

Control inalámbrico basado en sensores de aceleración aplicado a la asistencia de discapacitados

Tesis de Grado de Ingeniería Electrónica

Mauro Andrés Burger

Departamento de Electrónica
Facultad de Ingeniería
Universidad Nacional de Mar del Plata
E-mail: mauro.burger@fi.mdp.edu.ar

Director: Jorge Castiñeira Moreira
Co-Director: Pablo Daniel Agüero

Diciembre 2014



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Agradecimientos

A mis padres, Liliana y Alejandro, por el apoyo, la ayuda y la paciencia brindadas durante todos estos años, sin los cuales no hubiera sido posible lograr este gran objetivo. A mi hermano Javier y su esposa Melisa, los cuales me han brindado su ayuda y sus recomendaciones siempre que las necesité, y a mi tío Mario, hasta ahora el único ingeniero de la familia, que con su incomparable insistencia ha servido de ejemplo y motivación para nunca darse por vencido.

A mis compañeros de estudio con los que compartí incontables horas durante mi paso por la universidad, en especial a Santiago Pimentel, Manuel Stillitano, María Cecilia Antelo Gáspari, Mauro Di Vito y Emanuel Sajama, entre otros.

A Jorge Castiñeira por haber propuesto y dirigido este proyecto, a Pablo Agüero por su excelente predisposición para ayudar cada vez que surgió alguna dificultad o inquietud, y a todos los integrantes del Laboratorio de Comunicaciones que me brindaron su ayuda cuando fue necesaria.

Por último y no menos importante, a todos los integrantes de la República Argentina que, afortunadamente, hemos entendido la importancia de la educación pública y gratuita para el desarrollo de una mejor sociedad.

Resumen

Como requisito de la carrera de Ingeniería Electrónica, se desarrolló un Trabajo Final el cual consiste en un sensor que permite comandar distintos dispositivos mediante movimientos de cabeza, con el objetivo de ser utilizado para la asistencia a personas con discapacidades motrices. El sistema está implementado de forma modular, es decir que esta compuesto por diferentes elementos con fines específicos, los cuales se comunican entre sí. El dispositivo principal es un sensor que se coloca en la cabeza del usuario mediante una vincha elástica. Se basa en un sensor de aceleración de tipo capacitivo similar a los que se utilizan en dispositivos móviles de consumo masivo, el cual se comunica con los demás dispositivos de forma inalámbrica a través de ondas de radio. Se planteó un sistema flexible que pueda adaptarse a múltiples aplicaciones, como pueden ser el control de una silla de ruedas, una computadora, un juguete o un televisor.

Inicialmente en el primer capítulo se aborda el concepto de discapacidad, haciendo foco en parálisis y discapacidades motrices. Se estudian las alternativas existentes, comparando ventajas y desventajas de cada sistema, para luego presentar brevemente el método propuesto y como se compara con los otros. Es de destacar que desde el comienzo del proyecto se hizo énfasis en desarrollar un sistema económico, de forma que sea accesible para la mayor cantidad de personas posibles.

En el segundo capítulo se ofrece un marco teórico de todas las tecnologías utilizadas en el proyecto, como el principio de funcionamiento del sensor y la forma en que este se comunica, el transmisor de radiofrecuencia elegido, y el componente central de cada dispositivo que es el microcontrolador.

En el tercer capítulo se presenta en detalle el sistema desarrollado, comenzando por el funcionamiento del sistema en su conjunto y el protocolo de comunicación adoptado que utilizan todos los módulos del proyecto. Luego se encuentra explicado cada dispositivo por separado, incluyendo el circuito electrónico, el algoritmo que ejecuta el programa y el circuito impreso diseñado.

Por último se realiza un repaso de resultados obtenidos, objetivos alcanzados, próximas etapas o mejoras propuestas.

Índice general

Agradecimientos	a
Resumen	b
1. Introducción	1
1.1. Parálisis	1
1.1.1. Causas	2
1.1.2. Clasificación	2
1.2. Objetivo del proyecto	2
1.3. Soluciones actuales	4
1.3.1. Sensores mecánicos	4
1.3.2. Sensores infrarrojos	4
1.3.3. Sensores basados en seguimiento de iris	5
1.3.4. Sensores basados en ondas cerebrales	6
1.4. Método propuesto	6
1.4.1. Funcionamiento	7
1.4.2. Ventajas	7
1.4.3. Desventajas	8
2. Marco Teórico	9
2.1. Sensor de aceleración	9
2.1.1. Principio de funcionamiento	10
2.1.2. Tecnología MEMS	10
2.1.3. Sensor MMA8451	11
2.1.4. Interfaz I2C	13
2.1.5. Opciones de configuración	14
2.2. Enlace de radio frecuencia	16
2.2.1. Tranceptor nRF24L01+	16
2.2.2. Modulación GFSK	17
2.2.3. Interfaz SPI	18
2.2.4. Diagrama de estados	20
2.2.5. Protocolo propietario Enhanced Shockburst™	20
2.3. Microcontroladores	23
2.3.1. Familia PIC18F	23
2.3.2. Lenguaje y entorno de programación	24
2.4. Estándar USB 2.0	25
2.4.1. Velocidades	25
2.4.2. Características eléctricas	26
2.4.3. Protocolo	27
2.4.4. Paquetes	27
2.4.5. Transferencias	28
2.4.6. Clases	30

2.4.7. Microchip USB Device Framework	30
2.5. Radiación Electromagnética	30
2.5.1. Radiación Ionizante y No Ionizante	32
2.5.2. Regulaciones sobre Radiación no Ionizante	32
3. Desarrollo	34
3.1. Sistema propuesto	34
3.1.1. Modos de funcionamiento	34
3.1.2. Diagrama de estados	35
3.2. Protocolo de comunicación	35
3.2.1. Tipos de paquete	35
3.3. Transmisor	38
3.3.1. Circuito esquemático	38
3.3.2. Prototipo experimental	39
3.3.3. Circuito impreso	39
3.3.4. Programa	40
3.4. Receptor MATLAB®	44
3.4.1. Circuito esquemático	45
3.4.2. Prototipo experimental	46
3.4.3. Circuito impreso	46
3.4.4. Programa	47
3.4.5. Interfaz Matlab®	48
3.5. Receptor Mouse	49
3.5.1. Circuito esquemático	49
3.5.2. Prototipo experimental	49
3.5.3. Circuito impreso	49
3.5.4. Programa	51
3.6. Receptor RC	54
3.6.1. Circuito esquemático	54
3.6.2. Circuito impreso	55
3.6.3. Programa	56
3.7. Cálculos de radiación no ionizante	58
Conclusiones	60
A. Circuitos esquemáticos	I
A.1. Transmisor	II
A.2. Receptor MATLAB	III
A.3. Receptor Mouse	IV
A.4. Receptor RC	V
Índice de figuras	VI
Índice de tablas	VII
Bibliografía	VIII

Capítulo 1

Introducción

Para comprender la discapacidad, según el *Informe Mundial Sobre la Discapacidad*, publicado en 2011 por la OMS¹, es necesario analizar tanto el lado médico como el lado social:

El término genérico «discapacidad» abarca todas las deficiencias, las limitaciones para realizar actividades y las restricciones de participación, y se refiere a los aspectos negativos de la interacción entre una persona (que tiene una condición de salud) y los factores contextuales de esa persona (factores ambientales y personales).

La CIF² clasifica en tres categorías los problemas de funcionamiento humano:

- Deficiencias: Son problemas en la función corporal o alteraciones en la estructura corporal; por ejemplo, parálisis o ceguera.
- Limitaciones de la actividad: Son dificultades para realizar actividades; por ejemplo, caminar o comer.
- Restricciones de participación: Son problemas para participar en cualquier ámbito de la vida; por ejemplo, ser objeto de discriminación a la hora de conseguir empleo o transporte.

Se entiende entonces que la discapacidad involucra las dificultades que se presentan en cualquiera de las tres áreas de funcionamiento.

1.1. Parálisis

Parálisis es la pérdida o alteración de la función motora en una parte del cuerpo, debida a una lesión del mecanismo neural o muscular. Cuando se trata de la función motora, se llama parálisis a la pérdida del movimiento voluntario. Según la zona donde se encuentre la lesión, se puede producir paraplejía o cuadriplejía, como se ve en la Figura 1.1. Paraplejía se refiere a debilidad o parálisis de ambas piernas y la parte baja del cuerpo. Cuadriplejía o tetraplejía se refiere a debilidad o parálisis de los cuatro miembros.

¹Organización Mundial de la Salud (OMS): es el organismo de la Organización de las Naciones Unidas (ONU) especializado en gestionar políticas de prevención, promoción e intervención en salud a nivel mundial.

²Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud (CIF).

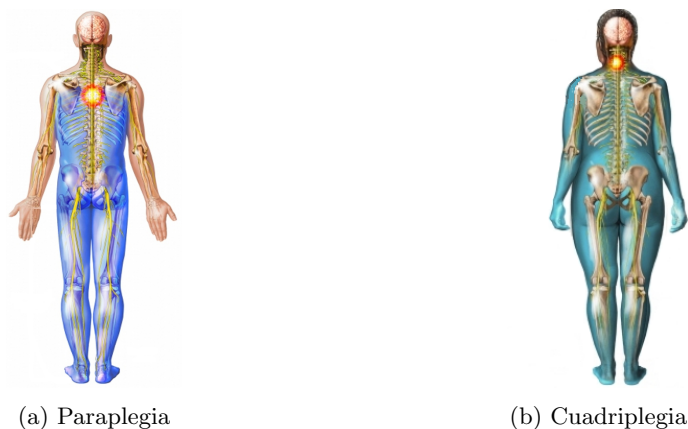


Figura 1.1: Parálisis según la ubicación de la lesión.

1.1.1. Causas

La lesión se produce, usualmente, cuando una zona de la columna vertebral se comprime o se dobla. Se puede producir en los siguientes casos:

- Lesiones durante el parto, en la zona del cuello.
- Caídas y golpes.
- Accidentes de tránsito.
- Accidentes de clavadistas.
- Disparos o puñaladas que atraviesan la médula espinal.

1.1.2. Clasificación

En la Tabla 1.1 se puede observar cómo se clasifican las distintas parálisis según su nivel de gravedad y las capacidades con las que cuenta el paciente.

1.2. Objetivo del proyecto

El objetivo de este proyecto consiste en permitir a las personas que sufran de cuadriplegia pero mantengan el movimiento en su cuello que controlen diversos dispositivos mediante el movimiento de sus cabezas. Los dispositivos pueden ser diversos: controlar el mouse de una computadora, mover una silla de ruedas eléctrica, mover un auto de juguete a radio control, controlar un televisor, etc.

Desde el comienzo el proyecto fue concebido como un sistema integral. Es decir, un mismo dispositivo que pueda ser usado para controlar todos los dispositivos mencionados. De esta forma, aparte del ahorro económico que esto significa, permite a una persona familiarizarse con el dispositivo mediante el control de, por ejemplo, el televisor, para luego pasar a un dispositivo más complejo como puede ser la silla de ruedas.

Nivel de la lesión	Posible deterioro	Potencial de rehabilitación
C2-C3	Generalmente mortal debido a la incapacidad para respirar	Totalmente dependiente para todos los cuidados
C4	Tetraplejía y dificultad para respirar.	Dependiente para todos los cuidados; generalmente necesita un respirador.
C5	Tetraplejía con cierta funcionalidad en el hombro y el codo.	Puede ser capaz de alimentarse por sí mismo utilizando dispositivos de asistencia; generalmente puede respirar sin respirador, pero puede necesitar otro tipo de soporte respiratorio.
C6	Tetraplejía con funcionalidad en el hombro y el codo y cierta funcionalidad en la muñeca.	Puede ser capaz de empujar una silla de ruedas en interiores y sobre pavimentos lisos; puede ser capaz de ayudar en la alimentación y el aseo y vestirse por sí mismo; dependiente de otros para trasladarse.
C7	Tetraplejía con funcionalidad en el hombro, el codo, la muñeca, y cierta funcionalidad en la mano.	Puede ser capaz de empujar una silla de ruedas en el exterior, trasladarse por sí mismo y conducir un coche especialmente adaptado; puede ser capaz de colaborar en los programas de intestino y vejiga.
C8	Tetraplejía con funcionalidad normal del brazo; debilidad de la mano.	Puede ser capaz de empujar una silla de ruedas en el exterior, trasladarse por sí mismo y conducir un coche especialmente adaptado; puede ser capaz de colaborar en los programas de intestino y vejiga.
T1 - T6	Paraplejía con pérdida de funcionalidad por debajo de la mitad del pecho; control total de los brazos.	Independiente con autocuidados en una silla de ruedas; capaz de trabajar a jornada completa.
T6 - T12	Paraplejía con pérdida de funcionalidad por debajo de la cintura; buen control del torso.	Buen equilibrio al estar sentado; gran capacidad para manejar una silla de ruedas y realizar actividades deportivas.
L1 - L5	Paraplejía con niveles variables de afectación de los músculos de las piernas.	Puede ser capaz de caminar distancias cortas con aparatos ortopédicos y dispositivos de asistencia.

Tabla 1.1: Categorías de parálisis.

1.3. Soluciones actuales

Existen diversas soluciones que permiten el control de dispositivos, mediante muy variadas técnicas. Muchos de los dispositivos están ideados para un único uso, mientras que otros contemplan el control de múltiples dispositivos. En las secciones siguientes se presentan algunas de las soluciones más comunes.

1.3.1. Sensores mecánicos

En la Figura 1.2 se observa un control tradicional para silla de ruedas. Muchos fabricantes, adicional al control tipo joystick, ofrecen un sensor mecánico que rodea la cabeza del usuario. El usuario presiona el sensor trasero para cambiar de velocidad/sentido y los sensores laterales para indicar giro. Como ventaja se puede destacar que no se requiere colocar ningún elemento en la cabeza del usuario, ya que el sensor se encuentra fijo en la estructura de la silla.



Figura 1.2: Controles mecánicos tradicionales para sillas de rueda.

1.3.2. Sensores infrarrojos

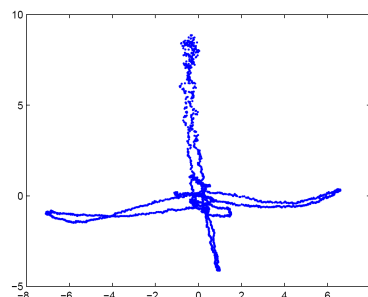
Una alternativa a los sensores mecánicos puede ser la implementación de sensores infrarrojos. Los mismos detectan la posición de la cabeza sensando la distancia mediante múltiples mediciones de distancia, como se ve en la Figura 1.3. Al igual que el sensor mecánico, no requiere colocar ningún elemento en la cabeza del usuario, pero tiene la ventaja de que no se debe presionar ni tocar ningún sensor. Debido a que los sensores están ubicados en la parte posterior de la cabeza, no obstruyen el campo visual.

1.3.3. Sensores basados en seguimiento de iris

En este caso los comandos se hacen mediante el movimiento de los ojos, con lo cual permite que el sistema sea utilizado por personas que no controlan el movimiento de su cuello. Típicamente se trata de una cámara de video conectada a una computadora, con lo cual el costo y complejidad del sistema es mayor. Una ventaja importante es que al conectarse a través de una computadora es que permite ampliar las posibilidades y flexibilidad del sistema.

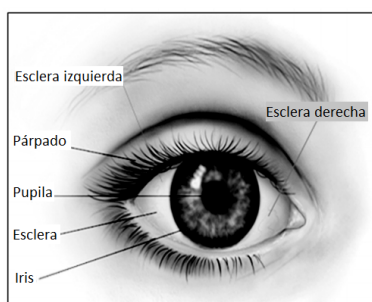


(a) Sensores infrarrojos atrás de la cabeza.



(b) Coordenadas obtenidas.

Figura 1.3: Control de silla de ruedas mediante sensores infrarrojos.



(a) Ojo humano.



(b) Control de una silla de ruedas a través de una computadora.

Figura 1.4: Control de silla de ruedas mediante detección de iris.

1.3.4. Sensores basados en ondas cerebrales

Existen diversos dispositivos basados en EEG³ ya sea para el control de sillas de ruedas o para otros tipos de dispositivos. Entre sus principales ventajas se puede destacar que no requieren del movimiento de la persona, y que existen múltiples aplicaciones simultáneas. En la Figura 1.5 se puede observar una silla de ruedas controlada mediante EEG. Esta tecnología resulta muy costosa y compleja de desarrollar, debido a la dificultades que presenta la medición de dichas corrientes. En primer lugar se trabaja con tensiones del orden de los microvolts, propensas a ser afectadas por ruido eléctrico. Además, debido a la colocación de los electrodos, es difícil conseguir mediciones confiables y repetibles, ya que un mínimo movimiento de los mismos afecta los resultados de las mediciones. Aún si se logra superar estos problemas, la correcta interpretación de estas señales no es sencilla. Sin embargo, esta tecnología se encuentra en constante desarrollo y ya se han obtenido resultados satisfactorios.

³EEG o electroencefalograma, es una técnica para medir la actividad eléctrica a lo largo del cuero cabelludo. Mide las variaciones de tensión que resultan de los flujos de corrientes iónicas dentro de las neuronas del cerebro.



Figura 1.5: Control de una silla de ruedas mediante ondas cerebrales diseñado por Toyota.

1.4. Método propuesto

El sistema desarrollado consiste en el control de distintos dispositivos mediante el movimiento de la cabeza. Este proyecto está orientado a personas con discapacidades físicas que no dispongan del movimiento de sus extremidades pero mantengan el movimiento de su cabeza. Básicamente se trata de colocar un módulo, tipo vincha, en la cabeza del usuario, el cual se encarga de leer los datos de un sensor de aceleración, de forma de poder determinar la inclinación de la cabeza, y por medio de un microcontrolador, transmitirlo mediante un transceptor⁴ de radio frecuencia. En el extremo receptor, se utilizará un transceptor idéntico, un microcontrolador, y la circuitería necesaria dependiendo de la aplicación en particular. En cada receptor, se interpretarán los datos de aceleración recibidos y se actuará dependiendo del objetivo del mismo. En la Figura 1.6 se puede observar lo dicho anteriormente mediante un diagrama genérico.

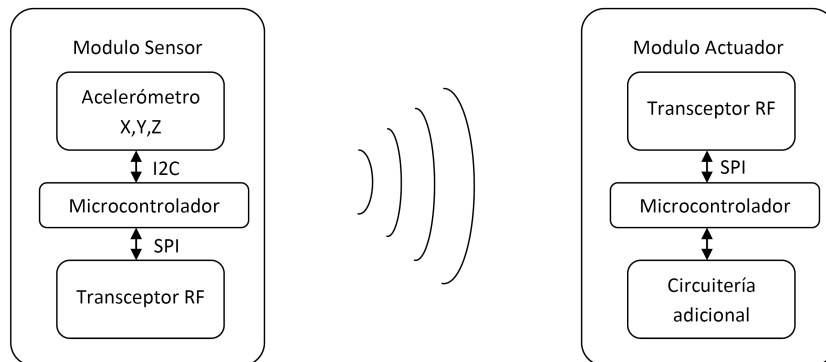


Figura 1.6: Sistema propuesto basado en sensor de aceleración y ondas de radio frecuencia.

⁴Transceptor: dispositivo inalámbrico que permite enviar y recibir información mediante ondas de radio.

1.4.1. Funcionamiento

El sistema dispone de un módulo sensor, que se puede vincular con distintos receptores simplemente presionando un botón en ambos terminales. Este mecanismo es similar a los dispositivos Bluetooth[®] que se pueden vincular y desvincular fácilmente. Esto requiere la asistencia inicial de un tercero que coloque la vincha y lo vincule con el receptor deseado. Entre las aplicaciones posibles para este sistema se pueden destacar:

- Controlar una computadora.
- Desplazar una silla de ruedas.
- Mover un auto a radio control.
- Controlar las funciones básicas de un televisor.

1.4.2. Ventajas

Comparando con las demás técnicas existentes en la actualidad, se pueden destacar las siguientes ventajas.

- Sistema más económico, debido a los costos de las tecnologías y debido a la posibilidad de usar un único sensor para múltiples usos.
- Control inalámbrico.
- Flexibilidad para controlar distintos dispositivos.
- Posibilidad de familiarizarse con el dispositivo mediante una aplicación libre de riesgo, para luego avanzar a algo más complejo.
- Debido a que es un sistema modular, queda abierta la posibilidad a nuevas aplicaciones con el mismo sensor.

Para lograr que el sistema sea modular y flexible, el transmisor está preparado para adaptarse a las necesidades de cada aplicación. Mediante un protocolo de handshake⁵ el receptor le indica cuáles son sus necesidades, relacionadas con frecuencia de muestreo, fondo de escala, filtrados, etc.

1.4.3. Desventajas

La desventaja principal de este sistema es la necesidad de que el usuario tenga relativamente buen control sobre su cabeza. Dependiendo de que grado de control tenga sobre la misma, variará las posibilidades de aplicación. También requiere de asistencia para la colocación del dispositivo y para la configuración del mismo. De acuerdo al funcionamiento actual, también es necesario asistencia para pasar de un uso a otro. Posteriormente se puede realizar una mejora que posibilite al usuario poder cambiar de usos mediante gestos con la cabeza. De esta forma permitiría al usuario poder desplazarse en una silla de ruedas eléctrica hasta el televisor y poder cambiar de canales. Luego, podría apagar el televisor y desplazarse a otro punto de la casa y realizar otra actividad.

