros. Este tipo de red es útil para la distribución de internet, telefonía, televisión por cable, etc.

Dentro de las categorías mencionadas, se encuentra la tecnología seleccionada para este proyecto, **ZigBee**.

Este describe un protocolo inalámbrico normalizado para realizar una conexión de tipo WPAN. Diseñada para integrarse en diversas aplicaciones con una conectividad mucho más sofisticada que otros estándares inalámbricos existentes como WiFi y Bluetooth, dando una solución en lugares donde éstas no pueden cubrir las necesidades, con baja tasa de transmisión de datos, bajo ciclo de servicio de conectividad y bajo costo. La razón por la cual se promovió este nuevo estándar, fue para permitir la interoperabilidad entre dispositivos fabricados por distintas compañías, algo impensado para la tecnología en ese momento. Esto quiere decir que el sistema puede funcionar con otros sistemas existentes o futuros.

ZigBee, fue cobrando cada vez más fuerza a lo largo de los años debido a sus interesantes características:

- bajo consumo de energía
- dispositivos más ecológicos que sus predecesores, consumiendo mucha menos energía en el despliegue total de la red
- bajo costo de dispositivo, mantenimiento e instalación
- redes con una gran cantidad de nodos
- en caso de utilizar baterías, prolonga su vida útil
- implementación global, sus bandas operativas son aceptadas en una gran cantidad de países

¿Cuáles son los objetivos de ZigBee?

Principalmente apunta a la sencillez. Permite el diseño y desarrollo ágil al utilizar esta tecnología, su arquitectura debe ofrecer versiones actualizadas y desarrolladas para lo que viene, pensar en el futuro.

Agregar componentes de Hardware a un sistema existente debe ser fácil, teniendo en cuenta que ZigBee reemplaza el cableado el costo debe ser bajo ya que el cambio a tecnología inalámbrica debe implicar una ventaja, no una desventaja.

Capítulo 3

ZigBee

3.1. Introducción a ZigBee

Las tendencias indican que en un futuro cercano este tipo de conexiones abarcará todos los procesos, razón por la cual las grandes empresas a nivel mundial encargadas del desarrollo de dispositivos electrónicos crearon la **Alliance**.

Según la Alianza Zigbee, su tecnología es la base y el futuro de IoT. Establecida en 2002, con una amplia membresía global colabora para crear y desarrollar estándares abiertos universales para los productos que transforman la forma en que vivimos. Con la profunda y diversa experiencia de sus miembros, programas de certificación sólidos y un conjunto completo de soluciones abiertas de IoT, lidera el movimiento hacia un mundo más intuitivo, imaginativo y útil[3].

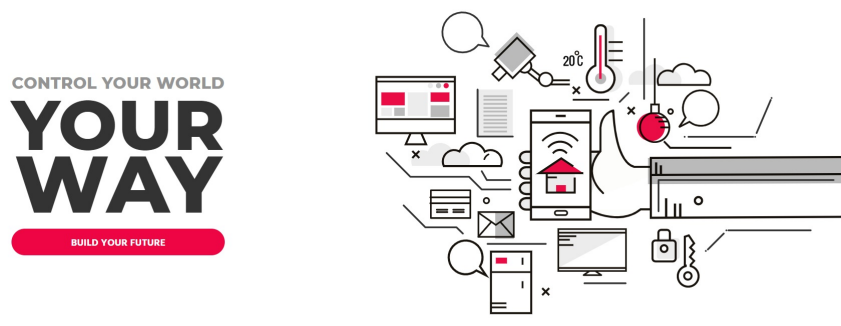


Figura 3.1: Imagen oficial de ZigBee Alliance haciendo referencia a la gran variedad de campos de aplicación [4].

Entrega beneficios como interoperabilidad de dispositivos estandarizados, actualizaciones constantes y características modernas e innovadoras. Concebida en 1998 cuando la tecnología Bluetooth y WiFi no podían presentar soluciones aptas para utilizar en todos los contextos, se puso en marcha el desarrollo de redes descentralizadas, es decir redes con varios nodos centrales ubicados en distintos puntos geográficos, funcionando como puntos de conexión y conectados entre si.

En 2004 se aprueba la especificación **ZigBee**, mismo año en el cual la *Alliance* anuncia la duplicación de sus miembros. Entre ellos se encuentran empresas reconocidas a nivel mundial:

- Amazon
- Apple
- COMCAST
- Google
- IKEA
- LUTRON
- NXP
- Silicon Labs
- Schneider Electric
- ST

- Texas Instruments

3.2. Estándar ZigBee

El estándar ZigBee es el nombre de un conjunto de protocolos de alto nivel orientada principalmente a la comunicación digital de bajo consumo, adopta al *Standard IEEE 802.15.4* para sus dos primeras capas, la física (PHY) y la subcapa de acceso al medio (MAC). Para comprender mejor como funciona este estándar, se observa el esquema de las capas del protocolo ZigBee (Figura 3.2):

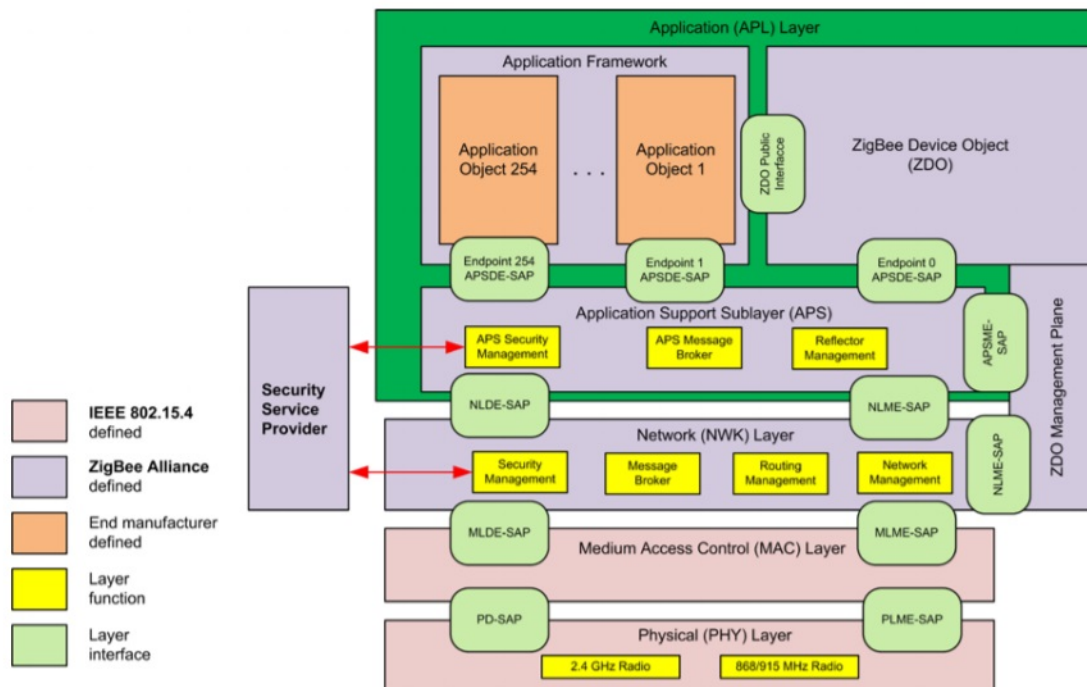


Figura 3.2: Capas de protocolo ZigBee[2].

En la Figura 3.2 se ven las capas del protocolo en relación al modelo ISO/OSI. Las 2 capas inferiores PHY y MAC son definidas por el *Standard IEEE 802.15.4* mientras que las capas de red (NWK) y de aplicación (APL) son definidas por ZigBee. Cada capa se conecta con la capa adyacentes por medio de un SAP (Service Access Point), esto hace referencia al pedido de servicios donde una capa superior

requiere un servicio de una capa inferior, las peticiones se realizan mediante un SAP. El *standard IEEE 802.15.4* define el nivel físico y de acceso al medio de redes inalámbricas de área personal (PAN) con baja tasa de transmisión de datos, proporciona enlaces de radiofrecuencia confiables con posibilidad de retransmisión, confirmación de recepción, mensaje broadcast y detección de errores. Veamos a continuación las características y operaciones que intervienen en las capas del estándar.

3.2.1. Capa Física (PHY)

Esta capa se encarga de la interacción con el medio físico para la transmisión y la recepción de datos en un canal de radio, acorde con las técnicas de modulación y spreading. Veamos las siguientes características de la capa:

■ Canales

La especificación incluye las siguientes bandas con sus respectivas velocidades (Figura 3.3):

- Global: 2.4 GHz a 250 Kbps (16 canales)
- Americanas: 915 MHz a 40 Kbps (10 canales)
- Europa: 868 MHz a 20 Kbps (1 canal)

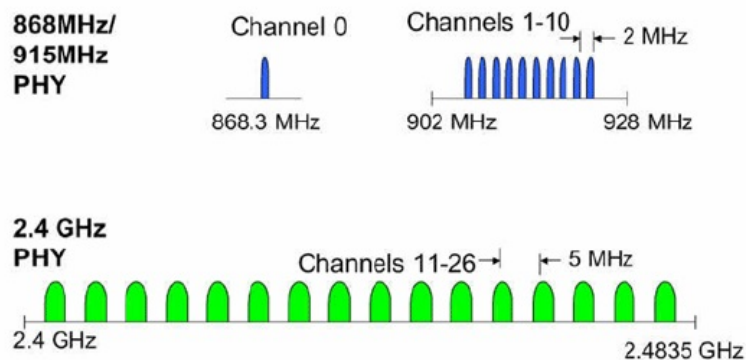


Figura 3.3: Estructura de canales[12].

Se puede ver en la Figura 3.3 que la mejor opción es 2.4 GHz con 16 canales (11 a 26) que van desde 2.4 GHz hasta 2.480 GHz con una separación de 5MHz por canal, tiene buenas características de propagación frente a los obstáculos aunque, por supuesto, un rango limitado. Esto parecería ser una desventaja, sin embargo la limitación produce una reducción de las interferencias. El rango puede expandirse simplemente agregando nodos en la red. Se modula a una velocidad de 62500 bauds (símbolos por segundo), cada símbolo se comprende de cuatro bits resultando una velocidad de transmisión de 250 Kbps.

Esta banda fue liberada por el ente regulador en la mayor parte del mundo, aunque no requiere licencias se limita la EIRP, energía efectiva irradiada por una antena isotrópica [6]. Este parámetro hace referencia a la suma de la potencia del transmisor y ganancia de la antena, menos las perdidas en el medio de propagación, con él los entes reguladores limitan la energía con la que pueden transmitir los equipos.

▪ Detección de Energía y Sincronización

La comunicación ocurrirá sobre un canal, cabe la posibilidad de que esté siendo utilizado por otros, por esta razón antes de comenzar el enlace se hace una evaluación del canal para comprobar si efectivamente puede realizarse la conexión inalámbrica por medio de los siguientes métodos:

- *Detección de Energía:* este procedimiento realiza una medición de energía sobre el canal para determinar el nivel con el cual se recibe la información (RSSI:Receive Signal Strenght Indication)[6].
- *Evaluación del Canal:* realiza la detección para comprobar si el canal se encuentra libre, puede hacerlo por el método de energía suficiente (pasado un umbral de energía, se considera canal ocupado), presencia de portadora (si hay presencia de portadora compatible con 802.15.4 se considera el canal ocupado) y presencia de portadora con nivel de energía suficiente (se considera ocupado si hay presencia de portadora con energía superior a cierto umbral).

Cuando se comprueba que el canal esta disponible para la transmisión, se sincronizan los dispositivos para generar el enlace. Para que el receptor detecte el inicio de una trama se realiza un preámbulo de treinta y dos ceros, seguido de la secuencia 10100111. Este ultimo indica que a continuación sigue un byte con longitud de trama. A medida que el lector avance en el texto, se mostrarán las estructuras de los frames y los detalles de sus campos.

SHR		PHR		PAYLOAD
PREAMBULO	SFD	LONGITUD	R	
32 ceros	10100111	7 bits		

Figura 3.4: Trama de sincronismo nivel físico.

Se ve en la Figura 3.4 los diferentes campos de la trama, SHR hace referencia al campo de encabezado para la sincronización con dicho preámbulo y secuencia de bits mientras que PHR es el encabezado de la PHY.

Luego de transmitir dicha trama, hay un silencio para que el transmisor se apague y se encienda el receptor, mientras que en el otro dispositivo sucede lo opuesto.

■ Modulación BPSK, QPSK y DSSS

Existen casos en donde no es posible transmitir la señal en banda base (que re presenta la información) debido a sus características de frecuencia y longitud de onda, por ejemplo, frecuencias del orden de los kHz tienen longitud de onda larga (de varios kilómetros), por lo cual requiere antenas de un tamaño considerable. Aquí es cuando la modulación tiene un papel fundamental, permitiéndonos realizar comunicaciones a través del medio de propagación aún en

condiciones adversas, por requerimientos de señal o infraestructura.

La técnica de modulación toma la señal en banda base y con ella modula otra señal conocida como portadora, esta señal modulada es la que efectivamente se transmite por el medio de propagación, en pocas palabras se cambian las características de la portadora de acuerdo a la señal en banda base (que tiene la información).

Dependiendo de la característica que se modifica en la portadora original se pueden clasificar distintos tipos de modulación:

- *Modulación Analógica:* dentro de esta categoría la portadora puede modularse en amplitud (AM), frecuencia(FM) o fase(PM).
- *Modulación Digital:* para el caso de modulación digital la portadora experimenta “saltos” dependiendo de la señal modulante. Puede realizar cambios respecto a la amplitud(ASK), frecuencia(FSK) o fase(PSK).

ZigBee utiliza modulación digital por desplazamiento de fase (PSK-Phase Shift Keying) con amplitud y frecuencia constante. El PSK es similar a la modulación en fase convencional, excepto que aquí la señal de entrada es una señal digital y cuenta con un número limitado de fases de salida. Se mencionarán los procesos, pero lejos está de entrar en los detalles más profundos.

Banda 868-928 MHz, BPSK

Para esta banda, la portadora experimenta “saltos” de fase dependiendo de los bits de información con la que se module. Por ejemplo, con una portadora cosenoidal, un bit 1 que la modula corresponderá a una fase de 0° por lo tanto la portadora no experimenta ningún cambio de fase, por el contrario cuando el bit modulante es 0, la portadora experimenta un cambio de fase de 180° (un “salto”), es decir la cosenoidal ahora estará invertida.

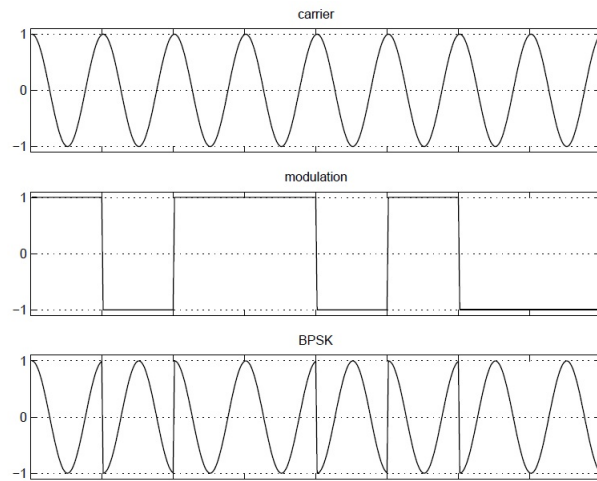


Figura 3.5: Modulación BPSK.

$$BPSK_{(t)} = \cos(2\pi f \cdot t + m_{(t)}) \quad (3.1)$$

De la ecuación anterior, f es la frecuencia de la señal portadora y $m_{(t)}$ es el valor de fase modulado por la señal digital, esta provocará el cambio en la portadora:

$$m_{(t)} = 1, BPSK_{(t)} = \cos(2\pi f \cdot t + 0) \quad (3.2)$$

$$m_{(t)} = 0, BPSK_{(t)} = \cos(2\pi f \cdot t + \pi) \quad (3.3)$$

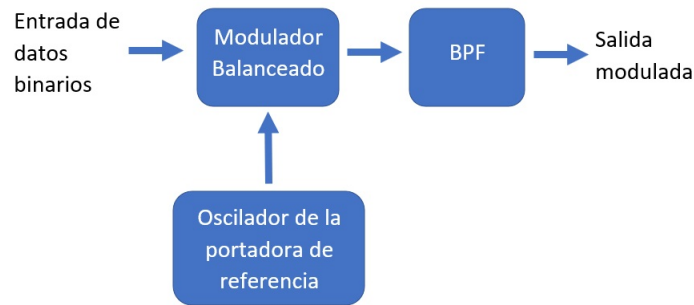


Figura 3.6: Diagrama de bloques simplificado, modulador BPSK.

La Figura 3.7 muestra un diagrama de bloques simplificado de un tipo de modulador BPSK, el modulador balanceado realiza la conmutación para invertir la fase, como se ha dicho dependiendo de la condición lógica de la entrada digital, la portadora es transferida a la salida con diferentes fases. Luego pasa por un filtro pasa banda que elimina los armónicos no significativos.

Por otro lado, la demodulación se realiza a través de un circuito de recuperación de portadora coherente. Este detecta y crea una portadora coherente en fase y frecuencia a la señal del transmisor.

La detección coherente se logra al multiplicar la señal modulada por una señal de la misma fase y frecuencia de la portadora, añadiendo un filtro pasa bajos es posible recuperar la señal original.

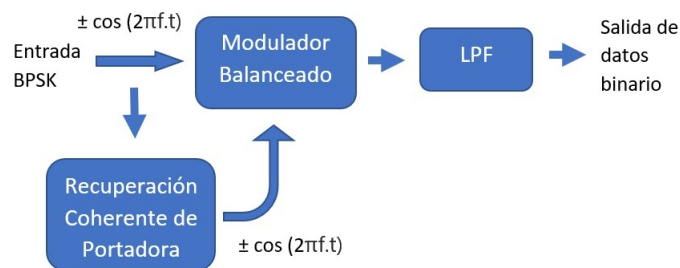


Figura 3.7: Diagrama de bloques simplificado, demodulador BPSK.

Banda 2.4 GHz, QPSK

Hasta el momento se ha mencionado el sistema binario BPSK donde sólo hay dos condiciones posibles de salida, uno para representar un 0 lógico y otro para 1 lógico.

Con frecuencia resulta una ventaja codificar a un nivel más alto que binario, en este caso se utilizan dos bits para referirse a cuatro fases de salida para una sola frecuencia de portadora resultado cuatro posibles condiciones: 00, 01, 10 y 11. Este tipo de modulación se conoce como QPSK, Q hace referencia a que las fases están en cuadratura (Quadrature) y son cuatro (Quadrature, Quaternary)[6].

Símbolo	Salto de fase
00	$+\frac{3\pi}{4}$
01	$-\frac{3\pi}{4}$
10	$+\frac{\pi}{4}$
11	$-\frac{\pi}{4}$

Cuadro 3.1: Relación entre símbolos y saltos de fase.

La tabla anterior muestra el mapa de los “saltos” que experimenta la portadora respecto a los dos bits de entrada:

$$QPSK_{(00)}(t) = \cos\left(2\pi f.t + \frac{3\pi}{4}\right) \quad (3.4)$$

$$QPSK_{(01)}(t) = \cos\left(2\pi f.t - \frac{3\pi}{4}\right) \quad (3.5)$$

$$QPSK_{(10)}(t) = \cos\left(2\pi f.t + \frac{\pi}{4}\right) \quad (3.6)$$

$$QPSK_{(11)}(t) = \cos\left(2\pi f.t - \frac{\pi}{4}\right) \quad (3.7)$$

Utilizando la identidad trigonométrica $\cos(a+b) = \cos(a).\cos(b) - \sin(a).\sin(b)$ es posible descomponer cada una de las ecuaciones:

$$QPSK_{(00)}(t) = -\frac{\sqrt{2}}{2}\cos(2\pi f.t) - \frac{\sqrt{2}}{2}\sen(2\pi f.t) \quad (3.8)$$

$$QPSK_{(01)}(t) = -\frac{\sqrt{2}}{2}\cos(2\pi f.t) + \frac{\sqrt{2}}{2}\sen(2\pi f.t) \quad (3.9)$$

$$QPSK_{(10)}(t) = +\frac{\sqrt{2}}{2}\cos(2\pi f.t) - \frac{\sqrt{2}}{2}\sen(2\pi f.t) \quad (3.10)$$

$$QPSK_{(11)}(t) = +\frac{\sqrt{2}}{2}\cos(2\pi f.t) + \frac{\sqrt{2}}{2}\sen(2\pi f.t) \quad (3.11)$$

Expresadas de esta manera se puede ver conceptualmente que QPSK es la suma de dos señales BPSK que se encuentran en fase y en cuadratura respectivamente. La siguiente figura muestra en los primeros tres gráficos las señales en fase y los siguientes tres en cuadratura para mostrar por último la señal QPSK resultante.

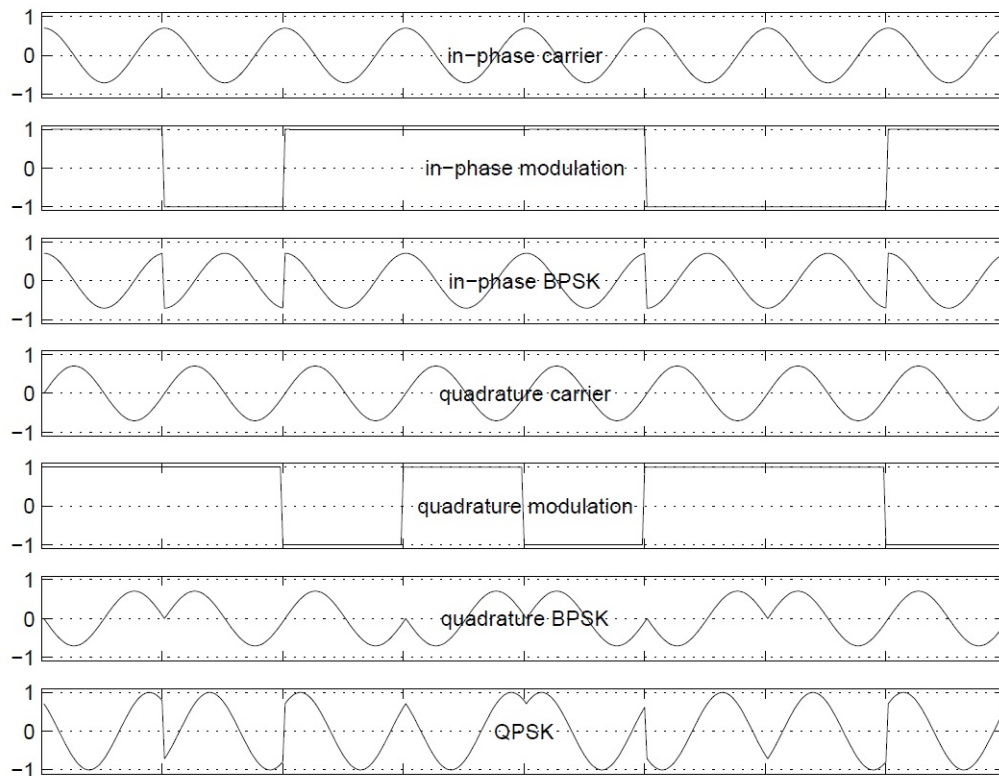


Figura 3.8: Modulación QPSK.

Este ejemplo corresponde a un caso común con una cuadratura de 90° cuyo punto de inicio de los “saltos” comienza en 45° lo que resulta en una constelación de la siguiente forma:

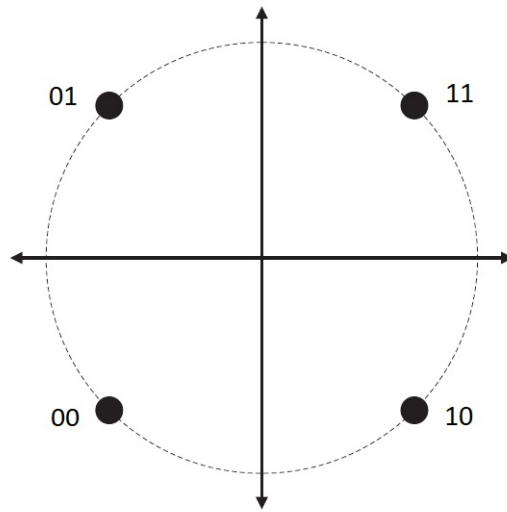


Figura 3.9: Constelación de modulación QPSK.

Como se ha mencionado, la señal QPSK resultante se forma con la adición de dos portadoras en fase y cuadratura desfasadas 90° entre sí, con ello, los bits de datos en la entrada del modulador se convierten a una forma paralela, dando origen a los canales I(In-Phase) y Q(Quadrature), cada grupo de bits modula a su correspondiente portadora, vemos a continuación el diagrama de bloques simplificado para ver como se genera la modulación:

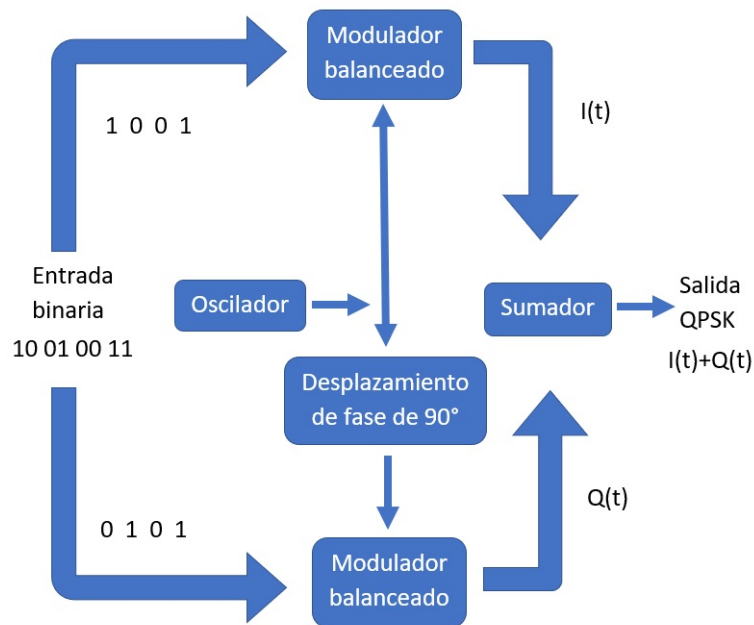


Figura 3.10: Diagrama de bloques, modulador QPSK.

El canal I modula con una portadora en fase con el oscilador local mientras que el canal Q lo hace con una portadora desfasada 90° (en cuadratura con la portadora de referencia) que luego se suman para generar la salida QPSK.

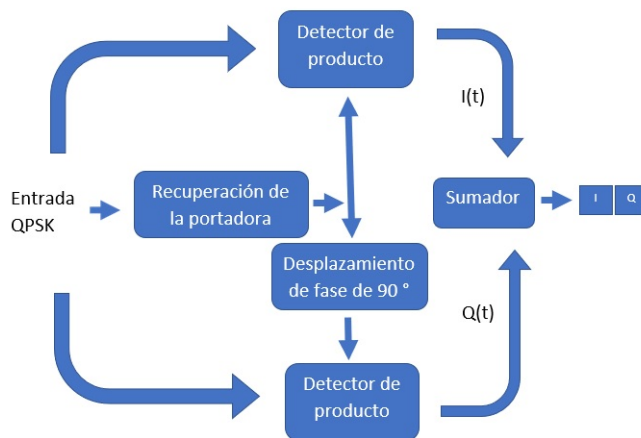


Figura 3.11: Diagrama de bloques, demodulador QPSK.

La Figura 3.11 muestra el diagrama de bloques simplificado del demodulador. Realiza una detección coherente, multiplica la señal modulada por la señal portadora original en fase y frecuencia, esta es captada por el circuito de detección coherente. La señal QPSK se heterodina con esta última para obtener los bits de información Q e I.

DSSS

Para poder mejorar la transmisión ante la posibilidad de interferencia en el medio, la *IEEE 802.15.4* especifica el uso de *Espectro Ensanchado de Secuencia Directa-DSSS-Direct Sequence Spread Spectrum*. Técnica creada para reducir la susceptibilidad de una comunicación a interferencias, para minimizar la posibilidad de generarlas o incluso para reducir la probabilidad de detección de estas comunicaciones[6]. En Ingeniería y Teoría de la Información, la capacidad de canal se refiere a la velocidad con la que los datos pueden ser transmitidos y se mide en bits por segundo, de esta teoría se establece para cualquier sistema de transmisión con un determinado ancho de banda y con una relación de señal ruido el límite de velocidad efectiva máxima en bps que se puede obtener sin importar la técnica de transmisión empleada. El *Teorema de Shannon-Hartley* establece que la capacidad del canal depende del ancho de banda B_w y de la relación señal ruido $\frac{S}{N}$ de la siguiente forma:

$$C = B_w \cdot \log_2\left(1 + \frac{S}{N}\right) \quad (3.12)$$

De la ecuación anterior se observa que para mejorar la capacidad debe aumentarse la relación señal ruido o lo que realiza esta técnica, aumentar el ancho de banda ensanchando el espectro.

El primer *spread spectrum* desarrollado se conoce como *Frequency Hopping Spread Spectrum (FHSS)*, aquí la técnica consiste en realizar cambios en la frecuencia de forma pseudoaleatoria, es decir aparentemente aleatoria. Cuando esto es empleado en una transmisión el receptor debe conocer estas frecuencias, de lo contrario no podrá recuperar la información del mensaje, es por eso que

los receptores a los cuales no se dirige la transmisión solo verán ruido. El ancho de banda se ensancha ya que depende de la cantidad de canales en los que ocurren estos cambios de frecuencia siempre y cuando los cambios de frecuencia son rápidos en comparación a la frecuencia con que se modula (bauds).

DSSS es diferente, en este caso la información se multiplica por una secuencia pseudoaleatoria haciendo la convolución de sus espectros, el receptor de destino deberá conocer esta secuencia para interpretar el mensaje y nuevamente como en el caso anterior los receptores a los que no se les destina el mensaje interpretarán la señal como ruido.

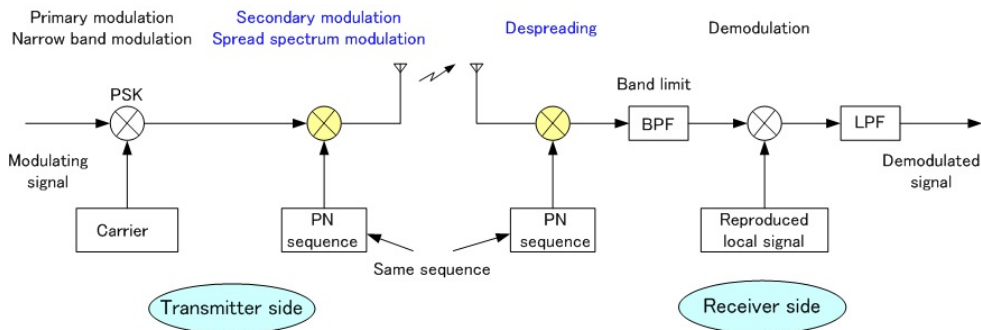


Figura 3.12: Etapas de modulación, transmisor y receptor[17].

Las secuencias pseudoaleatorias ortogonales se conocen como “Chips”, esta secuencia de código idealmente tiene una media nula y su autocorrelación es un impulso, estas características son propias del ruido, por esa razón a la secuencia se la llama ruido pseudoaleatorio (PN), típicamente son pulsos rectangulares de amplitud +1 y -1.

Cada símbolo de información es transmitido con una secuencia de N chips, este incremento en la velocidad de modulación produce una expansión de N veces el ancho de banda ocupado[6].

En las siguientes figuras se puede ver el proceso aplicado a una señal de entrada. Los receptores no autorizados obtendrán una señal que no podrán interpretar.

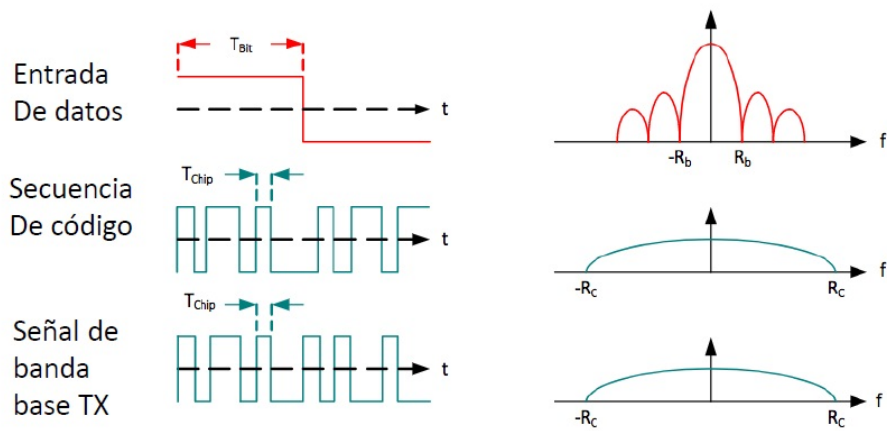


Figura 3.13: DSSS en el Transmisor[17].

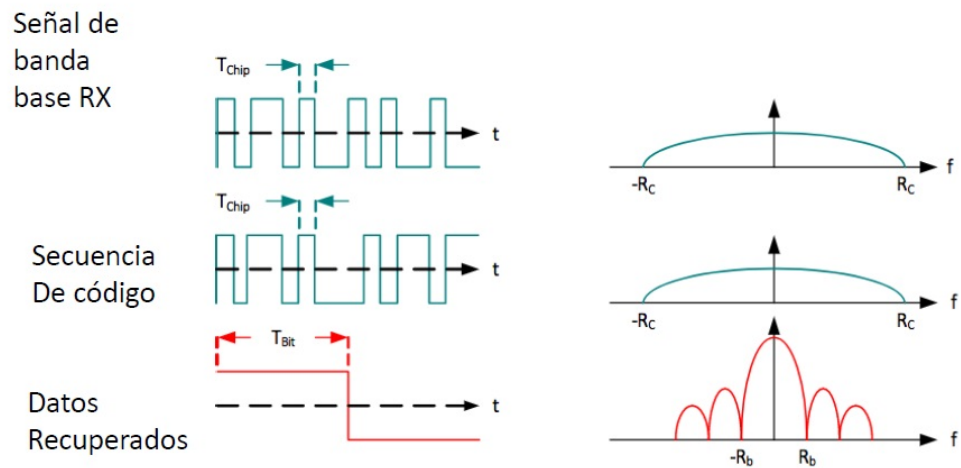


Figura 3.14: DSSS en el receptor[17].

El receptor realiza la correlación con la secuencia del transmisor para recuperar la información.

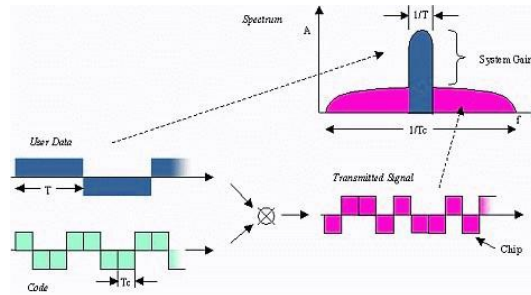


Figura 3.15: Secuencia de transmisión con DSSS[17].

Ensanchar el espectro tiene sus beneficios, hace que la señal aumente su resistencia al ruido e interferencia. Difundir la potencia transmitida sobre la banda mas ancha reduce la densidad espectral de potencia, ocultando la información que se transmite, además los transmisores pueden coexistir con otros tipos de tecnologías sin interferirse mutuamente.

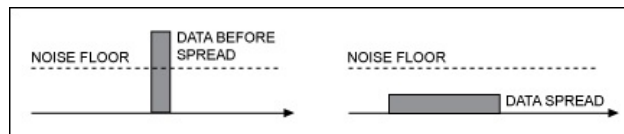


Figura 3.16: Barrera de espectro ensanchado bajo el nivel de ruido[17].

Como una desventaja el método presenta una dificultad en cuanto a su circuitería a la hora de implementarse, además del uso de un ancho de banda mayor al del mensaje original haciendo ineficiente este parámetro.

3.2.2. Capa de Control de Acceso al Medio (MAC)

Esta capa se encarga de la conexión de dispositivos en un mismo enlace físico, controla la información detectando posibles errores, se preocupa de la integridad de la misma pero no de su contenido. Hace posible que varios “interlocutores” puedan coexistir y compartir un medio de transmisión. Al igual que la capa física, es definida por el *standard 802.15.4*.

■ Estructura y Tipos de Tramas

La información se transmite en forma de tramas de la siguiente forma:

HEADER	PAYLOAD	FCS
3 a 23 bytes	variable	2 bytes
hasta 127 bytes		

Figura 3.17: Frame para el envío de información.

En el encabezado estarán los campos para el control y el FCS corresponde al campo de detección de errores.

Como ya se ha mencionado la comunicación entre dispositivos se realiza por un único canal, esto puede resultar caótico sin una organización ya que varios dispositivos pueden estar tratando de acceder al canal simultáneamente provocando lo que se conoce como “colisiones”. La capa física realiza un análisis del canal para saber si esta ocupado o libre, un dispositivo puede encontrarlo libre sin embargo otra entidad también puede estar a punto de enviar su propia información. Para evitar esto se utiliza un control de acceso al medio para organizar correctamente la gestión del canal.

Cada una de las tramas incluirá la dirección de destino y de origen, pueden ser de dos tipos:

- Dirección extendida: compuesta por 64 bits y única para cada dispositivo.
- Dirección corta: en este caso se compone de 16 bits y representa una dirección única pero dentro de la red, el coordinador es el encargado de asignar estas direcciones a los dispositivos que componen la red, por lo tanto, debe almacenar el listado de direccionamiento.

HEADER				PAYLOAD	FCS
FRAME CONTROL	SECUENCIA	DIRECCIONAMIENTO			
ACK		Fuente	PAN ID Y DIRECCION		
Modo direccion de destino		Destino	PAN ID Y DIRECCION		
Modo direccion de origen					

Figura 3.18: Trama con más detalle.

Dentro del encabezado, en el *frame control* para ser más específico, se encuentra el campo que indica cuál de los dos se está utilizando, si necesita confirmación de recepción y el tipo de trama que veremos a continuación.

Además de las direcciones anteriores también se utiliza una identificación llamada *PAN ID*, indica en que red se origina el mensaje y también es único para cada una de éstas. En caso de requerir enviar un mensaje de tipo *broadcast* para que llegue a todos los dispositivos, se utiliza la dirección corta y como destino un valor de 0xFFFF.

Además de incluir información en cuanto a las direcciones y ACK, el encabezado de trama indica de que tipo es dentro del frame control:

- *trama de datos*: como su nombre lo indica es utilizada para el tráfico de datos, lleva consigo la información.

HEADER				PAYLOAD	FCS
FRAME CONTROL	SECUENCIA	DIRECCIONAMIENTO			
ACK		Fuente	PAN ID Y DIRECCION		
Modo direccion de destino		Destino	PAN ID Y DIRECCION		
Modo direccion de origen					
Tipo de trama=DATOS					

Figura 3.19: El frame control incluido en el encabezado indica el tipo de trama de datos.

- *trama de comandos*: este tipo de trama es útil para gestionar parámetros de la red o configurar los dispositivos, el comando se envía dentro del PAYLOAD de la trama.

HEADER				PAYLOAD		FCS
FRAME CONTROL	SECUENCIA	DIRECCIONAMIENTO		COMANDO	PAYLOAD	
ACK		Fuente	PAN ID Y DIRECCION			
Modo direccion de destino		Destino	PAN ID Y DIRECCION			
Modo direccion de origen						
Tipo de trama=COMANDO						

Figura 3.20: El frame control incluido en el encabezado indica el tipo de trama de comando.

- *trama ACK*: se utiliza para confirmar la recepción cuando se lo solicita. No transporta información, solo indica el éxito de la recepción a quien lo solicita. No llevan información de direccionamiento es sus tramas.

HEADER			PAYLOAD	FCS
FRAME CONTROL	SECUENCIA	DIRECCIONAMIENTO		
Tipo de trama=ACK				

Figura 3.21: El frame control incluido en el encabezado indica el tipo de trama ACK.

- *trama Beacon*: esta trama le indica a los demás dispositivos la presencia del que la envía para que puedan asociarse a él. En la información de direccionamiento solo transporta la dirección de la fuente del mensaje.

HEADER				PAYLOAD	FCS
FRAME CONTROL	SECUENCIA	DIRECCIONAMIENTO			
ACK=0		Fuente	PAN ID Y DIRECCION		
Modo direccion de destino=0					
Modo direccion de origen					
Tipo de trama=BEACON					

Figura 3.22: El frame control incluido en el encabezado indica el tipo de trama ACK.

■ CSMA-CA

EL método de acceso al medio responde a sistemas CSMA(Carrier Sense Multiple Access), estos detectan la presencia de portadora en el medio y se hacen a un lado cuando otro esta utilizando el recurso. Dicho de una manera mas amigable consiste en “escuchar antes de hablar”, esto permite reducir las colisiones, pueden adoptar distintos comportamientos cuando detecta que el medio queda libre para ser utilizado:

- *l-persistent*: ocupa el medio de inmediato. En caso de que dos estaciones intenten hablar al mismo tiempo una vez listo el medio, se agrega un tiempo aleatorio *backoff* para evitar la colisión, este tiempo aumenta exponencialmente conforme se reiteren las colisiones.
- *p-persistent*: ocupa el medio a través de una probabilidad p , espera al slot siguiente $1-p$. Aunque no es inestable cuenta también con un tiempo de *backoff*.
- *nonpersistent*: espera un tiempo aleatorio o cantidad aleatoria de slots. Aunque no es inestable cuenta también con un tiempo de *backoff*, aunque este tipo además evalúa el canal antes de transmitir incluso habiendo finalizado el tiempo de *backoff*.

Para ZigBee el método de acceso al medio se conoce como Carrier Sense Multiple Access with Collision Avoidance(CSMA-CA) o acceso al medio por detección de portadora y prevención de colisiones, este protocolo de control intenta minimizar las colisiones cuando las comunicaciones tienen lugar en un mismo medio(canal) y deben realizarse de a una por vez. Existe una problemática en este tipo de red inalámbrica que tienen un alcance limitado, por lo que miembros de una misma red no se perciban entre si al estar en distintos lugares, si ambas estaciones no se escuchan pueden llegar a transmitir al mismo tiempo a otro que esta en medio de ellos por ejemplo, al superponerse nos exponemos a la perdida de información debido a la colisión, ambos dispositivos desconocen de este suceso por lo que no vuelven a intentar enviar los datos creyendo que el envío anterior fue exitoso, y así se pierde absolutamente todo. CSMA/CA

resuelve esto recurriendo a otro protocolo conocido: RTC(Request to Send) y CTS(Clear to Send). Este procedimiento tiene lugar previo al envío de datos utilizando ACK dentro de la MAC. Luego de que se realice la comprobación de disponibilidad de canal mediante los procesos de la capa PHY, si el dispositivo necesita enviar información puede hacerlo, aunque éste no sabe si otro dispositivo está a punto de hacer lo mismo, para ello el método establece que antes de utilizar el medio se espere un cierto tiempo pseudoaleatorio definido entre 0 y 2^T-1 (cada estación tendrá uno propio), donde T es conocido como exponente de backoff, cuyo valor comienza en macMinBE (IEEE 802.15.4 2003 indica por defecto 3) hasta el valor aMaxBE (IEEE 802.15.4 2003 indica como 5). Todos los dispositivos de la red tienen su propio tiempo, una vez finalizado si la comunicación no fue posible, los dispositivos incrementan T aumentando el tiempo de manera exponencial reduciendo la probabilidad de colisión. Si por algún motivo aun no es posible la comunicación luego de $\text{macMaxCSMABackoffs}$ (IEEE 802.15.4 2003 indica 4 intentos) se cancela el envío y se notifica internamente.

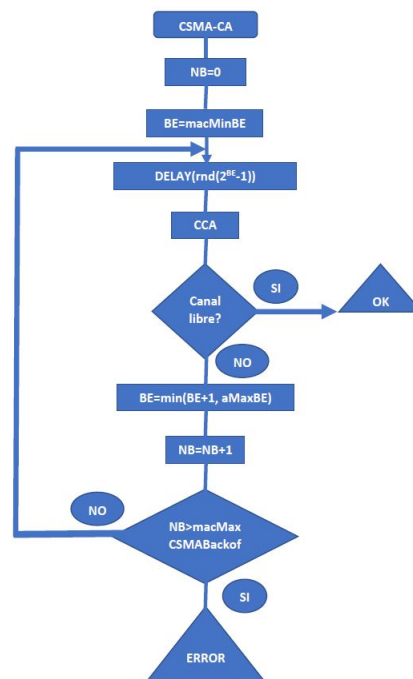


Figura 3.23: Diagrama de flujo del algoritmo CSMA-CA[6].

Si bien las colisiones son posibles, el método intenta agregar una aleatoriedad para minimizarlas, en algunos casos se asigna `macMinBE` en 0 para que una estación comience a transmitir inmediatamente al encontrar el canal libre ya que el término 2^T-1 se hace nulo.

■ Integridad de la Información

La capa MAC se encarga de la integridad y del control de la información, veamos algunos aspectos que ayudan en el control. La detección de errores se realiza mediante los campos de las tramas.

FCS (Frame Check Sequence) realiza el chequeo de redundancia CRC (Cyclic Redundancy Check) para realizar la detección. Cuando se envía una trama es posible exigir que el receptor informe que recibió el paquete de manera correcta solicitando en envío de la trama ACK antes mencionada, para realizar la petición debe indicarse con un flag dentro del campo header de la trama. Si la confirmación no es recibida se retransmitirá el mensaje hasta 3 veces. Además de CRC y la trama ACK, la capa gestiona el tiempo que existe entre trama y trama, debido a que los dispositivos deben procesar la información que le llega, el tiempo mínimo entre estas se conoce como IFS (Inter Frame Spacing) y está asociado al largo de las tramas, cuanto más grande sean estas últimas más largo será el tiempo entre las mismas.

■ Topologías

Para facilitar el desarrollo de una red, el *standard 802.15.4* define dos tipos de dispositivos los cuales presentan sus propias características funcionales:

- *Dispositivos de funcionalidad completa (FFD)*: también conocido como nodo activo, es capaz de recibir mensajes en formato 802.15.4 y utiliza de manera completa la capa MAC, dando la posibilidad de ser configurado como Router o Coordinador.
- *Dispositivos de funcionalidad reducida (RFD)*: también conocido como nodo pasivo, este tiene su funcionalidad y capacidad reducida, utiliza la capa

MAC de forma parcial y sólo funcionan en topología estrella. Conocidos como Dispositivos Finales o End Devices poseen características para lograr conseguir un bajo costo y simplicidad. Son los que adquieren el valor de los sensores en la red de este proyecto.

Los FFD pueden dialogar con nodos similares o con los RFD. Los RFD sólo podrán establecer comunicación con dispositivos FFD y no con los de su mismo tipo.

Se repasarán las topologías que puede adoptar las redes 802.15.4 y alguna de sus características. Dentro de éstas, se pueden diferenciar dos tipos, redes con o sin coordinador. Cuando no hay coordinador, se conocen como redes *AD HOC*, en este caso los dispositivos transmiten cuando lo desean. La comunicación ocurre por *CSMA-CA* y si se desea el receptor puede devolver una trama de reconocimiento (ACK).

El segundo tipo se refiere a las redes con coordinador, donde la topología es de tipo estrella. En el centro se ubica el coordinador y todos los demás dispositivos transmiten hacia él. Se encarga del sincronismo permitiendo a los dispositivos mas simples operar en bajo consumo. Dentro de esta categoría se desglosan las siguientes redes:

- *Redes sin beacon*: en este caso si bien existe un coordinador no envía *tramas beacon* indicando su presencia, sino que escucha a los demás en respuesta a tramas *Beacon Request*. Cuando un dispositivo quiere comunicarse con el coordinador, envía una trama de datos mediante *CSMA-CA* a lo cual el coordinador responde mediante un ACK, a este tipo se lo conoce como comunicación directa.

Cuando el coordinador es el que quiere comunicarse con el dispositivo final, espera a que este último le envíe una trama *data request*, el coordinador responderá con un ACK y mediante *CSMA-CA* enviará la trama de datos. Dicho esto podemos decir que el coordinador no podrá transmitir hacia el dispositivo final hasta que él lo contacte.

- *Redes con beacon*: el coordinador transmite periódicamente tramas *beacon* indicando el modo de operación y mantiene el sincronismo de la red.

■ Seguridad en 802.15.4

Como sabemos, la información se transmite por el medio de propagación libremente, por lo tanto cabe la posibilidad de ser interceptada, ante esto la seguridad es un aspecto muy importante en las conexiones inalámbricas.

