



Universidad Autónoma de Querétaro
Facultad de ingeniería.

Procesador digital inalámbrico de señales inerciales aplicado a
máquinas CNC y robots.

Tesis.

Que como parte de los requisitos para obtener el grado de Maestro
en Ciencias.

Presenta.

José Luis Contreras Hernández.

Dirigido por.

Dr. René de Jesús Romero Troncoso.

Santiago de Querétaro, Qro, México, Noviembre 2014.



Universidad Autónoma de Querétaro
 Facultad de Ingeniería
 Maestría en Ciencias Mecatrónica

Procesador digital inalámbrico de señales inerciales aplicado a máquinas CNC y robots.

TESIS

Que como parte de los requisitos para obtener el grado de

Maestro en Ciencias

Presenta:

Ing. José Luis Contreras Hernández

Dirigido por:

Dr. René de Jesús Romero Troncoso

SINODALES

Dr. René de Jesús Romero Troncoso
 Presidente

Firma

Dr. Roque Alfredo Osornio Ríos
 Secretario

Firma

Dr. Luis Morales Velázquez
 Vocal

Firma

Dr. Miguel Trejo Hernández
 Suplente

Firma

Dr. J. Jesús de Santiago Pérez
 Suplente

Firma

Dr. Aurelio Domínguez González
 Director de la Facultad de Ingeniería

Dr. Trino Torres Pacheco
 Director de Investigación y Posgrado

Centro Universitario
 Querétaro, Qro.
 Octubre 2014
 México

RESUMEN

En la actualidad, la comunicación inalámbrica está teniendo un gran auge dentro de los sistemas que realizan el procesamiento digital debido a la movilidad que presentan, la cual permite la colocación de procesadores en secciones de las cuales no se han adquirido señales o en las que se complica su instalación. Las señales inerciales permiten la estimación del funcionamiento de máquinas CNC y robots. Este trabajo contribuye en la adquisición, procesamiento y estimación de seis señales inerciales importantes para el análisis de maquinaria. Las señales inerciales obtenidas por el procesador digital inalámbrico son la aceleración, la velocidad, el jerk, la vibración, la inclinación y la rotación. Esta última no estimada en ninguno de los trabajos desarrollados hasta la fecha y las demás solamente parcialmente. Para la estimación de las señales antes mencionadas, se utilizan las señales provenientes de un acelerómetro y de un giroscopio y a las cuales se les aplican procesamientos digitales de filtrado para obtenerlas. Además del procesamiento en Hardware, se realizan un procesamiento en software para estimar las señales inerciales faltantes. Los parámetros a evaluar del procesador digital inalámbrico son la distancia máxima a la que se puede realizar la comunicación inalámbrica, el consumo de potencia para su instalación en zonas inaccesibles mediante la alimentación de una batería, la máxima frecuencia de muestreo y su adquisición de señales inerciales en Máquinas CNC y Robots.

(Palabras clave: Procesador digital, comunicación inalámbrica, señales inerciales, FPGA, Máquina CNC, Robot).

ABSTRACT

Nowadays, wireless communication is important for digital processing systems due to the mobility. This characteristic allows installing the processor in sections which have not been acquired signals or in which complicates installation. The inertial signals allow estimation of CNC and robots wellness. This work contributes to the acquisition, processing and estimation of six inertial signals for machinery analysis. The inertial signals obtained by the wireless digital processor are the acceleration, velocity, jerk, vibration, inclination and rotation. Rotation is not estimated in any of the work investigated. For the estimation of the above signals, accelerometer a gyroscope signals are applied digital processing filtering. In addition to processing in hardware, software processing is performed to estimate the missing inertial signals. The evaluated parameters of wireless digital processor is the maximum distance of wireless communication, power consumption for installation in inaccessible areas feeding by batteries, the maximum sampling rate and acquisition of inertial signals in CNC Machines and Robots.

(Keywords: Digital processor, Wireless communication, inertial signals, FPGA, CNC machine, Robot)

A Martha.

AGRADECIMIENTOS

A Dios, por todo.

A mis padres, mi hermana, familiares y amigos por el apoyo que siempre me han brindado.

Al Dr. René por su confianza, apoyo y sus consejos.

A todos los que aportaron para el desarrollo de este trabajo. En especial al Dr. Roque y al Dr. Luis por sus consejos.

A Martha por su paciencia, apoyo, ánimo y alegría que siempre me da para seguir adelante. Por ser una pieza tan importante.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por otorgarme la beca (487660) para realizar mis estudios de maestría en ciencias en la Universidad Autónoma de Querétaro.

ÍNDICE

	Página
RESUMEN	1
ABSTRACT	ii
AGRADECIMIENTOS.....	iv
ÍNDICE.....	v
INDICE DE FIGURAS.	ix
INDICE DE TABLAS.....	xii
1 Introducción	1
1.1 Antecedentes.....	2
1.2 Objetivos.....	6
1.2.1 Objetivo General.....	6
1.2.2 Objetivos Particulares.....	6
1.3 Hipótesis.	7
1.4 Justificación.	7
1.5 Planteamiento general.....	7
2 Revisión de Literatura.	9
2.1 Estado del arte.....	9
2.2 Señales inerciales.....	10
2.2.1 Velocidad.....	10
2.2.2 Aceleración.....	11
2.2.3 Jerk.	11
2.2.4 Vibración.	12
2.2.5 Inclinación.	12
2.2.6 Rotación.....	12
2.3 Sensores inerciales.....	13
2.4 Máquinas CNC y robots industriales.....	14
2.5 Comunicación serial.	15
2.5.1 Interfaz SPI.....	15
2.5.2 Protocolo RS-232.	17

2.5.3	Protocolo RS-485.	19
2.6	Comunicación inalámbrica.	19
2.6.1	IEEE 802.15.4.	20
2.6.2	ZigBee.	20
2.6.3	KNX.	20
2.6.4	Bluetooth.	21
2.6.5	WiFi.	21
2.7	Transceptor nRF24L01.	21
2.8	Lógica programable.	23
2.9	Técnicas de procesamiento digital de señales y algoritmos.	24
2.9.1	Filtro de diezmo por promedio.	25
2.9.2	Filtro FIR.	25
2.9.3	Filtro complementario.	26
2.9.4	Integral discreta	27
2.9.4.1	Regla de los trapecios	28
2.9.5	Derivada discreta	30
3	Metodología.	31
3.1	Sistema general.	31
3.2	Manejador para el nRF14L01.	35
3.2.1	Módulo de Comandos.	38
3.2.1.1	Memoria ROM.	38
3.2.1.2	Multiplexor de operación.	40
3.2.2	Módulo SPI.	40
3.2.3	FSM.	44
3.2.3.1	FSM principal.	46
3.2.3.2	FSM de configuración.	49
3.2.3.3	FSM de configuración de dirección.	50
3.2.3.4	FSM de transmisión de datos.	51
3.2.3.5	FSM de recepción de datos.	53
3.2.3.6	FSM de borrado del registro STATUS.	55
3.3	Sistema esclavo.	56

3.3.1	Módulo Sns_com.....	57
3.3.2	Máquina de estados NdTx_FSM.....	60
3.4	Sistema maestro.....	61
3.4.1	Máquina de estados del sistema esclavo.....	63
3.4.2	Filtrado.....	64
3.4.2.1	Diezmado por promedio.....	64
3.4.2.2	Filtro digital FIR.....	65
3.4.2.3	Máquina de estados de los filtros.....	66
3.4.3	Guardado de datos.....	67
3.4.3.1	Bloque Memo_Drv.....	71
3.4.4	RS-232.....	73
3.5	Módulo de comunicación RS-232.....	74
3.5.1	Módulo de transmisión.....	74
3.5.2	Módulo de recepción.....	76
3.6	Procesamiento en software.....	78
3.7	Hardware del sistema.....	79
4	PRUEBAS Y RESULTADOS.....	83
4.1	Prueba de distancia de comunicación inalámbrica.....	83
4.2	Prueba de tiempo de procesamiento.....	85
4.3	Prueba de señales inerciales adquiridas en máquina CNC.....	91
4.3.1	Graficas de aceleración.....	92
4.3.2	Graficas de velocidad.....	93
4.3.3	Graficas de vibración.....	94
4.3.4	Graficas de jerk.....	95
4.3.5	Graficas de inclinación.....	96
4.3.6	Grafica de rotación.....	97
4.4	Prueba de señales inerciales adquiridas en Robot.....	97
4.4.1	Graficas de aceleración.....	99
4.4.2	Graficas de velocidad.....	100
4.4.3	Graficas de vibración.....	101
4.4.4	Graficas de jerk.....	101

4.4.5	Graficas de inclinación.....	102
4.4.6	Grafica de rotación.....	103
4.5	Prueba consumo de potencia.....	104
5	Conclusiones.....	106
6	Bibliografía.....	107
	Apéndice A. Productos obtenidos.....	110
	Apéndice B. Código de manejador nRF24L01.....	117
	B.1 nRF_TxRx.vhd.....	117
	B.2 SPI.vhd.....	120
	B.3 FSM.vhd.....	122
	B.4 ROM_Cfg.vhd.....	127

INDICE DE FIGURAS.

Figura	Página
1.1. Planteamiento general del sistema	8
2.1. Esquema de conexión Maestro-Esclavo para protocolo SPI.	16
2.2. Modos de reloj del protocolo SPI. a) Modo A. b) Modo B.	16
2.3. Trama de comunicación RS-232.	18
2.4. Conexiones de terminales del protocolo RS-232.	18
2.5 Conexiones mínimas de terminales del protocolo RS-232	19
2.6. Operación de escritura del SPI.	22
2.7. Operación de lectura del SPI.	23
2.8 Conformación a bloques de un FPGA.	24
2.9. Diagrama a bloques de la estructura del filtro FIR.	26
2.10. Gráfica de trapecio.	28
2.11. Gráfica de trapecio compuesta.	29
3.1. Diagrama a bloques general del sistema digital inalámbrico de señales inerciales.	31
3.2. Diagrama a bloques de sistema esclavo.	32
3.3. Diagrama a bloques de sistema maestro.	33
3.4. Diagrama de flujo de la operación del procesador digital inalámbrico de señales inerciales.	34
3.5. Bloque de manejador nRF24L01.	35
3.6. Estructura del bloque del manejador nRF24L01.	37
3.7. Módulo del protocolo SPI.	41
3.8. Estructura interna del módulo SPI.	41
3.9. Máquina de estados del módulo SPI.	42
3.10. Simulación de las señales del bloque SPI_FlgCn.	43
3.11. Diagrama a bloques del bloque FSM.	44
3.12. Diagrama a bloques del analizador de modo anterior del transceptor.	45
3.13. Diagrama de flujo de la máquina de estados principal.	47
3.14. Diagrama de flujo de la máquina de estados de configuración.	49
3.15. Diagrama de flujo de la máquina de estados de configuración de dirección del transceptor	51
3.16. Diagrama de flujo de la máquina de estados de transmisión.	52
3.17. Diagrama de flujo de la máquina de estados de recepción.	54
3.18. Diagrama de flujo de la máquina de estados del borrado de STATUS.	55
3.19. Diagrama a bloques des sistema esclavo.	56
3.20. Módulo de comunicación con la unidad de medición inercial.	57
3.21. Estructura del módulo Sns_com.	58

3.22. Diagrama de flujo del módulo Sns_com.	59
3.23. Diagrama de flujo del funcionamiento del sistema esclavo.	61
3.24. Diagrama a bloques del sistema inalámbrico maestro.	62
3.25. Diagrama de flujo del funcionamiento del sistema maestro.	63
3.26. Diagrama a bloques de filtros.	64
3.27. Diagrama a bloques del filtro de diezmado por promedio.	65
3.28. Diagrama a bloques del filtro FIR.	66
3.29. Diagrama de flujo de la máquina de estados de los filtros.	67
3.30. Diagrama a bloques del guardado de datos.	68
3.31. Sección de escritura en la máquina de estados de guardado de datos.	69
3.32. Sección de lectura y envío en la máquina de estados de guardado de datos.	70
3.33. Diagrama a bloques del módulo Memo_Drv.	71
3.34. Diagrama de flujo del módulo manejador de memoria externa.	72
3.35. Diagrama a bloques del módulo de comunicación RS-232 para el sistema maestro.	73
3.36. Diagrama de flujo de la máquina de estados del módulo de comunicación RS-232 para el sistema maestro.	74
3.37. Módulo de transmisión del protocolo RS-232.	75
3.38. Diagrama de flujo de módulo de transmisión RS-232.	76
3.39. Módulo de recepción del protocolo RS-232.	76
3.40. Diagrama de flujo de módulo de recepción RS-232.	77
3.41. Diagrama de flujo del procesamiento en software.	78
3.42. Sensor.	80
3.43. Adaptador para Spartan del sensor.	80
3.44. Conexión entre sensor y adaptador.	81
3.45. Hardware del sistema esclavo.	81
3.46. Hardware del sistema maestro.	82
4.1. Gráficas resultantes de la prueba de distancia de comunicación inalámbrica. a) Gráfica de promedio. b) Gráfica de varianza. c) Gráfica de desviación estándar. d) Gráfica de error estándar.	84
4.2. Simulación del procesamiento del sistema maestro.	85
4.3. Tiempo de procesamiento simulado de QCf, QTx, QRx y QDir.	86
4.4. Tiempo de procesamiento simulado de QBs, QDiez y QSave.	87
4.5. Simulación del procesamiento del sistema esclavo.	88
4.6. Tiempo de procesamiento simulado del sistema esclavo.	88
4.7. Fresadora CNC utilizada.	91
4.8. Desplazamiento de la plancha de la fresadora.	92
4.9. Gráficas de las señales de aceleración en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z.	92
4.10. Gráficas de las señales de velocidad en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z.	93

4.11. Gráficas de las señales de vibración en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z.	94
4.12. Gráficas de las señales de jerk en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z .	95
4.13. Gráficas de las señales de inclinación en máquina CNC. a) Eje X. b) Eje Y.	96
4.14. Gráficas de las señales de rotación en máquina CNC.	97
4.15. Robot PUMA probado.	98
4.16. Desplazamiento de las revolutas. a) Desplazamiento de la primera revoluta del robot PUMA. b) Desplazamiento de la tercera revoluta del robot PUMA. c) Desplazamiento de la cuarta revoluta del robot PUMA.	98
4.17. Gráficas de las señales de aceleración del robot. a) Eje X. b) Eje Y. c) Eje Z.	99
4.18. Gráficas de las señales de velocidad del robot. a) Eje X. b) Eje Y. c) Eje Z	100
4.19. Gráficas de las señales de vibración del robot. a) Eje X. b) Eje Y. c) Eje Z	101
4.20. Gráficas de las señales de jerk del robot. a) Eje X. b) Eje Y. c) Eje Z.	102
4.21. Gráficas de las señales de inclinación del robot. a) Eje X. b) Eje Y	103
4.22. Gráficas de las señales de rotación del robot.	104

INDICE DE TABLAS.

Tabla	Página
3.1. Comandos y valores que contiene el módulo ROM.	39
3.2. Comandos y valores para las distintas operaciones del Transceptor.	40
3.3. Salidas correspondientes a cada valor de Mu.	43
4.1. Resultados de prueba de distancia de comunicación.	83
4.2. Resultados de los promedios de corriente, voltaje y potencia.	104

1 INTRODUCCIÓN

En esta tesis se presenta el desarrollo de un procesador digital con comunicación inalámbrica para la realización del monitoreo de señales inerciales aplicado a máquinas CNC y robots. Estas señales son adquiridas mediante un sensor inercial que utiliza el protocolo modbus-RS485 para la comunicación. Los datos de las señales inerciales adquiridos son procesados por un FPGA y posteriormente enviados inalámbricamente desde un transceptor a otro para aplicarles el procesamiento de filtrado en un FPGA colocado a distancia. Todas las señales procesadas se envían hacia el PC para su posterior análisis y estimación de señales inerciales. Todos los módulos necesarios para la realización de este trabajo fueron diseñados por medio del lenguaje descriptivo HDL. Una vez integrados los módulos de cada una de las partes que constituyen el procesador digital, se realizaron las pruebas correspondientes para verificar el correcto funcionamiento.

Este trabajo se desarrollará en cinco capítulos, en los cuales se tienen introducción, revisión de literatura, metodología, resultados y discusión, y para finalizar, conclusiones y prospectivas.

En la introducción se presentarán los antecedentes de este trabajo de una manera resumida, el cual nos ayuda para analizar la aportación de éste. Además de presentar los objetivos e hipótesis que nos ayudaron para desarrollar de una manera más clara el trabajo presentado. Una sección importante en este primer capítulo es la justificación, en la cual se plantea el problema que se solucionó y la manera en que se atacó. Posteriormente se presenta el planteamiento general, en el cual, se desarrollará un diagrama, presentando el problema y la forma de solucionarlo.

En el segundo capítulo se presentará el estado de arte del trabajo realizado, describiendo las diferentes herramientas que lo constituyen como son el FPGA, el transceptor y los sensores. Así como proporcionar datos importantes para la realización de la tesis como son la comunicación inalámbrica y los diferentes protocolos que existen y algunos protocolos alámbricos necesarios en este trabajo.

En el capítulo tres se desarrollará la metodología del trabajo, la constitución a bloques del procesador digital y la interacción que tiene cada uno con los demás bloques.

De igual manera se presenta el desarrollo implementado en software para la estimación de las señales inerciales.

En el cuarto capítulo se presentarán los resultados y la discusión de cada una de las pruebas realizadas por el sistema digital. En el último capítulo, se muestran las conclusiones obtenidas de la realización de este trabajo, así como las perspectivas para la continuación de este trabajo.

1.1 Antecedentes.

La mayoría de las industrias buscan tener una manufactura de alta calidad por medio del control y la reducción de las variaciones en los procesos de producción, obteniendo al mismo tiempo una optimización de costos. Estos requerimientos se pueden mejorar por medio de las máquinas con control numérico computarizado (CNC). Aunado a este tipo de maquinaria, es necesario el monitoreo de las herramientas de manufactura para evitar daños severos, tanto en productos, como en maquinaria. Además del uso de máquinas CNC, en la industria se emplean robots, los cuales han llegado a ser una parte indispensable para tener competitividad en el mercado actual. Los robots, además de realizar tareas peligrosas y repetitivas para el hombre, las realizan de una manera exacta, precisa y rápida por jornadas largas. Para que se pueda lograr esto, es necesario contar siempre con la posición y orientación del robot en la articulación principal o en cada una de las articulaciones con las que cuenta, dependiendo el caso. En ambas aplicaciones es necesario el monitoreo de los parámetros que permiten un correcto control y análisis del desgaste en las máquinas y robots por medio de las señales inerciales de sus diferentes secciones. En algunas de las partes de la maquinaria es complejo el monitoreo debido a sus grados de movilidad o a las secciones giratorias con las que cuenta. Se han desarrollado trabajos acerca del monitoreo de señales en máquinas CNC por medio de comunicación alámbrica como el elaborado por Mota-Valtierra (2010), en el cual presenta un método que utiliza un clasificador inteligente para determinar el desgaste en las herramientas de corte de una fresadora CNC. El monitoreo se efectúa a través del análisis de las fuerzas de corte que están contenidas en las corrientes del motor del husillo. La comunicación entre el controlador de la fresadora y la computadora personal (PC) se produce por medio de un sistema de adquisición de datos que es conectado alámbricamente con un protocolo propio del sistema. Otro trabajo que efectúa la adquisición de datos provenientes de sensores de

una manera alámbrica es el presentado por Albarbar (2008), en éste se presenta el monitoreo de señales inerciales por medio de acelerómetros de sistema microelectromecánico (MEMS) conectados a una máquina *shaker*. La adquisición es desarrollada mediante el uso de una tarjeta de adquisición NI-DAQ conectada a una PC, con el fin de que se analicen las señales obtenidas por la tarjeta. Por otro lado, Trejo-Hernández (2010) utiliza sensores primarios fusionados con un arreglo de compuertas programables en campo (FPGA) para generar un sensor inteligente que monitorea el desgaste del área en herramientas de las máquinas. El despliegado de resultados se produce mediante dos vías. La primera es haciendo uso de una interfaz de salida que se tiene desde el FPGA y la segunda opción es por medio del protocolo USB, éste permite el envío de datos hacia una PC para su posterior análisis. Además de las máquinas CNC, se han desarrollado trabajos de monitoreo de señales inerciales aplicado en robots, como el elaborado por Rodriguez-Donate (2010), en el que se fusionan sensores inteligentes y un FPGA para la extracción de parámetros de dinámica y vibración. En este trabajo se utilizó el robot 6-DOF PUMA, colocando acelerómetros y servomotores con *encoder*, y por medio del procesamiento de señales, obtener los resultados de vibración, inclinación y dinámica del robot. En un trabajo posterior Rodriguez-Donate (2011), desarrolló una red de sensores fusionados para la estimación de la cinemática de ejes múltiples delanteros en robots. En este trabajo son aplicados nuevamente, los acelerómetros y servomotores con *encoder* en el robot 6-DOF PUMA. Las señales de estos sensores son filtradas de ruido y se realizan los cálculos para obtener la posición y orientación de los ejes implementándolos en un FPGA. En ambos trabajos, los resultados obtenidos por el procesador digital de señales son enviados haciendo uso del protocolo USB hacia una PC y ésta, a su vez, es conectada con el control de movimiento del robot usando conexiones alámbricas. Con el mismo modelo de robot, se realizó el trabajo de Lomeli-Hernandez (2012), en donde se utilizan giróscopos digitales (MEMS) junto con tecnología FPGA para el monitoreo de la orientación de robots. La comunicación desarrollada entre el FPGA y los giróscopos digitales en ambos sentidos, se efectúa por medio del protocolo de periférico serial (SPI). Mientras que los parámetros de adquisición y el resultado de la orientación del robot son enviados por medio del protocolo USB hacia una PC. Como se observa en estos trabajos previos, los resultados obtenidos del monitoreo de señales se realizan utilizando interfaces alámbricas. Esto genera

la problemática de la adquisición de las señales inerciales en secciones giratorias de las máquinas CNC y robots. Así como la presencia de interferencias en el envío de datos resultantes en ambientes hostiles para las mediciones inerciales proporcionadas por los sensores. Además, la comunicación alámbrica presenta altos costos debido a las instalaciones de las redes a largas distancias y baja velocidad de transmisión de datos. Es por esto que es necesario el uso de transmisiones inalámbricas como se menciona en el trabajo de Ramamurthy (2007), donde se utilizan diversos tipos de protocolos de comunicación como son el *Bluetooth* y el de Fidelidad Inalámbrica (Wi-Fi), utilizando un transceptor administrado por medio de un microcontrolador para enviar los datos obtenidos por la plataforma de sensores inteligentes. Otro trabajo que desarrolla la administración de un transceptor por medio de un microcontrolador es el efectuado por Shekdar (2012), donde se hace uso de acelerómetros para realizar la adquisición de esta medida inercial mediante el microcontrolador AVR ATmega 16L. El transceptor utilizado para realizar el enlace de comunicación inalámbrica es el modelo CC2500. El sensor y el transceptor se comunican por el protocolo SPI con el microcontrolador. La comunicación inalámbrica se realiza de un modo de múltiples nodos, con lo cual se genera una red con la que se obtiene los datos medidos desde diferentes partes de una misma máquina o de múltiples máquinas que se desean monitorear. En el trabajo realizado por Jong-Woong (2013) se presenta la implementación de un algoritmo de estimación de desplazamiento por medio del uso de un filtro FIR. Este desarrollo es aplicado a estructuras colocadas en una mesa vibratoria. Para realizar el procesamiento de las señales y la comunicación inalámbrica, se utiliza un módulo de comunicación inalámbrico desarrollado por Intel. Este módulo cuenta con sensores de aceleración, tensión, temperatura, humedad y de luz. Su programación se desarrolla mediante un conjunto de herramientas desarrolladas por el grupo de investigación en el que se realiza dicho trabajo, el cual es de libre acceso. La frecuencia de muestreo de la señal de aceleración programada es de 25 Hz. El módulo del nodo que contiene el sensor o nodo esclavo utiliza batería como energía de alimentación. El modo de operación del sistema desarrollado se basa en un nodo maestro y un nodo esclavo. El nodo maestro se encarga de enviar parámetros hacia el esclavo para que éste realice los cálculos necesarios una vez adquiridas las señales de los acelerómetros. Una vez que se tienen los

valores del desplazamiento, son enviados hacia el nodo maestro para que éste los envíe hacia una PC.

El uso de FPGAs proporciona ventajas en el procesamiento de señales gracias a la capacidad computacional paralela con la que cuentan, pudiendo procesar múltiples señales provenientes de los sensores de una manera más rápida en comparación a los microcontroladores. En su trabajo, Kapoor (2007) sustituye los microcontroladores por FPGAs. Además, añade diferentes procesos para las señales como son el filtrado y la aplicación de una ventana. Tanto el código de la comunicación de radio frecuencia (RF), como el proceso de filtrado para el FPGA, son obtenidos mediante un generador de códigos HDL comercial. El transmisor utilizado cuenta con una dimensión amplia y una antena que dificulta su utilización en espacios reducidos. El receptor para la PC es conectado por medio del protocolo RS-232 y de esta manera se pueden analizar los resultados enviados desde el sensor colocado en la maquinaria. Como se ha mencionado con anterioridad, para el monitoreo de señales inerciales es necesario tener una comunicación inalámbrica que maneje diferentes nodos, en los cuales se puedan conectar los sensores. Esto fue presentado en el trabajo de Moreno-Tapia (2010), utilizando una red de sensores inteligentes para realizar las mediciones de diferentes variables en maquinaria CNC como son el monitoreo de vibración basado en un acelerómetro, el monitoreo del jaloneo utilizando un filtro adaptivo asimétrico FIR de orden alto, la medición de posicionamiento mediante detectores y la estimación de daños en herramientas mediante la medición de corrientes del huisillo. Para esto, se maneja un módulo de instrumentación diferente para cada uno de los elementos que monitorean las cuatro variables, haciendo cada uno uso de un convertidor analógico-digital de 12 *bits*. Una vez desarrollado esto, se conecta el módulo antes mencionado, con el módulo de transmisión inalámbrica. En este módulo se integra la unidad de procesamiento en hardware basada en FPGA. Este trabajo cuenta con un coordinador en un espacio equidistante de los nodos, el cual es implementado con una plataforma FPGA Spartan-3 XC3S200 que usa el transceptor MRF 24J40 para la comunicación inalámbrica. Este coordinador envía los datos guardados en una memoria RAM de la tarjeta Spartan a una PC por medio del protocolo RS-232.

Estos trabajos presentan diferencias en la forma de conectar los sensores de las máquinas CNC y los robots para la medición de las variables a analizar. Al ser aplicados a

la industria, el monitoreo inalámbrico mejora la adquisición logrando manejar diversas señales inerciales provenientes de una máquina, o de múltiples, sin tener una gran cantidad de entradas. Así mismo, se obtiene un menor costo de instalación al eliminar los tendidos alámbricos.

1.2 Objetivos.

Los objetivos de este trabajo se dividen en dos secciones. El objetivo general y los objetivos particulares.

1.2.1 Objetivo General.

El objetivo general de este trabajo es desarrollar un módulo de procesamiento digital inalámbrico para la adquisición de señales inerciales provenientes de maquinaria CNC o robots basado en FPGA y utilizando el transceptor inalámbrico nRF24L01.

1.2.2 Objetivos Particulares.

Para lograr el objetivo general, se plantean diferentes objetivos particulares, los cuales se enlistan de acuerdo a su desarrollo en este trabajo:

1. Desarrollar el manejador para el módulo inalámbrico del transceptor nRF24L01 en HDL.
2. Diseñar e implementar filtros pasa-bajas, pasa-altas y de diezmado por promedio en HDL para las señales inerciales.
3. Integrar el módulo modbus-RS485 en HDL para proveer la comunicación con sensores.
4. Realizar pruebas de adquisición y procesamiento de las señales inerciales para analizar el sistema inalámbricamente.
5. Realizar la adquisición de señales inerciales en máquinas CNC y robots para verificar la comunicación inalámbrica con el sensor.
6. Medir el desempeño del bus de comunicación inalámbrica para establecer el ancho de banda del bus mediante un banco de pruebas.
7. Analizar el consumo de potencia del sistema.

1.3 Hipótesis.

El desarrollo de un procesador digital inalámbrico de señales inerciales permite la colocación de sensores en áreas inaccesibles para equipos alámbricos en el monitoreo de la dinámica de máquinas CNC y robots.

1.4 Justificación.

Los sistemas de adquisición de señales inerciales desarrollados, no realizan la adquisición de las señales como son velocidad, aceleración, *jerk*, rotación, inclinación y vibración. Estos sistemas presentan la adquisición de una o hasta máximo cuatro señales. Al realizar el monitoreo de las señales inerciales que se presentan en alguna sección de las máquinas CNC o de los robots, existe la falta de señales para realizar un análisis completo, entre los que se encuentra el análisis de la señal de rotación. La adquisición y análisis de las señales inerciales en secciones giratorias de las máquinas CNC y robots se ven limitadas debido a las conexiones alámbricas que se presentan entre los sensores y los sistemas de procesamiento, por lo que una conexión inalámbrica permitiría al sistema obtener señales inerciales en este tipo de secciones con un consumo bajo de corriente. El tiempo de procesamiento es importante para obtener una frecuencia de muestreo mayor, de tal manera que un procesamiento paralelo aumenta la frecuencia en comparación a los microcontroladores o microprocesadores. El manejo del protocolo RS-485 para la adquisición de datos, permite la compatibilidad con otros sensores o sistemas desarrollados en el grupo de investigación.

1.5 Planteamiento general.

Para el desarrollo de este trabajo es necesario identificar las partes de las que se conforma. En la Figura 1.1 **¡Error! No se encuentra el origen de la referencia.** se presenta el planteamiento general del sistema. Este sistema se conforma por dos secciones, nombradas como sistema esclavo y sistema maestro.

El diagrama a bloques del sistema esclavo se muestra en la Figura 1.1 **¡Error! No se encuentra el origen de la referencia.** Inicialmente se tiene la máquina CNC o el robot al cual se acopla el sensor. Este sensor permite la obtención de las señales de aceleración y velocidad angular provenientes del acelerómetro y giroscopio, respectivamente. El módulo

del sistema de comunicación inalámbrica realiza la interfaz para la adquisición de las señales inerciales del sensor. Además, realiza la comunicación y la configuración necesaria usando el protocolo SPI para transmitir las señales inerciales inalámbricamente con el transceptor nRF24L01. Este módulo de comunicación cuenta con los protocolos RS-232 y SPI. Estos protocolos son desarrollados en Lenguaje de descripción de hardware.

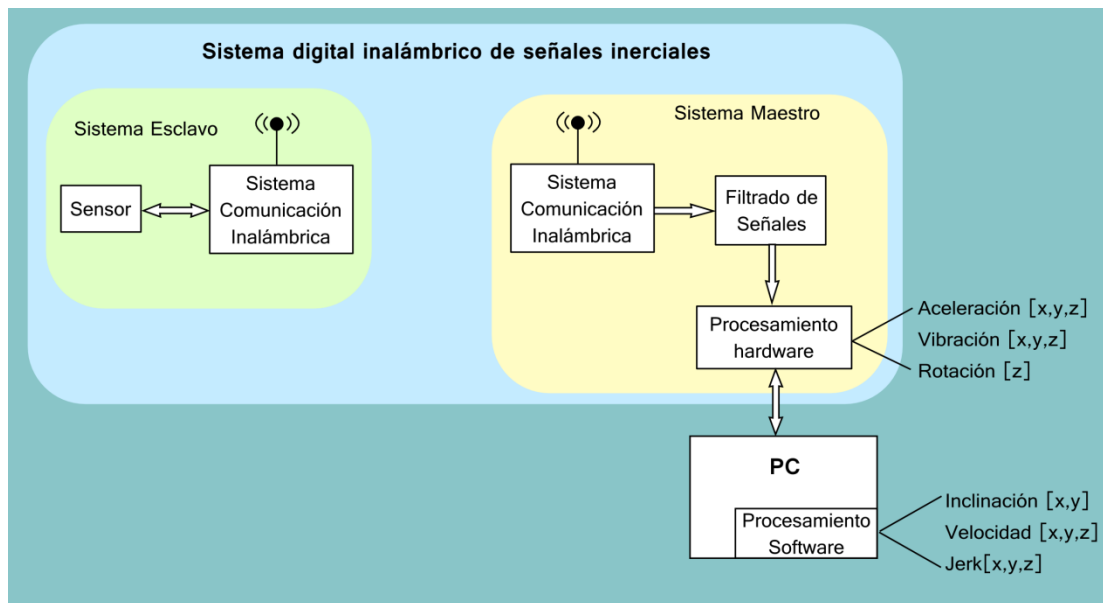


Figura 1.1. Planteamiento general del sistema

El diagrama a bloques del sistema maestro se muestra en misma Figura 1.1. Al igual que con el sistema maestro, la comunicación inalámbrica se realiza mediante el transceptor nRF24L01. El módulo del sistema de comunicación inalámbrica se encarga de configurar al transceptor y de procesar los datos de las señales inerciales recibidas mediante el protocolo SPI. A las señales inerciales recibidas se les aplica un filtro para la eliminación de señales de ruido de cuantificación que pudieran afectarlas. Enseguida se realiza un procesamiento en hardware para la obtención de aceleración y vibración en los ejes X, Y y Z y de la velocidad angular en el eje z, la cual corresponde a la rotación. Una vez realizado esto, el procesador del sistema maestro guarda los datos recibidos y procesados en una memoria RAM externa. Después los envía a la PC para estimar la inclinación, la velocidad y el jerk mediante los datos previamente obtenidos. La comunicación entre el sistema digital de

señales inerciales y la PC se efectúa con el protocolo RS-232. Obtenidas las seis señales inerciales, se procede a graficarlas en la PC mediante el software MATLAB.

2 REVISIÓN DE LITERATURA.

En este capítulo se presentará la fundamentación del sistema realizado, presentando la historia de la comunicación inalámbrica de una manera breve y presentando las razones por las cuales la industria ha optado por realizar el monitoreo con este tipo de comunicación. Además se describirán las diferentes herramientas que constituyen el proyecto de esta tesis como son el FPGA, el transceptor y los sensores. Además de proporcionar datos importantes como la comunicación inalámbrica, los diferentes protocolos que existen, algunos protocolos necesarios en este trabajo, los algoritmos implementados para realizar el filtrado de las señales y para la estimación de otras señales inerciales.

2.1 Estado del arte.

Heinrich Rudolph Hertz demostró la existencia de las ondas electromagnéticas en 1887, éstas ya había sido predichas por James Clerk Maxwell y Michael Faraday. Desde entonces se comenzó a desarrollar la comunicación inalámbrica muy lentamente, con la aparición del telégrafo inalámbrico en 1895, la radio en 1906, la televisión 1948 y la comunicación vía satélite en 1963. En los últimos 14 años han aparecido un gran número de estándares de comunicación inalámbrica a corta distancia, con lo que ha habido una revolución en el desarrollo y aceptación de productos móviles. Esto ha ocurrido por el deseo de obtener productos portables y su globalización. No todos estos estándares han sobrevivido, pero muchos se han vuelto muy famosos, al grado de tener billones de dispositivos que los utilizan. Un ejemplo es el estándar *Bluetooth* y *Wi-Fi*.

La industria ha comenzado a explorar las ventajas de la comunicación inalámbrica y de los tipos de estándares adecuados para sus necesidades. Existen infinitas aplicaciones para la comunicación inalámbrica en la automatización industrial donde el costo de sensores alámbricos es significativo. La tecnología inalámbrica permite la instalación de un gran número de sensores para el monitoreo de maquinaria en lugares donde la instalación

alámbrica es costosa o poco práctica. El uso de máquinas CNC y robots en la industria ha generado la necesidad de monitorear las acciones que afectan su funcionamiento o que generan un gran desgaste en estos. Los equipos utilizados son de un alto costo y con los cuales se llega a tener una alta producción, por lo que la existencia de un daño afecta seriamente a la economía de la empresa. La instalación y monitoreo permite recibir una mejor retroalimentación del estado de las máquinas para mejorar su mantenimiento, reducir los tiempos de reparación y los costos (Hunn, 2010).

2.2 Señales inerciales.

La fuerza inercial es una fuerza imaginaria, la cual cuando actúa sobre un cuerpo rígido, trae una posición de equilibrio. Es numéricamente igual a la fuerza de aceleración en magnitud, pero en dirección opuesta (Khurmi, 2006). En un movimiento rotacional ocurre algo similar, debido a que cuando se desea variar la velocidad de rotación de un cuerpo, llamada velocidad angular, se ejerce una fuerza proporcional a una constante, la cual expresa la resistencia de ese objeto a cambiar de su estado de rotación (Fabio, 2010).

Las señales obtenidas debido a la acción de la fuerza inercial son llamadas señales inerciales. Estas señales se ven reflejadas en la aceleración del objeto y en la velocidad angular. En otras palabras, son toda señal obtenida debido a la acción de las leyes de Newton en un sistema (Martins, 2007).

Las señales inerciales que se manifiestan debido a la fuerza inercial son la velocidad, aceleración, jerk o sacudida, vibración, inclinación y rotación.

2.2.1 Velocidad.

Es una magnitud física vectorial que nos expresa la rapidez con la cual un móvil cambia o tiene a cambiar de posición en un intervalo de tiempo. Esto se representa en la ecuación (2.1).

$$\vec{v} = \frac{x_2 - x_1}{t_2 - t_1} = \frac{\Delta x}{\Delta t} \quad (2.1)$$

En dónde, Δx es el desplazamiento del cuerpo y Δt es el tiempo que ha pasado durante el periodo de observación elegido (Giancoli, 2006).

2.2.2 Aceleración.

La aceleración es la razón del cambio de velocidad de un objeto entre el tiempo que toma en realizar este cambio. En la ecuación (2.2) se presenta la obtención de la aceleración.

$$\vec{a} = \frac{v_2 - v_1}{t_2 - t_1} = \frac{\Delta v}{\Delta t} \quad (2.2)$$

De otra manera:

$$a = \frac{dv}{dt} \quad (2.3)$$

De la ecuación (2.3) se despeja la velocidad obteniendo la ecuación (2.4)

$$dv = a dt \quad (2.4)$$

Integrando en ambos lados, se obtiene la ecuación (2.5)

$$v = \int a dt \quad (2.5)$$

Por lo que a partir de la integración de la aceleración se puede obtener la medida de la velocidad (Giancoli, 2006).

2.2.3 Jerk.

El jerk, sobreaceleración o sacudida es la tasa de cambio de la aceleración, o en otras palabras, la derivada de la aceleración con respecto al tiempo como se puede ver en la ecuación (2.6).

$$j = \frac{da}{dt} \quad (2.6)$$

El jerk también es la segunda derivada de la velocidad como se muestra en la ecuación (2.7).

$$j = \frac{d^2v}{d^2t} \quad (2.7)$$

El jerk es una magnitud vectorial debido a que la aceleración también lo es (Rodríguez-Donate, 2010) .

2.2.4 Vibración.

Cualquier movimiento que se repita por sí mismo después de un intervalo de tiempo es llamado vibración. La vibración de un sistema implica la transferencia de energía potencial a energía cinética y viceversa alternadamente. La vibración, en función de las aceleraciones, se considera como una aceleración seguida de una desaceleración y una nueva aceleración en el sentido inverso de traslación que ocurre rápidamente, todo esto de manera periódica y a frecuencias altas. Debido a esto, para obtener las vibraciones de un sistema, se hace uso de un acelerómetro con un filtro que permita el paso de las frecuencias altas y suprima las frecuencias bajas. (Rodríguez-Donate, 2010)

2.2.5 Inclinación.

La inclinación es la desviación de un cuerpo de su posición vertical u horizontal. En este caso se tiene como posición de referencia la gravedad, la cual es una aceleración. Mediante ésta, se puede obtener el ángulo en el que se encuentra posicionado el cuerpo. Por lo que se puede obtener indirectamente al utilizar la medida del acelerómetro, como se puede observar en la ecuación (2.8).

$$\theta = \text{atan2} \frac{acc_{md}}{g} \quad (2.8)$$

Como se puede observar en la ecuación (2.8), para obtener el ángulo del eje deseado se obtiene su lectura del acelerómetro acc_{md} y se divide entre la gravedad g . A este resultado se le aplica el arcotangente2 para obtener el ángulo del eje. (Gad-el-Hak, 2001).