⁵Handshake, etapa inicial de un protocolo de comunicación donde dos o más partes se dan a conocer e intercambian atributos.

Capítulo 2

Marco Teórico

En este capítulo se presentan brevemente las tecnologías y conocimientos técnicos que fueron necesarios para el desarrollo del proyecto. A su vez se describen las características de los dispositivos comerciales utilizados y los motivos de su elección.

2.1. Sensor de aceleración

La base de este proyecto consiste en detectar el movimiento de la cabeza. Más precisamente, lo que se busca detectar es la inclinación de la misma. Para ello se debe sensar las fuerzas que la gravedad terrestre ejerce sobre el sensor.

Existen diferentes tipos de sensores disponibles en el mercado, los cuales a grandes rasgos se pueden dividir en sensores de alterna o sensores de continua. Los primeros están acoplados en alterna, con lo cual no son aptos para medir aceleración constante como por ejemplo la aceleración de la gravedad terrestre \vec{g} . Los sensores de continua están acoplados a partir de los 0hz con lo cual si permiten medir aceleraciones estáticas o de variación lenta.

Algunos de los sensores disponibles son:

Sensores piezoeléctricos: Se basan en el desplazamiento de cargas que se genera a partir de la aceleración que se ejerce a la masa del sensor. Están acoplados en alterna, tienen un ancho de banda relativamente alto (típicamente mayor a 10KHz) y un gran rango de temperatura, entre $-200^{\circ}C$ y $+400^{\circ}C$.

Sensores piezoresistivos: Este tipo de sensores varía su resistencia en función de la aceleración a la que son sometidos. Están acoplados en continua, con anchos de banda de hasta 5Khz. Son susceptibles a cambios de temperatura, con lo cual deben ser compensados circuitalmente.

Sensores capacitivos: Se basan en la variación de capacidad de las placas de un capacitor sometidas a la fuerza que provoca la aceleración. Están acoplados en continua y tienen un ancho de banda relativamente bajo, de unos cientos de Hertz. Es la tecnología más común hoy en día, ya que se encuentra en dispositivos como sensores de airbags, teléfonos celulares, tablets o discos rígidos.

2.1.1. Principio de funcionamiento

El principio básico de funcionamiento del sensor de aceleración es descrito en la Figura 2.1, donde se puede observar un capacitor de tres placas paralelas.

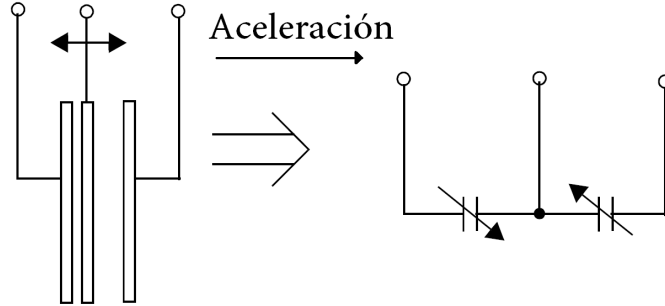


Figura 2.1: Principio de funcionamiento del sensor de aceleración

Las placas laterales se encuentran fijas respecto del encapsulado del sensor, mientras que la placa central es móvil. Se puede observar en la ecuación 2.1 que la capacidad aumenta al disminuir la distancia entre placas D , y disminuye cuando aumenta D . El sensor se encarga de convertir dicha capacidad en un valor de tensión, para luego traducirlo en un valor digital.

$$C = \frac{A \cdot \epsilon}{D} \quad (2.1)$$

Cabe destacar que el desplazamiento de la placa central se puede deber a la fuerza ficticia opuesta a la aceleración de un sistema no inercial, como muestra la ecuación 2.2, o bien por la acción de la fuerza de la gravedad, como muestra la ecuación 2.3. Ambos fenómenos pueden verse en la Figura 2.2.

$$\vec{f}^* = -m \cdot \vec{a}^* \quad (2.2)$$

$$\vec{P} = -m \cdot \vec{g} \quad (2.3)$$

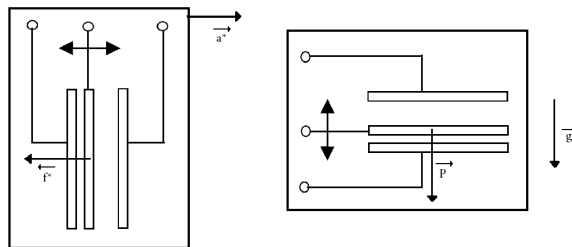


Figura 2.2: Distintas fuerzas que puede registrar el sensor de aceleración.

2.1.2. Tecnología MEMS

La tecnología MEMS (Micro-Electro-Mechanical Systems o Micro Sistemas Electro Mecánicos) permite la creación de pequeños dispositivos electromecánicos en escala reducida (de 1mm hasta 1 μ m). Se pueden recrear dispositivos mecánicos clásicos como poleas, agujeros, mensulas, engranajes, etc. Una importante ventaja de esta tecnología es que comparte los métodos de fabricación de los circuitos integrados, lo que permite la combinación de ambos dispositivos en un mismo encapsulado. La posibilidad de fabricar estos dispositivos en gran escala hace posible lograr bajos costos de producción.



Figura 2.3: Ejemplo de un dispositivo fabricado con la tecnología MEMS.

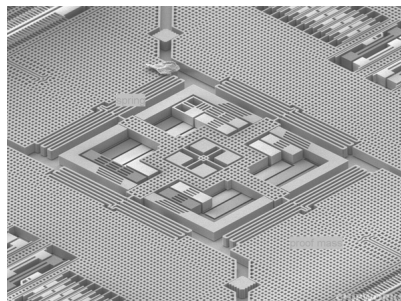


Figura 2.4: Ejemplo de un sensor de tres ejes fabricado con la tecnología MEMS.

2.1.3. Sensor MMA8451

En este proyecto se utilizó un acelerómetro MMA8451Q de la marca Freescale, cuyas características básicas son las siguientes:

- Tensión de alimentación 1.9-3.6V
- Rango de aceleración variable entre $\pm 2G$, $\pm 4G$ y $\pm 8G$
- Interfaz de salida digital I2C
- ADC con resolución de 14bits

- Muy bajo consumo de corriente: 6-165uA

Los motivos principales de su elección fueron su bajo consumo, interfaz digital, buena resolución, bajo costo, y la posibilidad de alimentarlo con dos pilas AAA recargables, cuya tensión nominal es $V = 2 * 1,2V = 2,4V$.

Se puede ver en la Figura 2.5 su diagrama en bloques simplificado que consiste en tres transductores colocados ortogonalmente conformando los ejes X, Y y Z, un convertor de capacidad a tensión y un convertor analógico a digital.

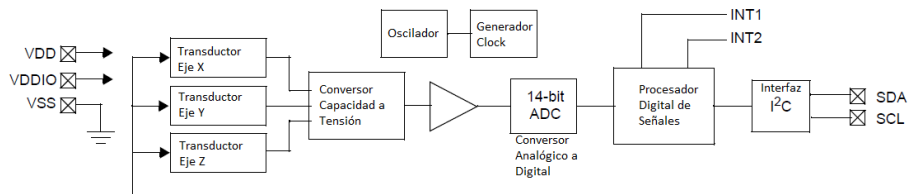


Figura 2.5: Diagrama en bloques del sensor de aceleración MMA8451.

El sensor está construido en un encapsulado QFN de 16 pines, cuyas dimensiones son 3x3x1mm, como se puede ver en la Figura 2.6.

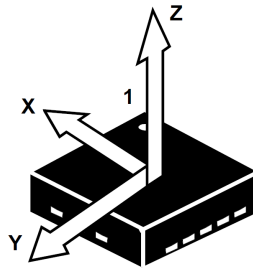


Figura 2.6: Encapsulado del sensor de aceleración MMA8451.

Como estos sensores están orientados al uso en teléfonos celulares y tablets, dispone de un procesador que analiza los datos y permite activar interrupciones ante eventos tales como cambio de orientación, caída libre, detección de movimiento, etc. De esta forma permite liberar al procesador del dispositivo de hacer ese tipo de cálculos, con el consiguiente ahorro de energía y tiempo de procesamiento. En este proyecto esas funciones no son utilizadas ya que el transmisor solo se encargará de transmitir los datos sin procesar, y cada receptor los interpretará en función de sus necesidades.

2.1.4. Interfaz I2C

Para la comunicación con el microcontrolador se dispone de un puerto I2C, mediante el cual se realizan todas las operaciones de lectura y escritura de registros para su configuración, así como para acceder a la información propiamente dicha. El bus I2C dispone de dos líneas únicamente: SDA para transmisión de datos y SCL para transmitir una señal de reloj, con lo cual se trata de un sistema sincrónico half-duplex como se puede ver en la Figura 2.7.

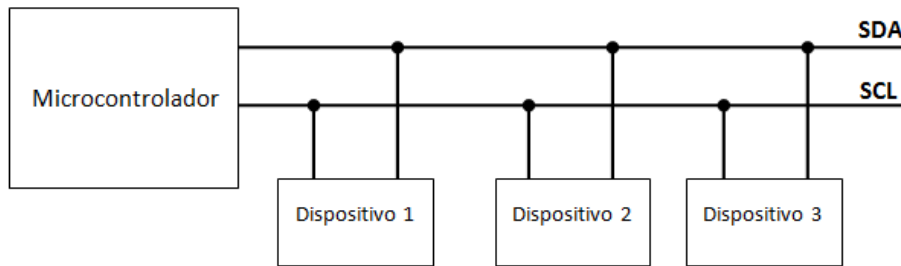


Figura 2.7: Diagrama conexión del bus I2C

Para identificar el dispositivo al que se quiere acceder, luego de una señal de Start (ST) se envía la dirección del dispositivo a acceder. En este proyecto solo se utiliza un dispositivo, pero aún así es necesario cumplir con el protocolo completo. Para leer un registro, se deberá operar de la siguiente manera: se envía una señal ST junto a la dirección del dispositivo y un bit 0 indicando que la operación es de escritura. Luego se espera una respuesta ACK y a continuación se envía la dirección del registro que se quiere leer, seguido de un bit 1 indicando lectura. Luego el sensor envía una señal ACK seguida del contenido del registro solicitado. Por último se envía una señal NAK (Not Acknowledge) seguido de una señal de Stop (SP) para finalizar la comunicación. El proceso se describe en la Figura 2.8.

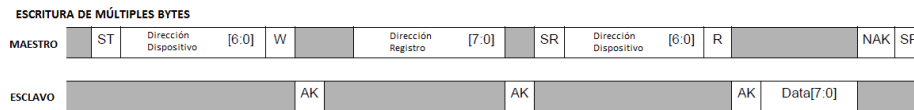


Figura 2.8: Diagrama de lectura de registro del bus I2C.

Para escribir un registro se envía una señal ST seguida de la dirección del dispositivo y un bit 0 indicando escritura, luego se escribe la dirección del registro deseado y por último la información que se desea grabar. Para indicar el fin de la comunicación se envía una señal SP. El proceso se describe en la Figura 2.9.

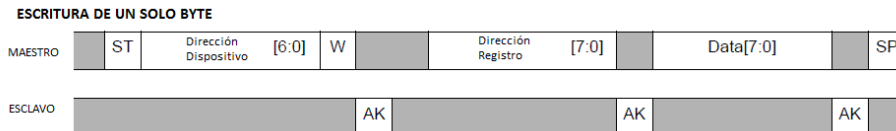


Figura 2.9: Diagrama de escritura de registro del bus I2C.

Como el proceso de lectura y escritura de registros individuales requiere más tiempo indicando direcciones que información, el protocolo permite hacerlo de forma secuencial. Como se puede ver en la Figura 2.10 luego de recibido el primer byte de información, si en vez de enviar una señal NACK se envía

un ACK, el sensor seguirá transmitiendo el contenido del registro siguiente. El mismo proceso puede observarse en la Figura 2.11 para el caso de escritura secuencial de múltiples registros.

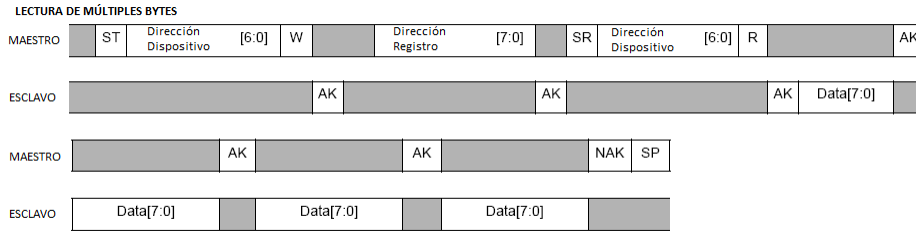


Figura 2.10: Diagrama de lectura de múltiples registros del bus I2C.

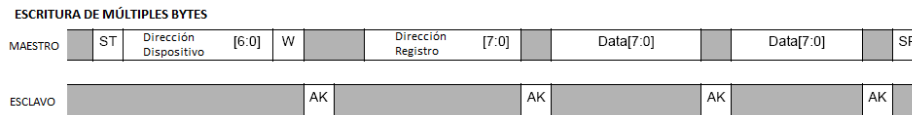


Figura 2.11: Diagrama de escritura de múltiples registros del bus I2C.

2.1.5. Opciones de configuración

El sensor MMA8451 permite varias opciones de configuración para brindar flexibilidad, ahorro de consumo y ahorro de recursos de CPU, entre ellos se encuentran:

Modos de operación

En la Figura 2.12 se observan los distintos modos de funcionamiento. En el modo *STANDBY* (reposo) solo permanece activa la parte digital del sensor, la cual permite la comunicación vía I2C, pero la parte analógica se encuentra desactivada para ahorrar energía. En los modos *SLEEP* (dormido) y *WAKE* (despierto) tanto la parte digital como la analógica se encuentran activadas. La diferencia entre ambos es que el primero sensa la aceleración a una frecuencia menor, con el objetivo de ahorrar energía hasta detectar cambios que despierten al dispositivo, como se verá próximamente.

Fondo de escala

El sensor permite cambiar el fondo de escala en valores de $\pm 2G$, $\pm 4G$ y $\pm 8G^1$. Si bien se usan 14bits para la conversión, los bits se alinean a la izquierda en dos registros de 8 bits, con lo cual se tienen 16bits para representar los valores de aceleración con signo incluido. Esto quiere decir que el sensor provee valores entre +32767 y -32768 respectivamente.

¹G: aceleración de la gravedad terrestre, $\approx 9,81 \frac{m}{s^2}$

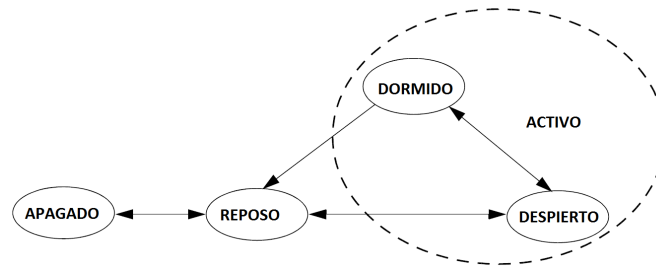


Figura 2.12: Modos de operación del sensor MMA8451.

Filtrado

Existe la posibilidad de emular que se está trabajando con un acelerómetro acoplado en alterna utilizando un filtro pasa altos interno. De esta forma se elimina, por ejemplo, el nivel de continua que provoca la aceleración de la gravedad terrestre.

Bajo consumo vs alta resolución

Se disponen 4 modos de operación, los cuales involucran una relación de compromiso entre consumo y ruido. Los modos de alta resolución logran reducir el ruido a través del sobre muestreo, a costa de un mayor consumo.

- Normal.
- Alta resolución.
- Alta resolución y bajo consumo.
- Bajo consumo.

Opciones avanzadas

Pensado para la aplicación de teléfonos celulares y tablets, este sensor dispone funciones avanzadas para facilitar su uso y liberar recursos del CPU:

- Detección de posición horizontal/vertical.
- Detección de movimiento y caída libre.
- Cambio automático entre modos WAKE y SLEEP.
- Detección de golpes.

2.2. Enlace de radio frecuencia

Debido a la comodidad y flexibilidad de uso, se determinó que el dispositivo debería ser inalámbrico. Como la finalidad de este trabajo no es diseñar un enlace de radio frecuencia, sino hacer una aplicación concreta en base a uno, se decidió la implementación de un circuito integrado para tal fin. En el proceso de selección se tuvieron en cuenta diversos factores, tales como costo, consumo,

tamaño, etc. Inicialmente se estudiaron sistemas estandarizados como Zigbee[®] o Bluetooth[®], ambos trabajan en la banda ISM² de 2.4Ghz. Pese a funcionar bajo un protocolo propietario, se optó finalmente por el transceptor nRF24L01+ de la firma Nordic Semiconductor[®], debido a su reducido tamaño, consumo y costo.

2.2.1. Tranceptor nRF24L01+

El módulo nRF24L01+ es un transceptor, es decir que se lo puede usar tanto como transmisor como receptor. Dispone de un encapsulado QFN³ de 20 pines y 4x4mm como se puede ver en la Figura 2.13. Entre sus características principales se puede destacar:

- Operación en la banda ISM de 2.4GHz.
- Modulación en GFSK.
- Velocidad de transmisión variable entre 250Kbps, 1Mbps y 2Mbps.
- Bajo consumo de corriente.
- Potencia de salida de 1mW.
- Tensión de alimentación 1.9-3.6V.
- Protocolo propietario Enhanced ShockBurst[™].
- Retransmisión automática.
- Respuesta automática con paquetes de información.
- Manejo de paquetes automático.

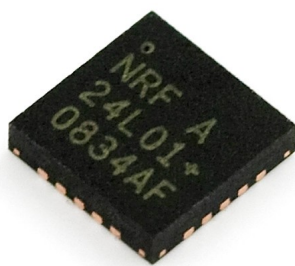


Figura 2.13: nRF24L01+ en su encapsulado QFN de 20 pines de 4x4mm.

²ISM (Industrial, Scientific and Medical) son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica.

³QFN (quad-flat no-leads) es un encapsulado de circuitos impresos de montaje superficial cuyos contactos se encuentran en la parte inferior del mismo, usualmente sobre los cuatro bordes.

En la Figura 2.14 se puede ver un diagrama en bloques del circuito. Incorpora en un mismo encapsulado toda la circuitería necesaria para la transmisión y recepción de radio frecuencia, con lo cual solo requiere unos pocos componentes externos pasivos.

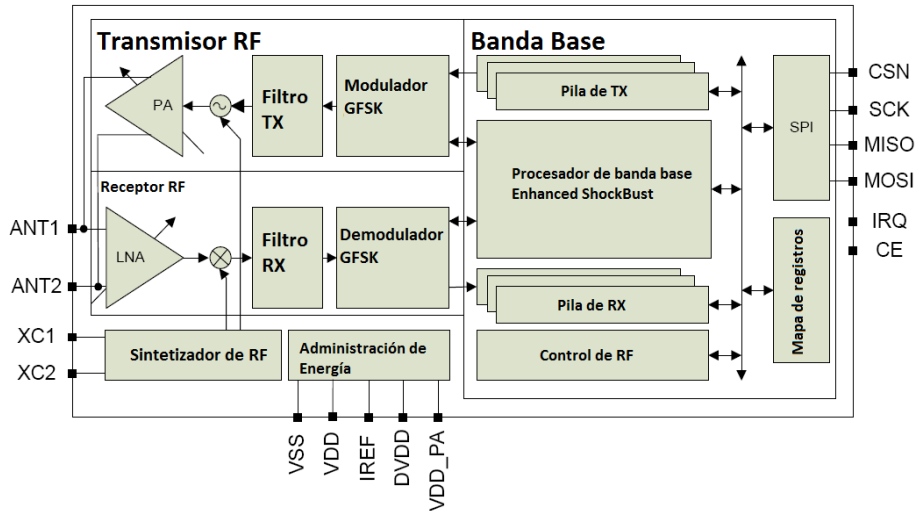


Figura 2.14: Diagrama en bloques del trancceptor nRF24L01+.

2.2.2. Modulación GFSK

La modulación utilizada es una variante de la modulación en frecuencia FSK (Frequency Shift Keying), la cual consiste en variar la frecuencia de una señal senoidal en función de la señal de entrada. Es decir, según el dato a transmitir sea un 1 o un 0, se transmitirá en una frecuencia o otra. Este proceso se puede ver en Figura 2.15, donde durante los pulsos en 1 se transmite a una frecuencia mayor a los pulsos en cero.