Dentro del encabezado de trama se encuentra el flag que indica el transporte protegido de la información. Esto ayuda a que el receptor sepa si debe realizar algún tipo de proceso para revertir la protección de la información o no. El *standard* provee además los siguientes servicios destinados a la seguridad[20]:

- *Control de acceso*: los dispositivos deben contar con una lista de afiliados a la red y rechace transmisiones de fuentes desconocidas.
- *Integridad de tramas*: el servicio garantiza que los mensajes recibidos no han sido modificados, utiliza MIC (Message Integrity Code) utilizando un hash dentro del payload de la trama, estos hash se refieren a un algoritmo matemático utilizado para dicho propósito. De esta forma se asegura que los mensajes no fueron modificados por dispositivos no autorizados.
- *Cifrado de datos*: la información del payload es codificada mediante criptografía de clave simétrica, los dispositivos que estén enlazados deberán conocer la clave para codificar y decodificar la información, la seguridad depende de esta y no del algoritmo utilizado.
- *Frescura*: utilizando un contador en las tramas, el objetivo de este servicio es asegurar que el mensaje no fue repetido por dispositivos no autorizados. El receptor verifica este contador y rechaza tramas con un contador igual o menor a este.

3.2.3. Capa de Red (NWK)

La capa de red permite conectar dispositivos ubicados en distintos enlaces físicos, se encarga de gestionar las rutas desde emisor y recuperarse ante problemas de conectividad. Proporciona servicios a la APS y utiliza servicios de la capa MAC, hace de interfaz entre ambas.

■ Tipos de Nodos y Topologías

Dentro de estas categorías de dispositivos existen tres tipos de nodos que son los actores principales definidos por el *standard ZigBee*:

- *Coordinador*: tipo de dispositivo más completo. Toda red debe tener un único coordinador. Éste es el encargado del inicio de red, mide la energía presente en los canales y selecciona el que menos ruido tenga. Creada la red, éste dispositivo nunca estará inactivo.
- *Router*: proporciona la comunicación de dispositivos distribuidos en la red. Es capaz de unirse a una red creada y como su nombre lo indica rutea el tráfico entre los distintos nodos. Puede dar servicios a otros funcionando como interfaz para repetir lo que recibe. Puede funcionar como coordinador para un grupo reducido de RFD en una red 802.15.4. Utilizados como repetidor generan una extensión del alcance de transmisión y en caso de requerirlo realiza el descubrimiento de otros nodos para que la información llegue a su destino.
- *End Device*: dispositivo más limitado, puede mantenerse dormido de forma periódica en caso de requerirlo, se asocia a un router (con funcionalidad de coordinador) y le envía su información de forma periódica.

Para unirse a la red inicia el proceso de asociación con un coordinador 802.15.4 (router) o un coordinador ZigBee. Si el dispositivo es reiniciado ya no posee dicha asociación, por lo tanto el nodo al que fue vinculado, su “padre” (coordinador 802.15.4) deben contestar nuevamente la petición para que se vuelva a generar la conexión entre ellos. Por lo general este nodo permanece dormido (Modo Sleep) la mayor parte del tiempo cuando se utilizan baterías aumentando su ciclo de vida, cuando despierta avisa a su coordinador y éste si tiene un mensaje para él, se lo envía.

Conocidos los intérpretes, veamos ahora las topologías de una red ZigBee:

- *Estrella*: topología mas limitada y simple, consiste en un coordinador y dispositivos end devices. El nodo coordinador participa en todas las comunicaciones, de modo similar a una red 802.15.4 [6]. Esta topología puede

ser una desventaja cuando las rutas entre el coordinador y los dispositivos finales no esta disponible, estos últimos deberán declararse huérfanos y requerirán volver a la red buscando un nodo “padre”.

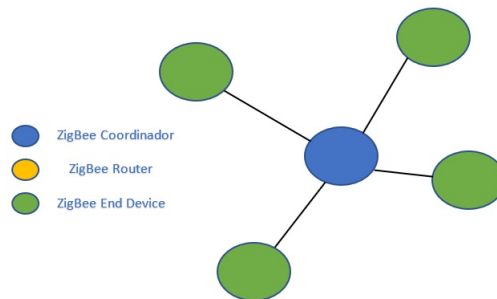


Figura 3.24: Topología estrella.

- *Árbol*: en este caso la red es formada por coordinador, end devices y routers. Estos últimos funcionan como una extensión para la cobertura de red, los end devices se conectan al “padre” (router o coordinador). ZigBee gestiona los nodos de forma automática, si se agrega o quita un nodo, la red se reconfigura nuevamente. Esta configuración tiene poco uso ya que es reemplazada por la forma de malla.

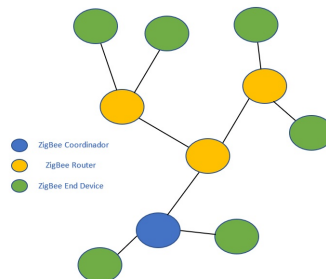


Figura 3.25: Topología árbol.

- *Malla*: este modo es el más fiable, el coordinador y los router se encargan de generar las rutas para el envío de información hacia el destino aún si la comunicación no es directa, el mensaje viaja a través de los routers por la red. Cuando una ruta no funciona, es detectada debido a la confirmación

de recepción propia de 802.15.4, de esta forma el router comienza un nuevo proceso de descubrimiento de ruta para volver a lograr la conexión en la red.

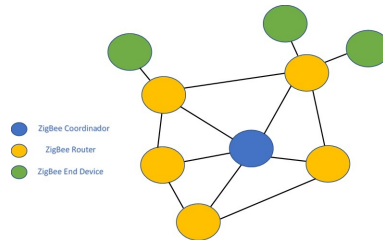


Figura 3.26: Topología malla.

■ Procesos de Red y Routing

En este apartado se verá el mecanismo de cómo son los procesos a la hora de iniciar una nueva red ZigBee.

Todo comienza en el Coordinador, este será el encargado de iniciar la red. Hemos visto que iniciar la red depende de la capa física del *standard 802.15.4*, toda la comunicación ocurre en un mismo canal, por lo cual el coordinador realiza la medición de cada uno de los canales para elegir el más adecuado. Una vez elegido el canal, comienza el “escaneo” mediante el envío de tramas *beacon request* y confecciona una lista de PAN ID’s disponibles o utiliza el valor impuesto por el usuario. A continuación, se asigna a si mismo la dirección de red 0x0000 y se queda a la espera[6].

Los dispositivos que requieran unirse a la red deben solicitarlo primero, el nodo que acepta esta petición pasa a ser el “padre” del solicitante “hijo”. Pueden adoptar distintos procesos dependiendo de la situación en la que se encuentren:

- *Asociación*: para asociarse el dispositivo realiza un descubrimiento de red enviando *tramas beacon request*, serán escuchadas por el coordinador o router que responden con *tramas beacon*. De esta manera el solicitante seleccionará al nodo que será su “padre”, a quien enviará la solicitud de asociación de red para que el coordinador conozca su dirección MAC y

que éste le asigne una dirección corta dentro de la red. En caso de que el proceso de asociación falle, intentará realizar el mismo procedimiento, pero con otro nodo “padre”.

- *Retorno a la red*: este proceso es similar al anterior en cuanto a las *tramas beacon*, sólo que la solicitud de asociación indica una reconexión de red, se utiliza cuando por algún motivo un dispositivo abandona la red.
- *Huérfano*: aquí el dispositivo envía una trama indicando que se encuentra en este estado (*orphan notification*). El nodo padre que lo tenía asociado responde indicándole su dirección corta de red.
- *Salir de la red*: cuando a un dispositivo se lo quita de la red, éste indica a los demás nodos vecinos que lo eliminen de sus tablas de direcciones.

Hasta aquí se mencionaron brevemente los procesos para el inicio de la red, veamos ahora el algoritmo de routing utilizado en ZigBee.

La capa de Red usa una versión modificada del protocolo *AODV (Ad hoc On-demand Distance Vector)* basado en el algoritmo *distance vector*. Este consiste en el envío de información acerca de las redes conocidas por parte de los nodos de forma periódica, de esta manera todos adquieren conocimiento acerca de la red que luego es distribuido nuevamente, por lo tanto habrá más de un camino para elegir, la elección dependerá de un parámetro llamado *costo*, es un algoritmo que permite generar el camino más corto en un gráfico[6]. Cada router tiene su costo calculado de forma individual y cuyo valor es transmitido a todos los demás nodos vecinos mediante mensajes de *link status*.

Un router posee internamente dos tablas que lo “ayudan” en la toma de decisiones al momento de realizar el routing. Una de ellas se llama *tabla de vecinos* y acumula las direcciones de los nodos que se encuentran a su alrededor en conexión directa, por otro lado la segunda tabla se conoce como *tabla de ruteo*, en este caso almacena direcciones de posibles caminos para extender la red, es decir llegar a los lugares más lejanos de la red. Cuando recibe un mensaje, estará la dirección de destino del mismo, a partir de allí enviará el paquete directamente si se trata de un nodo vecino o lo entregará a otro router que hará el mismo procedimiento hasta que el mensaje llegue a su destino final.

Cuando un dispositivo realiza un proceso de búsqueda (por desconocimiento

de la dirección de destino) la información se almacena en una tabla adicional llamada tabla de *rutas en proceso*. Si un nodo desaparece de la red, el nodo que detecta esta situación automáticamente informa al resto para que todos lo eliminen de sus tablas de ruteo. En ZigBee los nodos identifican a sus vecinos mediante mensajes llamados *hello*, cuando uno de ellos no conoce las rutas de destino para enviar información a un nodo más lejano (es decir, sin conexión directa) emite un mensaje de tipo *route request*, los vecinos intentarán ayudarlo brindándole la información necesaria. Si estos tampoco tienen información acerca del destinatario, el router inicia el proceso de *route discovery* y se realiza de manera conjunta entre los routers para encontrar un posible camino para entregar el mensaje. El proceso comienza con el mensaje de *route request*, el router al recibirlo verifica si el pedido lo involucra a él o a alguno de sus hijos. A partir de aquí pueden suceder distintos casos:

- si efectivamente el pedido los involucra directamente, contesta con un mensaje de *route reply*, este será enviado al dispositivo que hizo el pedido.
- Si no tiene información, crea una entrada nueva en su tabla de rutas en proceso y actualiza su propio costo, luego repite el pedido con una demora aleatoria para que todos los vecinos que reciben el *broadcast* no repliquen el pedido al mismo tiempo.
- Cada router que recibe el *route reply* compara la información con su propia tabla de rutas en proceso, los anuncios de costo actualizan la información y se pasan al siguiente nodo.
- Pasado un tiempo prudencial, si se recibió al menos un mensaje de *route reply* el router dispone de un intermediario inmediato para llegar a destino, el cual a su vez dispone de otro, y así sucesivamente[6]. La ruta ingresa en la tabla de ruteo de los dispositivos, pero carece de validación, ésta se realiza cuando se envía el primer mensaje (que generalmente lo hace el dispositivo que solicitó el pedido).
- Si un router no puede enviar mensajes de acuerdo a su tabla de ruteo, elimina la ruta y le avisa al resto de la situación por medio de un mensaje *network status*.

El proceso soporta rutas simétricas, es decir que el pedido de *route request* genera rutas idénticas en sentidos opuestos para descubrir la rutas de forma independiente y seleccionar la mejor acorde en relación al costo.

■ Frame y Funciones de la capa NWK

En NWK la información se transmite en forma de tramas con un encabezado que transporta los campos de control direccionamiento y ruteo.

En ocasiones se tiene un concentrador de información, es decir recibe mensajes de todos los puntos de la red. Supongamos que los dispositivos desconocen como enviar mensajes mas allá de su alcance, el concentrador debería contestar las peticiones de manera continua aumentando considerablemente la cantidad de tramas *route discoveries* y requeriría mas espacio en memoria para el almacenamiento en las tablas de ruteo. Para solucionar el caso se utiliza *source routing*, donde el concentrador especifica la ruta que deben tomar y así evitar que los routers deban averiguarlo. Otro problema es que los nodos finales no sepan llegar al concentrador, nuevamente habrá un incremento considerable de *route discoveries*, para solucionarlo los concentradores envían periódicamente su dirección para que todos puedan aprenderla.

Ya se han mencionado el problema con los mensajes broadcast, para evitar que un router vuelva a repetir una retransmisión, se agrega un contador para determinar la cantidad de veces que escucha una trama broadcast y así evitar las repeticiones de sus vecinos.

El al usuario tiene la posibilidad de trabajar en multicast, es decir enviar mensajes a un grupo selecto de dispositivos. Para ello emplea un campo especial dentro de la trama de red para el control de esta característica. Se ha dicho que la capa de NWK se relaciona directamente con la capa inferior definida por 802.15.4. Para minimizar el overhead la especificación utilizará direccionamiento corto en la capa definida por le IEEE. Toda la información se transmitirá sobre tramas de datos 802.15.4 y el protocolo NWK define direcciones de 16 bits, aunque puede requerir según sea el caso de las direcciones de 64 bits

propias de cada dispositivo, el encabezado de las tramas NWK consta de los campos que se ven en la siguiente figura:

HEADER								PAYLOAD
FRAME CONTROL	DEST ADDR	SRC ADDR	RADIO	SEQNO	IEEE DEST	MCAST	SOURCE ROUTING	
ACK								
DIRECCION IEEE								
SOURCE ROUTING								
MULTICAST								
Tipo de trama								

Figura 3.27: Trama NWK.

No solo especifica el direccionamiento, sino también tiene los campos de control de broadcast, control de multicast y source routing.

Este esquema general se aplica a las tramas que vimos en el apartado anterior en referencia a las trama de datos y de comando con la diferencia que el encabezado posee los nuevos campos mencionados.

3.2.4. Capa de aplicación (APL)

Esta capa es la encargada del transporte de datos de la aplicación y mantenimiento del flujo de datos.

APS (Application Support)

En la capa superior por encima de NWK se encuentra del protocolo APS, este provee circuitos virtuales para la comunicación entre dispositivos llamados *endpoints*. Así como existe una dirección única que identifica cada uno de los nodos, los *endpoints* identifican a la aplicación.

Cada aplicación utiliza distintos tipos de mensajes, cada uno de ellos tiene asociado un *Cluster ID* y al conjunto de mensajes de la aplicación se lo llama *profile ID*. Los perfiles son definidos por la *ZigBee Alliance*, el perfil público es para la interoperabilidad entre dispositivos similares mientras que a los privados sólo se le exige que

puedan coexistir con los demás.

Los dispositivos realizan las comunicaciones entre las aplicaciones por medio de los identificadores *endpoints*, pueden ser capaces de descubrir las aplicaciones y características de otros nodos mediante la aplicación *ZDO (ZigBee Device Objects)*, con ella se accede a estructuras que poseen dicha información, a éstas se las conoce como *descriptores*.

■ ZCL

Recordemos que los mensajes controlados por la *ZigBee Alliance* son identificados mediante *Clusters ID* y que cada aplicación utiliza un conjunto de estos. Permite que los nodos puedan utilizarlos e interpretarlos de la misma manera. Este grupo de mensajes identifica a cada aplicación, es a lo que llamamos *profile ID*.

Todos los mensajes de perfil público se encuentran dentro de la *ZCL (ZigBee Cluster Library)*, esta biblioteca fue creada con el fin de no reinventar todo ante cambios y que cada uno de los bloques constructivos puedan reutilizarse de forma ordenada.

No solo contiene los atributos asociados a los *Clusters* sino también los comandos para que realicen acciones. Los mensajes entre *Clusters* siguen un formato particular, puede haber hasta 65536 atributos y 256 comandos. El uso de esta biblioteca es obligatorio a la hora de crear perfiles públicos.

■ ZDO (ZigBee Device Object) y ZDP (ZigBee Device Profile)

Las aplicaciones del *endpoint 0* son reservadas para el ZDO, se ha dicho anteriormente la función que cumple y está formado por seis objetos, los dos primeros estarán presentes obligadamente mientras que el resto puede o no estar implementado en un perfil. Los objetos son los siguientes:

- *Devices and Service Discovery*: permite resolver direcciones NWK e IEEE[6], hallar dispositivos con capacidades puntuales leyendo sus descriptores.

- *Network Manager*: punto de acceso para selección de canal, puede cambiar a otro si lo desea de acuerdo a los reportes de interferencia.
- *Node Manager*: punto de acceso a las funciones de routing.
- *Binding Manager*: realiza tareas de binding, el termino binding se refiere al proceso de descubrimiento y asociación mediante el dialogo de los ZDO de los dispositivos.
- *Group Manager*: control de grupos de aplicaciones.

El ZDO posee al igual que los demás un perfil reservado con un conjunto de clusters, a estos se los llama ZDP (ZigBee Devide Profile), este perfil pertenece a ZigBee y está presente en todos los dispositivos.

■ Descriptores

Los descriptores son estructuras que contienen las características del nodo, los descriptores definidos son los siguientes[6]:

- *Descriptor del nodo*: identifica el tipo de nodo y sus funcionalidades.
- *Descriptor de la energía del nodo*: identifica la fuente de alimentación y su estado de carga.
- *Descriptor simple*: describe los *endpoints* activos y el *perfil ID*.
- *Descriptor complejo*: muestra con mas detalle las aplicaciones en el nodo.
- *Descriptor del usuario*: información que puede proveerse por el usuario.

■ Seguridad en ZigBee

La seguridad en ZigBee se aplica tanto en las transmisiones como en la asociación de nuevos dispositivos a la red. Utiliza criptografía CCM*, una versión modificada de CCM utilizado en 802.15.4. Este es un algoritmo de cifrado que brinda autenticación y confidencialidad mediante *AES-128, Advanced Encryption Standard con clave de 128 bits*.

La especificación ZigBee utiliza dos claves:

- *Clave de red*: utilizada para proteger el tráfico dentro de la red, todos los que la componen deben conocerla.
- *Clave de enlace*: utilizada para proteger el flujo de datos entre las aplicaciones.

Los dispositivos pueden obtener las claves de diferentes maneras, de manera previa es decir ya estaban en los dispositivos previamente, mediante transporte donde la clave viaja a través de la red (de forma protegida) o estableciéndola entre partes, generándola a partir de una clave que ambas partes comparten llamada clave maestra. También puede estar pre-instalada u obtenerse mediante transporte.

Dentro de la red es necesario un dispositivo seguro, esto hace referencia a la confianza que los demás dispositivos tienen en él. ZigBee establece este rol del *Centro de Seguridad*, por lo general es el coordinador donde todos los demás depositan su confianza. Éste será el encargado de distribuir las claves de seguridad.

Autenticación de Seguridad

Cuando un dispositivo se une a la red, lo hace escuchando *tramas beacon*, en respuesta envía el mensaje de *Association Request* y luego aguarda la respuesta de su petición para saber si efectivamente es aceptado o rechazado a unirse.

Realizada la asociación, el interesado se encuentra en el estado “asociado sin autenticar” [6], es decir aún no están completos los requerimientos para ser miembro de la red, debe pasar por un proceso extra de autenticación y dependerá de qué modo de seguridad esté presente al momento, existen dos:

- *Standard security mode*: este modo se utiliza en instalaciones residenciales donde la información no es crítica. El *Centro de Seguridad* distribuye las claves de red. En este modo la seguridad es provista por la capa NWK, mediante un campo de control ubicado en el encabezado de la trama que indica si la información transportada es protegida.

- *High security mode*: este modo se orienta a utilizarse en actividades comerciales, los dispositivos se comunican utilizando clave de red o de enlace. El *Centro de Seguridad* mantiene un listado de estas claves para distribuir las donde la clave de red es periódicamente cambiada. Para este modo además de la seguridad provista por la capa NWK, también la provee la capa APS similarmente mediante en campo del encabezado.

Veamos ahora los procesos de autenticación que completan el proceso para unirse a la red para ambos modos de seguridad:

- *Autenticación en Standard security mode*: al asociarse es posible que los dispositivos tengan predefinida la clave de red. El router que interviene en el primer enlace envía la información acerca de este nuevo miembro al *Centro de Seguridad*, este proporciona la clave de red en caso de que este nuevo miembro no la tenga o envía una clave nula si ya estaba predefinida. La información es enviada en forma de comando cifrado por la clave de red.
- *Autenticación en High security mode*: en este caso el nuevo dispositivo puede tener la clave de enlace pre-instalada o compartir una clave maestra con el *Centro de Seguridad*, nuevamente el router asociado envía la información del nuevo nodo a éste. Aquí podemos encontrar tres posibles escenarios:
 - *Clave de enlace pre-instalada*: el *Centro de Seguridad* envía la clave de red.
 - *Clave maestra pre-instalada*: el *Centro de Seguridad* en conjunto con el nuevo dispositivo el proceso para establece la clave de enlace.
 - *Sin claves pre-instaladas*: el *Centro de Seguridad* envía la clave maestra al router en una trama cifrada con la clave de enlace. Este último envía al nuevo miembro la clave de enlace sin cifrado.

Finalmente el proceso termina con la autenticación entre el router y el nuevo nodo el *protocolo de autenticación mutua*. Luego el nuevo dispositivo es considerado parte de la red, en caso de querer comunicarse con los demás nodos deberá pasar por el protocolo nuevamente.

■ Trama APS

La información es transmitida en tramas las cuales son transportadas a su vez por las tramas de datos NWK.

HEADER								PAYLOAD
FRAME CONTROL	DEST EP	Grp addr	Cluster ID	Profile ID	SRC ep	APS ctr	Ext.Header	
ACK	DIRECCIONAMIENTO							
Tipo de trama								
Seguridad								
Ext. Header								
Formato								
Delivery mode								

Figura 3.28: Trama APS.

El chequeo de integridad de la información no es realizado ya que es tarea para los niveles inferiores. El encabezado incluye flags presente en el campo de frame control, incluye las solicitudes de confirmación de recepción (ACK), el campo delivery mode se refiere al tipo de direccionamiento de entrega(unicast, multicast, broadcast o indirecto).

El campo Group address indica el número para un determinado grupo de dispositivos en multicast. Para evitar recibir tramas duplicadas se añade el campo APS counter para identificar todos los mensajes entre endpoints.

Veamos ahora los tres tipos de tramas en APS, similares a las vistas anteriormente:

- *Trama de datos*: llevan consigo los datos de la aplicación.



Figura 3.29: Trama de datos APS.

- *Trama de comando*: sólo transportan comandos relacionados a la seguridad.

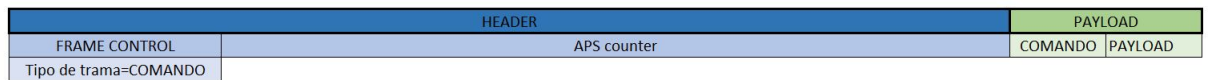


Figura 3.30: Trama de comandos APS.

- *Trama ACK*: respuestas de acuse de recibo. El campo de formato indica si el direccionamiento se incluye en la trama o no.

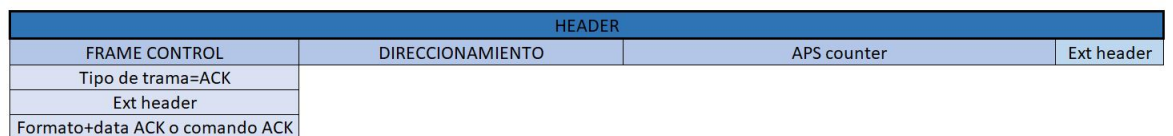


Figura 3.31: Trama de ACK de APS.

Dinámica de Comunicación

En este punto se han destacado todos los aspectos que abarcan cada una de las capas del *standard ZigBee*, si bien puede resultar algo abstracto a la hora

de leer de que trata, veamos ahora como interactúan las distintas capas de forma más amplia para poder comprender mejor el proceso de comunicación. Las tramas APS son transportadas por tramas de datos NWK que a su vez son transportadas por tramas de datos 802.15.4. La siguiente imagen describe lo anterior:



Figura 3.32: Relación entre tramas 802.15.4, NWK y APS.

En la parte inferior se encuentra remarcado de color negro la trama de datos 802.15.4 que transporta una trama de datos NWK que a su vez transporta la trama APS, veamos sus respectivos campos:

- **HEADER:** dentro de este campo se encuentra la trama 802.15.4 vista en capítulos anteriores rodeada de color rojo para poder identificarla, con los campos FRAME CONTROL, SECUENCIA para la sincronización e información de DIRECCIONAMIENTO.
- **PAYLOAD(NWK):** en este caso dentro se encuentra la trama de datos NWK remarcada en azul con su HEADER y su información de PAYLOAD(APS).

De esta manera se realiza la interacción entre las tramas. Veamos ahora el siguiente ejemplo, se encuentran cuatro dispositivos denominados 1, 2, 3 y 4. El objetivo de la aplicación 1 es comunicarse con la aplicación del nodo 4. La capa APS se encarga de la interacción entre aplicaciones, no sabe cómo enviar un mensaje hacia otro dispositivo, de eso se encargará la capa NWK, estas descubrirán las rutas a tomar, paso a paso hasta llegar al destino final del mensaje.

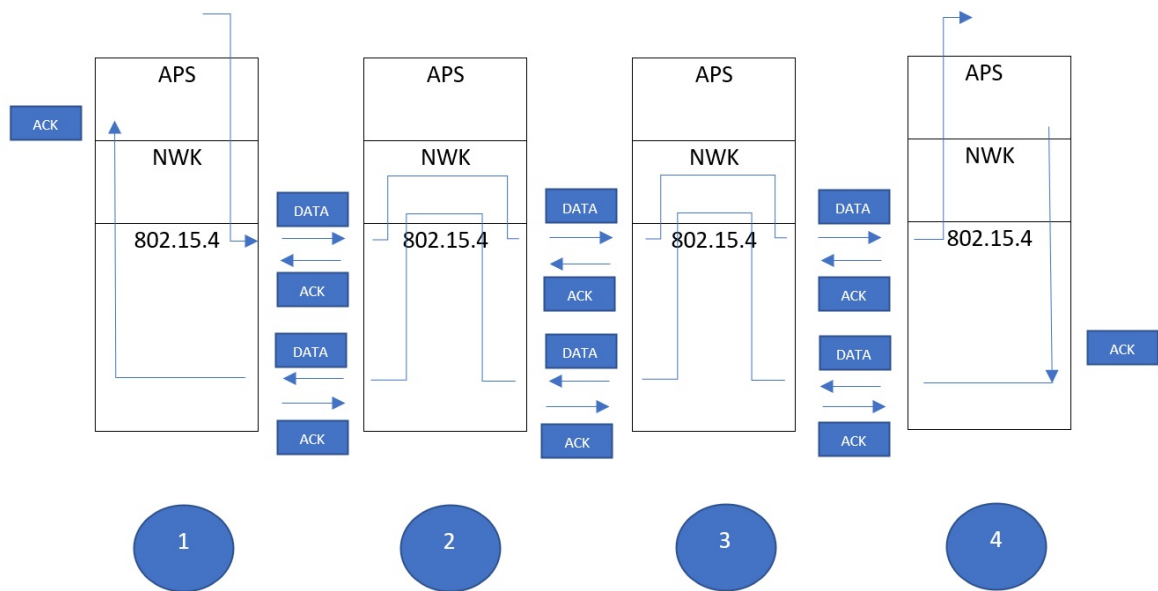


Figura 3.33: Interacción entre dispositivos.

La Figura 3.33 muestra un esquema simplificado del proceso de comunicación entre las capas del estándar. El endpoint 1 entrega la información a la capa APS del nodo 1, esta dialoga con la capa NWK para enviar la información al nodo 4. Se observa que para llegar a destino, el mensaje es transportado por los dispositivos 2 y 3 hasta el 4.

La interacción entre los nodos se realiza mediante tramas 802.15.4, asumimos que cada uno conoce a su vecino, el nodo 1 sabe el camino y entrega el mensaje al nodo 2 que en caso de recibir correctamente el mensaje responde con un ACK.

Ahora la información se encuentra en el nodo 2, el proceso vuelve a ocurrir de la misma forma, consulta sus tablas de ruteo en su NWK y por medio de otra trama 802.15.4 envía el mensaje al nodo 3, que a su vez, consulta sus tablas y lo hace llegar hacia el destino final, el nodo 4. A medida que llega la información a los nodos, envían las tramas ACK indicando el éxito de la recepción en cada una de las etapas.

Cuando el mensaje llega a destino final, el nodo 4 despacha la información a

la capa APS para que lo reciba el endpoint correspondiente. Una vez procesada toda la correspondencia, el proceso se invierte, el nodo 4 ahora envía la recepción hacia el nodo 1, el proceso nuevamente se repite de forma inversa mediante el dialogo de las NWK por medio de las tramas 802.15.4.

Los esquemas de tramas y el gráfico de interacción entre capas fueron consultados en la bibliografía[6].

3.3. Comparativa ZigBee con otras tecnologías

¿Por qué utilizar ZigBee? Existen diversas tecnologías disponibles en el mercado para utilizar en el proyecto, sin embargo se realizará una breve comparativa con Bluetooth de bajo consumo.

Cada uno es mejor que el otro en ciertos aspectos. En la actualidad la evolución de Bluetooth fue escalando hasta el punto que hoy es un competidor directo de ZigBee en cuanto al consumo energético y la creación de redes de área personal, aunque ambos tienen más diferencias que similitudes.

Al comienzo Bluetooth fue diseñado para el intercambio de datos a corta distancia, por eso está presente en una gran cantidad de dispositivos portátiles como un teléfono celular o unos simples parlantes. Sin embargo, necesita una fuente energética alta y esa era la principal desventaja comparado con el consumo de ZigBee. Hoy en día su desarrollo nos lleva a la una nueva versión llamada BLE (Bluetooth Low Energy) cuya tecnología trabaja en la banda de 2.4 GHz (al igual que su predecesor), los dispositivos que operen con esta especificación permanecerán en modo bajo consumo constantemente a menos que haya una conexión, por lo que resulta ideal para dispositivos IoT o M2M. Realiza conexiones de un bajo ancho de banda en distancias cortas. Implementa una arquitectura cliente/servidor permitiendo al hardware implementar las funciones de comunicación ahorrando dinero batería y ancho de banda. Este tipo de redes pueden contener una gran cantidad de dispositivos, aunque la distancia y el espacio físico limitan el tamaño de la red.

Los dispositivos no tienen que estar enlazados, pueden intercambiar datos de forma asincrónica en el momento que desea. Este protocolo es ampliamente aceptado e implementado en equipos OEM y millones de dispositivos. Como desventaja trabaja

punto a punto, eso quiere decir que los radios no podrán comunicarse mas allá de su rango individual, esto limita la distancia operativa de la red aproximadamente a diez metros quitándolo de posibles escenarios de aplicación que requieran una distancia mayor. El direccionamiento IP no está implementado lo que obliga a utilizar puertas de enlace para subir datos a internet.

Muchos teléfonos inteligentes llevan esta tecnología y son usados como las puertas de enlace, sin embargo, cuando no hay un dispositivo de este tipo no funciona. En aplicaciones comerciales e industriales no tiene tanto respaldo.

Para el caso de ZigBee hemos dicho que tiene un costo bajo, pero en relación a BLE puede extender la comunicación en un rango mucho más elevado. Tenemos la posibilidad de utilizar distintas topologías según sea el caso, en particular las redes mesh proporcionan un gran alcance y se reconfigurán dinámicamente en caso de que los nodos sean deshabilitados o eliminados. Los routers y coordinador deben estar siempre activos, eso debe tenerse en cuenta si se trabaja con baterías, los dispositivos finales serán los únicos que podrán estar en modo de bajo consumo. Tiene una mayor interoperabilidad entre distintos fabricantes y son ampliamente utilizados en domótica, IoT y redes industriales. Como hemos visto el direccionamiento, seguridad, reintentos y reconocimientos entre dispositivos están integrados dentro de los protocolos que integran el estándar, ahorrando un tiempo considerable de ingeniería. Cuanta con una amplia documentación y cubre mas casos de aplicación que Bluetooth.

Capítulo 4

Dispositivos XBee

4.1. Tipos

Los XBee son dispositivos de radio para comunicarse de forma inalámbrica. Pueden realizar tareas sencillas como simplemente reemplazar los cables en una conexión o tareas más complejas que involucren el uso de sus entradas y salidas digitales o analógicas. XBee es una implementación de Digi basada en el protocolo *ZigBee*. En estos últimos años se han presentado distintos modelos de esta tecnología:

- *XBee Serie 1 (XBee 802.15.4)*: primer módulo lanzado al mercado, ya vienen listos para trabajar con ellos y no necesitan ser configurados. Se utilizan en redes punto a punto o estrella. Consumen 45 mA al transmitir y aproximadamente 10 μ A en estado inactivo.

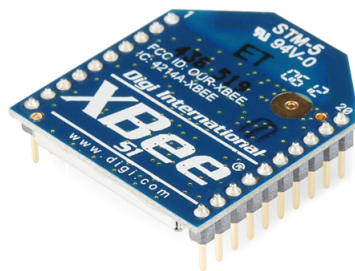


Figura 4.1: XBee Serie 1[1].

- *XBee Znet 2.5*: obsoleto, pueden funcionar en modo transparente o en modo API, pero depende del firmware con la que se configure los módulos.
- *XBee Serie 2*: esta es la serie elegida para realizar este proyecto ya que se adaptan a las necesidades requeridas del mismo. Es el mismo módulo mencionado antes pero con un nuevo firmware que permite utilizar además de 802.15.4, ZigBee. La potencia de su transmisor es mayor (+3dBm) al de la serie 1 logrando un mayor alcance con una sensibilidad de recepción de -95 dBm, su consumo también es menor utilizando 40 mA para transmisión/recepción y 1 μ A en estado inactivo.

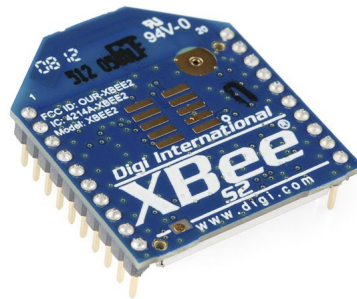


Figura 4.2: XBee Serie 2[1].

- *XBee PRO serie 1 y 2*: hay pocas diferencias entre el XBee regular y los PRO, y está en su hardware. El modelo PRO es un poco más grande, logra un mayor alcance que sus predecesores debido al aumento en la potencia del transmisor (+10dBm), a cambio de un consumo mayor de potencia, mientras que la sensibilidad de la recepción acusa en este caso -102 dBm, la corriente al transmitir es de 295 mA y para recepción 45 mA, mientras que en modo inactivo las especificaciones indican que es menor a 10 uA. Estos modelos son compatibles con los anteriormente mencionados.

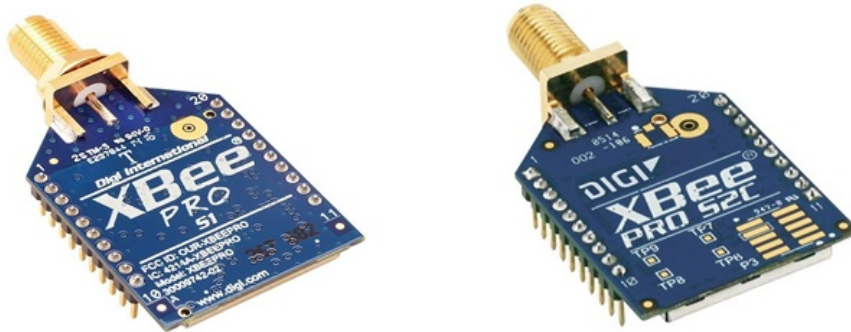


Figura 4.3: XBee PRO Serie 1 y Serie 2[1].

- *XBee Serie 3*: al igual que el predecesor permite crear redes punto a punto, punto a multipunto, broadcast y malla. Su firmware puede configurarse para utilizar distintas características como BLE, ZigBEE, DigiMesh y 802.15.4. Cuenta con un firmware con soporte de *micropython*.



Figura 4.4: XBee PRO Serie 3[1].

XBee presenta distintas antenas según el modelo:

- antena chip integrada en el módulo
- antena de cuarto de onda
- conector U.FL
- conector RPSMA

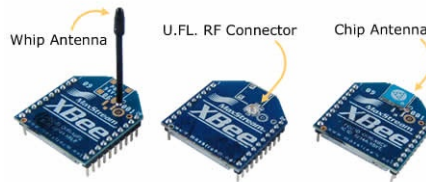


Figura 4.5: Tipo de antenas.

4.2. Características Generales XBee Serie 2

Este módulo es el que mejor se adapta al propósito de este proyecto. Cuenta con las características adecuadas para realizar mediciones tanto analógicas como digitales directamente en sus pines eliminando la necesidad de utilizar un microcontrolador, esto acota mucho el tiempo de desarrollo. Por supuesto que en caso de requerirlo puede adaptarse al proyecto fácilmente ya que los XBee cuentan con un puerto serie de comunicación para poder realizar una interacción con el micro deseado enviando la información adquirida por medio del módulo RF. Los pines pueden ser configurados de distinta manera, no todos contarán con las mismas funcionalidades pero presentan varias alternativas. Por último y más importante, el módulo trabaja con el estándar ZigBee visto el capítulo anterior.

El módulo Serie 2 presenta una potencia de salida de 2 mW (+3 dBm) con una sensibilidad en el receptor de -95 dBm. La tensión de operación es de 3.3 V y la corriente de operación en transmisión como en recepción es de aproximadamente 40 mA. Para el modo bajo consumo acusa una corriente menor a 1 μ A.

4.2.1. Modos de Operación

Los modos de operación, se refieren a los diferentes estados en los que el módulo trabaja. Adoptará alguno de ellos dependiendo la situación en la que se encuentre:

- *Modo Idle*: el XBee estará en este modo de reposo cuando no se encuentre recibiendo ni transmitiendo datos hacia otros nodos.

- *Modo de Transmisión:* cuando los datos llegan al puerto serie listos para enviarse, el módulo pasa de estado Idle a modo de Transmisión, listo para enviar un paquete hacia otro nodo. Previamente garantiza que se ha establecido una ruta confiable para el envío. En caso de no reconocer el destino en sus tablas de ruteo, realiza alguno de los procesos vistos en el capítulo anterior para encontrar otro camino.

Si luego del reconocimiento no se ha encontrado una nueva dirección para transportar el paquete, se descarta y los datos serán enviados una vez que se restablezca un camino posible para ellos.

- *Modo de Recepción:* Si el paquete recibido es válido, el módulo enviará los datos por el buffer del puerto serie.
- *Modo Comando:* será necesario para leer la configuración interna del XBee y para modificar parámetros. Las directivas se ingresarán vía puerto serie donde los comandos serán interpretados. Cuando este modo es activado, se inicia el temporizador del *Modo Comando (CT Command)* que lo habilita a recibir comandos AT en su pin de recepción *DIN*.

Para iniciar la comunicación, la secuencia comienza con el típico mensaje de comandos, *+++*. Si la conexión es exitosa se recibirá un *OK*.

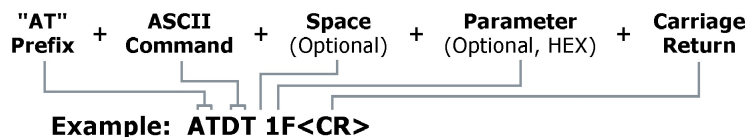


Figura 4.6: Estructura de envío de Comando AT[11].

Se observa la estructura de envío de comando, comenzando por el prefijo *AT*, seguido por el comando escrito es ASCII. Para finalizar se envía un *CR*, carriage return.

Cuando el XBee interpreta el comando de forma correcta, siempre devolverá como respuesta un *OK*, de lo contrario enviará el mensaje *ERROR*. La configuración se irá guardando en su respectivo registro y si se desean mantener los cambios debe enviarse un comando para escribirlos o modificar un bit en la

estructura API(se verá luego con más detalle).

Una vez realizado los cambios se debe salir de este modo, aplicando el comando *ATCN-Exit Command Mode* seguido de un *CR*. Otra manera es transcurrido el tiempo fijado mediante el parámetro CT (Tiempo de Espera de Modo Comando), cuando el tiempo de recepción de comando establecido se termina y no se recibe ningún comando válido el módulo automáticamente pasa a estado *Idle*.

- Modo Sleep:

Los modos de Sleep permiten pasar a un estado de muy poco consumo cuando se encuentra inactivo, puede ser configurado mediante uno de sus pines, o de forma cíclica de tiempo fijo. Muy útil para no gastar energía innecesariamente. ZigBee tiene entre sus características principales el bajo consumo energético, los módulos pueden ponerse temporalmente por si mismos en modo *sleep* y realizar transmisiones de forma controlada, alargando la vida útil de las baterías. Durante este modo el dispositivo estará *dormido* con un consumo muy pequeño y no podrá recibir o enviar datos hasta que *despierte* del modo reposo. Los routers y el coordinador participan activamente en el intercambio de paquetes, por lo que no pueden estar inactivos, los modos de bajo consumo son aplicables en los end devices. Los nodos “padre” en caso de querer comunicarse con los nodos dormidos, deberán esperar a que estos estén activos y envíen un mensaje de *data request* solicitando cualquier mensaje pendiente si es que los hay.

Los comandos *SM (Sleep Mode)*, *ST(Time Before Sleep)* y *SP(Cyclic Sleep Period)* son los utilizados para configurar los distintos modos. XBee cuenta con 4 comportamientos básicos de “sueño”, se pueden distinguir en dos categorías; modo controlado por pin y modo de sueño por medio de ciclos, aunque el módulo viene configurado por defecto con el modo de suspensión desactivado, es decir, la radio siempre estará despierta.

- *Modo de suspensión hibernate*: configurando el parámetro SM=1 el ingreso a bajo consumo es controlado por el pin DTR, el módulo se suspenderá hasta ser despertado operando sobre el pin mencionado, podrá estar operativo en un tiempo aproximado a 13.2 ms.

- *Modo de suspensión doze*: configurando el parámetro SM=2 el ingreso a bajo consumo es controlado por el pin DTR, el módulo en este caso dormita hasta ser despertado por el pin mencionado, se encontrará operativo aproximadamente en 2 ms. Este modo consume más corriente que el modo anterior.
- *Modo de suspensión cyclic sleep*: no depende del hardware sino de la configuración interna del XBee. Es el método más común para el ahorro de energía, dentro de esta categoría se encuentran dos posibilidades; modificando el parámetro SM=4 el dispositivo duerme durante un tiempo, determinado por el parámetro SP (periodo de sueño cíclico). Al despertar stará en modo operativo aproximadamente en 2 ms, se mantendrá despierto un cierto tiempo determinado por el parámetro ST.

La segunda forma corresponde a la configuración del parámetro SM=5, en este caso el módulo duerme un tiempo indicado por el parámetro SP a menos que se active el pin DTR. Al despertar interroga si hay correspondencia pendiente para él.

4.2.2. Comunicación con el Usuario

Los dispositivos utilizan dos formas diferentes para comunicarse con los usuarios: modo transparente y API. La elección depende del tipo de proyecto y su complejidad.

Modo Transparente

Este modo es utilizado principalmente en la conexión punto a punto o punto a multipunto, donde no es necesario ningún tipo de control. Es la configuración más sencilla, los módulos actúan en reemplazo de una conexión serie.



Figura 4.7: Comunicación Transparente entre módulos XBee.

Para lograr la comunicación entre dos módulos, el que envía los datos necesita la dirección de red del destinatario. Los dispositivos son capaces de almacenar esta dirección de 64 bits a través de dos parámetros: *DH* (*Dirección de destino alta*) Y *DL* (*Dirección de destino baja*). Para realizar una comunicación, deben fijarse estos dos parámetros con el valor de *SH* (*serial number high*) y *SL* (*serial number low*) del otro dispositivo.

A continuación, se observa un ejemplo de como se deben configurar.



Figura 4.8: Ejemplo de configuración para envío y recepción transparente[11].

Como se ha dicho, este modo de comunicación es el más básico y presenta dificultades a la hora de aumentar la complejidad del proyecto.

Cuando el número de nodos aumenta, se deben configurar las direcciones de red y MAC a cada uno de ellos cada vez que se requiere cambiar el nodo de destino para enviar datos.

Realizar cambios en la configuración implica escapar al modo comando para realizarlos por medio de comandos AT.

Como ventaja esta configuración es básica para comenzar a crear una red e intercambiar datos entre dos módulos.

Modo API

Este es el modo de comunicación más complejo. Permite la transmisión en forma de *tramas* (*frames*), con esto el flujo de datos que entra y sale por la interfaz serie XBee es estructurado y organizado para cada tipo de trama, que especifica que operaciones o datos transporta. De esta forma todos los datos recibidos o enviados

a través del puerto son frames API. Pueden tener datos o comandos AT, los datos se transportan a dispositivos remotos mientras que los comandos pueden tener dos alternativas de destino, hacia un nodo remoto para configurar sus parámetros o para configurar el propio XBee local. Esta interfaz se emplea en redes donde interactúan múltiples nodos permitiendo el acceso a cada uno de estos. Permite la configuración remota, la lectura y escritura de los pines I/O. Establece una comunicación compleja sin la necesidad de definir un protocolo, estas son algunas de las ventajas respecto a los modos mencionados anteriormente:

- identificación del origen y destino de cada frame
- posibilidad de enviar datos y comandos
- permite configuración remota de dispositivos unidos a la red
- permite recibir la confirmación de cada paquete transmitido.

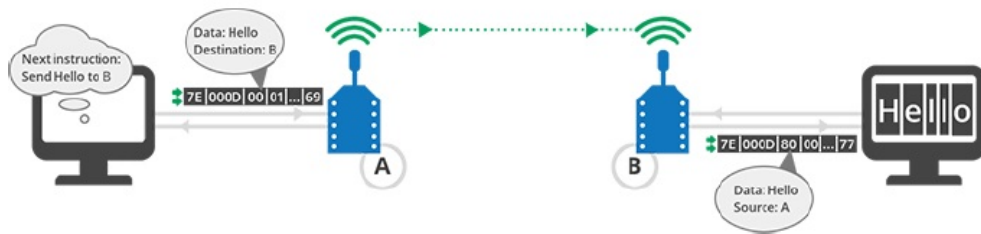


Figura 4.9: Comunicación API entre módulos XBee[9].

Gracias a los beneficios que ofrece la configuración API es la que se utilizara aquí. En una red da una ventaja en el manejo de la información y configuración remota. Se describirá en otro capítulo con más detalle el funcionamiento de este modo debido a su gran importancia dentro de este proyecto.

Capítulo 5

Modo API

5.1. Estructura de frame API

Las tramas API deben respetar una serie de especificaciones y estructuras determinadas. Como se mencionó anteriormente toda la información es empaquetada en forma de frames, con un orden y características específicas, los ejemplos para mostrar cada campo fueron tomados del reporte técnico[11]:

Start delimiter	Length		Frame data								Checksum
1	2	3	4	5	6	7	8	9	...	n	n+1
0x7E	MSB	LSB	API-specific structure								Single byte

Figura 5.1: Estructura de trama modo API[11].

La importancia que conlleva este modo de comunicación esta en los distintos frames con los que trabajan los XBee, mediante ellos se podrá configurar los módulos, consultar estados, configurar la red de trabajo y realizar peticiones de I/O específicas. En este proyecto se utilizarán dos tipos específicos, *0x17* (*Envío de Comando Remoto*), *0x97* (*Respuesta de Comando AT Remoto*) y *0x92* (*Recepción IO Remota*). Mediante ellos se configurarán los dispositivos remotos y se recibirán los valores analógicos y digitales provenientes de los End Device hacia el coordinador.

Los distintos frames hacen del modo API versátil a la hora de diseñar una aplicación con esta tecnología, veamos a continuación como se compone una trama de este tipo:

- *Start Delimiter*: el byte de inicio indicara el comienzo de cada trama, su valor es representado en hexadecimal con un valor de *0x7E*.
- *Lenght*: compuesto de dos bytes. Indica el tamaño de la trama luego del byte de inicio, nos facilita la cantidad de bytes que deben leerse a continuación.
- *Frame Data*: dentro de este espacio estará contenida la información acerca del tipo de trama y los datos que corresponden a cada una de ellas.
- *Checksum*: este byte chequea que los datos hayan sido enviados o recibidos correctamente mediante una operación aritmética. Se calcula como 0xFF menos la suma módulo 256 de todos los bytes del campo Frame Data. Para verificar se suman los bytes de Frame Data más el byte de Checksum cuyo resultado debe ser 0xFF.

Los frames abren un panorama mucho mas amplio, permitiendo al desarrollador contar con herramientas ya estructuradas. A continuación, veamos alguno de los tipos de trama:

API FRAMES	VALOR
Modem Status	0x8A
AT Command	0x08
AT Command-Queue Value	0x09
AT Command Response	0x88
Remote Command Request	0x17
Remote Command Response	0x97
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
ZigBee Transmit Request	0x8B
ZigBee Receive Packet	0x90

Cuadro 5.1: API Frames.

5.2. Comandos y Frames

Los comandos permite a los usuarios modificar configuraciones y realizar acciones sobre los módulos dependiendo el caso, repasemos algunos de ellos[10]:

■ Comandos de Configuración de Red

COMANDO	DESCRIPCIÓN
ND	Identifica todos los nodos de la red mediante broadcasts. El usuario envía este comando y los nodos contestarán a medida que la petición se extiende por la red.
NI	Almacena la cadena de identificación de cada modulo, definido por el usuario o una descripción del dispositivo.
CH	Lee el número de canal utilizado en la comunicación.
SH	Lee los 32 bits superiores de la dirección única de 64 bits de cada módulo.
SL	Lee los 32 bits inferiores de la dirección única de 64 bits de cada módulo.
VR	Interroga la versión de firmware de cada módulo.
AI	indica si un nodo está asociado o no a la red.
OP	Devuelve ID de la red a la que se enlace el módulo.