2.2.6 Rotación.

La rotación es un movimiento de cambio de orientación en un cuerpo que se produce de tal forma que cualquier punto del mismo permanece a una distancia constante del eje de rotación. Al igual que la señal inercial de inclinación, la señal inercial de rotación

es obtenida indirectamente. En este caso se utiliza el giroscopio para obtener la lectura de la velocidad del eje y mediante ésta obtener el ángulo del cuerpo de acuerdo a la ecuación (2.9). En donde ω corresponde a la velocidad obtenida mediante el giroscopio y τ es el tiempo de muestreo. Al realizar la multiplicación de ambas variables, se obtiene el ángulo de rotación desplazado en el muestreo presente. Para obtener el ángulo con respecto al eje de referencia, al ángulo calculado se le suma el ángulo anterior φ_p (Groves, 2013).

$$\varphi = (\omega * \tau) + \varphi_p \quad (2.9)$$

2.3 Sensores inerciales.

Un sistema inercial es aquel que cumple con las leyes de Newton. Para poder cuantificar las diferentes variables inerciales en un cuerpo, es necesario hacer uso de sensores. Los adecuados para realizar esta tarea son los acelerómetros y los giróscopos. Utilizando estos sensores se pueden determinar los movimientos físicos de aceleración, vibración, golpe, inclinación y rotación. La aceleración mide la variación de velocidad en una unidad de tiempo. Cuando la aceleración adopta un valor negativo se le conoce como desaceleración. La vibración, en función de las aceleraciones, se considera como una aceleración seguida de una desaceleración y una nueva aceleración en el sentido inverso de traslación que ocurre rápidamente, todo esto de manera periódica. De igual manera, el choque es una desaceleración que se produce de forma instantánea. Cuando un objeto se mueve de tal manera que se modifica su inclinación, existen cambios donde la acción gravitatoria está involucrada. Un acelerómetro detecta la inclinación al medir el efecto que la fuerza de la gravedad ejerce sobre los ejes del acelerómetro que estén expuestos a esta acción en función de su posición en el espacio. La rotación requiere el análisis de un movimiento angular. Este modo difiere de los anteriores, debido a que la rotación puede tener lugar sin observarse cambios en la aceleración. Por lo tanto, para detectar los movimientos de rotación de un cuerpo se utilizan los giróscopos (Bose, A. 2009).

Los sensores inerciales típicos están basados en el movimiento de una masa de prueba sísmica causada por una inercia. Estos sensores utilizan diferentes principios de medición como son capacitivo, piezoresistivo y piezoeléctrico (Rong, Z. 2010).

Los recientes avances de la tecnología han hecho posible la fabricación de pequeños acelerómetros de tecnología MEMS en los rangos de detección de bajas y altas unidades de gravedad con anchos de banda mucho más amplios que antes, aumentando de este modo el campo de las aplicaciones potenciales (Albarbar, A. 2008). Es importante para las máquinas CNC y robots el uso de este tipo de sensores para su monitoreo, obteniendo de esta manera, una señal de datos que describa el comportamiento de las herramientas o de la maquinaria utilizada.

2.4 Máquinas CNC y robots industriales.

Los procesos de manufactura han sido de gran relevancia en el desarrollo de la economía de varios países y la constante mejora de la productividad con alta calidad a bajo costo es un punto importante. Esto puede ser obtenido por medio del uso de máquinas CNC y robots (Trejo-Hernandez, 2010).

El desarrollo del CNC se originó a principios de los años cincuenta en el Instituto de Tecnología de Massachusetts (MIT), siendo utilizada en una fresadora. En la actualidad es aplicada a diferentes máquinas de la industria como son: tornos, taladradoras, centros de maquinado, rectificadoras, entre otros. El principio de operación común de todas las aplicaciones del control numérico es el control de la posición relativa de una herramienta o elemento de procesado con respecto al objeto a procesar. En una máquina CNC, a diferencia de una máquina convencional o manual, una computadora controla la posición y velocidad de los motores que accionan los ejes de la máquina. Gracias a esto, puede hacer movimientos que no se pueden lograr manualmente como círculos, líneas diagonales y figuras complejas tridimensionales. Además de brindarle una mayor seguridad al operador y reducción en los controles de calidad y desechos. Las máquinas CNC son capaces de mover la herramienta al mismo tiempo en los tres ejes para ejecutar trayectorias tridimensionales como las que se requieren para el maquinado de complejos moldes y troqueles. El término control numérico se debe a que las órdenes dadas a la máquina son indicadas mediante códigos numéricos, los cuales siguen una secuencia lógica. Estos códigos tuvieron que normalizarse para lograr una mayor versatilidad en los códigos de las máquinas del mismo tipo (Pabla, 1994).

Un robot es un sistema mecánico diseñado para emular diversas tareas humanas, controlado por un sistema electrónico donde los sensores son una parte esencial de todo el sistema, ya que éstos son los encargados de brindar la información necesaria para llevar a cabo las tareas requeridas. La industria es el lugar de mayor aplicación para los robots, realizando tareas repetitivas o peligrosas para el hombre. El desempeño de las tareas depende de la precisión con la que se realicen las mediciones de los sensores. Los más utilizados en los robots industriales son los *encoders*, los cuales son dispositivos mecánicos montados a la flecha de cada servomotor del robot con el que se mide su desplazamiento angular mediante un sistema de cuentas. Otro tipo de sensores importantes son los acelerómetros y giróscopos. Con estos, se puede medir la aceleración o velocidad del movimiento de los eslabones del robot y mediante un algoritmo se pueden determinar los desplazamientos del mismo (Lomelí-Hernández, 2012).

2.5 Comunicación serial.

La comunicación serial es el proceso de envío de datos de un bit por vez, secuencialmente, sobre un canal de comunicación o un bus. Los protocolos RS-232, RS-485 y SPI son utilizados en esta tesis para la realizar la comunicación alámbrica entre dispositivos o equipos.

2.5.1 Interfaz SPI.

El protocolo SPI es un estándar de enlace serial síncrono introducido por Motorola en 1979, posteriormente adoptado por otras industrias. Consta de un bus de alta velocidad con cerca de 9.5 Mbps, con comunicación *Full* dúplex. La interfaz SPI es usado por muchos microcontroladores y microprocesadores que permiten la comunicación con otros dispositivos. Esta comunicación es realizada por medio de una interfaz Maestro-Esclavo y puede soportar hasta una velocidad de transferencia de 10 Mbps. Por medio de este protocolo, se pueden tener uno o varios maestros a la vez. Los elementos que se pueden conectar por medio de este protocolo son convertidores ADC y DAC, módulos de memoria como EEPROM y FLASH, sensores de temperatura y presión, controladores de LCD, controladores de USB u otros dispositivos como los mezcladores de señales. Como se puede observar en la Figura 2.1, el protocolo SPI cuenta con cuatro líneas de control de datos. La señal de datos Salida-Maestro, Entrada-Esclavo (MOSI); es generada por el

maestro y recibida por el esclavo. La señal de datos Entrada-Maestro, Salida-Eslavo (MISO); es generada por el esclavo. La señal serial del Reloj (SCLK o SCK) es generada por el Maestro para sincronizar la transferencia de datos al Esclavo. La señal Selector de Esclavo (SS) es generada por el Maestro para elegir al Esclavo con el cual se comunicará (Shekdar, 2012).

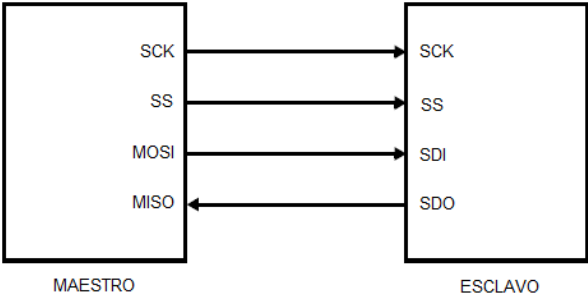


Figura 2.1. Esquema de conexión Maestro-Esclavo para protocolo SPI.

Debido a que el reloj con el que se efectúa la comunicación es controlado por el maestro, se tienen cuatro modos de reloj del protocolo SPI para la transferencia de datos por medio de la señal *MOSI* o *MISO*, conocidos como modo A, B, C y D. En la Figura 2.2 se muestran los diferentes modos del protocolo SPI.

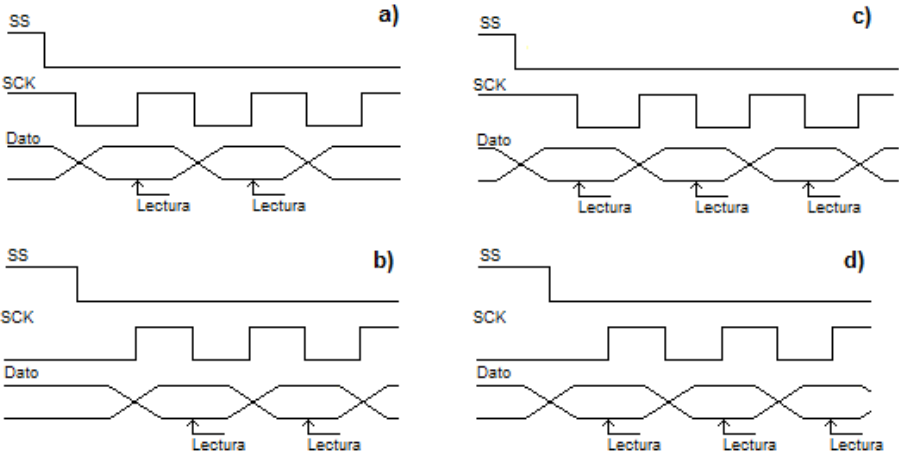


Figura 2.2. Modos de reloj del protocolo SPI. a) Modo A. b) Modo B. c) Modo C. d) Modo D.

En la Figura 2.2a se presenta el modo A. En este modo, los nuevos datos son colocados sobre la línea cuando el flanco de reloj SCK es descendente. El flanco de reloj se

mantiene en alto hasta un instante después de que la señal de selección de esclavo SS es colocada en bajo. La lectura del dato se realiza cuando el flanco de reloj es ascendente. En la Figura 2.2b se muestra el modo B del protocolo SPI, los datos son colocados sobre la línea cuando el flanco de reloj es ascendente. En este caso, el reloj se mantiene en bajo hasta un instante después de que la señal SS cambia de alto a bajo. La lectura del dato se realiza cuando el flanco de reloj es descendente. El modo C del protocolo SPI se realiza cuando los datos son colocados en la línea en el momento en el que la señal SS es activada en nivel bajo. El reloj se mantiene en alto hasta un instante después de que la señal es activada. El dato es leído en el primer flanco de bajada del reloj. Este modo es mostrado en la Figura 2.2c. Finalmente, al igual que en el modo C, el modo D se presenta cuando la señal SS se coloca en nivel bajo y al mismo instante, los datos también son colocados en la línea de salida. La diferencia radica en la señal de reloj, debido a que se mantiene en bajo hasta un instante después de que la señal SS es activada. La lectura de datos se efectúa en cada flanco de subida de la reloj como observa en la Figura 2.2d.

2.5.2 Protocolo RS-232.

El puerto serial envía información por medio de un tren de pulsos utilizando el protocolo RS-232. Este protocolo fue diseñado en los años sesenta para comunicar un equipo terminal de datos (DTE) y un equipo de comunicación de datos (DCE).

Este protocolo utiliza niveles lógicos desde +3 hasta +25 Volts para el '0' y desde -3 hasta -25 Volts para el '1'.

La velocidad de transmisión es la cantidad de datos transmitidos en unidad de tiempo expresada en bits por segundo (bps). En las transmisiones serie se utiliza el Baudio (Baud rate) como la medida de esta velocidad. Baudio se define como el número de veces que cambia la señal portadora en un segundo. Las velocidades que pueden emplearse son de 300, 600, 1200, 2400, 4800 y 9600 Bauds. La velocidad que puede emplearse depende de la calidad del medio de transmisión.

Para realizar la configuración principal del protocolo es básicamente la selección de la velocidad en baudios, la verificación de datos o paridad, los bits de parada luego de cada dato, y la cantidad de bits por dato. En la Figura 2.3 se muestra la trama de comunicación del protocolo RS-232.

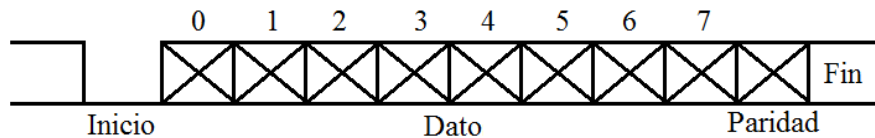


Figura 2.3. Trama de comunicación RS-232.

El protocolo RS-232 inicia por medio de un cero lógico. Este bit es necesario para indicar que se realizará un envío de datos. En seguida se envían los bits que conforman el dato, este puede variar desde cinco hasta ocho bits. Después del dato, se envía el o los bits de paridad. Este bit sirve para detectar errores en la comunicación. El transmisor calcula este bit a partir de la información enviada y es concatenado en la trama de comunicación. El receptor realiza el mismo cálculo de este bit y lo compara con el bit concatenado por el transmisor. La paridad puede escribirse con uno o dos bits. Al final se envía el bit de fin o paro, el cuál puede ser tardar entre uno o dos ciclos para indicar que el envío de la trama ha terminado. Para realizar la comunicación *full duplex*, se requieren de las conexiones mostradas en la Figura 2.4. Se requiere de conexiones para el envío y recepción de datos, tales como TxD y RxD respectivamente. Para realizar la petición de un envío de datos se cuenta con la conexión RTS. Para indicar que el receptor se encuentra listo para transmitir se utiliza la conexión CTS. También existen conexiones para indicar que tanto el receptor como el transmisor se encuentran listos para la comunicación de los datos, al es el caso de las señales DSR y DTR, respectivamente. Por último se presenta la conexión de tierra entre ambos para lograr un correcto acoplamiento de voltaje para la comunicación.

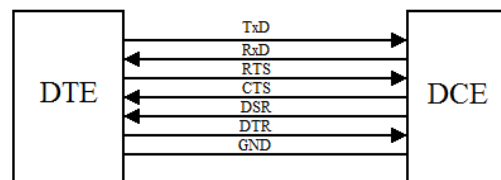


Figura 2.4. Conexiones de terminales del protocolo RS-232.

Anteriormente se presentaron los requerimientos para realizar la comunicación por medio del protocolo RS-232. Pero se pueden llegar a omitir varias de las conexiones dejando solamente TxD, RxD y GND, como se muestra en la Figura 2.5.

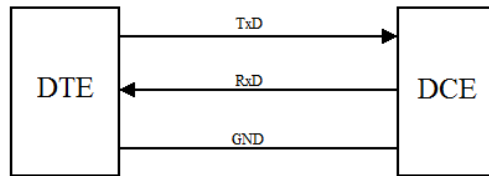


Figura 2.5 Conexiones mínimas de terminales del protocolo RS-232

2.5.3 Protocolo RS-485.

El protocolo RS-485 es una comunicación serial para computadoras y dispositivos. Este protocolo es el estándar de comunicación serial más versátil realizado por la Electronics Industry Association (EIA), por esto, es muy usado en aplicaciones de adquisición de datos y control donde múltiples nodos se comunican con otros. El protocolo RS-485 tiene una doble línea de comunicación, por lo que se eliminan los ruidos provocados por campos magnéticos debido a que el dato recibido no se encuentra diferenciado a tierra, se encuentra diferenciado entre ambas líneas. Este ruido es reducido o eliminado inclusive con las líneas enredadas entre sí. Los voltajes utilizados por el protocolo RS-485 es de máximo 6 Volts lo que ayuda también a la eliminación del ruido. Este protocolo permite realizar conexiones de hasta 1200 m y con una frecuencia de transmisión de 35 Mbps. Otra ventaja de este protocolo es el poder conectar varios transmisores y receptores en la misma red, llegando hasta 32 nodos con un simple par de cables, cosa que el protocolo RS-232 solamente logra punto a punto. Con una alta impedancia receptora, los enlaces con RS-485 pueden llegar a tener a lo máximo hasta 256 nodos. La velocidad de transmisión es de 10 Mbps. (Ingeniería en Microcontroladores, 2013).

2.6 Comunicación inalámbrica.

Heinrich Rudolph Hertz demostró la existencia de las ondas electromagnéticas y que por medio de éstas, se podía enviar información. Ésta es la base de la comunicación inalámbrica (*Wireless*). Para realizar dicha comunicación es necesario el uso de protocolos. Existe una gran variedad de protocolos inalámbricos para realizar la comunicación entre dos equipos o elementos. Existen estándares comerciales, los propios de una cierta marca y los que se realizan de acuerdo a la aplicación diseñada.

2.6.1 IEEE 802.15.4.

El estándar IEEE 802.15.4 especifica la capa física y el medio de acceso al control de la capa, las propiedades más importantes para las redes inalámbricas como la frecuencia de transferencia, el rango de frecuencias y la modulación. Muchos protocolos de redes de sensores inalámbricos populares están basados en este estándar de la IEEE. Hay muy pocas versiones de esta especificación disponibles como son las 802.15.4, 802.15.4-2006 y 802.15.4a-2007 donde el estándar IEEE802.15.4 es más conocido como 802.15.4-2003. La principal diferencia del estándar IEEE 802.15.4 y IEEE 802.15.4-2006 es la modulación ASK y O-QPSK que permite una frecuencia de transferencia más alta en bajas frecuencias (IEEE 802.15.4-2003, 2003). Las frecuencias específicas, modulación y la frecuencia de transferencia varía dependiendo de la región. El estándar IEEE 802.15.4a adiciona bandas de frecuencia y una modulación para las frecuencias de 2.45 GHz (IEEE 802.15.4a-2007, 2007).

2.6.2 ZigBee.

El estándar ZigBee es una red de sensores inalámbricos de bajo costo y bajo consumo, usado para el ahorro de energía y para tener una gran eficiencia. Su ahorro en el consumo de energía es debido a que los nodos pueden entrar en un modo de apagado mientras no es utilizado. Este estándar es usado en hogares y para la automatización de edificios e industrias. Este protocolo utiliza el estándar 802.15.4 y opera en el rango de frecuencias de 2.4 GHz con una transferencia de 250 kbps. El máximo número de nodos que puede alcanzar una red con este protocolo es de 1024 con un alcance de 200m. Teniendo algunos modelos que alcanzan los 1000m, consumiendo la potencia máxima posible. El protocolo puede usar una encriptación de hasta 128 bits (Zigbee, 2013).

2.6.3 KNX.

Este es un sistema de comunicación para hogares y automatización de edificios, que se basa en el estándar internacional ISO/IEC14543-3, los estándares europeos CENELEC, EN50090 y CEN EN 13321-1 y 13321-2, el estándar chino GB/Z 20965 and ANSI/ASHRAE 135. Este estándar opera a frecuencias de 868.3 MHz con modulación FSK y una velocidad de transferencia de 16.384 kbps. Los nodos de KNX, con conectores adicionales, pueden realizar la comunicación entre nodos que se encuentran bloqueados por algún objeto. El estándar KNX no es gratuito (KNX, 2013).

2.6.4 Bluetooth.

Este protocolo trabaja en la banda de los 2.4 GHz, además tiene una velocidad de transferencia de 3Mbps y una longitud de alcance de 100m. El protocolo Bluetooth utiliza perfiles para la comunicación. Un protocolo emergente de éste es el Wibree, que no es más que el protocolo Bluetooth con un consumo ultra bajo (Bluetooth SIG. 2009).

2.6.5 WiFi.

El protocolo WiFi es una red establecida para una red inalámbrica de área personal (WPAN). Está basado en el estándar IEEE 802.11 y opera en la banda de frecuencia de 2.45 GHz. La velocidad de transferencia va desde 54 Mbps para el estándar 802.11g hasta los 150 Mbps para el estándar 802.11n. El uso del protocolo adecuado depende de la aplicación a realizar y de las diferentes características con las que cuente el sistema (IEEE 802.11, 1997.).

2.7 Transceptor nRF24L01.

Un transceptor es un dispositivo que cuenta con un transmisor y un receptor que comparten parte de la circuitería. Debido a esto, un transceptor es manejado como una comunicación semidúplex, es decir, que pueden enviarse señales en ambos sentidos, pero no simultáneamente. El transceptor utilizado en este trabajo es el nRF24L01, este es un chip individual con un protocolo de banda base embebido (ShockBurst™). Este protocolo de comunicación fue diseñado para aplicaciones de comunicación inalámbrica de ultra bajo consumo de energía y bajo consumo de corriente. El transceptor nRF24L01 está diseñado para operar en la banda ancha de frecuencias de 2.400 - 2.4835 GHz, con 126 canales de radio frecuencia, además de permitir la comunicación entre diferentes puntos en un solo canal de frecuencia por medio de una recepción de datos secuencial. El transceptor es configurado y operado a través de la interfaz serial periférica (SPI). Utilizando el protocolo SPI, se pueden manejar las configuraciones de los registros del transceptor como son la frecuencia del canal de comunicación, la potencia de salida, la frecuencia del envío de datos, la dirección del transceptor, el número de bits del dato enviado o el modo en que opera el transceptor, ya sea como transmisor o como receptor.

Para configurar los registros del transceptor, se cuenta con dos tipos de operaciones, la escritura y la lectura.

En la Figura 2.6 se muestra la operación de escritura para los registros del transceptor, en la cual se presentan las cuatro señales del protocolo SPI. Como se puede observar, la señal *CSN* de activación del reloj controla el envío y recepción de los datos. Mientras se mantiene en alto, las demás señales se encuentran sin acción hasta que la señal cambia a estado bajo. *SCK* es el reloj de comunicación entre el maestro y esclavo. La señal *MOSI* se compone de dos secciones. La primera sección cuenta con ocho bits de comando, en la cual se ingresa la acción que se desea realizar en el transceptor, ya sea escritura de un registro o el envío inalámbrico de datos. En la segunda sección, la cual puede variar desde uno hasta treinta y dos *bytes*, se coloca el valor que se escribirá en el registro indicado por la sección del comando o el dato completo que se desea enviar inalámbricamente. En esta sección se inicia por el *byte* menos significativo. En la señal *MISO*, de acuerdo a su configuración, se recibe un señal de unos o ceros lógicos.

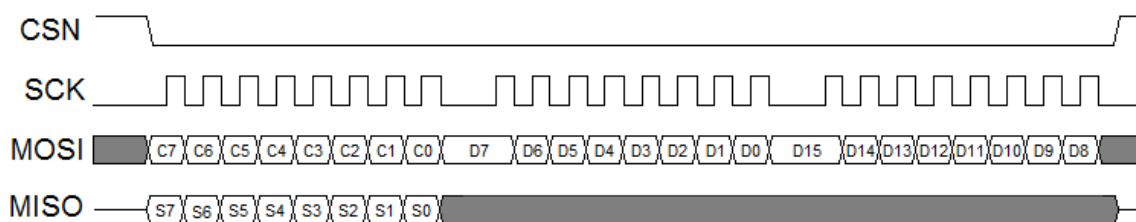


Figura 2.6. Operación de escritura del SPI.

En la Figura 2.7 se muestra la operación de lectura de datos provenientes del transceptor y sus registros. Al igual que la operación de escritura, se cuenta con la señal de activación *CSN* y la señal de reloj *SCK*. Para el caso de lectura, en la señal *MOSI* se envía el comando que indica la lectura de algún registro o la lectura del dato recibido inalámbricamente. En la señal *MISO* la primer trama recibida corresponde a una señal de unos o ceros, dependiendo de la configuración en la que se encuentre el transceptor. Enseguida se reciben las tramas del dato contenido en el registro que se desea leer o el dato recibido inalámbricamente, comenzando por el *byte* menos significativo.

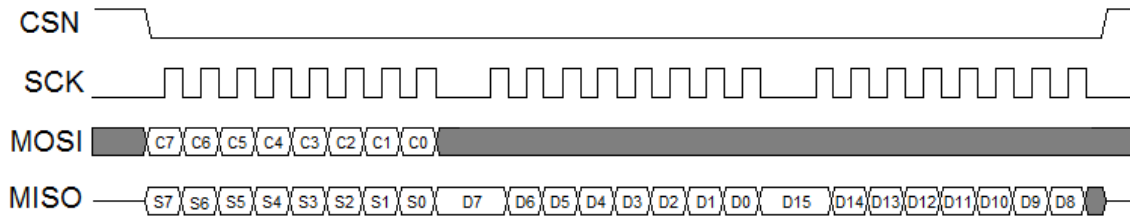


Figura 2.7. Operación de lectura del SPI.

Una vez configurado el transceptor, la comunicación inalámbrica se realiza por medio de un protocolo propio de la empresa *Nordic Semiconductor* conocido como ShockBurst™. Este protocolo está basado en un paquete de comunicación y soporta varios modos de la operación manual para protocolos de operación autónomos avanzados. El ShockBurst™ reduce el costo del sistema manejando un enlace de alta velocidad. El transceptor nRF24L01 hace uso de la modulación GFSK (Nordic Semiconductor, 2007).

2.8 Lógica programable.

El Arreglo de Compuertas Programables en Campo (FPGA) tiene la característica de tener altas densidades de compuertas, además de contar con un alto rendimiento y un número grande de entradas y salida que se pueden definir por el usuario. Esta familia fue creada para lograr una optimización de recursos en los bloques digitales de los dispositivos, una alternativa en la cual se pudiera hacer uso de una mayor cantidad de componentes y con una eficiencia superior (Romero Troncoso, 2007).

El FPGA se encuentra compreso bajo una arquitectura estilo Manhattan, en las cuales cada celda contiene estructuras lógicas simples y estas celdas se encuentran conectadas entre sí. Las celdas básicas usadas en una FPGA son las celdas lógicas, las celdas de entrada/salida y las celdas de distribución. Como se puede observar en la Figura 2.8, la celda lógica es la que ocupa una mayor cantidad de espacio y se compone por pequeños bloques lógicos de compuertas programables (CLB). Este bloque contiene lógica combinacional programable y registros de almacenamiento. Además de tener la capacidad de implementar cualquier función booleana de sus entradas. Cabe mencionar que el CLB consta de cuatro capas y cada capa consiste de varios componentes como son *look-up-tables* (LUT), multiplexores, acarrees lógicos, compuertas AND y elementos secuenciales.

Las salidas de los CLB se encuentran conectadas a segmentos cableados usando puntos de interconexión programables (PIP). Si el bloque no es usado, estos PIPs son desconectados. La arquitectura de estas celdas varía con respecto a cada familia de dispositivos. Las celdas de entrada/salida (I/O), son la interconectividad entre el FPGA y el exterior. Éstas se pueden programar independientemente para tener la función de entrada, la función de salida con control tri-estado o un pin bidireccional. Este bloque también cuenta con flip-flops que pueden ser usados como *buffers* de entrada y salida. La celda de distribución maneja las señales de reloj hacia adentro del circuito. La interconectividad de la FPGA está restringida a celdas adyacentes exclusivamente, lo cual permite un menor consumo de potencia, lo que refleja un mayor nivel de integración. Estos dispositivos han ganado un gran mercado en las aplicaciones modernas de diseño digital teniendo FPGAs que cuentan desde las 50,000 a las 10'000,000 compuertas (Woods, 2008).

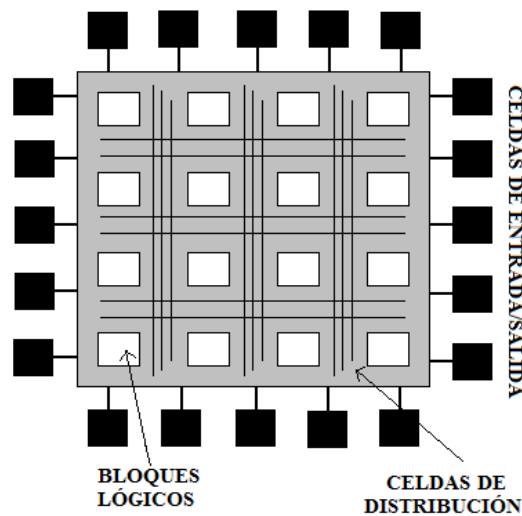


Figura 2.8 Conformación a bloques de un FPGA.

2.9 Técnicas de procesamiento digital de señales y algoritmos.

El procesamiento digital de señales es el análisis y procesamiento de señales discretas en tiempo y amplitud. Las técnicas y algoritmos del procesamiento digital de señales han logrado que el monitoreo se pueda implementar en circuitos digitales de costo y tamaño reducido.

2.9.1 Filtro de diezmado por promedio.

Este filtro es usado para la reducción del ruido de cuantificación mediante una técnica de sobremuestreo. En la ecuación (2.10) se muestra el algoritmo con el que se efectúa este filtrado. En la cual X es la señal a la que se le aplicará el filtro, k es el índice en tiempo discreto, i es el índice de diezmado, N es el orden del filtro o el número de muestras que contiene la ventana y finalmente, p es el promedio de la sumatoria de las N muestras seleccionadas (Rangel-Magdaleno, 2009).

$$p[k] = \frac{1}{N} \sum_{i=0}^{N-1} X[k - i] \quad (2.10)$$

2.9.2 Filtro FIR.

Los filtros de respuesta de impulso infinito (FIR) son llamados así debido a que si su entrada es un impulso, la salida será un número limitado de términos no nulos. Este tipo de filtros son recursivos, por lo tanto no tienen polos y son muy estables. Para obtener la salida de este filtro, solamente se emplean valores de las entradas actuales y las anteriores como se puede observar en la ecuación (2.11). En donde x es la señal a la que se aplica el filtro, n es el índice de la muestra filtrada, k es el índice de la sumatoria, b_k es el valor de los coeficientes del filtro y y_n es la señal resultante filtrada.

$$y[n] = \sum_{k=0}^{N-1} b_k x[n - k] \quad (2.11)$$

Como se puede observar, la ecuación (2.12) corresponde a la convolución de la señal de entrada $x[n]$ con la respuesta impulsional del filtro FIR $h[n]$.

$$y[n] = \sum_{k=0}^{N-1} h[k] x[n - k] \quad (2.12)$$

Aplicando la transformada Z a la respuesta impulsional del filtro FIR se tiene la ecuación (2.13)

$$H(z) = \sum_{k=0}^{N-1} h[k] z^{-k} = h_0 + h_1 z^{-1} + \dots + h_{N-1} z^{N-1} \quad (2.13)$$

De acuerdo con las ecuaciones (2.11) y (2.13), el diagrama a bloques de la estructura básica del filtro FIR es el que se muestra en la Figura 2.9 se muestra la estructura básica para la implementación del filtro FIR (García López, 2000).

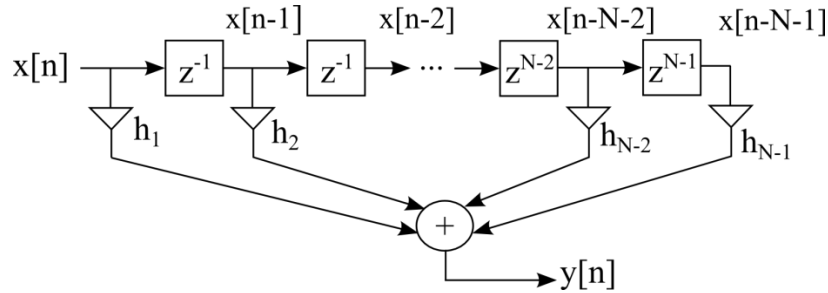


Figura 2.9. Diagrama a bloques de la estructura del filtro FIR.

2.9.3 Filtro complementario.

Este filtro es utilizado para estimar los ángulos de los ejes x , y y z , mediante los datos adquiridos por el acelerómetro y el giroscopio. Se usa la señal inercial obtenida con el giroscopio para tiempos cortos y la señal obtenida con el acelerómetro para tiempos largos. Esta última realiza la corrección de la deriva debido a estos tiempos. Un filtro complementario puede tomarse como un filtro Kalman de estado estacionario, el cual no considera ninguna descripción estadística del ruido que corrompe a las señales y es obtenido solamente por un simple análisis en el dominio de la frecuencia (Higgins, 2007). El filtro complementario es sencillo de implementar matemáticamente por lo que cuenta con un consumo bajo computacional. En la ecuación (2.14) se presenta el algoritmo desarrollado.

$$\theta_i = A_g(\theta_p + gyro * T) + A_a(acc_{ang}) \quad (2.14)$$

En la ecuación (2.14) la variable θ_i corresponde al ángulo calculado, θ_p es el ángulo anterior calculado, $gyro$ es el valor de la señal adquirida por el giroscopio, acc_{ang} es el ángulo calculado por el acelerómetro, A_g es la ganancia deseada para el valor del giroscopio y A_a es la ganancia deseada para el valor del acelerómetro.

El ángulo de la señal proveniente del acelerómetro acc_{ang} se obtiene de acuerdo con cada eje, como se muestra en la ecuación (2.15).

$$acc_{angx} = atan2 \frac{acc_x}{acc_z} \quad (2.15)$$

Por lo que para calcular el ángulo del eje x se tiene la ecuación (2.16)

$$\theta_{ix} = A_g(\theta_{px} + gyro_x * T) + A_a(acc_{angx}) \quad (2.16)$$

Para calcular el ángulo de los ejes y y z se tiene la misma ecuación (2.16) con sus correspondientes valores para cada uno.

Otros de los valores importantes en la ecuación (2.14) son las constantes A_g y A_a los cuales corresponden a la ganancia que se le asigna a la medición del ángulo de cada sensor. Para la asignación de estos valores, debe existir la correspondencia que se presenta en la ecuación (2.17).

$$A_g + A_a = 1 \quad (2.17)$$

Mediante este filtro se evita que en la estimación de los ángulos de cada uno de los ejes exista ruido proveniente del acelerómetro por su alta sensibilidad a las vibraciones y del giroscopio debido a la deriva. Logrando, de esta manera una mejor estimación de la inclinación del objeto a analizar (Gaydou, 2011).

2.9.4 Integral discreta

Con frecuencia se presenta la necesidad de integrar una función tabulada en donde los valores de x y $f(x)$ se dan en un conjunto de puntos discretos, como es el caso de datos experimentales. En estos casos es necesario considerar métodos numéricos para obtener un

valor aproximado de la cantidad de interés, independientemente de lo difícil que sea la función a integrar. La operación de la integral se observa en la ecuación (2.18).

$$I(f) = \int_a^b f(x)dx \quad (2.18)$$

Donde el intervalo de integración $[a,b]$ es finito. Existe un problema al usar los métodos analíticos de integración, en el que es posible que F no se pueda expresar en términos de funciones elementales o no se pueda evaluar fácilmente. Como es el caso en la existencia de una tabla de valores discretos.

2.9.4.1 Regla de los trapecios

Esta regla es un método de integración numérica, el cual, se basa en aproximar el valor de la integral de $f(x)$ por el de la función lineal que pasa por los puntos $(a, f(a))$ y $(b, f(b))$. Esta integral es igual al área del trapecio bajo la gráfica de la función lineal.

Regla del trapecio simple

Considerando la función $f(x)$, cuya grafica está entre los extremos a y b como se puede observar en la Figura 2.10.

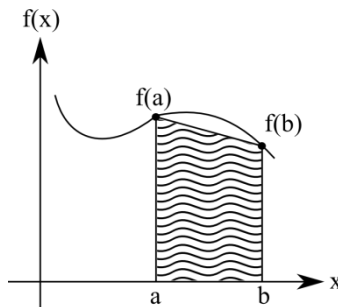


Figura 2.10. Gráfica de trapecio.

Si se utiliza un polinomio $P(x)$ de primer grado como una aproximación de $f(x)$ se tiene la ecuación (2.19).

$$P(x) = f(a) \frac{x - b}{a - b} + f(b) \frac{x - a}{b - a} \quad (2.19)$$

La cual es equivalente a la ecuación (2.20).

$$P(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \quad (2.20)$$

El área bajo esta línea recta será una aproximación del área bajo la curva entre los límites a y b . Integrando la ecuación (2.20), se obtiene la ecuación (2.21).

$$\int_a^b \left(f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \right) dx \cong (b - a) \frac{f(a) + f(b)}{2} \quad (2.21)$$

La ecuación (2.21) es conocida como regla de trapecio simple. Esta regla aproxima el área bajo la curva mediante el área del trapecio bajo la línea recta que une $f(a)$ y $f(b)$.

Regla del trapecio compuesta

Una aproximación suficiente al área bajo la curva se obtiene dividiéndola en n segmentos del ancho $h=(b-a)/n$ y aproximando el área de cada segmento mediante un trapecio, como se puede observar en la Figura 2.11.

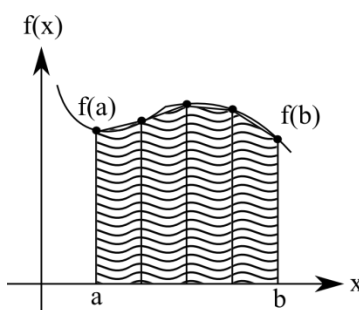


Figura 2.11. Gráfica de trapecio compuesta.

Sea $P=\{x_0, x_1, \dots, x_n\}$ la partición que se forma al hacer la subdivisión de la curva. Usando propiedades de la integral tenemos la ecuación (2.22)

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{n-1}}^{x_n} f(x)dx \quad (2.22)$$

Aplicando la regla del trapecio, se obtiene la ecuación (2.23)

$$\int_a^b f(x)dx \cong h \frac{f(x_0) + f(x_1)}{2} + h \frac{f(x_1) + f(x_2)}{2} + \dots + h \frac{f(x_{n-1}) + f(x_n)}{2} \quad (2.23)$$

Si se agrupan los términos, se obtiene la ecuación (2.24), la cual es conocida como la regla del trapecio múltiple o compuesta (Sese-Sanchez, 2013).

$$\int_a^b f(x)dx \cong \frac{h}{2} \left(f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right) \quad (2.24)$$

2.9.5 Derivada discreta

El cálculo del valor numérico de las derivadas de una función $y(x)$ con expresión analítica conocida no plantea, en principio, grandes problemas. Sin embargo, el cálculo de las derivadas numéricas de una función dada por una tabla de datos es un asunto que presenta incertidumbres. Para definir la derivada discreta de una función $f(n)$ se cuenta con la ecuación (2.25), la cual es denotada por $\Delta_n f(n)$ (Sese-Sanchez, 2013).

$$\Delta_n f(n) = f(n + 1) - f(n) \quad (2.25)$$

3 METODOLOGÍA.

En este capítulo se presenta la metodología seguida para el desarrollo del procesador digital inalámbrico, comenzando con el planteamiento general del sistema y sus características. Además, se describe el funcionamiento del sistema presentando los bloques que lo integran y el funcionamiento individual de éstos.

3.1 Sistema general.

Partiendo de un concepto general, se definen las partes que conforman el sistema de procesamiento inalámbrico para las señales inerciales. El sistema a bloques general del sistema se muestra en la **¡Error! No se encuentra el origen de la referencia..** El procesador digital consta de dos partes. El sistema esclavo y el sistema maestro.

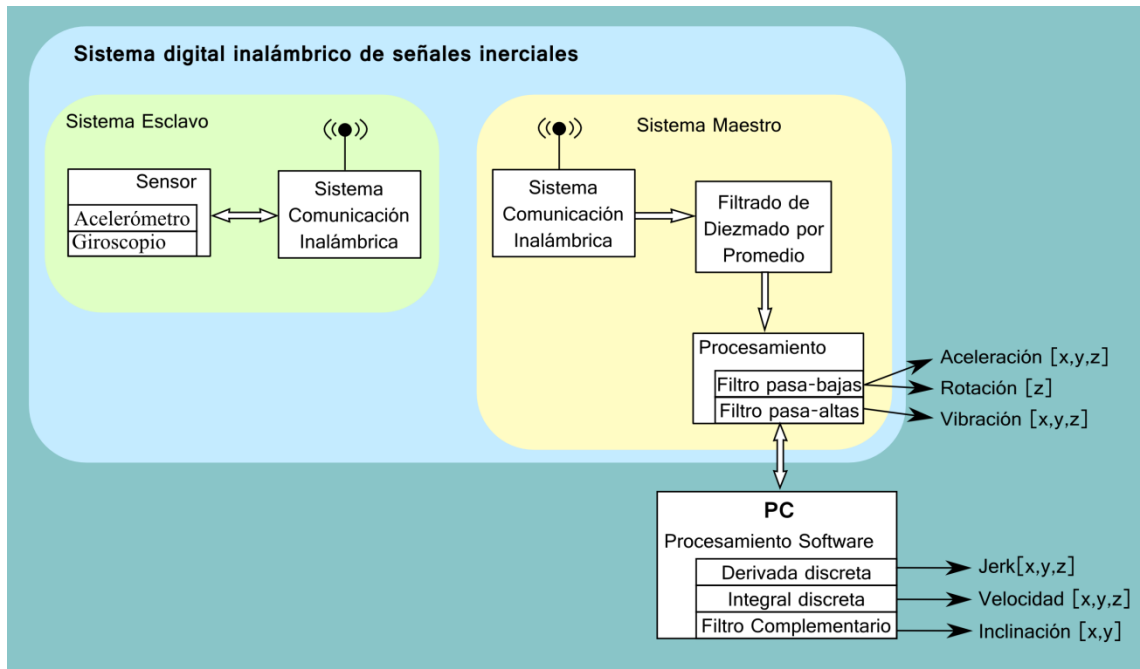


Figura 3.1. Diagrama a bloques general del sistema digital inalámbrico de señales inerciales.