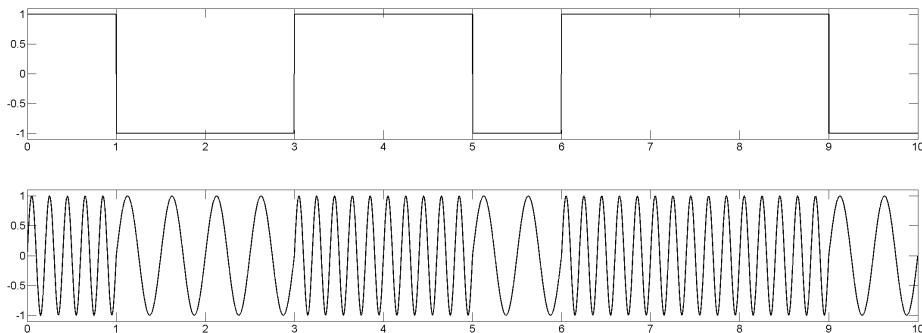


Figura 2.15: Modulación FSK.

Con el objetivo de reducir el ancho de banda de la señal, respecto a modulación en frecuencia estándar, se pueden filtrar los pulsos digitales antes de

ingresar al modulador, de manera de suavizar la señal y por lo tanto reduciendo el contenido en alta frecuencia. En el caso de GFSK se utiliza un filtro Gaussiano, como muestra la Figura 2.16.



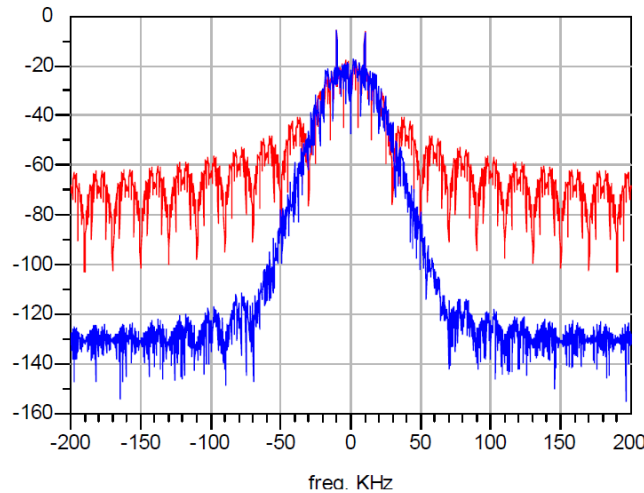
Figura 2.16: Diagrama en bloques de un sistema de modulación GFSK.

El equivalente pasa bajos del espectro de una señal modulada en FSK se puede ver en la ecuación 2.4, cuyo pulso y su espectro está dado por la ecuación 2.5. En el caso de GFSK, se ingresa previamente a un filtro cuya respuesta al impulso y su transformada están dadas por la ecuación 2.6. El resultado se puede ver de forma cualitativa en la Figura 2.17, donde se aprecia la reducción de ancho de banda provocada por el filtrado previo.

$$G_{LP} = \frac{1}{4} \left[\delta \left(f - \frac{r_b}{2} \right) + \delta \left(f + \frac{r_b}{2} \right) \right] + r_b |P(f)|^2 \quad (2.4)$$

$$p(t) = \text{rect} \left(\frac{t}{T_b} \right) \Leftrightarrow |P(f)|^2 = T_b^2 \text{sinc}^2(fT_b) \quad (2.5)$$

$$h(t) = e^{-\frac{1}{2} \left(\frac{t}{\tau} \right)^2} \Leftrightarrow H(f) = \tau \sqrt{2\pi} e^{-\frac{1}{2} (2\pi f \tau)^2} \quad (2.6)$$



Señal FSK **Señal GFSK**

Figura 2.17: Comparación cualitativa entre el espectro de FSK y GFSK.

2.2.3. Interfaz SPI

Para acceder al transceptor es necesario hacerlo mediante un enlace SPI, a través del cual se realiza la configuración inicial y la transmisión/recepción de datos. El mismo se compone de 4 líneas como se ve en la Figura 2.18.

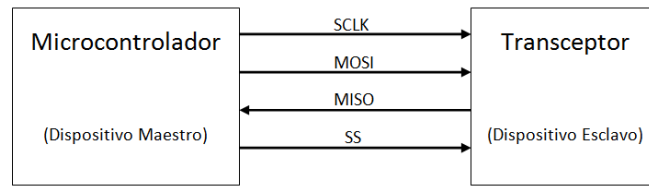


Figura 2.18: Diagrama de conexión SPI.

Donde:

- SCLK: Señal de reloj.
- MOSI (Master Out Slave In): Salida del dispositivo maestro, entrada del esclavo.
- MISO (Master In Slave Out): Entrada del dispositivo maestro, salida del esclavo.
- SS (Slave Select): Habilitación del dispositivo esclavo.

Se trata de una transmisión sincrónica del tipo full-duplex del tipo data exchange, es decir que siempre que se realiza una comunicación, se realiza un intercambio simultáneo de datos entre maestro y esclavo. Para conectarlo con múltiples dispositivos esclavos se necesitan múltiples SS. De esta forma los paquetes de transmisión no necesitan dirección para identificar el receptor.

Para la configuración y transmisión se deberán escribir y leer registros enviando datos en un orden determinado. Primero se deberá indicar el tipo de operación a realizar mediante un byte de comando, ya sea para lectura o escritura, y luego se deberá escribir o leer la cantidad de bytes que sean necesarios. Además, el transceptor al comienzo de cada operación, envía el byte de estado, el cual indica el estado de las transmisiones y recepciones, como se puede ver en la Figura 2.19.

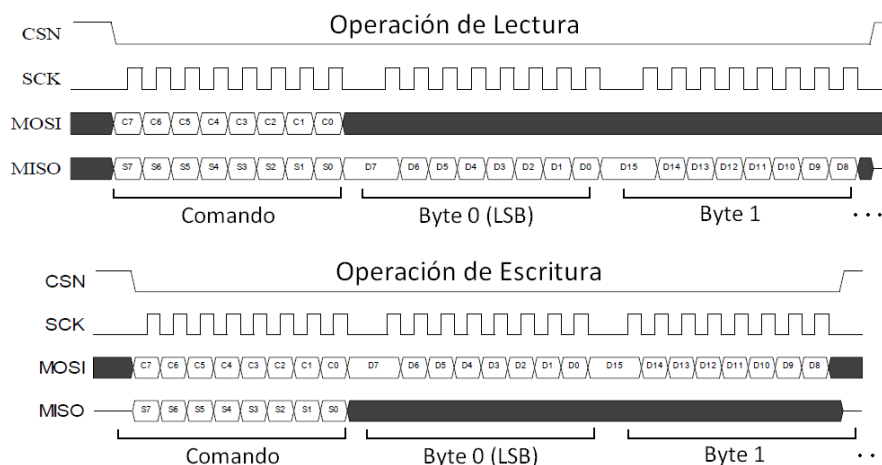


Figura 2.19: Operación de lectura y escritura del puerto SPI.

2.2.4. Diagrama de estados

Si bien el transceptor puede operar tanto como transmisor como receptor, se debe configurar su uso primario. Dependiendo de eso será su comportamiento. En la Figura 2.20 se puede ver un diagrama de estados que explica como opera el transceptor. Inicialmente al recibir alimentación se encuentra en modo *Power Down* hasta que se setea el bit PWR_UP en el registro correspondiente. Luego de un tiempo ingresa al modo *Standby-I*. Si se encuentra configurado en modo receptor y se habilita el pin CE, ingresa en *RX Mode* esperando que llegue un paquete válido. Si se encuentra configurado en modo transmisor y se habilita el pin CE, ingresa en *TX Mode* siempre y cuando tenga información guardada para transmitir. En caso contrario permanece en modo *Standby-II* hasta recibir información para transmitir. En ambos casos si se deshabilita el pin CE volverá al modo *Standby-I* y si se borra el flag PWR_UP volverá al modo *Power Down*.

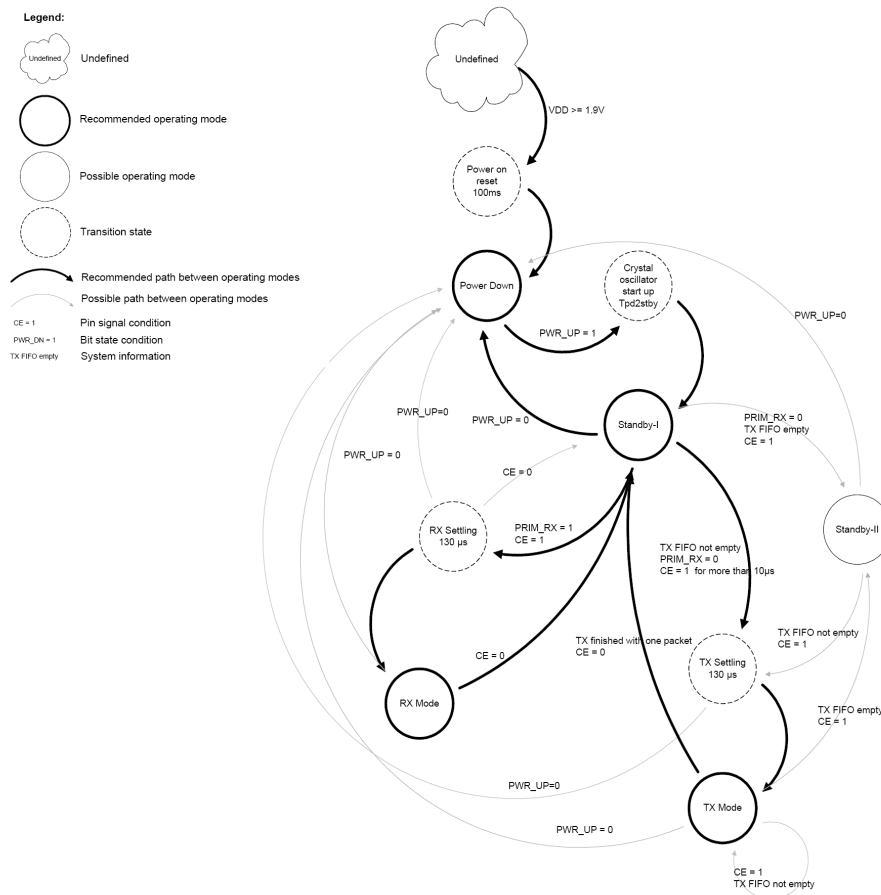


Figura 2.20: Diagrama de estados del transceptor nRF24L01+.

2.2.5. Protocolo propietario Enhanced Shockburst™

El transceptor nRF24L01+ provee un protocolo para facilitar al usuario ciertas tareas como pueden ser:

- Manejo automático de paquetes.
- ACK⁴ automático.
- Reenvío automático de paquetes.
- Comprobación de errores.

Formato de paquetes

El transceptor se encarga de generar automáticamente los paquetes de datos de acuerdo a la orden que recibe. En la Figura 2.21 se puede ver el formato genérico de los mismos.

Preámbulo 1 byte	Dirección 3 - 5 bytes	Registro de Control 9 bits	Carga 0 - 32 bytes	CRC 0 - 2 bytes
------------------	-----------------------	----------------------------	--------------------	-----------------

Figura 2.21: Formato del paquete del transceptor nRF24L01+.

El preámbulo es una sucesión de unos y ceros intercalados para sincronizar el demodulador del receptor. La dirección asegura que el paquete llegue al destino correcto, previniendo información cruzada en un sistema con múltiples transceptores. De acuerdo a la necesidad, puede ser de 3, 4, o 5 bytes. El campo de control es utilizado por el protocolo para el manejo automático de paquetes. El campo de Carga Útil es donde el usuario puede colocar la información que quiere transmitir, de acuerdo a la necesidad, puede ser de hasta 32 bytes. Por último se encuentra una sección donde puede ir código especial para la detección de errores. De acuerdo a la necesidad puede ser de 0 (desactivado), 1 o 2 bytes.

ACK automático

El ACK automático es una función que automáticamente transmite un paquete de ACK al transmisor después de que se recibe un paquete válido. Para que un paquete sea válido, debe tener la misma dirección y debe comprobarse correctamente el campo de CRC⁵. Es importante destacar que en los paquetes de respuesta, o ACK, también pueden tener un campo de Carga Útil, de forma de establecer un enlace bidireccional robusto.

Retransmisión automática

La retransmisión automática es una función que retransmite un paquete cuando no se recibe ACK válido. Se puede establecer el tiempo que se debe esperar el ACK antes de retransmitir (entre $250\mu s$ y $4000\mu s$) y la cantidad de

⁴ACK: Acknowledgment, tipo de paquete especial que envía el receptor para indicar al transmisor que se recibió correctamente

⁵CRC: Comprobación por Redundancia Ciclica, es un método de detección de errores frecuentemente usado en transmisiones digitales.

veces que debe retransmitirse. Se dice que es del tipo "mejor esfuerzo", ya que intenta retransmitir una cantidad de veces hasta que desiste.

Comprobación de redundancia cíclica

El sistema contempla la posibilidad de implementar detección y corrección de errores mediante un esquema del tipo ARQ (Automatic Repeat reQuest), donde se agrega una porción de código para permitir al receptor detectar errores, y en caso de ser necesario, solicitar una retransmisión. Se trata de un sistema tipo Stop and Wait, el cual envía un solo paquete, y espera su correspondiente ACK antes de enviar el próximo paquete, como muestra la Figura 2.22.

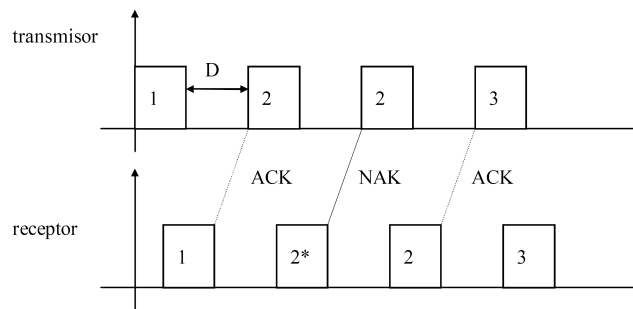


Figura 2.22: Esquema ARQ del tipo Stop and Wait.

Para la detección de errores se utilizan códigos cíclicos, cuya longitud se puede configurar entre 1 o 2 bytes. Para su cálculo considera como mensaje los campos de dirección, control y carga útil. Por ejemplo, si se utilizan 4 bytes de dirección, 8 bytes de carga útil, y 1 byte de CRC, sumado a los 9 bits de control, se tiene un total de bits $n = 113$ y $k = 105$ de mensaje. Eso permite una distancia mínima de Hamming⁶ dada por:

$$d_{min} \leq n - k + 1 = 10 \quad (2.7)$$

Lo que significa que se pueden detectar hasta $l = 9$ errores por paquete.

2.3. Microcontroladores

El dispositivo central de cada dispositivo del proyecto es un microcontrolador, es decir un circuito integrado capaz de ejecutar un programa grabado en su memoria, que se comunica al exterior a través de sus periféricos. Existen diversos fabricantes de microcontroladores, entre los cuales se eligió Microchip[®] debido a su mayor disponibilidad en el mercado local, sumado a que ya se disponía experiencia con los mismos y los elementos necesarios para su programación.

⁶Se llama distancia de Hamming entre dos vectores al número de coordenadas o elementos diferentes entre ellos.

2.3.1. Familia PIC18F

Se decidió trabajar con la familia *PIC18F*, la más avanzada dentro de los microcontroladores de 8 bits de Microchip® debido a las siguientes razones:

- Disponen de todos los periféricos necesarios: I2C, SPI, USB, PWM, etc.
- Existen versiones de muy bajo consumo con opciones de ahorro de energía.
- Relativo bajo costo.
- Facilidad en su programación, gracias a su extensa documentación.
- Disponibilidad en el mercado local.

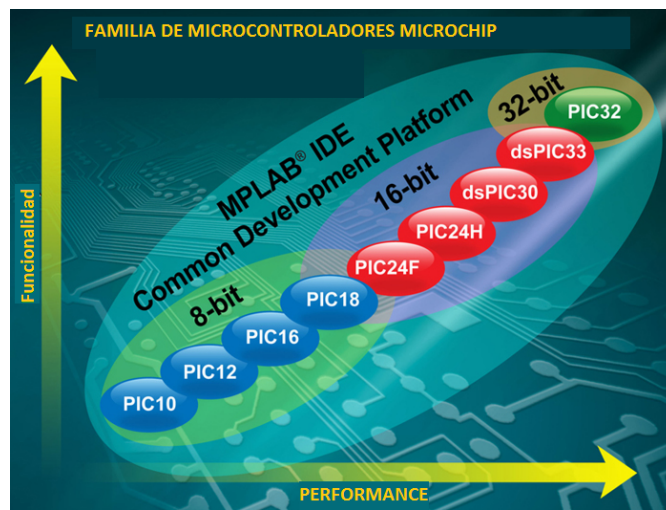


Figura 2.23: Familia de microcontroladores Microchip.

Se trabajó básicamente con dos microcontroladores, el *PIC18F2550* y el *PIC18F14K50*. En la Tabla 2.1 puede verse una comparación de sus características básicas. En ambos casos se utilizaron sus versiones en encapsulado DIP⁷, como se puede ver en la Figura 2.24, debido a su facilidad de soldado en etapa de prototipado. Para su programación se recurrió a un programador PICKIT 3TM de Microchip®, el cual se conectó mediante un terminal ICSP⁸, como se puede ver en la Figura 2.25.

2.3.2. Lenguaje y entorno de programación

Los microcontroladores fueron programados íntegramente en C, debido a su flexibilidad y facilidad en comparación con el lenguaje assembler. Se trabajó en el entorno de desarrollo MPLAB® IDE 8.87 junto al compilador C18, ambos

⁷Dual in-line package o DIP es una forma de encapsulamiento común en la construcción de circuitos integrados. La forma consiste en un bloque con dos hileras paralelas de pines, la cantidad de éstos depende de cada circuito.

⁸In Circuit Serial Programming, un método para grabar la memoria de un microcontrolador de forma serie. Facilita la programación ya que no requiere desmontar el microcontrolador de la placa.

Característica	PIC18F2550	PIC18F14K50
Frecuencia máxima	48Mhz	48Mhz
Memoria de programa	32KB	16KB
Memoria RAM	2048 Bytes	768 Bytes
Memoria EEPROM	256 Bytes	256 Bytes
Pines entrada/salida	24	15
Total pines	28	20

Tabla 2.1: Comparación entre los microcontroladores utilizados.

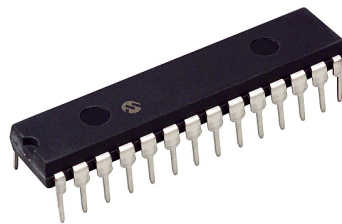


Figura 2.24: Microcontrolador en encapsulado DIP.

de Microchip[®]. Al comienzo del proyecto ya estaba disponible la nueva versión MPLAB[®] X y el compilador XC8[®], sin embargo todavía eran muy recientes y poseían errores e incompatibilidades aún no resueltas.

2.4. Estándar USB 2.0

En los desarrollos de circuitos electrónicos, a menudo resulta necesaria la comunicación entre un dispositivo y una computadora. La conexión clásica entre un proyecto microcontrolado es el puerto serie de la computadora. Durante los últimos años, la masividad con que avanzaron las computadoras portátiles hizo prácticamente desaparecer el puerto serie en reemplazo del popular USB. Si bien existen conversores USB-RS232 no siempre resulta conveniente su aplicación. Sumado a ello el puerto usb presenta dos ventajas interesantes: la posibilidad de alimentar el circuito desde el mismo puerto, y velocidades muy superiores al puerto serie.

El Bus Serie Universal (Universal Serial Bus) es un estándar creado en 1996 por las compañías Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips para solucionar el problema de conectividad entre dispositivos de PC de la época. De esta forma se logró reemplazar con éxito el puerto serie, el puerto paralelo, los conectores de teclado, mouse y joystick, el puerto midi, etc. Con este objetivo se desarrolló un puerto capaz de conectar múltiples dispositivos, con capacidad Plug & Play y un mínimo conocimiento necesario por parte del usuario para utilizarlo.

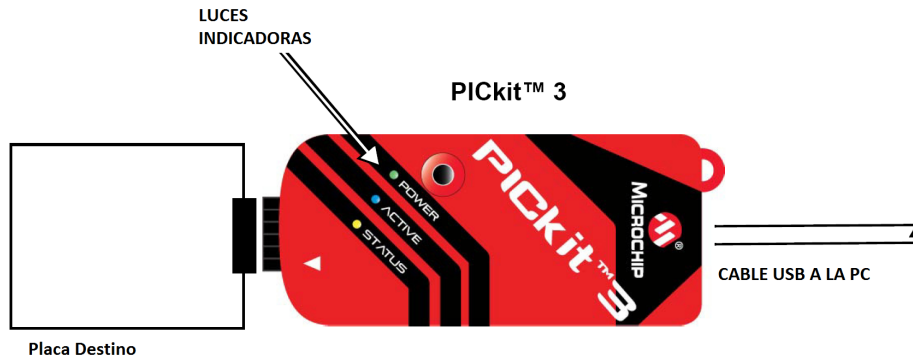


Figura 2.25: Programador PICKIT 3™ de Microchip®.

Denominación	Velocidad
Low Speed	1.5Mbps \pm 500ppm
Full Speed	12Mbps \pm 2500ppm
High Speed	480Mbps \pm 15000ppm

Tabla 2.2: Velocidades soportadas por el estándar USB 2.0.