Cuadro 5.2: Comandos de configuración de red.

Comando de Datos

COMANDO	DESCRIPCIÓN
IS	Al enviar este comando de forma remota, fuerza una lectura de todas las entradas analógicas y digitales activadas, el nodo que recibe la petición envía la información por medio del frame de respuesta a comando remoto 0x97.

IR[x]	Este comando es configurado para realizar una lectura periódica de todas las entradas analógicas y digitales. El periodo estará expresado en milisegundos y seteada de forma hexadecimal. Se transmitirán las tramas cada vez que se cumpla con el tiempo establecido hacia el nodo padre en caso de un End Device, o en el mismo momento en que se realiza el muestreo para un Router.
IC[x]	Al configurar los pines del módulo remoto, mediante este comando es posible detectar cambios sobre los mismos. Cuando esto ocurre transmite los cambios a su nodo padre.

Cuadro 5.3: Comandos de Datos.

■ Comandos de Configuración de Pin

Cada comando tiene asociado un pin del módulo y cada uno de ellos tiene distintas funcionalidades.

PIN	CÓDIGO	MODO
4	P2[x] DIO-12	0:Digital Input no monitorizada-3:Digital Input monitorizada-4:Digital Output LOW-5:Digital Output HIGH.
6	P0[x] DIO-10	0:Desactivado-1:RSSI PWM-3:Digital Input monitorizada-4:Digital Output LOW-5:Digital Output HIGH.
7	P1[x] DIO-11	0:Digital Input no monitorizada-3:Digital Input monitorizada-4:Digital Output LOW-5:Digital Output HIGH.
9	D8[x] DIO-8	0:Desactivado-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH.

11	D4[x] DIO-4	0:Desactivado-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH.
12	D7[x] DIO-7	0:Desactivado-1:Control de flujo CTS-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH.
15	D5[x] DIO-5	0:Desactivado-1:Led de asociación-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH
16	D6[x]	0:Desactivado-1:Control de flujo CTS.
17	D3[x] DIO-3	0:Desactivado-2:Analog Input-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH.
18	D2[x] DIO-2	0:Desactivado-2:Analog Input-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH.
19	D1[x] DIO-1	0:Desactivado-1:Botón de identificación de nodos habilitado-2:Analog Input-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH.
20	D0[x] DIO-0	0:Desactivado-1:Botón de identificación de nodos habilitado-2:Analog Input-3:Digital Input-4:Digital Output LOW-5:Digital Output HIGH.

Cuadro 5.4: Comandos de configuración de pin.

Los comandos brindan herramientas a la hora de tomar acciones o consultar parámetros de los dispositivos. A continuación, se verán los frames que se utilizan en el proyecto, nos permiten simplificar la operación entre varios módulos y proveer un control más detallado de los nodos de la red. Los frames que se muestran como ejemplo no tienen relación entre ellos, son simples ejemplos para ver la estructura en sí.

5.2.1. Frame 0x17 (Envío de Comando Remoto)

Esta trama es utilizada para realizar el envío de comandos hacia un dispositivo que sea parte de la red, de forma remota. Como parámetro adicional debe colocarse el byte conocido como *Remote Command Option* a un valor hexadecimal de *0x02*

para aplicar los cambios deseados instantáneamente. Un ejemplo de este tipo podría ser el siguiente:

”7E 00 11 17 01 00 13 A2 00 40 3A B3 10 FF FE 02 53 50 03 E8 68”

Nombre	Byte	Valor	Descripción
Start Delimiter	0	0x7E	Indica el comienzo de una trama.
Lenght	1(MSB) 2(LSB)	0x00 0x11	Numero de bytes del frame entre el Lenght y el Checksum.
Frame Type	3	0x17	Indica el tipo de trama.
Frame ID	4	0x01	Este byte se adjunta como un número de serie asociado al comando enviado. Permite relacionar la respuesta de un módulo a una acción seleccionada cuando se envían los comandos. Si este campo es 0x00 el usuario no obtiene respuesta, pero los comandos son ejecutados.
64-bit Destination Address (MAC)	5(MSB) 12(LSB)	0x00 0x13 0xA2 0x00 0x40 0x3A 0xB3 0x10	Dirección única del dispositivo remoto, allí se enviará el comando. Si el usuario requiere enviar un broadcast deberá utilizar la dirección especial ”00 00 00 00 00 00 FF FF” en este campo.

Cuadro 5.5: Frame detallado de Envío de Comando Remoto (0x17)

16-bit Destination Address	13,14	0xFF 0xFE	En este campo se establece la dirección de 16 bits del destino, si se conoce.
Remote Command Options	15	0x02	Aplica los cambios instantáneamente cuando su valor es "0x02".
AT Command	16,17	0x53 0xE8	Nombre del comando en codificación ASCII.
Command Parameter Value(opcional)	18,19	0x03 0xE8	Si esta presente, indica el valor del comando a modificar en binario. Si no se mandan parámetros, se omite.
Checksum	20	0x68	Campo de chequeo de errores, la operación es "0xFF" menos la suma módulo 256 de todos los bytes a partir de Length hasta el Checksum.

Cuadro 5.6: Frame detallado de Envío de Comando Remoto (0x17)

5.2.2. Frame 0x97 (Respuesta de Comando AT Remoto)

Este frame siempre será la respuesta a una trama de tipo 0x17, lo recibirá el nodo coordinador en respuesta a una petición de comando hecha por sí mismo. Puede haber varias tramas para completar la respuesta, un ejemplo de este tipo será el siguiente: "7E 00 11 97 01 00 13 A2 00 40 3A B3 10 FF FE 4E 50 00 01 00 D9".

Nombre	Byte	Valor	Descripción
Start Delimiter	0	0x7E	Indica el comienzo de una trama.
Length	1(MSB) 2(LSB)	0x00 0x11	Numero de bytes del frame entre el Length y el Checksum.
Frame Type	3	0x97	Indica el tipo de trama.
Frame ID	4	0x01	Este byte se adjunta como un número de serie asociado al comando enviado.

64-bit Source(remote) Adress (MAC)	5(MSB) 12(LSB)	0x00 0x13 0xA2 0x00 0x40 0x3A 0xB3 0x10	Dirección única del dispositivo remoto que devuelve la respuesta.
16-bit Source(remote) Adress	13,14	0xFF 0xFE	Dirección de 16 bits del dispositivo remoto que devuelve la respuesta.
AT Command	15,16	0x4E 0x50	Nombre del comando que provocó la respuesta remota.
Command Status	17	0x00	Reporta el resultado de la acción requerida por el comando, pudiendo ser: 0x00(OK), 0x01(ERROR), 0x02(COMANDO INVÁLIDO), 0x03(PARÁMETRO INVÁLIDO) y 0x04(TRANSMISIÓN DE COMANDO REMOTO FALLIDA).
Command Data(opcional)	18 19	0x01 0x00	Contiene la información del registro de comando AT. Si el host solicita un cambio de parámetro este campo será omitido mientras que si consulta el valor del parámetro en la solicitud, este campo tendrá el valor correspondiente.

Checksum	20	0xD9	Resta de "0xFF" menos la suma en 8 bits de todos los bytes a partir del Length hasta el Checksum.
----------	----	------	---

Cuadro 5.7: Frame detallado de Respuesta a un Pedido de Comando Remoto (0x97)

5.2.3. Frame 0x92 (Recepción de Estados I/O de Dispositivo Remoto)

Este frame es de suma importancia para el proyecto. Se recibe en el nodo configurado como destinatario con sus correspondientes direcciones, el Xbee remoto enviará su información de forma periódica incluyendo los valores de muestreo analógico y estados digitales, siempre y cuando sus pines estén configurados como tales. Se activa mediante el parámetro *Sample Rate (IR)* o *Change Detect (IC)*. En caso de querer "forzar" el envío debe realizarse mediante el comando *AT IS* a través del frame 0x17 (envío de Comando Remoto) y recibirá a una respuesta de tipo 0x97 (Respuesta de Comando AT Remoto) en este caso.

El frame se compone de campos como el número de muestras, las máscaras analógicas y digitales con sus respectivos valores correspondientes a los sensores, salidas y estados.

Como ejemplo se reciben dos muestras analógicas y digitales, el frame tendrá una estructura del siguiente estilo: "7E 00 16 92 00 13 A2 00 40 3A B3 10 FF FE 01 01 00 18 03 00 10 02 2F 01 FE 21".

■ Analog Channel Mask

Este campo indica que ADCs están activos para la toma de muestras, por ejemplo, 0x03=0011b quiere decir que ADC0 y ADC1 están activas como entradas analógicas en el módulo XBee remoto. Cada muestra tiene un tamaño de 2 bytes. Los XBee tienen un ADC de 10 bits de resolución, es decir, puede tomar valores entre 0x0000 y 0x03FF. Deben convertirse los valores en un nivel

de voltaje, para hacerlo simplemente es necesario aplicar la siguiente fórmula: $ADC/1023(V_{Ref}) = \text{Voltaje}$. El campo tendrá la siguiente correspondencia entre los bits y los ADC y será igual al campo “Analog Samples”.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
1	0	0	0	1	1	1	1
Vdd	N/A	N/A	N/A	ADC-3	ADC-2	ADC-1	ADC-0

Cuadro 5.8: Mascara de bits analógicos.

▪ Digital Channel Mask

Indica que pines digitales están activos, por ejemplo, 0x0018=00011000b se refiere a que las líneas DIO3 y DIO4 están activadas. El campo “Digital Samples” es 0x0010=00010000b por lo tanto DIO4 está en HIGH y DIO3 en LOW. El byte inferior y superior se presentan a continuación:

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
0	0	0	1	1	1	1	1
N/A	N/A	N/A	DIO-12	DIO-11	DIO-10	DIO-9	DIO-8

Cuadro 5.9: Mascara de bits digitales superior.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
1	1	1	1	1	1	1	1
DIO-7	DIO-6	DIO-5	DIO-4	DIO-3	DIO-2	DIO-1	DIO-0

Cuadro 5.10: Mascara de bits digitales inferior.

Una vez comprendido el significado y la manera de presentar las máscaras para los diferentes pines del XBee se analiza con detalle el resto de la trama:

Nombre	Byte	Valor	Descripción
Start Delimiter	0	0x7E	Indica el comienzo de una trama.

Lenght	1(MSB) 2(LSB)	0x00 0x11	Numero de bytes del frame entre el Lenght y el Checksum.
Frame Type	3	0x92	Indica el tipo de trama.
64-bit Destination Adress (MAC)	4(MSB) 11(LSB)	0x00 0x13 0xA2 0x00 0x40 0x3A 0xB3 0x10	Dirección de 64 bits del remitente.
16-bit Destination Adress	12,13	0xFF 0xFE	Dirección de 16 bits del remitente.
Receive Options	14	0x01	Puede tener dos estados: 0x01, envío de acknowledgement o 0x02 si fue un mensaje de Broadcast.
Number of Samples	15	0x01	Numero de muestras enviadas.
Digital Channel Mask	16 17	0x00 0x18	Mascara de bits, indica que I/O están configuradas como entrada digital. Los bytes son leídos en formato Big Endian.
Analog Channel Mask	18 19	0x00 0x03	Mascara de bits, indica que puertos de I/O están configurados como entradas analógicas.
Digital Samples(opcional)	20 21	0x00 0x10	Si el campo de máscaras digitales fuera nulo, este campo no será incluido en la trama. Caso contrario indicará los puertos de forma binaria mediante High(1) y Low(0).

Analog Samples(opcional)	22 23 24 25	0x022F 0x01FE	Bytes correspondientes al valor del ADC. Si el campo de máscara analógica es nulo este campo no será incluido en la trama.
Checksum	26	0x21	Resta de "0xFF" menos la suma en 8 bits de todos los bytes a partir del Length hasta el Checksum.

Cuadro 5.11: Frame 0x92 detallado, Recepción de Estados I/O de Dispositivo Remoto.

El detalle de cada frame a utilizar permite desglosar la información que contiene y utilizarla para un propósito destacando la importancia de este último, ya que será el frame que realice la medición continua de las variables que deseemos tanto analógicas como digitales. Mientras que los primeros dos frames mencionados son utilizados para generar acciones a través de eventos en caso que se requiera la activación de un actuador o desactivar algún PIN.

Utilizando con detalle estas tramas, se podrá realizar una interfaz de control para el usuario con mayor precisión.

Capítulo 6

Consideraciones de Hardware y Software

Para realizar un equipo de automatización y control deben tenerse en cuenta aspectos fundamentales de los posibles escenarios que debe enfrentar el desarrollador antes de poner en marcha los equipos electrónicos en campo.

Los ruidos sobre la señal, perturbaciones electromagnéticas alrededor del equipo pueden tener un efecto significativo sobre el funcionamiento. Para minimizar estos efectos, el desarrollador debe proveer un entorno seguro para la operación del dispositivo acorde a las necesidades y requerimientos presentes, eso incluye factibilidad económica y selección de componentes fáciles de hallar en el mercado electrónico local en el caso de este proyecto. Se pretende utilizar los circuitos para otros proyectos sin la necesidad de modificar demasiado el esquemático. Una vez listo el hardware, se pondrá a prueba la adquisición de datos analógicos y estados digitales para comprobar si la interpretación por parte del equipo es correcta, consumos y errores en la medición. Esta será la primer version y si bien no es un simple prototipo, sino un desarrollo mas complejo, acorde a los resultados de las evaluaciones se mejorarán los aspectos del diseño que así lo requieran.

Consideraciones

Los componentes seleccionados para el Hardware serán en su mayoría de tipo SMD (Surface Mounted Device) para reducir lo máximo posible el tamaño de la

placa final. Aún así, en esta primera etapa del diseño se considerará dejar un espacio adicional entre componentes debido a que será soldada a mano en primera instancia. Serán ubicados con un criterio tal que los circuitos ruidosos no estén cerca de los que son mas sensibles. Se generaran dos planos de tierra correspondiente a cada lado de la placa menos en la zona de optoacopladores y relés para evitar tenes un exceso de cobre en la zona donde debería aislarse completamente. Para el diseño de las entradas y salidas deberá tenerse en cuenta la posibilidad de ruido generado externamente o provocado por los circuitos, por lo tanto se deberá anteponer una protección para minimizar los errores.

Las entradas digitales serán de 12 - 24 VDC, ya que por lo general los sensores y los diseños de control de sistemas automatizados trabajan dentro de ese rango. Se deberá acondicionar la señal para poder conectarla al modulo XBee correctamente. Las señales analógicas, deberán ser capturadas y procesadas por el ADC interno de 10 bits. Con respecto a las salidas digitales, al ser la parte que implica más potencia de todo el circuito, se deberá aislar correctamente y tener en cuanta las corrientes que pueden ingresar al sistema desde la salida como posibles efectos provocados por los inductores. Se han desarrollado por separado los módulos que componen la placa , adquisición, control y salida. Para el diseño de cada una se tomo un criterio específico para lograr el objetivo esperado.

6.0.1. Alimentación

Para la alimentación, parte fundamental del diseño se seleccionaron previamente las tensiones a utilizar según las características de los posibles circuitos a implementar y sensores. Como primer medida el hardware funcionará con una alimentación de 24 VDC. A partir de ahí se adaptará el nivel a los valores que se requieren en los circuitos de acondicionamiento de señal y control. Para esto se empleará el uso de reguladores, los cuales llevarán la tension de entrada a 5VDC y luego a 3.3VDC, voltajes típico en componentes electrónicos. Debe contar con una buena regulación para mantener la estabilidad del sistema aún cuando la carga varíe.

Se utilizará un regulador de conmutación de bajo consumo y buenas condiciones de regulación llevando los 24VDC a 5VDC, utilizando la configuración recomendada por el fabricante para este caso (Figura 6.1).

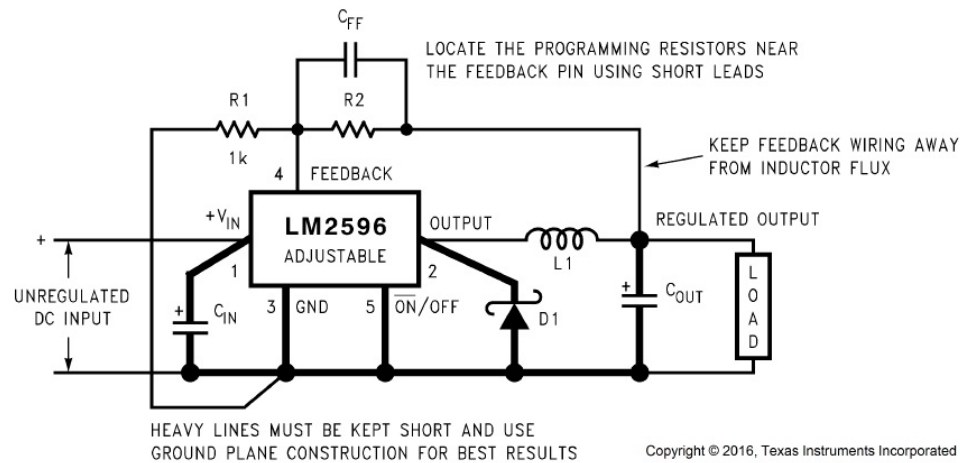


Figura 6.1: Configuración típica del regulador LM2596. Imagen tomada de [14].

Ésta será la primera etapa de acondicionamiento, el siguiente regulador llevará los 5VDC a 3.3VDC y allí se distribuirá a los componentes, principalmente al módulo de control XBee que trabaja a unos 3.3 VDC.

6.0.2. Entradas Digitales

Las entradas discretas están presentes en todos los diseños de control útiles para informar eventos y estados, en este proyecto serán entradas de tensión de 12 VDC o 24 VDC, comúnmente los sensores de las entradas dependen del voltaje con la que se alimentan los PLC.

Sin embargo, que se alimente de la fuente principal, no quiere decir que el módulo que las interprete trabaje al mismo nivel, si se conectara directamente implicaría la destrucción del sistema de control por la diferencia de potencial. Por este motivo, deben acondicionarse a los niveles de entrada recomendados por el fabricante.

Utilizando aislación galvánica, se separan ambas zonas de trabajo, cortando el flujo de corrientes entre ambas zonas, la de campo y la de adquisición del XBee.

Para el diseño de las entradas digitales se utilizará un componente muy común y fácil de encontrar, el optoacoplador. Existen distintos tipos cada uno con características diferentes, pueden encontrarse de tipo Triac, Mosfet, Darlington, etc. En particular

se escogió este último para el diseño ya que fue utilizado reiteradas veces en otros proyectos propios.

A pesar de estar aislado, las señales parásitas en campo pueden simular una señal de alto o bajo que erróneamente interpretara el equipo de control, por supuesto esta será una señal falsa y no informará el verdadero estado de la variable, por lo cual se debe agregar también un filtro para eliminar o reducir al máximo la posibilidad de ruido.

6.0.3. Entradas Analógicas

A lo largo de estos últimos años he tenido acceso y contacto con muchos tipos de sensores, y una de las conclusiones luego de trabajar con ellos es que una de las características principales que se incorporará en este proyecto es la posibilidad de que el módulo pueda interpretar sensores con salida analógica de tensión y también con salida de lazo de corriente 4-20 mA, esto hará mucho más versátil al dispositivo XBee ya que aumenta la posibilidad de selección de sensores a la hora de utilizar lo hecho en este proyecto. El módulo cuenta con una tensión de referencia de 3.3 VDC, por lo que cualquier tensión de señal de entrada no debe superar este valor. La resolución depende de esta tensión de referencia, y mediante ella se puede conocer el nivel de voltaje que efectivamente está ingresando al módulo si se la multiplica por el valor leído por el ADC.

$$Resolucion = \frac{3,3VDC}{1023} = 3,22mV$$

$$TensiondeEntrada = ValorADC * Resolucion$$

Al realizar el diseño del hardware, deben tenerse en cuenta nuevamente los casos no deseados, la presencia de posibles tensiones transitorias provocan variaciones o cambios de corta duración en los niveles de tensión de entrada, pudiendo ocasionar un problema de funcionamiento en el equipo o dañarlo permanentemente.

6.0.4. Salidas Digitales

Las salidas digitales se encargarán de activar o desactivar actuadores en caso de requerirlos, de manera automática dependiendo de una variable o por requerimiento propio del usuario por medio del software desarrollado para el proyecto.

A través de relés through hole se activarán las salidas con posibilidad de conectar una de ellas a 220 VDC. El módulo XBee no dispone de la potencia necesaria para realizar la conmutación del relé, por lo cual a través de un componente extra se amplificará la corriente y así poder activar las bobinas de relé.

La voltaje sobre la bobina corresponde a la siguiente relación: $V_L = L \frac{di}{dt}$, donde podemos decir que la tensión es cero cuando lo que circula por ella es una corriente constante, por lo que en continua se comporta como un cortocircuito, aunque una propiedad relevante del inductor es su oposición al cambio de corriente que fluye por él. De la ecuación anterior, un cambio discontinuo en la corriente por un inductor requiere una tensión infinita lo que no es físicamente posible. Así, un inductor se opone al cambio de corriente y su tensión puede cambiar abruptamente por lo que deben tomarse las precauciones a la hora de diseñar el circuito de salida.

6.0.5. Aplicación de Software

Ya se han descrito las consideraciones de los distintos bloques de operación en el Hardware.

También habrá consideraciones en las características del programa que maneja la aplicación. Haciendo un análisis de requerimientos se tomaron como objetivo los distintos aspectos que la aplicación debe brindar:

- envío de correo electrónico de alarma
- guardado de lecturas
- gráficos de variables en tiempo real
- envío de comandos
- archivo de instalación
- contraseña de usuario

Estas son las características principales, sin embargo detrás de ellos también hay requerimientos de diseño:

- el envío de correos por alarma debe poder utilizarse para enviar correos de forma manual y en respuesta a otros eventos, no solo por la presencia de una alarma.
- el envío de comandos remotos debe funcionar para enviar cualquier tipo de comando
- la pantalla principal de la aplicación debe ser subdividida en bloques mas pequeños, estos no solo deben servir para utilizarse en este proyecto sino también para cualquier desarrollo
- el diseño debe abarcar los cambios a futuro, con esto quiero decir que ante un cambio de escenario, el desarrollador debe poder realizar las modificaciones pertinentes de manera fácil y rápida sin tener que volver a diseñar todo el sistema

Al finalizar se harán las pruebas correspondientes para comprobar el funcionamiento de estos bloques.

Capítulo 7

Desarrollo

Luego de la recopilación de información y la toma de decisiones en cuanto al diseño del sistema, se puso en marcha el desarrollo del hardware y de la aplicación de software. Se irán desarrollando los distintos bloques que componen toda la interfaz.

7.1. Hardware

En este apartado se verá el Hardware propuesto para este proyecto, el desarrollo se realizó para adquirir los valores analógicos y digitales de los sensores por medio de los dispositivos End Device, Coordinador y Router no estarán en contacto con los sensores, los módulos con estas funcionalidades utilizarán un Hardware ya resuelto y presente en el mercado de muy bajo costo teniendo en cuenta que lo que buscamos en el nodo Coordinador es que sólo nos entregue los frames por el puerto serie y enviar comandos si se requiere en alguna ocasión, mientras que los Routers deben ampliar la red y realizar los procesos de Routing para la entrega de mensajes entre los dispositivos. Se optó por esta alternativa ya que el costo para desarrollar la placa de estos era mucho mayor que el Hardware comercial al que se hace referencia:

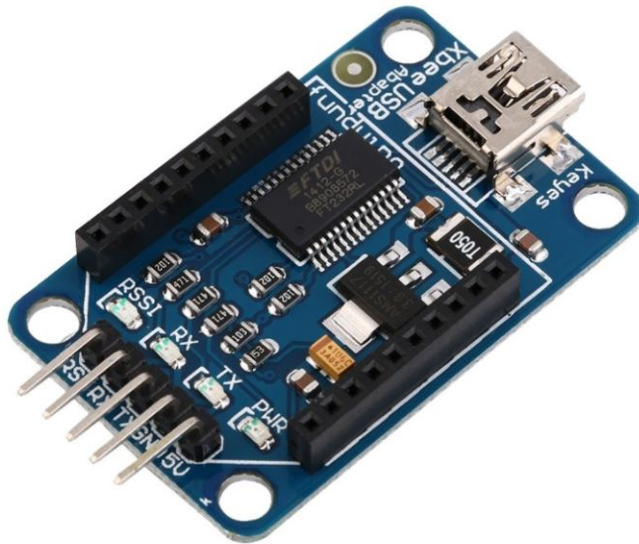


Figura 7.1: Adaptador mini USB Serial para módulos XBee.

La Figura 7.19 muestra la placa a utilizar, funciona como adaptador USB para los módulos y viene lista para conectar el XBee a la PC, cuenta con un chip FT232RL que proporciona la interfaz USB a UART, por lo tanto solo deben cargarse los drivers correspondientes en nuestra PC. Su tamaño es pequeño, y su tensión de alimentación es tomada directamente por el USB a 5 V, cuenta con un regulador acorde para llevarlos a los niveles de operación de XBee a 3.3 V, sólo necesitamos un cable mini para conectarlo, en caso del coordinador esta directamente ligado el puerto de la PC mientras que para los Routers se utilizará una fuente switching que entregue 5V, básica de bajo costo con salida USB similar a la que se observa en la siguiente figura:



Figura 7.2: Imagen representativa.

Los pines extra de la placa permiten utilizarlo directamente con un módulo Bluetooth, esto abre la posibilidad de trabajar con ello haciéndolo mas interesante ya que podemos transmitir los datos seriales vía BT, por supuesto no abordaremos ese desarrollo con Bluetooth en este caso, pero nos da la posibilidad de hacerlo si lo deseamos.

Por supuesto si en el futuro el proyecto implica tener un número considerable de dispositivos, seguramente en ese caso el desarrollo de una interfaz para Router y Coordinador debe ser realizada. Para este caso, se ha considerado reducir costos y tiempos utilizando el Hardware descrito, ya que la placa adaptadora, los cables USB necesarios y la fuente de 5V resultaban mucho mas baratos que comprar los componentes por separado y mandar a fabricar la placa.

7.1.1. Regulación de Voltaje

Para bajar el nivel de voltaje de 24 V a 5 V se escogió el regulador **LM2596** de Texas Instruments, presente principalmente en las fuentes Step-Down de bajo costo. Pueden adquirirse con salida de tensión variable o fija. Trabaja a una frecuencia de

150 KHz, puede soportar una entrada de hasta 40 V con salida entre 1.22 V a 37 V con carga. Su corriente máxima de salida es de 3 A(mucho más de lo requerido por el proyecto).

Mediante el Datasheet del regulador se aplicarán los conceptos recomendados para utilizarlo, en primera instancia se intentará utilizar los componentes de los ejemplos propuestos, en caso de no poder acceder a ellos se reemplazarán por otros con las mismas características, todo dependerá del mercado actual. Veamos los componentes del circuito en la siguiente figura:

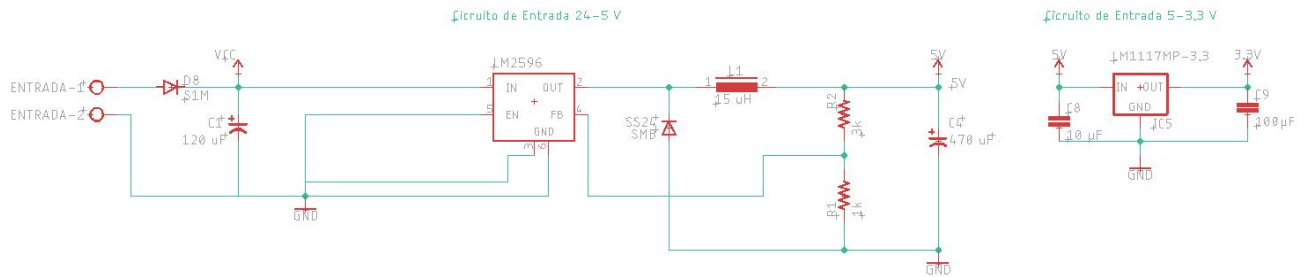


Figura 7.3: Circuito de regulación de voltaje.

De tipo ajustable, mediante el feedback y el divisor resistivo se fija la salida deseada en 5 V, la hoja de datos provee la ecuación específica y el criterio de selección de los demás componentes, la siguiente ecuación determina las resistencias de FB, tomando un valor recomendado de $R2=1\text{ K}\Omega$ de 1%, resulta una $R1=3\text{ K}\Omega$ con tensión de referencia del regulador de 1.23 V:

$$V_{OUT} = V_{REF} \left(1 + \frac{R1}{R2}\right) \quad (7.1)$$

$$5V = 1,23V \left(1 + \frac{R1}{1K\Omega}\right) \quad (7.2)$$

$$R1 = 3,06K\Omega \simeq 3K\Omega \quad (7.3)$$

Para la entrada se requiere un capacitor de aluminio o tantalio de baja ESR (resistencia serie equivalente), este parámetro puede definirse como la resistencia dinámica pura que opone el capacitor al paso de una corriente alterna e incluye a la resistencia de sus terminales, resistencia continua del material dieléctrico, de las

placas y la resistencia alterna en fase del dieléctrico a una determinada temperatura y frecuencia. La ESR no es una resistencia física dentro del capacitor, sino una resistencia que se genera sólo en corriente alterna. El capacitor de entrada debe prevenir los transitorios de voltaje y proveer la corriente instantánea requerida cuando el switch esta en estado ON, la corriente RMS que fluye en esta configuración es relativamente alta por lo que el parámetro de corriente RMS del capacitor debe ser alta.

El flujo de la corriente de ondulación (Ripple Current) a través de la ESR lleva al aumento de temperatura, la clasificación antes mencionada esta determinada por la cantidad de corriente requerida para elevar la temperatura interna. La capacidad de disipación del calor que posea el capacitor determinará la cantidad de corriente RMS que puede mantener el capacitor de forma segura.

La hoja de datos del regulador recomienda los siguientes aspectos al desarrollador de hardware: utilizar un capacitor cuyo voltaje sea de al menos 1.5 veces el valor de entrada, teniendo en cuenta que en este caso es de 24 V se tomará un valor aproximado de 36 V. La curva de capacitores diseñados para switching con baja ESR ayuda a la selección del componente en estas condiciones, el punto de interés se ubica en el cruce de 40 V y 500 mA en la siguiente figura:

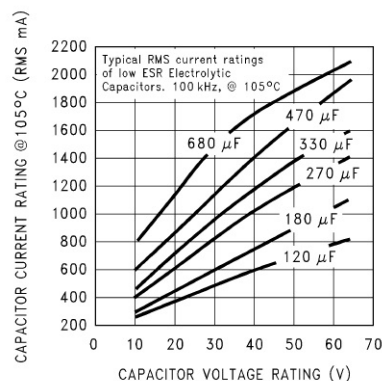


Figura 7.4: Curva típica de corriente RMS vs tensión del capacitor.

Se utilizará un capacitor de 120 µF/35 V para la entrada.

En cuanto a la salida se requiere uno de baja ESR ya que si éste parámetro es alto implicaría un ripple en la salida también elevado. Preferentemente el capacitor a

seleccionar deberá ser de tantalio ya que presenta baja ESR y buenas características a altas frecuencias, o un capacitor electrolítico con características de baja ESR, que se utilizará en primera instancia, la ESR se relaciona con el valor de capacidad y tensión, en muchos casos capacitores con tensión de operación elevada tiene una ESR baja por lo que es común encontrar capacitores cuyo voltaje de operación es mucho más elevado que la tensión de salida del circuito.

Que este valor sea bajo no quiere decir que es suficiente, en caso de muy bajas ESR existe la posibilidad de generar un bucle inestable en el feedback generando oscilaciones aun mayores. Se provee una tabla comparativa para un rápido diseño con las posibilidades de elección through hole y SMD de distintos tipos:

OUTPUT VOLTAGE (V)	THROUGH-HOLE OUTPUT CAPACITOR			SURFACE-MOUNT OUTPUT CAPACITOR		
	PANASONIC HFQ SERIES ($\mu\text{F/V}$)	NICHICON PL SERIES ($\mu\text{F/V}$)	FEEDFORWARD CAPACITOR	AVX TPS SERIES ($\mu\text{F/V}$)	SPRAGUE 595D SERIES ($\mu\text{F/V}$)	FEEDFORWARD CAPACITOR
2	820/35	820/35	33 nF	330/6.3	470/4	33 nF
4	560/35	470/35	10 nF	330/6.3	390/6.3	10 nF
6	470/25	470/25	3.3 nF	220/10	330/10	3.3 nF
9	330/25	330/25	1.5 nF	100/16	180/16	1.5 nF
12	330/25	330/25	1 nF	100/16	180/16	1 nF
15	220/35	220/35	680 pF	68/20	120/20	680 pF
24	220/35	150/35	560 pF	33/25	33/25	220 pF
28	100/50	100/50	390 pF	10/35	15/50	220 pF

Figura 7.5: Tabla de selección de capacitor de salida.

El caso mas cercano es el de 6V, con un capacitor de 470 $\mu\text{F}/24\text{V}$. Los de tantalio son recomendados para temperaturas menores a 25 grados bajo cero, debido a que los electrolíticos aumentan significativamente su ESR bajo estas condiciones de operación. En la mayoría de las aplicaciones capacitores electrolíticos de baja resistencia serie de valores de 82 μF y 820 μF proveen buenos resultados(dato del datasheet). Si la ESR aún es elevada (lo que implica un aumento de la temperatura y ripple) con el primer capacitor seleccionado, deberá colocarse otro en paralelo para compensar el consumo y elevar la vida útil de estos.

Los reguladores switching pueden operar en dos modos, continuo (CCM) o discontinuo(DCM). La diferencia está en la corriente del inductor, en modo continuo la corriente fluye de forma continua y nunca toma un valor de cero durante la conmutación. Por otro lado en modo discontinuo la corriente del inductor cae a cero durante dicho período.

En este caso el regulador opera en modo CCM, la hoja de datos nos da una guía de selección del inductor para minimizar los tiempos en el diseño en este modo de operación, la curva del inductor para el modelo de regulador ajustable es la siguiente:

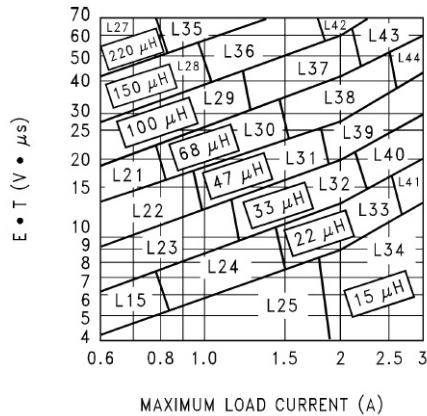


Figura 7.6: Valores de selección de inductor para LM2596-ADJ.

Primero veamos el valor de la constante del inductor Volt-microsegundo $E.T(V \cdot \mu s)$ con la ecuación provista por el fabricante:

$$E.T = (V_{IN} - V_{OUT} - V_{SAT}) \cdot \left(\frac{V_{OUT} + V_D}{V_{IN} - V_{SAT} + V_D} \right) \cdot \frac{100}{150kHz} (V \cdot \mu s) \quad (7.4)$$

De la hoja de datos $V_{SAT} = 1,16V$ y asumiendo una caída en el diodo de aproximadamente $V_D = 1,5V$, obtenemos como resultado $E.T=0,02(V \cdot \mu s)$. El inductor a utilizar en primera instancia será de $L=15\mu H$ ubicado en la zona del gráfico de los valores más pequeños, ya que nos da una $E.T$ menor y una corriente de consumo máxima mucho menor a la propuesta, se estima una corriente máxima de 300 mA (la corriente de operación es mucho menor) y un voltaje de entrada de 24V.

El regulador requiere un diodo para la corriente del inductor cuando el el switch esta en estado off, lógicamente deben ser rápidos para conmutar principalmente en aplicaciones de bajo voltaje de salida, por lo cual, los diodos de características “fast recovery” son requeridos. Se define el tiempo de recuperación inversa como el espacio de tiempo que transcurre desde que se inicia la conmutación hasta que el diodo adquiere un grado determinado de desconexión, es decir, un nivel de corriente inversa. Idealmente en la conmutación, la corriente iría instantáneamente desde un valor de

corriente directa en estado de conexión, a la corriente de saturación inversa de desconexión, sin embargo, antes de que el dispositivo pueda ser conmutado de ON a OFF deben ser retiradas las cargas portadoras minoritarias almacenadas en las regiones del semiconductor. La siguiente figura muestra la tabla de diodos recomendados por el fabricante:

VR	3-A DIODES				4-A TO 6-A DIODES						
	SURFACE-MOUNT		THROUGH-HOLE		SURFACE-MOUNT		THROUGH-HOLE				
	SCHOTTKY	ULTRA FAST RECOVERY	SCHOTTKY	ULTRA FAST RECOVERY	SCHOTTKY	ULTRA FAST RECOVERY	SCHOTTKY	ULTRA FAST RECOVERY			
20 V	SK32	All of these diodes are rated to at least 50V.	1N5820	All of these diodes are rated to at least 50V.		All of these diodes are rated to at least 50V.	SR502	All of these diodes are rated to at least 50V.			
			SR302				1N5823				
			MBR320				SB520				
30 V	30WQ03				1N5821						
	SK33				MBR330		50WQ03			SR503	
					31DQ03					1N5824	
			1N5822			SB530					
40 V	SK34		SR304	50WQ04		SR504					
	MBRS340		MBR340			1N5825					
	30WQ04	MURS320	31DQ04	MUR320		MURS620	SB540	MUR620			
50 V	SK35	30WF10	SR305			50WF10		HER601			
or	MBRS360		MBR350		50WQ05		SB550				
More	30WQ05		31DQ05				50SQ080				

Figura 7.7: Tabla de selección de diodo.

Para obtener un margen de seguridad, la clasificación de corriente en el diodo debe ser al menos 1.3 veces la corriente de carga y un voltaje inverso de por lo menos 1.25 veces el voltaje de entrada. Al operar con 1 A y 24 V a la entrada, está fuera de los rangos expuestos en la tabla. Para este caso se escogerá un diodo de montaje superficial SS24 de 2A/40V.

Luego de esta primera etapa de regulación, vuelve a disminuirse el nivel de tensión, pero en este caso a 3.3 V con un regulador LM1117. Este chip sólo requiere de dos capacitores para reducir el ripple, uno de entrada y otro de salida, como ventaja la tensión que ingresa a él ya fue regulada. Muy cercano a este dispositivo se encuentra el módulo XBee cuyo voltaje es provisto por este regulador, se espera que sea lo más estable y que no ingrese ruido al circuito principal debido a que podría ocasionar lecturas erróneas ya que la referencia que tome el módulo no es estable, en este proyecto la adquisición del conversor analógico digital es de suma importancia por lo cual esto afectaría directamente en el desempeño final.

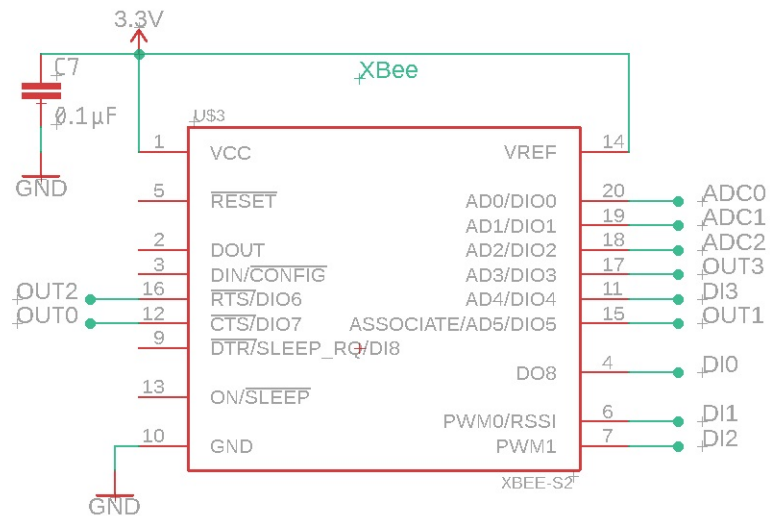


Figura 7.8: Modulo Xbee.

Por último a la entrada se agrega un diodo S1M de 1A como protección ante una posible polaridad inversa en la bornera al conectar la alimentación del circuito.

7.1.2. Entradas Digitales

El circuito de entradas digitales tiene distintas etapas para acondicionar la señal que ingresa al sistema. Al tratarse de señales discretas de 24 VDC lo primero es aislar galvánicamente las señales externas del circuito de control (XBee) ya que la tensión es excesiva para el rango operativo del módulo. Cuando se establece una corriente en el diodo emisor, este emite un haz de luz infrarrojo a través de la pequeña guía de onda de plástico o cristal hacia el fotoreceptor. La energía que incide sobre este último genera una diferencia de potencial en su salida. El optoacoplador **4N35** es un componente que ya e utilizado en otros proyectos y de los cuales pueden conseguir fácilmente en SMD.

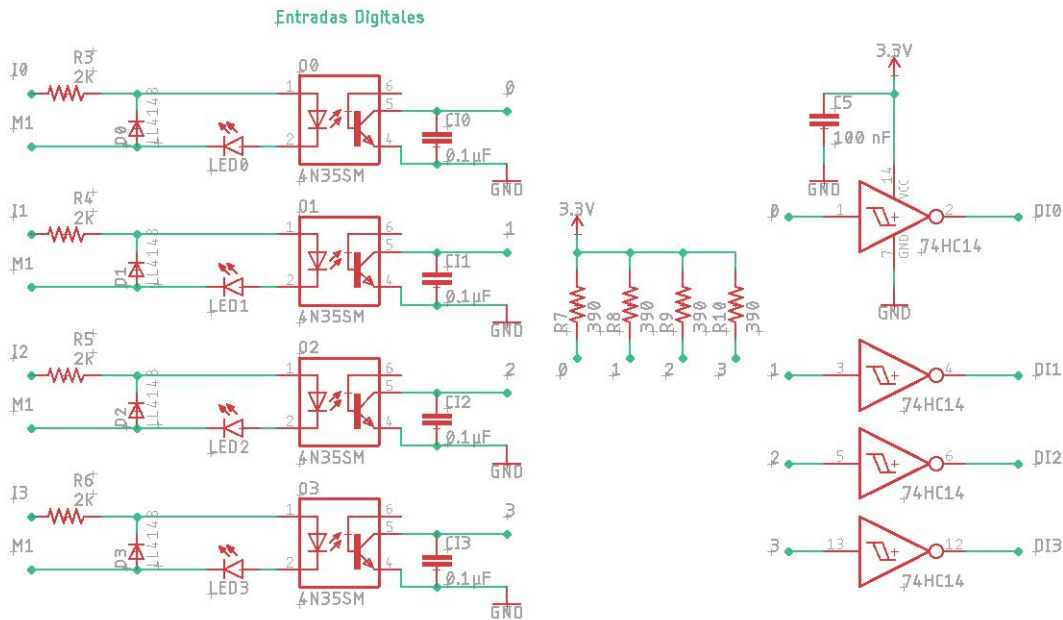


Figura 7.9: Circuitos para entradas digitales.

Para el diseño deben tenerse en cuenta los parámetros mas importantes de este tipo de dispositivo.

Similar a la ganancia del transistor, la relación de transferencia de corriente (CTR Current Transfer Ratio) es una características fundamental de los optoacopladores, representa la relación entre la corriente de salida y de entrada expresado en porcentaje, este parámetro puede verse afectado por la temperatura:

$$CTR_{\%} = \left(\frac{I_C}{I_F}\right)100 \%$$

En el mercado se pueden encontrar distintos tipos, con salidas transistorizadas, SCR o Triac.

Ya que el dispositivo consta de un transistor en este caso, opera en dos modos de operación:

- *modo lineal*: a la salida del optoacoplador se tiene una señal que es copia de la señal de entrada multiplicada por el CTR.

- *modo lógico digital(corte y saturación)*: en este caso la señal toma un valor alto o bajo dependiendo lo que ocurre en la entrada.

A la hora de escoger un optoacoplador, debe tener un CTR adecuado para cada aplicación, si funciona como interruptor generalmente se utiliza un dispositivo de CTR alto o con temperaturas extremas se debe buscar aquel cuyo CTR no se vea afectado significativamente por estas condiciones.

Deben consultarse los parámetros máximos para no sobrepasarlos, como la corriente máxima en la entrada(que deberá ser limitada por una resistencia) y voltaje inverso, de lo contrario funcionará mal o se dañará permanentemente.

La referencia de tensión que determina los niveles alto y bajo son tomados desde el colector en la salida del optoacoplador, si no hay corriente fluyendo en el diodo emisor, el valor de la salida será aproximadamente 3.3 V despreciando las mínimas corrientes, por el contrario, cuando el diodo emisor es activado por la presencia de la señal de 24 V en la entrada, el transistor saturará y llevará la tensión a un nivel bajo, idealmente 0 V despreciando el voltaje de saturación del transistor. En la entrada se agrega un diodo para proteger polarizaciones inversas.

Estas señales lógicas ingresarán en una compuerta Schmitt Trigger 74HC14, de tipo inversora posee una alta impedancia de entrada, rango de salida *rail to rail* y bajo consumo de potencia. Los ruidos pueden generar valores inapropiados a los establecidos como alto y bajo a la salida del optoacoplador, estas pequeñas variaciones originan transiciones en la salida que son eliminadas con la histéresis para estabilizarlos frente a posibles oscilaciones. Los valores umbrales de este dispositivo se encuentran alrededor de 60% de la alimentación para un estado alto y un 40% para un estado bajo, su tensión de suministro es de 3.3 V tomados del regulador.

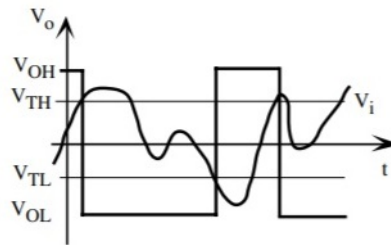


Figura 7.10: Respuesta Schmitt Trigger.

- cuando $V_i < V_p$, a la salida $V_o = V_{OH}$ y la tensión de referencia $V_p = V_{TH}$ (tramo 1)
- si V_i crece llegará a ser mayor a $V_p = V_{TH}$, esto provocará un cambio en la salida del comparador donde ahora $V_o = V_{OL}$ y el voltaje de comparación $V_p = V_{TL}$ (tramo 2 y 3)
- finalmente si se produce un cambio tal que $V_i < V_p = V_{TL}$ obtendremos a la salida $V_o = V_{OH}$ y $V_p = V_{TH}$ (tramo 4 y 1)

El comportamiento se ve reflejado en la forma de onda de la Figura 7.10, los cambios en la salida ocurren cuando la V_i alcanza los valores V_{TH} y V_{TL} . La activación de la compuerta no es inmediata, sino que lleva consigo un leve retardo de 6 ns, como el objetivo es mostrar estados lógicos no se hizo hincapié en la velocidad de conmutación ni de los retardos correspondientes.

Retomando el diseño del circuito con optoacoplador, su hoja de datos recomienda utilizarlo con una corriente en la entrada de $I_f = 10mA$, se selecciona este valor de operación ya que también presenta un máximo en la curva CTR en saturación que se mostrará a continuación. En la Figura 7.9 vemos el circuito de entrada, con I_f seleccionada se procede a comenzar los cálculos, se añade un led de presencia para poder visualizar la actividad en el puerto cuando se le aplica una diferencia de potencial de 24 VDC y una resistencia R_f , necesaria para limitar la corriente de activación del diodo interno del optoacoplador V_{Diodo} (cuya tensión es de 1.3 V). Se escogieron estados lógicos de 24VDC, principalmente por el hecho de que los sensores industriales generalmente operan a ese nivel:

$$24V = V_{LED} + V_{Diodo} + R_f \cdot I_f$$

$$24V = 2V + 1,3V + R_f(10mA)$$

$$20,7V = R_f(10mA)$$

$$R_f \simeq 2070\Omega$$

Un valor comercial cercano es 2000Ω , por lo que ahora la corriente $I_f = 10,3mA$. Los siguientes gráficos representan las curvas características del optoacoplador para una temperatura ambiente de 25 C . Para la corriente de entrada calculada, el valor CTR es 0.75 al cual le corresponde una corriente de colector de aproximadamente 8.5 mA , teniendo en cuenta cambios de temperatura y desgaste del dispositivo se selecciona un CTR menor, de 0.7 .

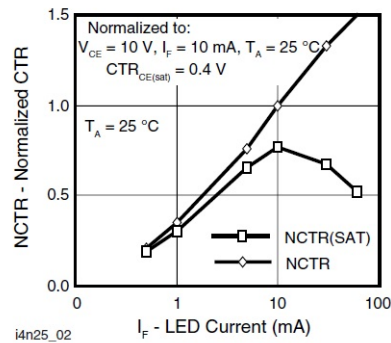


Figura 7.11: Curvas CTR lineal y en saturación con respecto a la corriente de entrada.