En la Figura 3.2 se presenta el sistema esclavo del procesador digital inalámbrico. Este sistema es instalado en la máquina o robot del que se desea adquirir las señales inerciales. Para este fin, se hace uso de un sensor que integra un acelerómetro y un giroscopio. Las señales de aceleración y velocidad angular son leídas en los tres ejes, por lo que el sensor proporciona una lectura de seis datos de dieciséis bits cada uno. Para la comunicación entre el manejador y el sensor se utiliza el protocolo RS-232. El manejador se encarga de efectuar la sincronización para entregar los datos leídos del sensor hacia el transceptor nRF24L01. Esta comunicación se efectúa mediante el protocolo SPI. Además de enviar los datos de las señales inerciales, el manejador se encarga de configurar el transceptor para la comunicación inalámbrica.

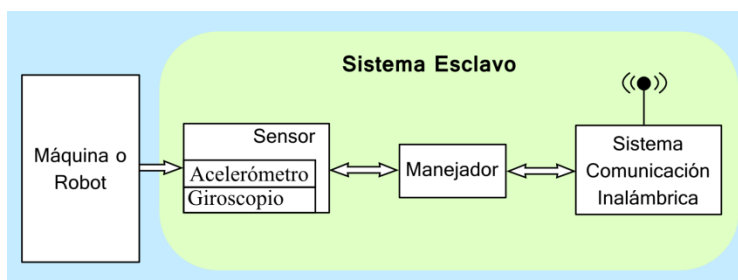


Figura 3.2. Diagrama a bloques de sistema esclavo.

El sistema maestro se muestra en la Figura 3.3. El sistema de comunicación inalámbrica es el encargado de realizar la configuración del transceptor para la comunicación inalámbrica utilizando el protocolo SPI. Una vez que se ha recibido la lectura de las señales inerciales enviadas inalámbricamente por el sistema esclavo, el sistema maestro aplica un filtro de diezmo por promedio para obtener las señales libres de ruido. Posteriormente, se aplica un filtro pasa bajas para filtrar las señales inerciales definitivas y de la cual se obtiene la aceleración y la rotación. Al final, se aplica un filtro pasa altas para obtener la señal inercial de vibración en cada uno de los ejes. Estos procesamientos de señales se realizan en paralelo, de tal manera que se realizan los filtrados al mismo tiempo para las seis señales inerciales. Una vez obtenidas estas señales, se almacenan en la memoria RAM externa, con el fin de que sean enviadas posteriormente por medio del protocolo RS-232 hacia la PC para obtener la estimación de las señales de jerk, velocidad, e

inclinación. En la PC se grafican estas señales, junto con las obtenidas mediante el procesamiento en hardware.

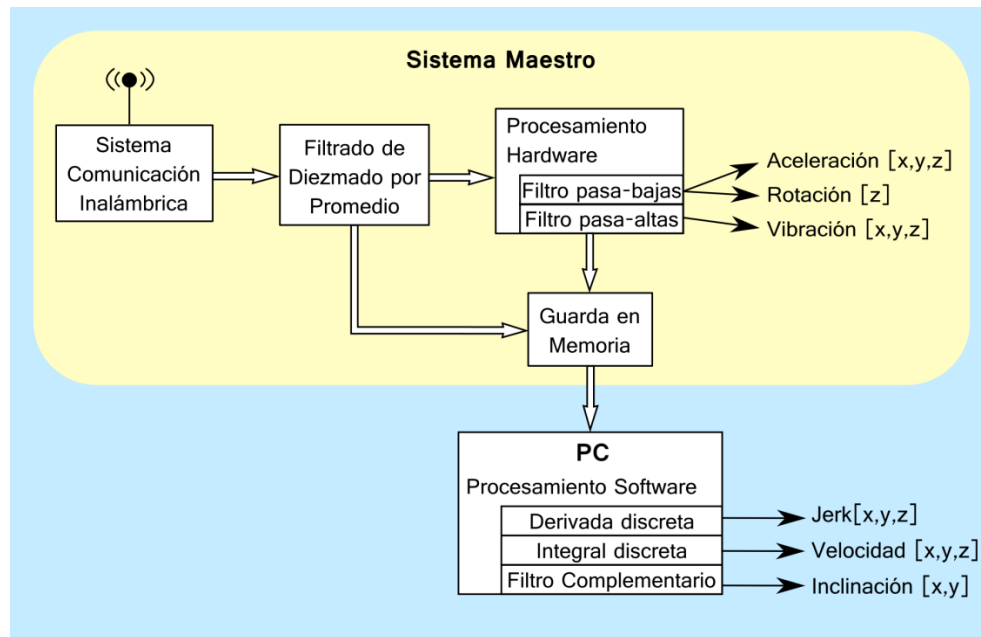


Figura 3.3. Diagrama a bloques de sistema maestro.

Para obtener las señales inerciales de una máquina o robot se realiza el procedimiento que me muestra en el diagrama de flujo de la Figura 3.4. Inicialmente, el sistema maestro está en modo de transmisor y envía una solicitud inalámbrica hacia el sistema esclavo, que se encuentra en modo receptor, para adquirir una lectura de los sensores. Recibida la petición, el sistema esclavo envía una señal al sensor para adquirir las lecturas de los tres ejes del acelerómetro y del giroscopio mediante el protocolo RS-232. Una vez obtenidas las señales inerciales, el sistema esclavo concatena las lecturas provenientes del sensor y cambia su modo inalámbrico a transmisor, mientras que el sistema maestro lo cambia a receptor. La concatenación se realiza para envían las seis lecturas en una sola trama de comunicación inalámbrica. Una vez lista la comunicación, se envía la trama de datos inalámbricamente. Después de que se ha recibido la lectura de las señales inerciales, el sistema maestro separa y filtra con el diezmado la señal. Una vez filtradas, a las señales se les aplica el filtro pasa-bajas y el filtro pasa-altas, con el fin de obtener las aceleraciones, la rotación y las vibraciones de la maquinaria. Ya procesados,

cada una de las lecturas es guardada en una localización de la memoria RAM. Una vez que se ha realizado esto, el sistema maestro cambia su estado a transmisor y el sistema esclavo a modo receptor para efectuar el siguiente muestreo. El procedimiento anterior es repetido, hasta que se ha guardado la cantidad de muestras deseadas por el usuario.

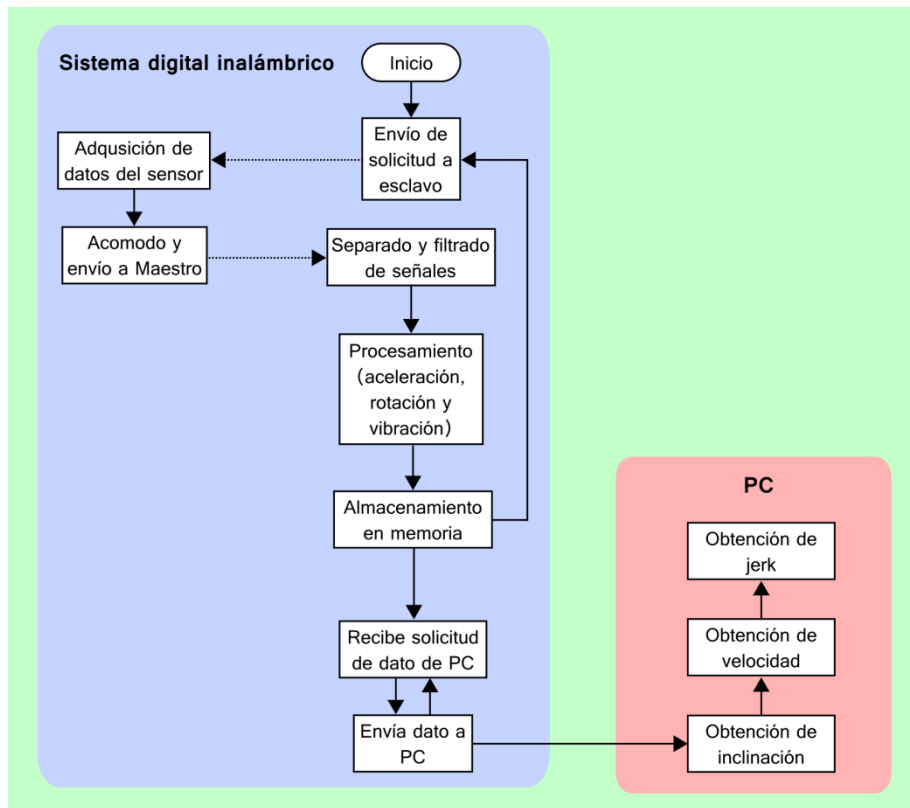


Figura 3.4. Diagrama de flujo de la operación del procesador digital inalámbrico de señales inerciales.

Ya que se han recibido todas las muestras deseadas, el sistema maestro espera la señal proveniente de la PC. Una vez recibida esta señal, el sistema maestro envía la lectura del dato contenido en la memoria RAM y vuelve a su estado de espera para enviar el siguiente dato. Este procedimiento es repetido hasta que el sistema maestro termina de enviar los datos almacenados en la memoria. Esta comunicación se realiza mediante el protocolo RS-232. Ya recibidos los datos en la PC, se procede a realizar los cálculos requeridos para la estimación de la inclinación, la velocidad y el jerk.

3.2 Manejador para el nRF14L01.

El manejador es el encargado de realizar la configuración y comunicación con el transceptor. Este manejador es utilizado tanto en el sistema maestro como en el esclavo, debido a que en ambos es necesario configurar el transceptor pasando del estado de transmisor a receptor y viceversa. El bloque general del manejador para el nRF24L01 se muestra en la Figura 3.5.

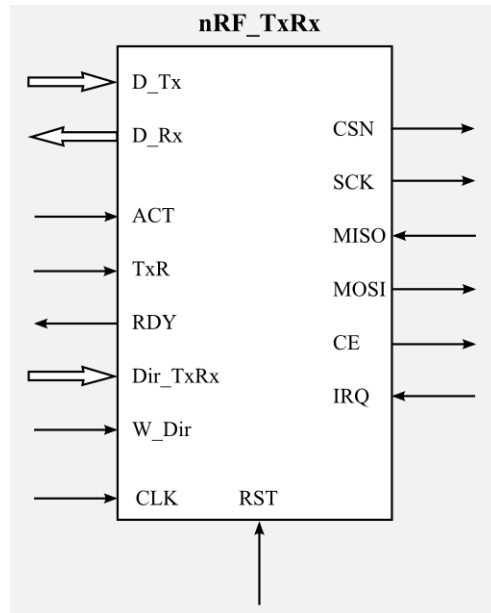


Figura 3.5. Bloque de manejador nRF24L01.

Como se puede ver en la Figura 3.5; **Error! No se encuentra el origen de la referencia.**, el manejador cuenta con las señales de Dato a enviar (D_Tx) y Dato recibido (D_Rx). La señal D_Tx y D_Rx varían de longitud, debido a que el transceptor cuenta con la propiedad de enviar y recibir desde 1 hasta 32 bytes de acuerdo con la configuración que el usuario le proporcione. El tamaño del dato es modificable desde los valores de configuración inicial del transceptor con los que cuenta el bloque manejador en su interior. La señal ACT es utilizada para activar el módulo del manejador, esta señal es activada en bajo. La señal TxR es para elegir el modo en que actuará el transceptor, ya sea como transmisor poniendo en bajo esta señal, o como receptor poniéndola en alto. La señal RDY indica en bajo cuando se ha terminado de ejecutar el envío o recepción de un dato. Esto es de gran ayuda para tener un buen control de las acciones. Otra propiedad con la que cuenta

este manejador es la de cambiar la dirección de los transceptores después de cada envío o recepción. Esto con la finalidad de poder crear una red de transceptores en los cuales se tiene un maestro y múltiples esclavos. Para realizar esto es necesario que el maestro se comunique con cada esclavo, los cuales, cuentan con una dirección diferente en el mismo canal de radiofrecuencia. Para realizar el cambio de dirección se tienen las señales Dir_TxRx y W_Dir. En la primera se coloca la dirección del transceptor con el cual se desea comunicar y la segunda activa la escritura de esta dirección en el transceptor. En ocasiones no es necesario cambiar la dirección, por lo cual no se activa la señal W_Dir y el transceptor mantiene el valor de dirección último en el que se le configuró. Esta señal se debe de establecer antes de activar el modo con la señal ACT.

El manejador cuenta con las señales utilizadas en el protocolo SPI para comunicarse con el transceptor nRF24L01 como son CSN, utilizada para seleccionar el transceptor; SCK, la cual es el reloj del protocolo; MOSI, que es la línea de comunicación del Maestro con el Esclavo; MISO, la línea de comunicación del Esclavo con el Maestro; CE, que habilita la comunicación con el transceptor e IRQ, la señal encargada de indicar cuando el transceptor ha terminado de enviar un dato o cuando se tiene en la pila un dato que ya ha sido recibido.

El módulo del manejador del transceptor nRF24L01 se encuentra estructurado como se muestra en la Figura 3.6. El bloque que realiza todo el control del módulo es la máquina de estados FSM. A este bloque entran las señales de control ACT, TxR, W_Dir y proporciona la señal RDY. Además de estas señales, el módulo FSM tiene las señales de activación de los registros de desplazamiento RegDes8n_8, RegDes8_8n y RegDes40_8. El registro de desplazamiento RegDes8n_8 es utilizado para almacenar el dato que se desea enviar, con un desplazamiento en tramas de 8 bits de para realizar la comunicación con el protocolo SPI. Este registro puede ser de hasta 32 *bytes*, siendo el máximo valor de tramas que puede enviar el transceptor en una solicitud de datos. El registro de desplazamiento RegDes8_8n es el inverso del registro anterior, es decir almacena cada *byte* recibido mediante un desplazamiento. Recibe hasta un máximo de 32 *bytes*. El Registro de desplazamiento RegDes40_8 es utilizado para configurar la dirección del transceptor, esta dirección está constituida por cinco tramas de 8 *bits*, por lo que es necesario un registro con 40 *bits* de entrada y 8 de salida para ser enviado por el protocolo SPI.

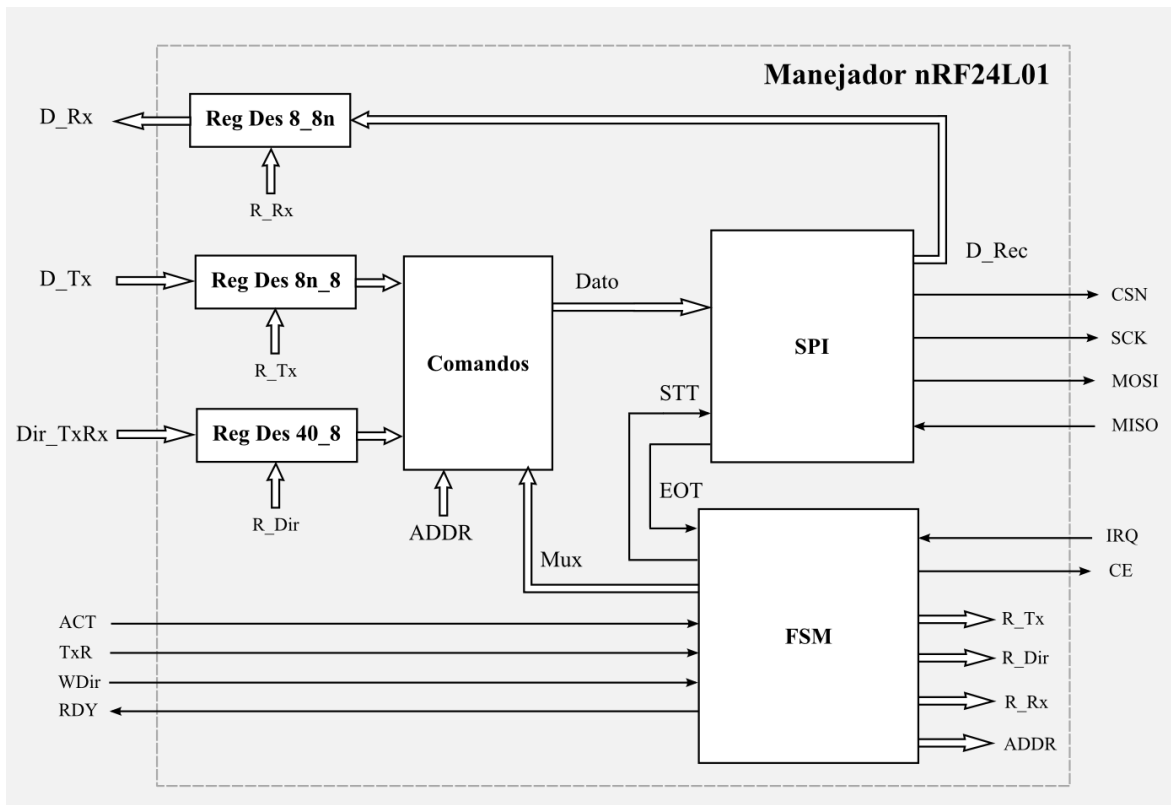


Figura 3.6. Estructura del bloque del manejador nRF24L01.

Para comunicarse con el transceptor se utiliza el protocolo SPI, el cual es implementado en el módulo que lleva éste nombre y que se muestra en la Figura 3.6. Este módulo cuenta con una señal de inicio de comunicación (STT) y una que indica el fin de la trama que se envió (EOT). Además contiene las señales necesarias para la comunicación SPI como son CSN, SCK, MOSI y MISO. La secuencia de datos para realizar la comunicación inalámbrica con el transceptor es controlada por medio de la señal MUX proveniente del bloque de las máquinas de estados FSM hacia el bloque de Comandos. Dentro de éste bloque se encuentra un módulo de una memoria ROM utilizada para configurar las características iniciales del transceptor como son la frecuencia de comunicación, el ancho de banda, la configuración como receptor o transmisor, la dirección correspondiente a cada transceptor, el accionamiento de las banderas, el número de tramas a enviar o recibir, la potencia de trabajo, entre otras. La secuencia con la que se envían los datos hacia el transceptor para su configuración es controlada por el módulo FSM con la señal ADDR. Esta señal es establecida para enviar una trama de 8 bits que corresponde a la

dirección en la que se encuentra almacenado un valor en la configuración, mientras q la máquina de estados espera el fin de la transmisión proveniente del módulo SPI. Al recibir esta señal de fin, la dirección es aumentada y envía el siguiente valor correspondiente a la configuración. El procedimiento se repite hasta terminar con todos los valores.

Además de estos valores de configuración inicial, el bloque de Comandos contiene los valores para configurar el transceptor como transmisor, para configurarlo como receptor, para realizar la transmisión inalámbrica de datos, para realizar la recepción inalámbrica de datos, para borrar el registro interno del transceptor llamado STATUS, para borrar su pila FIFO y para escribir la dirección del transceptor. De acuerdo con los valores de las señales ACT, TxR y W_Dir ingresados, el módulo FSM envía la dirección MUX correspondientes hacia el módulo Comandos para la selección de las instrucciones que el transceptor codifica en cada una de las acciones.

3.2.1 Módulo de Comandos.

En este módulo se encuentran los valores de configuración y de trabajo para el transceptor nRF24L01. Se encuentra dividido en dos partes. La primera es la memoria ROM, encargada de enviar los datos para la configuración inicial del transceptor y la segunda es el multiplexor de las operaciones, el cual contiene los comandos de trabajo.

3.2.1.1 Memoria ROM.

La memoria ROM que se encuentra dentro del bloque de Comandos, contiene los valores de la configuración inicial del transceptor. Como se puede observar en la Tabla 3.1, se encuentra dividida en tres secciones. La primera tiene el valor de la dirección del comando, la segunda tiene el valor que se le ingresa y la tercera una explicación breve de su acción. Estos valores permiten realizar la conexión entre transceptores, esperando las instrucciones necesarias para enviar y recibir los datos. Para modificar el estado del transceptor como transmisor o receptor es necesario modificar el valor del comando de escritura en el registro de configuración después de que se ha configurado el transceptor. Todos los demás valores quedan sin modificar. Los valores de la Tabla 3.1 presentados con el número hexadecimal *FF* son valores de no operación. Estos valores son utilizados para limitar entre un valor de configuración y otro, con el fin de que el transceptor los interprete correctamente.

Tabla 3.1. Comandos y valores que contiene el módulo ROM.

Dirección (HEX)	Valor (HEX)	Acción.
00	FF*	No Operación.
01	20	Comando Escritura en el Registro de Configuración.
02	5E	Configuración como Transmisor y activación de bandera IRQ.
03	FF*	
04	21	Comando Escritura en el Registro de Auto-Acknowledgment.
05	00	Desactivación de configuración.
06	FF*	
07	22	Comando Escritura en el Registro de Dirección de Rx disponible.
08	01	Disponible la primer línea.
09	FF*	
0A	23	Comando Escritura en el Registro Tamaño de Dirección.
0B	03	Cinco byte para la dirección.
0C	FF*	
0D	24	Comando Escritura en el Registro Retransmisión Automática.
0E	00	Desactivado.
0F	FF*	
10	25	Comando Escritura en el Registro Radiofrecuencia del canal.
11	04	Cuarto valor de frecuencia pre-establecido.
12	FF*	
13	26	Comando Escritura en el Registro Características de RF.
14	0F	Frecuencia de comunicación aérea de 2Mbps y 0dBm.
15	FF*	
16	2 ^a	Comando Escritura en el Registro Dirección como Transmisor.
17	78	Primer trama de dirección.
18	78	Segunda trama de dirección.
19	78	Tercera trama de dirección.
1A	78	Cuarta trama de dirección.
1B	78	Quinta trama de dirección.
1C	FF*	
1D	30	Comando Escritura en el Registro Dirección como Receptor.
1E	78	Primer trama de dirección.
1F	78	Segunda trama de dirección.
20	78	Tercera trama de dirección.
21	78	Cuarta trama de dirección.
22	78	Quinta trama de dirección.
23	FF*	
24	31	Comando Escritura en el Registro Cantidad de Tramas de Carga.
25	01	Un byte de envío o recepción.
26	FF*	

FF* son valores de no operación. Estos valores son utilizados para delimitar el fin del comando junto con su valor.

3.2.1.2 Multiplexor de operación.

Como se mencionó, el multiplexor de operación es una parte importante del módulo del manejador del transceptor nRF24L01. Este es el encargado de enviar la operación hacia el módulo SPI de acuerdo con el valor en la dirección seleccionada por el módulo FSM. En la Tabla 3.2 se muestra la dirección correspondiente a cada una de las operaciones, los valores de las operaciones y su acción.

Tabla 3.2. Comandos y valores para las distintas operaciones del Transceptor.

Dirección (HEX)	Valor (HEX)	Operación.
0	Variable	Al colocar esta operación envía uno a uno los valores de la ROM para realizar la configuración del Transceptor.
1	20	Comando para acceder al registro de configuración del transceptor.
2	5E	Valor para configurar el Transceptor como transmisor.
3	20	Comando para acceder al registro de configuración del transceptor.
4	3F	Valor para configurar el Transceptor como receptor.
5	A0	Comando para indicar el envío de un dato inalámbricamente.
6	Dato a enviar	Tramas de 8 bits a enviar inalámbricamente.
7	61	Comando para indicar la descarga de datos del transceptor recibidos inalámbricamente.
8	39	Comando para acceder al registro de STATUS.
9	6E	Valor para borrar el estado del STATUS.
A	30	Comando para acceder al registro de dirección de transmisión.
B	2A	Comando para acceder al registro de dirección de recepción.
C	Dirección	Dirección de transmisión o recepción.
D	E2	Borrar pila FIFO del transceptor.
E	FF	No operación.

3.2.2 Módulo SPI.

El transceptor utiliza el protocolo SPI para configurar los registros de sus propiedades, leer dichos registros, enviar las tramas de datos y leer las tramas recibidas. Por lo que es necesario un módulo que se encargue de realizar este protocolo, como el presentado en la Figura 3.7. El módulo SPI cuenta con las señales CSN, SCK, MISO y MOSI propias de este protocolo, las cuales permiten la comunicación con el transceptor.

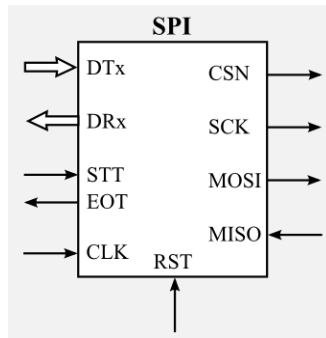


Figura 3.7. Módulo del protocolo SPI.

El módulo cuenta con la señal de dato de transmisión DTx de 8 bits, así como la señal del dato de recepción DRx con la misma longitud. La señal de inicio de comunicación STT indica cuando se desea que comience el protocolo SPI. Cada vez que se termina el envío de una trama de 8 bits la señal EOT baja por un ciclo de trabajo y vuelve a su estado en alto. Además, el módulo cuenta con la señal de reloj CLK y reset RST. Para un análisis más profundo se presenta en la Figura 3.8 la conformación del módulo SPI.

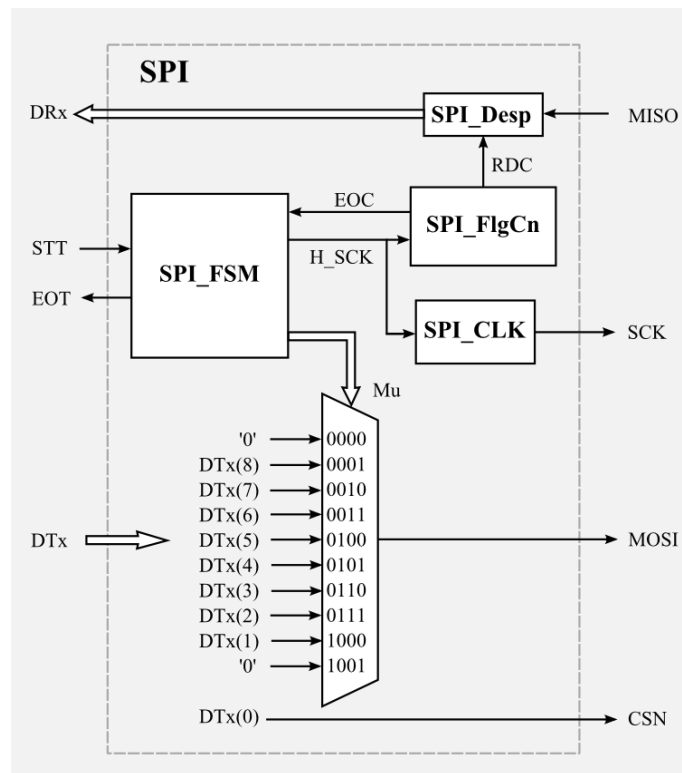


Figura 3.8. Estructura interna del módulo SPI.

El módulo que controla el envío de datos del multiplexor, la activación del módulo SPI_CLK y del módulo SPI_FlgCn es el módulo de la máquina de estados SPI_FSM. Recibe las señales de inicio STT y EOC para controlar los eventos. Además proporciona las señales de EOT, para indicar el fin de la trama enviada y H_SCK, para habilitar los módulos SPI_CLK y de SPI_FlgCn.

Para comenzar con su análisis es importante conocer el diagrama de flujo del comportamiento de la máquina de estados del módulo SPI. Este diagrama es mostrado en la Figura 3.9.

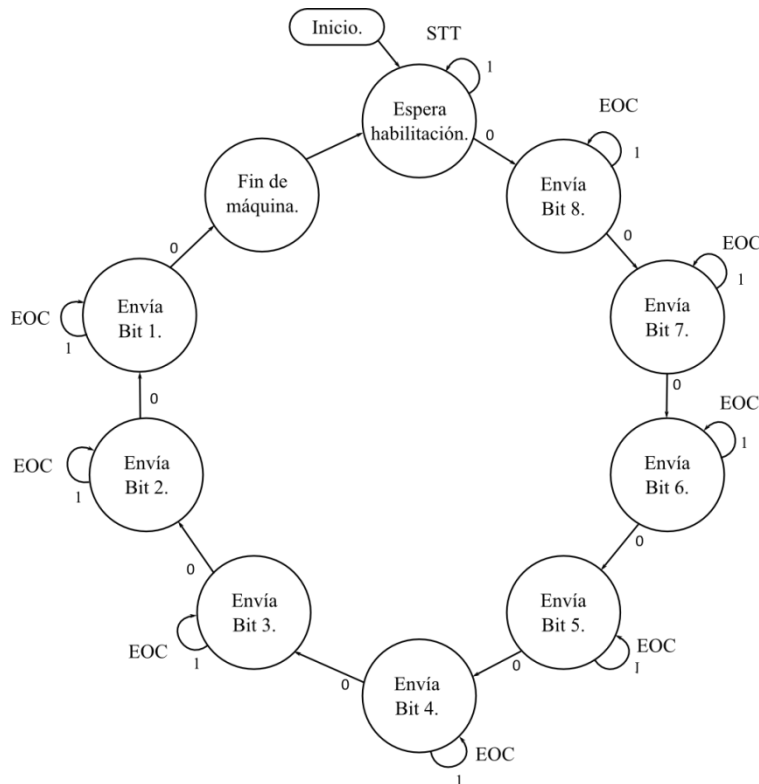


Figura 3.9. Máquina de estados del módulo SPI.

Como se observa en la Figura 3.9, la máquina de estados espera la habilitación de la señal STT del módulo SPI. Cuando esta señal se encuentra habilitada, la máquina pasa al primer estado, en donde se envía el *bit* 8 del dato DTx. Espera la indicación de la bandera de fin de cuenta EOC y pasa al siguiente estado. En este estado envía el bit 7 de DTx y espera nuevamente la señal EOC. El procedimiento se repite hasta enviar los 8 *bits* en protocolo SPI. Después de enviar el último *bit*, la máquina pasa al estado en donde envía la

señal de fin del proceso EOT, indicando que la trama de 8 *bits* ha sido enviada por el protocolo SPI correctamente y que se encuentra disponible para una nueva trama.

El bloque SPI_FlgCn indica el flanco en donde la lectura del *bit* MISO se realiza, de acuerdo al modo D de reloj del protocolo SPI mostrado en la Figura 2.2, así como el cambio de *bit* de la señal MOSI. La simulación del comportamiento de estas señales se muestra en la Figura 3.10. Al inicio del borde positivo de la señal SCK, se activa la adquisición del *bit* MISO por medio del bajo de la señal RDC y al final del mismo borde positivo, se indica el cambio de *bit* de la señal MOSI en el próximo ciclo de reloj con el bajo de la señal EOC.

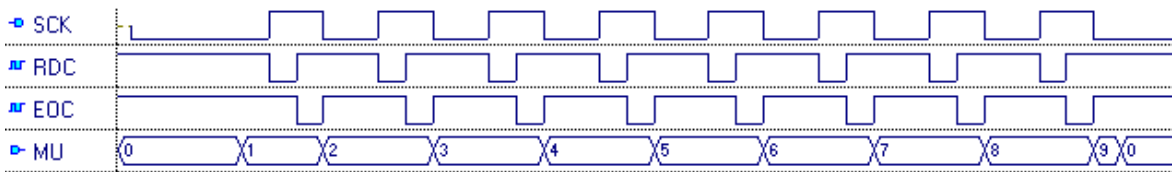


Figura 3.10. Simulación de las señales del bloque SPI_FlgCn.

El bloque SPI_FlgCn es activado por medio de la señal H_SCK, como se observa en la Figura 3.8. Esta señal activa de igual manera el bloque SPI_SCK generando la señal de reloj SCK. La señal de reloj se utiliza para saber el punto en que se debe de realizar la adquisición del dato de entrada MISO y el cambio de la señal Mu. Esta señal proveniente de la máquina de estados, concuerda con el *bit* correspondiente en el protocolo SPI. Como se puede ver en la Tabla 3.3.

Tabla 3.3. Salidas correspondientes a cada valor de Mu.

Mu	Salida
0	'0'
1	Bit 8
2	Bit 7
3	Bit 6
4	Bit 5
5	Bit 4
6	Bit 3
7	Bit 2
8	Bit 1
9	'0'

El bloque SPI_Desp es un registro serie-paralelo en el que se ingresa el *bit* MISO proveniente del transceptor nRF24L01 y se obtiene una trama de 8 *bits* de acuerdo con el protocolo SPI. Este registro es controlado por la señal RDC del bloque SPI_FlgCn.

3.2.3 FSM.

En la Figura 3.5 se tiene el diagrama general del módulo del manejador del transceptor nRF24L01. Para que se realice la selección de las operaciones necesarias para la comunicación inalámbrica, ya sea de transmisión o de recepción, es necesario el uso de máquinas de estados. En la Figura 3.11 se muestra el diagrama a bloques de las máquinas de estado.

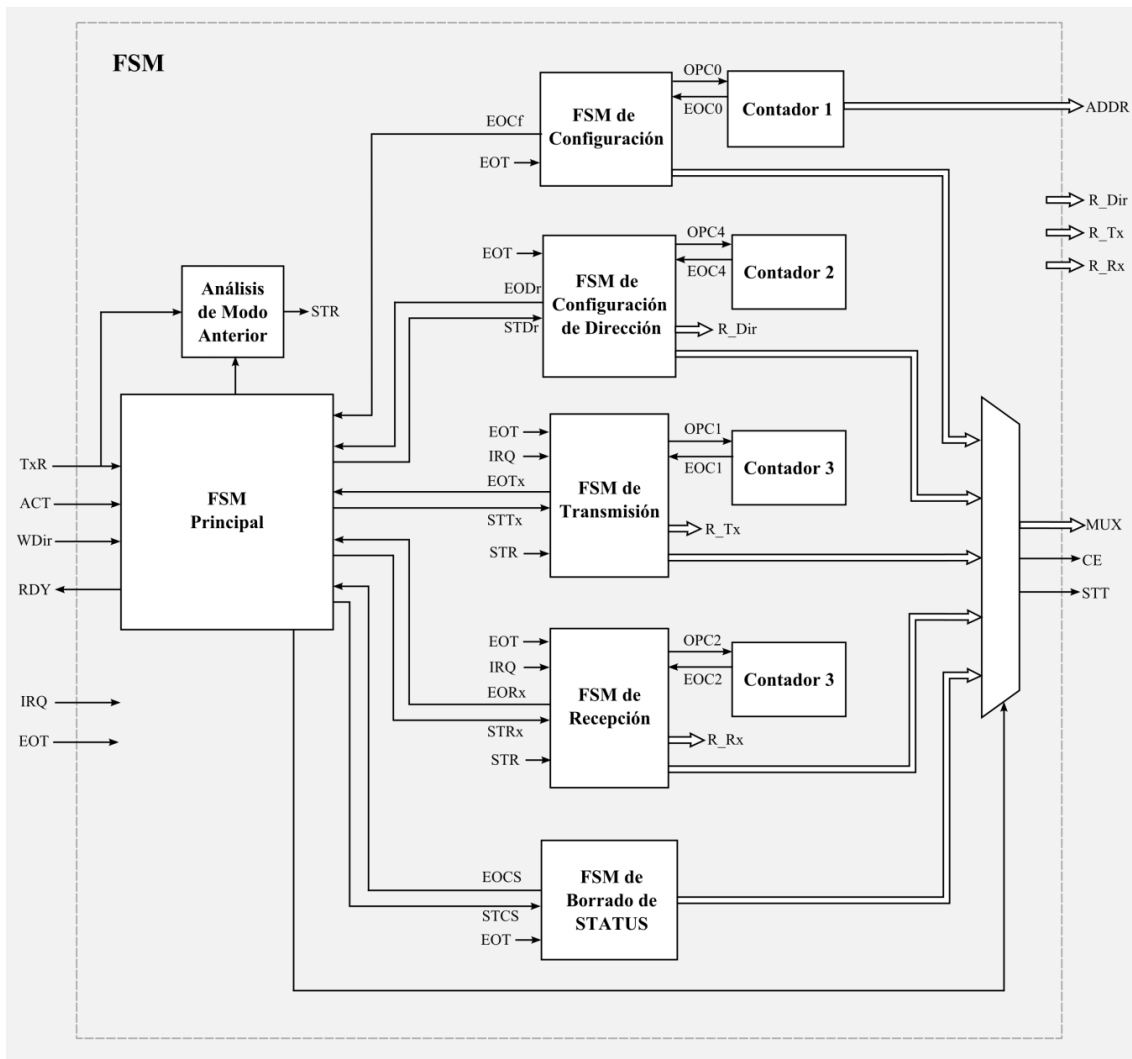


Figura 3.11. Diagrama a bloques del bloque FSM.

Como se puede observar en la Figura 3.11 el bloque FSM Principal se encarga de manipular las demás máquinas de estados, así como el multiplexor de salida de acuerdo a las señales TxR, ACT y W_Dir. Con el multiplexor de este módulo se controlan las salidas que actúan sobre el bloque de comandos de la Figura 3.5 por medio de la señal MUX, el inicio del módulo del protocolo SPI con la señal STT y habilita el transceptor con la señal CE.

Este módulo se conforma de otras máquinas de estados utilizadas para la configuración inicial, para configurar la dirección del transceptor, para realizar la transmisión de un dato inalámbricamente, para realizar la recepción de un dato inalámbricamente y para borrar el registro de STATUS del transceptor. Además, el módulo FSM cuenta con un módulo de análisis del estado anterior, esto con el fin de evitar la reconfiguración del transceptor. Es decir, si el transceptor se encuentra en modo de transmisor y se requiere cambiar a modo receptor, el módulo FSM de recepción realiza lo necesario para configurar el transceptor. En caso de que se mantenga en modo transceptor se evita su reconfiguración. De igual manera, si se tiene el transceptor en modo receptor y se cambia a modo transmisor, el módulo FSM de transmisión realiza la configuración necesaria. Si el transceptor se mantiene en el modo receptor, evita la reconfiguración en este mismo modo. El guardado del estado anterior se realiza con registros como se puede ver en la Figura 3.12.

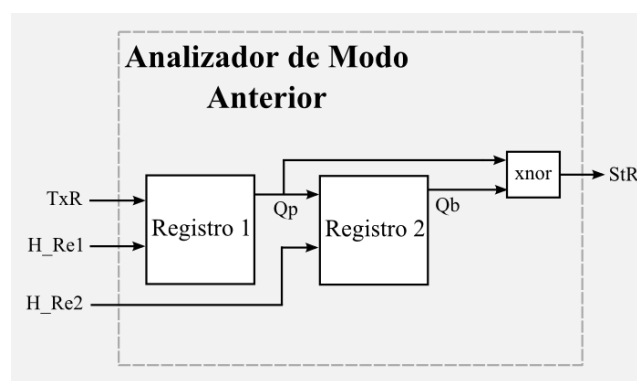


Figura 3.12. Diagrama a bloques del analizador de modo anterior del transceptor.

El analizador de modo anterior cuenta con dos registros. El registro 1 guarda el estado actual y el registro 2 guarda el estado anterior. La máquina de estados es la

encargada de habilitar los registros. Al seleccionar el modo de operación del transceptor, la máquina de estados principal habilita el registro 1 para almacenar la señal. Se realiza la comparación con la operación xnor entre el estado actual y el estado anterior guardado en el registro 2. Si las señales son iguales, la señal StR se mantiene en alto. Si son diferentes, la señal se mantiene en bajo. Esta señal es analizada en la máquina de estados de transmisión y en la máquina de estados de receptor.

El módulo FSM utiliza contadores con habilitación y reinicio para el control del número de tramas a enviar. El contador 1 tiene un módulo 39, el contador 2 tiene un módulo de 5 y el contador 3 debe de coincidir con el número de tramas a enviar inalámbricamente configuradas en el transceptor. Este contador puede tener un módulo desde 1 hasta 32. Es necesario realizar el análisis del funcionamiento de cada una de las máquinas de estado que conforman el bloque FSM.

3.2.3.1 FSM principal.

La máquina de estados principal controla las demás máquinas de acuerdo a la selección realizada por las señales TxR, ACT y W_Dir. La señal TxR permite seleccionar el modo de operación del transceptor, si es bajo opera como transmisor y si es alto opera como receptor. La señal ACT es la habilitación del modo de operación con activación en bajo. Finalmente, W_Dir indica si se desea ingresar una nueva dirección de transmisor y receptor para el transceptor. En bajo realiza esta operación y en alto no realiza la reconfiguración.

En la Figura 3.13 se muestra el diagrama de flujo de la máquina de estados principal. Después de iniciar, la máquina de estados principal espera a que la máquina de estados de configuración le envíe la señal de fin. La máquina de estados de configuración inicia su operación inmediatamente después de que la señal de *Reset* de la tarjeta es enviada. Esta es la encargada de enviar todas las señales necesarias para la configuración inicial del Transceptor. Cuando la señal de fin de la configuración del transceptor es recibida por la máquina principal, esta se mantiene en estado de espera hasta recibir la señal de activación ACT en bajo. Cuando la máquina principal es activada, se evalúa la operación seleccionada en la señal TxR. Si la señal se encuentra en bajo, se realiza el procedimiento para enviar los datos por el transceptor. Si la señal está en alto, se realiza el procedimiento

para la recepción de datos. Una vez que se selecciona el modo en que se operará, se guarda el tipo de operación en el registro 1 del analizador de modo anterior.