2.4.1. Velocidades

El estándar USB define tres velocidades de señalización y sus tolerancias, como se puede ver en la Tabla 2.2. Inicialmente Low Speed era utilizado para dispositivos pequeños, que requerían poca velocidad, como un Mouse, y Full Speed para los demás dispositivos. Con la versión 2.0 se incorporó la velocidad High Speed superando al puerto Firewire. Los microcontroladores de Microchip soportan las velocidades Low y Full speed actualmente, por lo que se centrará el enfoque en dichas velocidades. Es importante aclarar que el estándar 2.0 incluye los tres tipos de velocidad pero no necesariamente debe soportar el modo High Speed.

2.4.2. Características eléctricas

El USB requiere un cable mallado de 4 conductores, como se ve en la Figura 2.26 los conductores D+ y D- forman un par trenzado ambos referenciados al conductor GND. El cuarto cable, Vbus corresponde a +5V que pueden ser utilizados para alimentar el dispositivo.

Pin	Color	Función
1	Rojo	Vbus(+5V)
2	Blanco	D-
3	Verde	D+
4	Negro	GND

Tabla 2.3: Pines del estándar USB 2.0.

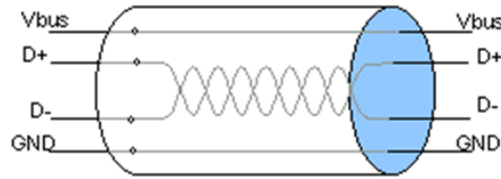


Figura 2.26: Pines del puerto USB.

	D+	D-
1 Diferencial	$> 2,8V$	$< 0,3V$
0 Diferencial	$< 0,3V$	$> 2,8V$

Tabla 2.4: Niveles de tensión USB 2.0.

Los terminales D+ y D- utilizan señalización diferencial, es decir se transmite la información utilizando ambos conductores y la salida se obtiene restándolos. Como se ve en la Figura 2.27, este método de señalización presenta mayor inmunidad al ruido, debido a que como ambos terminales se encuentran juntos, el ruido ingresa por ambos conductores y se anulan en el receptor. Los niveles de tensión mínimos y máximos se ven en la Tabla 2.4.

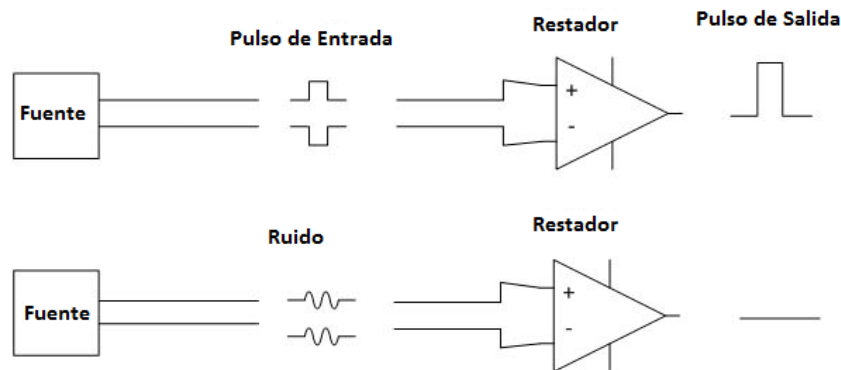


Figura 2.27: Señalización diferencial del puerto USB.

El USB utiliza codificación NRZI (Non-Return-to-Zero Inverted), el cual cambia de estado si el bit a transmitir es un 0, o se mantiene si es un 1, como muestra la Figura 2.28.

En cuanto a la potencia que puede suministrar, se definen tres tipos de dispositivos, dos de los cuales obtienen alimentación desde el bus, y uno que se auto alimenta. La Tabla 2.5 resume los tres tipos y sus características.

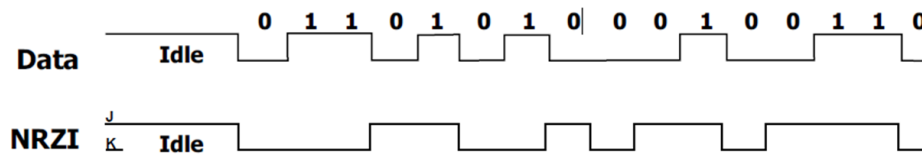


Figura 2.28: Codificación NRZI.

Dispositivo	Rango de V_{bus} a soportar	Corriente máxima
Low-power bus powered	$4,4V < V_{bus} < 5,25V$	100mA
High-power bus powered	$4,75V < V_{bus} < 5,25V$	500mA
Self-powered	—	100mA

Tabla 2.5: Niveles de corriente máximos USB 2.0.

2.4.3. Protocolo

El USB, como su nombre lo indica, es un bus, como se puede ver en la Figura 2.29. Esto quiere decir que solo se realizará una transmisión por vez. En este caso, al ser un bus del tipo “host controlled”, todas las comunicaciones las iniciará el host, y solo podrá haber un host por bus. Las transmisiones son recibidas por todos los dispositivos, pero solo responderá aquel que posea la dirección adecuada.

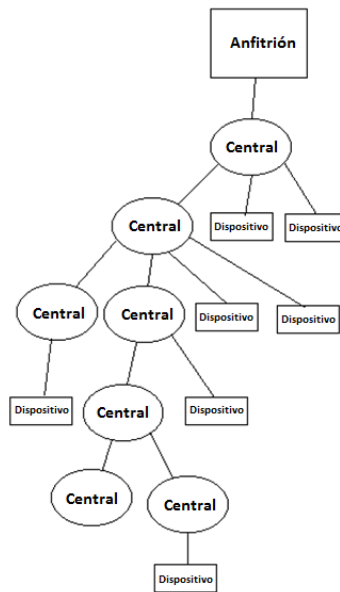


Figura 2.29: Bus USB.

Tipo de paquete	Subtipo	Descripción
Token	In	Informa al dispositivo que se requiere leer información.
	Out	Informa al dispositivo que se requiere escribir información.
	Setup	Usado para transferencias de control.
Data	Data0/1	Paquetes de información que pueden contener entre 0 y 1023 bytes.
Handshake	ACK	Indica que el paquete fue recibido correctamente.
	NAK	Indica que no puede recibir ni enviar información temporalmente.
	STALL	El dispositivo se encuentra en un estado que requiere intervención del host
Start of Frame	SOF	Paquete con un número de 11 bits enviado por el host cada 1ms.

Tabla 2.6: Niveles de corriente máximos USB 2.0.

2.4.4. Paquetes

La información es siempre transmitida en forma de paquetes, los cuales están compuestos por los siguientes campos:

- Sync: 8 bits utilizados para sincronización
- PID (Packet ID): 4 bits identifican el tipo de paquete transmitido.
- ADDR: 7 bits indican a que dispositivo está destinado el paquete, de 1 a 127, la dirección 0 está reservada para dispositivos que todavía no se les asignó una dirección.
- ENDP: 4 bits indican el endpoint, es decir, a que parte del dispositivo USB está destinado el paquete.
- CRC: 5 o 16 bits reservados para el chequeo de redundancia cíclica.
- EOP: fin del paquete.

Existen cuatro tipos de paquetes:

2.4.5. Transferencias

El USB especifica cuatro tipos de transferencia, como se puede ver en la Tabla 2.7, todas soportadas por la familia de microcontroladores utilizada.

2.4.6. Clases

Como existe gran diversidad de dispositivos, es necesario establecer un protocolo según la función que realice cada uno. Por ello se crearon distintas clases, como se puede ver en la Tabla 2.8, que identifican la función del dispositivo y permite la creación de drivers genéricos.

Tipo de transferencia	Descripción
Control	Este tipo provee el control de la configuración del dispositivo
Interrupt	Este tipo de transferencia asegura transferencias a tiempo de pequeños bloques de información, cuya integridad es asegurada.
Isochronous	Este tipo provee un método de transferencia para grandes cantidades de información con el tiempo de entrega asegurado, sin embargo, la integridad de la información no es asegurada.
Bulk	Este tipo de transferencia permite la transferencia de grandes cantidades de información cuya integridad es asegurada, pero no se asegura el tiempo de entrega.

Tabla 2.7: Transferencias USB 2.0.

2.4.7. Microchip USB Device Framework

En la página oficial de Microchip se pueden descargar las “Microchip Application Libraries”, un paquete de librerías que incluyen, además de lo relacionado al puerto USB, librerías para el control de displays gráficos, TCP/IP, WIFI, etc.

El microcontrolador utilizado puede funcionar en los siguientes modos:

- USB Human Interface Device (HID): utilizado para dispositivos de interfaz humana de baja velocidad, como por ejemplo un mouse. Tiene como ventaja que no requiere instalar drivers en la PC.
- USB Communication Device Class (CDC): esta clase requiere driver, pero tiene como ventaja que crea un puerto serie virtual en la PC, lo que resulta ideal para migrar de una aplicación que fue diseñada para trabajar con puerto serie.
- USB Mass Storage Device (MSD): es utilizada para discos rígidos externos, pen drives, etc.
- USB Custom Driver (GEN): esta clase utiliza drivers propietarios de Microchip, cuyas funciones pueden ser accedidas a través de la librería mpusbapi.dll.

2.5. Radiación Electromagnética

La radiación electromagnética es un tipo de propagación de energía sin transporte de masa. Está compuesta por un campo eléctrico y un campo magnético variable en el tiempo. Existen fuentes naturales de ondas electromagnéticas, como el sol y las estrellas, y fuentes artificiales, creadas por el ser humano. Dichas ondas no tienen un comportamiento continuo, sino que su energía se encuentra cuantificada en lo que denomina paquete de onda, o fotón. La energía que transporta cada fotón es directamente proporcional a su frecuencia, tal como indica la Ecuación 2.8, donde E es la energía, f es la frecuencia, y h es una constante denominada Constante de Planck.

Clase	Uso	Descripción	Ejemplos de aplicación
00h	Dispositivo	Sin especificar	Clase de dispositivo sin especificar, los descriptores de la interfaz son usados para determinar el driver necesario.
01h	Interfaz	Audio	Parlantes, microfonos, placas de sonido, MIDI
02h	Ambos	Comunicaciones y control CDC	Modems, adaptadores Ethernet o Wi-Fi
03h	Interfaz	Dispositivo de interfaz humana (HID)	Teclado, mouse, joystick
05h	Interfaz	Dispositivo de interfaz física (PID)	Joysticks con Force feedback
06h	Interfaz	Imagen	Cámara web, scanner
07h	Interfaz	Impresoras	Impresora laser, chorro de tinta, maquina CNC
08h	Interfaz	Almacenamiento masivo	Pendrives USB, lector de tarjetas de memoria, reproductor de audio digital, cámara digital, disco externo
09h	Dispositivo	Hub USB	Hub USB de banda ancha
0Ah	Interfaz	CDC-Información	Usado en conjunto con la clase 02h: Comunicaciones y control CDC
0Bh	Interfaz	Tarjeta inteligente	Lector de tarjetas inteligentes USB
0Dh	Interfaz	Seguridad de datos	Lector de huellas dactilares
0Eh	Interfaz	Video	Cámara web
0Fh	Interfaz	Cuidado de salud personal	Pulsera de monitor cardíaco
DCh	Ambos	Dispositivo de diagnostico	Dispositivo para comprobar la compatibilidad USB
E0h	Interfaz	Controlador Inalámbrico	Adaptador BluetoothS
EFh	Ambos	Miscelaneos	Dispositivo ActiveSync
FEh	Interfaz	Dispositivos de aplicaciones específicas	Adaptador IrDA, Clase de Prueba y Medición (USBTMC), USB DFU (Actualización directa de Firmware)
FFh	Ambos	Dispositivos específicos de fabricantes	Indica que el dispositivo requiere drivers específicos del fabricante

Tabla 2.8: Clases USB 2.0.

$$E = h \cdot f \tag{2.8}$$

En la Figura 2.30 se puede ver como se constituye el espectro electromagnético, es decir que tipo de ondas corresponden a cada rango de frecuencias. Hacia la derecha se encuentran las frecuencias más bajas (por ende menor energía), y desplazándose hacia la izquierda se encuentran las frecuencias de radio, las micro ondas, la luz infrarroja, la luz visible, los rayos x, los rayos gamma y los rayos cósmicos. Estos últimos siendo los de mayor frecuencia.

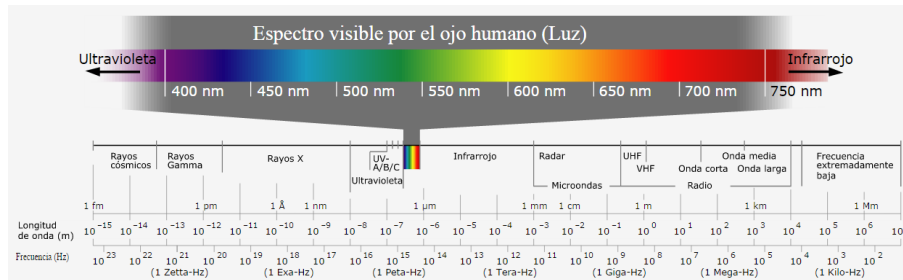


Figura 2.30: Espectro Electromagnético.

2.5.1. Radiación Ionizante y No Ionizante

Los átomos originalmente son eléctricamente neutros, es decir tienen igual carga negativa que positiva. Sin embargo, los electrones al ser impactados con un fotón con la suficiente energía se desprenden del átomo, produciendo un ión con carga positiva. Este fenómeno puede alterar la estructura molecular de la materia, y en el caso de seres vivos puede producir efectos no deseados.

La radiación electromagnética cuyas frecuencias se encuentran por debajo de la luz ultravioleta (por ejemplo, los 2.4Ghz de este proyecto) no tienen energía suficiente para ionizar la materia, por lo tanto no produce alteraciones celulares ni moleculares. Sin embargo las radiaciones no ionizantes, como toda radiación, transportan energía, con lo cual al radiar un objeto produce la elevación de temperatura del mismo. En el caso de los seres vivos, como animales o seres humanos, poseen la capacidad de regular su temperatura corporal, por ejemplo a través de la transpiración. Esta capacidad no es ilimitada, con lo cual se establecieron límites de intensidad de radiación varias veces por debajo del máximo tolerable.

2.5.2. Regulaciones sobre Radiación no Ionizante

En Argentina, el espectro electromagnético se encuentra regulado por la Comisión Nacional de Comunicaciones (CNC), la cual establece los valores máximos de potencia, campo eléctrico y campo magnético, como se puede ver en la Tabla 2.9. Para verificar si el dispositivo a evaluar se encuentra dentro de los umbrales permitidos existen dos formas: realizar el cálculo teórico en función de la potencia radiada, distancia de la antena a la persona, etc, o bien realizar la medición correspondiente.

Para explicar el procedimiento es conveniente definir los siguientes conceptos:

Rango de Frecuencia f [Mhz]	Densidad de potencia S [mW/cm^2]	Campo Eléctrico E [V/m]	Campo Magnético H [A/m]
0,3 - 1	20	275	0,73
1 - 10	$20/f^2$	$275/f$	$0,73/f$
10 - 400	0,2	27,5	0,073
400 - 2000	$f/2000$	$1,375\sqrt{f}$	-
2000 - 100000	1	61,4	-

Tabla 2.9: Valores máximos de radiación establecidos por la CNC.

Onda plana: Onda electromagnética en donde los vectores de campo eléctrico y magnético son ortogonales y perpendiculares a la dirección de propagación de la onda.

Región de campo cercano: En las cercanías de la antena (hasta una longitud de onda), los campos eléctricos y magnéticos no constituyen ondas planas sino que varían considerablemente punto a punto

Región de campo lejano: En esta región del campo radiado, la distribución angular es independiente de la distancia a la antena, el campo predominante es del tipo onda plana.

La densidad de potencia S, está vinculada con la magnitud del campo eléctrico E y el campo magnético H mediante la siguiente ecuación:

$$S = \frac{E^2}{Z_0} = H^2 Z_0 \quad (2.9)$$

Para determinar la densidad de potencia S, se puede usar la Ecuación 2.10:

$$S = \frac{2,56 \cdot PIRE \cdot F^2}{4 \cdot \pi \cdot r^2} \quad (2.10)$$

Donde:

S es la densidad de potencia medida en W/m^2 .

PIRE es el producto entre la potencia radiada por la ganancia de la antena respecto de un radiador isotrópico.

F es la atenuación en veces de la radiación para un cierto ángulo en el plano vertical, si es desconocido se toma $F = 1$.

r es la distancia de la antena, en metros.

La CNC establece que si bien los valores calculados mediante la Ecuación 2.10 corresponden a la radiación de campo lejano, pueden tomarse como el peor caso que se encontraría en la zona de campo cercano. A su vez establece que si superan los límites de la Tabla 2.9, deberán medirse los niveles de radiación emitidos.

Capítulo 3

Desarrollo

En este capítulo se presenta en primer medida el funcionamiento del sistema desarrollado, detallando su protocolo de comunicación. Luego se presenta cada dispositivo desarrollado, detallando sus componentes, funcionamiento, problemas encontrados y posibles mejoras.

3.1. Sistema propuesto

El sistema está pensado para poder vincular un mismo sensor con distintas aplicaciones, pero no simultáneamente. Esto permite disponer un único sensor, y poder utilizarlo para comandar un dispositivo en un momento, luego presionando un botón utilizarlo para comandar otro dispositivo. Para esto se establecieron 4 modos de funcionamiento y 4 tipos de paquetes, los cuales se explican a continuación.

3.1.1. Modos de funcionamiento

- **Modo Búsqueda:** en este modo tanto el transmisor como el receptor configuran una dirección preestablecida conocida como “dirección de búsqueda”. Luego el transmisor envía un paquete de búsqueda cada 500ms con el objetivo de buscar receptores compatibles. Desde el lado receptor al recibir un paquete de búsqueda devuelve un paquete de respuesta indicándole todas las especificaciones que requiere del sensor. Para entrar a este modo se deberá presionar un pulsador ubicado en cada dispositivo por más de 2 segundos.
- **Modo Play:** en este modo se ingresa desde el Modo Búsqueda después de haberse encontrado dos dispositivos. Se transmiten paquetes de datos con los valores de aceleración a la dirección y con los parámetros indicados en el paquete de respuesta a la búsqueda. Se envían con un periodo indicado por el receptor, entre 5ms y 250ms.
- **Modo Pausa:** en este modo se ingresa desde Modo Play, cuando el transmisor envía paquetes de datos y no obtiene respuesta. Cambia el periodo de transmisión a 500ms hasta que obtiene respuesta nuevamente, y vuelve al periodo indicado.

- **Modo Stop:** En este modo no se envía ningún paquete hasta que presiona el pulsador para entrar al modo búsqueda.

3.1.2. Diagrama de estados

En la figura 3.1 se puede apreciar mediante un diagrama de estados, como funciona la transición entre modos. Inicialmente se encuentra en modo Stop hasta que se presiona el pulsador durante al menos 2 segundos. Luego, en modo Búsqueda, el transmisor envía un paquete determinado a una dirección previamente definida. El receptor escucha dicha dirección y si recibe un paquete de búsqueda, le responde con un paquete especial con una serie de indicaciones que se explicarán más adelante. Una vez establecido el vínculo, el transmisor entra en modo Play y le envía continuamente al receptor la información del sensor de aceleración. Siempre que reciba respuesta seguirá enviando información a la frecuencia previamente especificada. Si deja de obtener respuesta durante un lapso de tiempo, el transmisor ingresa en modo Pausa, donde envía esporádicamente paquetes de información con el objetivo de retomar el contacto con el receptor. Cuando el receptor vuelve a responder, regresa automáticamente a modo Play.

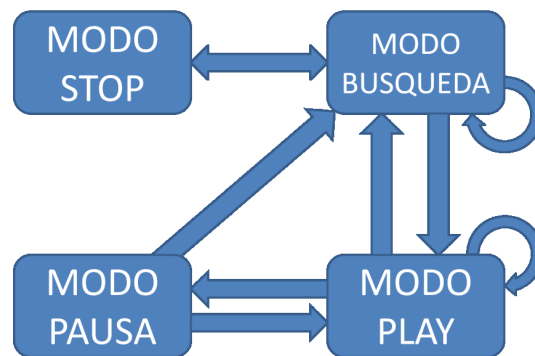


Figura 3.1: Diagrama de estados del sistema desarrollado.

3.2. Protocolo de comunicación

A continuación se especifica el protocolo de comunicación que deberán llevar a cabo tanto el transmisor como los receptores. En las especificaciones se encuentran los parámetros que se deberán configurar en el módulo transceptor nRF24L01+, así como la forma específica de cada tipo de paquete utilizado.

3.2.1. Tipos de paquete

Como se vió en la figura 2.21, el formato general de los paquetes está determinado por las especificaciones del fabricante, donde se puede variar la cantidad de bytes de dirección, carga útil y detección de errores. Dentro de la carga útil se arman los paquetes según el tipo de transmisión a realizar. En todos los paquetes enviados el primer byte de la carga útil es usado para identificar el tipo de paquete, como muestra la Tabla 3.1.

Tipo	Valor (Hex)	Descripción
P_ID_SRCH	0x8A	Paquete de búsqueda
P_ID_SRCH_ACK	0x8B	Paquete de respuesta a búsqueda
P_ID_DATA	0x8C	Paquete de datos
P_ID_CTRL	0x8D	Paquete de control

Tabla 3.1: Tipos de paquetes con su byte de identificación.