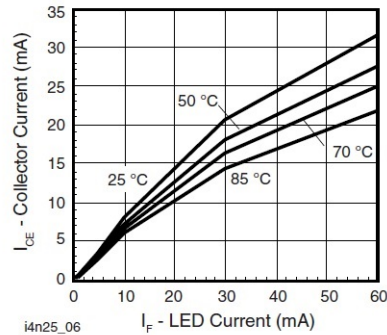


Figura 7.12: Curvas de corriente de colector vs corriente de entrada en función de la temperatura.

Con las consideraciones anteriores, se calcula la corriente de colector para el CTR dado y la resistencia R_c necesaria para asegurar la saturación del transistor tomando como referencia un valor de la hoja de datos $V_{CE} = 0,4V$ e $I_C = CTR(I_F)$:

$$I_C = (0,7)(10,3mA) = 7,21mA$$

$$R_c = \frac{V_{CC} - V_{CE}}{CTR(I_f)}$$

$$R_c = \frac{3,3V - 0,4V}{7,21mA}$$

$$R_c \simeq 402,22\Omega$$

Para un estado bajo se toma un valor de referencia máximo de 1 V. El Trigger es alimentado con 3.3 V y acusa en su hoja de datos un promedio aproximado V_{TL} del 30 % de su tensión de suministro, donde el máximo permitido ronda el 45 %. Para un estado bajo, se obtiene la resistencia mínima:

$$R_c > \frac{3,3V - 1V}{7,21mA}$$

$$R_c > 319\Omega$$

Finalmente el valor comercial de resistencia seleccionada es de 390Ω , con ella veamos la mínima corriente para no sobrepasar el valor de V_{TL} admitido en un

estado bajo:

$$I_c > \frac{3,3V - 1V}{390\Omega}$$
$$I_c > 5,89mA$$

Utilizando esta corriente de colector y nuevamente la relación $I_C = CTR(I_F)$, con $CTR=0.7$ obtenemos $I_f = 8,42mA$ Para este valor de corriente en el circuito de entrada obtenemos la tensión mínima, $V_{in} = 20,14V$, 16.08 % menor a la operativa de 24V. Como margen de operación para el circuito establecemos que se utilice con tensiones de $24V \mp 2V$.

Con voltaje de entrada mínima:

- Tensión de entrada mínima: 22 V
- $I_f=9.35$ mA
- $I_c=6.54$ mA
- $V_L=0.74$ V

Con voltaje de entrada máximo:

- Tensión de entrada máxima: 26 V
- $I_f=11.35$ mA
- $I_c=7.94$ mA
- $V_L=0.2$ V

En ese rango de tensiones aseguramos que no se supere el umbral máximo permitido de 1 V para un estado bajo en la entrada del trigger.

7.1.3. Entradas Analógicas

Para las entradas analógicas se utilizó un modelo de circuito diseñado para la placa industrial CIAA[7], en particular se escogió ya que se ha utilizado anteriormente y puesto a prueba en otros proyectos propios.

Este circuito tiene una versatilidad muy útil, ya que modificando sus jumpers es

posible utilizar dos configuraciones de distintas características, entrada lazo de corriente 4-20 mA o entrada de tensión 0-10 V. Estas son posteriormente adquiridas y procesadas por el conversor analógico digital del XBee. En la entrada se observan componentes de protección, el diodo TVS(Transient Voltage Suppression) protege al hardware de transitorios de tensión producidos por fuentes externas y el núcleo de ferrita suprime el ruido a altas frecuencias.

La señal también pasa por un amplificador operacional Rail to Rail, estos amplificadores tienen un rango dinámico que puede alcanzar niveles muy cercanos a su nivel de tensión de suministro(en este caso de 3.3 V tomados del regulador) en su entrada y salida.

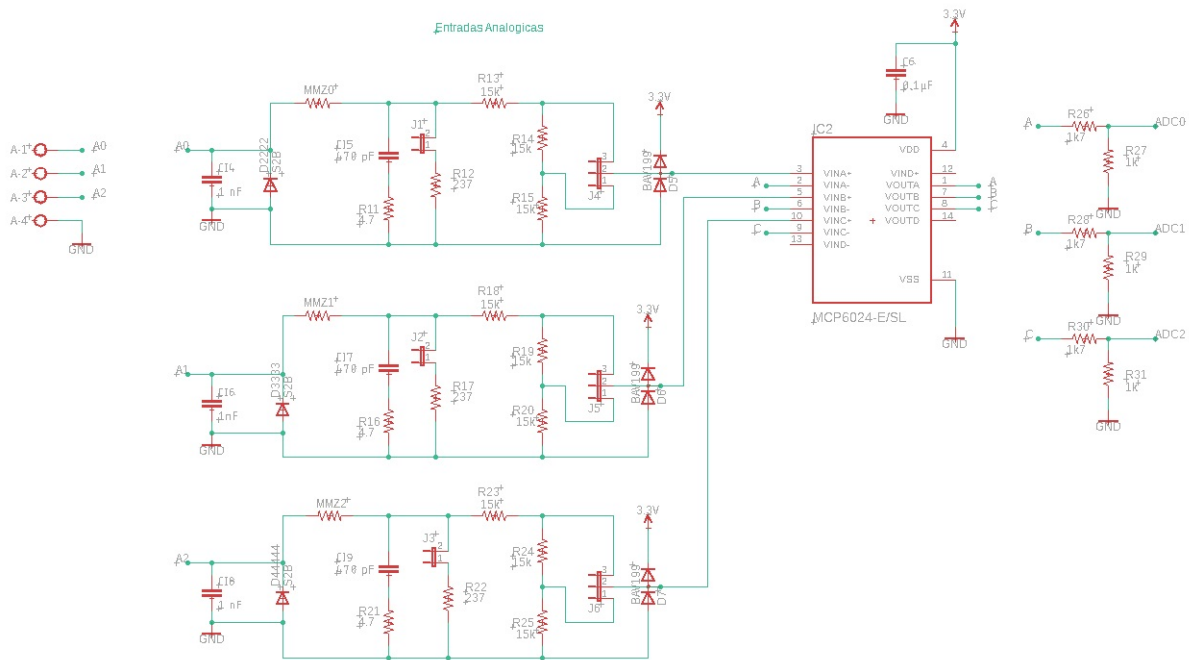


Figura 7.13: Circuito de adquisición para señales analógicas con protecciones, amplificador Rail to Rail y divisor resistivo.

Entrada 0-10 V

Para trabajar mediante sensores con salida de voltaje primero deben configurarse los jumpers conectando 1 – 2, ante una entrada máxima de 10 VDC la tensión caerá

sobre el divisor resistivo compuesto por las resistencias de $15K\Omega$. Realizando el calculo sobre la resistencia de entrada:

$$V_{In} = (VCC) \frac{R_{12}}{R_{10} + R_{11} + R_{12}} = (10V) \frac{15K\Omega}{45K\Omega} = 3,3V$$

La salida de tension calculada ingresa al operacional para asegurar correctamente el rango de entrada sin alinealidades y posteriormente mediante un divisor resistivo se reduce el voltaje a 1.2 V para respetar los niveles adecuados de operación en la entrada, recomendado por el fabricante del módulo XBee.

Entrada 4-20 mA

Para el lazo de corriente, nuevamente deben modificarse el estado de los jumpers, esta vez en 2 – 3 y el jumper que conecta a la resistencia de 237Ω , en este caso el lazo actuará sobre esta resistencia de carga. Para una entrada de 4 mA habrá una caída de tensión de 948mV, mientras que 20 mA provoca una diferencia de potencial de 4.74 V. Luego de pasar por el divisor resistivo (con las dos resistencias en serie de $15K\Omega$ se obtienen las tensiones de entrada al amplificador Rail to Rail:

$$V_{4Ma} = (237\Omega)(4Ma) = 0,95V$$

$$V_{20mA} = (237\Omega)(20Ma) = 4,74V$$

$$V_{In} = (V_{mA}) \frac{R_{14} + R_{15}}{R_{13} + R_{14} + R_{15}}$$

$$V_{In} = (0,95V) \frac{30K\Omega}{45K\Omega} = 0,63V$$

$$V_{In} = (4,74V) \frac{30K\Omega}{45K\Omega} = 3,16V$$

El fabricante recomienda operar en la entrada con tensiones de 1.2 V, a la salida de ambos casos se reduce la tension mediante un divisor resistivo de $1k7\Omega$ y $1k\Omega$.

7.1.4. Circuito de Salidas Digitales

Las salidas digitales son las encargadas de activar los actuadores de ser necesario. La placa cuenta con cuatro salidas a relé, una de ellas para circuitos de 220 VAC y las restantes para actuadores de 12 o 24 VDC.

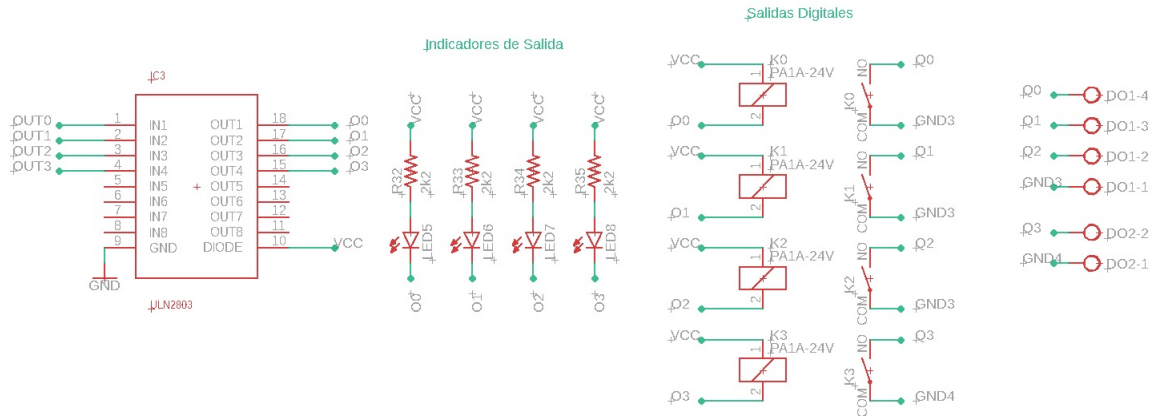


Figura 7.14: Circuito de salida a relé.

Los relé son de estado sólido, confiables y de tamaño reducido. Sus bobinas están conectadas a 24 VDC de la entrada de alimentación principal y son accionadas por los puertos digitales del XBee. Sin embargo el módulo no cuenta con la corriente suficiente para accionar el relé, para ello se emplea el dispositivo ULN2803 que consta de un array de transistores en configuración Darlington para amplificar la corriente. Como principal ventaja internamente el chip cuenta con diodos en cátodo común, estos diodos son necesarios principalmente por la presencia de cargas inductivas. Sin ellos, al interrumpir el paso de corriente la bobina intentará mantener el flujo induciendo una diferencia de potencial en sus bornes lo que llevaría a la destrucción interna de los transistores, el diodo brinda un “camino” correcto a las corrientes que fluyen hacia el chip protegiéndolo de este fenómeno.

Finalmente a continuación, las imágenes del Hardware en cuestión, veamos primero el diseño realizado en Eagle:

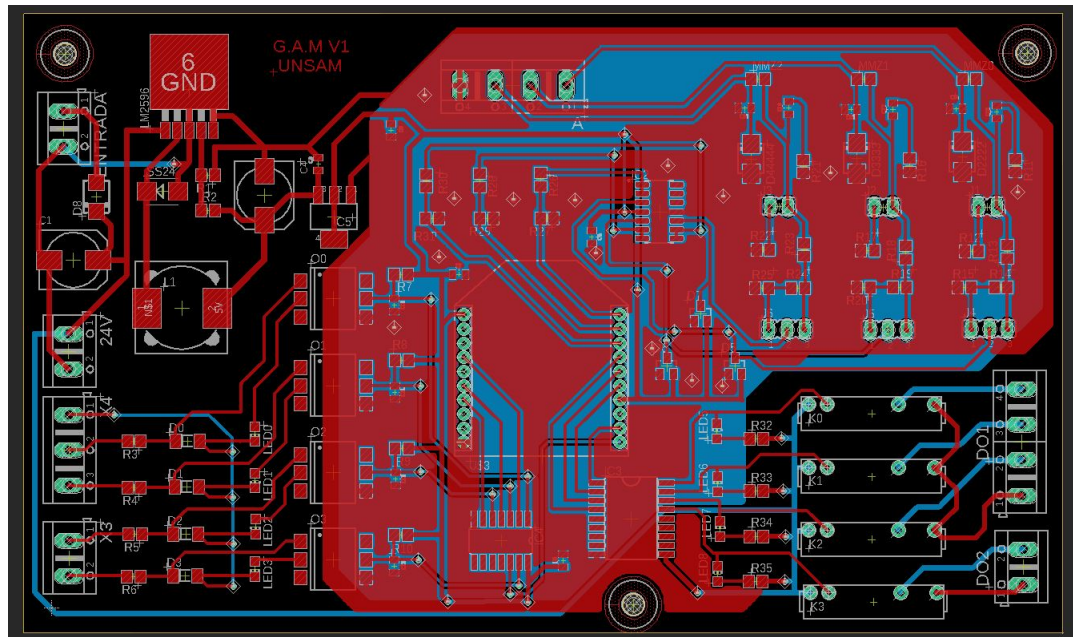


Figura 7.15: Diseño de Board en Eagle.

La imagen anterior muestra la Board ya terminada con sus respectivos planos de tierra en la capa Top y Bottom, como se ha mencionado no se realizó una reducción de tamaño de placa al colocar los componentes en esta version, fueron dispuestos para poder soldarlos de una forma cómoda teniendo en cuenta que son en su mayoría SMD.

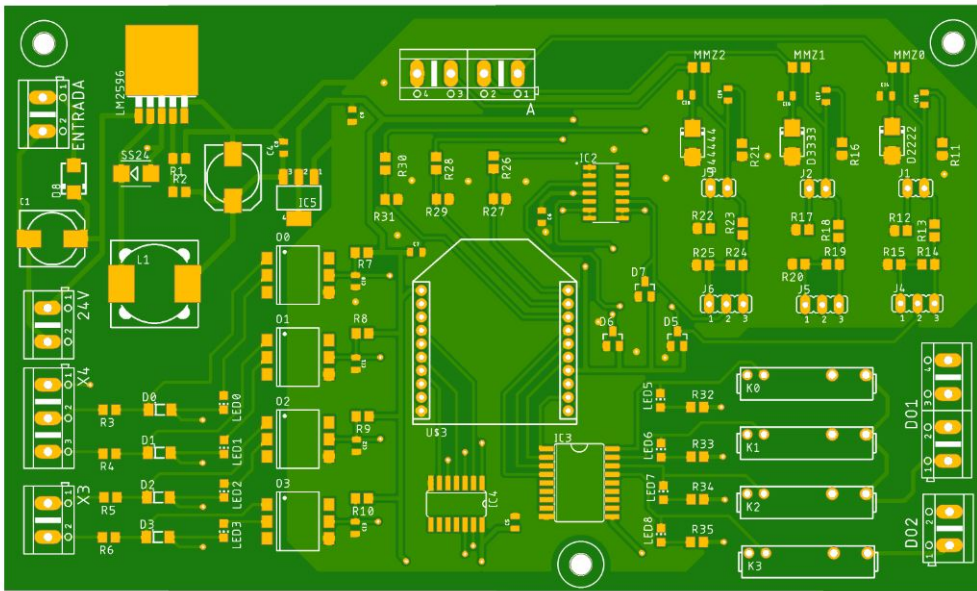


Figura 7.16: Capa Top.

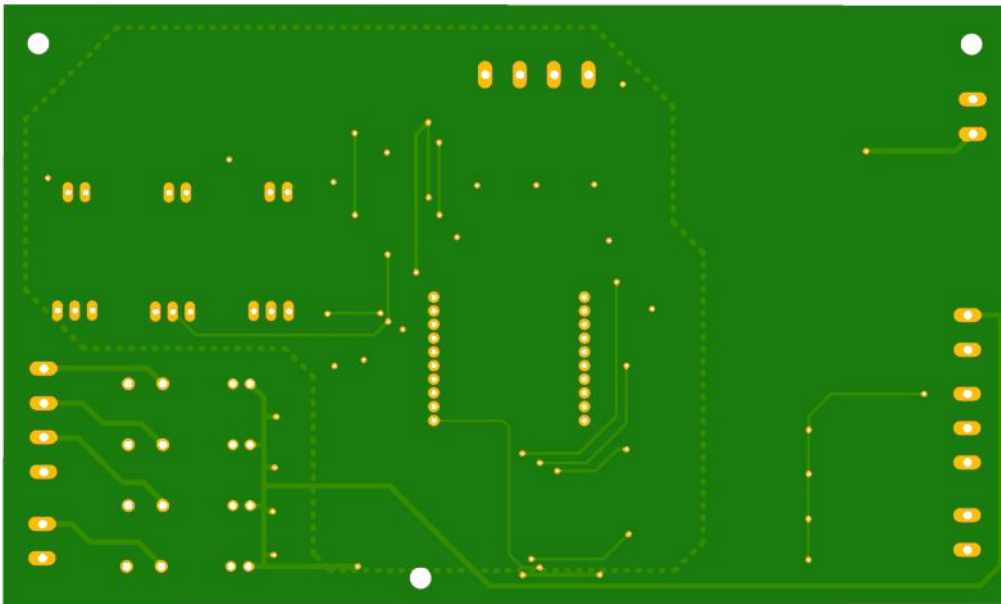


Figura 7.17: Capa Bottom.

Podemos ver las entradas digitales a la izquierda, las analógicas ubicadas arriba y

las salidas a la derecha. Luego de realizar el trabajo de impresión, la placa finalmente quedo de la siguiente manera:

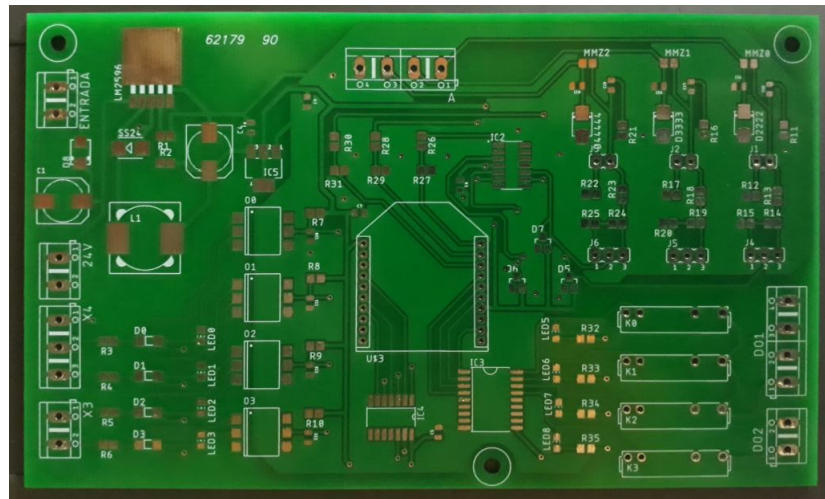


Figura 7.18: Capa Top.

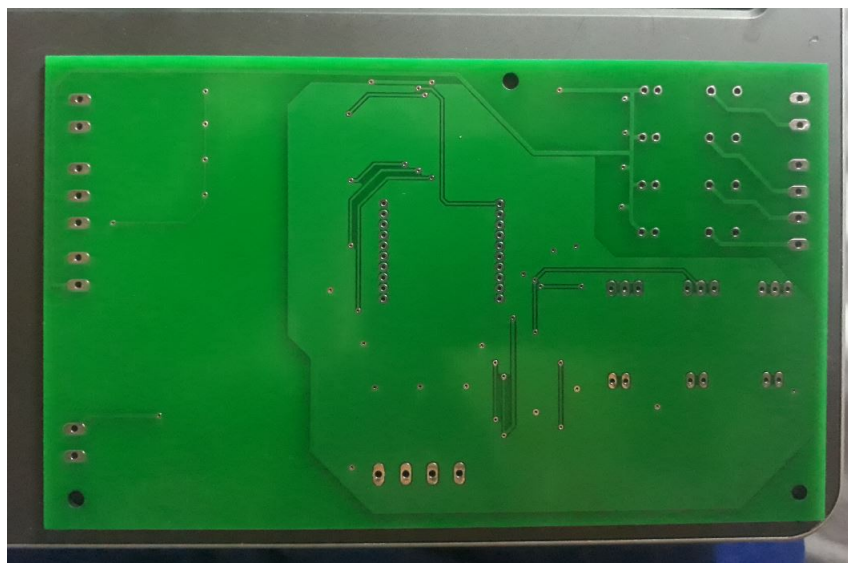


Figura 7.19: Capa Bottom.

Las pistas para la entrada del regulador son mas gruesas que el resto debido a que allí esta el grueso de la tension de entrada, lo mismo para los 24 V de los relé,

para evitar posibles ruidos generados por la bobina se recomendó alejarla un poco del resto de componentes. Los planos de tierra no se han implementado en las entradas digitales de 24 V ya que era un exceso de cobre teniendo en cuenta la cercanía de los componentes, allí en esa zona únicamente los optoacopladores actuarán como barrera de protección galvánica.

7.2. Software de Aplicación

La interfaz con el usuario fue desarrollada completamente en LabVIEW, a continuación se dará una breve introducción para poder comprender mejor como funciona este software.

7.2.1. Instrumentación Virtual

Para la interacción con el usuario, se realizará una interfaz de instrumentación virtual que permita trabajar con la adquisición de datos del Xbee. Como su nombre lo indica, un instrumento virtual está ligado al concepto de software, según la compañía National Instruments, mediante el desafío de crear un software capaz de ser utilizado en un ordenador como un instrumento para realizar mediciones surge el concepto de Instrumentación Virtual (VI) definido como *Instrumento que no es real, se ejecuta en una computadora y tiene sus funciones definidas por software*, esto le permitió a los ingenieros desarrollar nuevos sistemas utilizando estos nuevos conceptos. Mediante la representación gráfica en pantalla de elementos de visualización y control, permite al usuario interactuar con el desarrollo real de la aplicación de manera directa o mediante rutinas de programación.

INSTRUMENTO TRADICIONAL	INSTRUMENTO VIRTUAL
Definido por el proveedor	Definido por el usuario
Cumple una función específica, baja capacidad de interacción	Sistemas orientados a la aplicación, interactúa con redes, periféricos y otros
Basado en Hardware	Basado en Software
Dependiendo del instrumento puede tener costos elevados	Bajo costo, reprogramable

Tecnología del instrumento estable	Tecnología en constante desarrollo
Costo de desarrollo y mantenimiento elevados	El uso de software minimiza los costos de desarrollo y mantenimiento

Cuadro 7.1: Tabla comparativa Instrumento real vs Instrumento Virtual.

Utilizar un instrumento virtual, da versatilidad al proyecto y abre un abanico de posibilidades para utilizarse en distintos campos, se pueden encontrar cada vez más en ensayos, medidas y en el mercado de la automatización industrial donde los ingenieros la utilizan con el fin de reducir costos sin la necesidad de intervenir negativamente en el funcionamiento y demás procesos. A la hora de desarrollar con un instrumento virtual, se deben tener en cuenta distintos puntos importantes:

- Identificar los tipos de señales de entrada y salida, analógicas o digitales
- Protocolo de comunicación con el hardware
- Cantidad de señales que intervienen
- Diseño de un control en caso de requerirlo
- Implementación de timers en caso de necesitarlo
- Generación de reportes
- Generación de gráficos
- Requerimientos especiales, por ejemplo conversión de unidades, incorporación de código en otro tipo de lenguaje, etc

7.2.2. LabVIEW

Ademas de la investigación y posterior análisis en cuanto al tipo de tecnología que aplica este proyecto, siendo una parte fundamental en cuanto a la creación de

una red inalámbrica y la adquisición de los datos a través de sensores y señales discretas, también debe hacerse hincapié en el software de interfaz virtual donde se podrán observar las diferentes variables y donde el usuario tendrá contacto con el dispositivo XBee.

Como primer idea se utilizó un Software de Schneider Electric que reunía todos los requerimientos previstos para el diseño, además cuenta con una gran cantidad de herramientas y orientado principalmente al control y adquisición de datos, pero con una gran desventaja, su elevado precio.

El proyecto debe abarcar un costo reducido a comparación de las soluciones más sofisticadas presentes en el mercado de un alto valor monetario. Se seleccionó un software gratuito pero muy poderoso para realizar proyectos de este tipo, *LabVIEW* (*Laboratory Virtual Instrumentation Engineering Workbench*), que cuenta con una comunidad en constante desarrollo que aportan continuamente nuevas herramientas y proyectos.



Figura 7.20: Logo LabVIEW creado por National Instruments[13].

LabVIEW presenta una flexibilidad para realizar futuros cambios en los proyectos, tiene una interfaz con el usuario muy sencilla para programadores avanzados y personas con poco conocimiento previo. Proporciona una programación de tipo gráfica, posibilitando la creación de sistemas, con herramientas específicas para la adquisición, control, análisis y presentación de variables adquiridas por dispositivos electrónicos, simulado o real y embebido en proyectos de una alta complejidad. El entorno ofrece características interesantes:

- programación gráfica

- reducción de tiempo de desarrollo
- permite crear escenarios complejos y realizar cambios y actualizaciones de manera sencilla sobre los mismos
- permite la creación de bloques de procesamiento o control que pueden exportarse fácilmente a otros proyectos donde se requieran
- compatible con aplicaciones realizadas en otros lenguajes de programación

Se observa a continuación la estructura básica de software que permite la interacción usuario/hardware, que consta de distintas capas:

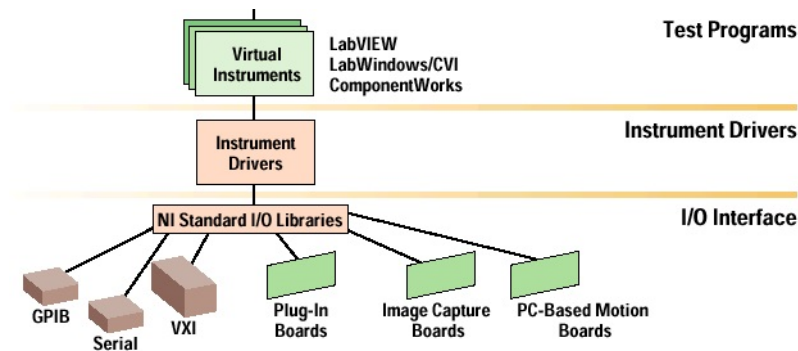


Figura 7.21: Estructura de Software[18].

Primero esta la capa de los elementos virtuales provistos por LabVIEW, luego los Drivers propios de los instrumentos, donde se programa que hacer con los datos que llegan a la aplicación, qué procesamiento conlleva mediante las herramientas del software y finalmente la capa de adquisición de entrada y salida de datos.

El software fue avanzando y actualizándose año tras año desde 1986 cuando se anunció su creación y se lanzó al mercado. Desde su primer Plug-In en tarjetas de adquisición de IBM que lo llevó al avance y desarrollo no sólo de la empresa sino de esta herramienta de software a tal punto que hoy en día es utilizado no sólo en tarjetas específicas de adquisición sino en todo tipo de áreas, proyectos con dispositivos electrónicos, programación embebida, comunicaciones, áreas de matemática y física, etc.

Cuenta con un gran soporte y se nutre de los foros oficiales ya que día a día los usuarios sorprenden con sus propias soluciones y en los lugares donde la aplican.

Entorno Virtual

Los programas creados por el entorno de LabVIEW se conocen como *VI's (Virtual Instruments)*. Cada uno de estos consta de dos partes fundamentales, una parte que interactúa con el usuario y otra que contiene el código y ambas están estrictamente relacionadas.

La interfaz con el usuario se conoce como **Panel Frontal**, allí se pueden encontrar pulsadores, gráficos, elementos de visualización entre otros. Los elementos se clasifican según el tipo, de control o de indicación.

La zona del código de fuente, se conoce como **Diagrama de Bloques**, y constituye los bloques de control, allí estará la programación que controlará lo que debe hacer nuestra aplicación. Todos los bloques se conectan mediante terminales.

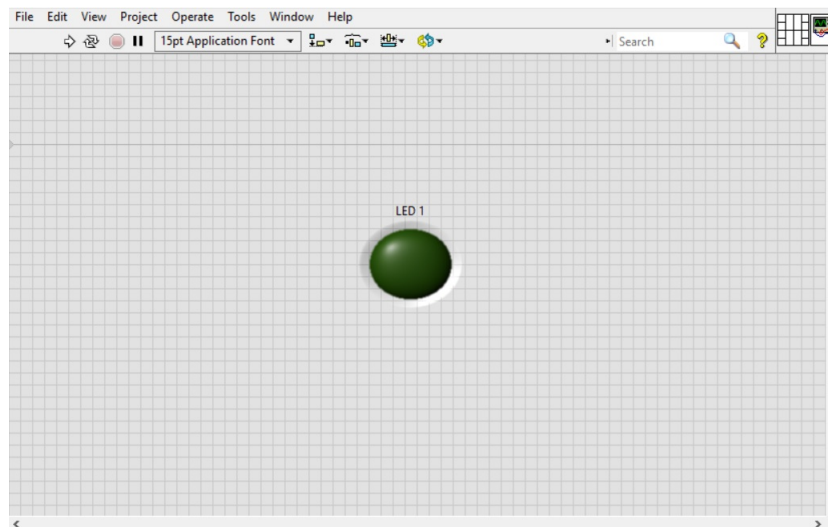


Figura 7.22: Pantalla de Panel Frontal.

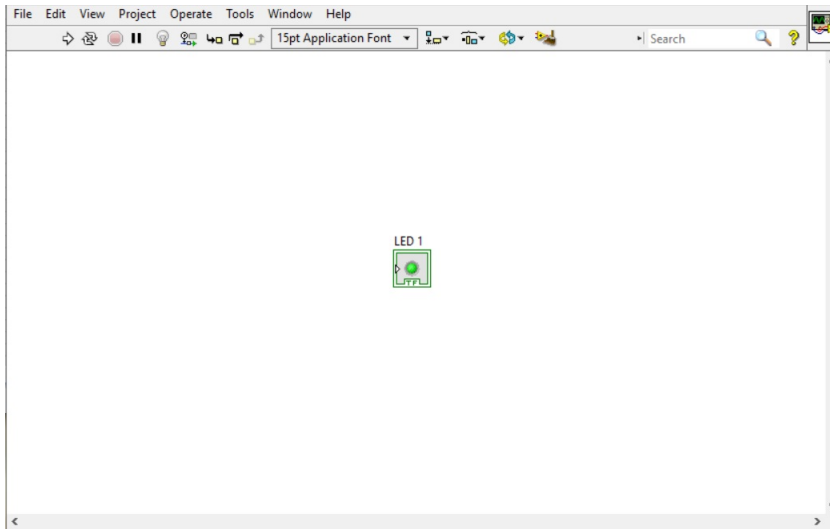


Figura 7.23: Pantalla de Diagrama de Bloques.

En el ejemplo de las Figuras 7.22 y 7.23 se muestra un led y su representación en el panel frontal como en el diagrama de bloques, ambos elementos constituyen un único elemento aunque se vean por separado, en uno se realiza el control y en el otro sólo la visualización.

En ambas figuras puede verse en la parte superior distintas opciones para la modificación del programa creado, esta es general para ambas pantallas.

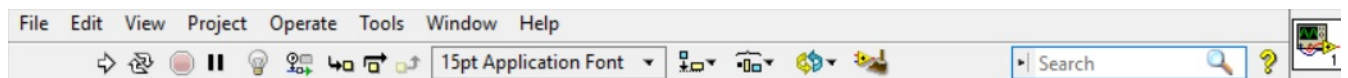


Figura 7.24: Barra de herramientas principal del LabVIEW.

Dentro de la barra principal se encuentran herramientas de distinto propósito:

- Control de ejecución del programa. Podrá ejecutar el programa de forma continua, pausarlo y detenerlo.
- Herramientas de depuración. Similar al que se suele utilizar en programación común, se puede observar la evolución de los datos a medida que el programa corre, con posibilidad de hacerlo manualmente paso a paso.
- Modificar el formato de los textos.

- Herramientas de diseño, con ellas se pueden alinear los distintos elementos presentes en las pantallas.
- En caso de requerir ayuda, se acude a este campo ingresando el nombre del elemento en cuestión.
- Ícono de descripción del programa actual. En caso de importarse en otro proyecto como un bloque, sólo se verá el ícono con sus respectivos terminales.

Además de la barra principal que es de uso general, se cuenta con una paleta de herramientas rápidas, también presente en ambas pantallas y se utilizan para la manipulación de elementos del sistema, herramientas de selección, cambio de posición de componentes, botón par realizar el cableado, Herramienta para movimiento de pantalla, herramienta de colores y creación de breakpoint similar al de los entornos a programación convencionales. Estas paletas de herramientas son las mas importantes y de las que se constituye un programa en LabVIEW, veamos un poco más en detalle.

Herramientas, Controles y Funciones

Dentro de las opciones particulares del **Panel Frontal** y el **Diagrama de Bloques** se encuentran todos los elementos necesarios para crear un proyecto en LabVIEW, veamos a continuación las mas utilizadas.

■ Panel Frontal

Cuando se trabaja desde el Panel Frontal, se utilizará la paleta de controles que se ve en la siguiente figura. Con ella se crean los elementos con los cuales el usuario interactúa en la aplicación. Dentro se encuentran controles y también indicadores, los controles se refieren a la entrada de datos y los indicadores a la salida. Puede verse en la imagen que hay diferentes tipos, modern, system, Express, etc. En la paleta **Modern** están los tipos de controles más utilizados.

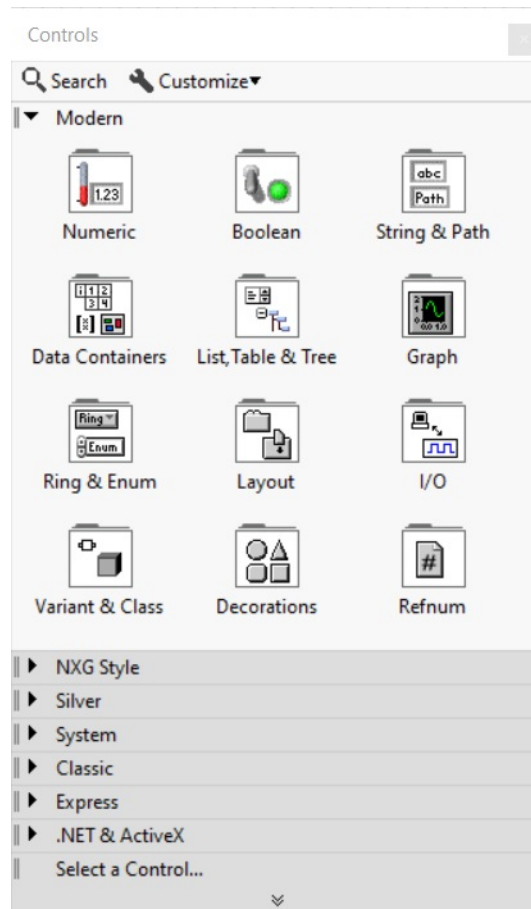


Figura 7.25: Controles del Panel Frontal.

- *Numeric*: introduce datos de tipo numérico
- *Boolean*: entrada y salida de indicadores booleanos
- *String and Table*: entradas y visualización de texto y tablas
- *List and Ring*: elementos para la creación de listas de opciones
- *Array and Clusters*: creación de arrays del mismo tipo, o de distintos formatos(Cluster)
- *Graph*: creación y edición de gráficos
- *Path and RefNum*: controles para el manejo de ficheros y archivos
- *Decorations*: agrega elementos a adicionales de decoración en el esquema del panel frontal

- *User Controls*: selecciona controles creados por el usuario
- *Active X*: controles para transferencia de datos hacia otros programas
- *Select Control*: busca un control por su nombre

■ Diagrama de Bloques

Así como la paleta de controles se despliega al utilizar el **Panel Frontal**, la paleta de funciones se encuentra en la pantalla de **Diagrama de Bloques**. Guarda los VI creados por el usuario para utilizar de forma general en cualquier proyecto y se los conoce como *SUB VI*. El menú Programming guarda las funciones usadas con más frecuencia y a su vez se sub-divide según el tipo de datos. Esta es muy similar a la paleta anterior en cuanto a la organización de las funciones y a que tipo pertenecen, solo se diferencian en los siguientes parámetros:

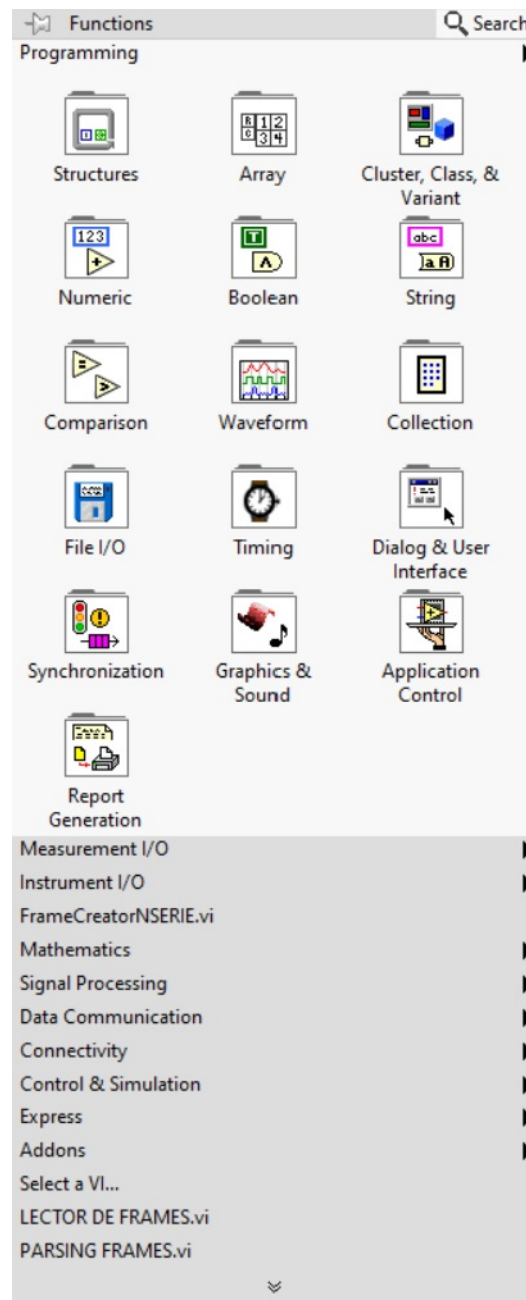


Figura 7.26: Funciones del Diagrama de Bloques.

- *Structures*: funciones conocidas en programación, bucles de tipo while, for, case y funciones de control
- *Comparison*: funciones de comparación de texto, números y booleanos

- *String*: funciones específicamente orientadas al manejo de Strings
- *Time and Dialog*: manejo de tiempo como retardos, contadores, timers y cuadros de dialogo
- *File I-O*: funciones para trabajar con ficheros
- *Communication*: funciones para intercomunicador ordenadores.
- *Instrument I-O*: muestra elementos virtuales creados por National Instruments para utilizar con tarjetas fabricadas de adquisición de datos, y controles de puertos para comunicación con el exterior
- *Data Acquisition*: funciones de procesamiento referidas a la adquisición de datos
- *Analysis*: funciones matemáticas, incluye operaciones básicas y otras más complejas como transformada de Fourier y Laplace

Se han visto brevemente los conceptos más básicos del Software para comenzar a utilizar el programa. Por supuesto, debe estudiarse de forma continua para poder programar a un nivel más avanzado en este entorno gráfico.

Siguiendo con el método de diseño, se dividió el desarrollo de software en pequeños módulos para luego integrarlos en una pantalla principal, por lo que a continuación se mostrarán los diferentes actores que intervienen en cada proceso de la interfaz, los denominados Sub-VI se intercomunican en el VI Principal, caso contrario el código sería ilegible por la cantidad de elementos presentes en pantalla, esto brinda un código más ordenado y fácil de interpretar.

Para los Sub-VI solo se mostrará la pantalla de diagrama de bloques, ya que es donde se puede apreciar el funcionamiento.

7.2.3. Estructura de Recepción

Como ya se ha dicho, se trabaja con frames y una estructura determinada que nos brinda el modo API de los XBee.

Para leer los datos que ingresan al puerto serie, se utiliza un patrón de diseño estándar que entrega LabVIEW a los desarrolladores, estas plantillas son reconocidas y aceptadas en la comunidad así como también simplifica el proceso de desarrollo sin tener que caer en soluciones complicadas y poco fiables. Han sido utilizadas por años y

probadas por los usuarios en campo.

Dentro del patrón de diseño se pueden encontrar diferentes conceptos, todos ellos para aplicarse dependiendo cual sea el caso:

- Máquina de Estados
- Interfaz de Usuario Basada en Eventos
- Productor/Consumidor
- Programación Orientada a Objetos

Este elemento virtual será el que reciba la información constantemente, por lo cual primero se plantea la siguiente pregunta:

¿qué hacer con los datos cuando ingresan al sistema?

Como la recepción es constante, el método debe asegurar no solo la lectura, sino también el procesamiento de estos antes de que llegue un nuevo mensaje, de lo contrario habrá pérdida de información.

Por lo tanto se necesitarán dos etapas, una de adquisición y una de interpretación de frame. El patrón de diseño Productor/Consumidor se basa en el patrón Maestro/Esclavo donde se maximiza el intercambio de información entre múltiples ciclos que pueden correr a distintas velocidades. Como su nombre lo indica estos se utilizan para la producir datos y para consumirlos en algún proceso.

Para la producción se emplea el uso de colas, utilizando **Queues**, estos son buffers de tipo FIFO (First Input - First Output), donde se guardará la información.

Para trabajar con ellos se utiliza su paleta de funciones, donde se pueden encontrar herramientas típicas como crear colas, insertar elementos en ella, obtener elementos de una ubicación específica, eliminar cola, etc.

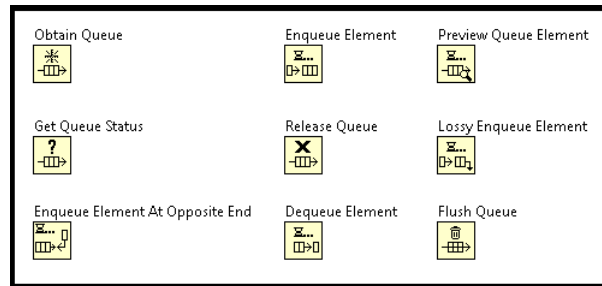


Figura 7.27: Funciones disponibles para el desarrollo con Queues.

Por esta razón comunmente este patrón de diseño es recomendable para utilizarse en procesamiento de datos ordena ya que beneficia el análisis ordenado del frame proveniente del XBee remoto.

Con esta información se procede a explicar detalladamente el funcionamiento del ciclo:

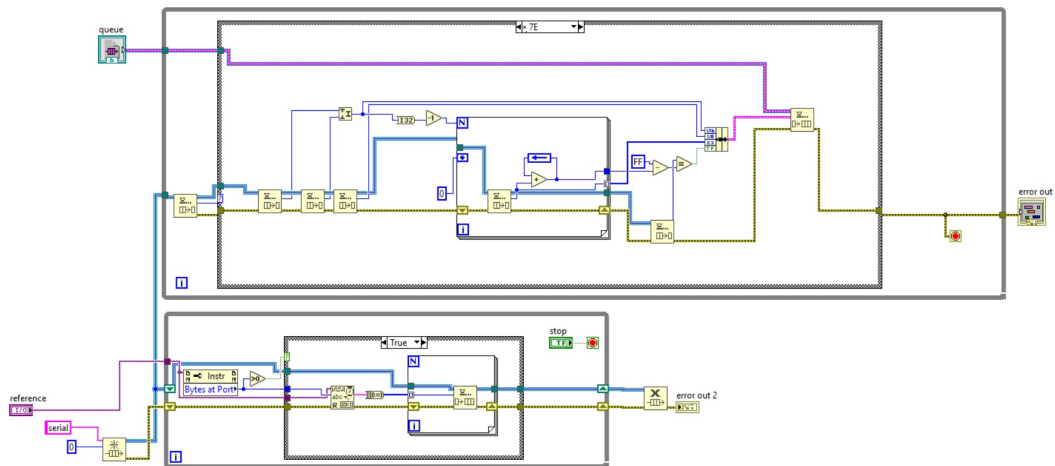


Figura 7.28: Patrón Productor/Consumidor con Queues.

Ciclo Productor

Este bloque se encarga exclusivamente de recolectar los bytes de información que llegan al puerto serie, si bien las formas API que se utilizan tienen una estructura formal, no se conoce con exactitud cuantos bytes van a ingresar ya que los frames

pueden cambiar dependiendo la situación.

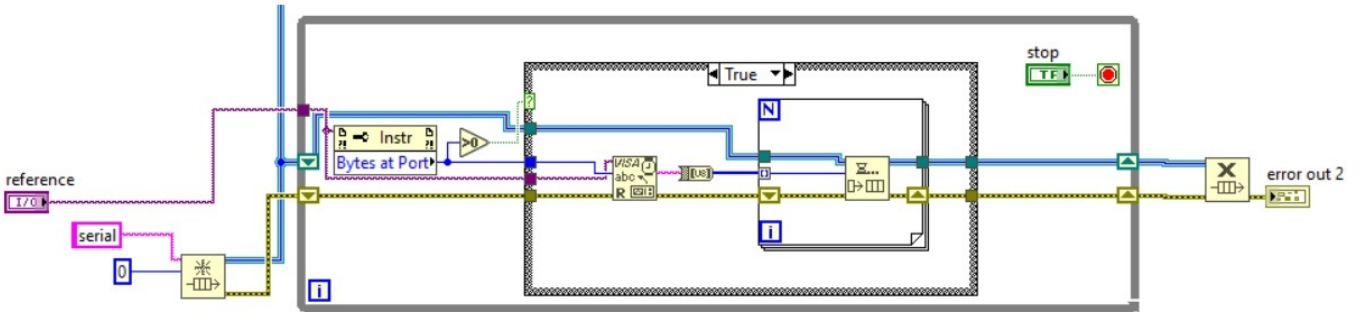


Figura 7.29: Patrón Productor.

En la figura se pueden ver las estructuras necesarias y los componentes para recolectar los datos, primeramente se tiene en cuenta que este proceso debe hacerse de forma continua, ya que deberá funcionar en todo momento monitoreando el puerto serie, para ello se realizó lo siguiente:

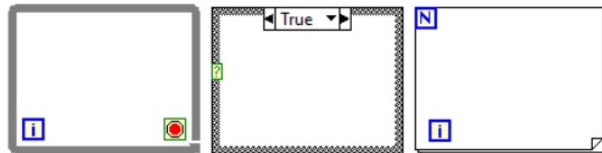


Figura 7.30: Estructura While (Izquierda) y Case (Medio) y Estructura For (Derecha).

- While:** se usa para crear Loops, todo proceso que este dentro del cuadro se ejecutará de manera continua. Posibilita agregar una condición de stop y brinda información entregando el número de iteración en la que se encuentra.
- Case:** lo que esté dentro de esta estructura se ejecutará sólo si cierta condición se cumple. Esta condición es muy versátil ya que puede ser un estado booleano, comparación de números y comparación de strings entre otros.
- For:** utilizado para realizar una acción un número definido de veces, brinda la información del número de iteración actual.

Todas las estructuras anteriores son iguales a los conceptos de programación en C. El while proporciona una ejecución continua de lectura y el case se utiliza para detectar la presencia de datos en el puerto.

Para monitorear el puerto se utiliza el nodo de propiedad llamado **Bytes at Port**, los nodos de propiedad son herramientas que permiten modificar características de los VI, conectando un control a su entrada, se puede seleccionar que puerto COM de la PC será el que ingrese al Loop, esta herramienta se activa al haber información disponible.

Una vez detectada la actividad del puerto, debe guardarse en alguna parte, se crea una cola por medio de las herramientas de **Queues** antes mencionadas.



Figura 7.31: Funcion Bytes at Port(Izquierda),Función de comparación(Medio) y Función de lectura de puerto(Derecha).

Cuando la cantidad de bytes en el puerto son mayor a 0, el bloque de comparación activa una señal booleana TRUE, que pone en marcha la ejecución de las tareas dentro de la estructura case. Allí dentro se leerá el contenido del puerto con la herramienta de lectura COM y se la convierte a un formato de Byte. Dentro del **For** se irán guardando los datos en la cola hasta que el puerto ya no tenga información haciendo inactiva la estructura case. De esta manera se hace la adquisición y guardado de los datos.

Ciclo Consumidor

El bloque se encarga de utilizar los datos obtenidos por medio del **Productor** y se observa que comparten su señal de **Queue**, esto es lo que genera el paralelismo de datos.