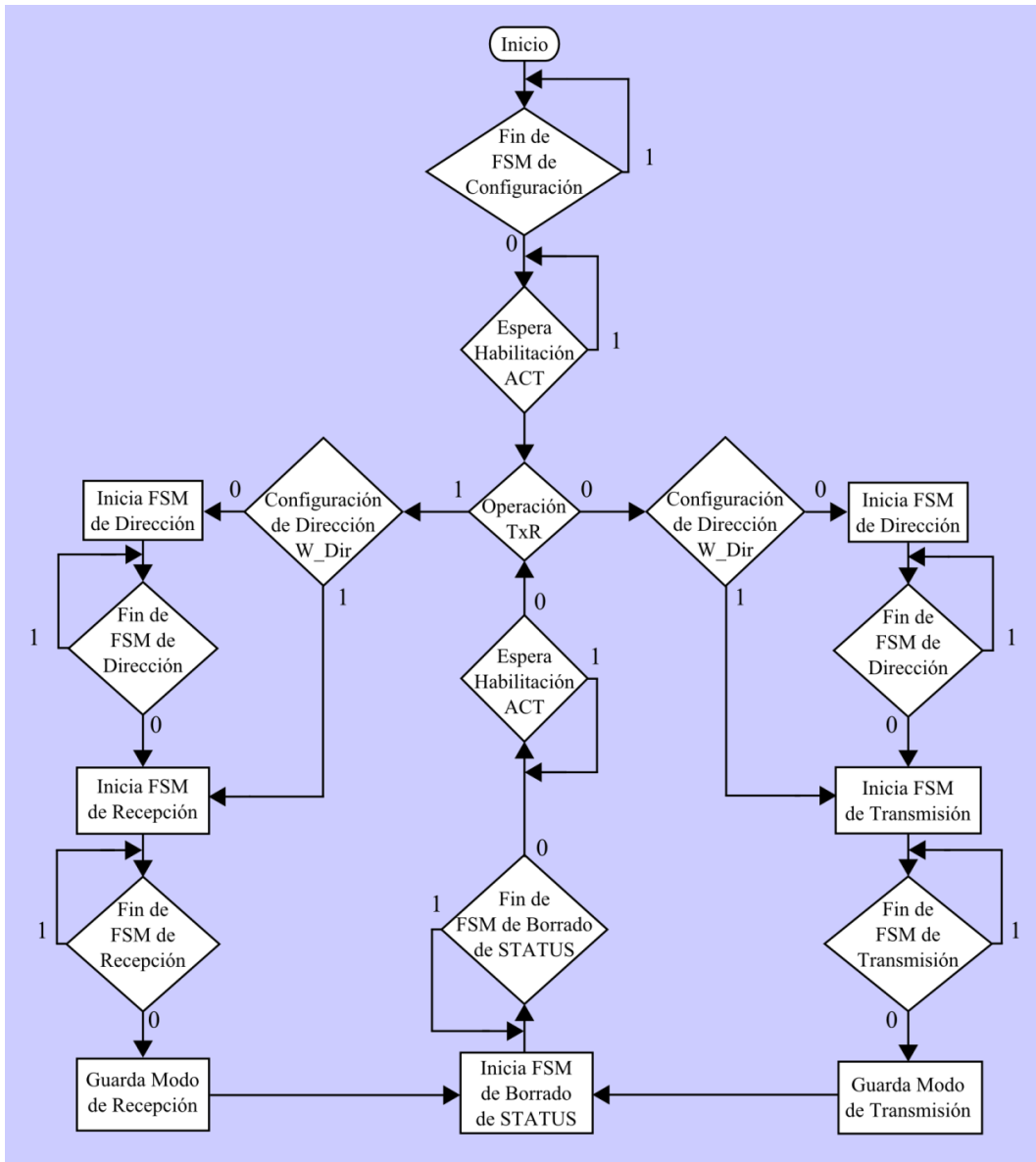


Figura 3.13. Diagrama de flujo de la máquina de estados principal.

Si el modo seleccionado es el de transmisor, la máquina de estados principal evalúa si se desea configurar la dirección del transceptor por medio de la señal W_Dir. En caso de que la señal se mantenga en bajo, la máquina principal activa la máquina de estados encargada de la configuración de la dirección del transceptor y espera a que se le devuelva

la señal de fin de máquina. Esta máquina envía las señales necesarias para lograr configurar el transceptor con la dirección deseada por el usuario. En caso de que la señal W_Dir esté en alto en el momento de su evaluación, no realiza la configuración de las direcciones.

Independientemente si se desea la configuración de dirección o no, se activar enseguida la máquina de estados encargada de realizar la transmisión, mientras la máquina principal espera a que esta le devuelva la respuesta de final de máquina. Cuando se recibe la señal de fin de la máquina encargada de enviar las señales para lograr la transmisión de datos, se guarda en el registro 2 del analizador el modo de transmisión almacenado en el registro 1 con anterioridad.

Una vez guardado el modo de transmisión, se inicia la máquina de estados encargada de borrar el registro STATUS del transceptor. Esta máquina de estados envía las señales necesarias para el borrado del registro STATUS. La máquina principal espera a recibir la señal de fin de la máquina de borrado. Después de recibir esta señal, la máquina principal vuelve al estado de espera de la activación por medio de la señal ACT y realizar una nueva operación.

Analizando el caso en el que se desea que el transceptor actúe como receptor. La máquina principal guarda el valor de TxR, el cual se encuentra en alto, en el registro 1 del analizador de modo anterior y posteriormente evalúa si se desea configurar la dirección del transceptor por medio de la señal W_Dir. Si esta señal se mantiene en bajo, la máquina principal activa la máquina de estados que realiza la configuración de la dirección y espera su señal de fin para proceder a la activación de la máquina de estados de recepción. En caso de que la señal W_Dir se mantenga en alto, la máquina principal irá directamente a la activación de la máquina de recepción sin activar la máquina de estados de la configuración de dirección. Después de que se activa la máquina de estados de la recepción, se envían las señales necesarias para recibir los datos inalámbricamente. La máquina principal se mantiene en espera de la señal de fin de recepción para guardar el modo de recepción en el registro 2 del analizador de modo anterior.

Al igual que con el modo transmisor, se activa la máquina de estados de borrado del registro de STATUS y se espera la recepción de su señal de fin para volver al estado de espera de activación para el envío o recepción de un nuevo dato, según se desee.

memoria ROM que se muestra en la sección 3.2.1.1. En caso de que aún no se envíe el dato del registro completo, correspondiente al comando y al valor del registro a modificar, la máquina de estados envía el valor guardado en la memoria ROM de acuerdo con la dirección del contador 1. Posteriormente, espera la señal de fin del módulo SPI, EOT. Una vez que se ha recibido esta señal, la máquina evalúa el fin del contador 1, con esto se analiza si se han enviado todas las tramas de configuración. En el caso de que no se hayan enviado todas las tramas almacenadas en la memoria ROM, se aumenta la dirección con la señal OPC0 del contador 1 y se vuelve a comenzar con la evaluación del fin de dato de cada valor de registro. Si el conjunto de tramas de comando y valor de registro han sido enviados, la máquina de estados evalúa directamente si el contador ha terminado con las direcciones. Si el contador ha terminado, la señal EOC0 le indica a la máquina de configuración y ésta envía la señal de fin a la máquina principal.

3.2.3.3 *FSM de configuración de dirección.*

Esta máquina de estados se encarga de realizar las operaciones necesarias para cambiar la dirección de transmisión y recepción establecida en el transceptor.

En la Figura 3.15 se presenta el diagrama de flujo de la máquina de estados de la configuración de dirección. Después de iniciar, la máquina de estados espera su activación por medio de la señal STDr. Si se recibe la señal de activación, la máquina envía el comando para modificar la dirección de transmisión del transceptor y posteriormente envía la primera trama de la dirección. Espera a que se reciba la señal de EOT proveniente del módulo SPI. Después, la máquina de estados evalúa la señal de fin del contador 2. En caso de que no se active, la máquina aumenta el contador por medio de la señal OPC4 y desplaza el registro RegDes40_8 mediante la señal R_Dir para enviar la siguiente trama de 8 *bits* de la dirección. El proceso se repite hasta que se hayan enviado los 5 *bytes* necesarios para la configuración de la dirección de transmisión. El contador 2 indica esto por medio de la señal EOC4. Una vez terminado, la máquina de estados comienza con el envío de la dirección de recepción del transceptor. Inicialmente se envía el comando de escritura para el registro de dirección del transceptor en modo de receptor y posteriormente, se envía la primera trama de la dirección. Al igual que con la dirección de transmisión, la máquina espera la señal EOT del módulo SPI. Una vez recibida la señal, se analiza el EOC del contador 2. Si no ha terminado de enviar todas las tramas de 8 *bits*, la máquina aumenta el

contador y recorre el registro RegDes40_8 para enviar la siguiente trama. Si terminó la cuenta, la máquina de estados envía la señal de fin a la máquina principal y vuelve a la espera de la señal de habilitación STDr.

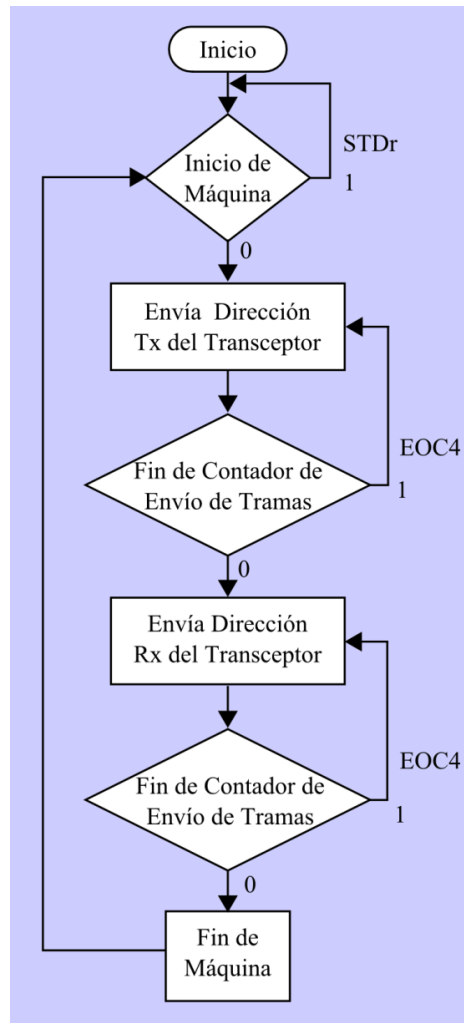


Figura 3.15. Diagrama de flujo de la máquina de estados de configuración de dirección del transceptor

3.2.3.4 FSM de transmisión de datos.

La máquina de transmisión de datos es la encargada de configurar al transceptor como transmisor y enviar los datos en *bytes* de acuerdo con el protocolo SPI para realizar la comunicación inalámbrica. El número de *bytes* enviados hacia el transceptor es configurado previamente y controlado por el contador 3 como se puede observar en la Figura 3.11 por

medio de la señal OPC2. En la Figura 3.16 se presenta el diagrama de flujo de la máquina de estados de la transmisión.

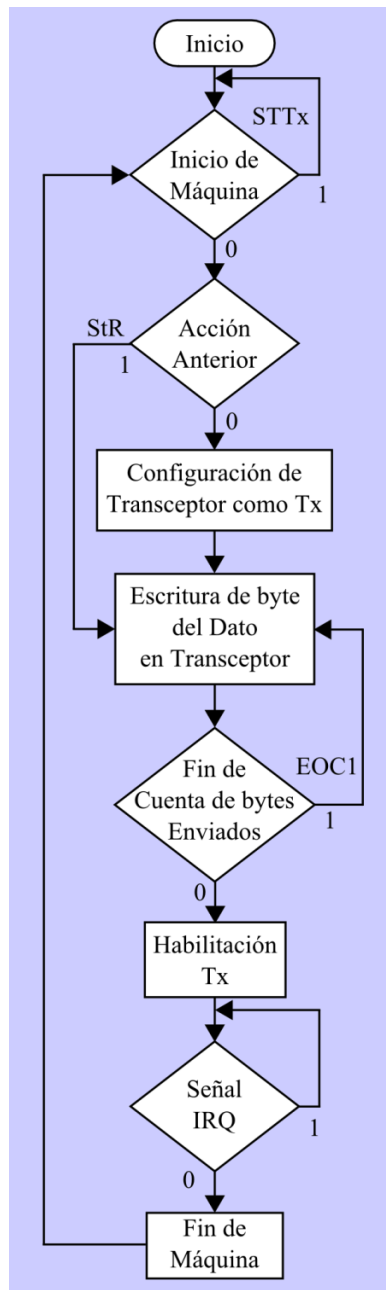


Figura 3.16. Diagrama de flujo de la máquina de estados de transmisión.

La máquina de estados de transmisión espera su activación por medio de la señal STTx enviada por la máquina de estados principal. Cuando la señal activa la máquina, ésta

evalúa la señal proveniente del analizador de modo anterior. Si el analizador indica que el transceptor se encontraba actuando como receptor, la máquina envía el comando y la instrucción para cambiarlo a modo de transmisor. Si el transceptor actuó como transmisor en la operación anterior, la máquina no realiza esta operación de cambio de modo.

Una vez configurado el transceptor como transmisor, la máquina de estados envía el comando adecuado para indicar que se realizará un envío inalámbricamente, seguido del primer *byte* con el dato a enviar. La máquina espera el fin del módulo SPI y evalúa si es el último *byte* del dato a enviar inalámbricamente. Si se requieren enviar más *bytes*, se aumenta el contador 3 y se recorre el registro de desplazamiento RegDes8n_8 con la señal R_Tx. Al recorrer el registro, se envía al transceptor el siguiente *byte* y así sucesivamente hasta que el contador devuelva la señal de fin de cuenta EOC1. Cuando se recibe la señal, la máquina de estados indica al transceptor que se envíen todos los *bytes*, almacenados previamente en el transceptor, por medio de la habilitación de la transmisión. Después de esto, la máquina de estados espera la señal IRQ proveniente del transceptor, la cual, indica que el envío inalámbrico de los *bytes* escritos en el transceptor se ha realizado correctamente. Una vez recibida la señal IRQ, la máquina de estados de transmisión envía su señal de fin a la máquina principal y vuelve a su estado de espera de activación.

3.2.3.5 FSM de recepción de datos.

Esta máquina de estados se encarga de configurar el transceptor en modo de receptor, leer el dato recibido cuando el transceptor lo indica y guardarlo en un registro para su posterior lectura por el sistema. En la Figura 3.17 se muestra el diagrama de flujo de la máquina de estados de recepción. Como se puede observar, la máquina de estados de recepción espera a que la máquina principal le envíe la señal de inicio STRx. Posteriormente, se analiza la señal StR para evaluar si el transceptor operaba en modo de receptor. En el caso de que se operara en modo de transmisión, la máquina de estados envía la trama de comando que indica al transceptor la escritura en su registro de configuración, la máquina espera que el módulo SPI le envíe la señal EOT para indicar que la trama del comando se ha enviado correctamente.

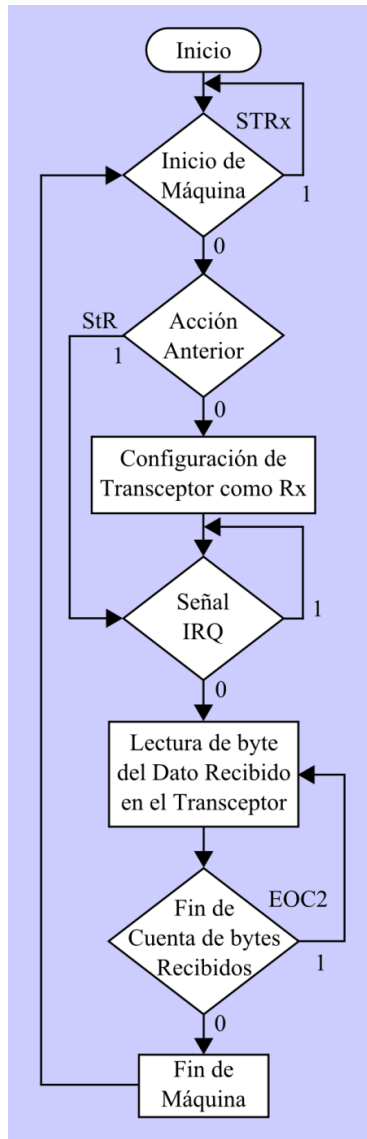


Figura 3.17. Diagrama de flujo de la máquina de estados de recepción.

Después de que se envía el *byte* de comando, es enviado el valor correspondiente para que el transceptor opere en modo de recepción. En el caso de que el transceptor ya se encuentre operando como receptor, la máquina de estados hace caso omiso del procedimiento de reconfiguración y va al estado de espera. En este estado de espera, la máquina se mantiene hasta que reciba el bajo de la señal IRQ, con lo que el transceptor indica que existe un dato recibido inalámbricamente en su pila y que se encuentra listo para que sea leído. Cuando se recibe la activación de la señal IRQ, se envía al transceptor el comando para leer los *bytes* del dato recibido. Enseguida, se envía al transceptor la primera trama con un valor al azar

para leer el primer *byte* del dato recibido, la máquina de estados evalúa la cantidad de *bytes* recibidos, si aún existen *bytes* por leer del transceptor, aumenta el contador 2 con la señal OPC2 y realiza el corrimiento del registro RegDes8_8n para leer el siguiente *byte* del dato, este proceso se repite hasta leer todos los *bytes* del dato recibido inalámbricamente. Después de que el contador 2 envía la señal EOC2 activada, la máquina de estados de recepción envía la señal de final a la máquina de estados principal y vuelve a su estado de espera de la señal de activación.

3.2.3.6 FSM de borrado del registro STATUS.

El registro STATUS del transceptor es el encargado de indicar el modo de operación y el valor de las banderas con las que cuenta. Estas banderas, una vez activadas, se mantienen en estado bajo hasta que sea reiniciado el registro. Entre estas banderas se encuentra la señal IRQ, por lo que después de realizar cada envío o recepción de datos inalámbricamente, es importante borrar el registro de STATUS mediante el envío del comando que indica su escritura y posteriormente, el valor que permite restablecer la señal IRQ.

En la Figura 3.18 se muestra el diagrama de flujo de la máquina de borrado del registro STATUS.

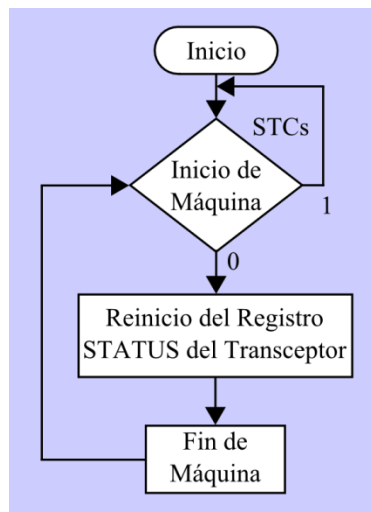


Figura 3.18. Diagrama de flujo de la máquina de estados del borrado de STATUS.

La máquina se mantiene en espera de la señal STCs proveniente de la máquina de estados principal. Cuando se recibe la señal baja, se envía el comando de escritura junto con el dato que reinicia el registro STATUS y envía el fin de máquina. Después de que se envía el fin, la máquina regresa a su estado de espera de la señal STCs.

3.3 Sistema esclavo.

El sistema esclavo es el sistema encargado de realizar la petición de datos de las señales provenientes de los sensores de aceleración y de velocidad angular que se encuentran en una unidad de medición inercial, de acuerdo con la señal de petición enviada inalámbricamente del sistema maestro. Para posteriormente, agrupar en una trama los datos provenientes de la unidad y enviarla inalámbricamente en una única comunicación.

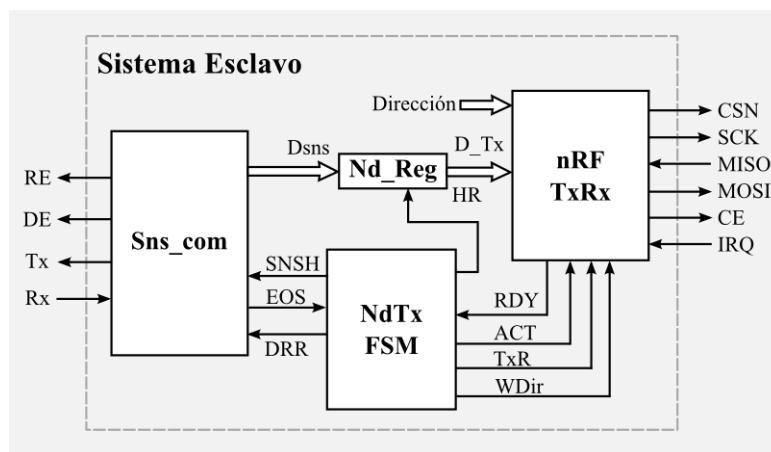


Figura 3.19. Diagrama a bloques del sistema esclavo.

En la Figura 3.19 se muestra el diagrama a bloques de la conformación del sistema esclavo. La cual corresponde de tres bloques llamados nRF_TxRx, Sns_com, Nd_Reg y Nd_TxFSM. El bloque nRF_TxRx es el encargado de realizar la interfaz necesaria con el transceptor para la comunicación inalámbrica. El bloque Sns_com se encarga de la interfaz entre la unidad de medición inercial y el sistema esclavo para la obtención de las señales de aceleración y velocidad angular. El bloque Nd_TxFSM se encarga de controlar las activaciones de los bloques Sns_com, nRF_TxRx y Nd_Reg. Siendo este último utilizado para el almacenamiento de los datos provenientes del bloque Sns_com y sincronizarlos para

el envío inalámbrico con el módulo nRF_TxRx. Como se observa, a este bloque se le agrega la dirección del dispositivo de una manera independiente, esto es para colocar múltiples sistemas esclavos con direcciones diferentes.

3.3.1 Módulo Sns_com.

Como ya se mencionó con anterioridad, el módulo Sns_com es el encargado de realizar la interfaz entre la unidad inercial y el sistema esclavo. En la Figura 3.20 se muestran las entradas y salidas del módulo Sns_com. Como se puede observar, el bloque cuenta con las señales de control SNSH, EOS y DDR. Las cuales son las encargadas de la habilitación del bloque, la señal que indica el fin de la operación y la señal que indica al bloque que la lectura ha sido realizada, respectivamente. De igual manera, cuenta con la señal que contiene los datos de las señales inerciales provenientes de los sensores de aceleración y velocidad angular Dsns. Las señales DE y RE son de control para la unidad de medición inercial, es decir, para indicarle cuando se le realiza una lectura de datos y una escritura. La señal Tx es utilizada para realizar la transmisión de datos hacia la unidad inercial de una manera serial y por lo contrario, la señal Rx es utilizada para realizar la recepción.

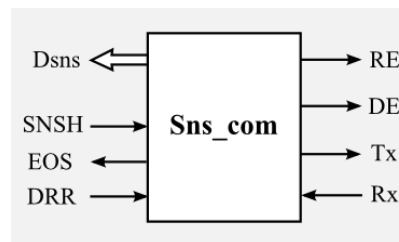


Figura 3.20. Módulo de comunicación con la unidad de medición inercial.

El módulo de comunicación con la unidad inercial se encuentra estructurado como se muestra en la Figura 3.21. El módulo Sns_FSM es el encargado de realizar el control sobre los demás bloques. Tales como Sns_Rx, Sns_Tx, Sns_Ret, Sns_Conta y Sns_Desplaza. Además de proporcionar la señal DR para indicarle a la unidad de medición inercial si se desea realizar una lectura o escritura. El módulo Sns_Rx se encarga de recibir los datos provenientes de la unidad inercial mediante la señal Rx y agruparlos para obtener una señal de 8 *bits* que, posteriormente, será concatenada con los demás *bytes* recibidos en

el registro de desplazamiento Sns_Desplaza. El módulo Sns_Rx es controlado mediante la señal de inicio RDD y la señal que indica el fin de la operación EOR. La operación de este módulo corresponde al módulo de recepción del protocolo RS-232 descrito en este trabajo. El registro de desplazamiento es controlado por la señal HRE de la máquina de estados y proporciona la señal Dsns que contiene todos los *bytes* leídos de la unidad inercial.

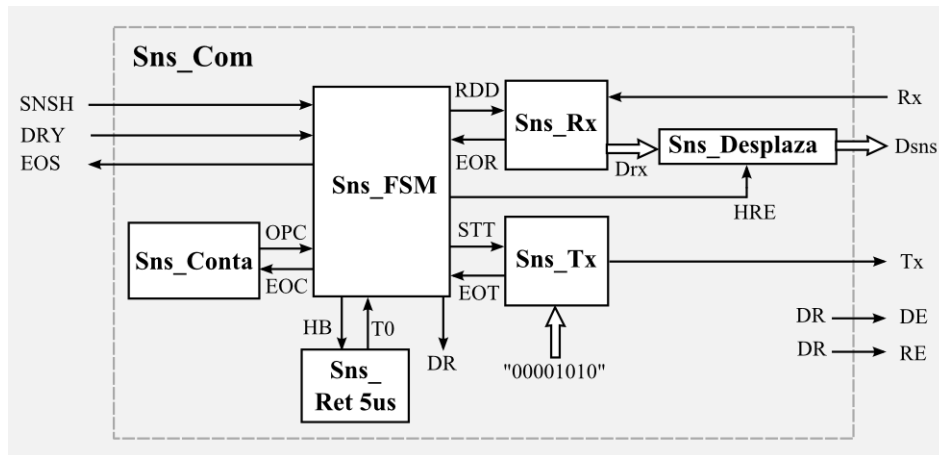


Figura 3.21. Estructura del módulo Sns_com.

El módulo Sns_Tx es el encargado de realizar el envío del dato que le indica a la unidad de medición inercial el deseo de la lectura de señales inerciales. Este envío se realiza de manera serial. El valor enviado hacia la unidad de medición es el “00001010” con lo cual se indica que se desea una lectura de señales provenientes del sensor de aceleración y de velocidad angular. La operación del módulo Sns_Tx corresponde al módulo de transmisión del protocolo RS-232 descrito en la sección 3.5 en este trabajo. Las señales que controlan este bloque son la de inicio STT y EOT, la cual indica el fin de operación del bloque.

Además de estos bloques, se tiene cuenta con el módulo Sns_Conta que realiza el conteo de las tramas recibidas de la unidad de medición inercial. Siendo tres señales recibidas por cada medida inercial y de dos tramas de ocho *bits* cada una, se tiene un total de doce lecturas cada vez que se realice una petición de datos a la unidad de medición inercial. Las señales que controlan este módulo son la señal OPC, con la cual se le indica si se desea incrementar la cuenta, mantener o reiniciar; y la señal EOC, que indica que la cuenta de las tramas recibidas ha finalizado.

Para que la unidad de medición inercial realice una correcta lectura, es necesario que entre la operación de escritura de dato y lectura exista un pequeño retardo. Para esto se cuenta con un módulo que realiza un retardo de 5 μ s activado mediante la señal HB. Cuando el retardo ha pasado, la señal T0 se lo indica a la máquina de estados.

El diagrama de flujo de las operaciones que realiza la máquina de estados se muestra en la Figura 3.22.

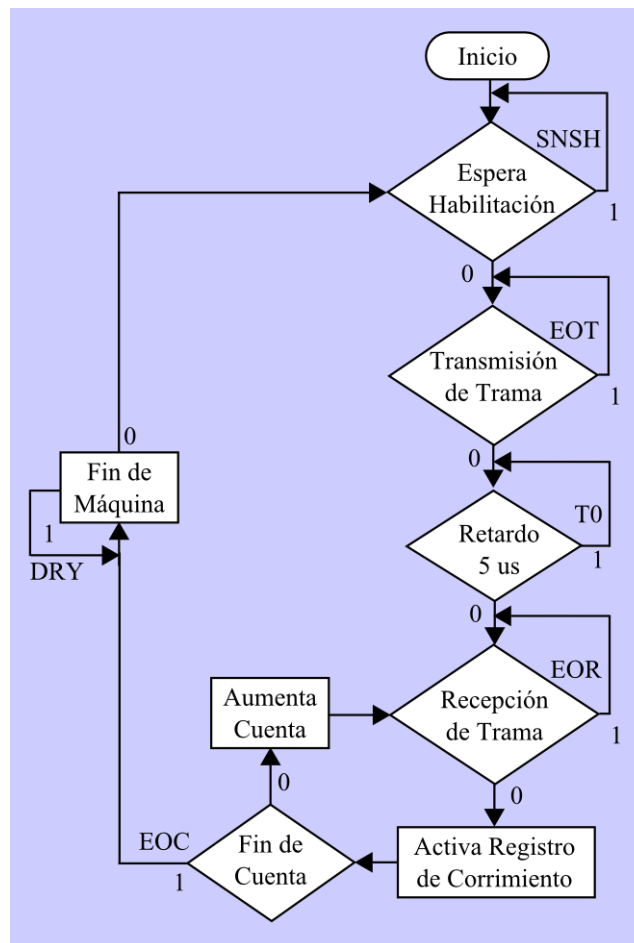


Figura 3.22. Diagrama de flujo del módulo Sns_com.

Como se ve en la Figura 3.22, después del inicio de la máquina, ésta se mantiene en espera de la señal de activación SNSH. Una vez que se ha habilitado la máquina, se habilita el envío de la trama “00001010”, la cual le indica a la unidad de medición inercial que se desea la lectura de las señales. Después de que ha terminado de enviar la trama, se activa el retardo de 5 μ s. Una vez que ha pasado el retardo, se inicia la recepción de las señales

provenientes de la unidad de medición inercial. Se activa el módulo Sns_Rx y se espera la recepción de la señal EOR que indica que se recibió la trama completa de 8 *bits*. Posteriormente, se activa el registro de corrimiento Sns_Desplaza para concatenar la trama recibida. Después de esto se analiza la cuenta de la trama concatenada, si aún se han recibido todas las tramas de la lectura de datos, se aumenta la cuenta y se recibe el siguiente *byte* correspondiente. Este procesamiento se repite hasta que se obtengan todos los *bytes* de la lectura de la unidad de medición inercial. Cuando esto ocurre, la máquina envía la señal de fin de operación EOS y espera la señal DRY, que indica que el dato ya fue leído por módulos posteriores, para volver a esperar la señal de habilitación. Esto es realizado para sincronizar el envío de datos. El módulo Sns_Rx corresponde al módulo de recepción mostrado en la sección 3.5 de este trabajo.

3.3.2 Máquina de estados NdTx_FSM.

Como se puede ver en la Figura 3.19, el módulo de la máquina de estados Nd_TxFSM es el encargado de manejar las operaciones entre los demás módulos. Para manejar el módulo de comunicación inalámbrico se utilizan las señales TxR, ACT, WDir y RDY, utilizadas para indicarle al transceptor la operación deseada, para su activación, para indicarle si se desea escribir su dirección y para que se indique cuando se haya terminado de realizar una operación, respectivamente.

El diagrama de flujo de los pasos que realiza el módulo de la máquina de estados se presenta en la Figura 3.23. Como se puede observar, una vez iniciado el diagrama de flujo, el módulo espera la señal RDY que indica que el transceptor ha sido configurado para operar y que se encuentra listo. Después de que se ha configurado, se activan las señales necesarias para que el transceptor se mantenga en modo de receptor, con el fin de que el sistema esclavo se mantenga en modo de espera hasta que el sistema maestro envíe un dato. Este envío de dato es la petición de la adquisición de señales de la unidad de medición inercial. El sistema se mantiene en la espera de la señal RDY, la cual indica que se ha recibido un dato del sistema maestro. Después de esto, la máquina de estado activa el módulo de comunicación con la unidad de mediciones inerciales y espera la activación de señal EOS, indicándole a la máquina que se tiene una lectura completa recibida. Posteriormente se activa el registro Nd_Reg para guardar el dato y sincronizarlo para su envío inalámbrico. Después que se ha hecho esto, se envía la señal DRY al módulo

Sns_com, indicándole que se ha leído el dato en su salida. Inmediatamente después de esto, se activan las opciones para que el transceptor trabaje en modo de transmisor. Se mantiene en espera después, la señal RDY, la cual indica que el dato se ha enviado correctamente. La máquina indica que ha terminado la operación y vuelve a configurarse el transceptor como receptor y se mantiene en espera de la siguiente indicación del sistema maestro para realizar una nueva lectura de señales inerciales.

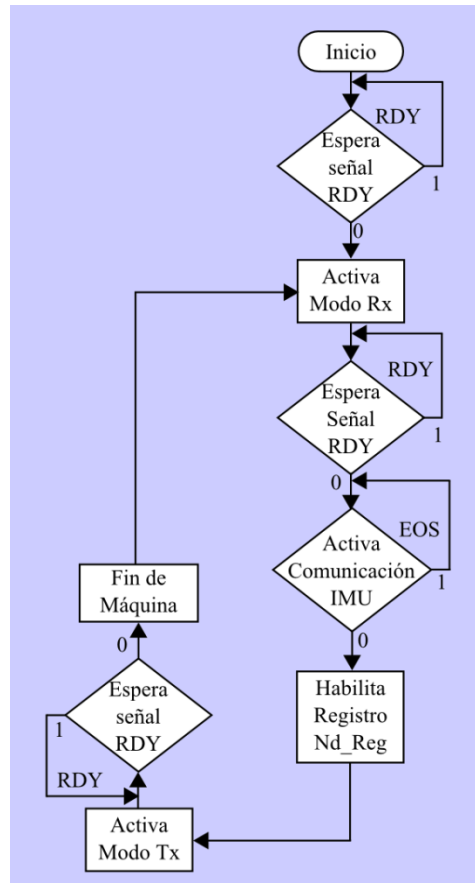


Figura 3.23. Diagrama de flujo del funcionamiento del sistema esclavo.

3.4 Sistema maestro.

El sistema inalámbrico maestro es el encargado de enviar una señal al sistema esclavo del que se desea obtener las señales inerciales. Después de recibirlas, se realiza el filtrado de diezmo por promedio y posteriormente un filtro digital pasa bajas para obtener las señales sin ruido. Para estimar las señales inerciales de vibración se utiliza un filtro

digital pasa altas después del filtro de diezmando. El bloque del sistema maestro se muestra en la Figura 3.24.

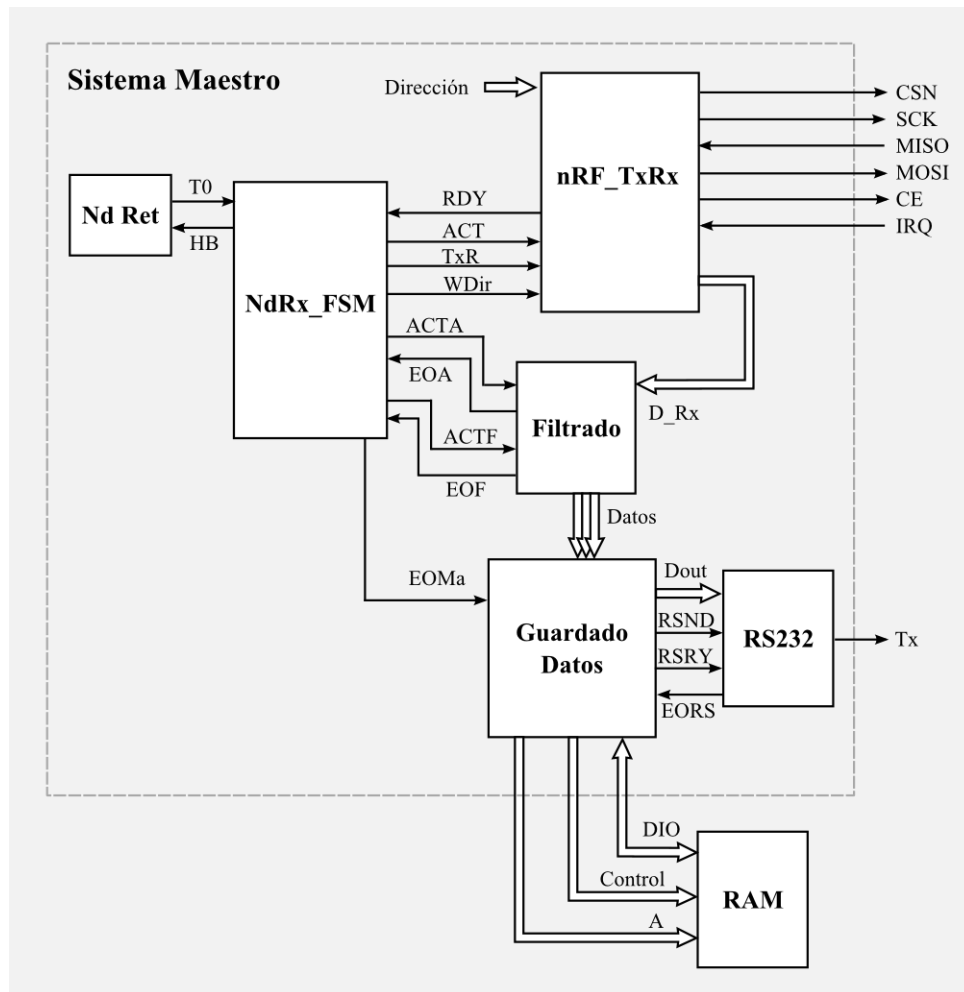


Figura 3.24. Diagrama a bloques del sistema inalámbrico maestro.

Como se puede ver en la Figura 3.24, se presentan los bloques que conforman el módulo del sistema maestro. El bloque NdRx_FSM es el encargado de realizar el control de los demás módulos. El bloque nRF_TxRx es el encargado de realizar la interfaz entre el transceptor y el sistema. Se cuenta con el módulo de filtrado, es en este dónde se realiza la separación de los seis datos obtenidos del acelerómetro y giróscopo recibidos inalámbricamente. A cada uno de estos datos se les aplican los diferentes filtros desarrollados y los resultados son enviados hacia el módulo de guardado de datos. Este módulo es el encargado de separar y controlar el almacenamiento en una memoria RAM de

todos los datos filtrados y en bruto. La señal nombrada control corresponde al conjunto de señales con las que se controla la escritura y lectura de los datos de memoria. El módulo de guardado de datos controla la lectura del total de datos almacenados y su envío por medio del protocolo RS-232 hacia la computadora. Esta interfaz se realiza por medio del módulo que lleva el nombre RS-232.

3.4.1 Máquina de estados del sistema esclavo.

La máquina de estados del sistema maestro se encarga de controlar los demás módulos para realizar el procesamiento de los datos recibidos. En la Figura 3.25 se muestra el diagrama de flujo de la máquina de estados del sistema maestro.

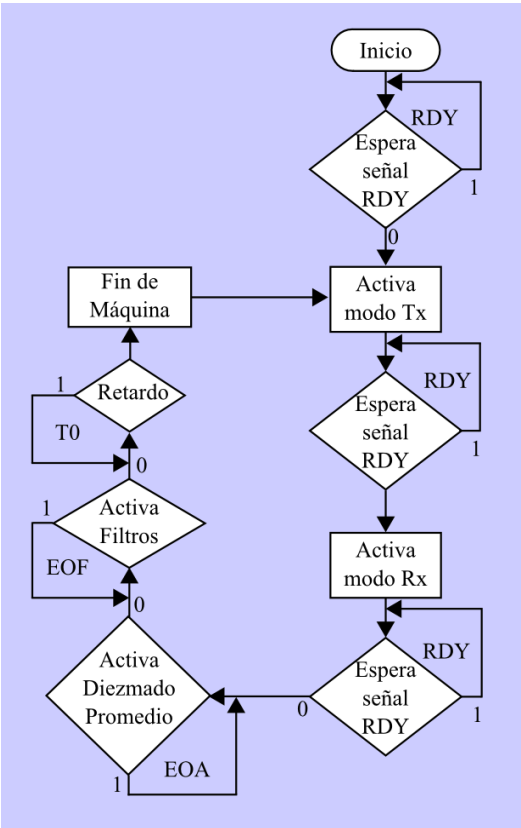


Figura 3.25. Diagrama de flujo del funcionamiento del sistema maestro.

Como se puede observar en la Figura 3.25, la máquina de estados inicia y se mantiene en espera de la señal RDY, la cual le indica que el transceptor ha sido configurado para realizar la comunicación inalámbrica. Una vez recibida esta señal, se activa el transceptor como transmisor y se realiza el envío de un dato con un valor de todos los bits en '1'. Con esto se le realiza la petición al sistema esclavo de una lectura de datos.

Después de esto, la máquina espera la recepción de la señal RDY nuevamente, esta indica que se ha enviado la petición correctamente. Posteriormente, el transceptor es activado en modo receptor para adquirir la trama con los datos enviados por el sistema esclavo. Al igual que con la transmisión, la máquina de estados espera la señal RDY indicando que se ha terminado de recibir la trama con los datos. Una vez terminado esto, se activa el bloque que realiza el diezmado por promedio. Cuando se obtiene la señal que indica que el diezmado se ha realizado, se activan los demás filtros, esperando a recibir la señal que indica su final. Después de recibida la señal, se envía la señal de fin de máquina EOMa hacia el módulo de guardado de datos para que realice almacene todos los datos, tanto recibidos inalámbricamente como los filtrados.