Para establecer un vínculo entre dos dispositivos se deberá iniciar el modo búsqueda en ambos terminales. De esta forma el receptor le enviará al transmisor todos los parámetros necesarios según la aplicación que realice el mismo. Una vez establecido el vínculo el transmisor enviará periódicamente datos, con la posibilidad que el receptor envíe paquetes de respuesta a modo de control.

A continuación se explica el funcionamiento y la sintaxis que deberán cumplir los módulos para su comunicación, recordando que el valor Auto Retransmit Delay indica el tiempo que espera el transmisor un ACK de respuesta antes de volver a enviar el paquete. El valor Auto Retransmit Count indica cuantas retransmisiones se intentarán antes de descartar el paquete. En todos los casos se usan direcciones de 4 bytes.

Paquete de búsqueda

Una vez presionado el pulsador del transmisor por más de 2 segundos se emitirá un paquete P_ID_SRCH a la dirección de broadcast en intervalos regulares de 0,5 segundos aproximadamente. Mediante el mismo procedimiento el receptor leerá dicha dirección habiendo preparado el correspondiente paquete de respuesta P_ID_SRCH_ACK. Una vez establecida la comunicación ambos terminales guardarán la dirección y datos referentes al formato de los paquetes en la memoria EEPROM, de forma que ambos dispositivos puedan apagarse y al ser encendidos nuevamente restablezcan la comunicación automáticamente. Ambos terminales quedarán vinculados hasta que se inicie nuevamente el modo búsqueda, ya sea por haber presionado el pulsador o por haber recibido el comando correspondiente (ver paquete de control). Tanto el transmisor como el receptor salen de modo búsqueda luego de 10 segundos sin encontrar respuesta. A su vez, lo hacen a mínima potencia para reducir el riesgo de vincular un dispositivo con otro que se encuentre en modo búsqueda dentro del rango de alcance.

Cantidad de Bytes:	2
Dirección:	0 x 33 00 00 01 (4 bytes)
Auto ACK:	Activado
Auto Retransmit Delay:	ARD_4000uS
Auto Retransmit Count:	ARC_15
Canal:	0 (2400Mhz)
Data Rate:	RF_DR_250kbps
Potencia:	RF_PWR_18dBm
CRC:	2 bytes

Tabla 3.2: Configuración del paquete de búsqueda.

P_ID	VERSION
------	---------

Tabla 3.3: Formato del paquete de búsqueda.

VERSION: versión del transmisor.

Paquete de respuesta

Al presionar el pulsador por más de 2 segundos, el receptor cargará el siguiente paquete mediante la instrucción W_ACK_PAYLOAD (para más información, ver hoja de datos del transceptor). En el mismo le indicará al transmisor todos los parámetros requeridos según el siguiente paquete:

Cantidad de Bytes:	10
Dirección:	0 x 33 00 00 01 (4 bytes)
Canal:	0 (2400Mhz)
Data Rate:	RF_DR_250kbps
Potencia:	RF_PWR_18dBm
CRC:	2 bytes

Tabla 3.4: Configuración del paquete de respuesta.

P_ID	ADDR0	ADDR1	ADDR2	ADDR3	RANGE_HPF	PWR_DR	TS	CHN
------	-------	-------	-------	-------	-----------	--------	----	-----

Tabla 3.5: Formato del paquete de respuesta.

ADDR3: Byte más significativo, indica el tipo de dispositivo, según la Tabla 3.6.

ADDR2-0: Número aleatorio generado por el receptor cada vez que se inicia el modo búsqueda. Una vez establecida la comunicación debe guardarse en EEPROM para retenerse en caso de apagado, tanto en el transmisor como en el receptor. Al iniciarse, se deberá cargar nuevamente.

RANGE_HPF: Indica el rango requerido de aceleración y habilita el filtro pasa alto (se suman los valores).

PWR_DR: Indica el nivel de potencia requerido y la velocidad de señalización. Se deberán sumar ambos valores antes de transmitir.

TS: Indica el periodo de muestreo requerido. Se calcula como:

$$T_S = 5ms + 1ms.TS \quad (3.1)$$

CHN: Indica el canal a utilizar, se calcula como:

$$Canal = 2400Mhz + 1Mhz.CHN \quad (3.2)$$

Paquete de datos

Este paquete es enviado periódicamente cada T_s segundos una vez recibido el paquete de respuesta (P_ID_SRCH_ACK). En todas las transmisiones se espera un ACK como respuesta. Si no lo recibe en ARD segundos, reenvía el paquete,

ADDR3	Dispositivo	Descripción
0x01	DEV_HID_MOUSE	Mouse USB
0x02	DEV_RC_CAR	Auto a Radio Control
0x03	DEV_WC	Silla de ruedas
0x04	DEV_TV	Televisor
0x33	DEV_SRCH	Reservado para paquetes de búsqueda
0xFF	DEV_GEN	Dispositivo genérico

Tabla 3.6: Campo ADDR3 según el tipo de dispositivo.

Rango	Etiqueta	Valor (Hex)
±2	RANGE_2G	0x00
±4	RANGE_4G	0x01
±8	RANGE_8G	0x02
HPF	Etiqueta	Valor (Hex)
Activado	HPF_OUT_ON	0x00
Desactivado	HPF_OUT_OFF	0x01

Tabla 3.7: Campo RANGE HPF.

un máximo de ART veces. Si se producen 10 transmisiones seguidas sin respuesta, el transmisor entrará en modo PAUSE. En el modo PAUSE el transmisor retiene los valores de configuración de forma de poder restablecer la comunicación de forma automática. Para ello envía el paquete de datos cada 0,5 segundos aproximadamente hasta que obtiene respuesta. Una vez obtenida la respuesta mediante un paquete ACK del tipo P_IP_CTRL lee el byte CMD y determina el modo siguiente.

NPT: Número de Paquete, variable que se incrementa cíclicamente entre 0 y 255, usada para llevar un control desde el receptor para comparar si hubo uno o más paquetes perdidos. Eventualmente se podrían interpolar los valores intermedios, o completar con el valor anterior o próximo.

OUT_X_MSB, OUT_X_LSB, OUT_Y_MSB, OUT_Y_LSB, OUT_Z_MSB, OUT_Z_LSB: Para independizar el valor recibido de la cantidad de bits del ADC del acelerómetro, los datos se alinean a la izquierda, de forma que el fondo de escala siempre es representado por el mismo valor, como muestra la Tabla 3.11.

En el caso del MMA8451Q usa un ADC de 14 bits, pero el sistema permitiría hasta 16bits sin modificaciones. Si se requieren solo 8 bits, simplemente se deben leer los respectivos MSB de cada eje.

Paquete de control

Este paquete es usado en respuesta al paquete de datos P_ID_DATA, enviado desde el receptor hacia el transmisor como ACK. En el mismo se envía un comando para controlar el transmisor con las características que se pueden ver en la Tabla 3.12.

CMD: Comando de respuesta, utilizado para enviar órdenes desde el receptor hacia el transmisor, según la Tabla 3.14.

Potencia	Valor (Hex)
0 dBm	0x06
-6 dBm	0x04
-12 dBm	0x02
-18 dBm	0x00
Data Rate	Valor (Hex)
RF_DR_2000kbps	0x08
RF_DR_1000kbps	0x00
RF_DR_250kbps	0x20

Tabla 3.8: Campo PWR DR.

Cantidad de Bytes:	8
Dirección:	[ADDR3 ADDR2 ADDR1 ADDR0] (Recibida dentro del paquete de respuesta)
Auto ACK:	Activado
Auto Retransmit Delay:	ARD_1000uS
Auto Retransmit Count:	ARC_5
Canal:	CHN (Recibido dentro del paquete de respuesta)
Data Rate:	DR (Recibido dentro del paquete de respuesta)
Potencia:	PWR (Recibido dentro del paquete de respuesta)
CRC:	1 bytes

Tabla 3.9: Configuración del paquete de datos.

P_ID	NPT	OUT_X_LSB	OUT_X_MSB	OUT_Y_LSB	OUT_Y_MSB	OUT_Z_LSB	OUT_Z_MSB
------	-----	-----------	-----------	-----------	-----------	-----------	-----------

Tabla 3.10: Formato del paquete de datos.

3.3. Transmisor

El dispositivo esencial de este sistema es el transmisor, que se encarga de sensar los valores de aceleración y transmitirlos por radio frecuencia. Este dispositivo se coloca en la cabeza del usuario, a través de una vincha. Inicialmente se implementó el circuito sobre una placa experimental, aunque para reducir el peso y volumen del dispositivo se realizó el diseño de un circuito impreso utilizando componentes de montaje superficial. Este último diseño no ha sido fabricado sino que se toma a modo de referencia para un eventual producto final.

3.3.1. Circuito esquemático

En el apéndice A.1 se puede ver el circuito del transmisor desarrollado. Es esencialmente un circuito digital, que se compone de tres circuitos integrados: un microcontrolador, un sensor de aceleración y un transceptor de radio frecuencia. Si bien los últimos dos se muestran en forma de bloques, sus circuitos internos fueron extraídos de sus respectivas hojas de datos. Para la interacción con el usuario se dispone de dos leds, para indicar el estado de la comunicación, y un pulsador para cambiar entre modos de funcionamiento. El circuito se alimenta

MSB	LSB	Valor Decimal	Equivalente	Ej: Rango=2G
0x7F	0xFF	32767	+Rango	+2G
0x3F	0xFF	16383	+Rango/2	+1G
0x00	0x00	0	0	0G
0xC0	0x00	-16384	-Rango/2	-1G
0x80	0x00	-32768	-Rango	-2G

Tabla 3.11: Ejemplos de valores enviados en el paquete de datos.

Cantidad de Bytes:	4
Dirección:	[ADDR3 ADDR2 ADDR1 ADDR0](Enviado dentro del paquete de respuesta)
Canal:	CHN (Enviado dentro del paquete de respuesta)
Data Rate:	DR (Enviado dentro del paquete de respuesta)
Potencia:	PWR (Enviado dentro del paquete de respuesta)
CRC:	1 byte

Tabla 3.12: Configuración del paquete de control.

P_ID	CMD	RANGE_HPF*	TS*
------	-----	------------	-----

Tabla 3.13: Formato del paquete de control. *Solo en el comando PLAY y PAUSE.

mediante dos baterías AAA de 1.5V cada una, a las cuales se les interpone un interruptor tipo deslizante para prender y apagar el dispositivo. Por último dispone de un conector ICSP para la programación del microcontrolador y un conector mini USB, que si bien actualmente no se utiliza, se colocó para un posible uso futuro.

3.3.2. Prototipo experimental

Debido al costo y complejidad que significa fabricar un circuito con tecnología SMD en bajos volúmenes, se optó por realizar un prototipo en placa experimental. Los circuitos integrados del sensor y transceptor se implementaron en módulos ya soldados, con conectores de tipo estandar, es decir de paso 0.1", como se puede ver en la figura 3.2. El microcontrolador utilizado se encuentra disponible tanto en encapsulados SMD como DIP, con lo cual para el prototipo se eligió este último.

En la figura 3.3a se puede ver el prototipo ensamblado en placa experimental, con sus dos pilas soldadas entre si y a dos cables que se conectan a la placa. Se pueden ver también los módulos del sensor y del transceptor respectivamente. El prototipo completo se puede ver en la figura 3.3b, que consiste en una caja plástica junto a una vincha elástica, ajustable, que se coloca en la cabeza del usuario.

Comando	Valor	Descripción
CMD_PLAY	0x11	Indica al transmisor que debe continuar enviando datos, o que debe volver a enviar datos si se encontraba en modo PAUSE.
CMD_PAUSE	0x22	Indica al transmisor que debe ingresar en modo PAUSE.
CMD_STOP	0x33	Indica al transmisor que debe dejar de enviar datos permanentemente, no se restablecerá la comunicación hasta que vuelva a iniciar el proceso de búsqueda mediante el pulsador.
CMD_SRCH	0x44	Indica al transmisor que debe iniciar el modo búsqueda. Este comando puede ser utilizado para modificar parámetros de la transmisión automáticamente sin intervención por parte del usuario.
CMD_RST	0x55	Indica al transmisor que debe enviar un paquete de búsqueda a la dirección previamente obtenida. Este comando es similar al comando CMD_SRCH pero evita el riesgo de que el transmisor se asocie con otro receptor que se encuentre en modo búsqueda y esté dentro del rango de alcance. Nota: este comando no fue implementado. En cambio, se agregaron los campos RANGE_HPF y TS en cada paquete de control para cambiar los parámetros de adquisición de forma directa.

Tabla 3.14: Opciones para el campo CMD.

3.3.3. Circuito impreso

Se desarrollaron dos versiones del circuito impreso, una para pequeña escala, y otra para mediana o gran escala. La primera, utiliza los mismos módulos que el transmisor hecho en placa experimental, lo cual simplifica el diseño ya que los componentes de menor tamaño y de radiofrecuencia se encuentran en los módulos comerciales. En la figura 3.4 se puede ver el circuito impreso y su vista 3D.

Para reducir el tamaño y peso del transmisor se diseñó un segundo circuito impreso que integra todos los componentes en una misma placa. Para ello se debió diseñar el circuito usando componentes de montaje superficial íntegramente. En la figura 3.5 se puede ver un boceto del mismo, donde el ruteo (trazado de caminos) está hecho a modo de ejemplo y no representa una versión final. En la figura 3.6 se puede ver un render 3D de la placa. Las pilas AAA se encuentran conectadas directamente a la placa, en el lado posterior, mediante unos contactos específicos para circuitos impresos.

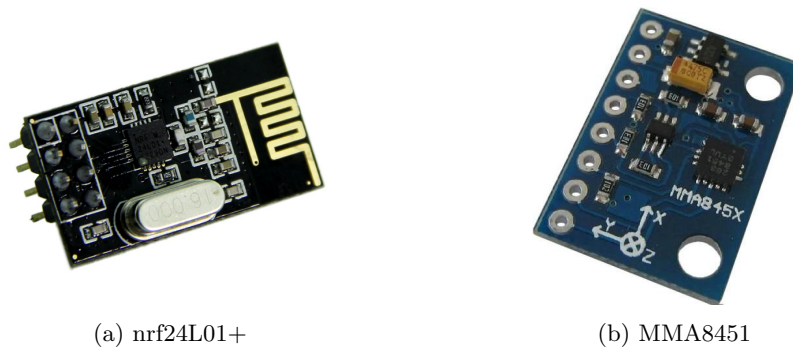


Figura 3.2: Módulos utilizados en el prototipo experimental.

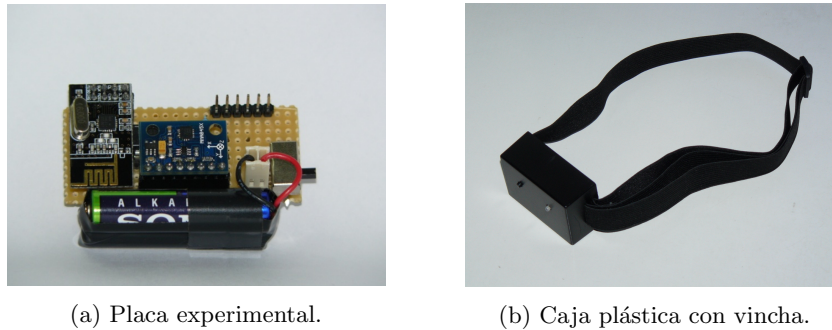


Figura 3.3: Primer prototipo del transmisor.

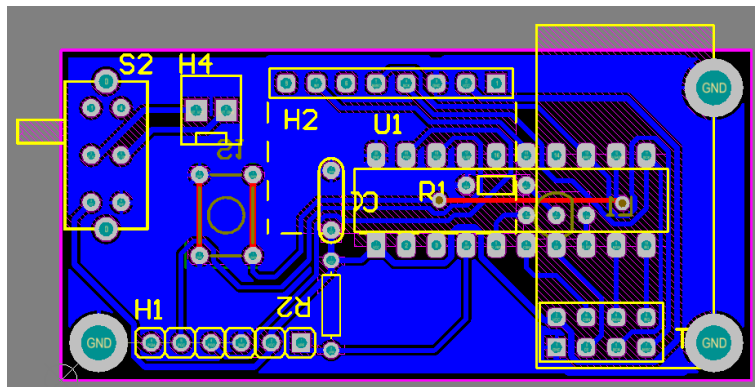
3.3.4. Programa

El programa que ejecuta el transmisor consiste en la realización práctica del protocolo de comunicación desarrollado en la sección 3.2. Fue desarrollado íntegramente en C mediante el entorno de desarrollo MPLAB[®] IDE 8.87 junto al compilador C18, ambos de Microchip[®].

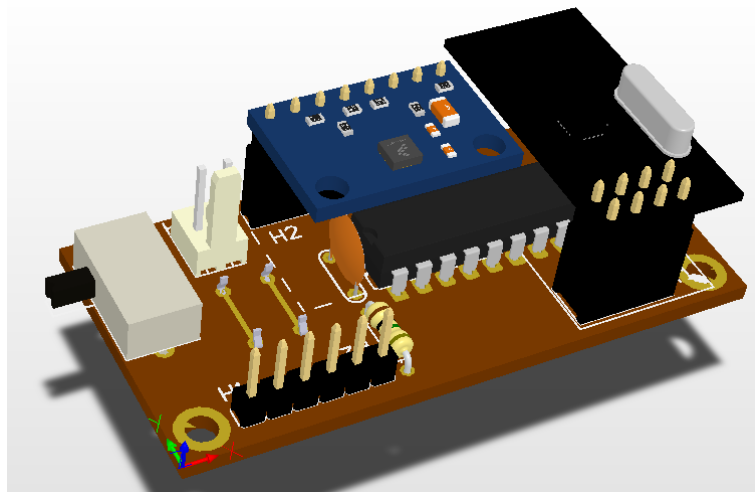
El algoritmo se basa íntegramente en interrupciones. El microcontrolador PIC18F14K50 utilizado dispone de dos niveles de prioridad para las interrupciones: alta y baja prioridad. La interrupción de baja prioridad es utilizada para determinar el estados de los leds en función del tiempo y del modo en que se encuentra el transmisor. La interrupción de alta prioridad es utilizada esencialmente para transmitir paquetes. Ambas interrupciones se activan mediante timers independientes. Si el programa está atendiendo una interrupción de baja prioridad, se puede producir una interrupción de alta prioridad, sin embargo, si se está atendiendo una interrupción de alta prioridad, no se atenderán las interrupciones de baja prioridad hasta finalizar la atención de la primera.

Interrupción de baja prioridad

La interrupción de baja prioridad es llamada cada 100ms, y verifica el estado del pulsador y determina el estado de las luces. Si el pulsador se encuentra presionado, incrementa una variable. Si no se encuentra presionado comprueba el valor de esa variable: si es menor que 20 (equivale a 2 segundos) no ejecuta



(a) Vista 2D.



(b) Vista 3D.

Figura 3.4: Circuito impreso del prototipo del transmisor.

ninguna tarea. Si es mayor a 20 y el dispositivo se encuentra en modo Stop, Play o Pausa pasa a modo Búsqueda. Si se encuentra en modo Busqueda, pasa a modo Stop. También se encarga de salir de modo búsqueda luego de 20 segundos de no encontrar un receptor disponible. En cuando a luces, depende del modo en que se encuentre:

Modo Stop: un destello rojo cada dos segundos.

Modo Play: un destello verde cada dos segundos.

Modo Pausa: un destello de ambos colores cada dos segundos.

Modo Búsqueda: un destello de cada color intercalados cada un segundo.

Interrupción de alta prioridad

La interrupción de alta prioridad es llamada con diferentes periodos según el modo de funcionamiento:

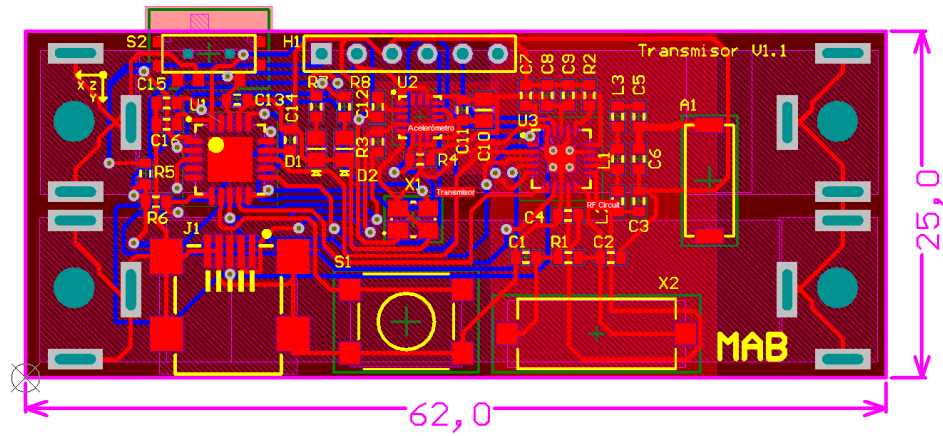
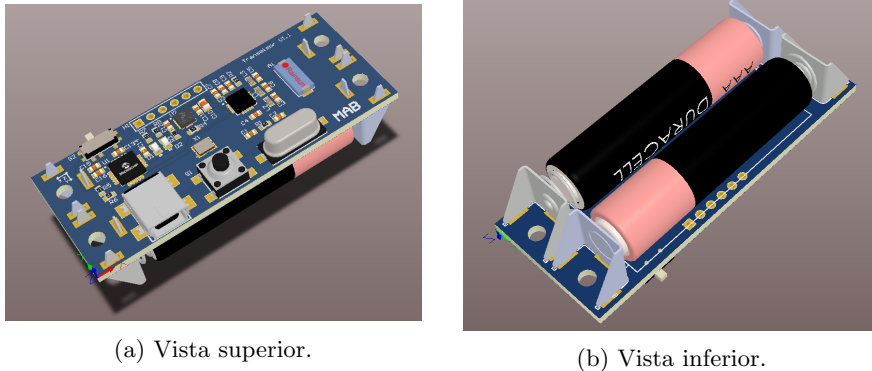


Figura 3.5: Circuito impreso tentativo del transmisor.