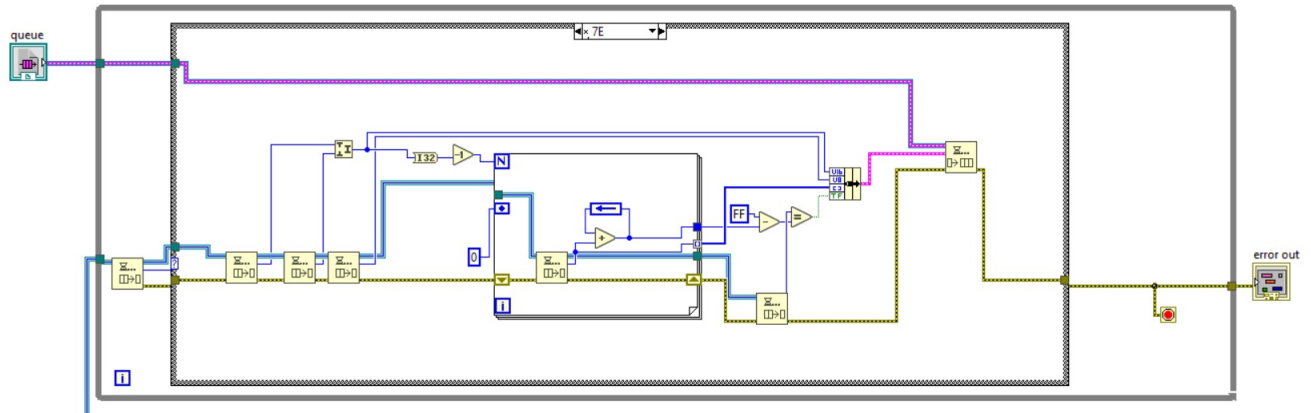


Figura 7.32: Patrón Consumidor.

Para este ciclo, nuevamente se acude a las estructuras antes mencionadas aunque en este caso el procesamiento es diferente. Dentro del **While**, el primer dato que llega al puerto es analizado, los API frames siempre comienzan con un valor hexadecimal de **0x7E**, de lo contrario la información será descartada inmediatamente. La estructura **Case** inicia el proceso de análisis siempre y cuando se cumpla la condición de que el byte leído sea del valor correcto. Se recuerda al lector que el frame luego del byte de inicio, informa el largo del mismo mediante los siguientes dos bytes.



Figura 7.33: De izquierda a derecha, Unión de Bytes, Función de Suma, Función de Comparación Igual, Función de Resta, Función de Formación de Cluster(Bundle).

Observar que en este caso se utiliza la función de **Queues** para extraer los elementos de la cola.

Al conocer el número de bytes, mediante una estructura **For** se suman N-1 (menos el último) para realizar el **Checksum** correspondiente y así validar cada trama con funciones aritméticas y de comparación .

Si la comprobación es exitosa, la información ya analizada se almacena en un Cluster por medio de la función **Bundle**, recordemos que Cluster es un simple array pero que contiene datos de distinto tipo. No solo se hace hincapié en el byte de inicio y el largo, sino también en el tipo de trama que se especifica en el cuarto byte.

Con toda esta información en el Cluster se almacenara:

- Largo de Trama
- Tipo de Trama
- Datos
- Señal booleana del Cheksum

Dicho esto, este Instrumento Virtual de recepción entrega la información enviada desde el módulo remoto para poder trabajar con ella. En el panel Frontal debe seleccionarse el **Queue** como una salida para poder interconectarla a los otros Instrumentos Virtuales, así como también una señal de error en caso de que algo inesperado suceda. Como entrada se indica el puerto serie de la PC en el que se va a conectar el XBee Coordinador. Para que el proyecto sea más legible es conveniente modificar el ícono del VI para familiarizarse con su funcionamiento y reconocerlo más fácilmente. Por supuesto este bloque puede utilizarse en cualquier otro proyecto que lo requiera.

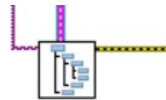


Figura 7.34: Ícono del VI Receptor con sus entradas y salidas.

7.2.4. Parsing

Luego de haberse concretado con éxito la etapa de adquisición de datos, se realiza el "Parsing", es decir se toma cada parte del frame y se analiza para extraer de él los valores de lecturas analógicas y valores discretos de entrada y salida.

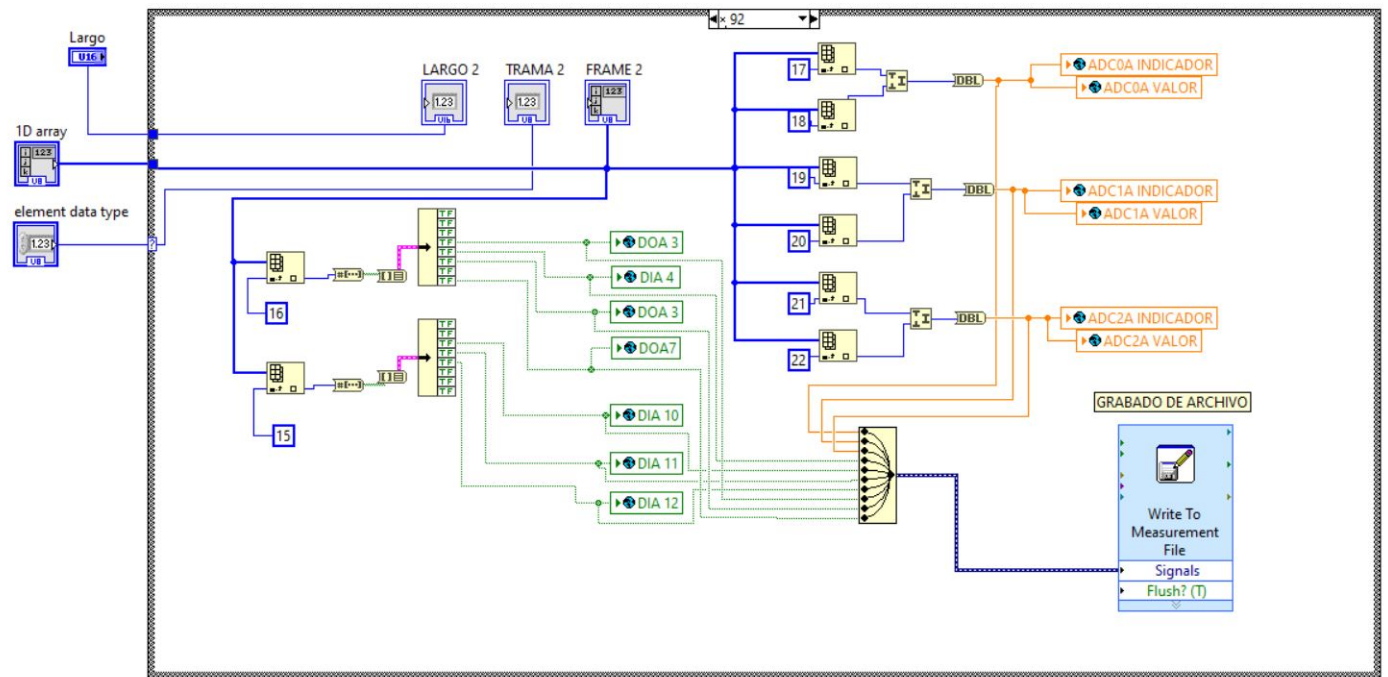


Figura 7.35: Elemento Virtual de Análisis y Guardado.

Este Instrumento Virtual recibe desde el exterior el largo de trama, tipo de trama e información de los demás bytes.

El frame que interesa aquí, es el **0x92-Recepcion IO Remota** analizada anteriormente en el Capítulo 6 - Modo API, con ella se extrae la información de los sensores leyendo bytes específicos. Con la estructura **Case** se comprueba si el tipo de trama es el que esperamos, una vez dentro se leen los estados de las variables.

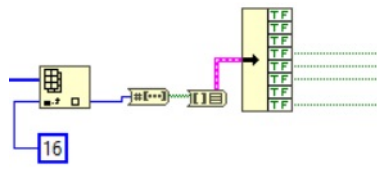


Figura 7.36: De izquierda a derecha, Función de Extracción por Índice, Conversion a Binario, Conversion a Cluster, Extracción de elementos del Cluster, Indicador.

En la Figura 7.36 se muestra el proceso para la extracción de los valores digitales

de entrada y de salida, estos están en la máscara de bits digitales en los bytes 14 y 15. Cada valor leído es transformado a su forma binaria y almacenado en un Cluster, a continuación mediante la función de extracción se pueden acceder a cada elemento de la máscara de manera independiente.

Para la parte analógica nuevamente se realiza el mismo principio aunque un poco diferente.

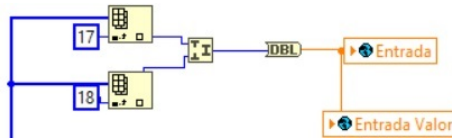


Figura 7.37: De izquierda a derecha, Función de Extracción por Índice, Unión de Bytes, Conversión a Double, Indicador.

En este caso, los valores analógicos se expresan en 16 bits, por lo tanto luego de leer dos bytes que corresponden a la máscara analógica se lo ingresa a la Función de Unión de Bytes para formar correctamente la magnitud, recordemos que el máximo valor posible es 1023 ya que el ADC es de 10 bits. Por último se utiliza una conversión a tipo Double.

Una vez leída la información se almacena en los indicadores para poder mostrarlas en la pantalla principal. Observar el color, ya que indica de que tipo es, por ejemplo el verde indica booleano, el azul un tipo numérico y el naranja un tipo numérico Double. Estas variables fueron creadas de manera global, esto nos permite escribirlas aquí, y leerlas en el Instrumento Virtual principal, hace que el código sea mas ordenado ya que no hay necesidad de crear variables en cada Instrumento Virtual, sobre todo si se utilizan las mismas, observar que tiene un icono con forma de planeta para diferenciarlas del resto e indicar su tipo. La siguiente figura contiene los indicadores utilizados para la lectura de un modulo remoto:

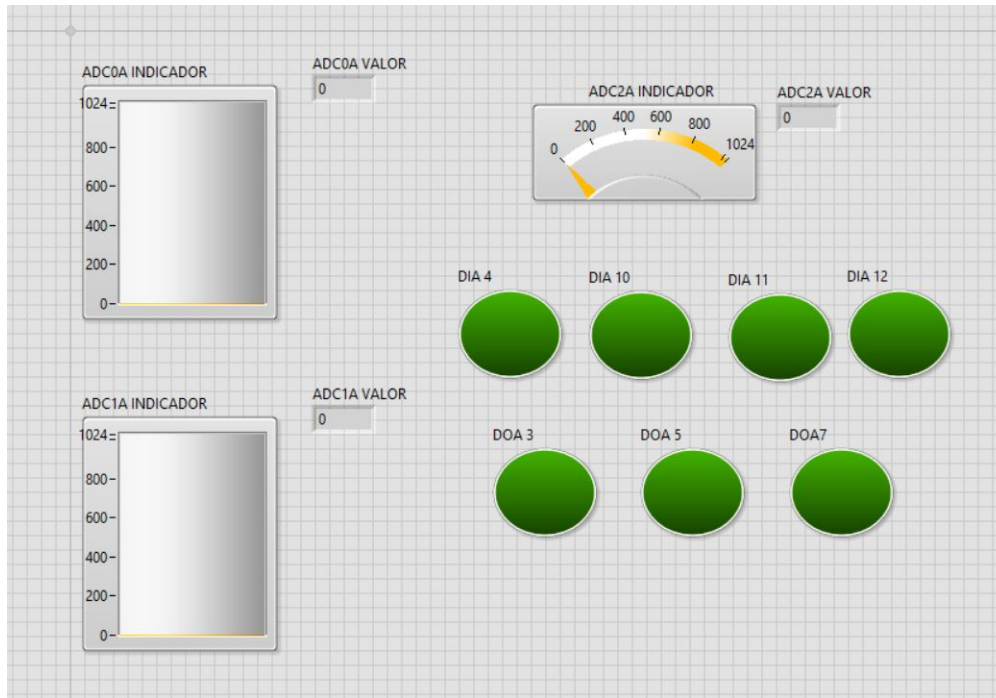


Figura 7.38: Variables Globales utilizadas en el proyecto.

Estas variables son creadas en un Instrumento Virtual nuevo pero con una particularidad, a pesar de observarse los indicadores en el **Panel Principal**, no tiene **Diagrama de Bloques**.

Cada una de ellas puede ser llamada desde cualquier punto del programa.

Las variables propuestas son las siguientes:

- ADC0A (gráfico y valor numérico)
- ADC1A (gráfico y valor numérico)
- ADC2A (gráfico y valor numérico)
- Digital Input DIA 4
- Digital Input DIA 10
- Digital Input DIA 11
- Digital Input DIA 12
- Digital Output DOA 3

- Digital Output DOA 5
- Digital Output DOA 7

Cada uno de los módulos remotos lleva esta configuración de pin, tres serán para adquisición de variables analógicas, cuatro para entradas de señales discretas y tres salidas para actuador. Los números se corresponden al número de PIN del PinOut de los XBee.

En la Figura 7.41 se ve un instrumento virtual propio de LabVIEW llamado **Write to Measurement File**. Este nos permite configurar distintas características a un archivo que deseemos crear:

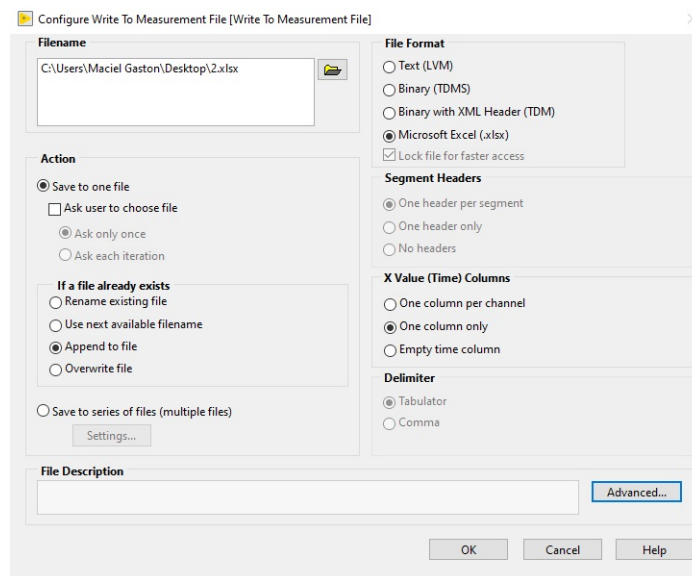


Figura 7.39: Configuración de VI para escritura.

Aquí podemos configurar la ubicación del archivo y de qué manera procede al guardar. Tenemos la opción de guardar en varios archivos en el caso de este proyecto se utiliza un archivo para cada dispositivo. Al escribir los datos también se puede agregar la fecha y hora en el que fue leído y guardado.

Paralelamente cuando se guardan los datos en sus respectivas variables globales, podemos ver que se envían a una herramienta que combina todas las señales en una sola.

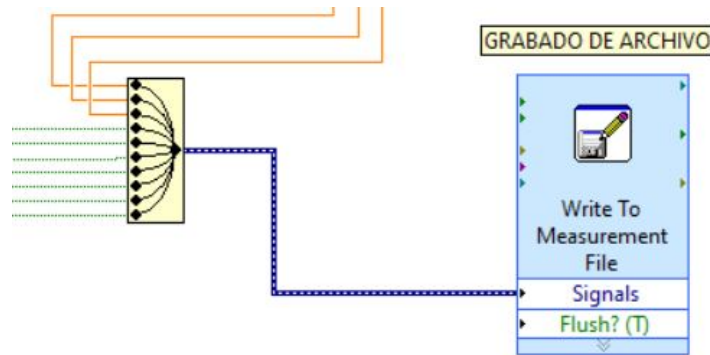


Figura 7.40: A la izquierda se observa la Herramienta de combinación de señales y a la derecha el VI para crear archivos EXcel.

Una vez combinadas las 10 variables, se ingresan al bloque de guardado, este las almacena con fecha y hora en el archivo creado cada vez que un frame valido ingrese en el VI Parsing.

Finalmente para terminar de crear este **Instrumento Virtual** se le proporciona un ícono y se adicionan los puertos de entrada de datos para que reciba la información desde el exterior.



Figura 7.41: Ícono virtual con puertos de entrada.

Aparenta no tener salida por lo que se ve en la Figura 7.41 sin embargo no es necesario ya que los datos son escritos en variables globales.

7.2.5. Generador de Comando

Una de las características a desarrollar desde el comienzo fue la posibilidad de enviar comandos de forma remota ya que de esa manera es posible controlar las salidas digitales del módulo XBee remoto o cambiar alguna configuración, para ello se utiliza el frame **0x17 - Envío de Comando Remoto**. El objetivo de este bloque

es crear el frame necesario en base a lo que el usuario desea, será necesario indicar el tipo de trama y agregar los distintos campos que la componen.

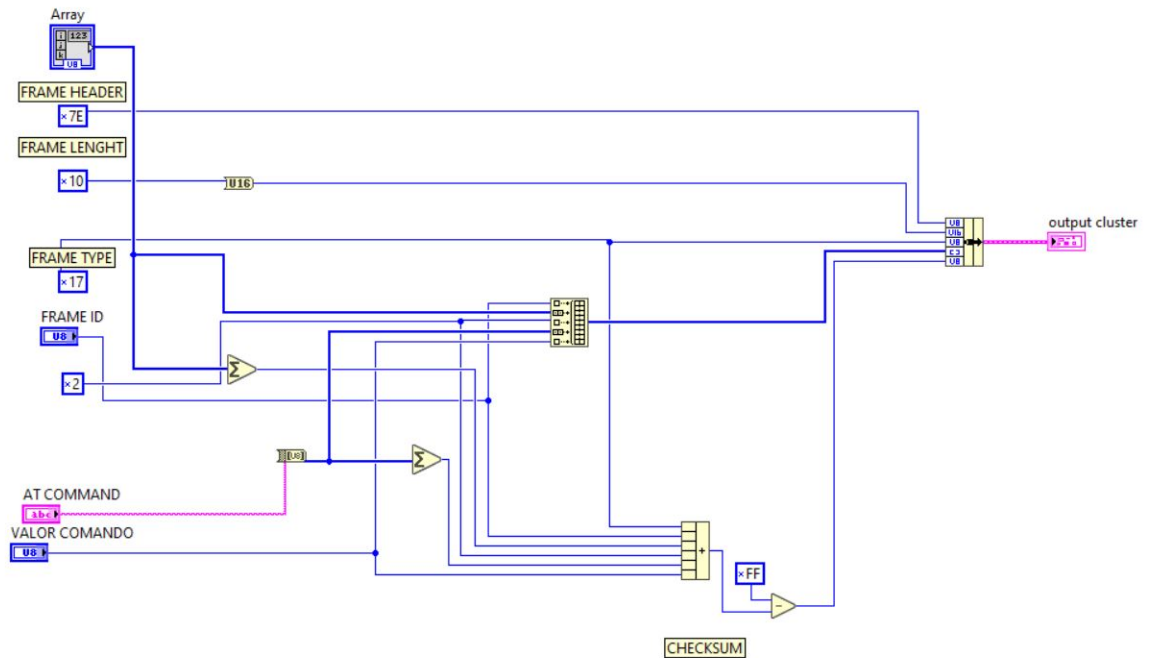


Figura 7.42: Instrumento Virtual para generar estructura del frame de comando.

En la Figura 7.42 se observa el diagrama de bloques del **Instrumento Virtual** para generar los campos, a la izquierda se encuentran cada uno de ellos.

En la parte superior hay un ejemplo del frame para el envío de comando remoto para modificar el estado de un PIN, al analizarlo se ven distintas características:

- Byte de Start
- Largo de Trama
- Tipo de Trama
- Dirección del XBee Remoto
- ID de la Comunicación
- Campo para aplicar los cambios inmediatamente

- Comando
- Valor del Comando
- Valor Checksum

El objetivo es generar cada uno de ellos empleando las siguientes herramientas:

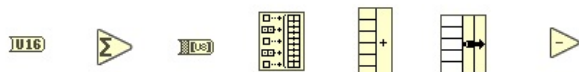


Figura 7.43: De izquierda a derecha, Convertir Número a Variable de 2 Bytes, Suma de los Elementos de un Array, Conversión de String a Byte, Construir Array, Suma Múltiple, Función de Formación de Cluster(Bundle) y la Función de Resta.

Analizando la situación se sabe que cualquier tipo de frame comienza con el hexadecimal **0x7E**, también que su longitud siempre sera constante para modificar el estado de un pin, ya que solamente se cambian los valores pero no la longitud final, por lo tanto estos serán constantes así como también el tipo de frame **0x17** y el campo de aplicación de cambios.

Por lo tanto se crean las siguientes **valores constantes**:

- **0x7E**:byte de inicio
- **0x10**:longitud de trama
- **0x17**:tipo de trama
- **0x02**:aplicar cambios

Mientras que los demás campos son variables que cambian dependiendo del comando y del módulo, por ejemplo si se utiliza solo un modulo remoto la dirección de destino del mensaje es constante, sin embargo para hacer escalable el código se programa de manera que puede ser modificado en cualquier momento para poder enviar comandos a distintos puntos de la red. El ID de la trama también puede variar y por supuesto el campo de comando y el valor del mismo, para que el usuario pueda

modificarlos estos se crean como **Controles**.

Observar la diferencia de íconos dejando claro cuales son los valores constantes y cuales son los controles(Variables). Las líneas poseen un color azul haciendo referencia a valores numéricos y el comando AT un color rosa, que indica que se trabaja con String.

Con estas consideraciones se procedió a generar el proceso para este **Instrumento Virtual**, todos los campos deben ser almacenados en un array para poder acceder a ellos de forma independiente, se crea un Cluster de almacenamiento como único array que contiene toda la información del frame.

Finalmente se suman los campos que corresponden y se restan a **0xFF** para obtener el Checksum correcto a enviar, esta última operación es vital para el funcionamiento debido a que ante un valor erróneo el módulo remoto descartará el frame al realizar la comprobación.

Se agrega un ícono para el nuevo bloque y se les asignan sus puertos.



Figura 7.44: Ícono de instrumento virtual para la creación de frames con sus puertos de entrada y salida.

Como se ha dicho, desde el exterior recibirá la información ingresada por el usuario estas serán las entradas y como salida el Cluster con la información creada.

7.2.6. Casting

Luego del proceso en el cual se crea el frame a transmitir, debe ser reconvertido para ser interpretado y enviado hacia el puerto serie donde se encuentra el módulo Maestro. Para esto se realiza un casting para llevar toda la información del Cluster de datos al formato String.

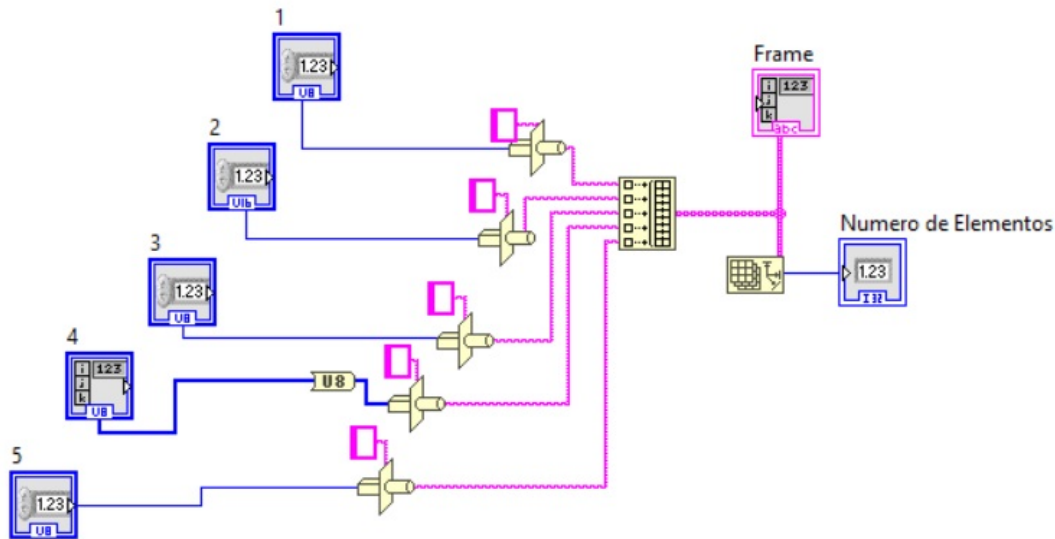


Figura 7.45: Instrumento virtual para realiza el casting.

Este **Instrumento Virtual** recibe los cinco datos y mediante la herramienta de casting los transforma a String. El campo de array al tener más de un byte dentro, se lo trata con la herramienta de conversión a byte para poder convertir uno a uno a string.



Figura 7.46: De izquierda a derecha, Función de Casteo, Función de Conversión a Byte, Constructor de Array y Lectura de Tamaño de Array.

Una vez convertidas todas son almacenadas en un nuevo array ordenados correctamente, este será entregado al sistema para ser enviado, en la Figura 7.45 puede verse la función de Lectura de Tamaño lo cual entrega la longitud del array. Esto es de suma utilidad ya que permite entregarle al puerto serie la cantidad específica de datos a enviar.

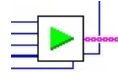


Figura 7.47: Ícono del Parsing con puertos correspondientes.

Como puertos de entrada se encuentran los datos y como salidas el string a enviar y el tamaño del mismo.

7.2.7. Detector de Alarma

Uno de los puntos principales es la generación de alarmas antes posibles anomalías en el funcionamiento, aunque antes de generarlas deben detectarse. Este Instrumento Virtual como su nombre indica fue creado para dicho propósito, veamos primero su estructura:

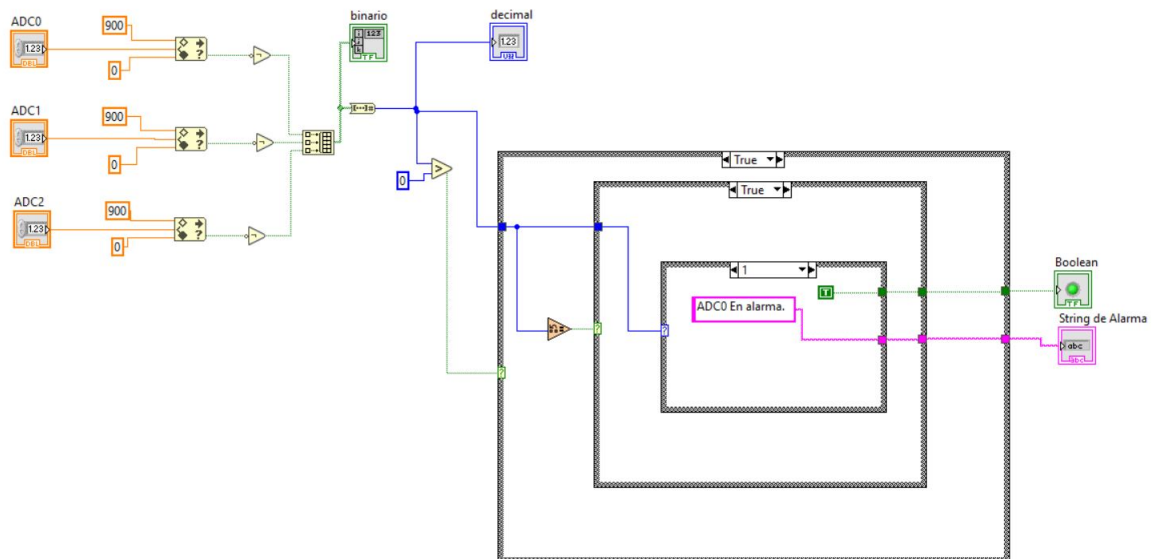


Figura 7.48: Estructura para la detección de magnitud fuera de rango.

A la izquierda de la Figura 7.48 se encuentran las entradas ADC (que serán proporcionadas externamente a este VI desde el programa principal), podemos ver un

nuevo elemento de LabVIEW llamado Función de Rango, esta herramienta nos permite ingresar la variable que deseamos monitorear y seleccionar un rango numérico (en este caso 0 para mínimo y 900 para máximo) para controlar si la magnitud ingresada está dentro o fuera del rango acotado, indicando la situación mediante una señal booleana en su salida. Cuando el valor esta dentro del rango la salida es TRUE, sin embargo aquí utilizamos la lógica inversa, siendo TRUE solamente cuando supera el valor máximo o mínimo, por eso se han agregado compuertas NOT en las tres salidas de las herramientas, con ellas creamos un array de bits para formar un número binario con estas señales, como hay tres es posible formar 8 posibilidades:

Salida	ADC2	ADC1	ADC0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Cuadro 7.2: Tabla de estados de alarma.

Como vemos, cada uno de los estados anteriores se corresponde a una situación en particular, utilizamos un conversor para cambiar la forma binaria a decimal. Dicho esto, podemos decir que estamos en condiciones de generar la señal de alarma ya que conocemos cada una de las transiciones posibles. En la Figura 7.48 a la derecha pueden verse las estructuras Case que modelan el comportamiento que deseamos. En primer Case es para ver si hay presencia alguna de alarma, cuando las magnitudes son normales la transición 0 es la que estará activa(sin alarma) de manera continua, como no queremos realizar ninguna acción cuando estemos en este estado evaluamos si el número en concreto de los detectores es mayor a 0, es decir, cuando se detecta alguna alarma. Si no hay presencia alguna de estas simplemente el bloque se mantiene inactivo ya que el número no es mayor a 0, es 0.

Ahora, cuando alguna de las variables se sale del rango especulado, a la salida de

los detectores podemos tener 1,2,..7, por lo tanto ingresaremos dentro del primer Case ya que todas son mayores a 0, en ese momento nos encontramos con el segundo Case, este se encarga de evaluar si la alarma en cuestión sigue activa, esta estructura fue necesaria debido a que ante la presencia de alarma, la señal quedaba activa continuamente, esta será enviada al bloque del correo electrónico para dar los avisos, esta situación con la señal de salida provocaba el envío incontrolable de mails y tildaba todo el programa. Por lo tanto antes de ingresar al segundo Case, se evalúa con la herramienta de detección de cambios si el número de alarma es el mismo que en el ciclo anterior, esta herramienta devuelve una señal booleana TRUE si hay cambio y FALSE si sigue siendo el mismo. Dicho esto cuando se detecta una alarma ingresa en el primer case y en el segundo ya que efectivamente hubo un cambio en el número de alarma, por lo tanto ingresa en el último Case que activa la salida. Al siguiente ciclo si la misma alarma sigue activa el detector de cambios veraél mismo número nuevamente por lo que no nos dejará ingresar al tercer Case y desactivará la señal de alarma. La tercer estructura Case tiene todos los casos posibles, aquí los condicionales son números y no estados booleanos, allí tenemos las etiquetas para enviar por Mail indicando que alarma esta activa, y un valor TRUE continuo para activar la señal de salida dependiendo que número de alarma esta activo:

- 0: sin alarma
- 1: ADC0 en alarma
- 2: ADC1 en alarma
- 3: ADC0 y ADC1 en alarma
- 4: ADC2 en alarma
- 5: ADC0 y ADC2 en alarma
- 6: ADC1 y ADC2 en alarma
- 7: ADC0, ADC1 y ADC2 en alarma

Con esto nos aseguramos que ante la presencia alguna de alarmas, se genere la señal por única vez y que indique mediante Strings cual de ellas esta activa. Los indicadores binario y decimal que se ven en el bloque principal simplemente son agregados para ayudar al desarrollador a comprobar el funcionamiento el bloque, las

salidas de este Instrumento Virtual son la señal de alarma en cuestión y el String de Alarma, mientras que sus entradas serán por supuesto, las magnitudes ADC extraídas de los frames.



Figura 7.49: Icono correspondiente al detector, con sus tres entradas y dos salidas.

Crear este bloque aparte nos da la ventaja de importarlo en otros proyectos así como también la posibilidad de cambiar los rangos de funcionamiento en las variables de interés desde aquí sin modificar el programa principal.

7.2.8. Envío de Correo Electrónico

Este **Instrumento Virtual** fue creado como una de las piezas principales a la hora del desarrollo de una aplicación orientada al control, es algo que pude ver en todos los software's a lo largo de estos años, útil y versátil a la hora de utilizarse en distintas situaciones. Labview ofrece herramientas para utilizar correo electrónico de ser necesario mediante la paleta de SMTP(Simple Mail Transfer Protocol), con estas diferentes funciones es posible desarrollar un bloque para el envío de correos. En este proyecto se utiliza para indicar al responsable o a un conjunto de personas si es necesario, la presencia de algún tipo de alarma que seleccione el usuario, por lo tanto esto debe configurarse para cada aplicación específica.

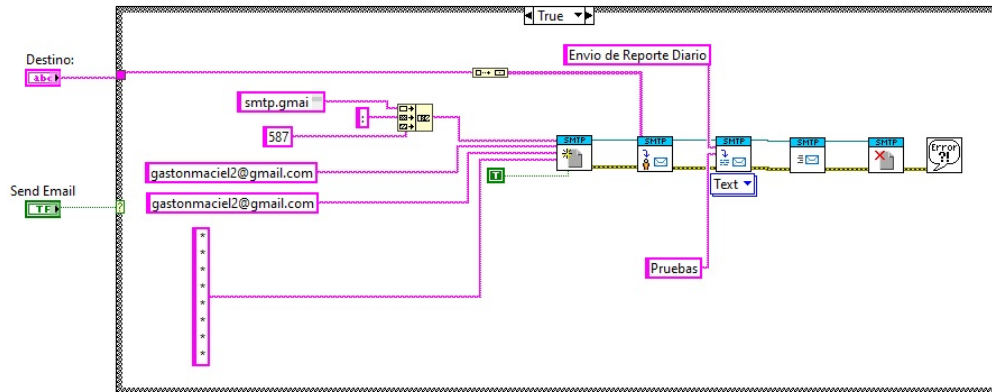


Figura 7.50: Estructura para el envío automático de correo electrónico.

Sin embargo también puede utilizarse para el envío de por ejemplo el reporte diario de datos, simplemente adjuntando la ubicación del archivo y agregándolo en la línea de configuración.

En la Figura 7.50 se observa como previamente deben establecerse ciertos parámetros para lograr el objetivo final, si bien Labview nos da facilidades hay que aprender y familiarizarse con ellas para sacar mayor provecho y no cometer errores o al menos minimizarlos al máximo.



Figura 7.51: De izquierda a derecha, Función de Concatenación de Strings, Constructor de Array, Configuración SMTP, Configuración de Destino, Herramienta de Texto Adjunto, Función de Envío de Correo, Cierre de Herramientas de Correo.

El funcionamiento es simple, todo esta dentro de una Estructura Case, cuando éste detecte un valor verdadero en su señal de entrada(que viene del detector), dará pie a la configuración y creación del correo. Por lo tanto esta señal booleana será uno de los puertos de entrada al bloque. También puede verse una señal de String fuera del recuadro, esta es de tipo control y es la que utiliza el usuario para indicar

el destinatario del mail. Particularmente se tomo como una señal de control para poder cambiar los destinos desde la pantalla principal pero podría ser un valor constante si el destino es siempre el mismo, en ese caso eliminaríamos un puerto de este **Instrumento Virtual** cambiando el tipo de señal, de control a constante.

Para iniciar un proceso SMTP en Labview primero se configura el servidor utilizado ingresando los siguientes datos:

- Host o IP del servidor SMTP
- Dirección y contraseña del emisor
- Habilitar seguridad de capa de transporte TLS

Aquí se utiliza el servicio de mensajería de Google, en la imagen se aprecia mi dirección de correo personal para realizar las pruebas de funcionamiento, al utilizar este host debe habilitarse el TLS ya que es una de las exigencias impuestas por el servicio.

Para crear el parámetro de Host se crean constantes y se las concatenan para formar "smtp.gmail.com:587" el ultimo termino se refiere al puerto a utilizar recomendado por Google. Pueden verse ademas, los demás parámetros constantes que ingresan a la función, la dirección de emisor y la señal booleana verdadera en el TLS.

Una vez lista la configuración, se ingresa en el bloque de dirección de destino el control del usuario antes mencionado y finalmente se ingresa el asunto del correo y el texto que habrá en el, aquí el asunto es "Envío de reporte diario" y solamente contiene la palabra "Pruebas". Si el usuario desea enviar un archivo, se deberá utilizar la herramienta de adjuntos de la paleta de funciones SMTP y se conectara luego de este último bloque. Finalmente el proceso debe ser finalizado con la función de cierre para este tipo de procesos, de lo contrario el programa quedará en un ciclo infinito provocando un mal funcionamiento de la aplicación.



Figura 7.52: Ícono para el envío de correo electrónico.

La Figura 7.52 muestra el ícono con sus dos puertos de entrada, la señal booleana y el control de destino de correo.

7.2.9. Pantalla Principal

Finalmente habiendo visto los Sub-VI anteriores, es posible explicar el funcionamiento del **Instrumento Virtual Principal** del sistema. Este es el corazón de la interfaz, contiene todos los bloques anteriores y realiza la conexión entre ellos además de tomar el control total de como aplicarlos. A continuación se observa el diagrama de bloques que gobierna el comportamiento:

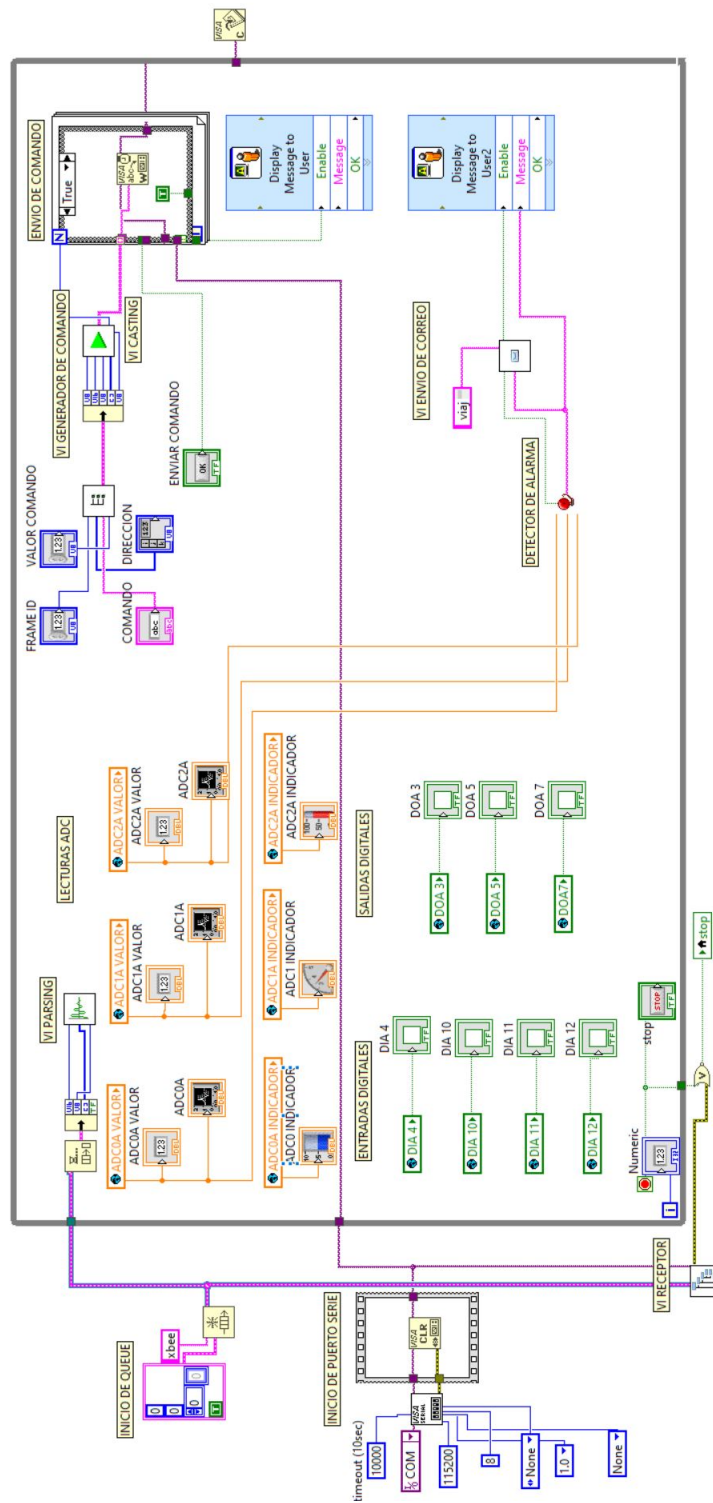


Figura 7.53: Diagrama de bloques principal del sistema.

Antes de comenzar a entender como funciona el bloque principal debemos tener algunas consideración que ayudarán a comprender mejor el diagrama, observar que dentro del esquemático hay etiquetas que clasifican cada sector de funcionamiento:

- **VI RECEPTOR**
- **INICIO DE PUERTO SERIE**
- **INICIO DE QUEUE**
- **VI PARSING**
- **LECTURAS ADC**
- **ENTRADAS DIGITALES**
- **SALIDAS DIGITALES**
- **DETECTOR DE ALARMA**
- **VI ENVÍO DE CORREO**
- **ENVÍO DE COMANDO**
- **VI GENERADOR DE COMANDO**
- **VI CASTING**

En ese orden se irán relacionando a medida que avance este texto.

Para que la interfaz comience a recibir hemos visto la función del **VI RECEPTOR**, para poder comenzar a trabajar necesita que le proporcionen el puerto, para ello se inicia uno y se le da las características deseadas.

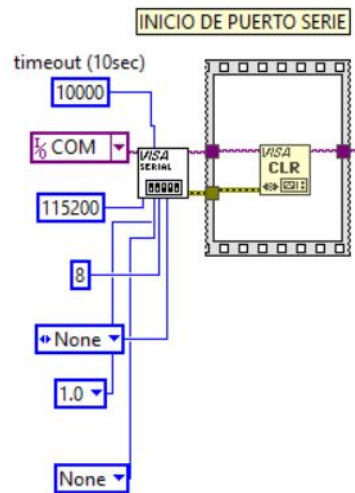


Figura 7.54: Configuración de puerto serie.

En la sección de **INICIO DE PUERTO SERIE** se encuentran dos herramientas nuevas, a la izquierda la herramienta de Apertura de Puerto para iniciarlo y configurar los baudios, cantidad de bits, bit de parada, paridad y control de flujo. En este caso todos los parámetros son constantes, aunque fácilmente pueden cambiarse a señales de control para ingresarlos a mano cada vez que el usuario lo desee.

A la derecha se encuentra la Herramienta de Vaciado de Buffer, como su nombre lo indica nos asegura que el puerto este vacío para trabajar. Pueden verse que la línea de conexión va hacia dos lugares, eso es por que una de las líneas alimenta al bloque de recepción y otro al envío de comando donde también se necesita utilizar el puerto serie.

Una vez listo el medio físico, ahora se debe dar formato a los datos. Recordemos que lo que entrega el **VI RECEPTOR** es un **Queue**, debemos iniciarlo es esta pantalla principal para mantener el formato de lo contrario no podemos utilizar lo que recibimos, puede verse su inicio en la sección **INICIO DE QUEUE**.

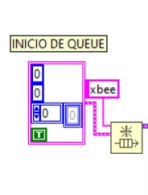


Figura 7.55: Inicialización de Colas para el uso de datos.

Cuando se crea esta herramienta debe aclararse el tipo de datos que conlleva, por eso a la izquierda puede verse el formato del Cluster que manejamos en toda la interfaz: dato de longitud, de tipo de trama, array con datos con la dirección remota y el valor booleano de Checksum.

Con esto nos aseguramos la recepción de datos para posterior análisis. Como queremos que la lectura se realice de manera continua todos los procesos están dentro una Estructura While.

Lo primero que sucede al ingresar un frame, es el análisis para poder extraer los valores analógicos y digitales así como también el guardado de los mismos.

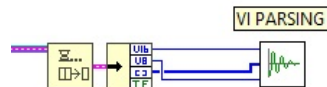


Figura 7.56: Ingreso de datos a analizar.

Para ello extraemos primero el Cluster del **Queue** y utilizamos la herramienta **Unbundle** para obtener los campos que contiene de manera separada, estos datos son lo que ingresan al **VI PARSING** para ser interpretados y escritos en las variables globales las cuales deben ser vueltas a llamar en la pantalla principal pero en este caso para leerlas y brindarles un medio al usuario para observarlas, eso se ve en el sector de **LECTURAS ADC**, **ENTRADAS DIGITALES** y **SALIDAS DIGITALES**.

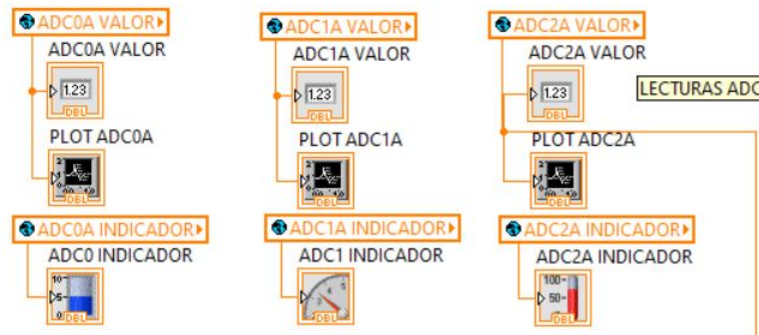


Figura 7.57: Valores de lectura analógicos.

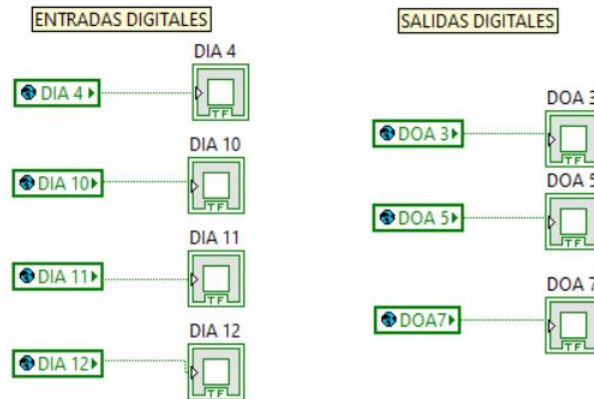


Figura 7.58: Valores de lectura digitales.

Aquí se empieza a ver un poco mas claro el funcionamiento de las variables globales, cuando realizamos estas llamadas para leer su valor las mostramos en nuevos indicadores que se crean únicamente para la pantalla principal, en este caso tenemos los siguientes indicadores:

- DIA 4 - DIA 10 - DIA 11 - DIA 12
- DOA 3 - DOA 5 - DOA 7
- ADC0A VALOR - ADC1A VALOR - ADC2A VALOR
- PLOT ADC0A - PLOT ADC1A - PLOT ADC2A

Con ellos mostramos en la pantalla que ve el usuario el estado de lo que estamos sensando:

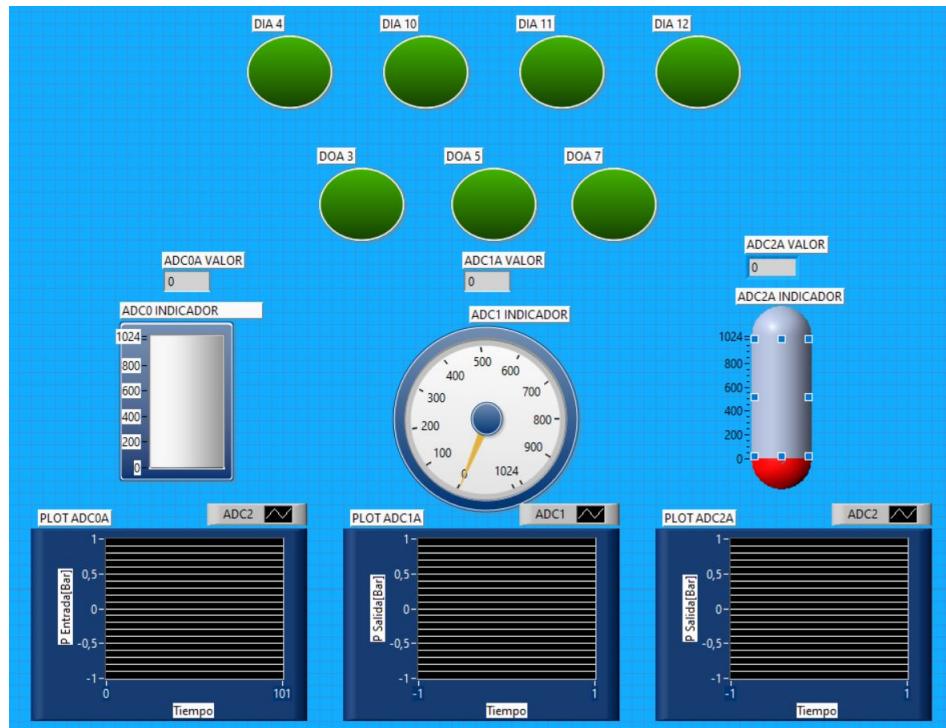


Figura 7.59: Panel Frontal de variables.