3.4.2 Filtrado.

El módulo de filtrado realiza las operaciones necesarias para el filtrado de diezmado por promedio, el filtrado digital pasa-bajas y el filtrado digital pasa-altas. En la Figura 3.26 se muestra el diagrama a bloques de la constitución del bloque de filtros.

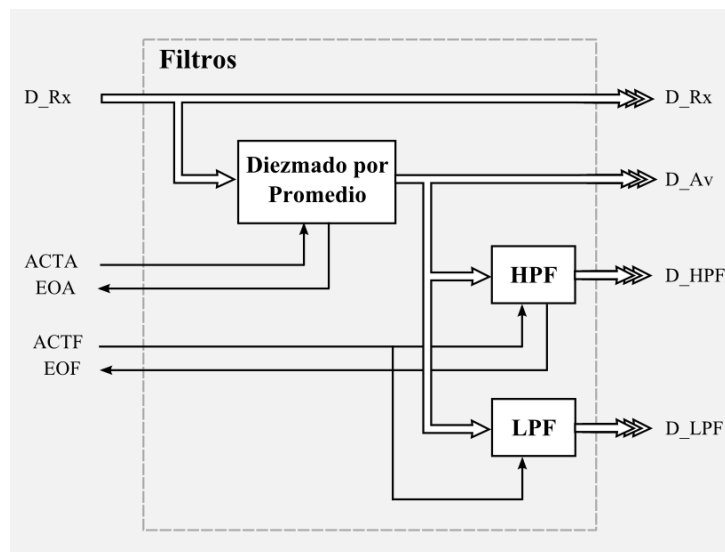


Figura 3.26. Diagrama a bloques de filtros.

3.4.2.1 Diezmado por promedio.

Para desarrollar el diezmado se utilizó el modelo mostrado en la Figura 3.27, el cual consta de una memoria RAM interna para realizar el almacenamiento de los datos; dos contadores de dirección, el de entrada utilizado para escribir el dato nuevo en la memoria y

el de salida para enviar los valores guardados en memoria y obtener la sumatoria de estos datos; el acumulador se encarga de realizar esta sumatoria para posteriormente utilizar un corrimiento a manera de divisor para obtener el promedio; y por último la máquina de estados, la cual se encarga de activar los contadores de dirección, la memoria y el acumulador de acuerdo con el procesamiento.

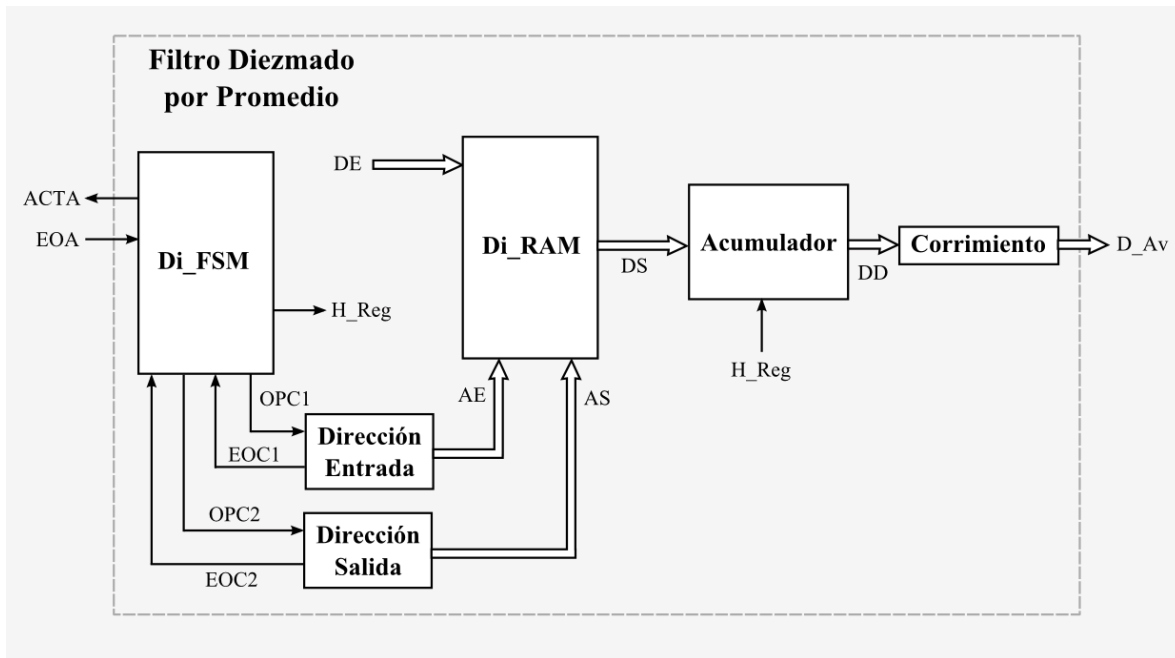


Figura 3.27. Diagrama a bloques del filtro de diezmo por promedio.

3.4.2.2 Filtro digital FIR.

El filtro digital FIR cuenta con los bloques mostrados en la Figura 3.28. Los cuales, al igual que el filtro de diezmo, cuentan con una memoria RAM interna, un contador de dirección de entrada y un contador de dirección de salida y una máquina de estados, pero existen algunos cambios en lugar de un acumulador solamente, este filtro cuenta con un módulo multiplicador acumulador y no tiene un registro de corrimiento. Además se le integra un contador de dirección para una memoria ROM interna, en la cual, se tienen los valores de las constantes b para este filtro. Las constantes b son multiplicadas con los valores de los datos guardados en la RAM para realizar el procesamiento de filtrado. De acuerdo con estas constantes los filtros serán pasa-bajas o pasa-altas.

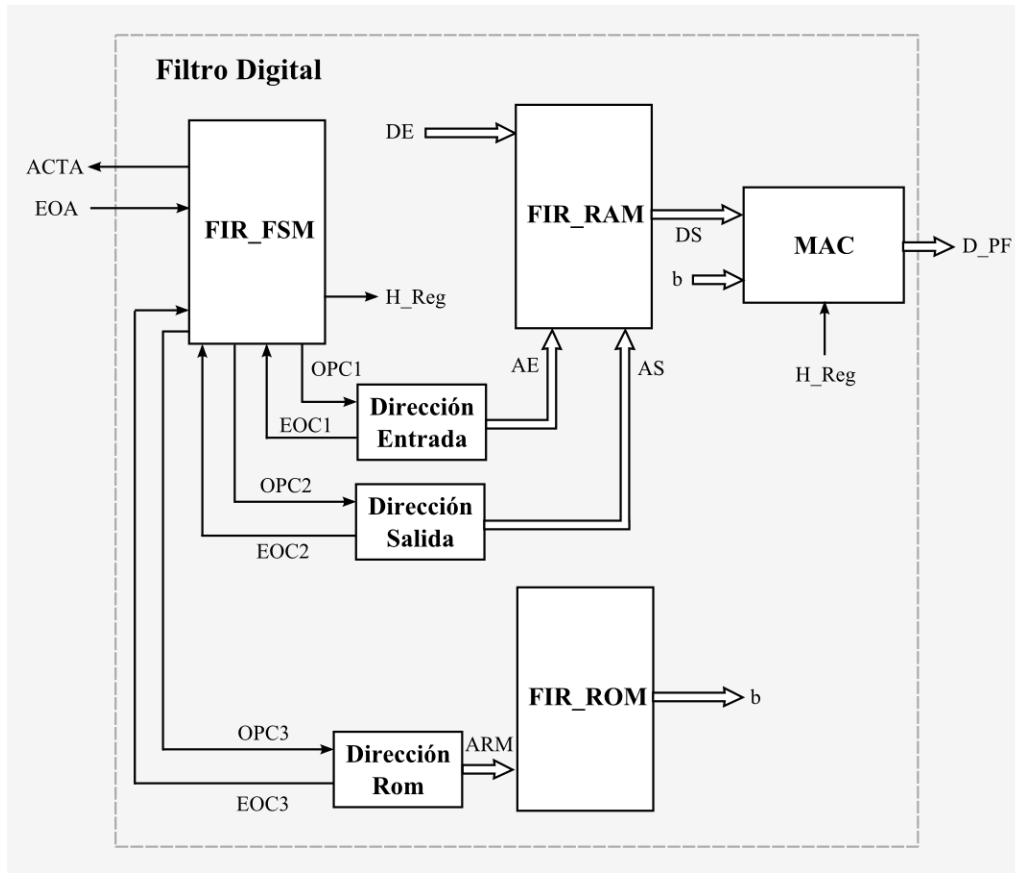


Figura 3.28. Diagrama a bloques del filtro FIR.

3.4.2.3 Máquina de estados de los filtros.

La máquina de estados de los filtros tanto de diezmado por promedio, como para el filtro FIR, realizan el mismo procesamiento. Solamente varía en la activación de la dirección ROM que es añadida. En todo lo demás se presenta como se puede ver en la Figura 3.29. Inicialmente se reinicia la memoria RAM con ceros para evitar errores o adquisiciones de datos erróneos. Después de esto se espera su activación, posteriormente, se guarda el dato recibido en la memoria RAM. Una vez que se tiene el dato en la memoria RAM, se activa el acumulador y el contador de la dirección de salida, por lo que se realizan las iteraciones necesarias para sumar todos los datos que se encuentran en la memoria. En el caso del filtro FIR, se realiza lo mismo, pero se activan los contadores de la dirección de salida de la RAM y de la ROM para realizar la multiplicación de los valores y posteriormente la acumulación. Después de que se obtienen todas las acumulaciones de los datos guardados en la RAM, se evalúa si se ha terminado de escribir en todas las

direcciones de la memoria. En caso de que aún no se termine, se aumenta la dirección y se envía un fin de máquina para indicar que se ha terminado el proceso. En caso de que haya terminado, se reinicia la dirección de entrada de la RAM y se envía la misma señal de fin de máquina. La máquina vuelve a mantenerse en espera de la activación de la siguiente señal.

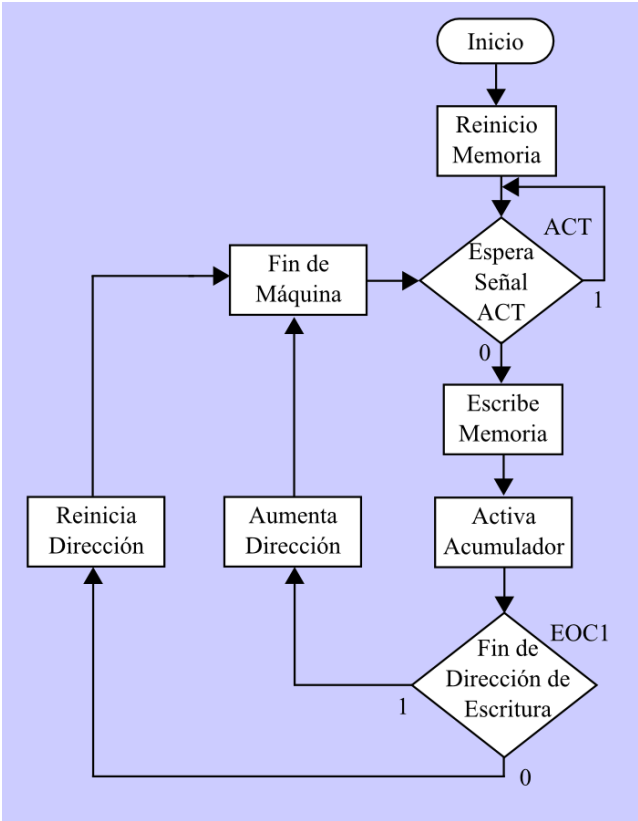


Figura 3.29. Diagrama de flujo de la máquina de estados de los filtros.

3.4.3 Guardado de datos.

El bloque de guardado de datos es el encargado de realizar la separación de datos procesados y en bruto, además de guardarlos en la memoria RAM. En la Figura 3.30 se muestra como se conforma el bloque de guardado de datos. Se cuenta con el manejador de la memoria RAM externa, Memo_Drv, el cual es controlado por la máquina de estados Dato_FSM. Además de controlar el manejador, la máquina de estados controla el contador de la dirección de la memoria RAM, CNA. También controla el contador de tramas recibidas de los datos CNT y el registro de cada uno de los datos provenientes del filtro.

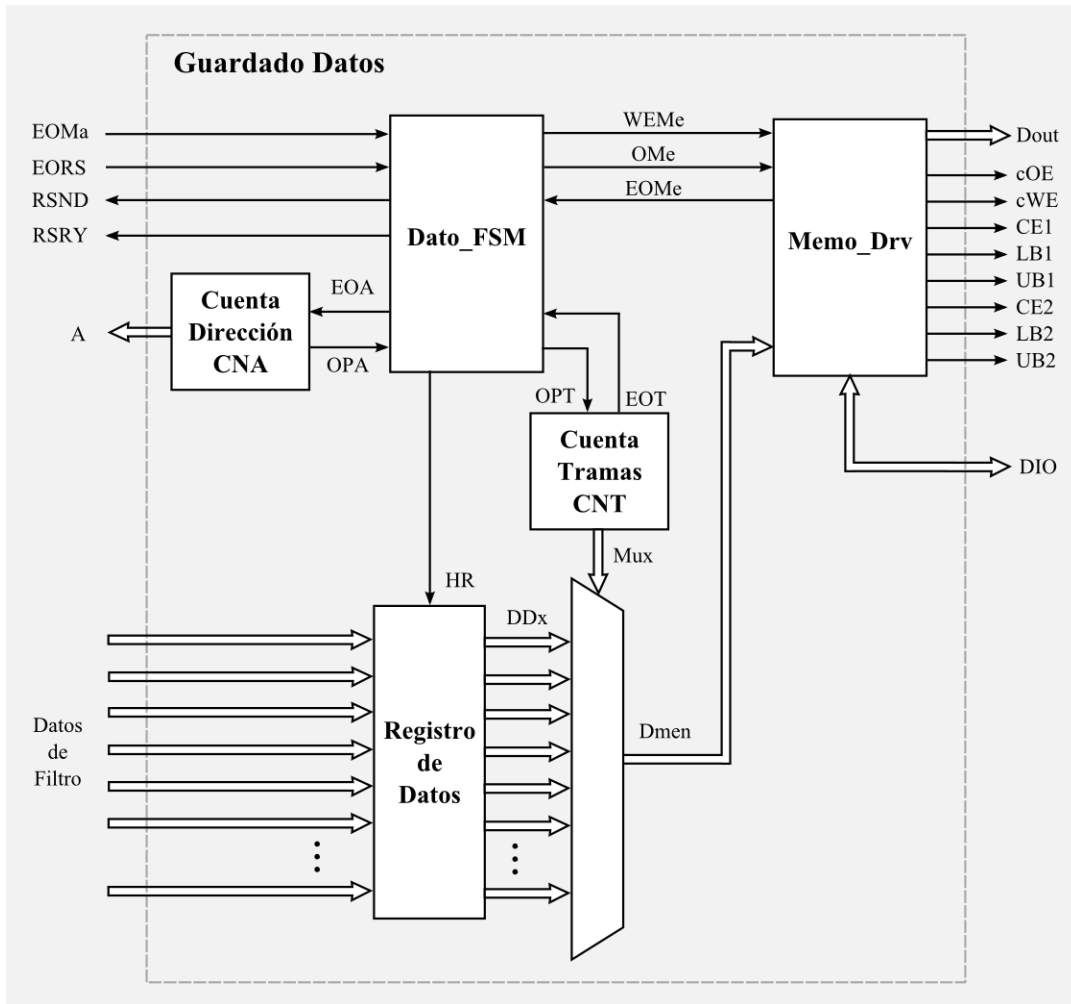


Figura 3.30. Diagrama a bloques del guardado de datos.

Para comprender el funcionamiento del bloque es necesario analizar el diagrama de flujo de la máquina de estados. Con el fin de una mejor comprensión del procesamiento realizado, la máquina de estados se dividió en dos partes: la parte de guardado de datos y la parte de lectura de datos.

La sección de guardado de datos se muestra en la Figura 3.31, en la cual se puede observar que después del inicio, la máquina de estados se mantiene en espera de la señal EOMa. Una vez recibida, se cargan los registros con los datos provenientes del bloque filtro. Todos los registros son activados por medio de la señal HR. Por medio del contador CNA se va seleccionando la dirección de la memoria en la que se ha de guardar el dato que se selecciona con el multiplexor y el contador CNT. Se activa el módulo Memo_Drv, el

cual realiza el protocolo para guardar el dato en la memoria RAM. Con la señal OMe se selecciona si se desea escribir o leer de la memoria.

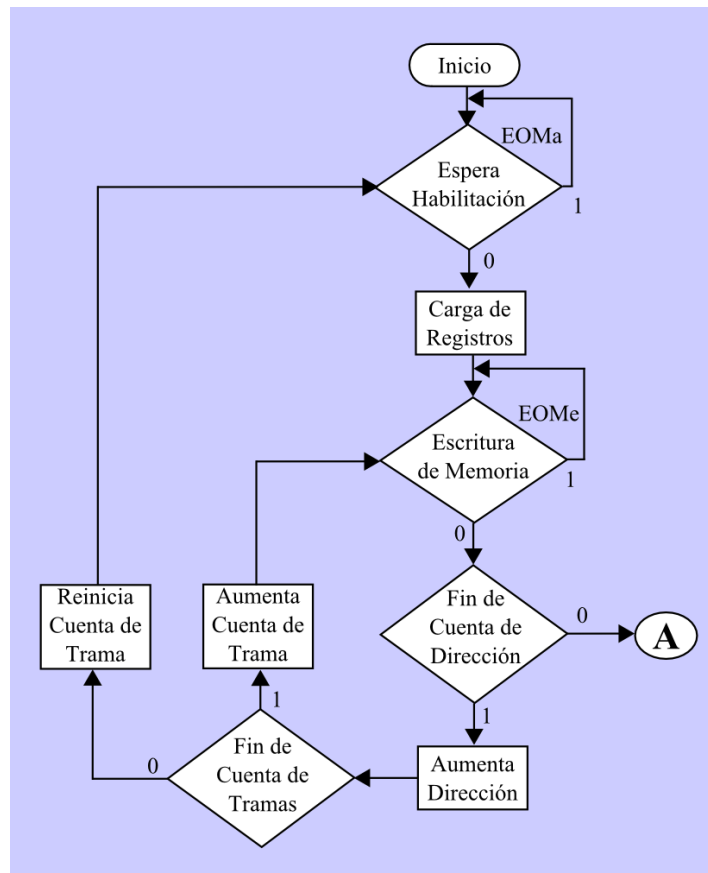


Figura 3.31. Sección de escritura en la máquina de estados de guardado de datos.

Una vez que se ha recibido la señal de fin de escritura de la memoria, se analizan el número de datos guardados por medio de la cuenta de dirección CNA, en caso de que aún no haya terminado la cuenta, se evalúa si todos los datos enviados por el bloque de filtrado han sido guardados. En caso contrario, la máquina de estados va a la sección de lectura de datos para enviarlos hacia la computadora por medio del protocolo RS-232. Si aún no se han guardado todos los datos de filtrado, la máquina de estados aumenta el contador CNT y vuelve a escribir en una diferente dirección de la RAM. En caso de que haya guardado todos, el contador se reinicia y espera la nueva activación del bloque.

Cuando el almacenamiento de datos ha terminado, la máquina de estados cambia a la parte de la lectura de datos para enviarlos por medio del protocolo RS-232 a la computadora. En la Figura 3.32 se presenta esta sección.

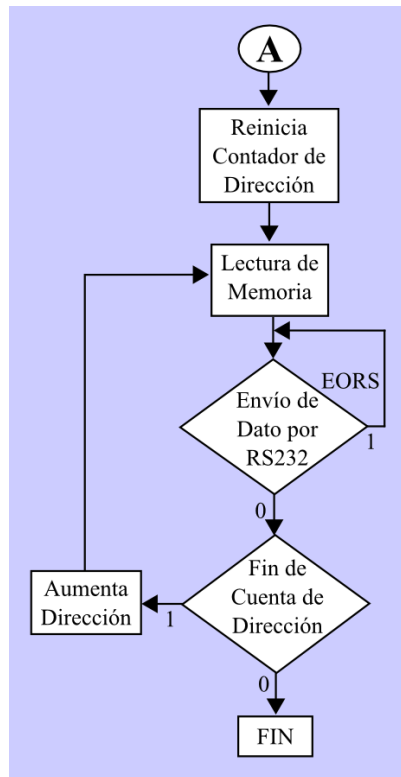


Figura 3.32. Sección de lectura y envío en la máquina de estados de guardado de datos.

Como se puede observar, después de que se han guardado todos los datos deseados en la memoria RAM, la máquina de estados entra en la sección del envío de estos datos. Para esto, se reinicia el contador de Dirección y se le indica al módulo Memo_Drv que se desea una lectura de la memoria, una vez recibido el dato leído en la dirección inicial, la máquina de estados activa el envío de datos de una manera serial por el protocolo RS-232. Después que se ha enviado el dato, la máquina de estados evalúa si la dirección corresponde al del último dato en memoria. En caso de que no sea el último dato, la máquina de estados aumenta la dirección de la memoria y continúa con la siguiente lectura del dato correspondiente. En caso contrario, la máquina envía una señal de fin.

3.4.3.1 Bloque Memo_Drv

El bloque manejador de memoria Memo_Drv es el encargado de enviar las señales correspondientes hacia la memoria RAM externa para realizar una lectura o escritura de acuerdo con lo que le indique la máquina de estados del guardado de datos. En la Figura 3.33 se muestra la estructura interna del bloque manejador.

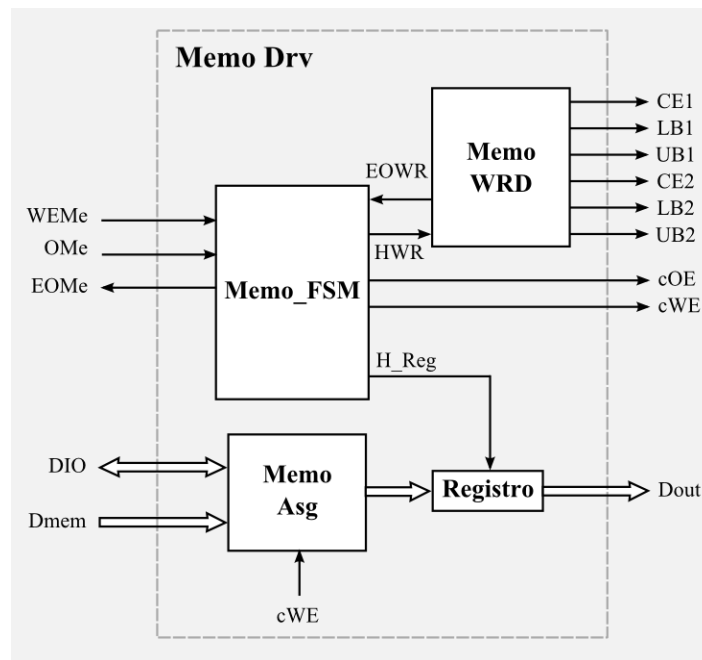


Figura 3.33. Diagrama a bloques del módulo Memo_Drv.

Como se puede ver, se cuenta con la máquina de estados memo_FSM, encargada de controlar los bloques para el procesamiento y generar las señales para la escritura y lectura de la memoria RAM externa. El bloque Memo_Drv cuenta con el cambio necesario de las diferentes señales para realizar la habilitación y deshabilitar la memoria. Es decir, tiene el procedimiento base para las comunicarse con la memoria. Este bloque cuenta con una señal de habilitación HWR y una señal que le indica a la máquina de estados memo_FSM cuando ha terminado. El bloque MemoAsg es utilizado para colocar en alta impedancia o escribir los datos en el bus de la memoria. La operación de este seleccionador es controlada por medio de la señal cWE correspondiente a la señal *Write Enable* de la memoria. Cuando se desee escribir, el bus DIO tomará el valor Dmem que es el dato proveniente de los filtros. Cuando se desee leer, el bus estará en alta impedancia y el dato obtenido se almacenará en

un registro habilitado por la máquina de estados para su posterior envío por el protocolo RS-232.

La máquina de estados del procesamiento para realizar la comunicación con la memoria RAM externa se muestra en la Figura 3.34. Después de iniciar, la máquina de estados espera su habilitación por medio de la señal HMe. Una vez habilitado, se evalúa la opción que el usuario desea realizar por medio de la señal OMe. En caso de que esta señal se haya activado en bajo, la máquina de estados Memo_FSM envía las señales correspondientes cWE y cOE para realizar la escritura de datos en la memoria. En caso de que la señal OMe sea activada en alto, la máquina de estados envía las señales correspondientes para realizar la lectura de datos. En ambas operaciones se mantiene hasta recibir la señal EOWR proveniente del módulo MemoWRD. Después de realizar cualquiera de las operaciones, la máquina de estados indica su fin de operación y vuelve al estado de espera de la señal de habilitación.

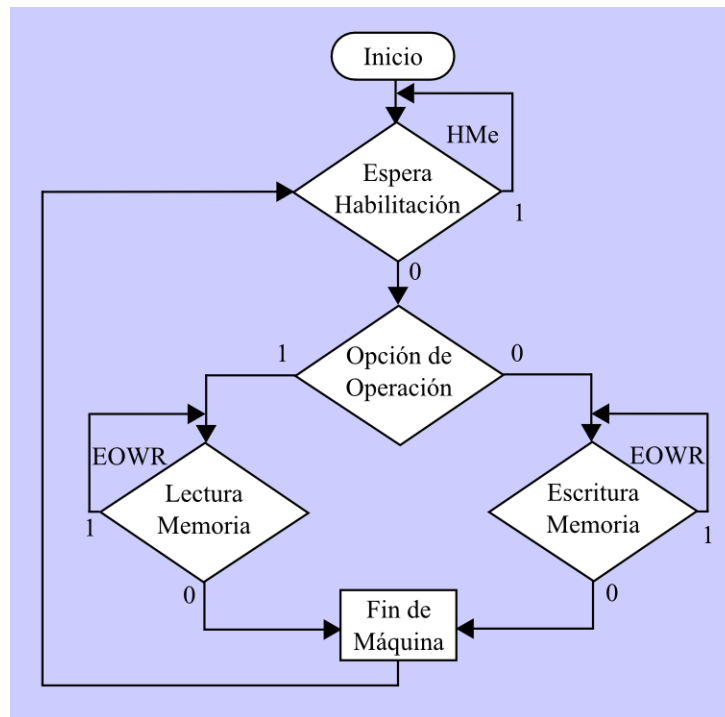


Figura 3.34. Diagrama de flujo del módulo manejador de memoria externa.

3.4.4 RS-232.

El módulo que realiza la comunicación con la computadora es el módulo RS-232 mostrado en la Figura 3.24. Este módulo de comunicación se encuentra estructurado como se observa en la Figura 3.35. Este módulo cuenta con el bloque de la máquina de estados RS_FSM, encargado de controlar los demás bloques. El bloque RS_Registro es un registro de desplazamiento en el cual se ingresan 16 *bits* y a su salida se tiene 8 *bits*. Es controlado por medio de la señal ORe. Además, para la comunicación con la computadora, se utiliza el módulo de transmisión de datos presentado en la sección 3.5. La máquina de estados se encarga de activar este módulo por medio de la señal STT y espera a que le indique su fin con la señal EOT. La señal Tx es la encargada de conducir los datos de una forma seria hacia la computadora.

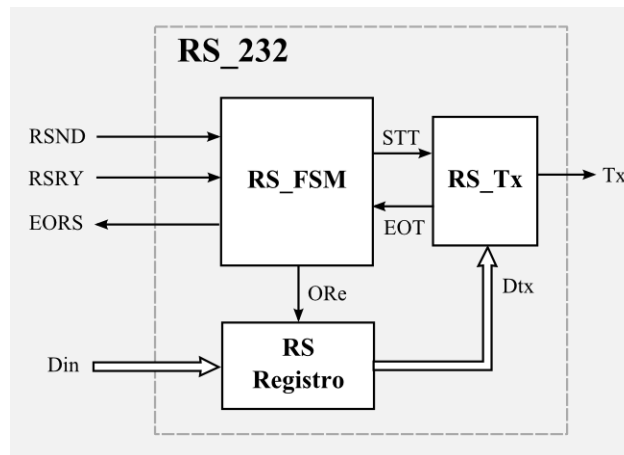


Figura 3.35. Diagrama a bloques del módulo de comunicación RS-232 para el sistema maestro.

Para realizar esto, la máquina de estados debe de llevar una secuencia de activación de módulos. Cabe señalar que la señal de entrada Din tiene un ancho de 16 *bits* y la entrada del bloque RS_Tx es de 8 *bits*, por lo que es necesario realizar dos envíos por dato ingresado. Debido a esto, se hace uso de la máquina de estados con el fin de controlar los dos envíos. En la Figura 3.36 se muestra el diagrama de flujo de la comunicación, después de inicializar, la máquina de estados se mantiene en espera de la activación de la señal RSND. Una vez recibida la señal, la máquina guarda en el registro el dato de 16 *bits* en la que se encuentra en Din, después inicia el módulo de transmisión y espera la señal de fin de transmisión EOT. Después de que se ha recibido, se corre el registro de desplazamiento

para enviar la segunda trama de 8 bits. Se inicia la transmisión después del corrimiento y se espera nuevamente a la recepción de la señal EOT del módulo de transmisión. Después de que se ha recibido, se envía el fin de máquina por medio de la señal EORS. Y espera la señal RSRY para volver al estado de espera de inicio de la máquina para enviar un nuevo dato.

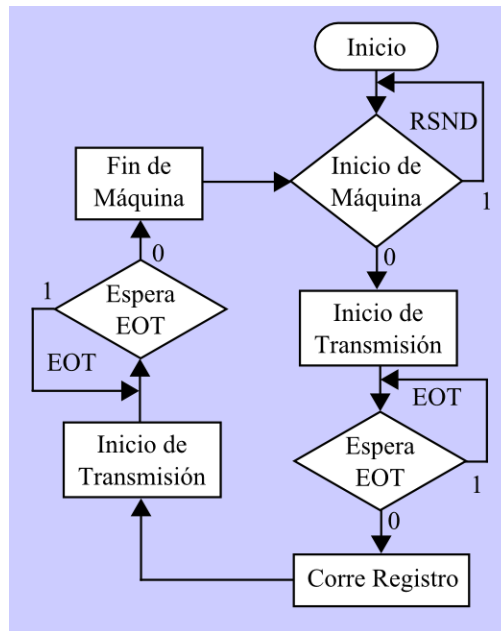


Figura 3.36. Diagrama de flujo de la máquina de estados del módulo de comunicación RS-232 para el sistema maestro.

3.5 Módulo de comunicación RS-232.

El módulo de comunicación serial RS-232 está conformado por el módulo de transmisión y el módulo de recepción. Ambos trabajan independientemente por lo que pueden instalarse ambos módulos para realizar la transmisión y recepción o solamente alguno de estos módulos.

3.5.1 Módulo de transmisión.

Este módulo es el encargado de realizar la transmisión de datos de modo serial con el protocolo RS-232. El diagrama a bloques se presenta en la Figura 3.37. El bloque se encarga de tomar los datos de la señal DTx y enviar los datos de manera serial por la señal de salida Tx. Este módulo cuenta con los bloques FSM_Tx, el cual es encargado de

controlar los demás bloques; el módulo divisor, este realiza la sincronización de la comunicación; el módulo detector de paridad realiza el cálculo para obtener el *bit* de paridad necesario en la comunicación y el multiplexor, encargado de enviar cada uno de los *bits* de acuerdo a su protocolo. El módulo divisor se programa de acuerdo con la velocidad de *bauds* de la comunicación serial. Este módulo es activado mediante la señal STT. Al terminar, la señal EOT es activada en bajo.

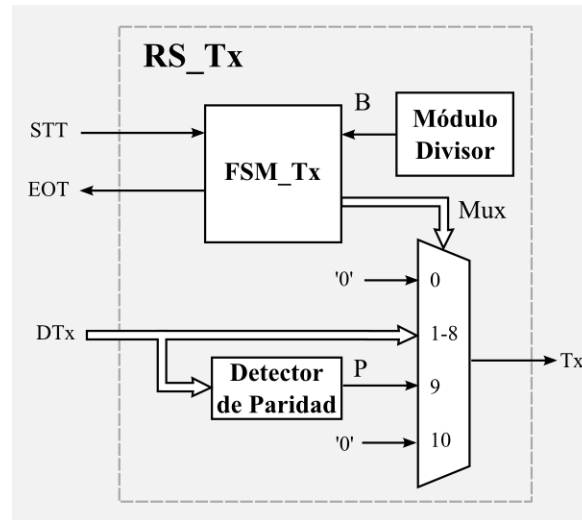


Figura 3.37. Módulo de transmisión del protocolo RS-232.

La máquina de estados se muestra en la Figura 3.38. El módulo de la máquina de estados espera la activación de la señal STT. Mientras espera, la señal de salida se mantiene en alto. Una vez activada, se envía la señal de inicio en bajo. Espera la señal B proveniente del módulo divisor. Después, la señal Mux va cambiando de acuerdo con el *bit* del dato correspondiente al protocolo RS-232. Cada uno de estos *bits* son sincronizados con el módulo divisor. Después del envío de datos, se envía el *bit* de paridad calculado y el *bit* de paro. Posteriormente, la señal de salida vuelve a estado alto. La máquina activa la señal EOT. Para indicar que la máquina ha finalizado y vuelve al estado inicial en espera de la señal STT.

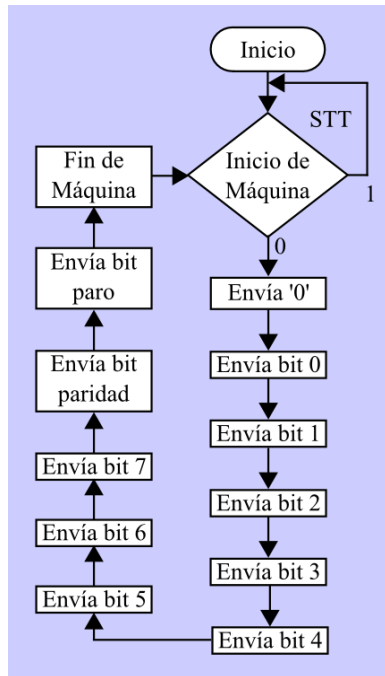


Figura 3.38. Diagrama de flujo de módulo de transmisión RS-232.

3.5.2 Módulo de recepción.

Este módulo es encargado de realizar la recepción de una forma serial por medio del protocolo RS-232. En la Figura 3.39 se observa la estructura interna del bloque de recepción. Al igual que con la transmisión, se cuenta con el módulo divisor para sincronizar la comunicación y la máquina de estados, encargada de controlar los demás bloques. El registro de desplazamiento almacena los *bits* provenientes de la línea de recepción, proporcionando el dato DRx después de cada recepción de datos y proporciona el *bit* de paridad calculado en el bloque que lleva el mismo nombre.

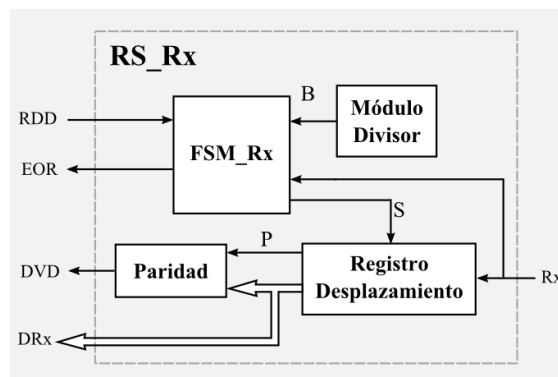


Figura 3.39. Módulo de recepción del protocolo RS-232.

Como se puede observar, en la Figura 3.40 se presenta el diagrama de flujo de la máquina de estados del módulo de recepción del protocolo RS-232.

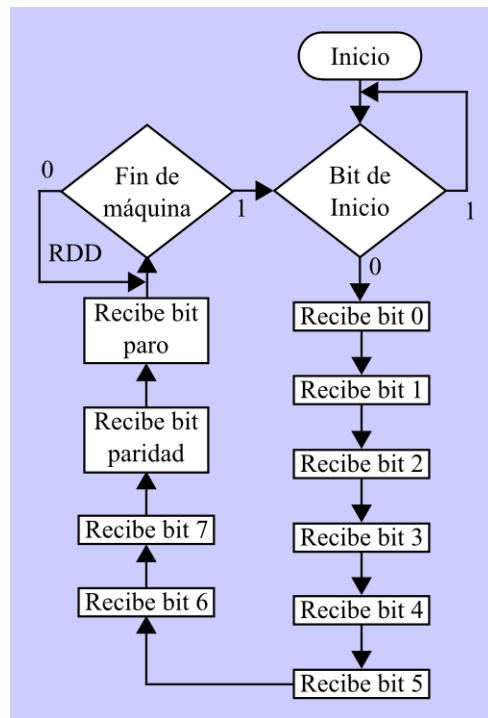


Figura 3.40. Diagrama de flujo de módulo de recepción RS-232.

Después de iniciar, la máquina de estados se mantiene en espera del *bit* de inicio en bajo proveniente de la señal Rx. Una vez recibido, se recibe el *bit* menos significativo del dato, se espera a que se reciba la señal del módulo divisor y se almacena en el registro de desplazamiento el siguiente *bit* del dato. Esto se repite hasta recibir el dato completo de ocho *bits*. Después de que se recibió el dato completo y que se encuentra en el registro de desplazamiento, se recibe el *bit* que indica el paro de la comunicación y se mantiene en el estado en donde se envía la señal de fin de máquina, esperando la señal RDD la cual indica que el dato almacenado en el registro de desplazamiento ha sido leído. Ya recibido, la máquina de estado vuelve a su estado de espera del *bit* de inicio que indica que la siguiente trama de datos debe de ser leída.

3.6 Procesamiento en software.

El procesamiento en software que se realiza en este trabajo es con el fin de estimar las señales inerciales sin aumentar los recursos utilizados en el procesamiento en hardware, ya que hace uso de operaciones complejas. Las señales inerciales obtenidas mediante el procesamiento en software son las señales de inclinación, de velocidad y de jerk. El programa en que se realiza el procesamiento es MATLAB. En la Figura 3.41 se muestra el diagrama de flujo del procesamiento en software del sistema.

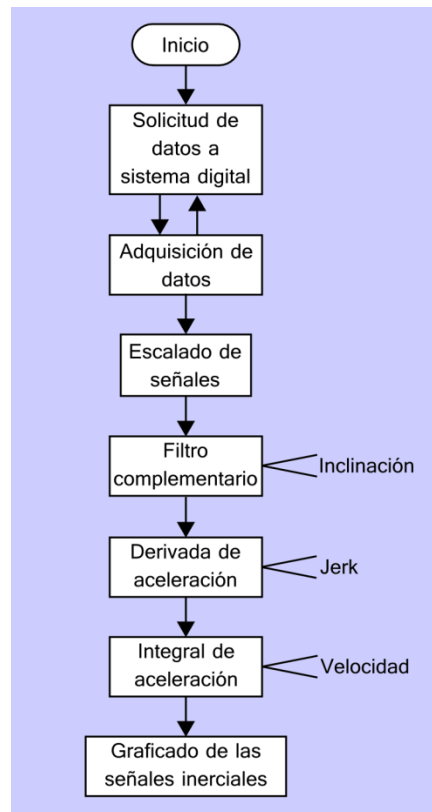


Figura 3.41. Diagrama de flujo del procesamiento en software.

De acuerdo a lo que se muestra en la Figura 3.41, la PC realiza la petición de datos al sistema digital. Posteriormente, adquiere los datos enviados por el sistema digital y los almacena en software MATLAB. Los pasos anteriores son repetidos hasta recibir los datos completos de las señales inerciales. La comunicación entre el sistema digital y la PC se realiza mediante protocolo RS-232. Después de recibidos los datos, se realiza el escalado de las señales para manejar los datos en las unidades adecuadas. Con los datos escalados, se

aplica un filtro complementario para obtener la señal de inclinación mediante las señales de aceleración y velocidad angular. Después de que se estima la señal de inclinación, se utiliza la derivada de la aceleración para obtener la señal de jerk. Para estimar la velocidad, se integra la señal de aceleración. Una vez estimada la velocidad, se cuenta con las seis señales inerciales. El software realiza el graficado de las señales inerciales y de sus ejes, con los cuales se puede analizar el comportamiento de las señales inerciales en la máquina CNC o robot en el que se colocaron los sensores.

3.7 Hardware del sistema.

Como ya se ha visto en la descripción del sistema general en la sección 3.1, para el desarrollo en hardware del sistema digital inalámbrico de señales inerciales es necesario la implementación del sistema maestro y del sistema esclavo. Para el sistema maestro se utilizó la tarjeta *Spartan3 Starter Kit* con 1 000 000 de compuertas, la cual cuenta con 4320 celdas lógicas equivalentes, 30 000 bits de RAM distribuida, 216 000 bits en bloques de RAM y 12 multiplicadores. Para el sistema esclavo se utilizó la tarjeta *Spartan3 Starter Kit* con 200 000 compuertas, la cual cuenta con 17280 celdas lógicas equivalentes, 120 000 bits de RAM distribuida, 432 000 bits en bloques de RAM y 24 multiplicadores. Ambas tarjetas cuentan con memorias SRAM, contando con una capacidad de 1 Megabyte.