(a) Vista superior.

(b) Vista inferior.

Figura 3.6: PCB tentativo del transmisor.

Modo Stop: no se llama a esta interrupción, ya que se encuentra desactivado.

Modo Play: se llama con un periodo entre 5ms y 260ms dependiendo del requerimiento del receptor.

Modo Pausa: se llama cada 500ms aproximadamente.

Modo Búsqueda: se llama cada 500ms aproximadamente.

También, dependiendo del modo de funcionamiento se ejecutan distintas tareas:

Modo Play: primero se lee la información del sensor de aceleración por el puerto I2C, luego se arma el paquete de datos y se transmite por SPI al transceptor. Se espera a que el transceptor responda mediante el pin IRQ y se consulta el resultado de la operación. Si la operación fue satisfactoria se verifica el paquete de respuesta, es decir el paquete de control. Dicho paquete, como indica la sección 3.2.1, puede ajustar los parámetros de transmisión (rango y periodo de muestreo), así como obligar al transmisor

a cambiar de modo. En cambio, si la operación no fue satisfactoria, es decir se llegó al máximo de reenvíos sin obtener respuesta, se incrementa una variable que al llegar a 10 paquetes transmitidos sin éxito, se pasa a modo pausa. Al primer paquete de respuesta obtenido, se reinicia dicho contador y se transmite en modo play nuevamente.

Modo Pausa: se procede de forma similar a modo play, solo que el periodo con que se envían los paquetes pasa a ser de aproximadamente 500ms.

Modo Búsqueda: en este modo se envía un paquete de búsqueda según se describe en la sección 3.2.1 con el objetivo de determinar si hay receptores disponibles. Una vez enviado el paquete se espera que el transceptor responda mediante una IRQ el estado de la operación. Si no hubo respuesta, no se hace nada hasta el próximo intervalo de búsqueda, es decir dentro de 500ms. Si hubo respuesta, se verifican los datos del ACK, es decir del paquete de respuesta a la búsqueda descrito en la sección 3.2.1. Si el paquete es correcto se guardan los datos en memoria EEPROM y se activa un flag indicando que hay un dispositivo sincronizado. Luego, se ingresa a modo play con los parámetros recién obtenidos.

Para facilitar las atenciones a las interrupciones ya descriptas, se crearon seis funciones auxiliares descriptas en la Tabla 3.15.

Función	Descripción
Use_Spi();	Como el puerto SPI y el I2C comparten pines del microcontrolador, se debe configurar el puerto antes de cada uso. Esta función configura el puerto serie del microcontrolador en modo SPI.
Use_I2C();	Esta función configura el puerto serie del microcontrolador en modo I2C.
Modo_Busqueda();	Ingresa en modo búsqueda. Configura el timer en 500ms y el transceptor según lo explicado en la sección 3.2.1.
Modo_Play();	Ingresa en modo play. Configura el timer según lo requerido y el transceptor según lo explicado en la sección 3.2.1.
Modo_Pause();	Ingresa en modo pausa. Configura el timer en 500ms y el transceptor según lo explicado en la sección 3.2.1.
Modo_Stop();	Ingresa en modo stop. Desactiva el transceptor y el timer.

Tabla 3.15: Funciones auxiliares del transmisor.

3.4. Receptor MATLAB®

Este dispositivo permite una interfaz con el programa Matlab® a través del puerto USB. No es una aplicación final, sino una herramienta que fue utilizada durante el desarrollo del proyecto. Permite no solo recibir información del sensor de aceleración, sino también el control del mismo desde la PC, tal como puede

hacer cualquier otro receptor. Está basado en un PIC18F2550 el cual posee puerto USB incorporado en el mismo.

3.4.1. Circuito esquemático

En el apéndice A.2 se pueden ver los componentes del sistema: un microcontrolador PIC18F2550, un bloque de RF idéntico al del transmisor, dos luces LED, un pulsador, un conector ICSP y un conector USB. La alimentación de 5V se toma directamente del puerto USB, y los 3.3V necesarios para alimentar el transceptor de RF se logran a través de un regulador lineal LM78L33. Nuevamente la comunicación con el transceptor de RF se logra mediante un puerto SPI integrado en el microcontrolador. Este microcontrolador se encuentra sobre dimensionado para la sencilla tarea que realiza. Sin embargo, fue conveniente su uso en una etapa de desarrollo debido a que puede ser utilizado en modo depuración. En este modo se puede ejecutar el programa desarrollado paso a paso o mediante bloques, desde el mismo microcontrolador. Permite entonces una especie de simulación pero en hardware, de forma de poder ver el contenido de las variables, instante a instante. Mediante este método resultó mucho más conveniente la depuración del programa cuando se presentaban errores.

3.4.2. Prototipo experimental

Al igual que el transmisor, se desarrolló un primer prototipo en placa experimental, como se puede ver en la figura 3.7. Se puede observar el conector USB tipo A el cual se conecta directamente al puerto USB de una computadora, tal como lo hace una memoria flash.

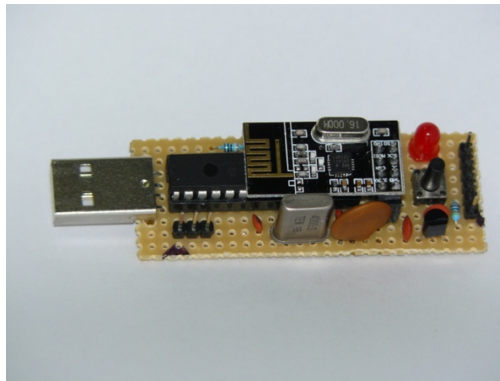


Figura 3.7: Primer prototipo del receptor Matlab.

3.4.3. Circuito impreso

Al ser un circuito utilizado únicamente en la etapa de desarrollo, no se diseñó ni se construyó un circuito impreso del mismo. Tampoco se planea hacerlo en una etapa posterior.

3.4.4. Programa

El programa se puede dividir en dos partes: la primera con todo lo relacionado con la comunicación por RF y la implementación descrita en la sección 3.2, y la segunda parte relacionada con la comunicación USB hacia Matlab. Se utilizó nuevamente un timer de baja prioridad para el control de las luces y el pulsador, mientras que la interrupción de alta prioridad se reservó para el protocolo USB. El programa principal se encarga de verificar la llegada de nuevos datos y transmitirlos por USB, en los casos que corresponde hacerlo.

Para la primer parte, es decir para implementar el protocolo de comunicación, se crearon cuatro funciones para entrar en los respectivos modos de funcionamiento: play, pausa, stop y búsqueda. En este caso, las funciones se desarrollaron de forma paramétrica, es decir que se ingresan parámetros de entrada para modificar el comportamiento de la función. Esto se debe a que el receptor es quien fija los parámetros de la comunicación. La función `Modo_Busqueda` se debe llamar ingresando `ADDR0`, `ADDR1`, `ADDR2`, `RANGE_HPF`, `PWR_DR`, `TS` y `CHN`, es decir los parámetros del paquete de respuesta que se detalla en la sección 3.1. Las funciones `Modo_Play` y `Modo_Pause` permiten ingresar con los parámetros `RANGE_HPF` y `TS`, para indicarle al transmisor que debe cambiar dichos valores. La función `Modo_Stop` es igual a la del transmisor, ya que desactiva el transceptor de RF totalmente.

Para la segunda parte, es decir la comunicación USB, se tomo como referencia el programa de muestra en las librerías de Microchip®. Para facilitar la lectura inicialmente se hizo una limpieza de los archivos principales, es decir, se eliminó todo el contenido que no era esencial para el uso que se le iba a dar. Se pueden destacar las siguientes funciones:

USBDeviceInit() Inicializa la pila USB en su estado por defecto, reinicia el módulo y borra todas las variables, registros e indicadores de interrupción.

USBDeviceAttach() Indica al host (PC) que se ha conectado un dispositivo al bus. Es necesario llamar esta función para iniciar la enumeración en el bus.

USBGenRead(BYTE ep, BYTE* data, WORD len) Lee la información recibida en el endpoint "ep", de longitud "len", a guardarse en la variable "data".

USBGenWrite(BYTE ep, BYTE* data, WORD len) Transmite información a través del endpoint "ep", la información de la variable "data", de largo "len".

USBDeviceTasks() Esta es la función principal del módulo USB, se encarga de transmitir y recibir información, detectar eventos, etc. Si se trabaja en modo encuesta (`USB_POLLING`), se debe llamar periódicamente cada 1.8ms aproximadamente. Si se trabaja con interrupciones no es necesario llamarla de forma manual, sino que una interrupción de alta prioridad se encarga de ejecutar esta instrucción.

ServiceRequests() Se encarga de atender los pedidos del host (PC), interpretando sus comandos y respondiendo según lo solicitado.

En la Tabla 3.16 se puede ver una lista de comandos a los que responde la función `ServiceRequests()` vista anteriormente.

Comando	Valor (Hex)	Descripción
READ_VERSION	0x00	Solicita el número de versión.
START	0x1E	Indica que comience a transmitir información por USB, si se encuentra en modo Play y recibe datos del sensor.
STOP	0x1F	Indica que no envíe información por USB en caso de recibir datos.
CMD_USB_BUSQUEDA	0x20	Indica al receptor que entre en Modo Búsqueda.
CMD_USB_STOP	0x21	Indica al receptor que entre en Modo Stop.
CMD_USB_PLAY	0x22	Indica al receptor que entre en Modo Play.
CMD_USB_PAUSE	0x23	Indica al receptor que entre en Modo Pausa.

Tabla 3.16: Comandos USB para controlar el receptor.

3.4.5. Interfaz Matlab[®]

Con el objetivo de poder analizar, procesar y visualizar los movimientos realizados con el sensor, se realizó una interfaz mediante el programa Matlab[®]. Se accedió a los mismos mediante código, es decir no se implementó una interfaz gráfica, ya que este dispositivo receptor no contempla ningún uso por parte del usuario final, sino que es utilizado como una herramienta de desarrollo propia.

Para su implementación se trabajó con una librería provista por Microchip[®], llamada "mpusbapi.dll". Un resumen de las funciones principales se puede ver a continuación:

MPUSBGetDeviceCount(PCHAR pVID_PID) Para un determinado Vendor ID y Product ID, devuelve la cantidad de dispositivos conectados y enumerados.

MPUSBOpen(instance,pVID_PID,pEP,dwDir,dwReserved) Abre la comunicación USB del dispositivo dado por el Vendor ID y Product ID, en el endpoint pEP.

MPUSBRead(handle,pData,dwLen,pLength,dwMilliseconds) Función utilizada para leer información recibida del puerto USB. Se carga en la variable pData, cuyo largo es dwLen.

MPUSBWrite(handle,pData,dwLen,pLength,dwMilliseconds) Función utilizada para enviar información al puerto USB. Se envía la variable pData, cuyo largo es dwLen.

El programa inicialmente comprueba que existan dispositivos conectados. En ese caso abre una conexión en el endpoint 0 para transmitir, y en el endpoint

1 para recibir. Luego envía el comando START descrito en la Tabla 3.16. Por último, espera a recibir nuevos datos y a medida que los recibe los almacena en un vector, de longitud creciente, a la vez que los grafica en función del tiempo, como se ve en la Figura 3.8. La primer fila corresponde al indicador de número de paquete, para control de orden, que varía entre 0 y 255. Los datos obtenidos y graficados en las tres filas inferiores corresponden a los valores de aceleración de los ejes X, Y y Z respectivamente, para distintos movimientos del sensor.

3.5. Receptor Mouse

Como primera aplicación concreta, se desarrolló un receptor capaz de emular un mouse USB completo. Inclinando la cabeza hacia los costados, o hacia arriba o abajo, el usuario puede mover el cursor de una PC. Mediante un tipo especial de movimiento, también se puede hacer click izquierdo y derecho, así como mover el scroll hacia arriba y abajo. No requiere configuración de la PC, ni instalación de *drivers*.

3.5.1. Circuito esquemático

En el apéndice A.3 se puede observar el circuito esquemático de este receptor. Es básicamente idéntico al Receptor MATLAB®, con la salvedad de usar otro microcontrolador, el PIC18F14K50, el mismo del transmisor. Esto permitió reducir el costo y tamaño del receptor significativamente. Por otro lado, en una versión de producción, permitiría elegir un encapsulado QFN de forma de reducir aún más el tamaño del dispositivo. Por último, el hecho de usar el mismo microcontrolador que el transmisor permite facilitar los procesos de compra y logística en caso de producirse a mediana o gran escala.

3.5.2. Prototipo experimental

Como en los anteriores dispositivos, se implementó este receptor en una placa experimental como se puede ver en la figura 3.9. Si bien el puerto USB trabaja a 12Mhz, frecuencia que no es recomendable en placas experimentales, no se han detectado problemas de funcionamiento.

3.5.3. Circuito impreso

Al igual que con el transmisor, se diseñó un circuito impreso intermedio para pequeña escala, usando una placa sencilla de una sola capa, dejando el circuito de radiofrecuencia a cargo de un módulo externo. De esta forma se puede acceder de forma económica y rápida a la fabricación de una pequeña escala de dispositivos, que pueden ser usados durante la etapa de pruebas de campo. En la figura 3.10 se puede observar el diseño del PCB y una vista 3D del mismo.

3.5.4. Programa

El programa que ejecuta este receptor se puede dividir en tres partes:

1. Implementación del protocolo de comunicación de radiofrecuencia.

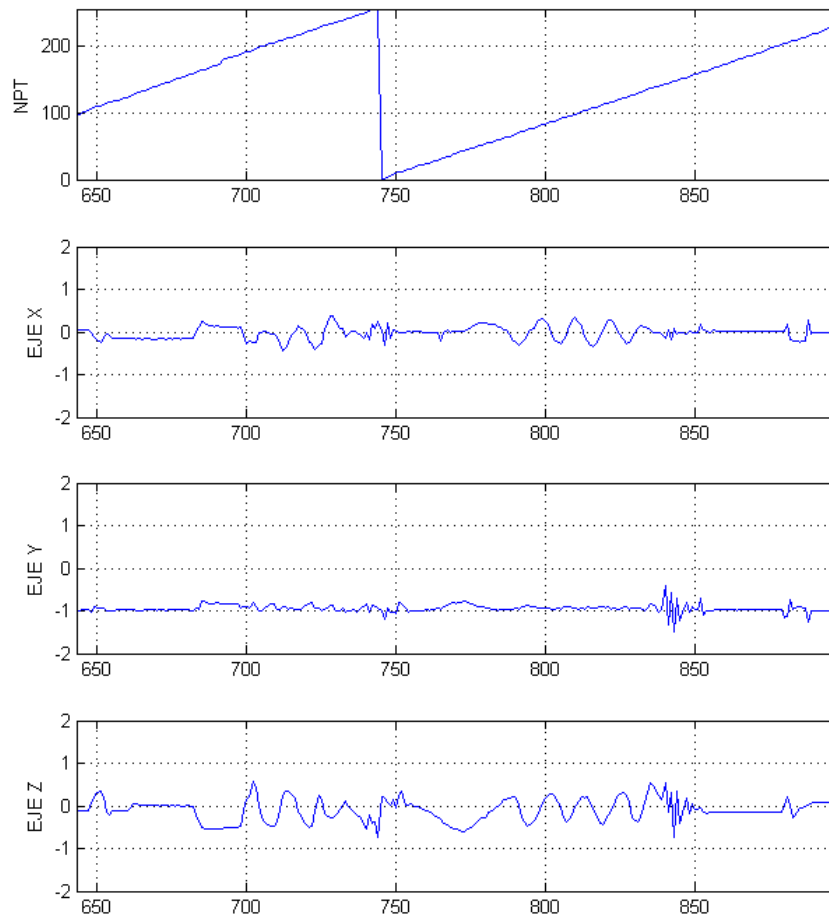


Figura 3.8: Datos de aceleración recibidos directamente en Matlab.

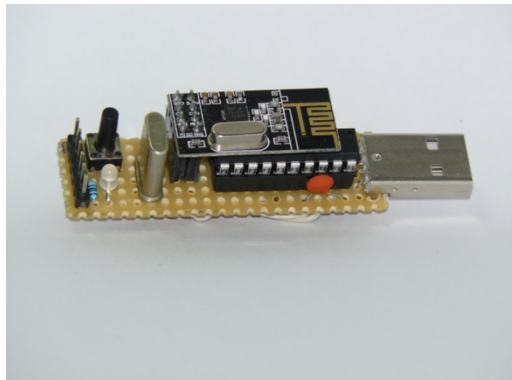


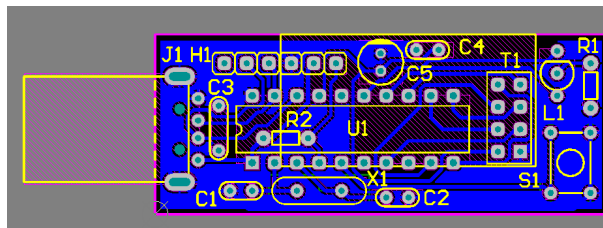
Figura 3.9: Primer prototipo del receptor Mouse.

2. Análisis y procesamiento de la información recibida.
3. Envío de comandos de mouse a través del puerto USB.

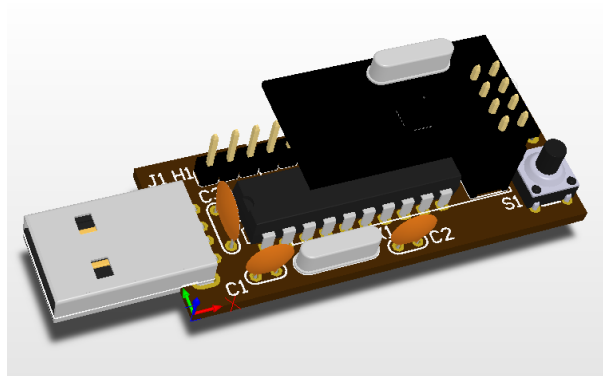
El protocolo de comunicación se implementó igual que el receptor Matlab, es decir siguiendo los lineamientos de la sección 3.2. En este caso se le solicitó al transmisor que envíe los datos de aceleración 20 veces por segundo, con un fondo de escala de $\pm 2G$, y sin filtro pasa altos (ya que se debe sentir la inclinación de la cabeza respecto del vector de aceleración terrestre de la gravedad \vec{G}).

Como se ve en la figura 3.11, el eje Y coincide con el vector \vec{G} , el eje X coincide con inclinaciones laterales y el Z con inclinaciones hacia adelante y hacia atrás. Por lo tanto, solo se tuvieron en cuenta estos ejes para determinar completamente la inclinación de la misma. Inicialmente se utilizó el Receptor Matlab para analizar con mayor facilidad los valores recibidos en función de los movimientos realizados. Para ello se graficó el eje X en función del -Z (el signo menos, para que sea positivo cuando se inclina la cabeza hacia arriba), como se ve en la figura 3.12. El punto rojo del centro representa el valor actual de inclinación. Una inclinación de 90° hacia la derecha, llevará el punto rojo hacia (1,0), una inclinación a la izquierda de 90° lo llevará a (-1,0) El centro de la figura, corresponde a la cabeza perfectamente centrada. Luego, se observan 4 círculos de radios R1, R2, R3 y R4 respectivamente, los cuales dividen el espacio en cuatro diferentes zonas.

Para el control del mouse se utilizó una máquina de estados, la cual se observa en la figura 3.13. La transición entre estados se ejecuta cada vez que se recibe un nuevo dato. Se calcula el radio ($radio = \sqrt{X^2 + (-Z)^2}$) dado por las coordenadas del punto (X,-Z) y en función de dicho valor se determina la acción y el siguiente estado. Inicialmente se encuentra en estado Stop, y permanece ahí mientras el radio sea menor a R2, es decir con la cabeza derecha, en la posición de reposo. Una vez que se aparta la cabeza de la posición de reposo, cambia al estado Count, que básicamente se encarga de incrementar el contador en una unidad cada vez que se recibe un dato nuevo. Si el contador llega al valor Nmov, se interpreta que se deberá mover el cursor del mouse. En este caso, Nmov=20, y como se reciben 20 datos por segundo, se deberá esperar 1 segundo para mover el cursor. Mientras el punto se encuentre afuera del círculo R2 se seguirá moviendo el cursor. Una vez que vuelve a la posición de reposo,



(a) Vista 2D.



(b) Vista 3D.