Aún siendo indicadores los colores vuelven a hacer las mismas referencias antes vistas en cuanto al formato de datos, los verdes siguen siendo booleanos, y los naranjas valores double numéricos. Allí se ven los estados digitales y analógicos. Para la selección de los indicadores se tomaron tres tipos diferentes como ejemplo, uno en forma de tanque, otro en forma de indicador de aguja y un termómetro con el fin de mostrar que hay muchísimas más, por ejemplo válvulas, motores, ventiladores y todo tipo de herramienta industrial. También se utiliza una paleta muy útil llamada **Gráficos** para mostrar los datos a medida que van ingresando al sistema, además permite configurar ajustes de escala y ejes entre otras. Una de las preguntas que pueden surgir es por que no realizar el parsing en la pantalla principal, esto podría hacerse pero es más conveniente hacerlo aparte ya que si debemos realizar alguna operación sobre los datos solo hay que modificar al **VI PARSING** y no la panta-

lla principal. Además este desarrollo fue hecho con la filosofía que recomiendan los ingenieros de National Instruments, los procesos del VI principal deben entrar en la pantalla sin sobresalir de ella. Como dato adicional para el lector, LabVIEW no permite acercar o alejar la pantalla, los tamaños de las herramientas y bloques son constantes, por lo tanto el diseño debe caber en la pantalla sin tener que deslizarse hacia los costados, si incorporase todos los procesos juntos en la pantalla principal, sin duda no alcanzaría para que todos entren sin necesidad de utilizar el espacio adicional no recomendable.

Ya se ha visto el modo en que se ven los datos, ahora se procede a explicar el envío de alarma.

Vemos en la pantalla principal el sector **DETECTOR DE ALARMA**, a este Instrumento Virtual se conectan las tres variables de ADC, los procesa y retorna una señal que indica la presencia de alarma y en caso afirmativo entrega un String que indica cual de ellas o que conjunto esta fuera de rango. Estas son las que controlan el envío de correo electrónico.

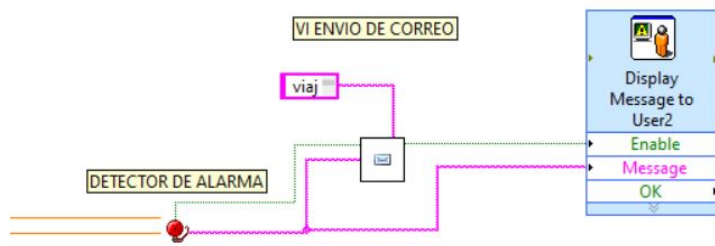


Figura 7.60: Proceso de envío de correo por alarma.

Se le debe proveer la dirección de destino desde aquí, para ello se crea un control para ingreso de String llamado DESTINATARIO y se conecta. También se agrega mediante un VI de Dialogo estándar de LabVIEW un cartel indicando el envío exitoso. El control para el destino se verá de la siguiente manera en la pantalla de Panel Frontal:

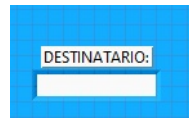


Figura 7.61: Ingreso de destinatario.

Con esto finaliza el proceso de detección de alarma y envío de correo. Por último queda analizar el **ENVÍO DE COMANDO** para el control remoto. Como deben ingresarse los distintos campos que componen el frame, se crearon controles para ser ingresados por el usuario, estos son:

- DIRECCIÓN
- VALOR COMANDO
- COMANDO
- FRAME ID
- OK

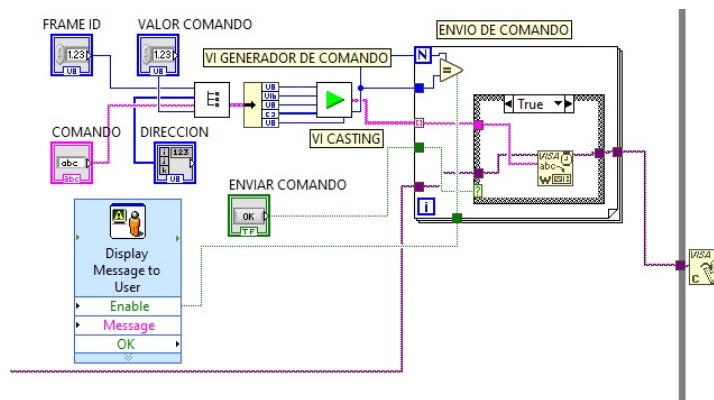


Figura 7.62: Proceso de envío de comando.

De esta forma el **VI GENERADOR DE COMANDO** hace su trabajo y entrega el Cluster de datos armado (recordemos que Cluster es un array pero con datos de distinto tipo). Este entra directamente al **VI PARSING** que les cambia el formato e indica la cantidad de caracteres a enviar. En este punto el frame ya está listo para ser enviado, para eso se utiliza una Estructura For con N iteraciones

que es este caso es la cantidad de caracteres que tiene la trama. Dentro de este se encuentra la herramienta de Escribir en Puerto, con ella enviamos el frame hacia nuestro módulo Maestro que hará el trabajo de envío. Cuando el usuario use el botón de ENVIAR COMANDO habilitara la escritura y aparecerá un indicador si el envío fue exitoso.

Con esto se han completado los diferentes procesos que forman a la aplicación.

7.3. Configuración de Red

Como ya sabemos los dispositivos se comunicarán entre ellos por medio de radio-frecuencia y no podrán gestionar los datos recibidos, de eso se encargará el programa creado en LabVIEW tomando los frames a través de la interfaz serie que posee el XBee.

Antes de comenzar repasemos brevemente algunos conceptos mencionados en capítulos anteriores que intervienen directamente en la formación de la red:

- *dirección 64-bits*: esta es única para cada dispositivo, también llamada dirección MAC el valor esta contenido en los parámetros SH y SL de los dispositivos. Por lo general SH tiene un valor 0013A200, este prefijo identifica a los módulos de Digi. Por otro lado, la dirección de 64 bits 000000000000FFFF se reserva para enviar un mensaje de tipo broadcast.
- *dirección 16-bits*: cuando un dispositivo se una a una red existente se le asigna una dirección aleatoria de 16 bits, mas conocida como dirección de red. Esta puede cambiar si ocurre un problema de direccionamiento o si el dispositivo abandona y vuelve a unirse a la red. Puede verse el valor de la dirección en el parámetro MY. El valor 0000 se reserva al coordinador, mientras que un valor FFFE indica que el dispositivo aún no se ha unido a la red.
- *identificador de nodo*: esta cadena de caracteres especifica el nombre que se le da al dispositivo, de esta forma se busca facilitar la interfaz con el usuario visualmente, para leer la identificación debe consultarse el parámetro NI.
- *identificación PAN*: cada red debe tener una ID PAN que sera común a todos los dispositivos que componen la red y puede ser leído con el parámetro ID.

- *canales*: si desea conocer el canal donde funciona el dispositivo debe consultarse el parámetro CH. Si se desea establecer un canal operativo específico puede realizarse mediante el parámetro SC, obliga al coordinador a trabajar sobre ese canal así como también a routers y end devices. Si el canal no tiene importancia puede usarse el SC para utilizar múltiples canales(0xFFFF), también es posible configurar el tiempo de escaneo de canales mediante el parámetro SD, responde a la siguiente fórmula exponencial: $SC \cdot (2^{SD}) \cdot 15,36\text{ms}$ que determina el tiempo.
- *node join time*: para los dispositivos que permiten la asociación, es posible imponer un tiempo máximo para que los demás nodos puedan hacerlo a través del parámetro NJ. Un valor 0xFF indica que siempre se permitirá la asociación.
- *zigBee stack profile*: perfil a utilizar por el XBee. Se configura por medio del parámetro ZS, puede tomar un perfil de red, ZigBee2006 o ZigBee PRO.
- *IO sampling rate*: los valores de I/O son transmitidos periódicamente cada vez que el tiempo establecido por el parámetro IR(en milisegundos) se concrete en caso de Router o cuando despierte para un End Device.
- *coordinador activo*: para indicar si un módulo es coordinador o no, se utiliza en el parámetro CE.
- *modo sleep*: el parámetro de sleep SM determina si el módulo es un Router o End Device. Pueden configurarse el tiempo que el dispositivo permanecerá dormido mediante SP y el tiempo que se mantiene activo luego de despertar por medio de ST. Los dispositivos Router y Coordinador deberán tener el mismo valor de SP para estar al tanto del tiempo que duermen sus hijos.

Los XBee utilizados en este proyecto se configuran para operar en modo API y se le debe asignar un rol dentro de la red los cuales ya conocemos. Por medio de los parámetros CE(Coordinator Enable) y SM(Sleep Mode) se les da la funcionalidad, veamos las combinaciones:

- *Coordinador*: estará activo todo el tiempo y debe asignarse el rol de coordinador, CE=1 y SM=0.
- *Router*: al igual que el coordinador no puede estar inactivo, aunque no cumple con las funciones de coordinador, CE=0 y SM=0.

- *End Device*: los dispositivos finales son los únicos habilitados para estar en estado de bajo consumo y no tienen todas las funcionalidades del coordinador, CE=0 y SM=4.

Dentro de la red, los nodos pueden comunicarse con múltiples dispositivos o con uno en particular:

- *Broadcast*: en este caso el mensaje se transmite a todos los nodos de la red, el coordinador y router que recibe el broadcast vuelve a retransmitirlo hasta tres veces. Si se utiliza la dirección de 64 bits(000000000000FFFF), la de 16 bits debe configurarse como desconocida(FFFE). Si se utiliza la dirección de 16 bits(FFFF) debe configurarse la de 64 bits como desconocida(FFFFFFFFFFFFFFFF).
- *Unicast*: este consiste en transmitir un mensaje hacia un nodo específico. Puede ser hacia un nodo vecino o mas lejano. Si se utiliza la dirección de 64 bits debe configurarse la dirección de 16 bits como desconocida(FFFE), para el caso de 16 bits debe configurarse la de 64 bits en desconocida(FFFFFFFFFFFFFFFF).

Escenarios a Utilizar

La aplicación XCTU brinda al usuario la posibilidad de configurar de manera mas rápida y sencilla los dispositivos. Como primer medida se carga el firmware correspondiente a cada uno de los dispositivos para darle su funcionalidad y set de instrucciones acorde conectando cada XBee a la PC con la placa de la Figura 7.19.

Product family	Function set	Firmware version
XB24-B XB24-SE XB24-ZB	ZigBee Coordinator API	21A7 (Newest)
	ZigBee Coordinator AT	21A0
	ZigBee End Device API	218C
	ZigBee End Device AT	2170
	ZigBee End Device Analog IO	2164
	ZigBee End Device Digital IO	2141
	ZigBee End Device PH	2121
	XB24-B XB24-SE XB24-ZB	ZigBee Router API
ZigBee Router AT		23A0
ZigBee Router AT (WALL RT)		238C
ZigBee Router Sensor		2370
ZigBee Router/End Device Analog IO		2364
ZigBee Router/End Device Digital IO		2341
ZigBee Router/End Device Sensor		2321
XB24-B XB24-SE XB24-ZB		End Device - LTH
	ZigBee Coordinator API	29A0
	ZigBee Coordinator AT	298C
	ZigBee End Device API	2970
	ZigBee End Device AT	2964
	ZigBee End Device Analog IO	2941
	ZigBee End Device Digital IO	2921

Figura 7.63: Firmware API a utilizar en el proyecto para coordinador, router y end device.

En la Figura 7.63 vemos el Firmware a cargar en los dispositivos, para Coordinador, Router y End Device. A partir del mismo, se configuran los parámetros adicionales para formar el tipo de red que deseamos sabiendo que el modo API ya está dentro de los módulos y los roles que tendrán también son configurados automáticamente al cargar el FW, esto último hace referencia al parámetro CE. No es necesario especificar las direcciones de destino (DH y DL) ya que esta modalidad lleva más información que el mensaje en sí, como la dirección de origen de la transmisión, la intensidad de señal y el valor de verificación, todo ello se encuentra dentro del paquete que se transmite, particularmente la dirección 0 es la que se asigna al Coordinador (donde irán todos los frames en nuestro caso).

Antes de comenzar con la red primero se establecen las siguientes consideraciones funcionales en los pines de los End Devices, todos los dispositivos de este tipo en las redes a utilizar tienen los puertos configurados de la misma manera para la toma de muestras analógicas y digitales provenientes de los sensores:

- DO/D1/D2 - ADC
- D3/D5/D7 - DIGITAL OUT
- D4/P0/P1/P2 - DIGITAL INPUT

Con estas características de entrada y salida se crean distintos escenarios posibles para obtener los resultados del proyecto, los End Devices en particular fueron configurados para no dejar de transmitir, es decir no estaremos trabajando en bajo consumo. Dicho tiempo y características de la red se seteados en los parámetros descritos en el comienzo de esta sección, a partir de ellos se exponen a continuación las tablas con los valores seleccionados para cada uno de los dispositivos:

- *RED A - Comunicación Coordinador y End Device:* configuración más básica, el End Device enviará periódicamente al Coordinador las muestras analógicas y estados digitales.

Parámetro	XBeeM	XBee A
ID(PAN ID)	12	12
CE(Coordinator Enable)	1	0
NI(Node Identifier)	COORD	ED1
NJ(Node Join Time)	FF	-
SM(Sleep Mode)	0	4
SP(Cyclic Sleep Period)	20	20
ST(Time Before Sleep)	-	FFFE
SC(Scan Channel)	FFFF	FFFF
SD(Scan Duration)	3	3(default), .
ZS(ZigBee Stack Profile)	0	0
IR	-	BB8(3 s)

Cuadro 7.3: Configuración de parámetros de cada módulo.

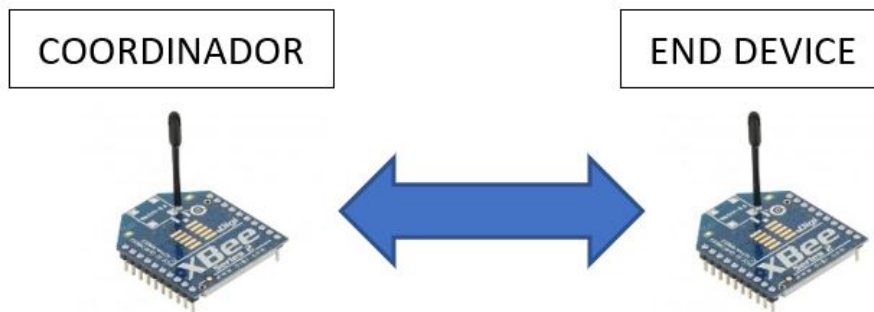


Figura 7.64: Comunicación Coordinador y End Device.

- *RED B - Comunicación Coordinador y Múltiples End Devices*: en este caso se dos nodos simples transmitiendo hacia el coordinador.

Parámetro	XBee M	XBee A	XBee B
ID(PAN ID)	12	12	12
CE(Coordinator Enable)	1	0	0
NI(Node Identifier)	COORD	ED1	ED2
NJ(Node Join Time)	FF	-	-
SM(Sleep Mode)	0	4	4
SP(Cyclic Sleep Period)	-	20	20
ST(Time Before Sleep)	-	FFFE	FFFE
SC(Scan Channel)	FFFF	FFFF	FFFF
SD(Scan Duration)	3	3	3
ZS(ZigBee Stack Profile)	0	0	0
IR	-	1388(5 s)	1388

Cuadro 7.4: Configuración de parámetros de red.

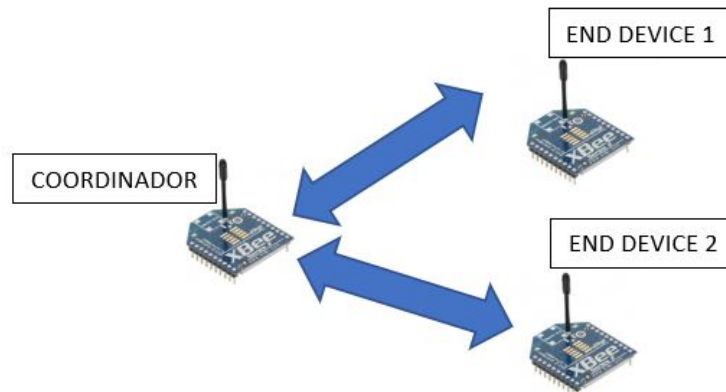


Figura 7.65: Comunicación Coordinador y Múltiples End Devices.

- *RED C - Comunicación Coordinador y Múltiples End Devices:* en este caso se tienen múltiples nodos simples transmitiendo hacia el coordinador.

Parámetro	XBee M	XBee A	XBee B	XBee C
ID(PAN ID)	12	12	12	12
CE(Coordinator Enable)	1	0	0	0
NI(Node Identifier)	COORD	ED1	ED2	ED3
NJ(Node Join Time)	FF	-	-	-
SM(Sleep Mode)	0	4	4	4
SP(Cyclic Sleep Period)	-	20	20	20
ST(Time Before Sleep)	-	FFFE	FFFE	FFFE
SC(Scan Channel)	FFFF	FFFF	FFFF	FFFF
SD(Scan Duration)	3	3	3	3
ZS(ZigBee Stack Profile)	0	0	0	0
IR	-	3A98	3A98	3A98

Cuadro 7.5: Configuración de parámetros de red.

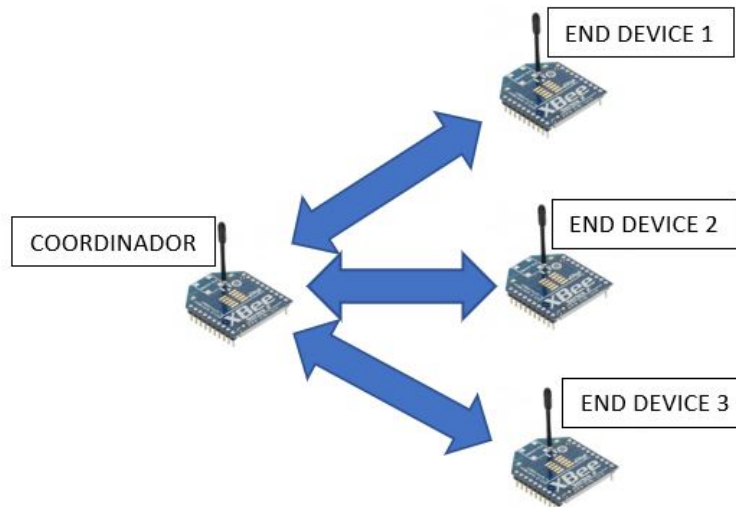


Figura 7.66: Comunicación Coordinador y Múltiples End Devices.

- *RED D - Red Mesh, Comunicación Coordinador, Múltiples Routers y End Devices*: configuración mas compleja y a la que apunta el proyecto.

Parámetro	XBee A	XBee B	XBee C	XBee D	XBee E
ID(PAN ID)	12	12	12	12	12
CE(Coordinator Enable)	1	0	0	0	0
NI(Node Identifier)	COORD	RTA	RTB	ED1	ED2
NJ(Node Join Time)	FF	FF	FF	-	-
SM(Sleep Mode)	0	0	0	4	4
SP(Cyclic Sleep Period)	384	384	384	384	384
ST(Time Before Sleep)	-	-	-	7D0	7D0
SC(Scan Channel)	FFFF	FFFF	FFFF	FFFF	FFFF
SD(Scan Duration)	3	3	3	3	3
ZS(ZigBee Stack Profile)	0	0	0	0	0
IR	-	-	-	3E8	3E8

Cuadro 7.6: Configuración de parámetros de red.

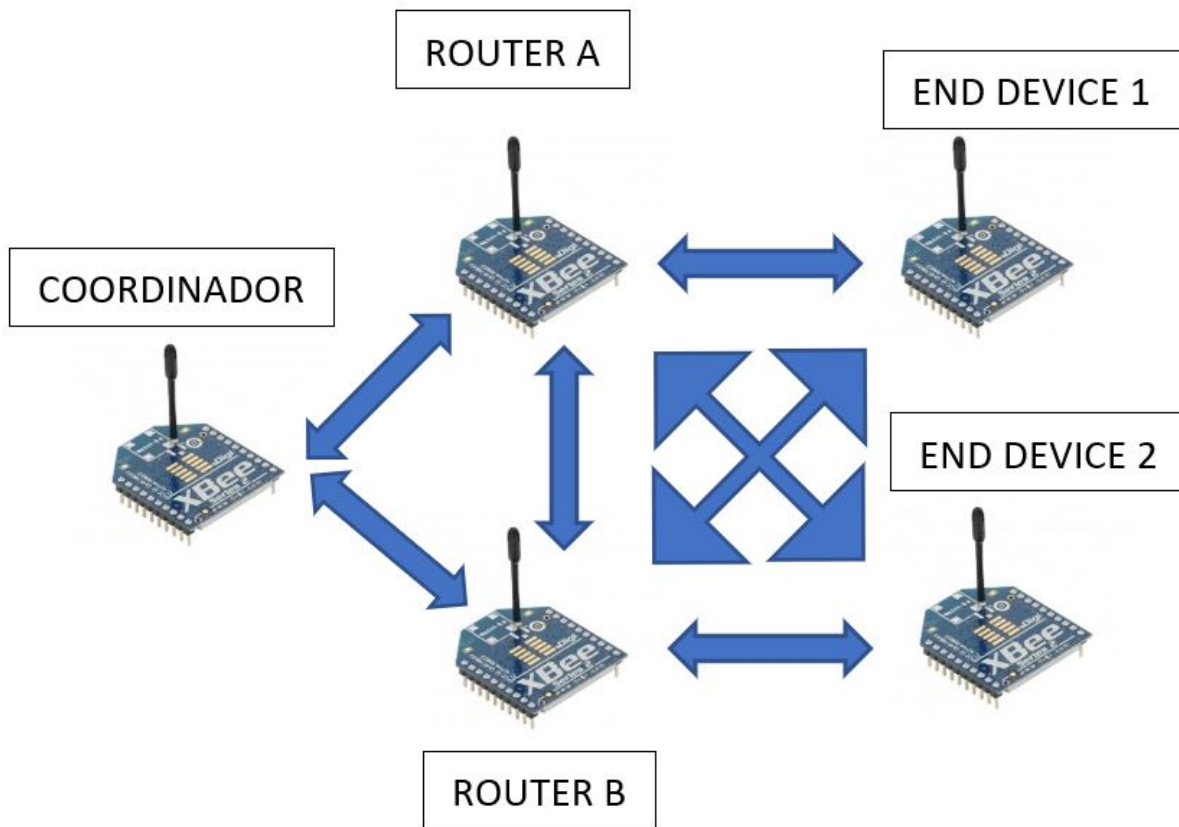


Figura 7.67: Red Mesh, comunicación Coordinador, Múltiples Routers y End Device.

Hasta ahora hemos repasado como se han configurado cada uno de los nodos de las redes, las primeras dos se utilizan principalmente para comprobar el funcionamiento del sistema creado en LabVIEW al recibir frames de un sólo nodo o de varios, la Red C, por otro lado, es la mas importante debido a que el proyecto esta pensado para ser utilizado en topología Mesh.

El Coordinador se encarga de crear la red, sabemos que un Router es capaz de tener un papel de Coordinador en una red 802.15.4 para un grupo reducido de RFD, aquí se encarga de transportar las tramas entre otros Routers dentro de la red para hacer llegar los mensajes a destino si es que el mensaje no es para si mismo, también deben estar atentos para almacenar los mensajes pendientes para sus nodos “hijos”, por lo que el Coordinador es un Router con características especiales, como el inicio y

mantención de la red que repasaremos a continuación. El último eslabón hace referencia a los dispositivos End Devices, estarán en campo tomando los valores de los sensores y enviándolos a destino.

En topología Mesh, los dispositivos mas completos es decir Coordinador y Routers se encargarán de generar y descubrir las rutas de destino para los mensajes a enviar. Todo comienza en el Coordinador, crear la red ZigBee implica crear primero una red 802.15.4, al trabajar en 2.4 GHz seleccionará uno de los dieciséis canales disponibles y todo el tráfico sucederá por el canal elegido, esto hace referencia a los procesos vistos en la sección 4.2.1 donde se emplean los métodos de medición de energía en todos los canales para finalmente escoger el mas silencioso.

Seleccionado el canal operativo comienza el proceso de búsqueda activa donde emite tramas 802.15.4 Beacon Request formando una lista de PAN ID encontrados para no escoger un ID que este en uso, en este caso se ha establecido una PAN ID=12 desde el programa XCTU directamente por el usuario, dicho esto se auto asigna la dirección de red 0x0000 y aguarda, por este motivo no hemos especificado las direcciones de destino en los End Devices (tienen un valor de 0) debido a que por el Coordinador toma un valor 0 al iniciar la red. se ha configurado el parámetro NJ en el Coordinador y Routers para que acepten la asociación siempre.

En este momento la red ha sido conformada, por lo que ahora deben unirse los demás participantes, quien quiera participar de ella debe pedir permiso a un miembro de la red, creando una relación “padre-hijo”. En la sección 4.2.3 se detallan los procesos para unirse a la red ZigBee: asociación, retorno y declararse huérfano. Cuando la red se conforma con todos los participantes comienza el tráfico de frames que tanto nos interesa. Cuando un Router recibe un mensaje chequea si es para él, en caso de que no le corresponda comienza el proceso de Routing. Recordemos que poseen tablas internas de Vecinos y de Ruteo, la primera hace referencia a los nodos en contacto directo mientras que la segunda hace referencia a los posibles caminos para entregar mensajes a nodos lejanos, el camino que seguirá el mensaje se hará en función del menor costo asociado. Si el Router no encuentra el destino en sus tablas comienza el proceso cooperativo entre pares de Route Discovery alojando la información del proceso de búsqueda en la tabla de rutas en proceso, como mayor ventaja las redes ZigBee Mesh no necesitan supervisión adicional a la hora de realizar las transmisio-

nes, cuando se envía un mensaje los módulos escogerán la ruta optima a utilizar y en caso de que alguno de los nodos quede inutilizado, realizará los procesos Routing vistos(sección 4.2.3) para generar un camino alternativo de forma autónoma manteniendo la red operativa en todo momento.

Si alguno de nuestros dispositivos falle por algún motivo y debe reemplazarse, debemos configurarlo tal como hemos hecho al comienzo de este apartado, cargando Firmware y demás parámetros(PAN ID,NJ,SP,etc). De manera sencilla XCTU nos da la posibilidad de guardar perfiles de los dispositivos, por lo tanto es importante tener un backup de los tres perfiles que siempre utilizaremos en este proyecto (Coordinador, Router y End Devices), de esta forma cuando un dispositivo nuevo debe configurarse, se le cargará el perfil correspondiente y ya tendrá consigo los parámetros que hemos modificado para cada caso, al conectarlo nuevamente ocurrirán los procesos para que el nuevo integrante pueda unirse a la red.

Capítulo 8

Resultados

Luego de lo desarrollado en el capítulo anterior, se exponen en éste los resultados del proyecto. Se puso a prueba el rendimiento de los módulos realizando un análisis de RSSI (Received Signal Strength Indicator) para determinar el alcance efectivo así como también el impacto que causa en el sistema la incorporación de Routers y paquetes de distinto tamaño. Las configuraciones utilizadas ya han sido descritas. Mediante el programa XCTU se cambiaron los parámetros acordes y se realizaron las pruebas utilizando las herramientas que provee, Range Test y Throughput. Cada prueba experimental fue realizada cinco veces sobre un pasillo de cuarenta metros con presencia de WiFi en la trayectoria de los módulos. Todas las pruebas se realizaron sobre la banda de 2.4GHz y abarcan el alcance efectivo de transmisión, verificación de funcionamiento de circuitos, verificación de adquisición e interpretación de frames, pruebas de guardado y alarmas.

8.1. RSSI

El indicador de intensidad de señal recibida es evaluado aquí en función de la distancia, se utiliza un nodo Coordinador y otro End Device para corroborar cómo afecta sobre el parámetro RSSI.

En la Figura 8.2 pueden observarse las distintas distancias propuestas, en cada punto se realizó el análisis de señal mediante la herramienta Range Test de XCTU enviando 100 paquetes de 32 bytes y promediando los resultados obtenidos para cada distan-

cia, y una velocidad de 115200bps en el puerto. Cuando los XBee se vinculan el dispositivo final transmite hacia el coordinador que envía un acuse de recibo(ACK). Las mediciones experimentales se han hecho variando la potencia del transmisor en los siguientes valores: -2dB, 0dB y +2dB.

Para este apartado se hace uso de modelos de propagación para presentar una visión general, con ellos se busca predecir la potencia recibida a distintas distancias del transmisor, siempre en campo lejano donde el frente de onda ya se ha establecido. El Modelo de *Propagación en Espacio Libre* es utilizado cuando entre transmisor y receptor existe una línea clara de vista entre ellos, dando como resultado la potencia recibida en función de la distancia entre los nodos:

$$P_r = P_t G_r G_t \left(\frac{\lambda}{4\pi d} \right)^2 \quad (8.1)$$

Cabe recordar que la ecuación anterior fue vista ya en el Capítulo 3, para modelar se utiliza la máxima potencia de transmisión (2mw) con ganancias de antena (según la hoja de datos) de 1.5 dBi. La frecuencia de operación es de 2.4 GHz y longitud de onda $\lambda = c/f$, tomando la velocidad de la luz $c=2.99 \times 10^8$ m/s.

Esta teoría es válida siempre y cuando las antenas puedan verse en línea recta de visión LOS (Line of Sight) sin la presencia de obstáculos en la trayectoria que alteren la transmisión.

Cuando esto sucede, los obstáculos presentes generan reflexiones provocando que el receptor adquiera las señales desfasadas debido a que recorren caminos con diferentes longitudes dando lugar a una interferencia destructiva o constructiva según sea el caso. El fenómeno fue estudiado dando como resultado las zonas de Fresnel, este término hace referencia a un conjunto de elipsoides que se forman en el medio de propagación entre la antena emisora y receptora. Las zonas están definidas por los límites donde las ondas dentro de estos llegan al receptor con la misma fase de la señal transmitida, Fresnel demostró que el efecto de las demás zonas es compensado por el efecto de la primera, de modo que un reflector dentro de la primera zona define el efecto ocasionado, razón por la cual es necesario dejar esta zona libre[6].

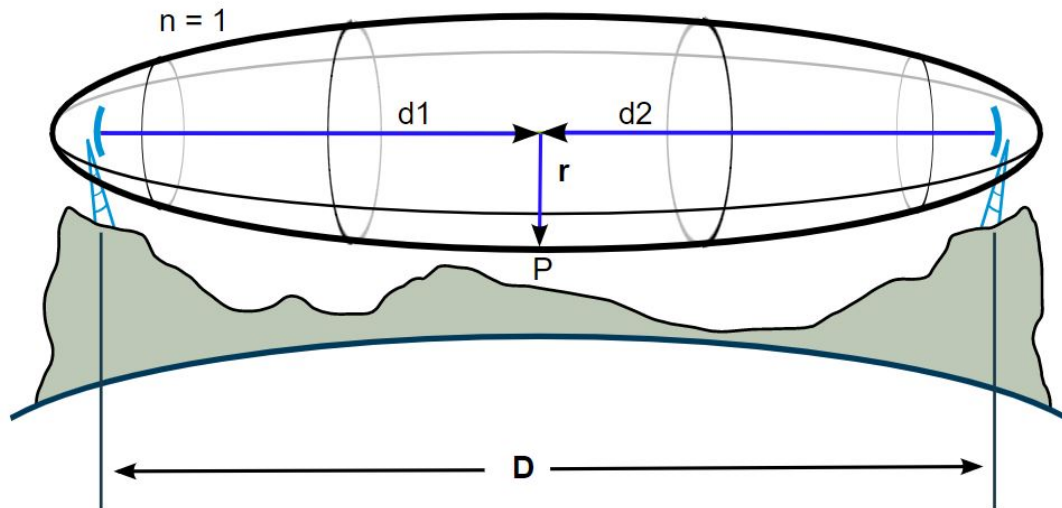


Figura 8.1: Imagen representativa de las zonas de Fresnel [19] para el caso $n=1$ (primer zona).

Observar que las elipses no se encuentran sólo en un plano sino que es perpendicular a la recta en todas las direcciones. La siguiente ecuación permite calcular el radio de la zona en las diferentes distancias de interés:

$$r = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \quad (8.2)$$

Donde la ecuación está definida por la longitud de onda (λ), las distancias d_1 y d_2 respecto al punto de interés y n el número de la zona de Fresnel. Se calcula el radio central utilizando la distancia propuesta experimental de 40 metros, por lo tanto se toma una distancia media en el centro de la elipse de 20 metros lo cual nos da un valor $r = 1.11$ metros, este será el radio máximo de la primera zona de Fresnel. Teniendo en cuenta que el radio debe ser tomado en todas las direcciones adyacentes a la trayectoria central se asume que la presencia de reflexiones ya que para el radio calculado se encuentran objetos, si bien los nodos están a una altura de 2 metros, la presencia de paredes estrechas en la trayectoria por ejemplo, empiezan a tener un impacto significativo en las transmisiones por lo tanto ya no estaremos en el espacio libre como se propuso al comienzo, por esa razón hacemos uso del *modelo de Dos*

Rayos(Reflexiones en el Suelo) para estimar los resultados RSSI[15], este modelo incluye los efectos de reflexión.

El siguiente gráfico nos muestra la antena receptora la cual recibe una onda directa y otra reflejada:

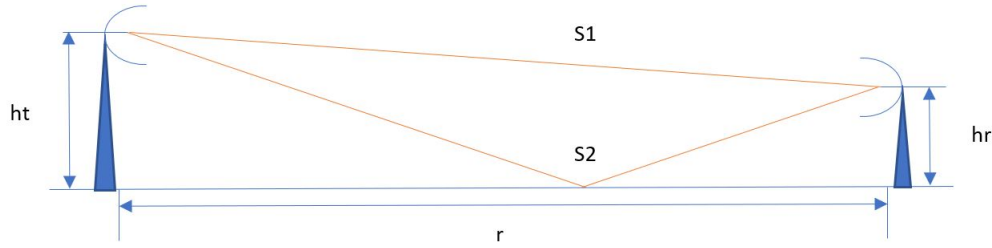


Figura 8.2: Gráfico del modelo de reflexiones en el suelo .

Extendiendo las líneas representativas podemos formar dos triángulos rectángulos, aplicando geometría sobre ambos trayectos S1 y S2 se utiliza el teorema de Pitagóricas para calcular la diferencia entre ambos caminos:

$$S_1 = \sqrt{r^2 + (ht - hr)^2} \quad (8.3)$$

$$S_2 = \sqrt{r^2 + (ht + hr)^2} \quad (8.4)$$

Teniendo en cuenta que hablamos de una onda, considerando la longitud podemos expresar la diferencia de fase como:

$$\Delta\phi = \left(\frac{2\pi}{\lambda}\right)(\sqrt{r^2 + (ht + hr)^2} - \sqrt{r^2 + (ht - hr)^2}) \quad (8.5)$$

El campo eléctrico de las antenas está dado por la expresión $E = E_0(1 + \rho e^{j\Delta\phi})$ donde ρ es el coeficiente de reflexión de la tierra (que depende de la permitividad dieléctrica y ángulo de incidencia), al aumentar la distancia r el ángulo decrece y tiene el coeficiente de reflexión a cero, utilizando identidades trigonométricas se puede expresar el módulo cuadrado de la transferencia entre el campo resultante y el directo:

$$|(E/E_0)|^2 = [2\text{sen}(\frac{2\pi(ht)(hr)}{\lambda r})]^2 \quad (8.6)$$

Finalmente incorporamos este último termino a la ecuación de Friis que vimos al comienzo:

$$P_r = P_t G_r G_t (\frac{\lambda}{4\pi d})^2 [2\text{sen}(\frac{2\pi(ht)(hr)}{\lambda r})]^2 \quad (8.7)$$

Aquí la distancia es relativamente corta (40m), sin embargo bajo ciertas condiciones donde $r > \frac{(ht)(hr)2\pi}{\lambda}$ la ecuación puede expresarse de la siguiente forma:

$$P_r = P_t G_r G_t \frac{((ht)(hr))^2}{r^4} \quad (8.8)$$

La expresión anterior solo es valida para la condición mencionada, aquí utilizamos la Ecuación 9.7.

Por último, el *modelo de perdidas "Log Distancia"* también llamado Log Distance Path Loss Model, este modelo puede considerarse como una generalización de la ecuación de Friis de espacio libre, donde la potencia disminuye a una tasa de $(\frac{1}{d})^2$ y donde se agrega una variable aleatoria para tener en cuenta los efectos de "sombreado", este termino hace referencia a la fluctuación de la potencia recibida debido a objetos que obstruyen la ruta de propagación entre el transmisor y receptor, puede expresarse mediante la siguiente ecuación:

$$PL_{(d)} = PL_0 + 10\gamma \log_{10}(\frac{d}{d_0}) + \sigma \quad (8.9)$$

Donde $PL_{(d)}$ es la pérdida en función de la distancia en dBm, PL_0 nuevamente hace referencia a las pérdidas pero a una distancia de referencia (d_0) la cual se recomienda que sea de 1 metro [8]. PL_0 puede obtenerse a través de la ecuación de Friis, γ es llamado exponente de pérdida de trayectoria que depende del tipo de entorno, en condición de visión directa entre las antenas varía entre 1.2 y 2 (valor utilizado en este caso) y σ la desviación estándar calculada con los valores de la ecuación de Friis, lo cual arroja un valor de 8.

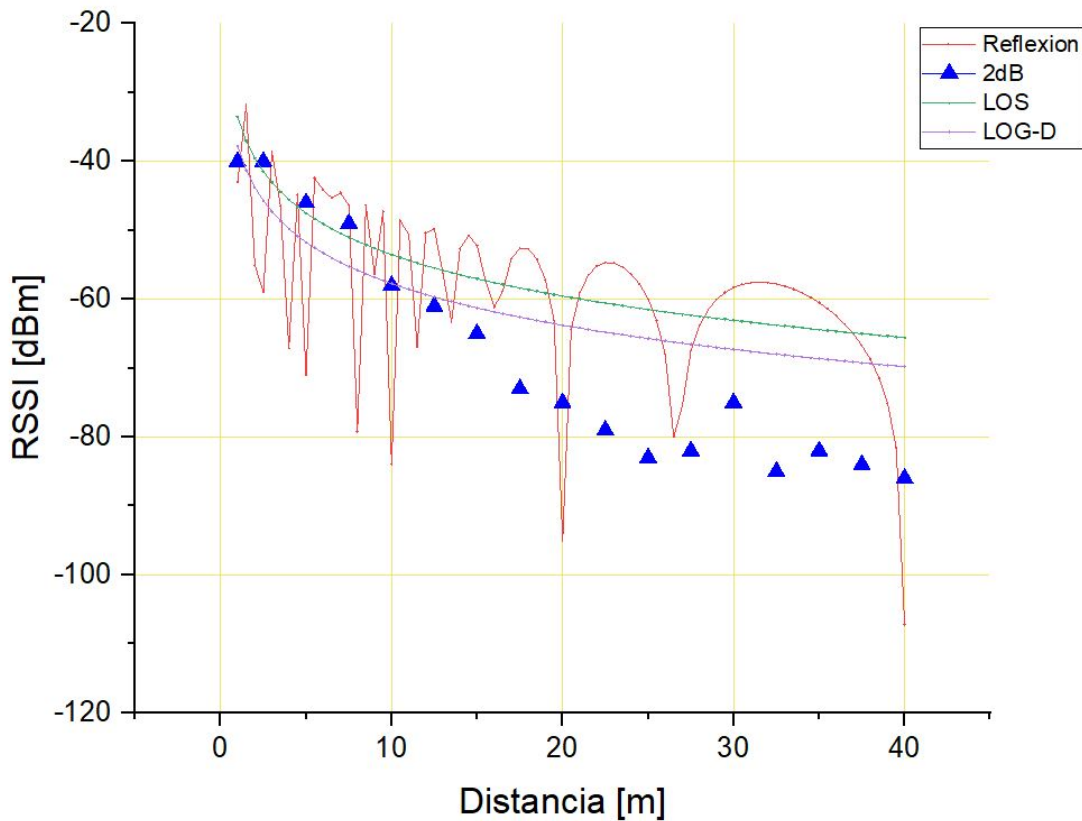


Figura 8.3: Resultados RSSI en función de la distancia para los distintos modelos propuestos.

Dados los niveles de transmisión en los escenarios descritos, la Figura 8.3 muestra los diferentes valores esperados para cada modelo de propagación y el resultado experimental, el modelo de Dos Rayos presenta una mayor discontinuidad que los otros debido a las variaciones de fase en las diferentes distancias, mientras que los otros una curva decreciente mucho más continua ya que para el modelo de Propagación Libre no contemplamos obstáculo alguno, lo cual es un escenario demasiado positivo ya que en realidad las antenas no tienen la altura suficiente para verse libremente en el espacio, mientras que el Log Distancia al tener algunas consideraciones en cuanto a obstáculos muestra una curva más pesimista que la anterior. Observando los resultados experimentales con los XBee, ninguno de los modelos coincide con los datos

obtenidos de manera precisa, si bien la curva presenta un descenso a medida que la distancia aumenta igual que todos los modelos propuestos, se observan distintos picos a lo largo de la trayectoria de transmisión. Siendo optimista se optó por utilizar el modelo de dos Rayos como posible aproximación, ya que los dispositivos no se encuentran en línea directa de visión (LOS) por lo que descartamos el modelo de Espacio Libre, si bien Log Distancia contempla las pérdidas por objetos dentro de la trayectoria, para las zonas de Fresnel calculadas se comprueba que dentro del radio no sólo hay presencia de objetos sino también la presencia de paredes, por lo tanto podemos decir que inevitablemente se producen reflexiones significativas mostradas en la curva por este motivo. Si bien no se encuentran los resultados en este apartado, siguiendo este modelo para colocar los módulos, aumentar la altura de las antenas mejoró considerablemente los resultados debido a que la mayoría de las reflexiones fueron mucho menos considerables ya que se alejan los objetos de la primer zona de Fresnel.

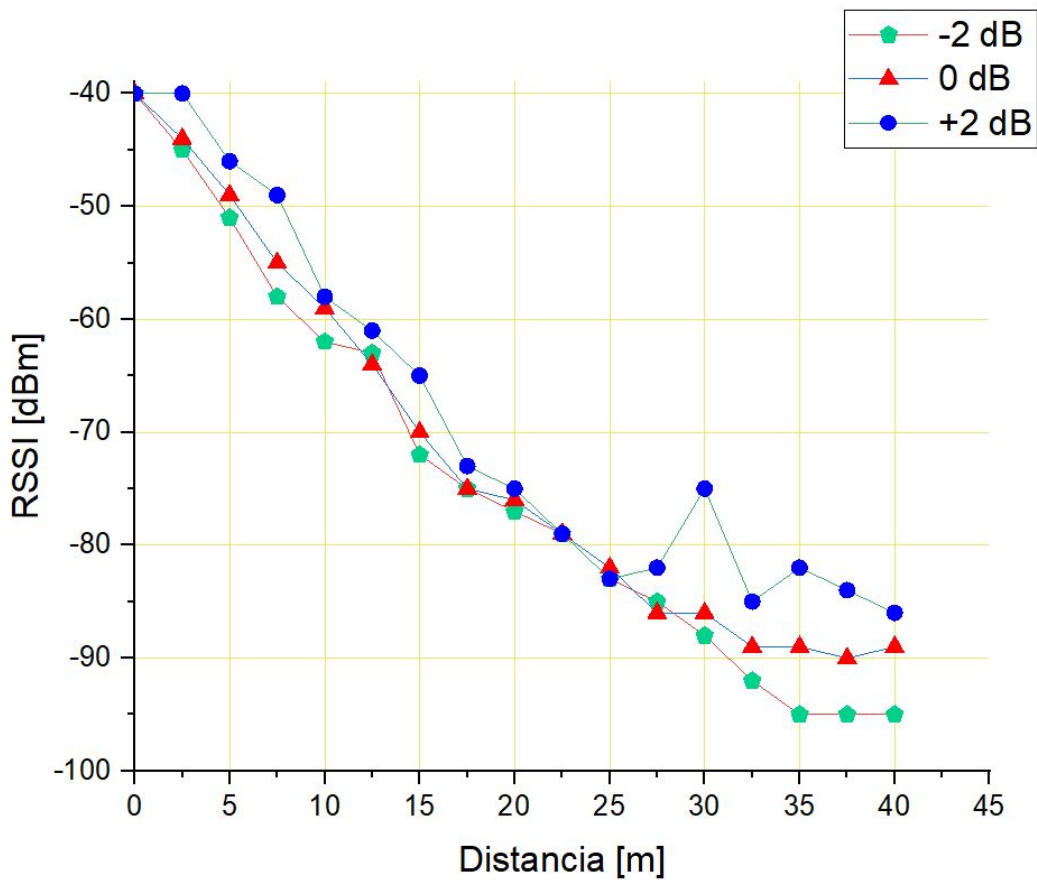


Figura 8.4: Resultados RSSI en función de la distancia y potencia de transmisión.

Por último, como era de esperarse, la Figura 8.4 muestra que a medida que los dispositivos se alejan la intensidad disminuye, se observan puntos inestables que pueden atribuirse a fenómenos presentes en el medio de transmisión. El aumento en la potencia de transmisión mejoró el desempeño de los módulos.

8.2. Throughput

Veamos ahora el rendimiento de las transmisiones, el throughput es la medida de datos enviados o recibidos en un período de tiempo definido, considerando aspectos como el tiempo en que se incluye la información dentro de la trama y el tiempo de

espera que tienen para transmitir. Los frames que lleguen al receptor del nodo Coordinador ingresan al puerto serie de la PC vía usb, el objetivo es medir el rendimiento del módulo XBee en función de la velocidad de transmisión (baudios) y tamaño de paquete. El análisis es realizado con un Coordinador y End Device, utilizando la herramienta Throughput de XCTU, se envían paquetes hacia el Coordinador cuya longitud irá aumentando hasta un valor máximo de 84 bytes para cada una de las velocidades de prueba: 9600bps, 19200bps, 38400bps, 57600bps y 115200bps. No es posible predecir el tiempo empleado de forma exacta debido a que se desconoce cuando el dispositivo tiene acceso al medio para transmitir, lo que si sabemos es que si la comunicación se realiza sobre un mismo canal hay posibilidad de que sea utilizado por mas de una PAN si así se desea, los dispositivos compartirán medio aguardando su turno, por lo que si se quiere maximizar el throughput (y reducir la latencia) se debe asignar sólo una PAN para dicho canal.

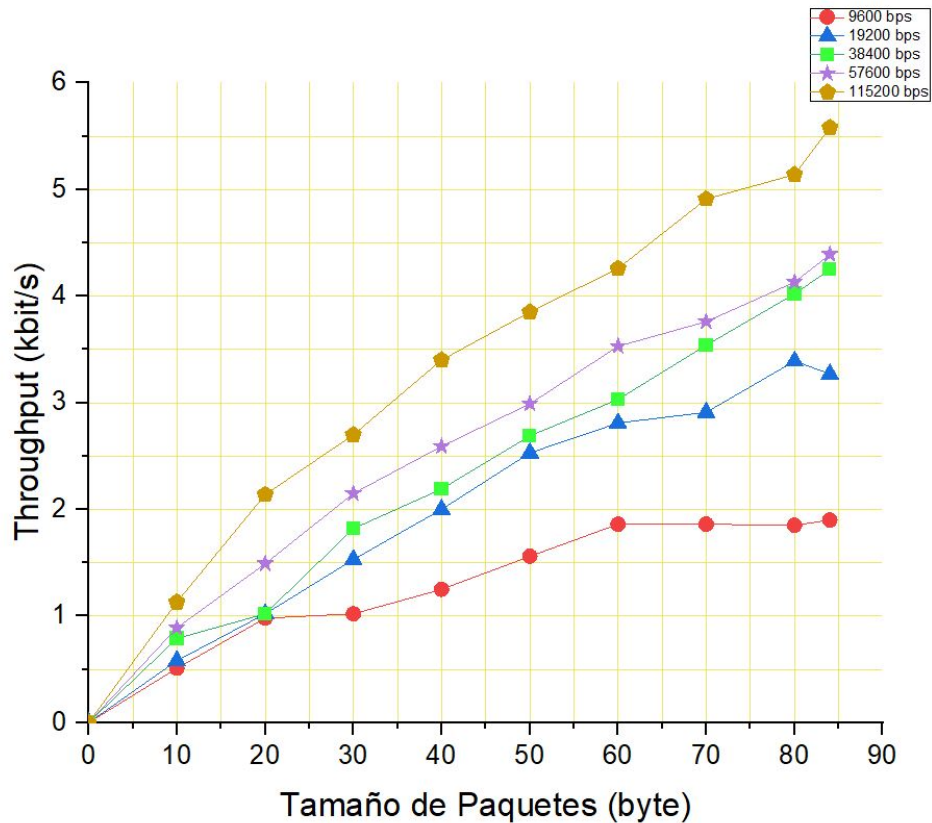


Figura 8.5: Resultados medición de throughput en función de la velocidad de transmisión y tamaño del paquete.

En la Figura 8.5 se exponen los resultados obtenidos, variando el número de bytes del paquete hasta un tamaño máximo admitido en el PAYLOAD de 84 bytes. El estándar ZigBee nos dice que pueden alcanzarse velocidades de transmisión de hasta 250kbps, sin embargo podemos ver que el desempeño a máxima capacidad de carga útil y baudios llegó sólo a un valor máximo de 5.58kbps, lejos del nivel de rendimiento expuesto por el fabricante, esto se debe a que sólo estamos midiendo la carga útil de los frames con esta herramienta (ya que no se dispone de un analizador de protocolo), hemos visto en el Capítulo 4 que a la hora de realizar el envío de información no sólo se envía lo que nos interesa(PAYLOAD), sino que viene acompañado de los campos que conforman el estándar, preámbulos, encabezados de MAC, NWK y APS, acuse

de recibos, etc. A la hora de ver los resultados deben tenerse en cuenta ya que el verdadero Throughput debe medirse teniendo en cuenta todos estos aspectos que resultan invisibles para la herramienta provista por XCTU.