Además de estas tarjetas, se hace uso del transceptor de comunicación inalámbrica nRF24L01. Este transceptor opera en la banda de frecuencia de 2.4 GHz con 126 canales de radio frecuencia utilizando una modulación GFSK. Su frecuencia de comunicación inalámbrica puede ser variada en 1Mbps o 2Mbps. De igual manera, el envío de datos inalámbricamente puede ser establecido de 1 a 32 bytes a deseo del operador. La potencia de salida puede ser programada de 0, -6, -12 o -18 dBm por lo que la corriente consumida se encuentra en el rango de 900nA hasta 22 uA. El transceptor cuenta con un sistema de registros para establecer las características de comunicación. Estos registros son modificados mediante el protocolo SPI.

Otra pieza importante en la constitución en hardware del sistema esclavo es el sensor de las señales inerciales. El sensor utilizado se diseñó en la Universidad Autónoma de Querétaro, Facultad de Ingenierías, por el Dr. Luis Morales Velázquez. El sensor se muestra en la Figura 3.42.

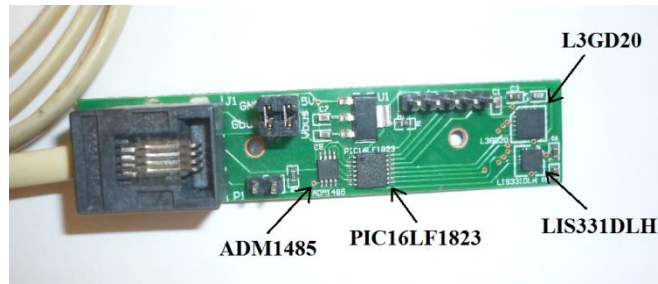


Figura 3.42. Sensor.

El sensor de la Figura 3.42 cuenta con el acelerómetro LIS331DLH y el giroscopio L3GD20, con éstos se adquieren la aceleración y la velocidad angular. Además, cuenta con un microcontrolador PIC16LF1823 para realizar la adquisición y envío de las lecturas del acelerómetro y el giroscopio cuando se le solicite. El sensor cuenta con un ADM1485, el cual es un transceptor de línea diferencial adecuado para altas velocidades. Este circuito integrado está diseñado para realizar una transmisión de datos equilibrada entre los protocolos RS-485 y RS-422. Por lo que la comunicación entre un sistema externo y el sensor se realiza mediante el protocolo RS-422 y este circuito integrado adecúa el protocolo para que el microcontrolador reciba la señal RS-485 y pueda realizar la lectura de las señales ingresadas. Junto con el sensor, se diseñó un adaptador para las tarjetas *Spartan3*, como se puede ver en la Figura 3.43

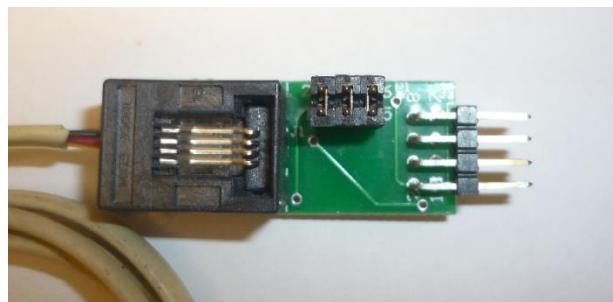


Figura 3.43. Adaptador para Spartan del sensor.

Este adaptador cuenta con otro circuito integrado ADM1485, pero conectado de una forma inversa, lo que facilita el procesamiento de las señales desde la tarjeta *Spartan3* ya que son ingresados los valores mediante el protocolo RS-232 y el circuito integrado adapta las señales al protocolo RS-422. Esta conversión permite ingresar las instrucciones de

lectura de dato al sensor utilizando los módulos ya desarrollados para este trabajo. La conexión entre el sensor y su adaptador se muestra en la Figura 3.44

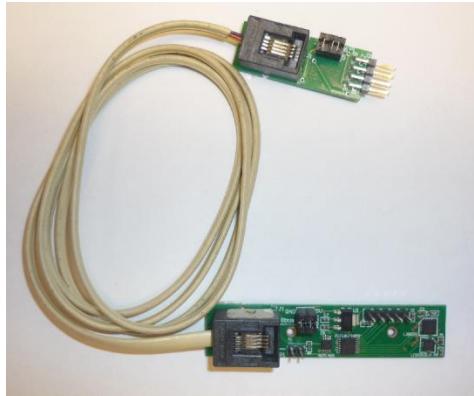


Figura 3.44. Conexión entre sensor y adaptador.

El sistema esclavo en hardware se muestra en la Figura 3.45. Como se puede observar, el sensor es conectado alámbricamente a la tarjeta *Spartan3* utilizando el adaptador antes descrito. Este adaptador es colocado en los puertos de expansión de la tarjeta, al igual que el transceptor nRF24L01, encargado de realizar la comunicación inalámbrica con el sistema maestro. El microcontrolador es programado para que, en el momento que reciba un dato proveniente de la tarjeta del sistema esclavo, éste realice la lectura de los tres ejes del acelerómetro y del giroscopio y en respuesta, envíe esos datos leídos hacia la tarjeta *Spartan*.

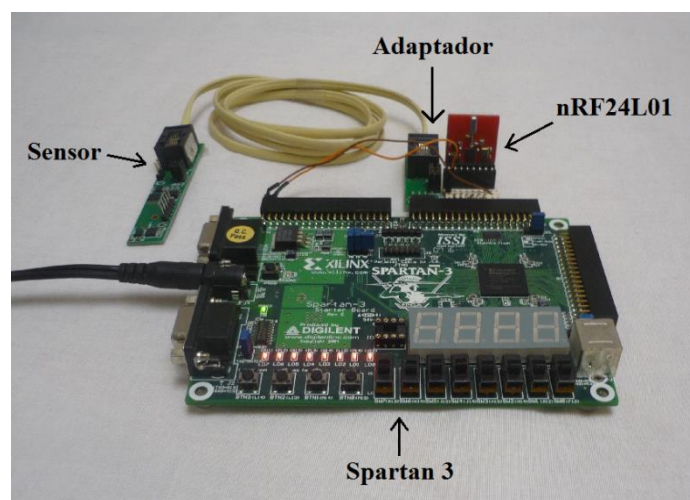


Figura 3.45. Hardware del sistema esclavo.

El sistema maestro en hardware se muestra en la Figura 3.46. Como se puede observar, el transceptor nRF24L01 es colocado en los puertos de expansión de la tarjeta para la comunicación inalámbrica con el sistema esclavo. Además del transceptor, a la tarjeta *Spartan* se le conecta un cable USB-DB9 para comunicarse con la computadora. Como se puede ver, este cable se conecta en el puerto serial RS-232 de la tarjeta.

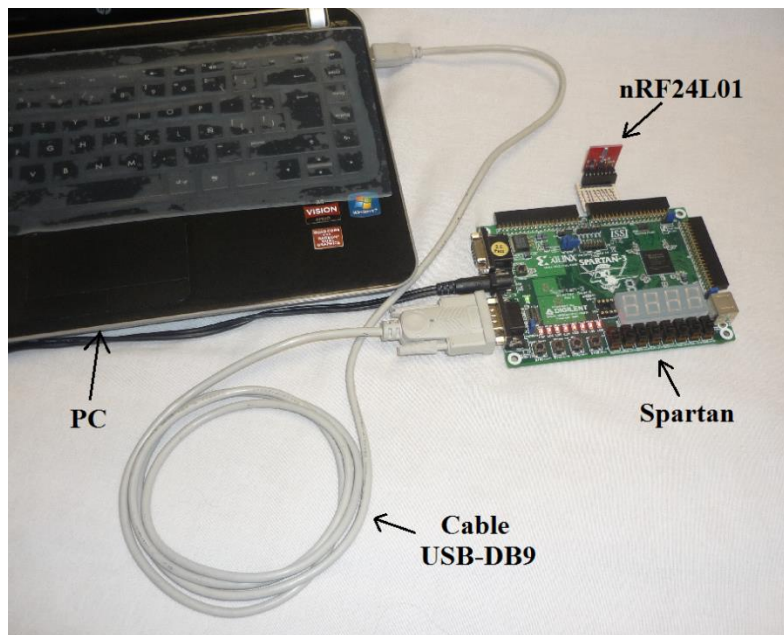


Figura 3.46. Hardware del sistema maestro.

4 PRUEBAS Y RESULTADOS.

En este capítulo se presentan las pruebas realizadas para validar el funcionamiento del sistema. Estas pruebas fueron de tiempo de procesamiento, de distancia de comunicación, de las señales adquiridas en la máquina CNC y en el robot y finalmente, de consumo de potencia. A continuación se presentan las características de las pruebas realizadas y sus resultados.

4.1 Prueba de distancia de comunicación inalámbrica.

Es importante obtener el rango de distancia en la cual el sistema envía correctamente los datos en condiciones normales de operación. La prueba consistió en colocar en una posición fija el sistema maestro e ir incrementando la distancia a la que se encontraba el sistema esclavo con una variación de 1 metro. Las pruebas se realizaron en un espacio libre, es decir, sin objetos que interfirieran en la comunicación inalámbrica. Esto con el fin de obtener la distancia máxima a la que el sistema trabaja correctamente. Se realizaron adquisiciones de datos durante 10 segundos en cada una de las pruebas, con una frecuencia de muestreo de 1 KHz. Para obtener la descripción correcta del comportamiento del sistema, se realizaron 40 pruebas para cada variación de distancia entre el sistema maestro y el sistema esclavo. En cada prueba se registraron la cantidad de datos que se recibieron correctamente para posteriormente obtener el promedio de datos recibidos, su varianza, su desviación estándar y su error estándar.

En la Tabla 4.1 se presentan los resultados de la prueba de distancia de comunicación. Como se puede observar, desde las distancias de 1 metro hasta 19 metros, el sistema trabaja de una manera correcta con un promedio de 10000 datos recibidos por muestra. Su varianza, desviación y error estándar tienen el valor de cero.

Tabla 4.1. Resultados de prueba de distancia de comunicación.

Distancia [m]	Promedio	Varianza	Desviación	Error Estándar
1 a 19	10000	0	0	0
20	9999.6	2.99	1.729161647	0.273404462
21	9855.05	3168.8475	56.29251726	8.900628489
22	8776.05	53775.4475	231.8953374	36.66587225
23	7130.5	251025.25	501.024201	79.2188819
24	5612.225	296074.9744	544.1277188	86.03414647
25	2830.075	314055.9194	560.406923	88.60811466

De acuerdo con la Tabla 4.1, a partir de 20 metros de distancia, el sistema comienza a tener problemas al recibir las lecturas provenientes de los sensores con datos erróneos, debido a la potencia de alcance. Con las 40 repeticiones de cada prueba de 10 000 datos, se tiene un promedio de 9999.6 datos correctos en cada prueba. Como se puede observar, también se aumenta la varianza, la desviación estándar y se cuenta con un valor de error estándar. En las gráficas de la Figura 4.1 se presentan los resultados del cálculo de promedio, varianza, desviación estándar y error del procesador digital inalámbrico a las distintas longitudes probadas.

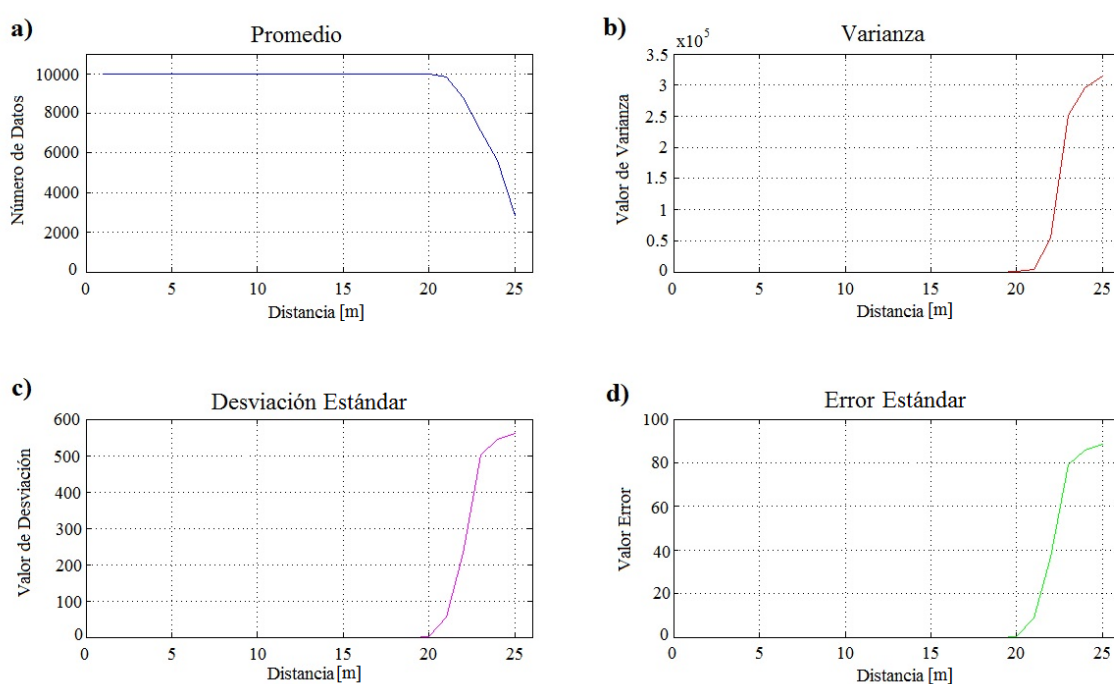


Figura 4.1. Gráficas resultantes de la prueba de distancia de comunicación inalámbrica. a) Gráfica de promedio. b) Gráfica de varianza. c) Gráfica de desviación estándar. d) Gráfica de error estándar.

En la Figura 4.1a) se muestra el resultado del promedio de datos recibidos para las distintas distancias en las que se realizaron las pruebas. A partir de la distancia de 20 metros el promedio comienza a reducirse hasta los 25 metros, en el cual se recibe una cantidad de datos muy reducida. Como se puede ver en las gráficas de la Figura 4.1b), c) y d), la varianza, la desviación estándar y el error estándar se incrementan en una proporción muy similar. Con esto se corrobora que el sistema tiene una comunicación inalámbrica

correcta desde 0 hasta 19 metros de distancia en un área libre. A partir de los 20 metros se pueden realizar adquisiciones menores de datos de una manera correcta de acuerdo a la gráfica del promedio de datos.

4.2 Prueba de tiempo de procesamiento.

Es importante obtener el tiempo de procesamiento para tener la frecuencia de trabajo máxima a la que operará el sistema digital inalámbrico de señales inerciales. Con este fin, es necesario realizar la prueba del tiempo de muestreo en el que se lleva a cabo el procesamiento en hardware. Este tiempo se divide en dos secciones, el tiempo del procesamiento esclavo, el tiempo de procesamiento maestro y el tiempo en aire, es decir, el tiempo que tarda en realizarse la comunicación inalámbricamente por parte de los transceptores. Estos tiempos se suman y de esta manera se obtiene el tiempo máximo de adquisición. Para la comprobación de este tiempo, se implementa un contador de ciclos desde que el maestro realiza una petición hasta que recibe una adquisición de datos del sensor y se realiza una nueva petición. De esta manera se obtiene el tiempo máximo de muestreo del sistema.

Para obtener el tiempo de procesamiento del sistema maestro mediante la simulación, se dividen las operaciones que realizan las diferentes máquinas de estados como se puede ver en la Figura 4.2.

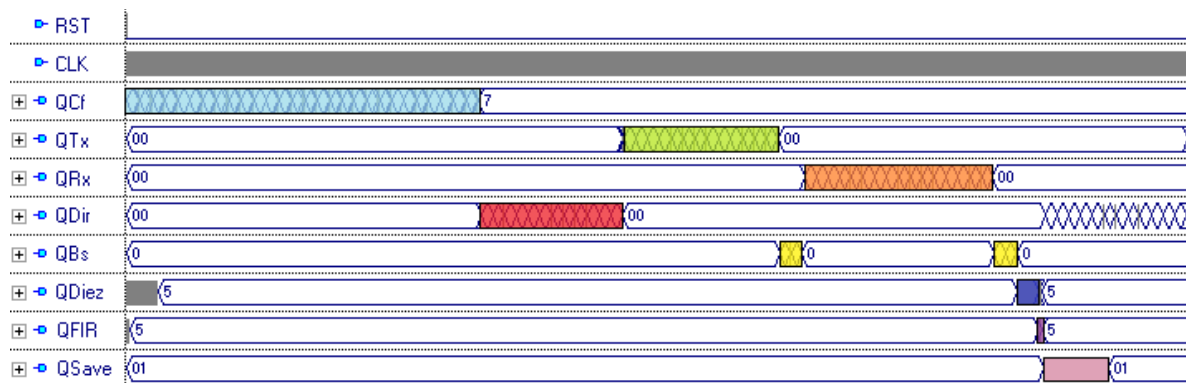


Figura 4.2. Simulación del procesamiento del sistema maestro.

Como se puede ver en la Figura 4.2, se cuenta con diferentes máquinas de estados donde *QCf* son los estados de la configuración de los registros del transceptor nRF24L01,

QTx es la máquina de estados encargada de realizar la transmisión de la petición de datos enviada al esclavo, *QRx* es la máquina de estados encargada de realizar la adquisición de los datos provenientes del sistema esclavo, *QDir* es la máquina que configura la dirección del transceptor del sistema maestro para comunicarse con el transceptor del sistema esclavo, *QBs* es la máquina encargada de borrar la pila de datos que se encuentra en el transceptor para evitar que los datos recibidos se solapen, *QDiez* es la máquina de estados que se encarga de realizar el filtro de diezmado por promedio, *QFIR* es la máquina que se encarga de realizar los filtros FIR pasa-bajas y pasa-altas y finalmente, *QSave* que es la máquina de estados encargada de realizar el guardado de datos obtenidos del sistema maestro en la memoria RAM externa.

Una vez que se ha explicado la separación del procesamiento en el sistema maestro, se presentan sus tiempos en la Figura 4.3. Como se puede ver, el tiempo de la configuración *QCf* es 83.01 μ s por lo cual no influye por realizarse una sola vez. El tiempo que influye en la frecuencia de muestreo es a partir de la configuración de la dirección del transceptor del sistema maestro con *QDir*. Posteriormente, se realiza la petición de datos enviando una señal con *QTx* y borra la pila del transceptor mediante *QBs*. El sistema se mantiene en espera de la respuesta del sistema esclavo. Una vez recibida, se descargan los datos con *QRx* y se borran los datos de la pila del transceptor. Finalmente se aplica el filtrado de diezmado, los filtros FIR pasa-altas y pasa bajas y se guardan los datos.

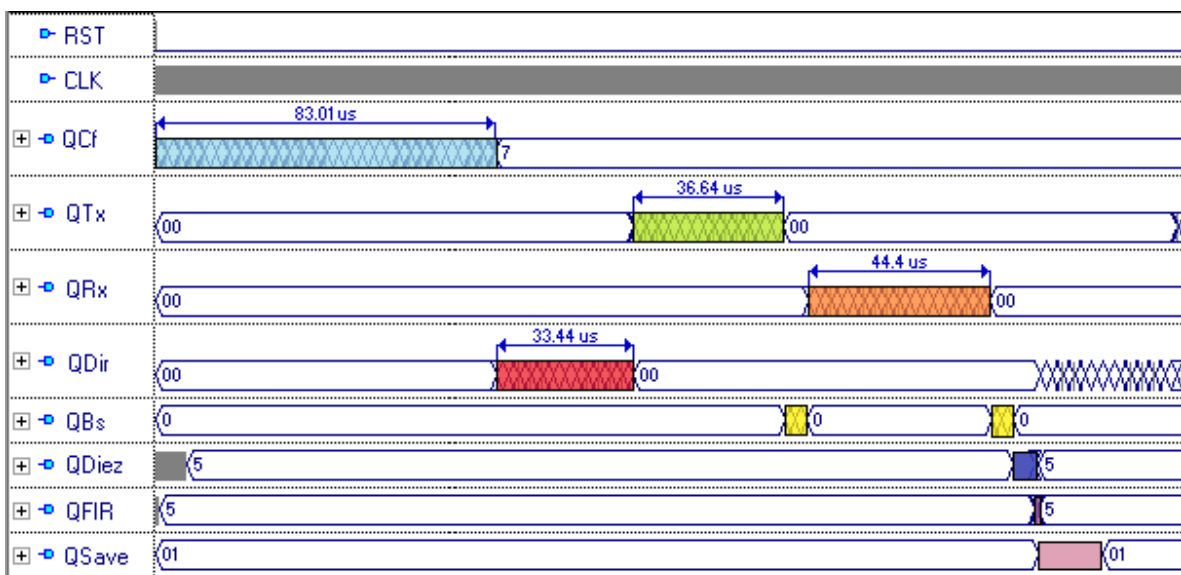


Figura 4.3. Tiempo de procesamiento simulado de QCf, QTx, QRx y QDir.

En paralelo con el guardado de datos, se realiza nuevamente la configuración de dirección del transceptor, por lo que el tiempo en el que se guardan los datos se encuentra dentro el tiempo que se lleva $QDir$. Por lo que el tiempo de cada muestreo comprende desde $QDir$ hasta $QFIR$. Al realizar el reinicio del sistema con la señal RST, $QDiez$ y $QFIR$ tienen unos ciclos de operación, estos son para la inicialización de valores de registros, por lo que no afectan en el tiempo de muestreo. En la Figura 4.4 se presenta el tiempo de procesamiento de las máquinas de estado QBs, $QDiez$ y $QSave$

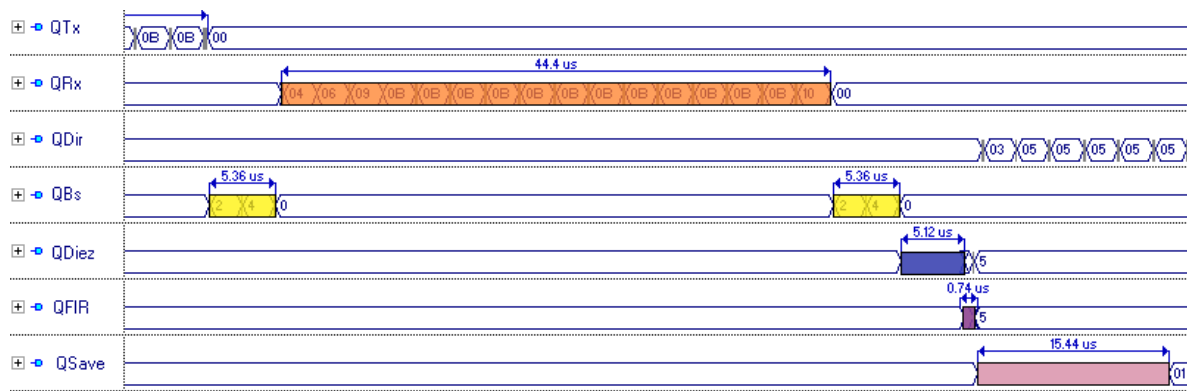


Figura 4.4. Tiempo de procesamiento simulado de QBs, $QDiez$ y $QSave$.

La suma del tiempo del procesamiento de las máquinas de estado da el tiempo de procesamiento del sistema maestro denotado por la ecuación (4.1).

$$T_{maestro} = T_{QDir} + T_{QTx} + T_{QRx} + 2T_{QBs} + T_{QDiez} + T_{QFIR} \quad (4.1)$$

El resultado del tiempo de procesamiento es:

$$T_{maestro} = 33.44 \mu s + 36.64 \mu s + 44.4 \mu s + 10.72 \mu s + 5.12 \mu s + 0.74 \mu s = 131.06 \mu s$$

Para obtener el tiempo de procesamiento del sistema esclavo mediante la simulación, de igual manera se dividen las operaciones que realizan las diferentes máquinas de estados como se puede ver en la Figura 4.5. Este sistema cuenta con menos máquinas de estados, pero mantiene la estructura de configuración y comunicación del sistema maestro con sus

máquinas QCf , QTx , QRx , $QDir$ y QBs . Además de estas, cuenta con la máquina de estados $Qsns$, encargada de la comunicación y adquisición de datos con sensor.

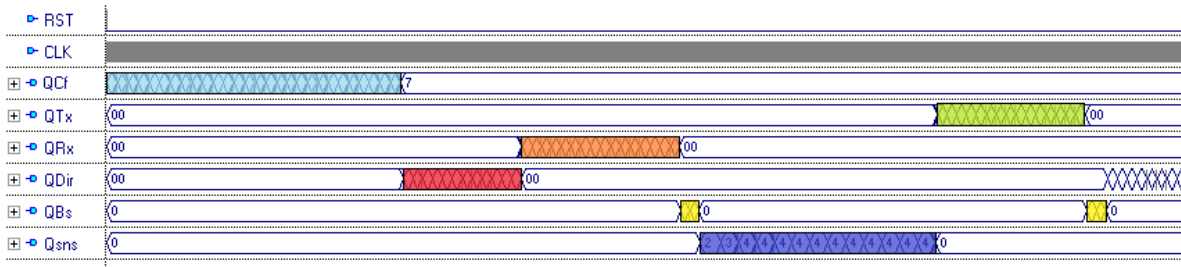


Figura 4.5. Simulación del procesamiento del sistema esclavo.

Al igual que el sistema esclavo, se realiza la configuración de las características del transceptor con la máquina de estados QCf . Enseguida se configura la dirección de comunicación inalámbrica del transceptor y se espera hasta que el sistema maestro envíe una petición. Una vez recibida, la máquina de estados $QDir$ descarga el valor del dato recibido y borra la pila del transceptor que lo almacena con QBs . Después de esto, se realiza la comunicación y adquisición de datos del sensor mediante la máquina $Qsns$. Ya adquiridos los datos, se envían hacia el sistema maestro con la máquina de estados QTx y se borra la pila del transceptor. Al igual que con el sistema maestro, se analiza al tiempo de procesamiento del sistema esclavo, el cual se muestra en la Figura 4.6.

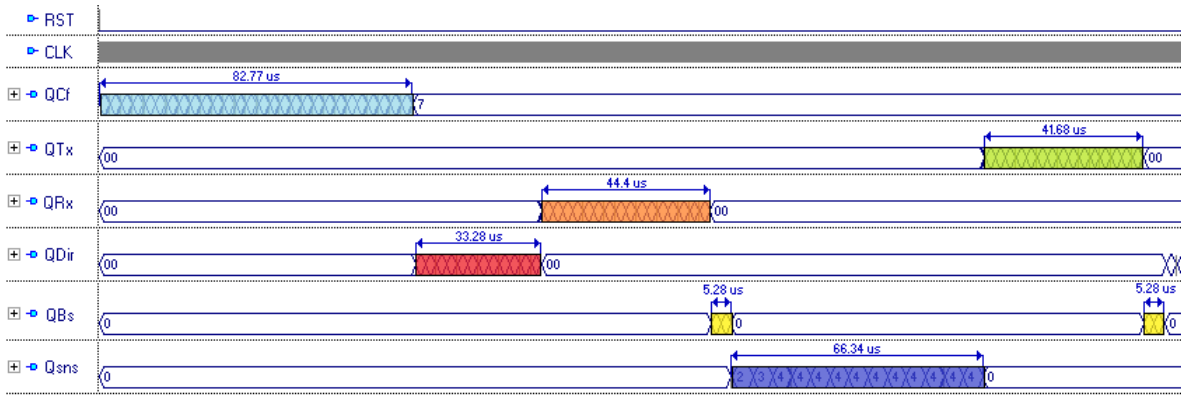


Figura 4.6. Tiempo de procesamiento simulado del sistema esclavo.

Para el análisis del tiempo de procesamiento del sistema esclavo cabe señalar que el tiempo de configuración de la máquina QCf se realiza en un tiempo de 82.77 μs y no

influye en éste. El tiempo de procesamiento de la máquina *QDir* se realiza en un tiempo de 33.28 μs en un tiempo muerto, es decir que no entra dentro del tiempo de muestreo. Por lo que el tiempo de procesamiento del sistema esclavo se expresa en la ecuación (4.2).

$$T_{esclavo} = T_{QRx} + T_{QTx} + 2T_{QBs} + T_{Qsns} \quad (4.2)$$

El resultado del tiempo de procesamiento esclavo es:

$$T_{esclavo} = 44.4 \mu s + 41.68 \mu s + 10.56 \mu s + 66.34 \mu s = 162.98 \mu s$$

Hasta el momento se tiene el tiempo de procesamiento del maestro y del esclavo, para obtener el tiempo de muestreo, es necesario el tiempo que se lleva la comunicación inalámbrica, este tiempo es denominado tiempo de aire y se determina mediante la ecuación (4.4), la cual es descrita por la hoja de datos del transceptor nRF24L01.

$$T_{aire} = \frac{\text{longitud del paquete}}{\text{frecuencia de transmisión}} \quad (4.3)$$

Desarrollando la ecuación (4.3), se tiene que la longitud del paquete se constituye de 1 *byte* de preámbulo de la comunicación entre transceptores, 5 *bytes* de la dirección del transceptor con el que se realiza la comunicación, N *bytes* del paquete de datos a enviar, 2 *bytes* de la comprobación de datos denominada CRC y 9 *bits* para el control del paquete. La longitud del paquete se define en bits y es dividido entre la frecuencia de transmisión como se puede ver en la ecuación (4.4).

$$T_{aire} = \frac{8[\text{bit/byte}] * (1[\text{byte}] + 5[\text{bytes}] + N[\text{bytes}] + 2[\text{bytes}]) + 9 [\text{bit}]}{\text{frecuencia de transmisión} [\text{bit/s}]} \quad (4.4)$$

Para el procesador digital inalámbrico de señales inerciales se realiza una transmisión de 12 bytes en el paquete de datos a enviar y de frecuencia de transmisión de 2Mbps, por lo que el tiempo aire de transmisión resultante es:

$$T_{aire} = 84.5 \mu s$$

Para obtener el tiempo total de procesamiento se realiza la suma del tiempo del maestro, el tiempo del maestro y el doble del tiempo aire debido a que se realiza la comunicación inalámbrica dos veces como se puede ver en la ecuación

$$T_{procesamiento} = T_{maestro} + T_{esclavo} + 2T_{aire} \quad (4.5)$$

La suma anterior da como resultado del tiempo de procesamiento:

$$T_{procesamiento} = 131.06 \mu s + 162.98 \mu s + 169 \mu s = 463.04 \mu s$$

En el resultado del contador de ciclos implementado, se obtiene un valor de 5814 por lo que se puede obtener el tiempo de procesamiento real descrita por la ecuación

$$T_{procesamiento\ real} = Conteo * \tau \quad (4.6)$$

En donde τ es el periodo de cada ciclo y Conteo es el valor obtenido. El periodo de cada ciclo utilizado en el procesador es de 80 ns.

$$T_{procesamiento\ real} = 5814 * 80\ ns = 465.12\ \mu s$$

Se puede ver una diferencia de 2.08 μs entre el tiempo de procesamiento real y el estimado por la simulación, esto es debido al tiempo de acción de las banderas y de las diferentes máquinas de estados y el tiempo de subida del transmisor para realizar la transmisión inalámbrica, como se explica en la hoja de datos. La frecuencia máxima de muestreo a la cual se puede realizar la comunicación se obtiene mediante el inverso del tiempo de procesamiento, por lo que la frecuencia máxima de muestreo es de:

$$f_{max} = 2149.98\ muestras/s \approx 2150\ muestras/s$$

4.3 Prueba de señales inerciales adquiridas en máquina CNC.

Un punto importante en la comprobación del sistema es su evaluación en la maquinaria CNC, por lo cual se adquieren las variables de aceleración, velocidad, jerk, inclinación, rotación y vibraciones. Para realizar la adquisición de estas señales se utilizó la fresadora que se muestra en la Figura 4.7. El sensor es colocado en la plancha inferior de la manera que se muestra y la cual es manejada en un solo eje, en este caso el eje X. El manejo del movimiento tal eje de la fresadora se realizó mediante el hardware y software desarrollado en la Facultad de Ingenierías de la Universidad Autónoma de Querétaro. Para realizar la prueba de la adquisición de las señales inerciales, el procesador digital inalámbrico presenta una frecuencia de muestreo de 1 KHz, los filtros pasa-bajas y pasa-altas son de orden 14 con una frecuencia de corte de 100 Hz y 200 Hz, respectivamente y el filtro de diezrado por promedio es de orden 128. Para corroborar el desplazamiento de la base de la máquina CNC, se adquirieron los datos provenientes de los motores a pasos con una frecuencia de muestreo de 1 KHz.



Figura 4.7. Fresadora CNC utilizada.

El movimiento de la plancha seleccionado para la prueba se muestra en la Figura 4.8. En esta prueba se decidió mover la plancha con un desplazamiento largo hacia la izquierda de 40000 pasos, como se puede ver en la gráfica de la Figura 4.8. Y posteriormente, realizar tres desplazamientos de 20 000 cada uno hacia la derecha hasta llegar a la posición de 20000 pasos a la derecha de nuestra posición de referencia.

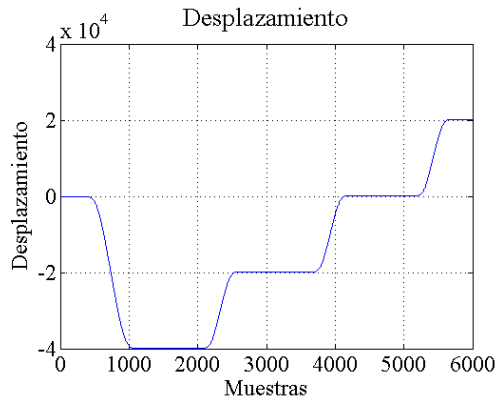


Figura 4.8. Desplazamiento de la plancha de la fresadora.

Con la lectura adquirida del acelerómetro y del giroscopio, se obtienen las seis señales inerciales que son graficadas a continuación.

4.3.1 Gráficas de aceleración.

Las gráficas de aceleración se muestran en la Figura 4.9, la aceleración influye principalmente en el eje X, pero los otros ejes también se ven afectados a ésta con una amplitud menor. Esto es debido a la colocación del acelerómetro con el cual se presentan errores en los otros ejes tales como el desfase en la amplitud.

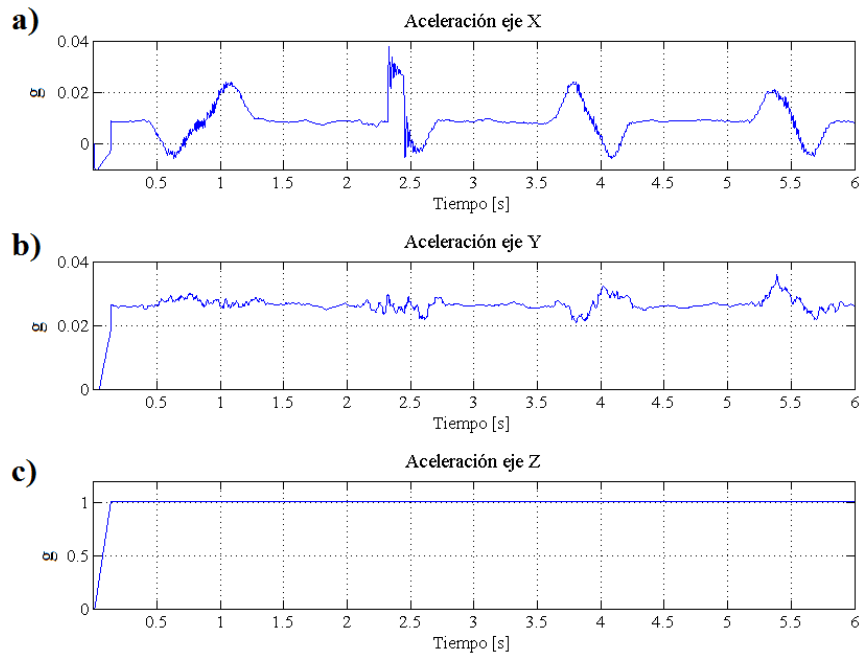


Figura 4.9. Gráficas de las señales de aceleración en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z.

La gráfica de aceleración en el eje x del sensor se presenta en la Figura 4.9a), como se puede observar, en el primer movimiento se tiene una aceleración con una pendiente positiva, lo cual nos indica que se mueve hacia la derecha y posteriormente tres aceleraciones con pendiente negativa, lo cual indica que se mueve hacia la izquierda. Además, en el segundo movimiento se tiene una pendiente una medición con ruido, lo que nos indica que la máquina tuvo un problema al momento de realizar este movimiento. En el inciso b) de la Figura 4.9 se presenta la aceleración en el eje Y, como ya se mencionó, corresponde a señales de ruido del sistema ya que idealmente, esta señal se mantendría con amplitud cero. La señal de aceleración del inciso c) de la Figura 4.9 corresponde a la gravedad y como se puede ver, se mantiene en 1 g. El eje Z también se ve afectado por la aceleración del movimiento, pero no influye tanto.

4.3.2 Gráficas de velocidad.

Las gráficas de la velocidad estimada se presenta en la Figura 4.10, al igual que con la aceleración, se presentan los ejes X, Y y Z.

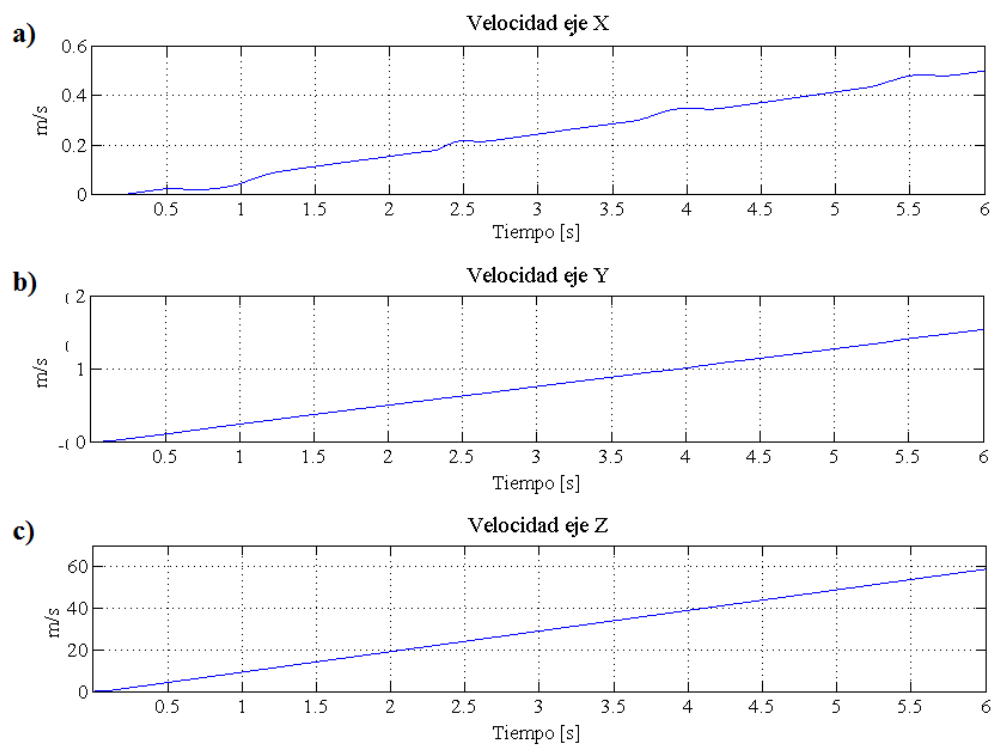


Figura 4.10. Gráficas de las señales de velocidad en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z.

Como se puede observar en la gráfica del inciso a) de la Figura 4.10 la aceleración muestra cambios en la velocidad en el eje X, mientras que en el inciso b), correspondiente al eje Y y el inciso c), correspondiente al eje Z, no se observan cambios notables a pesar de que existen alteraciones en su aceleración. Por lo que esos ruidos no afectan de manera notable la velocidad.

4.3.3 Gráficas de vibración.

Las gráficas de las vibraciones obtenidas por el procesador digital inalámbrico se presentan en la Figura 4.11 En estas se presentan los ejes X, Y y Z en los incisos a), b) y c) respectivamente.