Figura 3.10: Circuito impreso del prototipo del Receptor Mouse.

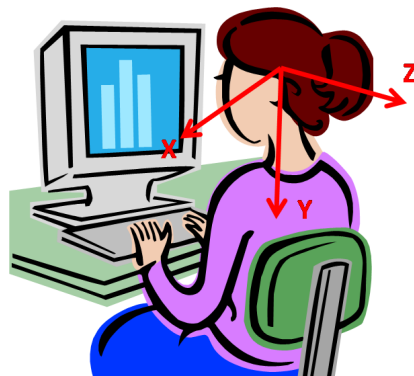


Figura 3.11: Ubicación de los ejes respecto de la posición de la cabeza.

se inicia un contador que una vez que llega a N_{stop} , vuelve al estado Stop. Si el punto sale de R2 antes que se alcance N_{stop} , volverá inmediatamente a mover el cursor y dejar de contar. Esto es así para que el usuario pueda cambiar de dirección el movimiento del cursor pasando por el círculo central, sin tener que volver a esperar que el contador principal llegue hasta N_{mov} .

En el caso de que se encuentre en el estado Count y entre en el círculo R1 antes de que el contador alcance N_{mov} , se interpretará que se quiere hacer un gesto. Los gestos posibles son: click derecho, click izquierdo, scroll hacia arriba, y scroll hacia abajo. Para ello se deberá inclinar la cabeza hacia la derecha, izquierda, arriba, y abajo respectivamente, y volver a reposo rápidamente. Para

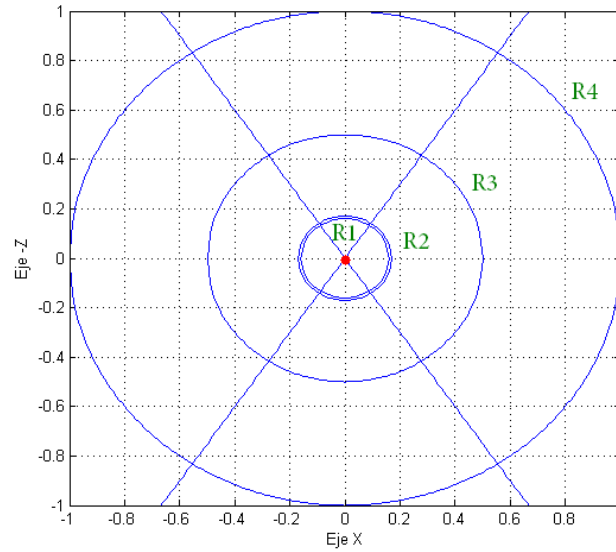


Figura 3.12: Determinación de la inclinación de la cabeza.

determinar que tipo de gesto debe hacerse, se establecieron cuatro cuadrantes, que se dividen por las diagonales que se observan en la figura 3.12.

Para determinar la velocidad de movimiento del cursor, se establece una velocidad mínima v_{min} , en este caso 1, y una velocidad máxima v_{max} , en este caso 15. La velocidad mínima corresponde al radio R2, y la máxima corresponde al radio R3. Luego de R3 la velocidad se mantiene en v_{max} . Los valores de v_{min} y v_{max} corresponde a la cantidad de píxeles que se deberá desplazar el cursor con cada muestra recibida. Es decir si se inclina la cabeza levemente, el cursor se moverá lentamente, y a medida que se aumenta la inclinación, aumenta la velocidad linealmente hasta alcanzar v_{max} cuando se alcanza R3.

3.6. Receptor RC

El desarrollo de un receptor para un auto a radio control cumple dos propósitos: por un lado puede servir como elemento lúdico para que usuarios del sistema puedan entretenerse mientras se familiarizan con el dispositivo, y por otro lado sirve como un simulador para el posterior desarrollo del control de una silla de ruedas. Se desarrolló un prototipo totalmente funcional cuyos elementos mecánicos se basan en un kit para armar de robótica, y cuya electrónica se desarrolló sobre un circuito impreso simple faz.

3.6.1. Circuito esquemático

En el apéndice A.4 se observa el circuito esquemático de este receptor. Consiste en un microcontrolador PIC18F2550 que se comunica con un módulo de transmisión nRF24L01+, dos luces Led y un pulsador igual que los demás módulos del sistema. Además posee un zumbador, para emitir una señal sonora que

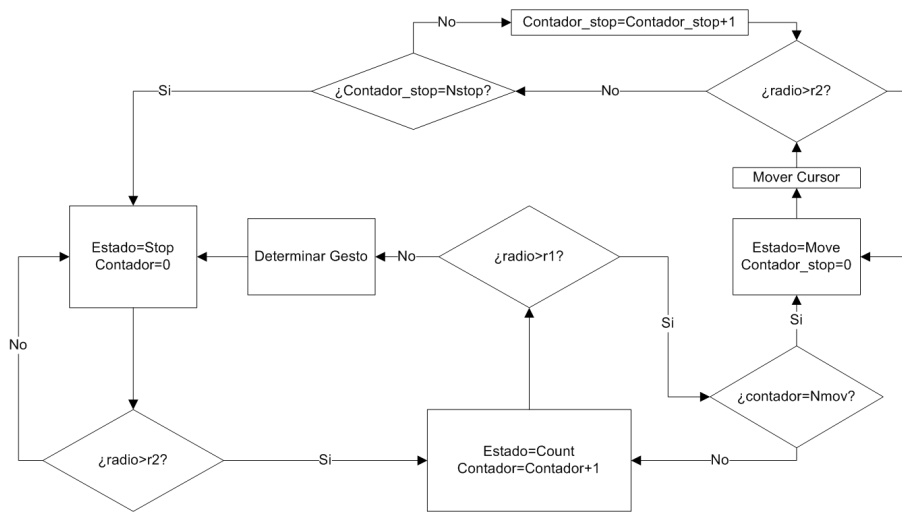


Figura 3.13: Algoritmo ejecutado por el receptor.

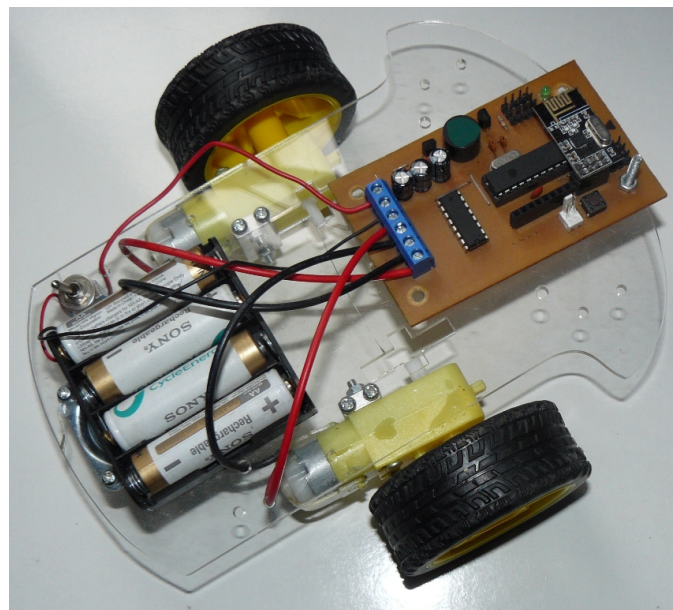


Figura 3.14: Prototipo de auto a radio control ensamblado.

sirva para alertar al usuario de determinados eventos. El control de los motores se hace a través del circuito integrado L293D, el cual incorpora dos puentes H completos. Es decir, permite alimentar dos motores en ambos sentidos, para poder girar en sentido horario y antihorario de forma independiente cada rueda. Para determinar la velocidad, se ingresa al puente H con señales moduladas por ancho de pulso, o PWM. Estas señales se caracterizan por ser de forma cuadrada, de frecuencia constante pero de ciclo de trabajo variable entre 0 y 100 por ciento. De esta forma, debido al efecto pasa bajos que produce un motor de

corriente continua, se varía el valor medio que se le aplica al motor.

En la figura 3.15 se observa el circuito de entrada y salida de cada canal del L293D. Si la entrada se conecta a V_{cc1} , a la salida obtengo V_{cc2} , y si la entrada la conecto a 0V, a la salida obtengo 0V, pero siempre a través de los transistores. Por lo tanto, si se conecta cada terminal de un motor de corriente continua a un circuito como este, se puede controlar la dirección de rotación del mismo, intercalando V_{cc2} y 0V en cada terminal del motor.

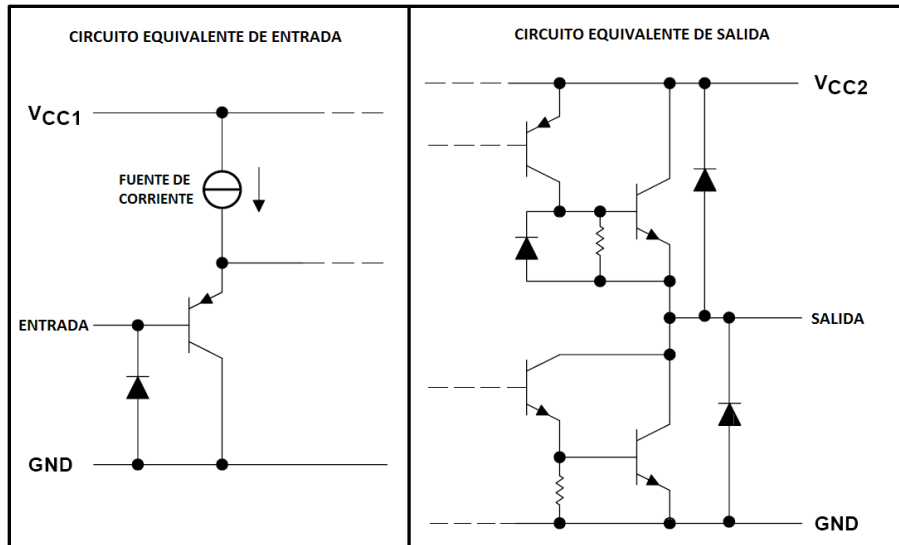


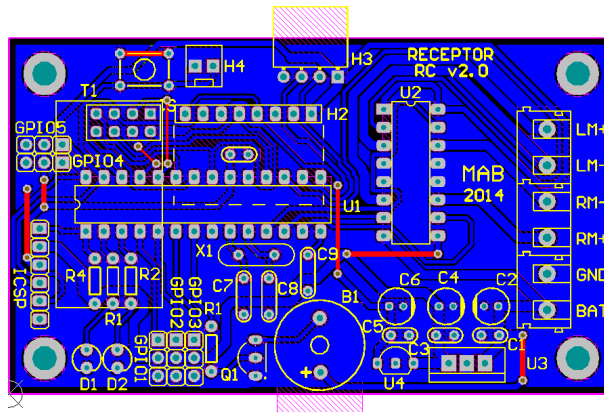
Figura 3.15: Circuito de entrada y salida de cada bloque del L293D.

3.6.2. Circuito impreso

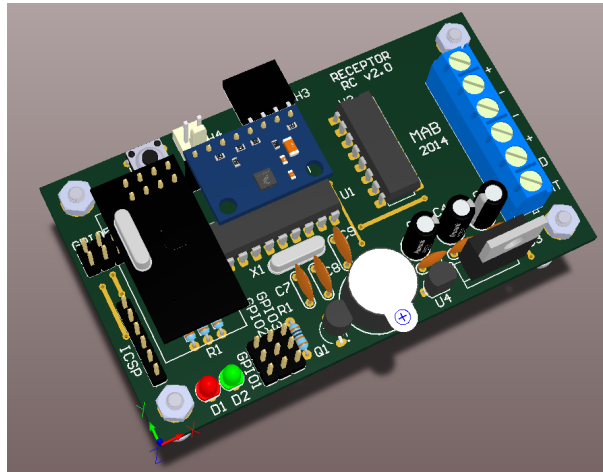
En la figura 3.16 se observa un diseño del circuito impreso y su vista 3D. Se puede ver que también se incluye la posibilidad de conectarle un sensor de aceleración adicional, idéntico al que usa el transmisor, por si llega a resultar necesario su uso en el control de una silla de ruedas. Se usan conectores tipo borneras para conectar la batería y los dos motores de las ruedas. El circuito integrado L293D dispone de sus cuatro pines centrales conectados a masa, los cuales a su vez permiten transferir calor hacia el circuito impreso para disminuir la resistencia térmica del mismo. Por tal motivo, se puede observar que el plano de masa se extendió lo más posible en esa zona para aumentar el área de cobre.

3.6.3. Programa

En primera medida se ha desarrollado la rutina que controla la velocidad y dirección de los motores. Para ello se utilizaron dos salidas PWM y dos salidas convencionales, para regular la velocidad y dirección de cada motor, respectivamente. Se ingresa a la función con dos valores de velocidad, una para cada motor, entre 1000 y -1000. Valores positivos representan el giro en un sentido, y valores negativos en el otro. Para detener el motor completamente se debe ingresar con 0. La función se encarga de generar dos ondas cuadradas de 500Hz



(a) Circuito impreso.



(b) Proyección 3D.

Figura 3.16: Circuito impreso del receptor RC.

a través del módulo PWM integrado en el microcontrolador. Si se exceden los valores 1000 y -1000 se los limita a sus valores máximos. Para valores positivos, se coloca en 0V la salida convencional (que define la dirección), y se varía entre 0 y 100 por ciento el ciclo de trabajo de la otra salida. Para valores negativos, la salida convencional se coloca a VCC, y se invierte el ciclo de trabajo de la salida PWM, es decir entre 100 y 0 por ciento. De esta forma, si se desea máxima velocidad en sentido inverso, se ingresa con VCC a la salida convencional, y como el ciclo de trabajo es 0 por ciento, se obtiene una salida en 0V constantes. Para valores intermedios, por ejemplo -300, se genera una salida con un ciclo de trabajo de 70 por ciento, de forma que solo circule corriente por el motor durante el periodo negativo de la onda cuadrada, es decir el 30 por ciento del periodo completo.

El resto del programa se encarga de implementar el protocolo de comunicación de igual forma que lo hacen los demás receptores. La diferencia radica en la forma de interpretar los datos de aceleración obtenidos. Para determinar la

velocidad (a través del ciclo de trabajo de las señales PWM) en función de la posición de la cabeza se realiza el siguiente cálculo:

$$PWM_{DER} = \frac{Y}{16} - \frac{X}{10} \quad (3.3)$$

$$PWM_{IZQ} = \frac{Y}{16} + \frac{X}{10} \quad (3.4)$$

El primer término de las ecuaciones 3.3 y 3.4 corresponde a movimientos de la cabeza hacia adelante y hacia atrás, con lo cual inclinar la cabeza hacia adelante incrementa la velocidad de ambas ruedas en igual cantidad. De igual forma, inclinar la cabeza hacia atrás implicaría valores negativos en el eje Y (que correspondería al eje -Z de la Figura 3.12), con lo cual ambas ruedas giran en sentido inverso y el vehículo gira en reversa. El segundo término corresponde a la dirección, por lo tanto movimientos de la cabeza hacia la derecha incrementan el valor de X y en consecuencia aumenta el valor de PWM de la rueda izquierda y disminuye el de la rueda derecha. Como resultado el vehículo se desplazara hacia la derecha o hacia la izquierda siguiendo los movimientos de la cabeza. Los valores de los denominadores se determinaron experimentalmente. En la escala utilizada, y como el valor máximo que puede tomar cada eje es de 1G, el valor numérico máximo obtenido en cada eje es

$$\frac{2^{16}}{4} = 16384$$

. Por lo tanto dividiendo dicho valor por 16 se obtiene velocidad máxima (1000) para inclinaciones de la cabeza de 90°. Por supuesto el usuario nunca llegaría a tan ángulo de inclinación (completamente horizontal), pero se eligió dicho valor para que el vehículo sea más fácil de controlar.

3.7. Cálculos de radiación no ionizante

Teniendo en cuenta los conceptos teóricos y regulaciones sobre la radiación no ionizante desarrollados en la Sección 2.5, se presentará un análisis para los diferentes casos de potencia y frecuencia de muestreo. Según la Tabla 2.9, para la frecuencia de trabajo (2400Mhz), se establece un límite máximo de densidad de potencia de $S = 1mW/cm^2$. Despejando el radio de la Ecuación 2.10 se obtiene:

$$r = \sqrt{\frac{PIRE, 2, 56 * F^2}{4. \pi. S}} \quad (3.5)$$

De forma conservadora, al desconocer F, se tomara igual a la unidad. El valor de PIRE, consiste en el valor de potencia por la ganancia de la antena respecto de la antena isotrópica. Si bien la ganancia de la antena real se desconoce, se utilizará de forma conservadora el valor de 3dB, tomado como referencia de productos de similares características.

El transceptor utilizado permite regular su potencia de salida entre 0dBm, -6dBm, -12dBm y -18dBm. Sin embargo, el transmisor no se encuentra transmitiendo continuamente sino que lo hace cada vez que se requiere transmitir datos nuevos, cuyo intervalo depende de la frecuencia de muestreo. El sistema

desarrollado permite variar el periodo de muestreo entre 5 y 250ms. Nuevamente, aplicando un criterio conservador, se supone que cada transmisión dura 5ms, de forma que el tiempo que se encuentra transmitiendo varía entre el 2% y el 100%. En los receptores utilizados se utilizó un periodo de muestreo lo que corresponde a un *ciclo de trabajo* de 10%. En la Tabla 3.17 se pueden observar las distancias mínimas, en milímetros, que deben ser respetadas entre la persona y la antena del dispositivo transmisor.

Potencia / Ciclo de Trabajo	2 %	10 %	100 %
0dBm	2.85	6.38	20.18
-6dBm	1.43	3.19	10.11
-12dBm	0.71	1.60	5.06
-18dBm	0.35	0.80	2.54

Tabla 3.17: Distancia a la antena mínima para diferentes potencias y ciclos de trabajo.

En el peor de los casos, es decir cuando se transmite a máxima potencia el 100% del tiempo, se deberá dejar una separación de aproximadamente 2cm. Por ejemplo, para las condiciones de funcionamiento del mouse, se utilizó potencia mínima transmitiendo el 10% del tiempo, con lo cual la distancia mínima debe ser de 0.8mm. En el caso del auto RC, se usó potencia máxima transmitiendo el 10% del tiempo, lo que requiere una distancia mínima de 6.38mm.

El prototipo desarrollado inicialmente no contempló estas medidas de seguridad, con lo cual una versión posterior deberá considerar colocar la posición de la antena al menos 20mm separadas del contacto con la cabeza. Otra medida posible podría ser colocar una capa metálica sobre el interior de la caja plástica, en las caras que están en contacto con la piel, con el objetivo de ofrecer una barrera contra la radiación electromagnética. En caso de continuar este proyecto de forma comercial, se deberán considerar estas cuestiones seriamente, y realizar las mediciones y certificaciones pertinentes, cuestiones que escapan al alcance de este proyecto.

Conclusiones

Desde el inicio del proyecto como una propuesta de trabajo para aplicar a una beca, hasta la finalización del mismo como proyecto final de grado, se han mantenido los lineamientos generales sin grandes cambios. El concepto de sistema modular e inalámbrico de bajo costo se pudo materializar tal como se había imaginado en su comienzo. En retrospectiva, las tecnologías seleccionadas han cumplido los requisitos esperados. Como los avances tecnológicos suceden de forma cada vez más veloz, es posible re considerar posibilidades que habían sido descartadas debido a su costo. Por ejemplo, los módulos de comunicación Bluetooth en la actualidad resultan más cercanos en costo respecto de los módulos utilizados. También han surgido nuevas opciones en materia de microcontroladores y sensores de aceleración que podrían reducir los costos en caso de llevarse a la producción comercial.

La posibilidad de permitir que cada receptor le indique al transmisor los parámetros de potencia, frecuencia, etc, le brindó al sistema la flexibilidad necesaria para adaptarse a todas las pruebas y usos que se diseñaron. Durante la etapa de desarrollo facilitó las pruebas y ajustes de cada dispositivo, solo con cambiar una línea de código en los receptores se adaptaba el sistema a las necesidades. Si bien la elección de un sistema inalámbrico por sobre uno cableado aumenta considerablemente el costo del sistema, el mismo es necesario para aumentar las posibilidades de aplicación. Por ejemplo, el control del mouse se podría haber hecho sin ninguno problema con un sistema cableado, al igual que una futura implementación de silla de ruedas, sin embargo el control del auto a radio control y el eventual control de un televisor no hubiera sido posible.

Desde el punto de vista de lo aprendido durante este trabajo, se puede destacar el conocimiento adquirido en comunicaciones serie en banda base (I2C, SPI, USB) así como los conceptos sobre paquetes de información, capas de protocolos de comunicación, y fundamentalmente el diseño de un enlace digital a partir de módulos comerciales. El hecho de haber tenido que definir cómo se realizarían todas las comunicaciones, con sus especificaciones, tipos y formas de paquete y parámetros de transmisión requirió una meticulosa documentación que respalde constantemente lo elaborado. Se obtuvieron también conocimientos sobre regulaciones de radiación no ionizante, y se estudió de forma teórica el desarrollo de un circuito impreso de radiofrecuencia.