8.3. Throughput en Red con Múltiples Saltos

Uno de los objetivos es el aumento del alcance efectivo de transmisión añadiendo Routers en la trayectoria de los módulos. Se transmitirán paquetes comenzando por una red simple de Coordinador y único End Device, luego se añadirá un Router para que retransmitir el mensaje y luego dos respectivamente, todos trabajando sobre el mismo canal. Se han configurado los módulos en 115200bps. Para forzar las conexiones correctamente se controlaron las asociaciones limitando el parámetro NJ, obligando a los dispositivos a conectarse a los nodos que permitían la asociación, de manera controlada.

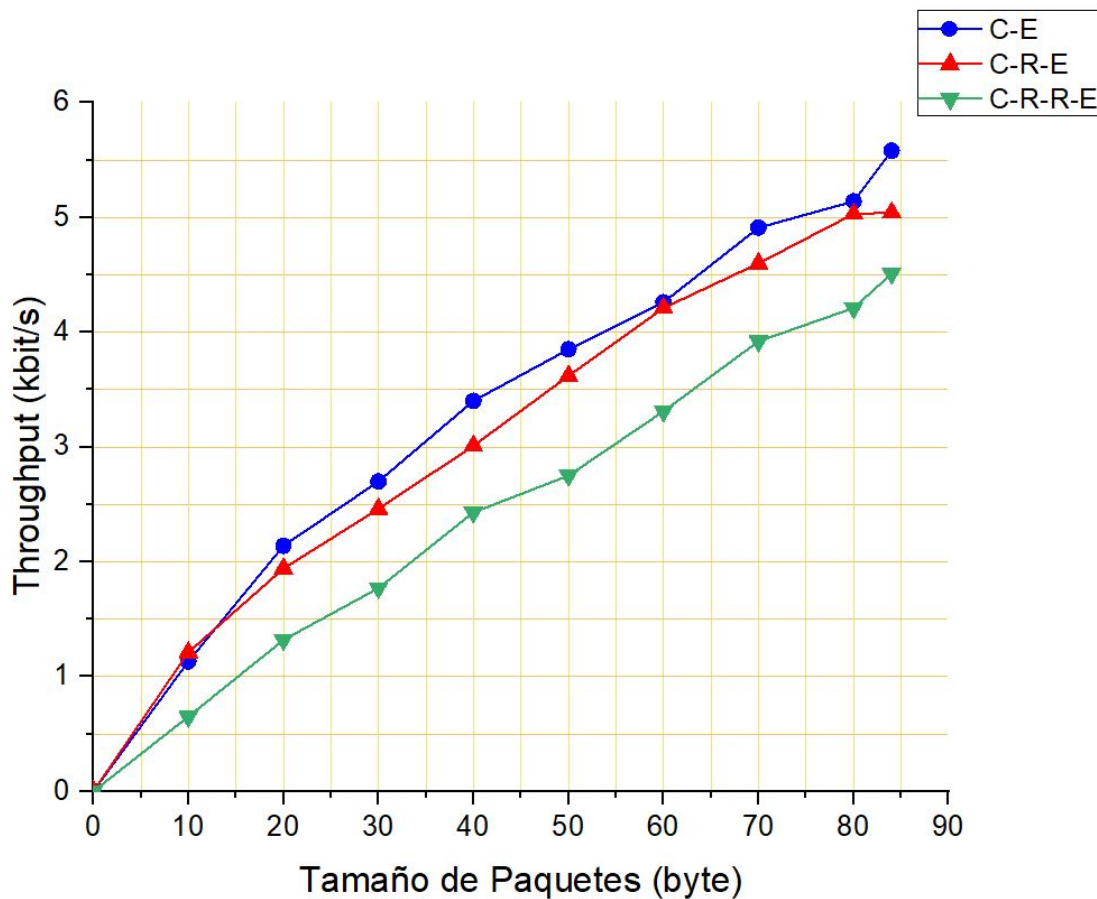


Figura 8.6: Resultados medición de throughput en función de configuración de red y tamaño del paquete.

La Figura 8.6 revela que la presencia de Routers en la red tiene un efecto significativo en el Throughput y rendimiento. Como era de esperarse el número de saltos impacta directamente en el throughput. La comunicación directa arroja los mejores resultados ya que el canal siempre está libre para que el dispositivo final transmita al Coordinador. Cuando se añadió un Router en la red, mientras éste le deja el mensaje al Coordinador, el End Device debe esperar su turno debido a que el canal está siendo ocupado, finalmente añadir otro Router redujo aún más el rendimiento tal como se esperaba. En primera instancia como se ha visto en el Capítulo 3, 802.15.4 responde a una red WPAN (Wireless Personal Area Network) con características como

alcance relativamente corto y una baja velocidad de transferencia, ZigBee extiende al estándar agregando sus propias características, esto produce mas overhead al sistema, es decir, el envío de datos de control y señalización dentro de los frames propia del estándar reduce el throughput y aumenta la latencia, si a eso sumamos los tiempos de retransmisión y la cantidad de nodos por canal, hacen que ZigBee no sea muy amigable en aplicaciones que requieran baja latencia y poca tolerancia a los cambios de tiempo, por lo tanto debe haber un equilibrio entre cantidad de saltos y throughput según las necesidades requeridas a la hora de diseñar la red ya que ZigBee no fue creado para maximizar estos detalles.

8.4. Consumo

En la practica se espera un consumo de algunos miliamper, se ha configurado el End Device para enviar cada 2 segundos sin descanso el envío de muestras, utilizando todas las entradas analogicas activas, todas las entradas analogicas en estado High y las tres salidas de actuador en estado alto.

La siguiente imagen es un oscilograma de la corriente de operación con las características anteriores, para obtenerla se utilizó una resistencia de 10Ω en serie con alimentación.

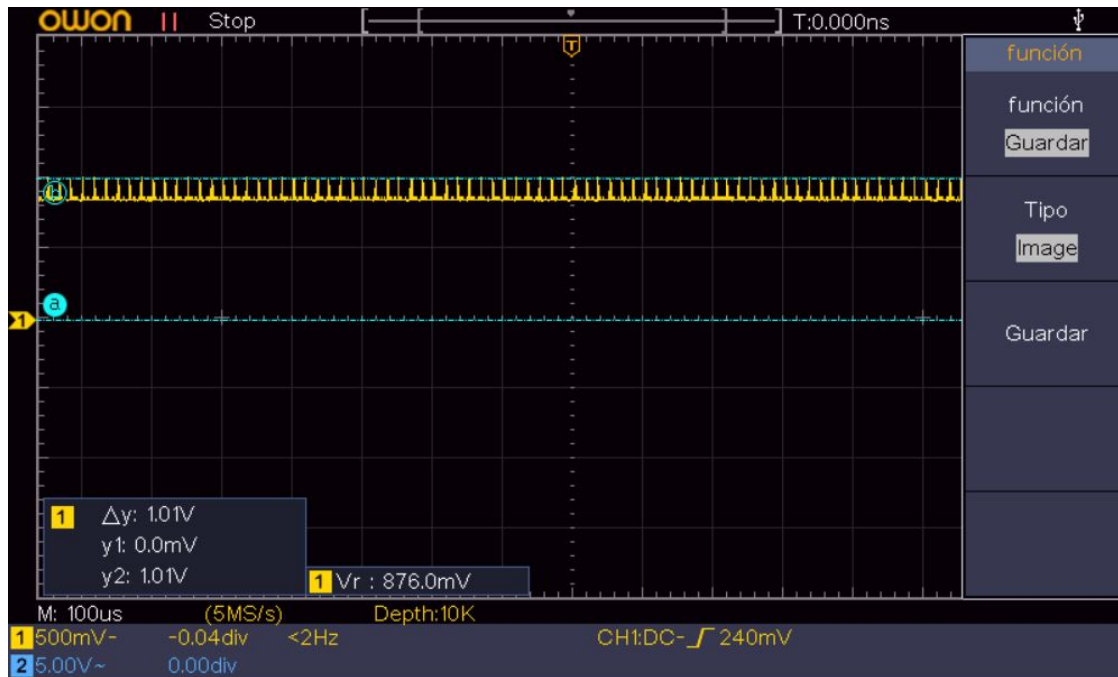


Figura 8.7: Oscilograma para el estudio de corriente consumida.

Se observa un Poll correspondiente al envío del Frame 0x92(Recepción de Estados I/O de Dispositivo Remoto), vemos un valor de tensión RMS de 876 mV, dando una corriente consumida total de aproximadamente 87.6 mA.

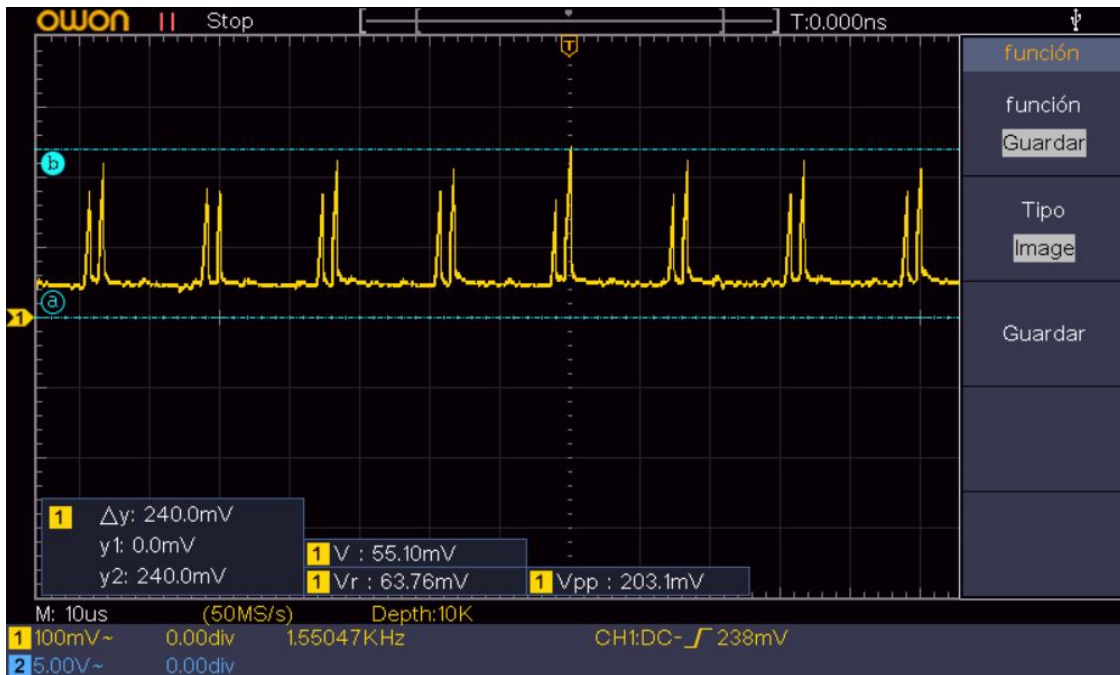


Figura 8.8: Oscilograma para el estudio de corriente consumida.

En la Figura 8.8 se observa con más detalle el espectro de corriente sin el acople de continua, vemos que los picos tienen un voltaje pico a pico de 203.1 mV con un promedio de 55.10 mV y valor RMS de 63.76 mV.

Consideramos que el consumo de la placa con todas las entradas y salidas activas se encuentra en un rango aceptable, menor a 100 mA. Teniendo en cuenta que alimentamos el Hardware con 24 V, tenemos una potencia de 2.1 W.

8.5. Recepción de Frames

Se comprobó la correcta adquisición de los datos provenientes de los módulos comenzando por el caso más simple, la RED A donde sólo un dispositivo final se encarga del envío periódico de información hasta hasta el caso más complejo RED C, donde varios dispositivos lo hacen en simultáneo.

Para dicho propósito se evalúa la cantidad de datos almacenados en el BackUp creado por la aplicación, cada frame que ingresa es procesado por el instrumento virtual VI PARSING ya descrito, recordemos que discrimina los datos pertinentes (muestras

analógicas y valores discretos), luego son guardados en un archivo Excel con fecha y hora de cada dato. Teniendo en cuenta que se conoce el periodo de envío, es posible predecir cuantas muestras deberían almacenarse en el archivo para un determinado tiempo en primera instancia.

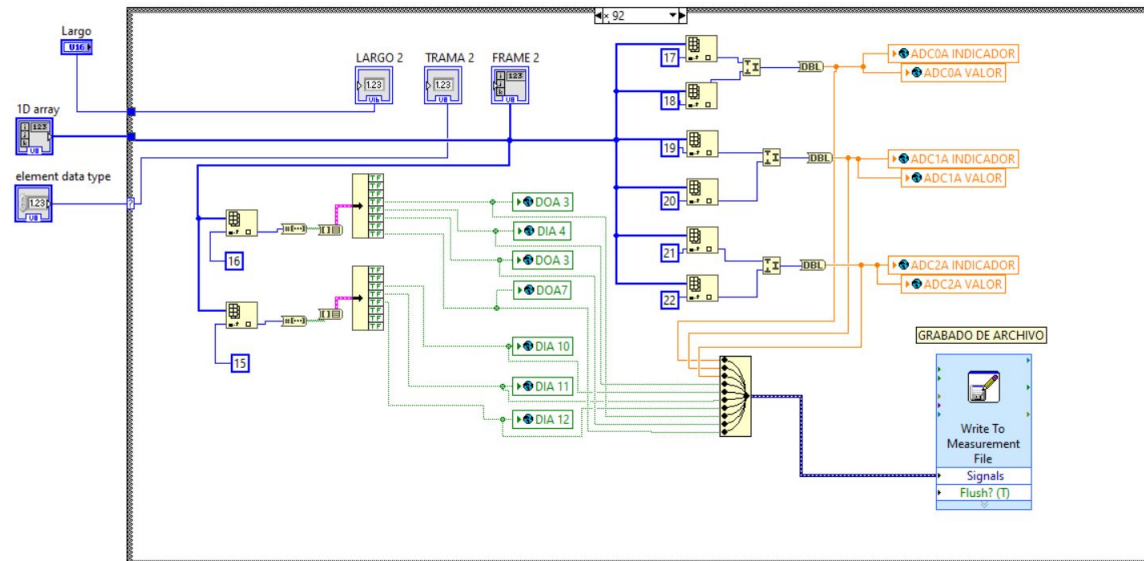


Figura 8.9: Recolección de variables y guardado.

En la Figura 8.9 se encuentra el instrumento virtual donde se realiza la extracción de las variables contenidas en el frame y se guarda en su correspondiente hoja de Excel. Como tenemos un sólo dispositivo no se realiza un análisis del origen de los frames ya que siempre provienen del mismo XBee.

Creada la red donde el dispositivo End Device recolecta los valores, se establecen voltajes definidos en distintos horarios para comprobar que efectivamente el dispositivo Coordinador percibe correctamente los cambios. Cabe destacar que en este apartado no se hará juicio de las muestras adquiridas por lo que trabajaremos directamente con los valores ADC entre 0 (mínimo), aleatorio y 1023 (máximo) aplicando una diferencia de potencial en las entradas y restaremos importancia por el momento de los valores discretos.

La siguiente tabla muestra los rangos forzados para cada tramo de tiempo, los resultados expuestos corresponden a una hora de loggeo completa, aunque el dispositivo

fue utilizado el día completo, se hacia bastante mas extenso exponer todo el rango horario por lo que se ha acotado a una hora aleatoria:

Tiempo[Hora:Minutos]	ADCX
18:55	Máximo
19	Mínimo
19:05	Máximo
19:10	Aleatorio
19:15	Mínimo
19:20	Aleatorio
19:25	Máximo
19:30	Aleatorio
19:35	Mínimo
19:40	Máximo
19:45	Aleatorio
19:50	Máximo
19:55	Mínimo

Cuadro 8.1: Valores forzados para cada rango de tiempo para un único nodo.

Como el dispositivo End Device transmite cada 3 segundos, en un minuto realiza 20 transmisiones, tomando los 60 minutos de una hora deberían haber $(20) \times (60 \text{ minutos}) = 1200$ frames guardados.

30	11/9/2020 18:55:02	1023	1023	1023
31	11/9/2020 18:55:05	1023	1023	1023
32	11/9/2020 18:55:08	1023	1023	1023
33	11/9/2020 18:55:11	1023	1023	1023
34	11/9/2020 18:55:14	1023	1023	1023
35	11/9/2020 18:55:17	1023	1023	1023
36	11/9/2020 18:55:20	1023	1023	1023
37	11/9/2020 18:55:23	1023	1023	1023
38	11/9/2020 18:55:25	1023	1023	1023
39	11/9/2020 18:55:28	1023	1023	1023
1265	11/9/2020 19:54:52	0	0	0
1266	11/9/2020 19:54:55	0	0	0
1267	11/9/2020 19:54:58	0	0	0
1268	11/9/2020 19:55:01	0	0	0

Figura 8.10: Celdas de Excel correspondientes al BackUp.

En la Figura anterior podemos ver el primer y último frame tenido en cuenta para la prueba dando la siguiente cantidad de registros en una hora:

- *XBee A*: [18:55:02 - 19:55:01] ha guardado 1239 frames.

Analizando los datos, se observa que el aumento de la cantidad de frames a lo esperado se debe a transmisiones que llegan con una diferencia de 2 segundos en vez de 3 como en el recuadro azul que se ha añadido en la Figura 8.10.

A partir de esto veamos a continuación los resultados obtenidos correspondiente a cada tramo horario:

Time	ADC0	ADC1	ADC2
11/9/2020 18:55:02	1023	1023	1023
11/9/2020 18:55:05	1023	1023	1023
11/9/2020 18:55:08	1023	1023	1023
11/9/2020 19:00:01	0	0	0
11/9/2020 19:00:04	0	0	0
11/9/2020 19:00:07	0	0	0
11/9/2020 19:05:00	1023	1023	1023
11/9/2020 19:05:03	1023	1023	1023
11/9/2020 19:05:06	1023	1023	1023
11/9/2020 19:10:01	865	915	678
11/9/2020 19:10:04	866	914	672
11/9/2020 19:10:06	866	916	0
11/9/2020 19:15:01	0	0	0
11/9/2020 19:15:03	0	0	0
11/9/2020 19:15:05	0	0	0
11/9/2020 19:20:00	617	881	716
11/9/2020 19:20:03	617	881	716
11/9/2020 19:20:06	617	881	716
11/9/2020 19:25:02	1023	1023	1023
11/9/2020 19:25:05	1023	1023	1023
11/9/2020 19:25:08	1023	1023	1023
11/9/2020 19:30:01	477	501	542
11/9/2020 19:30:04	477	502	542
11/9/2020 19:30:07	477	502	541
11/9/2020 19:35:01	0	0	0
11/9/2020 19:35:04	0	0	0
11/9/2020 19:35:07	0	0	0
11/9/2020 19:40:00	1023	1023	1023
11/9/2020 19:40:03	1023	1023	1023
11/9/2020 19:40:06	1023	1023	1023
11/9/2020 19:45:02	469	514	662
11/9/2020 19:45:05	469	515	662
11/9/2020 19:45:08	469	515	662
11/9/2020 19:50:02	1023	1023	1023
11/9/2020 19:50:05	1023	1023	1023
11/9/2020 19:50:08	1023	1023	1023
11/9/2020 19:55:01	0	0	0
11/9/2020 19:55:04	0	0	0
11/9/2020 19:55:07	0	0	0

Figura 8.11: Resultados obtenidos.

Tal como se ha anunciado previamente en el Cuadro 9.1 con los valores forzados, se llega a la conclusión de que efectivamente los datos adquiridos por la placa End Device y enviados al Coordinador corresponden con lo que se había propuesto inicialmente con una leve latencia, por eso se toman tres frames para corroborar

que los datos realmente cambian sólo cuando fueron forzados a hacerlo, estamos en condiciones de decir que la recepción funciona correctamente, el siguiente paso es comprobarla cuando aumenta la cantidad de nodos.

Ahora la aplicación debe adquirir datos de dos XBee diferentes, nuevamente se realiza el mismo procedimiento forzando tensiones a la entrada:

Tiempo[Hora:Minutos]	ADCX
16:40	Mínimo
16:45	Máximo
16:50	Mínimo
16:55	Aleatorio
17:00	Mínimo
17:05	Máximo
17:10	Mínimo
17:15	Máximo
17:20	Aleatorio
17:25	Mínimo
17:30	Máximo
17:35	Mínimo
17:40	Máximo

Cuadro 8.2: Valores forzados para cada rango de tiempo con dos nodos.

En este caso el Instrumento Virtual VI Parsing resulta de suma importancia, ya que hay más de un dispositivo en la red, debemos discriminar que nodo envía la información, para ello se añade una estructura Case, dependiendo la dirección leída dentro del frame guarda los valores en la variable que corresponda ADCXA o ADCXB.

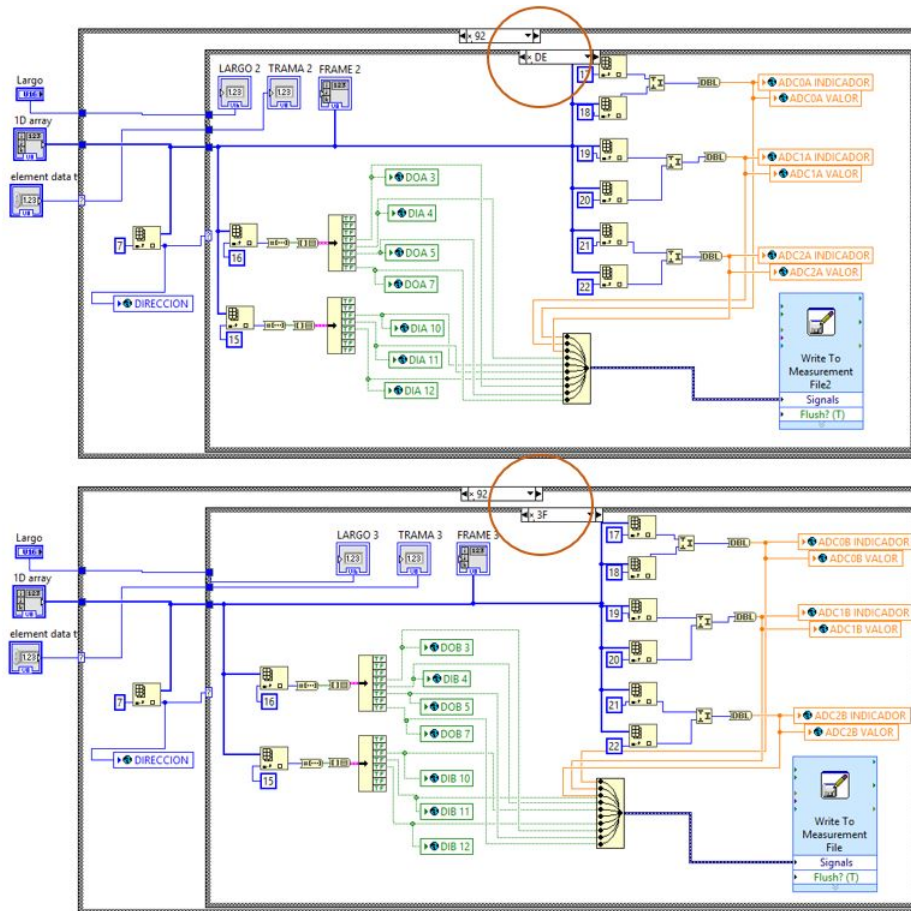


Figura 8.12: Parsing con estructura Case adicional.

La Figura 8.12 muestra dentro del círculo un byte (extraído del frame que llega al puerto) de la dirección única de los dispositivos, en este caso xDE y x3F, estos son los condicionales del Case para ingresar al grupo de variables que corresponde a cada XBee y su Excel particular.

Los módulos ahora son configurados para enviar los datos cada 5 segundos (RED B), esto es importante ya que al aumentar la cantidad de dispositivos por canal aumenta la latencia de la recepción.

47	22/9/2020 16:40:06	1	1	1
48	22/9/2020 16:40:11	1	1	1
49	22/9/2020 16:40:16	1	1	1
777	22/9/2020 17:40:01	788	3	1
778	22/9/2020 17:40:06	1023	1023	1023
779	22/9/2020 17:40:11	1023	1023	1023

Figura 8.13: Celdas de inicio y fin de guardado correspondiente al dispositivo A.

46	22/9/2020 16:40:04	0	0	0
47	22/9/2020 16:40:09	0	0	0
48	22/9/2020 16:40:14	0	0	0
778	22/9/2020 17:40:03	1023	1023	0
779	22/9/2020 17:40:08	1023	1023	1023
780	22/9/2020 17:40:13	1023	1023	1023

Figura 8.14: Celdas de inicio y fin de guardado correspondiente al dispositivo B.

La Figura 8.16 y 8.14 muestran la cantidad de frames guardados para cada dispositivo (ADCXA y ADCXB). Para este caso, como el envío periódico es configurado en 5 segundos, cada dispositivo debería enviar 720 frames en una hora dando un total de 1440. Los resultados obtenidos son los siguientes:

- *XBee A*: [16:40:06 - 17:40:01] ha guardado un total de 731 frames.
- *XBee B*: [16:40:04 - 17:40:03] ha guardado un total de 733 frames.

Nuevamente como hemos dicho antes, el tiempo configurado de 5 segundos no es exacto, hay frames que llegan antes de previsto, acumulando en una hora 11 frames adicionales para *XBee A* y 13 para *XBee B*. La siguiente figura contiene los valores ADCX correspondientes al forzado.

Time	ADCOA	ADC1A	ADC2A	Time	ADC0B	ADC1B	ADC2B
22/9/2020 16:40:06	1	1	1	22/9/2020 16:40:04	0	0	0
22/9/2020 16:40:11	1	1	1	22/9/2020 16:40:09	0	0	0
22/9/2020 16:40:16	1	1	1	22/9/2020 16:40:14	0	0	0
22/9/2020 16:45:02	1023	1023	680	22/9/2020 16:45:00	0	1023	0
22/9/2020 16:45:07	1023	1023	1023	22/9/2020 16:45:05	1023	1023	1023
22/9/2020 16:45:12	1023	1023	1023	22/9/2020 16:45:09	1023	1023	1023
22/9/2020 16:50:02	1	1	1023	22/9/2020 16:50:00	1023	0	1023
22/9/2020 16:50:07	1	1	1	22/9/2020 16:50:05	0	0	0
22/9/2020 16:50:12	1	1	1	22/9/2020 16:50:10	0	0	0
22/9/2020 16:55:03	935	257	2	22/9/2020 16:55:00	0	0	0
22/9/2020 16:55:08	953	698	778	22/9/2020 16:55:05	697	954	778
22/9/2020 16:55:12	953	698	778	22/9/2020 16:55:10	697	954	778
22/9/2020 17:00:03	1	2	1	22/9/2020 17:00:01	0	0	0
22/9/2020 17:00:08	1	1	1	22/9/2020 17:00:06	0	0	0
22/9/2020 17:00:13	1	1	1	22/9/2020 17:00:10	0	0	0
22/9/2020 17:05:03	1023	1023	1	22/9/2020 17:05:01	0	1023	0
22/9/2020 17:05:08	1023	1023	1023	22/9/2020 17:05:06	1023	1023	1023
22/9/2020 17:05:13	1023	1023	1023	22/9/2020 17:05:11	1023	1023	1023
22/9/2020 17:10:04	0	0	0	22/9/2020 17:10:01	1023	0	1023
22/9/2020 17:10:09	1	1	1	22/9/2020 17:10:06	0	0	0
22/9/2020 17:10:14	1	2	2	22/9/2020 17:10:11	0	1	1
22/9/2020 17:15:04	1023	1023	1	22/9/2020 17:15:02	0	0	0
22/9/2020 17:15:09	1023	1023	1023	22/9/2020 17:15:07	1023	1023	1023
22/9/2020 17:15:14	1023	1023	1023	22/9/2020 17:15:11	1023	1023	1023
22/9/2020 17:20:04	847	835	1023	22/9/2020 17:20:02	1023	1023	1023
22/9/2020 17:20:09	841	827	758	22/9/2020 17:20:07	828	842	760
22/9/2020 17:20:14	842	827	760	22/9/2020 17:20:12	827	841	758
22/9/2020 17:25:00	842	828	761	22/9/2020 17:25:02	823	0	761
22/9/2020 17:25:05	1	1	1	22/9/2020 17:25:07	0	0	0
22/9/2020 17:25:10	2	2	2	22/9/2020 17:25:12	0	0	0
22/9/2020 17:30:00	1023	2	1	22/9/2020 17:30:03	1023	0	0
22/9/2020 17:30:05	1023	1023	1023	22/9/2020 17:30:07	1023	1023	1023
22/9/2020 17:30:10	1023	1023	1023	22/9/2020 17:30:12	1023	1023	1023
22/9/2020 17:35:01	1	1023	1023	22/9/2020 17:35:03	0	0	1023
22/9/2020 17:35:05	2	1	1	22/9/2020 17:35:08	0	0	0
22/9/2020 17:35:10	2	2	1	22/9/2020 17:35:13	0	0	0
22/9/2020 17:40:01	788	3	1	22/9/2020 17:40:03	1023	1023	0
22/9/2020 17:40:06	1023	1023	1023	22/9/2020 17:40:08	1023	1023	1023
22/9/2020 17:40:11	1023	1023	1023	22/9/2020 17:40:13	1023	1023	1023

Figura 8.15: Resultados obtenidos con dos End Devices.

Lo primero que se observa es que efectivamente ahora la respuesta no es tan rápida como el caso anterior, nuevamente se toman los primeros tres guardados de cada horario seleccionado para ver la evolución de las variables que ahora tardan unos segundos más en establecerse correctamente, si bien no es exacto al momento del cambio se comprueba que son tomados correctamente y que se guardan con pocos segundos de diferencia y mantienen la magnitud correctamente entre cada tramo horario.

Finalmente veamos lo que sucede cuando tres dispositivos finales envían datos, ahora se ha añadido un nuevo condicional en la estructura Case con la dirección del nuevo

integrante de la red (0xD4):

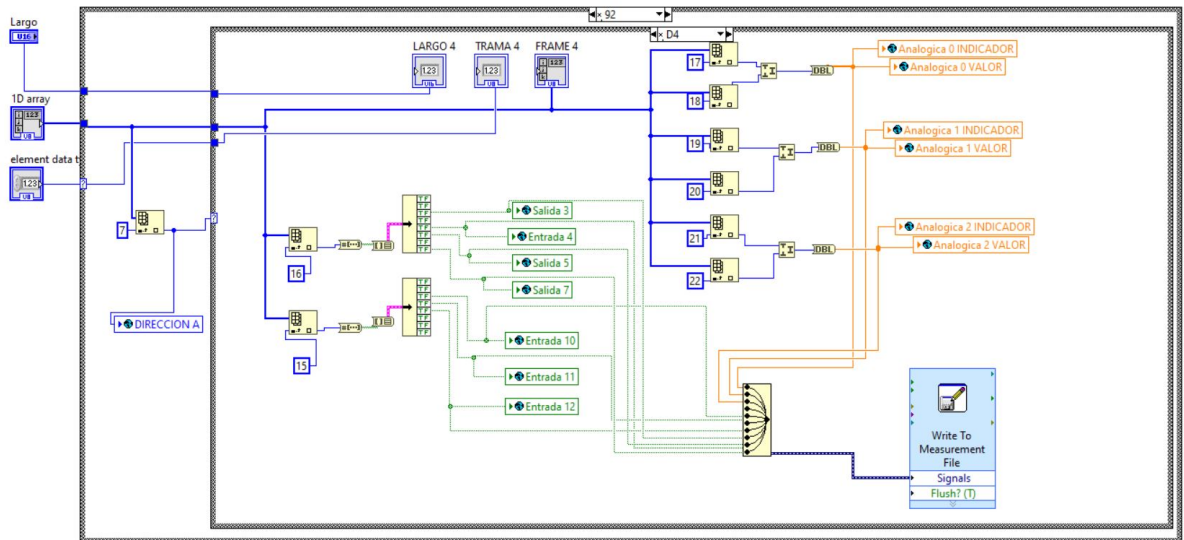


Figura 8.16: Resultados obtenidos con dos End Devices.

Ahora el período es de 15 segundos, lo que corresponde a 240 frames guardados durante una hora para cada uno de ellos. Veamos primero la tabla de estados:

Tiempo[Hora:Minutos]	ADCX
15:30	Mínimo
15:35	Máximo
15:40	Mínimo
15:45	Máximo
15:50	Mínimo
15:55	Aleatorio
16:00	Máximo
16:05	Mínimo
16:10	Máximo
16:15	Mínimo
16:20	Máximo
16:25	Mínimo
16:30	Máximo

Cuadro 8.3: Valores forzados para cada rango de tiempo con tres nodos.

Finalizado el tiempo de loggeo, nos arrojan la siguiente cantidad de celdas:

32	22/9/2020 15:30:15	0	0	0
33	22/9/2020 15:30:32	1	0	0
34	22/9/2020 15:38:13	0	0	0
284	22/9/2020 16:30:09	1023	1023	1023
285	22/9/2020 16:30:14	1023	1023	1023
286	22/9/2020 16:30:29	1023	1023	1023

Figura 8.17: Inicio y Fin de guardado correspondiente al dispositivo A.

28	22/9/2020 15:30:04	0	0	0
29	22/9/2020 15:30:21	0	0	0
30	22/9/2020 15:30:38	0	0	0
283	22/9/2020 16:30:10	1023	1023	1023
284	22/9/2020 16:30:16	1023	1023	1023
285	22/9/2020 16:30:30	1023	1023	1023

Figura 8.18: Inicio y Fin de guardado correspondiente al dispositivo B.

29	22/9/2020 15:30:07	0	0	0
30	22/9/2020 15:30:27	0	0	0
31	22/9/2020 15:30:45	1	0	0
286	22/9/2020 16:30:04	1023	1023	1023
287	22/9/2020 16:30:19	1023	1023	1023
288	22/9/2020 16:30:24	1023	1023	1023

Figura 8.19: Inicio y Fin de guardado correspondiente al dispositivo C.

Los tres dispositivos tienen su hoja de Excel correspondiente, el guardado arroja las siguientes cantidades:

- *XBee A*: [15:30:15 - 16:30:09] ha guardado un total de 253 frames.
- *XBee B*: [15:30:04 - 16:30:10] ha guardado un total de 256 frames.
- *XBee C*: [15:30:07 - 16:30:04] ha guardado un total de 258 frames.

Por lo tanto hay un excedente de 13 frames para *XBee A*, 16 para *XBee B* y 18 para *XBee C*. A medida que aumenta el tiempo para cada envío presenta un aumento en los frames debido a la falta de precisión en el timing, generado en conjunto por los *XBee* y la adquisición realizada por el programa.

Time	ADC0A	ADC1A	ADC2A	Time	ADC0B	ADC1B	ADC2B	Time	ADC0C	ADC1C	ADC2C
22/9/2020 15:30:15	0	0	0	22/9/2020 15:30:04	0	0	0	22/9/2020 15:30:07	0	0	0
22/9/2020 15:30:32	1	0	0	22/9/2020 15:30:21	0	0	0	22/9/2020 15:30:27	0	0	0
22/9/2020 15:30:13	0	0	0	22/9/2020 15:30:38	0	0	0	22/9/2020 15:30:45	1	0	0
22/9/2020 15:35:01	0	1023	0	22/9/2020 15:35:04	1023	1023	1023	22/9/2020 15:35:10	1023	1023	1023
22/9/2020 15:35:16	1023	1023	1023	22/9/2020 15:35:09	1023	1023	1023	22/9/2020 15:35:25	1023	1023	1023
22/9/2020 15:35:30	1023	1023	1023	22/9/2020 15:35:24	1023	1023	1023	22/9/2020 15:35:40	1023	1023	1023
22/9/2020 15:40:01	1023	0	1023	22/9/2020 15:40:02	1023	0	1023	22/9/2020 15:40:10	1	1	1
22/9/2020 15:40:06	0	0	0	22/9/2020 15:40:07	0	0	0	22/9/2020 15:40:16	1	1	1
22/9/2020 15:40:21	0	0	0	22/9/2020 15:40:22	0	0	0	22/9/2020 15:40:30	1	1	2
22/9/2020 15:45:06	1023	1023	0	22/9/2020 15:45:07	1023	1023	0	22/9/2020 15:45:01	1	1	1
22/9/2020 15:45:21	1023	1023	1023	22/9/2020 15:45:22	1023	1023	1023	22/9/2020 15:45:16	1023	1023	1023
22/9/2020 15:45:26	1023	1023	1023	22/9/2020 15:45:27	1023	1023	1023	22/9/2020 15:45:30	1023	1023	1023
22/9/2020 15:50:12	0	0	0	22/9/2020 15:50:12	0	0	0	22/9/2020 15:50:06	1	1	3
22/9/2020 15:50:26	0	0	0	22/9/2020 15:50:27	0	0	0	22/9/2020 15:50:21	1	1	1
22/9/2020 15:50:41	0	0	0	22/9/2020 15:50:42	0	0	0	22/9/2020 15:50:36	1	1	1
22/9/2020 15:55:02	907	762	0	22/9/2020 15:55:03	937	787	0	22/9/2020 15:55:06	761	906	760
22/9/2020 15:55:17	908	763	761	22/9/2020 15:55:18	937	787	786	22/9/2020 15:55:12	761	906	760
22/9/2020 15:55:32	908	763	761	22/9/2020 15:55:32	937	787	786	22/9/2020 15:55:26	761	906	760
22/9/2020 16:00:02	907	1023	761	22/9/2020 16:00:03	1023	1023	787	22/9/2020 16:00:12	1023	1023	1023
22/9/2020 16:00:17	1023	1023	1023	22/9/2020 16:00:18	1023	1023	1023	22/9/2020 16:00:26	1023	1023	1023
22/9/2020 16:00:22	1023	1023	1023	22/9/2020 16:00:23	1023	1023	1023	22/9/2020 16:00:32	1023	1023	1023
22/9/2020 16:05:08	0	0	1	22/9/2020 16:05:08	0	0	0	22/9/2020 16:05:02	1	1023	1023
22/9/2020 16:05:22	0	0	0	22/9/2020 16:05:23	0	0	0	22/9/2020 16:05:17	2	3	1
22/9/2020 16:05:37	0	0	0	22/9/2020 16:05:38	0	0	0	22/9/2020 16:05:32	4	3	1
22/9/2020 16:10:13	1023	1023	1023	22/9/2020 16:10:14	1023	1023	1023	22/9/2020 16:10:03	1023	1023	1
22/9/2020 16:10:28	1023	1023	1023	22/9/2020 16:10:28	1023	1023	1023	22/9/2020 16:10:08	1023	1023	1023
22/9/2020 16:10:42	1023	1023	1023	22/9/2020 16:10:43	1023	1023	1023	22/9/2020 16:10:23	1023	1023	1023
22/9/2020 16:15:13	0	0	0	22/9/2020 16:15:14	0	0	0	22/9/2020 16:15:08	1	1	1
22/9/2020 16:15:18	0	0	0	22/9/2020 16:15:19	0	0	0	22/9/2020 16:15:23	1	1	1
22/9/2020 16:15:33	0	0	0	22/9/2020 16:15:34	0	0	0	22/9/2020 16:15:28	1	1	1
22/9/2020 16:20:04	1023	1023	0	22/9/2020 16:20:05	1023	1023	1023	22/9/2020 16:20:13	1023	1023	1023
22/9/2020 16:20:18	1023	1023	1023	22/9/2020 16:20:19	1023	1023	1023	22/9/2020 16:20:28	1023	1023	1023
22/9/2020 16:20:33	1023	1023	1023	22/9/2020 16:20:34	1023	1023	1023	22/9/2020 16:20:43	1023	1023	1023
22/9/2020 16:25:09	0	0	0	22/9/2020 16:25:10	0	0	0	22/9/2020 16:25:04	1	1	1
22/9/2020 16:25:24	0	0	0	22/9/2020 16:25:25	0	0	0	22/9/2020 16:25:19	0	0	0
22/9/2020 16:25:38	0	0	0	22/9/2020 16:25:39	0	0	0	22/9/2020 16:25:33	1	1	1
22/9/2020 16:30:09	1023	1023	1023	22/9/2020 16:30:10	1023	1023	1023	22/9/2020 16:30:04	1023	1023	1023
22/9/2020 16:30:14	1023	1023	1023	22/9/2020 16:30:16	1023	1023	1023	22/9/2020 16:30:19	1023	1023	1023
22/9/2020 16:30:29	1023	1023	1023	22/9/2020 16:30:30	1023	1023	1023	22/9/2020 16:30:24	1023	1023	1023

Figura 8.20: Resultados obtenidos con tres End Devices.

La Figura 8.20 muestra en detalle los frames para el rango de una hora, si bien este caso arroja un número mucho mayor de frames guardados comprobamos que lo almacenado es correcto y corresponde a lo que esperaba donde nuevamente entre cada franja de tiempo (5 minutos), los valores se estabilizan a lo esperado, no presentan ninguna variación aleatoria. Si bien el aumento en el parámetro IR de los dispositivos genero que aumentarían la cantidad de datos almacenados, en la PC no tenemos ningún problema en cuanto al espacio de almacenamiento disponible, aunque si se

cuenta con un rango acotado de memoria debe tenerse en cuenta.

Por último la RED D, configurada en mesh con dos dispositivos finales y dos routers, fue probada con el programa que se utilizó en la red B ya que recibimos frames por parte de 2 dispositivos. La única diferencia es que al tener saltos en la transmisión, provoca un aumento de latencia por lo que el período de envío de muestras que se había utilizado para dos End Devices de 5 segundos, se aumentó a 15 segundos para asegurar la recepción.

8.6. Envío de Comando Remoto

Nuevamente en este apartado volvemos a recurrir a la metodología adoptada en los puntos anteriores, en un rango de tiempo se enviaron los comandos para controlar las salidas digitales desde el Coordinador a través del Instrumento Virtual creado para dicho propósito. La siguiente tabla muestra en detalle de los comandos:

Tiempo[Hora:Minutos]	DO3	DO5	DO7
12:15	0	0	1
12:20	0	1	0
12:25	0	1	1
12:30	1	0	0
12:35	1	0	1
12:40	1	1	0
12:45	1	1	1
12:50	0	0	0
12:55	0	0	1
13:00	0	1	0
13:05	0	1	1
13:10	1	0	0
13:15	1	0	1

Cuadro 8.4: Valores forzados para cada rango de tiempo para tres salidas digitales.

En el Capítulo 8 se ha explicado en detalle como funciona el VI Generador de

Comando, con esta herramienta creamos el frame necesario para que el dispositivo remoto interprete correctamente lo que deseamos.



Figura 8.21: Panel de configuración para envío de comando remoto.

La Figura 8.21 muestra el panel frontal para que podamos utilizar el VI con un ejemplo de aplicación, en él se carga la dirección del dispositivo receptor, el comando con su correspondiente valor e ID del frame. En el Capítulo 6 se encuentran los comandos de Configuración de Pin, estos son los que se utilizan aquí para modificar las salidas que deseamos, en este caso:

- *Digital Output 3*: comando=D3, 4=LOW,5=HIGH
- *Digital Output 5*: comando=D5, 4=LOW,5=HIGH
- *Digital Output 7*: comando=D7, 4=LOW,5=HIGH

La imagen anterior indica que se trata del dispositivo remoto cuya dirección es: 00 13 A2 00 40 3C BF DE, como los últimos dos bytes corresponden a la dirección corta de 16 bits, como estamos utilizando la dirección larga de 64 bits, completamos con FF FE. Se está modificando la salida del Pin 3 por ello el comando es D3 y cuyo valor es 4 lo que indica que queremos apagar la salida. Finalmente se asigna un ID de 1.

Time	DOA3	DOA5	DOA7
29/9/2020 12:15:07	0	0	1
29/9/2020 12:20:19	0	1	0
29/9/2020 12:25:03	0	1	1
29/9/2020 12:30:16	1	0	0
29/9/2020 12:35:05	1	0	1
29/9/2020 12:40:21	1	1	0
29/9/2020 12:45:05	1	1	1
29/9/2020 12:50:18	0	0	0
29/9/2020 12:55:04	0	0	1
29/9/2020 13:00:12	0	1	0
29/9/2020 13:05:13	0	1	1
29/9/2020 13:10:15	1	0	0
29/9/2020 13:15:04	1	0	1

Figura 8.22: Resultados obtenidos a partir de la recepción luego de enviar el comando.

En la Figura 8.22 se observan los frames con el momento exacto donde el cambio se establece, teniendo en cuenta que algunas de las transiciones requieren mas de un comando (por ejemplo pasar de 001 a 010) para establecer lo deseado, se obtuvo un máximo de 21 segundos para el estado 110 lo cual es tolerable para nuestro proposito. Esta prueba se realizo con un solo dispositivo, sin embargo se lograron resultados similares al enviar el comando cuando se tienen dos y tres End Devices, solo aumentaron algunos segundos más en establecerse.

8.7. Envío de Alarma

Uno de las características deseadas desde un principio para este proyecto fue la detección de situaciones anormales en cuanto a las variables y posterior aviso del evento a la persona que corresponda. En este caso se envía un mail indicando la alarma detectada. Para ello se han realizado las pruebas utilizando los tres ADC, se ha fijado un valor máximo como ejemplo de 900. Cuando el programa recibe el frame y separa cada magnitud en su respectiva variable, detecta si este umbral es sobrepasado o no, en caso de que esto ocurra realiza el envío de mail a una dirección

de correo que el usuario deberá ingresar para dar el aviso. Se ha variado el potencial en las entradas analógicas para generar la alarma, a continuación se encuentra la tabla correspondiente a los horarios en que se han generado:

Tiempo[Hora:Minutos]	ADC0A	ADC1A	ADC2A
20:00	Alarma	-	-
20:05	-	Alarma	-
20:10	-	-	Alarma
20:15	Alarma	Alarma	-
20:20	Alarma	-	Alarma
20:25	-	Alarma	Alarma
20:30	-	-	Alarma
20:35	-	Alarma	-
20:40	Alarma	-	-
20:45	Alarma	Alarma	-
20:50	Alarma	-	Alarma
20:55	Alarma	Alarma	Alarma
21:00	-	Alarma	Alarma

Cuadro 8.5: Rango horario de generación de alarma.

Dicho esto, veamos primero los frames almacenados por nuestro programa para ver si efectivamente los datos se corresponden a lo que se había planeado:

Time	ADCOA	ADC1A	ADC2A
2/10/2020 20:00:07	1023	0	0
2/10/2020 20:05:03	1	1023	0
2/10/2020 20:10:05	0	0	1023
2/10/2020 20:15:06	1023	1023	0
2/10/2020 20:20:23	1023	9	1023
2/10/2020 20:25:08	0	1023	1023
2/10/2020 20:30:06	0	0	1023
2/10/2020 20:35:05	1	1023	0
2/10/2020 20:40:05	1023	15	2
2/10/2020 20:45:02	1023	1023	10
2/10/2020 20:50:05	1023	25	1023
2/10/2020 20:55:08	1023	1023	1023
2/10/2020 21:00:06	12	1023	1023

Figura 8.23: Resultados obtenidos a partir de la recepción luego de enviar el comando.

La Figura 8.23 contiene los valores leídos por los tres ADC, podemos ver que cada uno de los datos se han leído correctamente en cada rango horario con una diferencia de algunos segundos, particularmente el horario 20:20 presenta la mayor diferencia con 24 segundos. Ahora bien, se ha dicho que como ejemplo se tomo un valor máximo de ADC=900, correctamente los valores se han establecido en 1023 como se esperaba, por lo cual las alarmas concretamente fueron generadas. Veamos a continuación, la casilla de mails donde se han recibido los avisos.

Principal	Social	Promociones
<input type="checkbox"/> ☆ gastonmaciel2 6	ADC2 En alarma. - Alarma detectada.	2 oct.
<input type="checkbox"/> ☆ gastonmaciel2 2	ADC1 y ADC2 En alarma. - Alarma detectada.	2 oct.
<input type="checkbox"/> ☆ gastonmaciel2 6	ADC1 En alarma. - Alarma detectada.	2 oct.
<input type="checkbox"/> ☆ gastonmaciel2	ADC0, ADC1 y ADO2 En alarma. - Alarma detectada.	2 oct.
<input type="checkbox"/> ☆ gastonmaciel2 2	ADC0 y ADC2 En alarma. - Alarma detectada.	2 oct.
<input type="checkbox"/> ☆ gastonmaciel2 2	ADC0 y ADC1 En alarma. - Alarma detectada.	2 oct.
<input type="checkbox"/> ☆ gastonmaciel2 5	ADC0 En alarma. - Alarma detectada.	2 oct.

Figura 8.24: Mails recibidos por alarma en la variable ADC2.