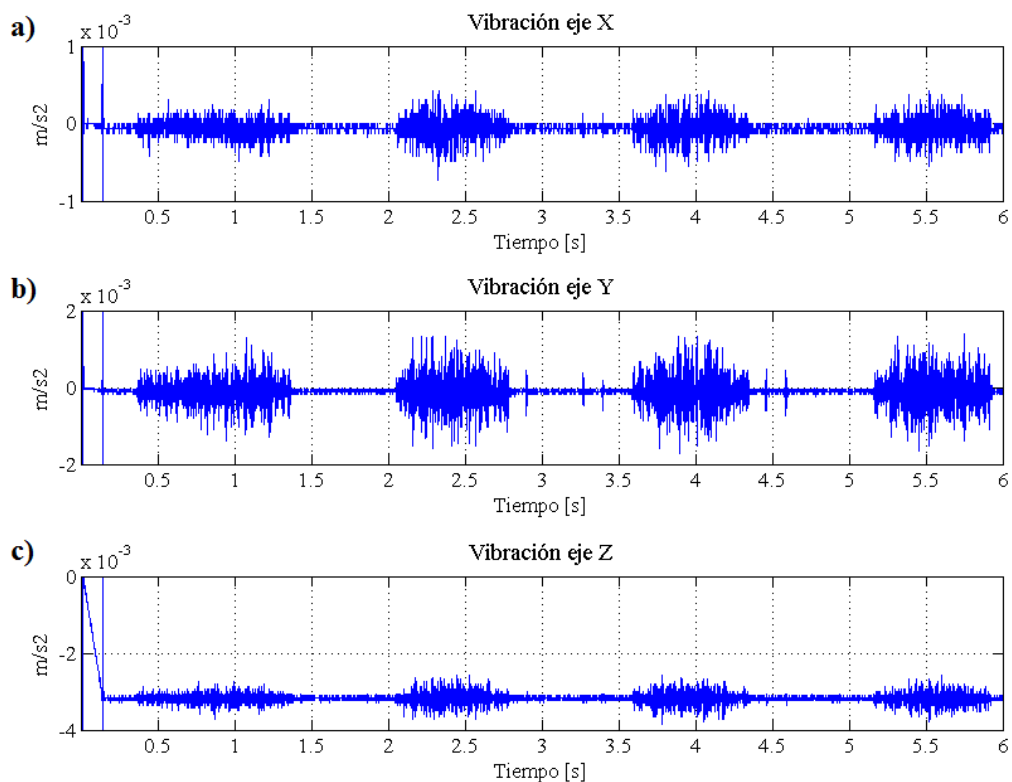


Figura 4.11. Gráficas de las señales de vibración en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z.

Como se puede observar en la Figura 4.11, se presentan las vibraciones naturales por el funcionamiento de la fresadora y en cada movimiento se generan vibraciones con una amplitud más grande que corresponden con el desplazamiento de la plancha. Estas gráficas en las señales, sirve de referencia para evaluar el funcionamiento correcto de una máquina

en general o si es que se tiene alguna anomalía, ya sea en la maquinaria o en las herramientas con las que se trabaja.

4.3.4 Gráficas de jerk.

La estimación del jerk es importante debido a las mediciones que proporciona de la máquina y las cuales sirven para evaluar el comportamiento y funcionamiento de la maquinaria. El jerk de los ejes X, Y y Z se muestra en la Figura 4.12 en los incisos a), b) y c) respectivamente. Como se puede notar, las amplitudes de los incisos b) y c) son más pequeñas que en el inciso a), por el efecto que tiene la aceleración sobre este eje. Como se mencionó en la gráfica de la aceleración, en la gráfica de la Figura 4.12 a), el segundo desplazamiento muestra una mayor amplitud que la de los demás, con lo cual se puede estimar que la máquina fresadora presentó un fuerte jaloneo posiblemente causado por alguna falla y de esta manera estimar el funcionamiento de la máquina. Las amplitudes de las señales mostradas en la Figura 4.12b) y c) son señales de ruido que se presenta en los ejes Y y Z respectivamente.

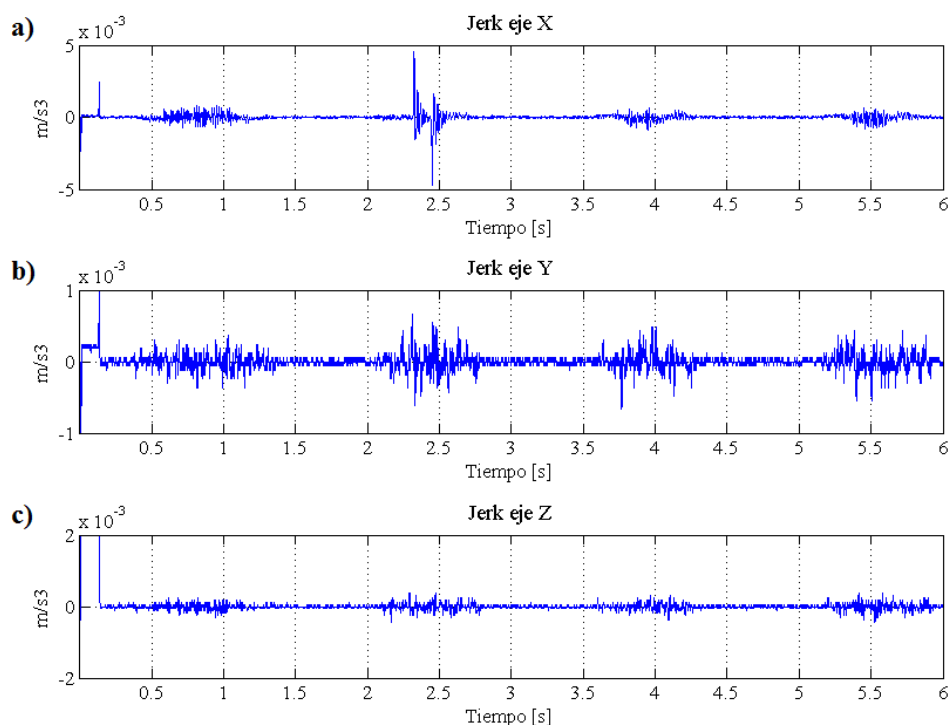


Figura 4.12. Gráficas de las señales de jerk en máquina CNC. a) Eje X. b) Eje Y. c) Eje Z .

4.3.5 Gráficas de inclinación.

Es importante evaluar la inclinación con la que cuenta la sección de la máquina en la que se ha colocado el sensor. Al realizar las pruebas en la máquina se presentan las señales de inclinación en la Figura 4.13. En este caso se presenta la inclinación en el eje X y en el eje Y, los cuales corresponden a la Figura 4.13a) y Figura 4.13b). El eje Z no se presenta debido a que no se puede registrar una inclinación correcta en el eje haciendo uso de la fusión de datos del acelerómetro y del giroscopio.

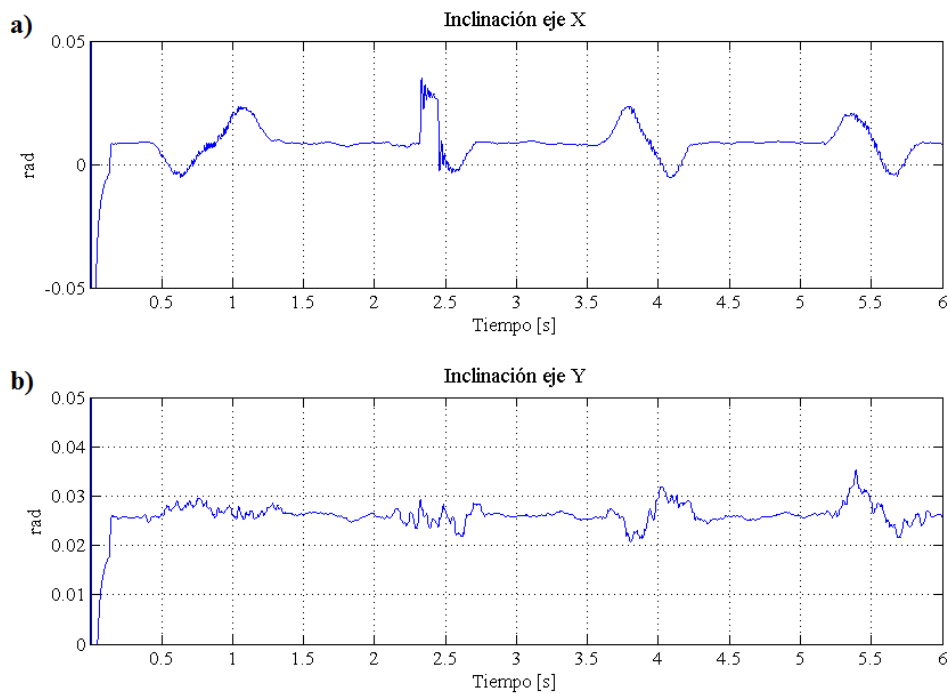


Figura 4.13. Gráficas de las señales de inclinación en máquina CNC. a) Eje X. b) Eje Y.

Como se observa en la Figura 4.13a), se genera una inclinación en cada caso en el que la plancha de la fresadora se mueve. En la teoría esta señal no debe de tener amplitud, al igual que la señal que se presenta en la Figura 4.13b). Pero analizando los valores de amplitud que presentan ambas señales de los ejes, se observa que su amplitud es muy pequeña, por lo que no afecta considerablemente el ángulo. En estos también se observa un desfase que indica que el sensor no se encuentra colocado de una manera horizontal perfecta. Debido a lo anterior, se establece que las señales presentadas son ruido que afecta a la inclinación.

4.3.6 Grafica de rotación.

Para obtener la velocidad de posición en el eje Z del sistema, se adquiere la señal de rotación, la cual es presentada en la Figura 4.14.

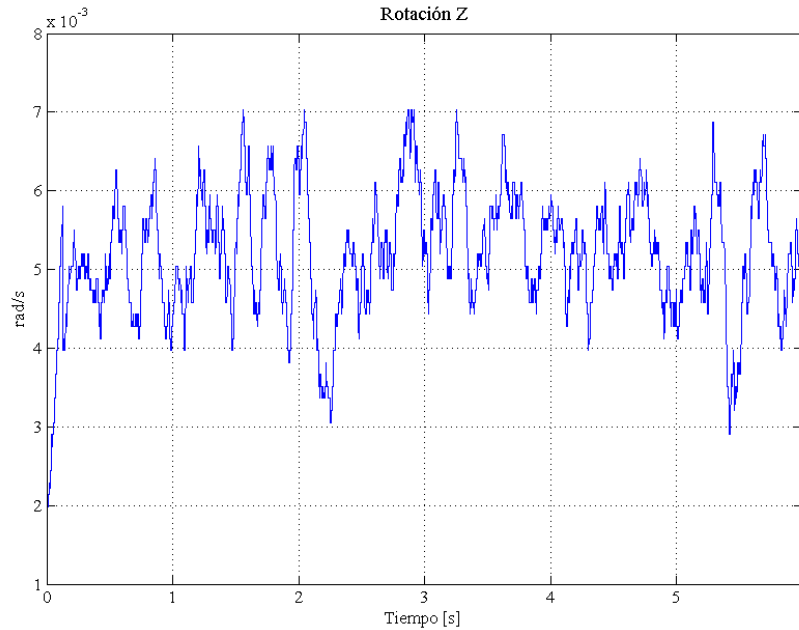


Figura 4.14. Gráficas de las señales de rotación en máquina CNC.

Como se puede observar en la Figura 4.14, al igual que con las gráficas de la inclinación, la amplitud de la señal de rotación es pequeña, además de variar de tal forma que no cuenta con el patrón del desplazamiento de la base de la fresadora, por lo que queda claro que se trata de una señal de ruido proveniente del giroscopio y en la cual, la acción de la aceleración no afecta de forma significativa. Esto es de esperarse, debido a que no se está realizando un movimiento giratorio a través del eje Z.

4.4 Prueba de señales inerciales adquiridas en Robot.

De igual manera que con la máquina CNC, se realizaron las pruebas de la adquisición de señales inerciales en un robot, obteniendo las variables de aceleración, velocidad, jerk, inclinación, rotación y vibraciones. El robot utilizado para esta prueba es el robot PUMA que se muestra en la Figura 4.15. El sensor es colocado en la punta del robot. El manejo del robot PUMA se realizó mediante el hardware y software desarrollado en la Facultad de Ingenierías de la Universidad Autónoma de Querétaro. Para realizar la prueba

de la adquisición de las señales inerciales en el robot, el procesador digital inalámbrico presenta una frecuencia de muestreo de 1 KHz, los filtros pasa-bajas y pasa-altas son de orden 14 con una frecuencia de corte de 100 Hz y 200 Hz, respectivamente y el filtro de diezmado por promedio es de orden 128.

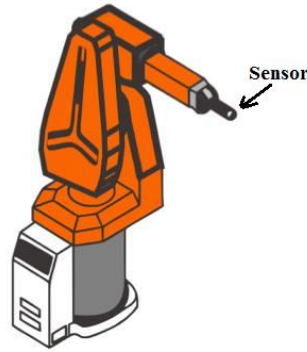


Figura 4.15. Robot PUMA probado.

Para revisar el movimiento del robot se tomaron los valores del conteo de los servomotores. El conteo de cada revoluta se muestra en la Figura 4.16. En esta prueba, se tomó un desplazamiento ya programado en el control del robot.

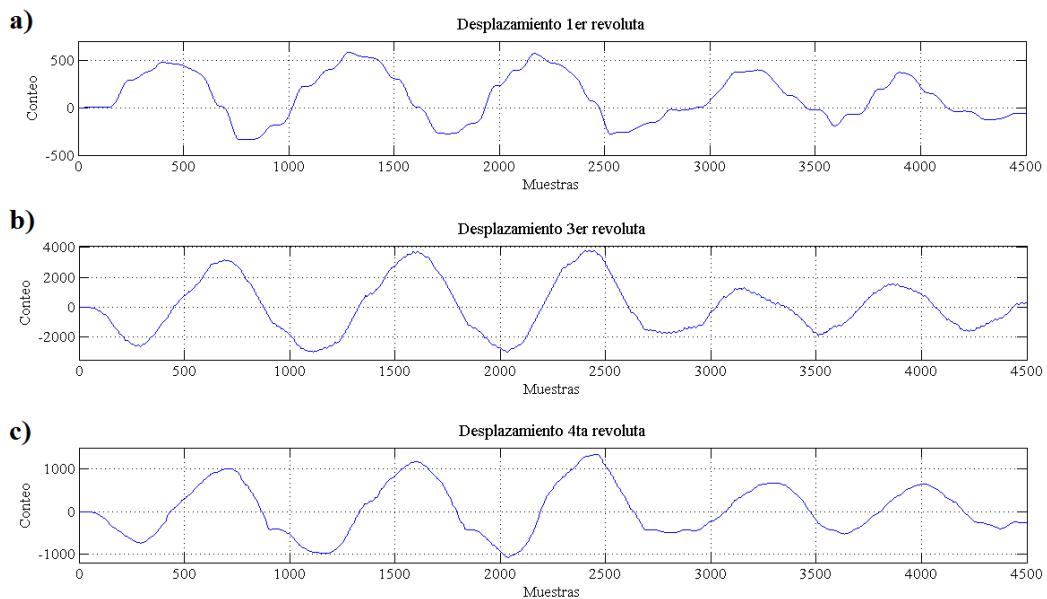


Figura 4.16. Desplazamiento de las revolutas. a) Desplazamiento de la primera revoluta del robot PUMA. b) Desplazamiento de la tercera revoluta del robot PUMA. c) Desplazamiento de la cuarta revoluta del robot PUMA.

El movimiento generado es un movimiento circular en el eje X y en el eje Y. Al realizar el desplazamiento de las revolutas, se adquirieron los datos del acelerómetro y del giroscopio para obtener las seis señales inerciales con el sistema digital.

4.4.1 Gráficas de aceleración.

Las gráficas de aceleración del robot obtenidas con el sistema digital se muestran en la Figura 4.17, la aceleración se presenta en los tres ejes, pero en donde se observa un mayor cambio es en el eje X y Y, ya que es donde se presenta el movimiento circular. En el eje Z se observa que se mantiene el valor cercano del nivel de 1g, las variaciones que presenta es debido a las vibraciones que presenta el robot.

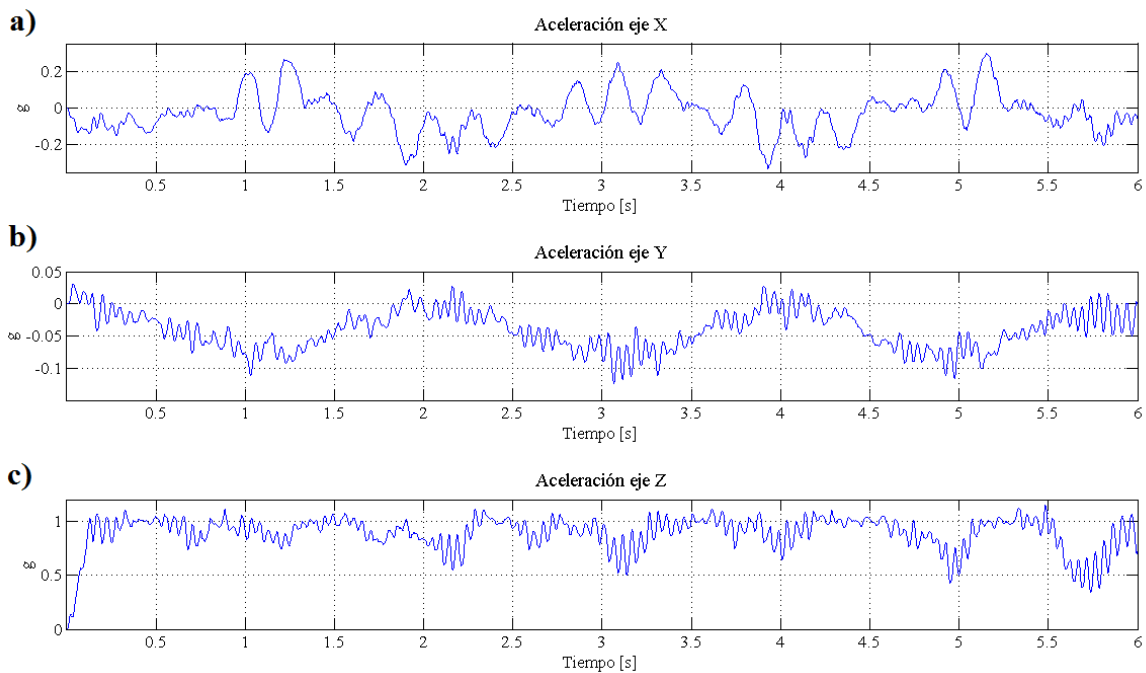


Figura 4.17. Gráficas de las señales de aceleración del robot. a) Eje X. b) Eje Y. c) Eje Z.

La gráfica de aceleración del robot en el eje x se presenta en la Figura 4.17a), como se puede observar, debido a que el robot no presenta un comportamiento circular perfecto, se obtiene una gráfica con variaciones grandes, pero con un comportamiento con tendencia oscilatoria. Por lo que si se genera el movimiento deseado en este eje. En el inciso b) de la Figura 4.17 se presenta la aceleración medida en el eje Y, en este eje se observa una señal con una tendencia oscilatoria mayor que el eje X. Al igual que con el eje X, en el eje Y se

presentan señales de ruido en la señal, esto debido a la alta vibración del brazo robótico. La señal de aceleración del inciso c) de la Figura 4.17 corresponde a la gravedad y como ya se mencionó, se mantiene en 1 g. El eje Z también se ve afectado por la vibración del robot.

4.4.2 Gráficas de velocidad.

Las gráficas de la velocidad estimada para el robot se presenta en la Figura 4.18, al igual que con la aceleración, se presentan los ejes X, Y y Z.

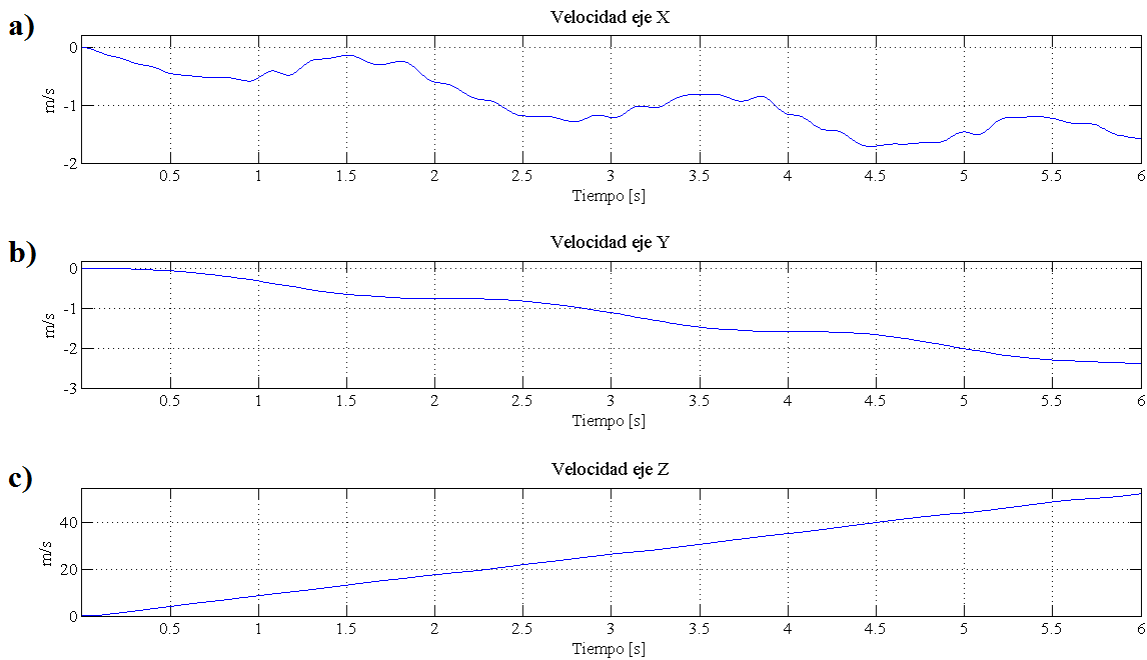


Figura 4.18. Gráficas de las señales de velocidad del robot. a) Eje X. b) Eje Y. c) Eje Z

Como se puede observar en la gráfica del inciso a) de la Figura 4.18, la aceleración presenta cambios oscilatorios en la velocidad en el eje X. De igual manera, pero con una menor amplitud, la velocidad del eje Y presentado en el inciso b) de la Figura 4.18 cuenta con oscilaciones. Por otro lado, el inciso c) de la Figura 4.18 que corresponde al eje Z, se mantiene prácticamente constante, y los cambios que presenta son debidos al ruido detectado en este eje. Analizando estas gráficas, se observa que el robot tiene el movimiento circular en el eje X y Y, pero las vibraciones afectan también el eje Z.

4.4.3 Gráficas de vibración.

Las gráficas de las vibraciones del robot obtenidas por el procesador digital inalámbrico se presentan en la En estas se presentan los ejes X, Y y Z en los incisos a), b) y c) respectivamente.

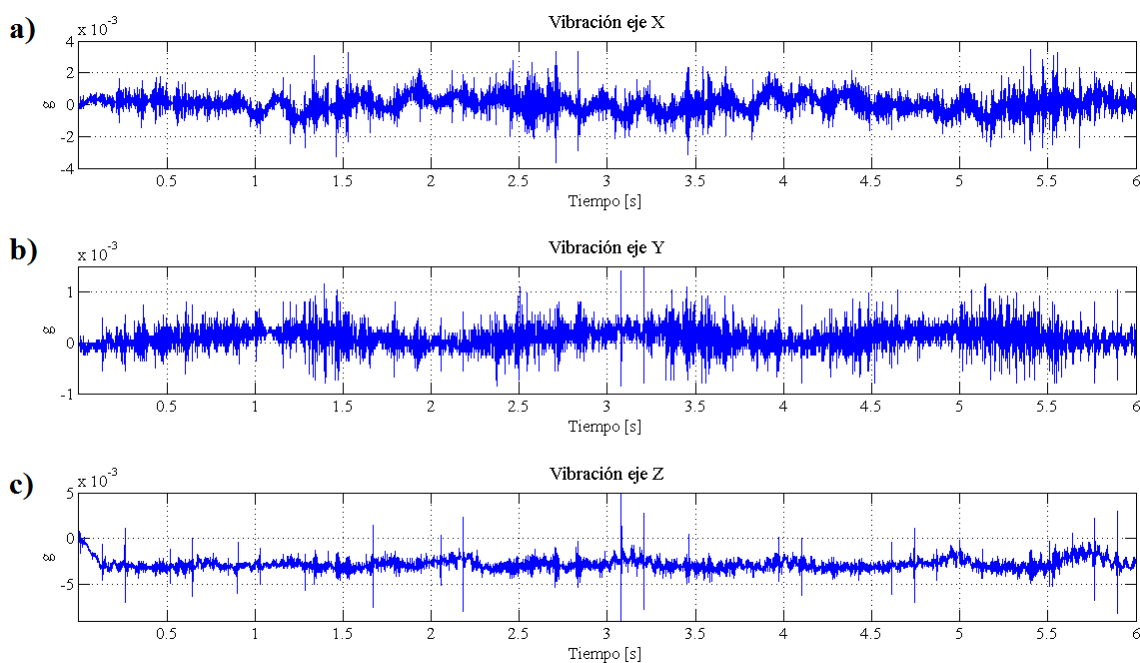


Figura 4.19. Gráficas de las señales de vibración del robot. a) Eje X. b) Eje Y. c) Eje Z

Como se puede observar en la Figura 4.19, se presentan las vibraciones naturales del movimiento circular generado en el robot en los ejes X, Y y Z correspondientes a los incisos a), b) y c) de la Figura 4.19, respectivamente. Estas gráficas muestran en qué parte del movimiento circular existe una mayor vibración en el brazo robótico. En el eje X y Y se tiene una pequeña tendencia oscilatoria debido al movimiento circular, relación que el eje Z no presenta de manera clara.

4.4.4 Gráficas de jerk.

La gráficas obtenidas de la estimación del jerk en los ejes X, Y y Z, se presenta en los incisos a), b) y c) de la Figura 4.20, respectivamente.

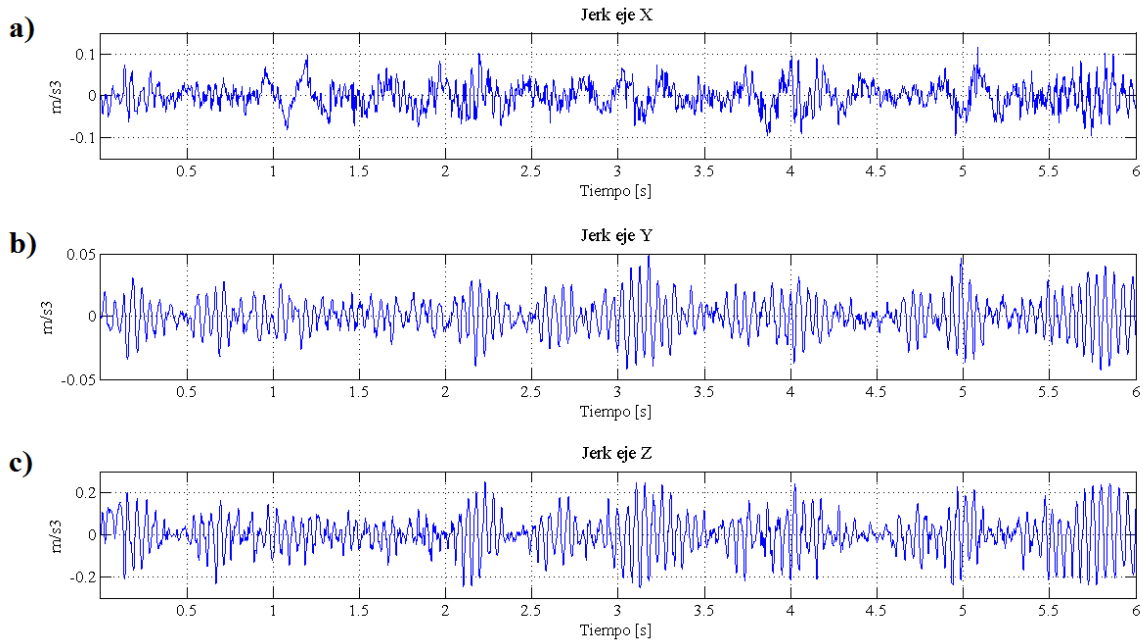


Figura 4.20. Gráficas de las señales de jerk del robot. a) Eje X. b) Eje Y. c) Eje Z.

En el inciso a) de la Figura 4.20, correspondiente al jaloneo del eje X, se muestran variaciones debido al movimiento circular programado en el robot y a sus vibraciones. Esta gráfica no presenta la relación tan marcada que tienen las gráficas de los incisos b) y c). Como se puede ver, el jaloneo en el eje Y y Z tiene un comportamiento muy parecido.

4.4.5 Gráficas de inclinación.

Al igual que con la máquina CNC, es importante evaluar la inclinación del brazo robótico. Debido a esto, se obtiene una posición con mayor exactitud. En este caso para la punta del robot. Al realizar las pruebas en el robot, se obtienen las señales de inclinación que se muestran en la Figura 4.21. En este caso se presenta la inclinación en el eje X y en el eje Y, los cuales corresponden a la Figura 4.21a) y Figura 4.21b). El eje Z no se presenta, como se mencionó para las máquinas CNC, debido a que no se puede registrar una inclinación correcta en el eje haciendo uso de la fusión de datos del acelerómetro y del giroscopio.

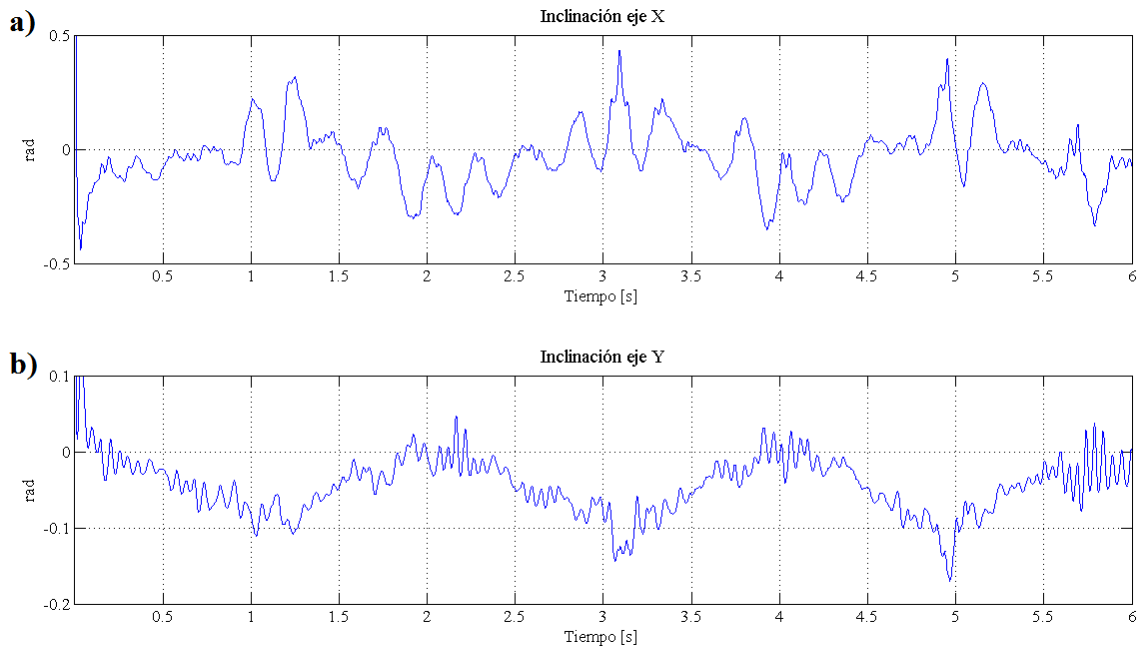


Figura 4.21. Gráficas de las señales de inclinación del robot. a) Eje X. b) Eje Y

Como se observa en los incisos a) y b) de la Figura 4.21, al generar el movimiento circular en el eje X y Y, se presenta una inclinación en dichos ejes, teniendo un comportamiento oscilatorio. Para el caso del eje X, se observa de una manera más clara esta oscilación que con las pruebas realizadas con el acelerómetro. Las vibraciones afectan de igual manera la estimación de la inclinación como se puede observar.

4.4.6 Gráfica de rotación.

Para obtener la rotación, se adquiere la velocidad de posición en el eje Z leída del sistema digital, la gráfica resultante se presenta en la Figura 4.22. Al igual que con las gráficas de la inclinación, la gráfica de la rotación en Z presenta la oscilación generada por el movimiento circular del robot. Aunque la amplitud es pequeña, puede observarse este comportamiento acompañado de las vibraciones que afectan al robot y pudieran dañarlo o tener un posicionamiento inexacto en el caso de que se desee realizar un control sobre éste.

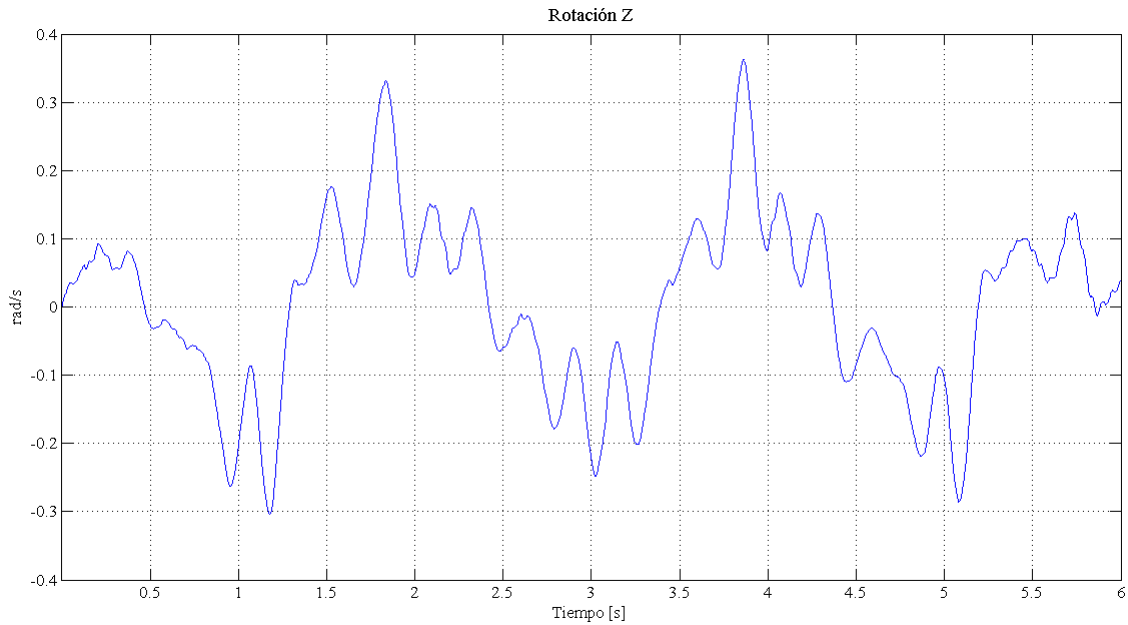


Figura 4.22. Gráficas de las señales de rotación del robot.

4.5 Prueba consumo de potencia.

La prueba del consumo de potencia se realizó para analizar la posibilidad de conectar el sistema esclavo del procesador digital inalámbrico a una batería, esto con el fin de montarlo en secciones donde la alimentación mediante un eliminador sea difícil. Para esta prueba, se realizó la medición de corriente y voltaje que consume el sistema esclavo del procesador digital inalámbrico a su entrada de alimentación mientras se encontraba en operación. Este procedimiento se repitió cincuenta veces, registrando cada una de las lecturas y sacando un promedio de las corrientes y voltajes obtenidos como se puede observar en la Tabla 4.2.

Tabla 4.2. Resultados de los promedios de corriente, voltaje y potencia.

Promedio de Corriente	Promedio de Voltaje	Potencia Promedio
191.2 mA	5.25 V	1.0038 W

Es sistema esclavo del procesador digital cuenta con un consumo de potencia de 1.0038 W, en donde la mayor parte se presenta debido al FPGA. El consumo del transceptor y el sensor es de 0.2625 W solamente. El consumo de la potencia del sistema

esclavo se mantiene dentro del rango aceptable, por lo que puede ser alimentado por batería, pero con la desventaja de que su consumo disminuye el tiempo que puede operar o necesita de una batería de alto costo. Un ejemplo de esto es que con una batería de 5V y 1050 mAh duraría, idealmente 5.44 horas en operación. Dependiendo de la aplicación en la que se use, puede ser conveniente o inconveniente.

5 CONCLUSIONES.

Se ha comprobado que el procesador digital inalámbrico de señales inerciales tiene un alcance local de hasta 19 metros, lo cual es útil para el monitoreo de las señales inerciales de máquinas CNC y robots a una distancia considerable. Esta distancia varía dependiendo de los obstáculos que se tengan entre los sistemas esclavo y maestro de la comunicación inalámbrica. El sistema cuenta con la capacidad de cambiar el transceptor de la familia utilizada con el fin de ampliar esta longitud de comunicación aumentando la potencia en el sistema o añadiendo una antena de largo alcance. Además de contar con la portabilidad buscada en el sistema.

De acuerdo con las simulaciones, el sistema alcanza una frecuencia de muestreo máxima de 2150 muestras/segundo. Lo cual es una frecuencia aceptable y superior a la de trabajos investigados, en los cuales se tienen registradas frecuencias de muestreo de hasta 1500 muestras/segundo.

Se presentaron los resultados obtenidos en la medición y estimación de señales inerciales con el fin de corroborar el funcionamiento del procesador digital inalámbrico. En todas las gráficas que se presentaron, se obtuvo el comportamiento estimado para el movimiento en el eje X del sensor. Lo cual avala el correcto funcionamiento del procesador digital de señales inerciales en máquinas CNC. En lo que corresponde a las pruebas realizadas por el robot, se observó que el comportamiento obtenido por el procesador digital corresponde al movimiento generado en el robot, consiguiendo observar el comportamiento de señales inerciales que afectan su funcionamiento, como es el caso de las vibraciones y los jalones. En las señales inerciales obtenidas, se presentan vibraciones con frecuencias muy bajas, estas vibraciones pueden ser eliminadas aumentando el orden de los filtros. Este orden se modifica en el sistema digital de manera fácil y con un consumo de recursos bajo. En caso de que el orden sea muy grande, la estructura de estos filtros en el sistema digital permite emigrar a un FPGA con mayores características y con un aumento económico menor. El consumo de la potencia del sistema esclavo se mantiene dentro del rango aceptable, por lo que puede ser alimentado por batería, pero con la desventaja de que su consumo disminuye el tiempo que puede operar o hace necesario añadir una batería con un costo alto.

6 BIBLIOGRAFÍA.

1. Badri, A.E. 2010. A typical filter design to improve the measured signals from MEMS accelerometer. *Measurement*. 43. 1425-1430.
2. Chourasia, V.S.; Tiwari, A.K. Wireless data acquisition system for fetal phonocardiographic signals using bluetooth. *Int. J. Comput. Healthc.* 2012, 1, 240–253.
3. Gad-el-Hak, M. 2001. *The MEMS Handbook Mechanical and Aerospace Engineering Series*. CRC Press.
4. García López, J. 2000. *Procesamiento digital de señales. Serie libros electrónicos*. Mexico, DF.
5. Gaydou, D. 2011. Filtro complementario para estimación de actitud aplicado al controlador embebido de un cuatrirrotor. *Simposio Argentino de Sistemas Embebidos. SASE*.
6. Giancoli, C.D. 2006. *Física.Principios con aplicaciones*. Pearson educación. Sexta ed. pp. 848.
7. Groves, P.D. 2013. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems (Segunda Edición)*. Artech House.
8. Higgins W. 2007. A comparison of complementary and Kalman filtering. *Aerospace and Electronic Systems. IEEE Transactions on*. 3. 321–325.
9. Hunn, Nick. 2010. *Essentials of Short Range Wireless Standards*. Cambridge University Press. Primera ed._

10. Jong-Woong, P. 2013. Development of a Wireless Displacement Measurement System Using Acceleration Responses. *Sensors*. 13.8377-8392.
11. Khurmi, R. S, Gupta, J. K. 2008. *Theory of Machines*. Eurasia Publishing House.
12. Martins, A.A. 2007. The Connection between Inertial Forces and the Vector Potential. *AIP Conference Proceedings*. 880. 1189-1201.
13. Moreno-Tapia, S. V. 2010. A Field Programmable Gate Array-Based Reconfigurable Smart-Sensor Network for Wireless Monitoring of New Generation Computer Numerically Controlled Machines. *Sensors*. 10. 7263-7286.
14. Mota-Valtierra G.C. 2010. ANN Based Tool Condition Monitoring System for CNC Milling Machines. *Ingeniería Investigación y Tecnología*. 12. 461-468.
15. Nordic Semiconductor. 2007. nRF24L01. Single Chip 2.4GHz Transceiver. Product Specification. www.nordicsemi.com.
16. Ramamurthy, H. 2007. Wireless Industrial Monitoring and Control using a Smart Sensor Platform. *Sensors Journal, IEEE*. 7. 611-618.
17. Rangel-Magdaleno, J.J. 2010. FPGA-Based Vibration Analyzer for Continuous CNC Machinery Monitoring With Fused FFT-DWT Signal Processing. *IEEE Transactions on instrumentation and measurement*. 59. 3184-3194
18. Rodriguez-Donate, C. 2010. FPGA-Based Fused Smart Sensor for Dynamic and Vibration Parameter Extraction in Industrial Robot Links. *Sensors*. 10. 4114-4129.
19. Sese-Sanchez, L.M. 2013. *Cálculo numérico y estadística aplicada*. UNED.