Como se pretendió desde un comienzo, se logro desarrollar un sistema que reúne las características de varios proyectos ya existentes en uno solo, de forma de reducir costos y mejorar la experiencia del usuario. El proyecto continua abierto a nuevas aplicaciones, como las ya planteadas (control de silla de ruedas o televisor), o cualquier uso que surja en el futuro, solo limitados por la imaginación.

Apéndice A

Circuitos esquemáticos

A.1. Transmisor

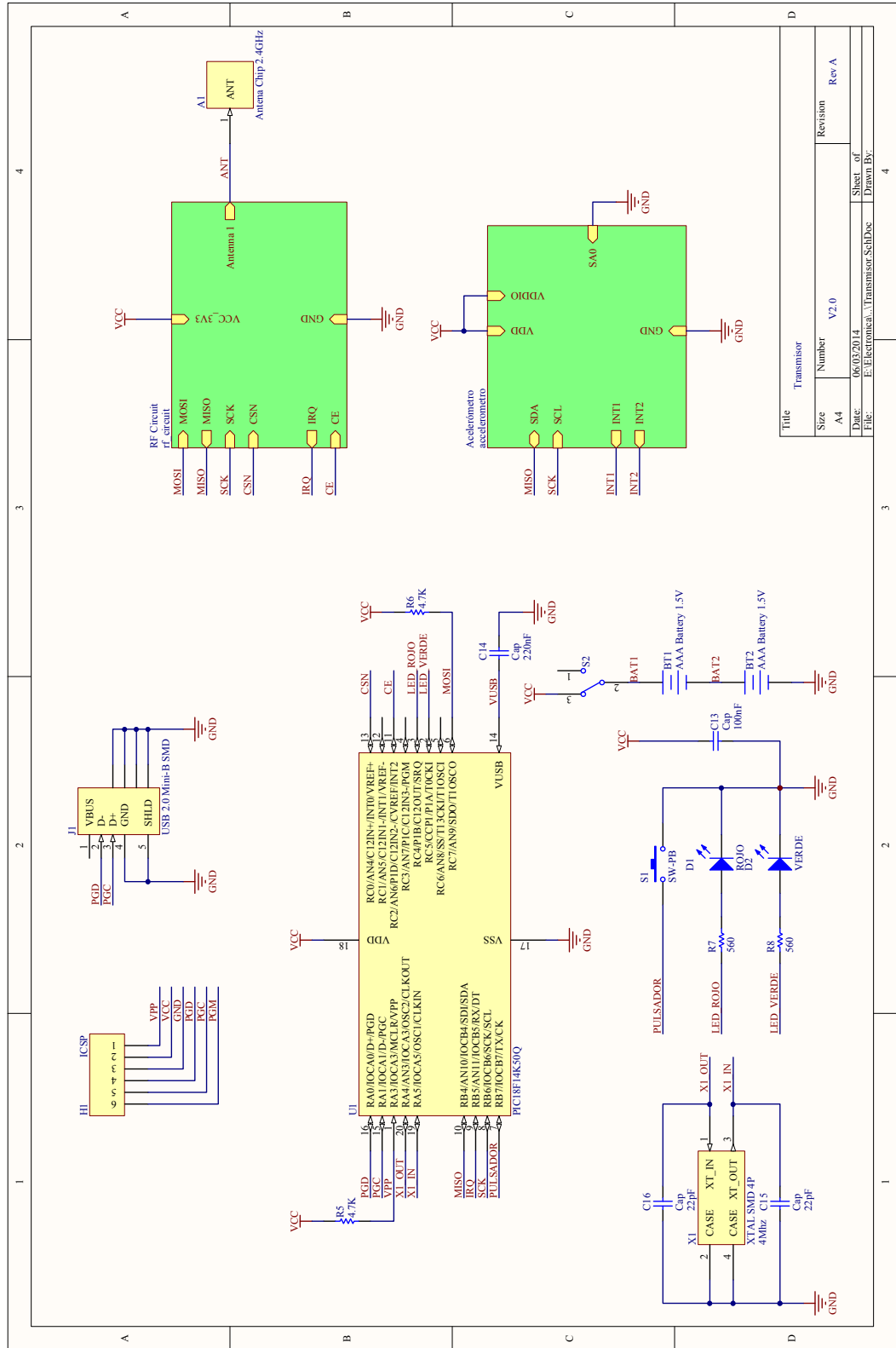


Figura A.1: Circuito esquemático del transmisor.

A.2. Receptor MATLAB

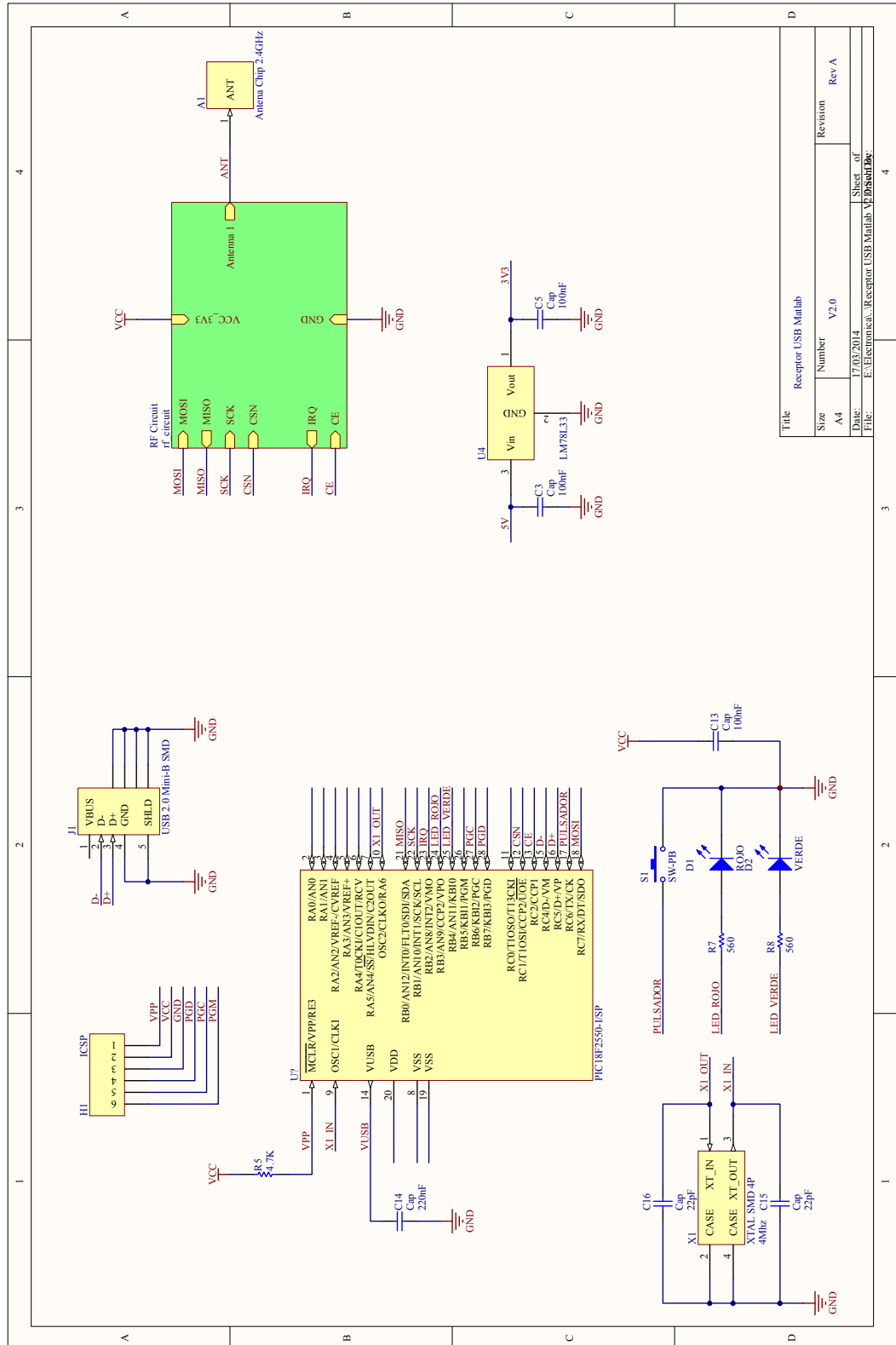


Figura A.2: Circuito esquemático del Receptor USB Matlab.

A.4. Receptor RC

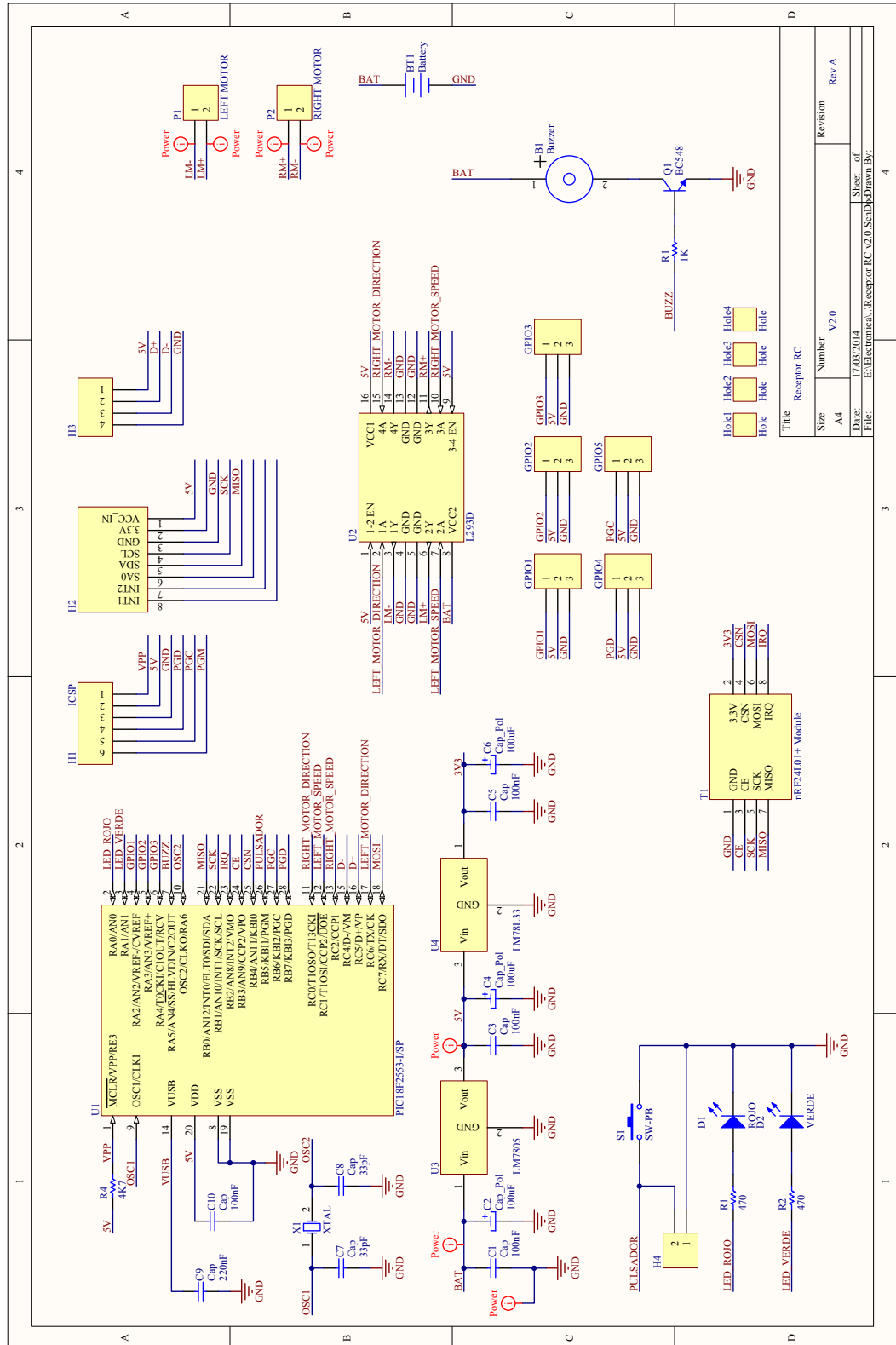


Figura A.4: Circuito esquemático del Receptor RC.

Índice de figuras

1.1. Parálisis según la ubicación de la lesión.	2
1.2. Controles mecánicos tradicionales para sillas de rueda.	4
1.3. Control de silla de ruedas mediante sensores infrarojos.	5
1.4. Control de silla de ruedas mediante detección de iris.	5
1.5. Control de una silla de ruedas mediante ondas cerebrales diseñado por Toyota.	6
1.6. Sistema propuesto basado en sensor de aceleración y ondas de radio frecuencia.	7
2.1. Principio de funcionamiento del sensor de aceleración	10
2.2. Distintas fuerzas que puede registrar el sensor de aceleración.	11
2.3. Ejemplo de un dispositivo fabricado con la tecnología MEMS.	11
2.4. Ejemplo de un sensor de tres ejes fabricado con la tecnología MEMS.	12
2.5. Diagrama en bloques del sensor de aceleración MMA8451.	12
2.6. Encapsulado del sensor de aceleración MMA8451.	12
2.7. Diagrama conexión del bus I2C	13
2.8. Diagrama de lectura de registro del bus I2C.	13
2.9. Diagrama de escritura de registro del bus I2C.	14
2.10. Diagrama de lectura de múltiples registros del bus I2C.	14
2.11. Diagrama de escritura de múltiples registros del bus I2C.	14
2.12. Modos de operación del sensor MMA8451.	15
2.13. nRF24L01+ en su encapsulado QFN de 20 pines de 4x4mm.	17
2.14. Diagrama en bloques del transceptor nRF24L01+.	17
2.15. Modulación FSK.	18
2.16. Diagrama en bloques de un sistema de modulación GFSK.	18
2.17. Comparación cualitativa entre el espectro de FSK y GFSK.	19
2.18. Diagrama de conexión SPI.	19
2.19. Operación de lectura y escritura del puerto SPI.	20
2.20. Diagrama de estados del transceptor nRF24L01+.	21
2.21. Formato del paquete del transceptor nRF24L01+.	21
2.22. Esquema ARQ del tipo Stop and Wait.	22
2.23. Familia de microcontroladores Microchip.	23
2.24. Microcontrolador en encapsulado DIP.	24
2.25. Programador PICKIT 3 TM de Microchip [®]	25
2.26. Pines del puerto USB.	26
2.27. Señalización diferencial del puerto USB.	27
2.28. Codificación NRZI.	28

2.29. Bus USB.	28
2.30. Espectro Electromagnético.	32
3.1. Diagrama de estados del sistema desarrollado.	35
3.2. Módulos utilizados en el prototipo experimental.	43
3.3. Primer prototipo del transmisor.	43
3.4. Circuito impreso del prototipo del transmisor.	44
3.5. Circuito impreso tentativo del transmisor.	45
3.6. PCB tentativo del transmisor.	45
3.7. Primer prototipo del receptor Matlab.	46
3.8. Datos de aceleración recibidos directamente en Matlab.	50
3.9. Primer prototipo del receptor Mouse.	51
3.10. Circuito impreso del prototipo del Receptor Mouse.	51
3.11. Ubicación de los ejes respecto de la posición de la cabeza.	52
3.12. Determinación de la inclinación de la cabeza.	53
3.13. Algoritmo ejecutado por el receptor.	54
3.14. Prototipo de auto a radio control ensamblado.	55
3.15. Circuito de entrada y salida de cada bloque del L293D.	55
3.16. Circuito impreso del receptor RC.	56
A.1. Circuito esquemático del transmisor.	II
A.2. Circuito esquemático del Receptor USB Matlab.	III
A.3. Circuito esquemático del Receptor USB Mouse.	IV
A.4. Circuito esquemático del Receptor RC.	V

Índice de tablas

1.1. Categorías de parálisis.	3
2.1. Comparación entre los microcontroladores utilizados.	24
2.2. Velocidades soportadas por el estándar USB 2.0.	26
2.3. Pines del estándar USB 2.0.	26
2.4. Niveles de tensión USB 2.0.	27
2.5. Niveles de corriente máximos USB 2.0.	27
2.6. Niveles de corriente máximos USB 2.0.	29
2.7. Transferencias USB 2.0.	29
2.8. Clases USB 2.0.	31
2.9. Valores máximos de radiación establecidos por la CNC.	33
3.1. Tipos de paquetes con su byte de identificación.	36
3.2. Configuración del paquete de búsqueda.	37
3.3. Formato del paquete de búsqueda.	37
3.4. Configuración del paquete de respuesta.	38
3.5. Formato del paquete de respuesta.	38
3.6. Campo ADDR3 según el tipo de dispositivo.	39
3.7. Campo RANGE HPF.	39
3.8. Campo PWR DR.	40
3.9. Configuración del paquete de datos.	40
3.10. Formato del paquete de datos.	40
3.11. Ejemplos de valores enviados en el paquete de datos.	41
3.12. Configuración del paquete de control.	41
3.13. Formato del paquete de control. *Solo en el comando PLAY y PAUSE.	41
3.14. Opciones para el campo CMD.	42
3.15. Funciones auxiliares del transmisor.	46
3.16. Comandos USB para controlar el receptor.	48
3.17. Distancia a la antena mínima para diferentes potencias y ciclos de trabajo.	58

Bibliografía

- [1] *Informe mundial sobre la discapacidad*, Organización Mundial de la Salud, 2011.
- [2] Dr. Alejandro Andersson, *Artículo Parálisis Cerebral*. Instituto de Neurología Buenos Aires, Ciudad Autónoma de Buenos Aires, Argentina.
- [3] Henrik Vie Christensen ,Juan Carlos Garcia, *Infrared Non-Contact Head Sensor, for Control of Wheelchair Movements*. Aalborg University, Dinamarca, Alcalá University, España, 2003.
- [4] Mark Nolan, Edward Burke, Frank Duignan, *Accelerometer Based Measurement of Body Movement for Communication, Play, and Creative Expression*. Dublin Institute of Technology, 2008.
- [5] Carlo Alberto Avizzano, Patrizio Sorace, Damaso Checcacci, *Head Traking based on accelerometer sensors*. Percro Laboratory, Pisa, Italia.
- [6] Danish Chopra, *Detection of Iris Movement by Estimating Sclera Size and Sync it with the Caster Wheel of a Wheelchair*. Department of computer science engineering, DAVIET, Jalandhar, Punjab, India, 2010.
- [7] Aleksandar Pajkanović, Branko Dokić, *Wheelchair Control by Head Motion*. University of Banja Luka, Faculty of Electrical Engineering, Banja Luka, The Republic of Srpska, Bosnia and Herzegovina, 2013.
- [8] *Resolución 3690/2004 (Boletín Oficial N° 30.524,10/11/04)*. Comisión Nacional de Comunicaciones, Buenos Aires, Argentina, 2004.
- [9] Ing. Victor Daniel Frizzera, *Radiaciones No Ionizantes*. Comisión Nacional de Comunicaciones, Buenos Aires, Argentina, 2007.
- [10] *In-Circuit Serial Programming Guide*. Microchip Technology Inc, Arizona, USA, 2003.
- [11] *PIC18F2455-2550-4455.4550 Data Sheet*. Microchip Technology Inc, Arizona, USA, 2009.
- [12] *PIC18F-LF1XK50 Data Sheet*. Microchip Technology Inc, Arizona, USA, 2010.
- [13] *PICkit3 Programmer-Debugger User's Guide*. Microchip Technology Inc, Arizona, USA, 2010.

- [14] *MPLAB-CXX Compiler User's Guide*. Microchip Technology Inc, Arizona, USA, 2000.
- [15] *Microchip USB Device Firmware Framework User's Guide*. Microchip Technology Inc, Arizona, USA, 2008.
- [16] Anthony Chu, *Choosing the Right Type of Accelerometers*. Measurement Specialties, Inc. California, USA.
- [17] *Accelerometers and How they Work*. Texas Instruments, Dallas, USA.
- [18] *MMA8451Q Data Sheet: 3-Axis, 14-bit/8-bit Digital Accelerometer*. Freescale Semiconductor, Texas, USA, 2012.
- [19] *Application Note: Embedded Orientation Detection Using the MMA8451*. Freescale Semiconductor, Texas, USA, 2011.
- [20] *Application Note: Data Manipulation and Basic Settings of the MMA8451*. Freescale Semiconductor, Texas, USA, 2012.
- [21] *nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0*. Nordic Semiconductor, Trondheim, Noruega, 2008.
- [22] Brennen Ball, *Everything You Need to Know about the nRF24L01 and MiRF-v2.*, 2007.
- [23] .Compaq, HP, Intel, Lucent, Microsoft, NEC, Philips, *Universal Serial Bus Specification Revision 2.0.*, 2000.
- [24] *USB Overview*. Silicon Labs, Texas, USA.
- [25] Balkar Erdoğan, *Implementation of a USB based PIC-to-PC communication*, 2011.
- [26] Bud Caldwell, *USB Generic Function on an Embedded Device*. Microchip Technology Inc, Arizona, USA, 2008.
- [27] Ing. Gabriel Jesús Pool Balam, *Transferencia y procesamiento de datos a alta velocidad, mediante el uso de MATLAB, el puerto USB 2.0 y PIC18F2455 de Microchip*. Universidad Modelo, Mérida, Yucatán, México, 2009.