En la figura anterior se muestra la casilla de mails que se ha dispuesto como destinatario de la pantalla principal, contiene efectivamente los correos recibidos, veamos con mas detalle cada uno de ellos:

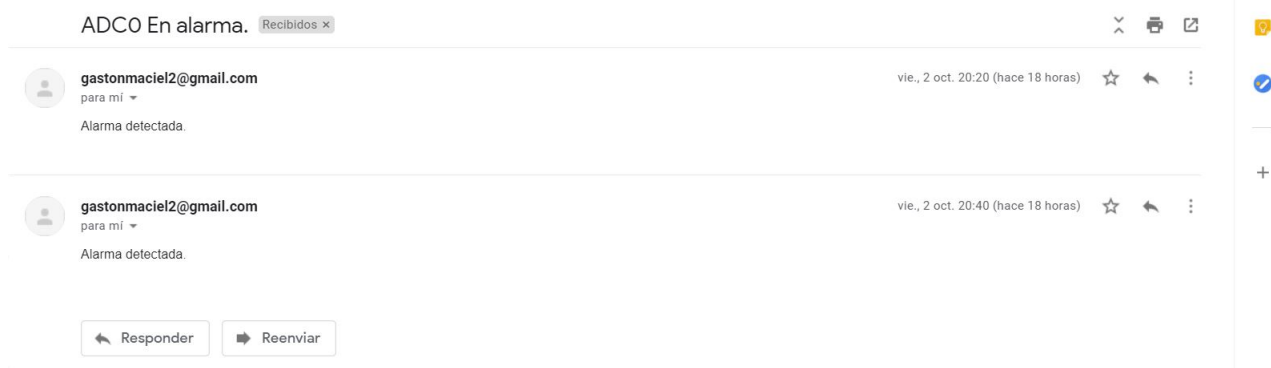


Figura 8.25: Mails recibidos por alarma en la variable ADC0.

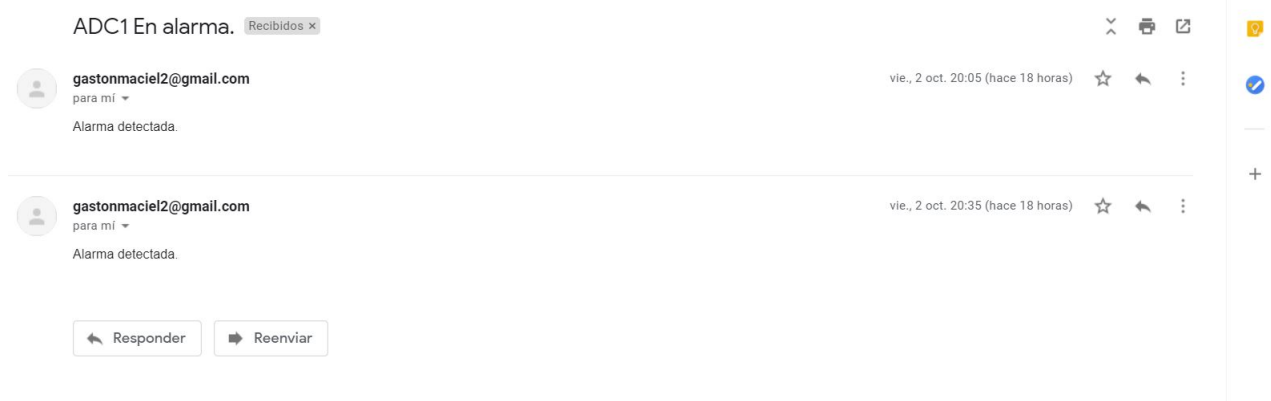


Figura 8.26: Mails recibidos por alarma en la variable ADC1.

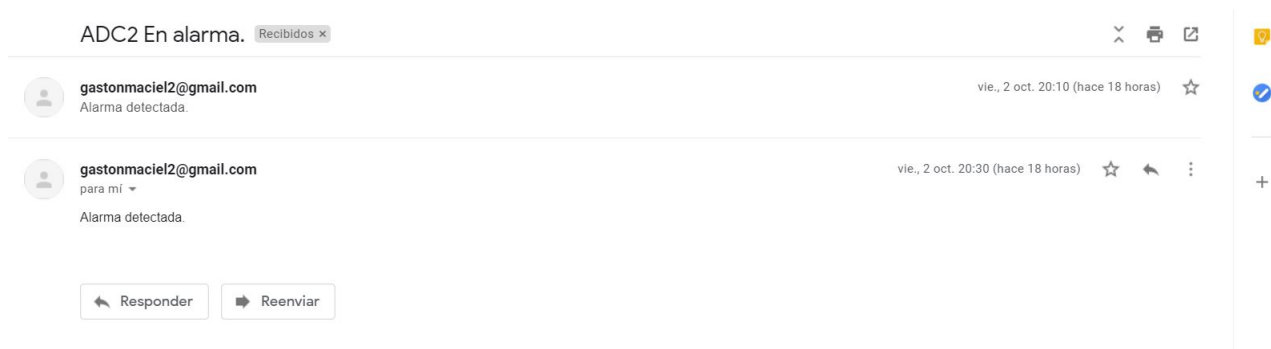


Figura 8.27: Mails recibidos por alarma en la variable ADC2.

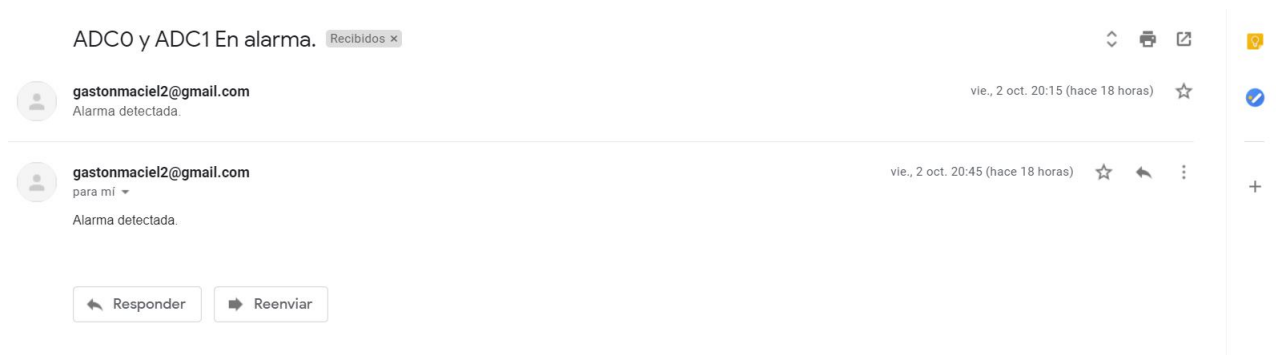


Figura 8.28: Mails recibidos por alarma ADC0 Y ADC1.

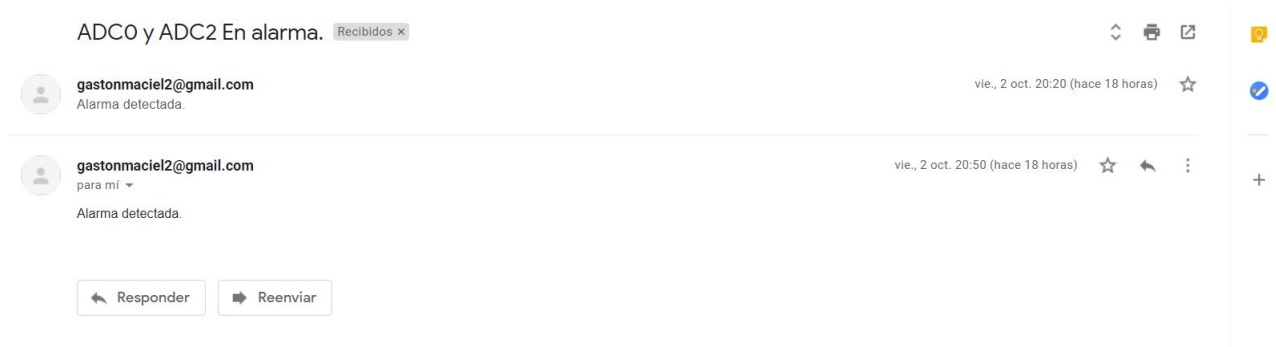


Figura 8.29: Mails recibidos por alarma ADC0 Y ADC2.

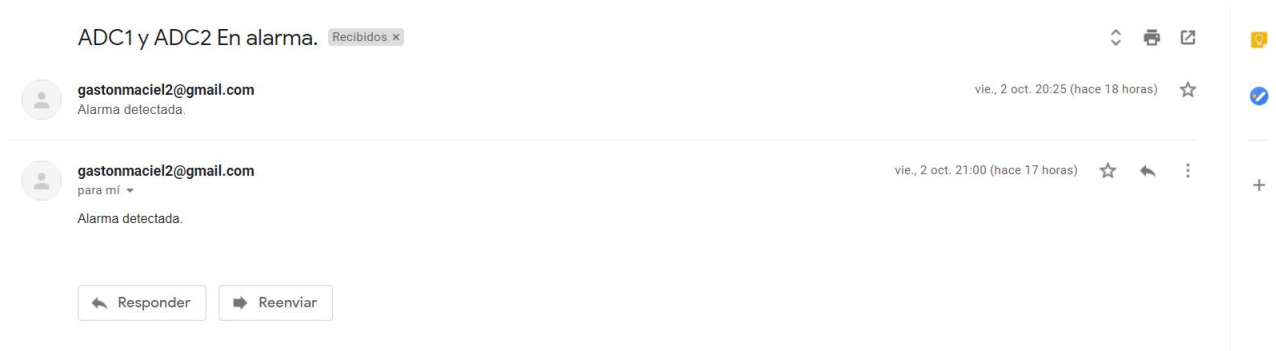


Figura 8.30: Mails recibidos por alarma ADC1 Y ADC2.

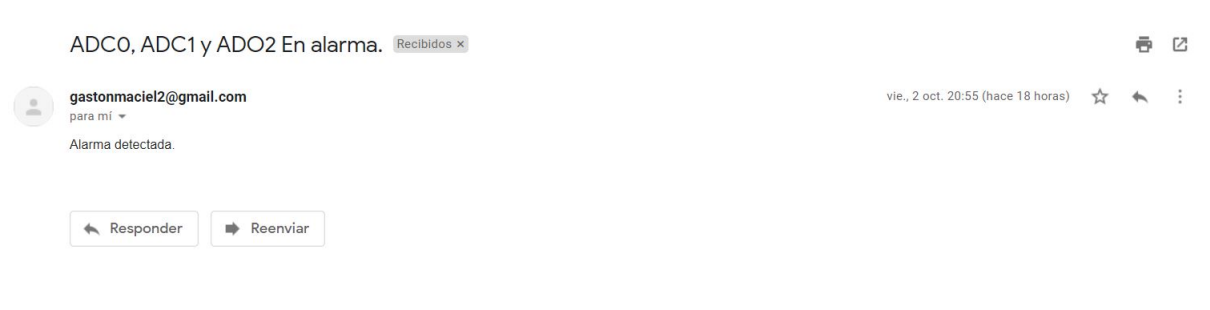


Figura 8.31: Mails recibidos por alarma ADC0, ADC1 Y ADC2.

Luego de revisar la casilla se comprueba que han llegado todas las alarmas generadas, el horario no especifica los segundos aun así todas han llegado en el rango de minutos esperados y no se ha perdido ningún dato de alarma. Al realizar las pruebas con más dispositivos arrojo un buen funcionamiento sin embargo se debe aumentar el tiempo IR de los dispositivos finales para darle tiempo al programa en el procesamiento de alarmas, para el caso de dos dispositivos 5 segundos arrojo excelentes resultados mientras que para tres dispositivos finales se aumento el tiempo de envío periódico de variables de 15 a 20, con eso se mantuvo el funcionamiento correctamente.

8.8. Lecturas ADC 0-10V/4-20mA

Comprobado el correcto funcionamiento de la recepción, se ingresa una tension y una corriente controlada en las entradas analógicas y se verifica la fiabilidad de la lectura en los modos 0-10V y 4-20mA. En este punto debemos hacer una aclaración, debido a que no hay resistencias de 1.7k Ω SMD, se utilizó un valor de 1.8k Ω en el divisor resistivo que esta conectado directamente al ADC, lo cual modifica levemente los niveles en la entrada del módulo.

Para el caso del modo 0-10V, veamos como esta resistencia afecta a la entrada ADC luego de pasar por el divisor resistivo compuesto por las resistencias de 15k Ω y el divisor en cuestión(V_{in}):

Tensión de entrada [Volt]	$V_{in}(R=1.7k\Omega)$	$V_{in}(R=1.8k\Omega)$
0	0	0
1	0.12	0.11
2	0.24	0.23
3	0.37	0.36
4	0.49	0.47
5	0.61	0.59
6	0.71	0.71
7	0.86	0.83
8	0.98	0.95
9	1.11	1.07
10	1.2	1.18

Cuadro 8.6: Comparativas V_{in} para cada resistencia.

Finalmente, sabiendo esto se procedió a ingresar el rango dinámico propuesto de 0 a 10 V y ver a través del BackUp los resultados que leen los ADC, los tres están conectados al mismo potencial para evaluarlos bajo las mismas condiciones por medio de una fuente regulable calibrada y colocando el jumper en la posición 1-2 para utilizar este modo de trabajo. Se han tomado al menos 10 muestras en cada nivel de voltaje para cada entrada ADC y realizado un promedio, obteniendo los siguientes resultados experimentales los cuales comparamos con los teóricos:

Tensión de entrada [Volt]	Teórico	Medido
0	0	0
1	93	93
2	200	200
3	303	300
4	400	400
5	502	500
6	608	601
7	707	704
8	809	801
9	912	900
10	1005	1002

Cuadro 8.7: Cuentas correspondientes a los ADC para entrada 0-10V.

Si bien solo hay una columna con el valor medido experimentalmente, los tres ADC de la placa respondieron de la misma manera, en cuanto a las cuentas podemos decir que el rango de variación es mínimo teniendo en cuenta el ripple de la fuente y posibles anomalías a la hora de aplicar la tensión al circuito de entradas analógicas, ya que se fue variando el nivel de voltaje a mano cubriendo todo el rango.

En cuanto a la entrada 4-20 mA, el lazo actuará sobre la resistencia de carga de 237Ω , en este caso se utiliza un valor de 220Ω ya que no contamos con la resistencia deseada. Esto provoca un cambio en la diferencia de potencial en el nodo correspondiente, que se mostrará a continuación:

Corriente de entrada[mA]	$V=I.237\Omega$)[V]	$V=I.220\Omega$)[V]
4	0.94	0.88
6	1.42	1.32
8	1.89	1.76
10	2.37	2.2
12	2.84	2.64
14	3.31	3.08
16	3.79	3.52
18	4.26	3.96
20	4.74	4.4

Cuadro 8.8: Comparativas caída de tension en resistencia de carga.

Nuevamente realizamos la comparativa entre el valor teórico calculado (utilizando un valor de resistencia de carga de 220Ω) y el resultado experimental obtenido a la entrada de los ADC utilizando un simulador Fluke calibrado con potencia de bucle y colocando los dos jumpers en posición 2-3 para utilizar el modo 4-20mA:

Corriente de entrada [mA]	Teórico	Medido
4	170	170
6	264	266
8	349	350
10	443	445
12	528	530
14	622	622
16	707	709
18	801	805
20	886	888

Cuadro 8.9: Cuentas correnpondientes a los ADC para 4-20mA.

Nuevamente los resultados son positivos ya que se acercan a los cálculos teóricos. En ambos casos podemos adjudicar esta variación no sólo a la fluctuaciones provocadas por el usuario o los equipos empleados aquí a la hora de establecer los

niveles de tensión o corriente sino que también, se están utilizando resistencias con una tolerancia mayor al 1 % que en teoría deberíamos utilizar aquí, sin embargo por los motivos de público conocimiento(COVID) fueron muy complicadas de conseguir de manera rápida.

8.9. Conclusiones

La primer version de este proyecto logro sus objetivos, se han puesto en marcha y probado cada una de las características propuestas desde el principio. El proyecto logró detectar alarmas, enviar comandos remotos para activar posibles actuadores, recepcionar correctamente las tramas, enviar correos electrónicos y crear un BackUp con las mediciones en fecha y hora tal cual llegan al dispositivo Maestro. Los circuitos de entrada digital funcionaron correctamente logrando detectar todos los cambios de estado en las entradas, en cuanto al circuito de entradas analógicas concluimos que los resultados obtenidos son muy buenos ya que se encuentran muy cerca de los valores teóricos calculados para ambos modos(0-10V/4-20mA), teniendo en cuenta que no se utilizaron aquí resistencias del 1 %. La placa trabajando con todas sus entradas y salidas activadas arrojo un consumo bajo, si bien se esperaba poco consumo por parte del XBee, el resto de los componentes no apporto un consumo significativo extra de corriente.

Aumentar la cantidad de nodos implica que la recepción tenga mas trabajo, lo cual aumenta la latencia propia del programa en LabVIEW, por lo que aumentar la cantidad de nodos conectados a sensores implica que debamos aumentar el tiempo del parámetro IR de los XBee(aumentar el periodo de envío de muestras periódico) para darle tiempo a la aplicación de procesar correctamente los datos, con tres dispositivos utilizamos 15 segundos entre cada muestra enviada, lo cual resulta aceptable, ya que en sistemas mas complejos por lo general las transmisiones se realizan cada x minutos y no cada x segundos.

Tras realizar un análisis en detalle acerca de ZigBee, se llega a la conclusión de que esta tecnología fue correctamente seleccionada, por su versatilidad, bajo costo, bajo consumo y de fácil configuración e integración al proyecto logrando un alcance mayor al esperado en las transmisiones(teniendo en cuenta que utilizamos un módulo

estándar sin antena con mayor ganancia). La facilidad para crear la red en cuestión y la capacidad de autoruteo por parte de los módulos agilizó muchísimo los tiempos. Debe tenerse en cuenta la baja tasa de transmisión así como también si se requiere un *timming* preciso ya que ZigBee no es tan ameno a esas características.

Se aprendió a utilizar LabVIEW desde cero, habiendo adquirido los conocimientos para poder crear cualquier tipo de sistema a partir de ahora, la estructura API que ofrece ZigBee ayudo a reducir la complejidad a la hora del diseño ya que nos entrega frames que previamente podemos analizar y a través de eso, realizar nuestro programa. Se han descubierto muchas herramientas interesantes además de las que aquí se han utilizado, ecuaciones diferenciales, integrales y derivadas, transformadas (entre ellas Laplace y Fourier), interpolación y aproximación, álgebra lineal, análisis de señal y muchas más. LabVIEW logro cumplir con las expectativas y será una herramienta más en el ámbito profesional del que escribe este texto. Terminado el programa se realiza un instalador para que cualquiera que quiera utilizar el programa pueda instalarlo en su PC sin necesidad de contar con LabVIEW aunque deberá contar con sistema operativo Windows.

La manera en que se han ido cumpliendo los objetivos y avanzando en el proyecto fueron de gran aporte formativo y personal en cuanto a como se debe proceder a la hora de realizar un proyecto, habiendo pasado por todas las etapas que se mencionaron al comienzo, diseño de circuitos, hardware y software pudiendo aplicar en ellos, los conocimientos adquiridos a lo largo de estos años en la Universidad Nacional de San Martín y en la experiencia laboral.

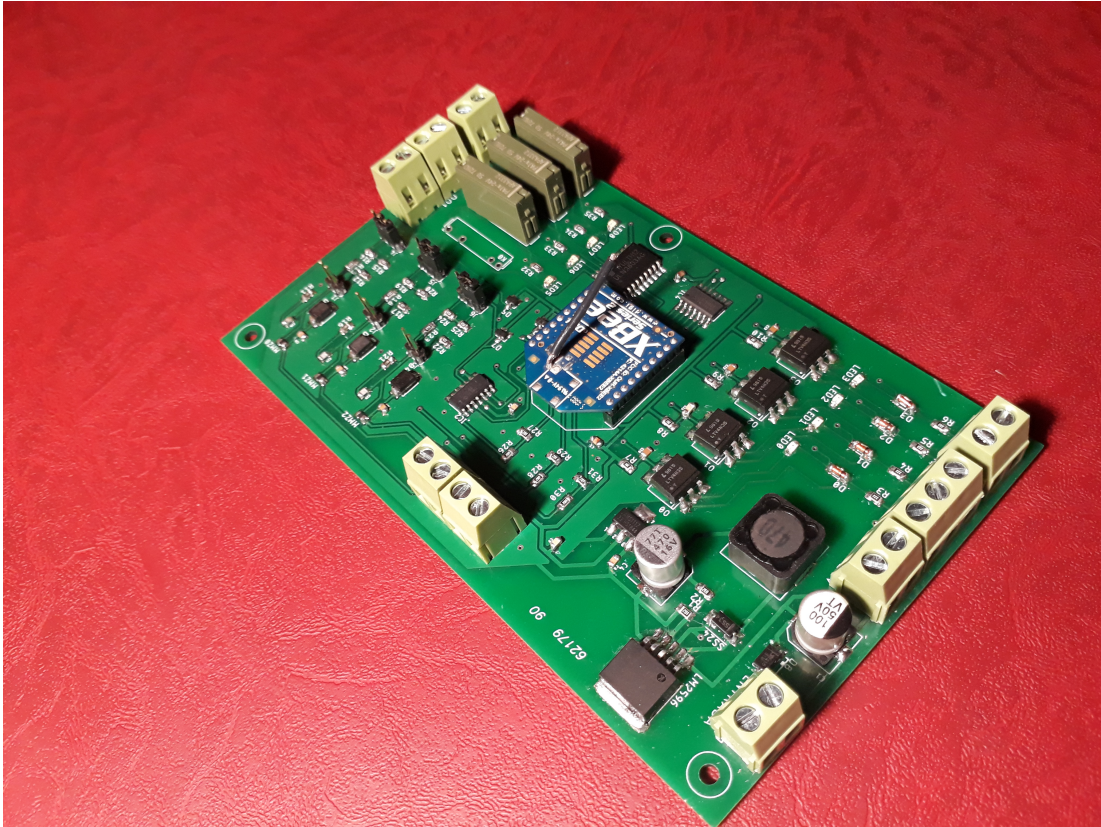


Figura 8.32: Hardware final.

La Figura 8.32 muestra el Hardware finalizado, con algunos cambios respecto al modelo diseñado, principalmente en la fuente de entrada donde se han puesto otro valor de capacitor de entrada y de inductor para ahorrar tiempo. Esto podría traer efectos de rizado sin embargo son mitigados por el regulador lineal que le sigue a esta etapa, por lo que no afectan al funcionamiento en absoluto.

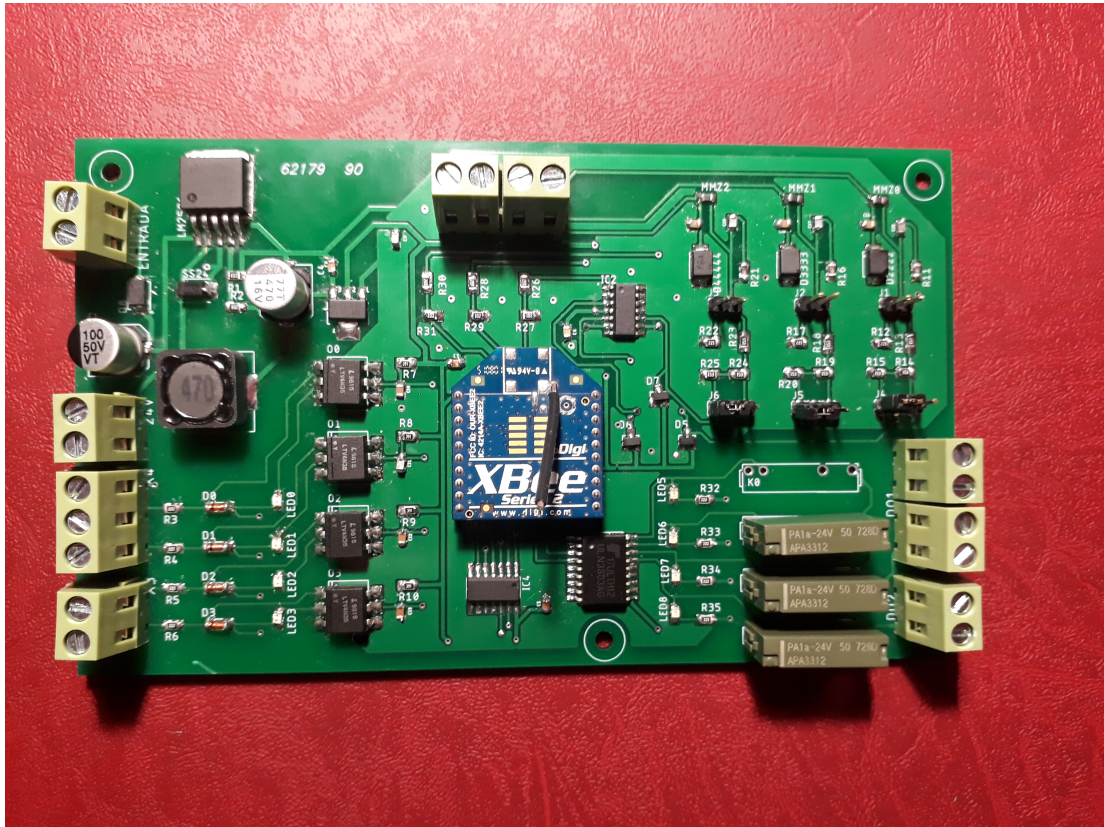


Figura 8.33: Hardware final.

La Figura 8.33 muestra una vista superior, donde podemos ver los distintos circuitos de funcionamiento, a la izquierda arriba vemos el circuito de alimentación con ambos reguladores, abajo a la izquierda las entradas digitales con sus respectivas borneras, arriba a la derecha el circuito de entradas analógicas con las borneras arriba, abajo a la derecha las salidas para actuadores, por último en el centro se ubica el corazón del proyecto, el módulo XBee.

8.10. Posibles Trabajos Futuros

Dentro del diseño actual, aún pueden modificarse diferentes partes del proyecto para agregar características y corregir distintos aspectos y problemas que han surgido a lo largo de la elaboración.

- *Programación orientada a objetos:* se han repasado los conceptos empleados en cuanto a la programación hecha para este proyecto, sin embargo este es una pequeña parte y el comienzo de algo mucho mayor. Hemos hecho las pruebas utilizando tres nodos adquiriendo datos de los sensores, sin embargo en el futuro se pretende utilizar una gran cantidad de nodos, por lo que no podemos simplemente agregar estructuras case con la dirección del nuevo nodo para diferenciar sus variables de las demás, sería poco eficiente. Como dije, este fue el puntapié inicial para una idea de sistema que tengo en mente, por lo que requiere otro enfoque si se requiere tener una solución que abarque una cantidad mucho mayor de dispositivos. Este enfoque se basará en la Programación Orientada a Objetos. Este paradigma de programación, organiza el código en entidades conocidas como **CLASES**, en base a ella se crean los objetos que heredan las características de dicha clase, con esto se busca crear una única clase de dispositivo con los atributos y funciones que hemos desarrollado para este proyecto. De esta manera, cuando se requiera agregar un dispositivo nuevo al programa, solo debemos crear un objeto de la clase dispositivo y agregarlo, ya que al hacerlo directamente heredará las características deseadas. Permite reducir la cantidad de Instrumentos Virtuales en el proyecto, el desarrollador puede identificar de manera sencilla las áreas en las que desean agregar funciones especiales reduciendo la posibilidad de cometer errores en otros sectores de código no relacionado. Si bien cuento con sólidos conocimientos en programación, emplear esta técnica no es sencilla, teniendo en cuenta que se ha aprendido como utilizar LabVIEW para cumplir con el propósito, utilizar OOP(Object Oriented Programming) requería un aumento considerable de tiempo para poder manejarlo con facilidad, por lo que se escogió una programación estructurada similar a C para empezar.
- *Evaluación de cambios, tecnología RF y procesamiento:* al comienzo, se intentó minimizar el tiempo de desarrollo para poder avanzar con los tiempos estipulados, utilizar el módulo XBee ahorra al desarrollador cuestiones como ADC, PWM, Digital Input, circuito transmisor, etc, sin embargo, lo que aquí se quiere lograr al final, es que el Hardware sea mucho más poderoso, empleando directamente un microcontrolador que se comunique con el XBee vía puerto serie,

pudiendo utilizar aún el modo API. Sin embargo en este escenario, se presentan nuevas oportunidades en cuanto a lo que podemos hacer, por ejemplo utilizar EEPROM, protocolos serie, DAC, DMA, ADC de mayor resolución, PWM.. con esto la funcionalidad del proyecto aumentaría considerablemente. Evaluar nuevamente la posibilidad en cuanto a los módulos XBee para aumentar el alcance, ya que disponemos de otras antenas y placas con micro mucho mas poderoso al utilizado en este proyecto. Se deben rehacer los cálculos financieros para ajustar los cambios y que aún sea un proyecto redituable.

- *Reducción de espacio en la placa:* para este proyecto no se aprovecho el espacio del cobre para realizar la placa final, sino que se dio una gran libertad en la distancia entre componentes para poder ser soldados de manera más cómoda, sin embargo claro está, que si se requieren fabricar una gran cantidad de placas, ahorrar en el espacio también implica un beneficio en la economía del proyecto ya que podremos imprimir una mayor cantidad en el mismo espacio.
- *Desarrollo de Hardware para dispositivo Maestro:* en este caso no se centro el desarrollo en el dispositivo maestro, el cual en un principio solo requiere de un puerto serie y su correspondiente voltaje de operación, por lo cual hemos utilizado un modulo comercial estándar para cumplir dicha función, sin embargo en un futuro cercano se pretende realizar un Hardware acorde para el dispositivo Maestro, agregando alguna funcionalidad en sus pines ademas del puerto serie.
- *Cambio de niveles:* se ha desarrollado y calculado todo alrededor de una tension industrial típica de 24 V, sin embargo se harán los cálculos para poder utilizar este proyecto en tensiones mas chicas como 12 V y 5 V, ya que existen una infinidad de sensores y actuadores que operan en ese nivel.
- *Posibilidad de control por medio de IP:* uno de los puntos no desarrollados pero que se tuvo en cuenta desde el comienzo fue realizar el control y seguimiento de variables vía web, dándole al usuario la posibilidad de ver la planta funcionando en cualquier lugar con acceso a internet, sin embargo el que suscribe carece de un servidor e IP fija. Como punto máximo este proyecto puede monitorearse desde otra PC , siempre y cuando esté en red con la computadora que contiene al maestro. No hemos incluido este apartado en el proyecto ya que lo considero

de poca relevancia si no brinda la posibilidad de verlo en cualquier parte como los sistemas mas sofisticados.

Apéndice A

Gerenciamiento del Proyecto

A.1. Anteproyecto

Previo al desarrollo del proyecto se describe el planeamiento que se realizó, siguiendo los lineamientos del Project Management.

Project charter

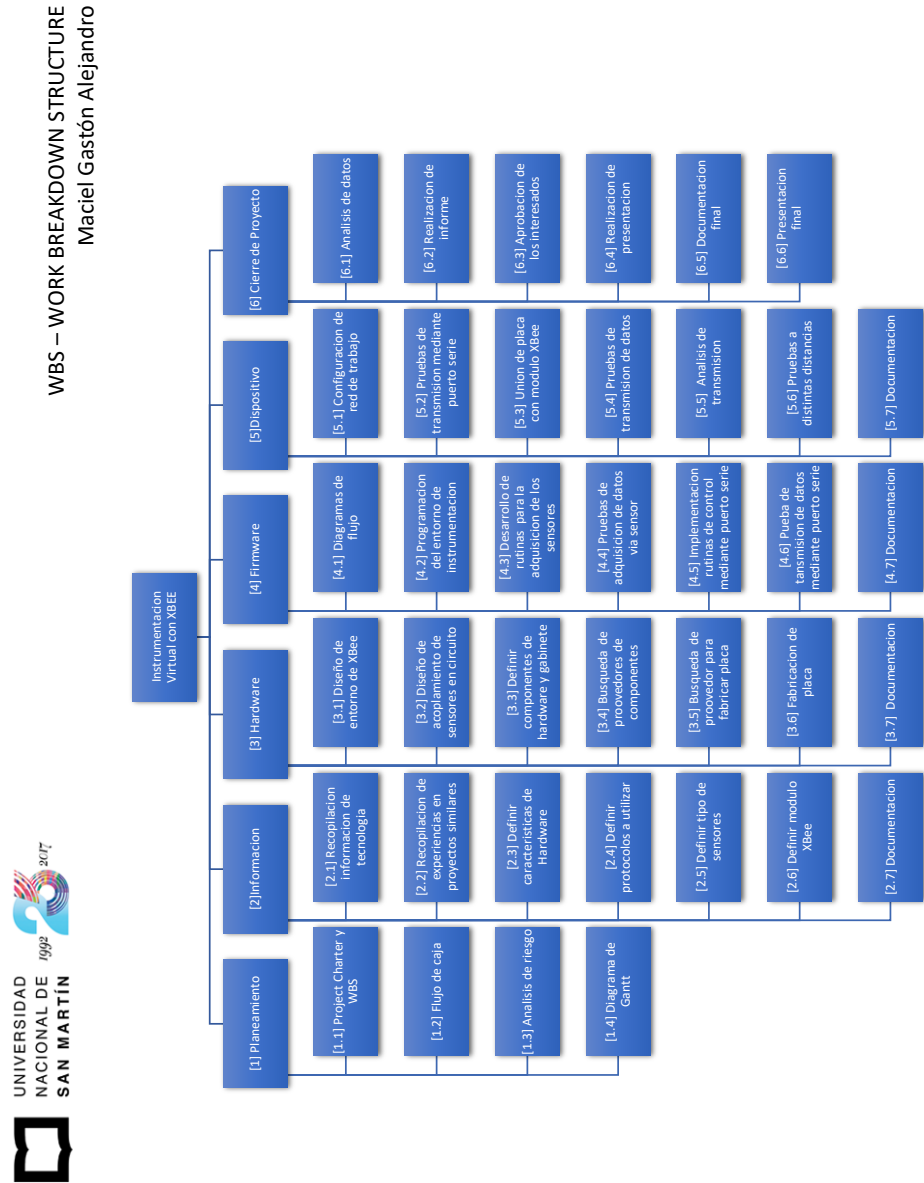
PROJECT CHARTER

Fecha	Nombre del Proyecto
XX/12/2019	Instrumentación Virtual con XBEE
Fecha de inicio del proyecto	Fecha tentativa de finalización de proyecto
01 de Septiembre de 2019	01 de Mayo de 2020
Área de Aplicación(Sector/Actividad)	
➤ Sensores y adquisición de datos, Telecomunicaciones, Control Industrial	
Objetivos del Proyecto (general y específicos)	
➤ Generales: Diseño de Hardware, implementación de Firmware, transmisión mediante XBee. ➤ Específicos: Determinación de componentes para Hardware, adquisición de datos, selección de módulos de tecnología XBee, programación de software, implementación de solución para comunicaciones a distancia con posibilidad de bajo consumo, implementación de señales digitales de control.	

PROJECT CHARTER

Project Scope Statement											
<ul style="list-style-type: none"> ➤ Propósito del Proyecto/Justificación: El propósito de este proyecto, es desarrollar un dispositivo que realice la lectura de variables analógicas y digitales, implementación de transmisión de datos de manera inalámbrica, utilización de sensores de tipo industrial principalmente orientado a dispositivos de 12 VDC o 24 VDC, diseño de salidas digitales para actuadores y realización de una interfaz de control para el usuario. ➤ Antecedentes: La necesidad de control y automatización se hace presente en las diferentes industrias, lo más común es encontrar soluciones a base de PLC's acompañados de un software de control. Sin embargo los costos de implementación de este tipo de desarrollos se han hecho muy elevados, y en muchos casos, solo las grandes empresas pueden tener el poder adquisitivo para contar con este tipo de tecnología, este proyecto brinda la posibilidad de poder adquirir una solución de características similares pero a un costo menos elevado. ➤ Metodología: Se investigará sobre la tecnología propuesta para lograr una buena performance basándose en la teoría, se realizará el hardware necesario para adquirir los datos de los sensores, configurar la red de trabajo, e implementar la comunicación requerida. Todos los avances serán documentados. ➤ Factibilidad técnica y económica: como primera medida se buscará realizar el hardware con componentes comerciales comunes y accesibles. Los módulos XBee serán lo más caro del desarrollo. ➤ Tiempos previstos: los tiempos previstos en la elaboración de Hardware será de dos meses, al igual que el Firmware. Se harán las pruebas pertinentes de funcionamiento por un mes y luego un mes para la documentación completa del proyecto. ➤ Análisis de riesgo: 											
<table border="1"> <thead> <tr> <th>RIESGO</th> <th>IMPACTO</th> </tr> </thead> <tbody> <tr> <td>Alcance de transmisión menor al esperado</td> <td>Alto</td> </tr> <tr> <td>Tiempo de desarrollo de Hardware</td> <td>Medio</td> </tr> <tr> <td>Tiempo de desarrollo de Firmware de Instrumentación</td> <td>Alto</td> </tr> <tr> <td>Adquisición de los componentes necesarios</td> <td>Medio</td> </tr> </tbody> </table>		RIESGO	IMPACTO	Alcance de transmisión menor al esperado	Alto	Tiempo de desarrollo de Hardware	Medio	Tiempo de desarrollo de Firmware de Instrumentación	Alto	Adquisición de los componentes necesarios	Medio
RIESGO	IMPACTO										
Alcance de transmisión menor al esperado	Alto										
Tiempo de desarrollo de Hardware	Medio										
Tiempo de desarrollo de Firmware de Instrumentación	Alto										
Adquisición de los componentes necesarios	Medio										
Plan de Comunicación											
<ul style="list-style-type: none"> ➤ Toda la comunicación será mediante correo electrónico, aclarando fecha, tema pertinente y comentarios. 											
Personas intervinientes	Firma										
Ing. Marcelo Romeo											
Maciel Gastón Alejandro											
Caprile Sergio											

Work breakdown structure



A.2. Gestión de Riesgos

Identificación de los riesgos:

- *Riesgo 1*: No disponer de tiempo suficiente para llevar a cabo el proyecto.
 - *Severidad(9)*: Implica que el proyecto no es terminado antes de la fecha de entrega.
 - *Probabilidad de ocurrencia(7)*: El responsable del proyecto tiene una jornada laboral de 8 horas o más dependiendo de la carga de trabajo.
- *Riesgo 2*: La planificación del proyecto no es adecuada.
 - *Severidad(8)*: Implica que el proyecto no es terminado antes de la fecha de entrega.
 - *Probabilidad de ocurrencia(5)*: El responsable tiene experiencia la realización de proyectos aunque este en particular abarca una gran cantidad de temas en su elaboración(Hardware, Software y Firmware).
- *Riesgo 3*: Demoras en la adquisición de los componentes electrónicos.
 - *Severidad(8)*: Implica que los tiempos planificados serán modificados, afectando al esquema de ejecución de tareas. Puede implicar mayores costos al no contar con algunos componentes.
 - *Probabilidad de ocurrencia(4)*: El principio la filosofía del proyecto lleva al desarrollador a pensar en circuitos sencillos y componentes presentes en el mercado local.
- *Riesgo 4*: Tiempo excesivo en la fabricación de la placa.
 - *Severidad(7)*: Implica que los tiempos planificados serán modificados, afectando al esquema de ejecución de tareas.
 - *Probabilidad de ocurrencia(2)*: El responsable del proyecto cuenta con un proveedor confiable y varias opciones disponibles, donde ya se han realizado en el pasado proyectos personales.

- *Riesgo 5:* Daño en placa o módulo XBee al realizar las pruebas de funcionamiento.
 - *Severidad(7):* Implicaría importantes retrasos en el análisis final a la hora de generar las conclusiones, debiendo reparar o sustituir el Hardware.
 - *Probabilidad de ocurrencia(2):* Cada bloque funcional del Hardware se analizará por separado y en detalle para evitar posibles errores de diseño, adjudicando los posibles errores a una mala manipulación por parte del usuario.

- *Riesgo 6:* Alcance de transmisión menor al esperado.
 - *Severidad(8):* Se espera con este desarrollo reemplazar los enlaces físicos por transmisiones inalámbricas, se espera un alcance aceptable para que el cambio realmente valga la pena.
 - *Probabilidad de ocurrencia(4):* Previo a comprar los módulos XBee, se realiza un análisis teórico y en base a experiencia de terceros para estimar el desempeño de los dispositivos.

Riesgo	Severidad	Ocurrencia	RPN	Severidad*	Ocurrencia*	RPN*
1	9	7	63	9	2	18
2	8	5	40	8	2	16
3	8	4	32	8	2	16
4	7	2	14			
5	7	2	14			
6	8	4	32	8	2	16

Cuadro A.1: Tabla de Gestión de Riesgos (RPN=Severidad x Ocurrencia).

Las medidas de mitigación serán adoptadas cuando el riesgo arroje un valor RPN mayor a 20. En la tabla anterior los valores con * corresponden a los riesgos luego de aplicar el plan adoptado en los riesgos con RPN mayor al máximo establecido:

- *Riesgo 1*: Se dedicaran al menos 10 horas semanales a la elaboración del proyecto y se coordinara periódicamente con el tutor a cargo. En caso de ser necesario se solicitaran días en el trabajo para poder atender cuestiones urgentes de la tesis.
 - *Severidad(9)*: Sin modificación.
 - *Probabilidad de ocurrencia(2)*: Se dispone de mas tiempo para realizar el proyecto.

- *Riesgo 2*: Para evitar problemas de tiempo se han dividido las tareas en pequeñas tareas dependiendo el tipo (Informe, Hardware, Software y Firmware). Con esto se pretende cumplir con el calendario estipulado previamente manteniendo el orden en tareas menos tediosas, o con menos carga horaria. Adicionalmente se utilizará un software de gerenciamiento de proyectos para realizar el seguimiento.
 - *Severidad(8)*: Sin modificación.
 - *Probabilidad de ocurrencia(2)*: Se reduce la posibilidad de una mala planificación y extension de tiempos.

- *Riesgo 3*: Se ha hecho una selección de al menos 5 proveedores distintos para la compra de componentes electrónicos, como se ha dicho se pretende utilizar componentes que son comunes en el mercado si es posible.
 - *Severidad(8)*: Sin modificación.
 - *Probabilidad de ocurrencia(2)*: Se reduce la posibilidad de retrasos por falta de stock al considerar componentes electrónicos de uso frecuente.

- *Riesgo 6*: Los fabricantes que utilizan ZigBee ofrecen una variedad de dispositivos con diferentes características, entre ellas el alcance de transmisión de los módulos.
 - *Severidad(8)*: Sin modificación.
 - *Probabilidad de ocurrencia(2)*: Se reduce la posibilidad de tener un alcance menor al cambiar de módulo.

A.3. Cash Flow

A continuación, el flujo de caja donde se hace referencia a las salidas y entradas netas de dinero en cuanto al proyecto expresado en dólares.

Año	2021	2022	2023	2024	2025
Ingreso de Caja	7.800,00	10.800,00	14.000,00	17.400,00	21.000,00
Ventas	6.300,00	8.600,00	11.000,00	13.500,00	16.100,00
Cantidad	30,00	40,00	50,00	60,00	70,00
Precio	210,00	215,00	220,00	225,00	230,00
Servicio de Mantenimiento Total	1.500,00	2.200,00	3.000,00	3.900,00	4.900,00
Mantenimiento de placa y Sistema	50,00	55,00	60,00	65,00	70,00
Egresos de Caja	7.400,00	9.850,00	13.450,00	17.550,00	21.450,00
Costos Directos					
Materia Prima Total	4.500,00	6.600,00	9.000,00	11.700,00	14.000,00
Insumos(por placa)	150,00	165,00	180,00	195,00	200,00
Operario de ensamble Total	1.500,00	2.400,00	3.500,00	4.800,00	6.300,00
Ensamblaje(por placa)	50,00	60,00	70,00	80,00	90,00
Inversion Inicial	350,00				
Maquinas y Herramientas	300,00				
Costos Indirectos					
Logistica/Servicios	500,00	550,00	600,00	650,00	700,00
Operaciones de Venta	250,00	300,00	350,00	400,00	450,00
Flujo de Año	400,00	950,00	550,00 -	150,00 -	450,00
Flujo Acumulado	400,00	1.350,00	1.900,00	1.750,00	1.300,00

Figura A.1: Flujo de caja proyectado.

Se calcula el VAN y TIR, estas formulas financieras son herramientas para analizar que tan oportuno es el proyecto, independientemente del área o del tipo de producto en cuestión.

VAN corresponde a Valor Actual Neto y TIR a Tasa Interna de Retorno, ambos conceptos estan estrictamente relacionados con el flujo de caja.

A.3.1. VAN y TIR

Esta herramienta financiera es la diferencia entre el dinero que ingresa y la cantidad que se invierte en el producto para analizar los beneficios.

$$VAN = -I + \prod_{n=1}^N \frac{Q_n}{(1+r)^n} \quad (\text{A.1})$$

Donde I representa la inversión inicial, n representa el año, Q_n el flujo de caja y r la tasa de descuento que en este caso sera del 10 %.

El TIR realiza el mismo calculo llevando el VAN a cero, su resultado se expresa en porcentaje y expresa un rendimiento interno del proyecto.

$$TIR = - \prod_{t=0}^N \frac{Q_n}{(1+r)^n} = 0 \quad (\text{A.2})$$

Inversion Inicial	- 350,00
1	400,00
2	950,00
3	550,00
4	- 150,00
5	- 450,00
Rentabilidad Minima	10,00%

VAN	830,12
TIR	145%

Figura A.2: Cálculo de VAN y TIR con una tasa mínima de 10 %.

El VAN positivo indica que el proyecto es rentable teniendo en cuenta que se utiliza una tasa del 10 %, mientras que LA TIR arroja un valor positivo de 145 %, a mayor tasa interna de retorno es mayor rentabilidad.

Bibliografía

- [1] Zigbee Alliance. Zigbee products. https://zigbeealliance.org/product_type/certified_product/.
- [2] Zigbee Alliance. Zigbee specification. <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf>, 2015.
- [3] ZigBee Alliance. The standard bearer of the open iot. <https://zigbeealliance.org/about/>, 2019.
- [4] ZigBee Alliance. Control your world, your way. <https://zigbeealliance.org>, 2020.
- [5] W. Bauer y G. Westfall. *University Physics Volume 2: Second Edition*. McGraw-Hill Higher Education, 2013. ISBN 9780077654474. URL <https://books.google.com.ar/books?id=CcUjAAAAQBAJ>.
- [6] S.R. Caprile. *Equisbí: Desarrollo de aplicaciones con comunicación remota basadas en módulos ZigBee y 802.15.4*. Sergio R. Caprile. ISBN 9789871301171. URL <https://books.google.com.ar/books?id=xTXv5-AhOhMC>.
- [7] CIA. Computadora industrial abierta argentina. <http://www.proyecto-ciaa.com.ar/>.
- [8] H. Cotuk, K. Bicakci, B. Tavli, y E. Uzun. The impact of transmission power control strategies on lifetime of wireless sensor networks. *IEEE Transactions on Computers*, 63(11):2866–2879, 2014. ISSN 1557-9956. doi:10.1109/TC.2013.151.

-
- [9] DIGI. Api mode in detail. https://www.digi.com/resources/documentation/Digidocs/90001942-13/concepts/c_api_mode_detailed.htm?TocPath=XBee%20API%20mode%7C_____1, .
- [10] DIGI. Digi. <https://www.digi.com/>, .
- [11] DIGI. Zigbee rf modules. Inf. téc., Digi International, 2018.
- [12] Sinem Coleri Ergen. Zigbee/ieee 802.15.4 summary. <http://users.eecs.northwestern.edu/~peters/references/ZigtbeeIEEE802.pdf>, 2014.
- [13] National Instruments. Labview. www.ni.com.
- [14] Texas Instruments. Lm2596 simple switcher[®] power converter 150-khz 3-a step-down voltage regulator. Inf. téc., Texas Instruments, 2016.
- [15] G. V. Mahesh, Dhanavanthan Balachander, y T. Rama Rao. Rf propagation measurements in agricultural fields for wireless sensor communications. *2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, págs. 808–812, 2013.
- [16] Katsuhiro Naito. A survey on the internet-of-things: Standards, challenges and future prospects. *Journal of Information Processing*, 25:23–31, 2017. doi:10.2197/ipsjjip.25.23.
- [17] Marcelo Romeo. Zigbee. Inf. téc., Universidad Nacional de San Martin, 2018.
- [18] Fernandez Veramendi. Arduino based data acquisition system. Inf. téc., Sofia Technical University, 2016.
- [19] Wikipedia. Zonas de fresnel. <https://upload.wikimedia.org/wikipedia/commons/5/5c/FresnelSVG1.svg>, 2020.
- [20] Yang Xiao, Hsiao-Hwa Chen, Ruhai Wang, y Sakshi Sethi. Mac security and security overhead analysis in the ieee 802.15.4 wireless sensor networks. *EU-RASIP Journal on Wireless Communications and Networking*, 2006, 2006. doi:10.1155/WCN/2006/93830.