20. Shekdar, P. V. and S. R. Hirekhan. 2012. Implementation of Wireless Smart Sensor Platform for Industrial Application. IJERT. 1.
21. Stauffer, P. 2012. State estimation on flexible robots using accelerometers and angular rate sensors. Mechatronics. 22. 1043-1049.
22. Tana, H. 2010. Efficient FPGA implementation of a wireless communication system using Bluetooth connectivity. Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. 1767-1770.
23. Vera-Salas L.A. 2010. Sistema de medición y análisis de vibraciones inalámbrico basado en FPGA, aplicado a una máquina ensambladora de componentes de chip. Universidad Autónoma de Querétaro, Facultad de ingeniería.

APÉNDICE A. Productos obtenidos.

1.- Artículo publicado en congreso CIINDET 2014 de la IEEE. Sección Morelos.

Sistema digital de comunicación inalámbrica usando el transceptor nRF24L01. Id. 319.



Sistema digital de comunicación inalámbrica usando el transceptor nRF24L01.

J. L. Contreras-Hernández, R.A. Osornio-Rios, R. J. Romero-Troncoso, L. Morales-Velázquez, A. García-Pérez.

Resumen:

El sistema de comunicación inalámbrico presentado usa el transceptor nRF24L01 para realizar una comunicación punto a punto. Este cuenta con un consumo bajo de corriente y puede enviar hasta 32 bytes en un paquete de transmisión. El sistema se describe en el Lenguaje para Descripción de Hardware (HDL) y se implementa en un Arreglo de Compuertas Programable en Campo (FPGA) de la familia Xilinx SPARTAN-3. El sistema de comunicación inalámbrica se encuentra dividido en dos partes, el sistema de transmisión y el sistema de recepción. El sistema de transmisión se encarga de configurar al transceptor para enviar el dato deseado. El sistema de recepción lee el dato recibido cuando el transceptor le indica y lo almacena en memoria para enviarlo por RS-232 a la computadora. Para comprobar la correcta comunicación del sistema, se genera una señal senoidal discreta con valores binarios. Esta señal se envía inalámbricamente para su comparación con la señal de origen. Este trabajo presenta las características y ventajas del sistema digital de comunicación inalámbrica utilizando el transceptor nRF24L01.

Palabras Clave: Comunicación inalámbrica, transceptor nRF24L01, sistema inalámbrico.

Abstract:

This paper presents a wireless communication system which uses the transceiver nRF24L01 to perform a point-to-point communication. It has low current consumption and sends up to 32 bytes in a transmission packet. The Hardware Description Language (HDL) describes and implements the system in a Field

Programmable Gate Array (FPGA) Xilinx SPARTAN-3. The wireless communication system is divided into two parts, transmission system and reception system. The transmission system sets the transceiver to send the desired data. The reception system reads the data received when the transceiver indicates it and saves data in a memory to send it through an RS-232 protocol to a CPU. This system generates a discrete sinusoidal signal with binary values to check right communication. The wireless communication system sends this signal to compare it with the original.

This work presents the features and advantages of a wireless communication digital system using a nRF24L01 transceiver.

Keywords: Wireless communication, transceiver nRF24L01, wireless system..

Introducción

La comunicación inalámbrica es la transferencia de información entre dos o más puntos que no se encuentran conectados entre sí por un conductor eléctrico. En los últimos 15 años han aparecido un gran número de estándares de comunicación inalámbrica a corta distancia, con lo que ha habido una revolución en el desarrollo y aceptación de productos móviles. Esto ha ocurrido por el deseo de obtener productos portables y por su globalización. No todos estos estándares han sobrevivido, pero muchos se han vuelto muy famosos como el Bluetooth y Wi-Fi, al grado de tener billones de dispositivos que los utilizan.

Trabajos Relacionados

La industria ha comenzado a explorar las ventajas de la comunicación inalámbrica y de los tipos de estándares adecuados para sus necesidades. Existen muy diversas aplicaciones para la comunicación inalámbrica, desde el monitoreo con una red inalámbrica para cuidados médicos, como los desarrollados en [1] y en [2], hasta el monitoreo de señales en maquinaria industrial como en [3]. La tecnología inalámbrica permite la instalación

Ing. José Luis Contreras Hernández. (jlc.uaq@gmail.com).
Dr. Roque Alfredo Osornio Ríos. (raosornio@hspdigital.org).
Dr. Luis Morales Velázquez. (lmorales@hspdigital.org)
Facultad de Ingeniería, Universidad Autónoma de Querétaro, 76807.
San Juan del Río, Querétaro, México.
Dr. René de Jesús Romero Troncoso. (troncoso@hspdigital.org).
Dr. Arturo García Pérez. (agarcia@hspdigital.org).
División de Ingenierías, Universidad de Guanajuato, 36700.
Salamanca, Guanajuato, México.



de un gran número de sensores en lugares donde la red alámbrica es costosa o poco práctica. Con el monitoreo se recibe una mejor retroalimentación del estado de las máquinas para mejorar su mantenimiento, reducir los tiempos de reparación y los costos [4].

El monitoreo inalámbrico utilizado en la industria se ha desarrollado con transceptores, lo cual facilita en gran medida su manejo. Como se puede ver en [5], se utiliza una plataforma con el transceptor CC2500, utilizando el estándar IEEE 802.15.4. Este transceptor opera de 2.4 a 2.483 GHz y 16 canales. Además de contar con un bajo consumo de corriente, con 900nA en estado apagado, 17mA en estado de recepción y 21.1mA en estado de transmisión. Para realizar la comunicación entre el microprocesador y el transceptor se hace uso del protocolo de interfaz periférica serial (SPI).

Otra aplicación en la industria en la que se hace uso de un transceptor se presenta en [6]. El transceptor utilizado es el MRF 24J40, el cual maneja el estándar IEEE 802.15.4. Con una frecuencia de operación de 2.4 GHz y un consumo de corriente de 2µA en estado apagado, 19mA en estado de recepción y 23 mA en estado de transmisión. Para realizar la configuración y comunicación con el transceptor, es utilizado un FPGA. Cada dato enviado puede variar desde los 12 hasta los 24 bits. La comunicación entre el transceptor y el FPGA utilizado es realizada por medio del protocolo SPI.

El desarrollo de sistemas digitales en FPGA ha tomado mucha popularidad en estos últimos años, esto debido a que permite la reconfiguración del sistema de una manera más fácil, así como fusionar los módulos necesarios para crear un sistema de mayor complejidad sin modificar los que ya se tienen previamente. Para lograr esto, se deben de tomar en cuenta las resoluciones de las señales, por lo que el sistema desarrollado debe de contar con la habilidad de acoplarlas sin efectuar grandes cambios en el sistema. Aunado a esto, se desea tener un consumo de potencia bajo en los sistemas de comunicación inalámbrica, lo cual permiten desarrollar aplicaciones en las que se aproveche al máximo la alimentación.

El trabajo presentado en este artículo es un sistema digital de comunicación inalámbrica descrito en HDL y desarrollado en FPGA Xilinx SPARTAN-3 utilizando el transceptor nRF24L01. Este transceptor cuenta con 126 canales de comunicación de radio frecuencia con una separación de 1MHz entre cada uno. Por lo que el

transceptor trabaja en la banda de frecuencias que van desde los 2.4 a los 2.525 GHz. En México, la banda de frecuencia permitida para el uso del WiFi es de 2.4 a 2.483 GHz. Por lo que el transceptor puede trabajar en los primeros 84 canales. Para la comunicación entre transceptores, se utiliza el protocolo embebido (ShockBurst™), diseñado para aplicaciones de comunicación inalámbrica de ultra bajo consumo de energía y bajo consumo de corriente, con 900nA en estado apagado, 12.3 mA en estado de recepción y 11.3mA en estado de transmisión, las cuales son corrientes tan bajas, que el transceptor puede ser alimentado por medio de una batería por un tiempo mayor. La capacidad del transceptor para enviar desde 1 hasta 32 bytes en un paquete de transmisión, junto con la capacidad de mantener hasta 3 paquetes de recepción en espera para su lectura, permiten que el sistema desarrollado cuente con la capacidad de acoplarse con las señales de otros sistemas por la alta resolución que maneja, además de trabajar de forma pipe-line, reduciendo su tiempo de procesamiento [7].

Interfaz periférica serial

El bus SPI es un estándar síncrono introducido por Motorola de comunicación full duplex que fue adoptado por la industria. Este estándar tiene una interfaz Maestro-Esclavo y soporta hasta 10 Mbps de velocidad. Este protocolo especifica cuatro líneas de datos y control como se puede ver en la figura 1.

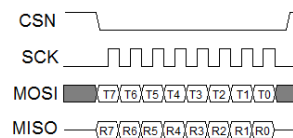


Fig. 1 Líneas de datos y control del estándar SPI.

La señal en la parte superior de la figura 1 corresponde a *Chip Select* (CSN), esta señal es generada por el maestro para seleccionar el esclavo con el cual se ha de comunicar. La siguiente señal es *Serial Clock* (SCK), esta es generada por el maestro para sincronizar los datos de transferencia entre el maestro y el esclavo. La tercera señal es *Master Out, Slave In* (MOSI), es generada por el maestro y recibida por el esclavo, en ésta se colocan los datos a transmitir iniciando con el bit más significativo. La última señal es *Master In, Slave Out* (MISO), esta es generada por el esclavo y



recibida por el maestro, el primer bit que recibe corresponde al más significativo [3].

Transceptor nRF24L01

Un transceptor es un dispositivo que cuenta con un transmisor y un receptor que comparten parte de la circuitería. Debido a esto, un transceptor es manejado como una comunicación semidúplex, es decir, que pueden enviarse señales en ambos sentidos, pero no simultáneamente. Este transceptor permite la comunicación entre diferentes puntos en un canal de frecuencia por medio de una recepción de datos secuencial. El transceptor es configurado y operado utilizando la interfaz SPI. Haciendo uso de este estándar, se pueden manejar las configuraciones de los registros y acceder a todos los modos de operaciones del chip. Además de los parámetros como son la frecuencia del canal, la potencia de salida y la frecuencia de envío de datos, que es de hasta 2 Mbps. La operación de escritura para los registros del transceptor es presentada en la figura 2.

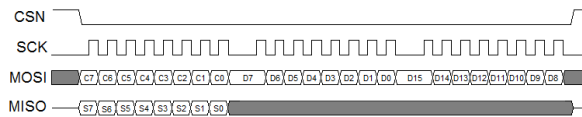


Fig. 2 Operación de escritura para el transceptor.

En la figura 2 se puede observar la señal CSN, con la que se habilita al transceptor para comenzar la comunicación. La señal SCK es el reloj que controla el envío y recepción de los datos, dependiendo el caso en el que se encuentre. La señal MOSI se compone por tres secciones. La primera sección cuenta con ocho bits de comando que envían la dirección del registro a modificar. La segunda sección es de ocho bits y en ésta se envía el byte menos significativo del dato que se requiere al registro seleccionado en la primera sección de la trama. La tercera sección cuenta con el byte más significativo que se requiere enviar al registro deseado. En la señal MISO, de acuerdo a su configuración, se recibe el dato contenido en el registro de STATUS del transceptor.

En la figura 3 se muestran las señales necesarias para la lectura de datos provenientes del transceptor y sus estados. Se cuenta con la señal de activación CSN y la señal de reloj SCK. Para el caso de lectura, en la señal MOSI se envía el comando del registro de lectura. En la

señal MISO se recibe el resultado contenido en el registro STATUS del transceptor.

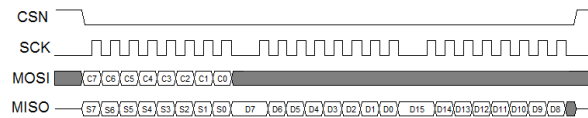


Fig. 3 Operación de lectura para el transceptor.

Una vez configurado el transceptor, la comunicación inalámbrica se realiza por medio de un protocolo propio de la empresa Nordic Semiconductor conocido como ShockBurst™. Este protocolo está basado en un paquete de comunicación y soporta varios modos de la operación manual para protocolos de operación autónomos avanzados. El ShockBurst™ reduce el costo del sistema manejando un enlace de alta velocidad. El transceptor hace uso de la modulación GFSK [7].

Diseño del Sistema

El sistema digital de comunicación inalámbrica se encuentra conformado por los bloques que se observan en la figura 4.

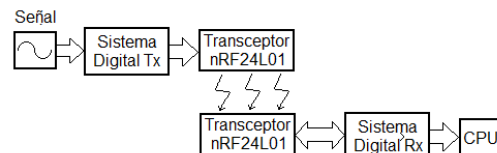


Fig. 4 Diagrama a bloques de sistema digital de comunicación inalámbrica.

El sistema digital de comunicación inalámbrica se encuentra dividido en dos subsistemas: El sistema digital de transmisión Tx y el sistema digital de recepción Rx. Ambos sistemas se encuentran descritos en HDL e implementados en FPGAs de la familia Xilinx SPARTAN 3.

El diagrama a bloques del módulo del sistema digital Tx se muestra en la figura 5.

En este sistema se tiene un módulo que genera el protocolo SPI necesario para la comunicación con el transceptor. El bloque WRITE se encarga de controlar el número de tramas enviadas por el módulo SPI para escribir en los registros del transceptor, con el fin de configurarlo y para el envío de datos de la señal. Los comandos y los datos correspondientes a la configuración del transceptor se encuentran guardados



en la memoria ROM, el contador con habilitación Tx_Counter proporciona su dirección. La máquina de estados Tx_FSM controla el proceso de configuración del transceptor y el envío de la señal.

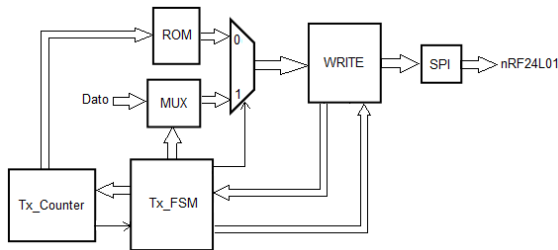


Fig. 5 Módulo del sistema digital de transmisión Tx.

Finalizada la configuración, el módulo del sistema digital de transmisión Tx pasa al modo de envío del dato de la señal con un multiplexor. El módulo MUX contiene el comando necesario para realizar la transmisión, así como el dato de la señal que se desea enviar. Una vez finalizado el envío, el módulo del sistema digital de transmisión Tx entra en modo de espera, hasta que se obtenga el siguiente dato de la señal a enviar.

El módulo del sistema digital de recepción Rx es un poco más complejo, debido a que realiza la configuración del transceptor y pasa al modo de espera de recepción, en el cual, el sistema lee la información del puerto transceptor, cuando éste le indica que ha recibido un dato. En la figura 6 se muestra el diagrama a bloques del módulo del sistema digital de recepción Rx.

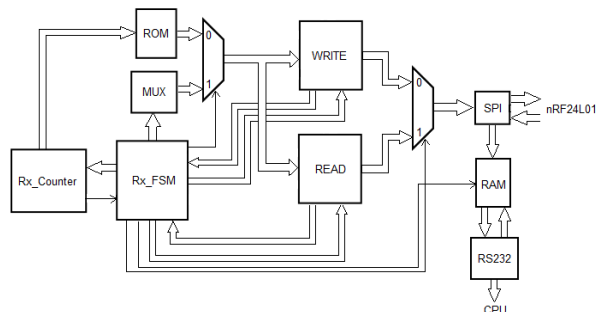


Fig. 6 Módulo del sistema digital de recepción Rx.

Este sistema de recepción cuenta con los mismos módulos de WRITE, ROM y Rx_Counter que el

sistema de transmisión para la configuración del Transceptor. Una vez terminada la configuración, el sistema cambia a modo de recepción, por lo que son necesarios los multiplexores que se muestran en el diagrama a bloques. En el módulo MUX se encuentran los comandos necesarios para realizar la recepción de datos. El módulo READ controla el SPI para enviar el comando que ordena al transceptor la descarga de los datos recibidos. Estos datos son almacenados en una memoria RAM, para posteriormente ser enviado hacia la computadora y analizar sus gráficas. Esta comunicación se realiza con un módulo que maneja el protocolo RS-232. La máquina de estados del receptor Rx_FSM controla todo el sistema de recepción.

La fusión del sistema de transmisión y el sistema de recepción genera el sistema de comunicación inalámbrica completo. Teniendo la ventaja de configurar ambos a voluntad del usuario, sin necesidad de introducirse en el código, solamente se realizan los cambios deseados directo en las memorias ROM de los sistemas.

Experimentación y Resultados

Para confirmar el correcto funcionamiento del sistema de comunicación inalámbrica, se generó una señal senoidal discreta en descripción HDL mostrada en la gráfica de la figura 7. Esta señal se encuentra formada por 99 muestras. La amplitud de la señal es desde -127 hasta 127 representado con 8 bits. La comunicación inalámbrica se desarrolló con 32 bytes, que es la resolución máxima que el transceptor es capaz de enviar y recibir.

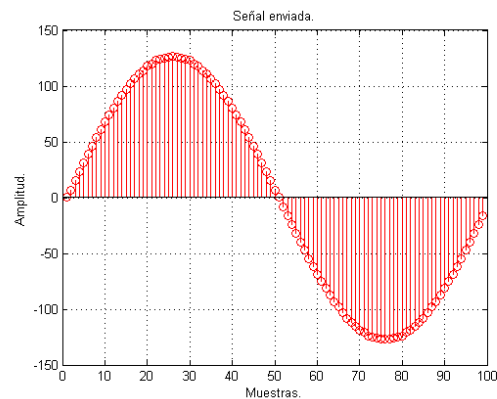


Fig. 7 Señal generada en HDL para su envío.



La señal se desarrolló en una Tabla de Consulta (LUT) en descripción HDL, añadida al sistema de transmisión para efectuar las pruebas.

Se ejecutaron veinte repeticiones en el envío de la señal inalámbricamente.

La señal recibida se muestra en la figura 8. Como se puede observar, la recepción de la señal se realiza de una forma correcta ya que corresponde con la señal transmitida. Las pruebas se realizaron con diferentes distancias, evaluando un alcance máximo de 70m a 100m en aéreas completamente despejadas.

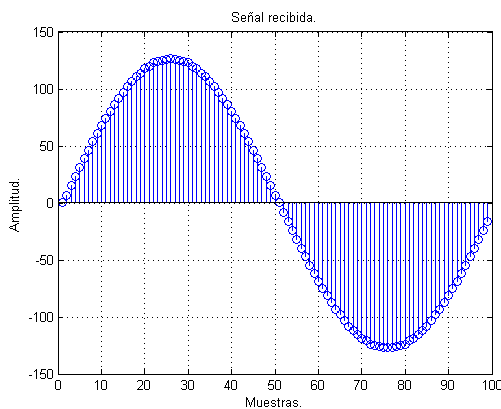


Fig. 8 Señal recibida inalámbricamente

Es importante el análisis de los recursos utilizados en el desarrollo del sistema digital de transmisión, éstos son presentados en la tabla 1.

Tabla 1: Recursos utilizados en el sistema de transmisión en tarjeta Xilinx Spartan-3 FPGA: XC3S200

Elemento	Usado	Disponible	Porcentaje %
Slice Flip Flops	76	3,840	1%
4 input LUTs	280	3840	7%
Slices	145	1920	7%
Block RAMs	0	12	0%
Multipliers	0	12	0%

Como se puede observar en la tabla 1, el uso de los *Slice Flip Flops* es muy bajo, usando apenas el 1% del total disponible. Para las LUTs y los *Slices* ambos usan el 7%. Mientras que los bloque de RAM y *Multipliers* no son usados. Gracias a esto, el módulo del sistema digital de transmisión Tx puede ser acoplado a otros sistemas de procesamiento digital de señales

desarrollados en HDL para contar con la comunicación inalámbrica sin un consumo excesivo de recursos.

Los recursos utilizados en el desarrollo del sistema digital de recepción son presentados en la tabla 2.

Como se puede ver en la Tabla 2, este sistema se ve afectado en el consumo de recursos debido a que contiene los bloques necesarios para la configuración del transceptor. A pesar de esto, los recursos utilizados siguen siendo reducidos con 3% de *Slice Flip Flops*, 12% de LUTs y 13% de *Slices*. Los bloques de RAM y *Multipliers* no son usados. Por esto, el sistema de recepción permite su acoplamiento en otros sistemas digitales de procesamiento desarrollados en HDL o como en este trabajo, la adición de un módulo que genere algún protocolo para el envío de los datos recibidos hacia un equipo para su análisis.

Tabla 2: Recursos utilizados en el sistema de recepción en tarjeta Xilinx Spartan-3 FPGA: XC3S200

Elemento	Usado	Disponible	Porcentaje %
Slice Flip Flops	136	3,840	3%
4 input LUTs	479	3840	12%
Slices	263	1920	13%
Block RAMs	0	12	0%
Multipliers	0	12	0%

El sistema digital de comunicación inalámbrica es una herramienta de bajo consumo, permitiendo una gran adaptabilidad a sistemas de procesamiento de señales ya desarrollados. Aportándoles la movilidad que presenta este sistema.

Conclusión

El sistema digital de comunicación inalámbrica usando el transceptor nRF24L01 es una herramienta de fácil configuración. Con una distancia de comunicación perfecta para espacios en los que se requiere monitoreo de maquinaria o de sistemas que se encuentren cercanos, así como los que necesitan tener una mayor movilidad y en los que la instalación de cableado puede llegar a obstruir su manejo. Debido a su reducido tamaño, puede adaptarse a las maquinarias sin la necesidad de realizar modificaciones mayores. El sistema cuenta con la capacidad del transceptor para enviar una gran cantidad de bytes de datos, esto permite la adaptabilidad del sistema inalámbrico con otros ya existentes, sin la limitante de enviar datos seccionados en paquetes o no contar con la cantidad de bits



requerida para la lectura de un dato. El uso bajo de recursos en el FPGA, permite la fusión de otros sistemas en un mismo dispositivo. Debido al desarrollo en la actualidad de sistemas de bajo consumo de corriente, el sistema de comunicación inalámbrica es una herramienta que permite llevar a cabo su tarea con alimentación de baterías, utilizando FPGAs de bajo consumo.

Referencias

- H. Tana, "Efficient FPGA implementation of a wireless communication system using Bluetooth connectivity", Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pp. 1767-1770, 2010.
- V.S. Chourasia and A.K. Tiwari, "Wireless data acquisition system for fetal phonocardiographic signals using bluetooth", Int. J. Comput. Healthc, vol. 1, pp 240-253, 2012.
- H. Ramamurthy, "Wireless Industrial Monitoring and Control using a Smart Sensor Platform", Sensors Journal, vol. 7, pp. 611-618, 2007.
- N. Hunn, "Essentials of Short Range Wireless Standards", Cambridge University Press, Primera ed. 2010.
- P. V. Shekdar, and S. R. Hirekhan, "Implementation of Wireless Smart Sensor Platform for Industrial Application", IJERT, vol. 1, 2012.
- S. V. Moreno-Tapia, "A Field Programmable Gate Array-Based Reconfigurable Smart-Sensor Network for Wireless Monitoring of New Generation Computer Numerically Controlled Machines", Sensors, vol 10, pp. 7263-7286, 2010.
- Nordic Semiconductor, "nRF24L01. Single Chip 2.4GHz Transceiver. Product Specification", www.nordicsemi.com. 2007.

Ing. José Luis Contreras Hernández.

Recibió el grado de Ingeniero por parte de la Universidad de Guanajuato, Salamanca, México, en 2012. En la actualidad se encuentra en la Universidad Autónoma de Querétaro. Su área de interés es el procesamiento digital de señales para aplicaciones en mecatrónica.

Dr. Roque Alfredo Osornio Ríos.

Recibió el grado de Ingeniero por parte del Instituto Tecnológico de Querétaro, Querétaro, México. Y los grados de Maestro en ingeniería y Doctor por parte de la Universidad de Querétaro, Querétaro, México, en 2007. Es un investigador nacional CONACYT. En la

actualidad es profesor titular en la Universidad de Querétaro. Ha sido asesor de más de 30 tesis y coautor de más de 40 artículos en revistas internacionales y conferencias. Sus áreas de interés son el procesamiento de señales en hardware y mecatrónica.

Dr. René de Jesús Romero Troncoso.

Recibió el grado de Ingeniero y de Maestro por parte de la Universidad de Guanajuato, Salamanca, México. Y el grado de Doctor por honores por parte de la Universidad Autónoma de Querétaro, Querétaro, México, en 2004. Es un investigador nacional nivel 2 CONACYT. En la actualidad es profesor titular en la Universidad de Guanajuato y es un investigador invitado de la Universidad Autónoma de Querétaro. Ha sido asesor de más de 180 tesis, autor de dos libros en sistemas digitales y coautor de más de 80 artículos en revistas internacionales y conferencias. Sus áreas de interés son el procesamiento de señales en hardware y mecatrónica.

Dr. Luis Morales Velázquez.

Recibió el grado de Doctor de la Universidad de Querétaro, Querétaro, México. Es reconocido por CONACYT como investigador. En la actualidad es profesor titular en la Universidad de Querétaro. Ha sido asesor de varias tesis. Sus áreas de interés son el procesamiento de señales en hardware, Mecatrónica y el desarrollo de plataformas de procesamiento de señales en tiempo real basado en FPGA.

Dr. Arturo García Pérez.

Recibió el grado de Ingeniero y Maestro por parte de la universidad de Guanajuato, Salamanca, México en 1992 y 1994 respectivamente. Y el grado de Doctor por parte de la Universidad de Texas, Dallas, E.U.A, en 2005. En la actualidad es profesor titular en la Universidad de Guanajuato. Ha sido asesor de más de 40 tesis. Sus áreas de interés son el procesamiento digital de señales para aplicaciones en mecatrónica.

APÉNDICE B. Código de manejador nRF24L01.

B.1 nRF_TxRx.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
entity nRF_TxRx is
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
MISO : in std_logic;
IRQ : in std_logic;
ACT : in std_logic;
WRD : in std_logic;
WDir : in std_logic;
Din : in std_logic_vector(7 downto 0);
Dir_a : in std_logic_vector(39 downto 0);
-- Outputs
RDY : out std_logic;
SCK : out std_logic;
CSN : out std_logic;
CE : out std_logic;
MOSI : out std_logic;
H_RTx : out std_logic_vector(1 downto 0);
H_RRx : out std_logic_vector(1 downto 0);
Dout : out std_logic_vector(7 downto 0);
Q3 : out std_logic_vector(2 downto 0);
Q0 : out std_logic_vector(3 downto 0);
Q1,Q2 : out std_logic_vector(4 downto 0);
Q4,Qm : out std_logic_vector(4 downto 0)
);
end nRF_TxRx;
architecture Main_TxRx of nRF_TxRx is
component SPI
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
STT : in std_logic;
MISO : in std_logic;
DTx : in std_logic_vector(7 downto 0);
-- Outputs
EOT : out std_logic;
SCK : out std_logic;
MOSI : out std_logic;
DRx : out std_logic_vector(7 downto 0)
);
end component;
component FSM
```

```

port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
CSNp : in std_logic;
ACT : in std_logic;
IRQ : in std_logic;
WRD : in std_logic;
EOT : in std_logic;
WDir : in std_logic;
-- Outputs
RDY : out std_logic;
CE : out std_logic;
STT : out std_logic;
HRe1 : out std_logic_vector(1 downto 0);
HRe2 : out std_logic_vector(1 downto 0);
H_RDr : out std_logic_vector(1 downto 0);
MU : out std_logic_vector(3 downto 0);
ADDR : out std_logic_vector(5 downto 0);
Q3 : out std_logic_vector(2 downto 0);
Q0 : out std_logic_vector(3 downto 0);
Q1,Q2 : out std_logic_vector(4 downto 0);
Q4,Qm : out std_logic_vector(4 downto 0)
);
end component;
component ROM_Cfg
port(
--Inputs
ADDR : in std_logic_vector(5 downto 0);
--Outputs
D : out std_logic_vector(8 downto 0)
);
end component;
component Multiplex
port(
-- Inputs
Mu : in std_logic_vector(3 downto 0);
D_Re : in std_logic_vector(7 downto 0);
Dire : in std_logic_vector(7 downto 0);
DROM : in std_logic_vector(8 downto 0);
-- Outputs
CSN : out std_logic;
D_Tx : out std_logic_vector(7 downto 0)
);
end component;
component R_IRQ
port(
--Inputs
CLK : in std_logic;
D : in std_logic;
--Outputs
Q : out std_logic
);
end component;
component RD40_8

```

```

port(
--Inputs
RST : in std_logic;
CLK : in std_logic;
D : in std_logic_vector(39 downto 0);
H : in std_logic_vector(1 downto 0);
--Outputs
S : out std_logic_vector(7 downto 0)
);
end component;
signal EOT,CSNp : std_logic;
signal STT,IRQD : std_logic;
signal H_RDr : std_logic_vector(1 downto 0);
signal Mu : std_logic_vector(3 downto 0);
signal ADDR : std_logic_vector(5 downto 0);
signal DTx,DRx : std_logic_vector(7 downto 0);
signal D_Re,Dire : std_logic_vector(7 downto 0);
signal DROM : std_logic_vector(8 downto 0);
begin
D_Re <= Din;
Dout <= DRx;
CSNp <= DROM(0);
CH_01 : SPI port map(RST,CLK,STT,MISO,DTx,EOT,SCK,MOSI,DRx);
CH_02 : FSM port map(RST,CLK,CSNp,ACT,IRQD,WRD,EOT,WDir,RDY,CE,STT,H_
RTx,H_RRx,H_RDr,Mu,ADDR,Q3,Q0,Q1,Q2,Q4,Qm
);
CH_03 : ROM_Cfg port map(ADDR,DROM);
CH_04 : Multiplex port map(Mu,D_Re,Dire,DROM,CSN,DTx);
CH_05 : RD40_8 port map(RST,CLK,Dir_a,H_RDr,Dire);
CH_06 : R_IRQ port map(CLK,IRQ,IRQD);
end Main_TxRx;

```

B.2 SPI.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
entity SPI is
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
STT : in std_logic;
MISO : in std_logic;
DTx : in std_logic_vector(7 downto 0);
-- Outputs
EOT : out std_logic;
SCK : out std_logic;
MOSI : out std_logic;
DRx : out std_logic_vector(7 downto 0)
);
end SPI;
architecture Main_SPI of SPI is
component SPI_FSM
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
H : in std_logic;
EOC : in std_logic;
-- Outputs
EOT : out std_logic;
H_S : out std_logic;
Mu : out std_logic_vector(3 downto 0)
);
end component;
component SPI_CLK
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
H_SCK : in std_logic;
-- Outputs
SCK : out std_logic
);
end component;
component SPI_FlgCn
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
H_SCK : in std_logic;
-- Outputs
EOC : out std_logic;
RDC : out std_logic
);
```

```

end component;
component SPI_Desp
port(
--Inputs
RST : in std_logic;
CLK : in std_logic;
MISO : in std_logic;
RDC : in std_logic;
--Outputs
D : out std_logic_vector(7 downto 0)
);
end component;
component SPI_Mux
port(
-- Inputs
Din : in std_logic_vector(7 downto 0);
H_MUX : in std_logic_vector(3 downto 0);
-- Outputs
Dout : out std_logic
);
end component;
signal EOC,H_SCK : std_logic;
signal RDC : std_logic;
signal Mu : std_logic_vector(3 downto 0);
begin
JL_01 : SPI_FSM port map (RST,CLK,STT,EOC,EOT,H_SCK,Mu);
JL_02 : SPI_CLK port map (RST,CLK,H_SCK,SCK);
JL_03 : SPI_FlgCn port map (RST,CLK,H_SCK,EOC,RDC);
JL_04 : SPI_Desp port map (RST,CLK,MISO,RDC,DRx);
JL_05 : SPI_Mux port map (DTx,Mu,MOSI);
end Main_SPI;

```

B.3 FSM.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
entity FSM is
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
CSNp : in std_logic;
ACT : in std_logic;
IRQ : in std_logic;
WRD : in std_logic;
EOT : in std_logic;
WDir : in std_logic;
-- Outputs
RDY : out std_logic;
CE : out std_logic;
STT : out std_logic;
HRe1 : out std_logic_vector(1 downto 0);
HRe2 : out std_logic_vector(1 downto 0);
H_RDr : out std_logic_vector(1 downto 0);
MU : out std_logic_vector(3 downto 0);
ADDR : out std_logic_vector(5 downto 0);
Q3 : out std_logic_vector(2 downto 0);
Q0 : out std_logic_vector(3 downto 0);
Q1,Q2 : out std_logic_vector(4 downto 0);
Q4,Qm : out std_logic_vector(4 downto 0)
);
end FSM;
architecture Main_FSM of FSM is
component FSM_MN
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
EORx : in std_logic;
EOTx : in std_logic;
EOCS : in std_logic;
EOCf : in std_logic;
EODr : in std_logic;
WRD : in std_logic;
Hab : in std_logic;
WDir : in std_logic;
-- Outputs
STRx : out std_logic;
STTx : out std_logic;
STCS : out std_logic;
STDr : out std_logic;
H1 : out std_logic;
H2 : out std_logic;
RDY : out std_logic;
Mux : out std_logic_vector(2 downto 0);
S : out std_logic_vector(4 downto 0)
);
```



```

end component;
component FSM_Cfg
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
CSN : in std_logic;
EOT : in std_logic;
EOC : in std_logic;
-- Outputs
CE : out std_logic;
STT : out std_logic;
EOM : out std_logic;
Mux : out std_logic_vector(3 downto 0);
OPC : out std_logic_vector(1 downto 0);
S : out std_logic_vector(3 downto 0)
);
end component;
component FSM_Tx
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
EOT : in std_logic;
EOC : in std_logic;
STM : in std_logic;
IRQ : in std_logic;
StR : in std_logic;
-- Outputs
CE : out std_logic;
STT : out std_logic;
EOM : out std_logic;
HRe : out std_logic_vector(1 downto 0);
OPC : out std_logic_vector(1 downto 0);
Mux : out std_logic_vector(3 downto 0);
S : out std_logic_vector(4 downto 0)
);
end component;
component FSM_Rx
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
EOT : in std_logic;
EOC : in std_logic;
STM : in std_logic;
IRQ : in std_logic;
StR : in std_logic;
-- Outputs
CE : out std_logic;
STT : out std_logic;
EOM : out std_logic;
HRe : out std_logic_vector(1 downto 0);
OPC : out std_logic_vector(1 downto 0);
Mux : out std_logic_vector(3 downto 0);

```

```

S : out std_logic_vector(4 downto 0)
);
end component;
component FSM_CS
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
EOT : in std_logic;
STM : in std_logic;
-- Outputs
CE : out std_logic;
STT : out std_logic;
EOM : out std_logic;
Mux : out std_logic_vector(3 downto 0);
S : out std_logic_vector(2 downto 0)
);
end component;
component FSM_Dir
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
EOT : in std_logic;
EOC : in std_logic;
STM : in std_logic;
-- Outputs
CE : out std_logic;
STT : out std_logic;
EOM : out std_logic;
Mux : out std_logic_vector(3 downto 0);
Hre : out std_logic_vector(1 downto 0);
Hco : out std_logic_vector(1 downto 0);
S : out std_logic_vector(4 downto 0)
);
end component;
component Counter
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
OPC : in std_logic_vector(1 downto 0);
-- Outputs
SAL : out std_logic_vector(5 downto 0);
EOC : out std_logic
);
end component;
component CounterD
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
OPC : in std_logic_vector(1 downto 0);
-- Outputs
EOC : out std_logic

```

```

);
end component;
component CountDr
port(
-- Inputs
RST : in std_logic;
CLK : in std_logic;
OPC : in std_logic_vector(1 downto 0);
-- Outputs
EOC : out std_logic
);
end component;
component Mux_FSM
port(
-- Inputs
CE0 : in std_logic;
CE1 : in std_logic;
CE2 : in std_logic;
CE3 : in std_logic;
CE4 : in std_logic;
STT0 : in std_logic;
STT1 : in std_logic;
STT2 : in std_logic;
STT3 : in std_logic;
STT4 : in std_logic;
H_MUX : in std_logic_vector(2 downto 0);
M0 : in std_logic_vector(3 downto 0);
M1 : in std_logic_vector(3 downto 0);
M2 : in std_logic_vector(3 downto 0);
M3 : in std_logic_vector(3 downto 0);
M4 : in std_logic_vector(3 downto 0);
-- Outputs
CE : out std_logic;
STT : out std_logic;
MU : out std_logic_vector(3 downto 0)
);
end component;
component R_OP
port(
--Inputs
RST : in std_logic;
CLK : in std_logic;
D : in std_logic;
H : in std_logic;
--Outputs
S : out std_logic
);
end component;
signal EORx,EOTx,EOCS,EODr : std_logic;
signal EOCf,H1,H2 : std_logic;
signal STRx,STTx,STCS,STDr : std_logic;
signal StR : std_logic;
signal EOC0,EOC1,EOC2,EOC4 : std_logic;
signal STT0,STT1,STT2,STT4 : std_logic;
signal STT3,S1,S2 : std_logic;

```

```

signal CE0,CE1,CE2,CE3,CE4 : std_logic;
signal OPC0,OPC1,OPC2,OPC4 : std_logic_vector(1 downto 0);
signal M0,M1,M2,M3,M4 : std_logic_vector(3 downto 0);
signal H_Mux : std_logic_vector(2 downto 0);
begin
FSM_01 : FSM_MN port map(RST,CLK,EORx,EOTx,EOCS,EOCf,EODr,WRD,ACT,WDir,S
TRx,STTx,STCS,STDr,H1,H2,RDY,H_Mux,Qm
);
FSM_02 : FSM_Cfg port map(RST,CLK,CSNp,EOT,EOC0,CE0,STT0,EOCf,M0,OPC0,Q0)
;
FSM_03 : FSM_Tx port map(RST,CLK,EOT,EOC1,STTx,IRQ,StR,CE1,STT1,EOTx,HRe
1,OPC1,M1,Q1
);
FSM_04 : FSM_Rx port map(RST,CLK,EOT,EOC2,STRx,IRQ,StR,CE2,STT2,EORx,HRe
2,OPC2,M2,Q2
);
FSM_05 : FSM_CS port map(RST,CLK,EOT,STCS,CE3,STT3,EOCS,M3,Q3);
FSM_06 : FSM_Dir port map(RST,CLK,EOT,EOC4,STDr,CE4,STT4,EODr,M4,H_RDr,OP
C4,Q4
);
FSM_07 : Counter port map(RST,CLK,OPC0,ADDR,EOC0);
FSM_08 : CounterD port map(RST,CLK,OPC1,EOC1);
FSM_09 : CounterD port map(RST,CLK,OPC2,EOC2);
FSM_10 : CountDr port map(RST,CLK,OPC4,EOC4);
FSM_11 : Mux_FSM port map(CE0,CE1,CE2,CE3,CE4,STT0,STT1,STT2,STT3,STT4,H_
Mux,M0,M1,M2,M3,M4,CE,STT,MU
);
FSM_12 : R_OP port map(RST,CLK,WRD,H1,S1);
FSM_13 : R_OP port map(RST,CLK,S1,H2,S2);
StR <= S1 xnor S2;
end Main_FSM;

```

B.4 ROM_Cfg.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity ROM_Cfg is
port(
--Inputs
ADDR : in std_logic_vector(5 downto 0);
--Outputs
D : out std_logic_vector(8 downto 0)
);
end ROM_Cfg;
architecture Cfg_ROM of ROM_Cfg is
type vector_array is array (63 downto 0) of std_logic_vector (8 downto 0);
constant ROM_v : vector_array :=(
"11111111"&'1', -- CONFIG
"00100000"&'0',
"01011110"&'0',
"11111111"&'1', -- EN_AA
"00100001"&'0',
"00000000"&'0',
"11111111"&'1', -- EN_RXADDR
"00100010"&'0',
"00000001"&'0',
"11111111"&'1', -- SETUP_AW
"00100011"&'0',
"00000011"&'0',
"11111111"&'1', -- SETUP_RETR
"00100100"&'0',
"00000000"&'0',
"11111111"&'1', -- RF_CH
"00100101"&'0',
"01010011"&'0',
"11111111"&'1', -- RF_SETUP
"00100110"&'0',
"00001110"&'0',
"11111111"&'1', -- RX_ADDR_P0
"00101010"&'0',
"10110011"&'0',
"10110100"&'0',
"10110101"&'0',
"10110110"&'0',
"11110001"&'0',
"11111111"&'1', -- TX_ADDR
"00110000"&'0',
"10110011"&'0',
"10110100"&'0',
"10110101"&'0',
"10110110"&'0',
"11110001"&'0',
"11111111"&'1', -- RX_PW_P0
"00110001"&'0',
"00001100"&'0',
"11111111"&'1',
```

