

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de un dispositivo implantable para
el tratamiento de la epilepsia por medio de la estimulación del
nervio vago**

Trabajo de graduación presentado por Rodolfo José Cacacho Estrada
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2019

Diseño e implementación de un dispositivo
implantable para el tratamiento de la epilepsia por
medio de la estimulación del nervio vago

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




**Diseño e implementación de un dispositivo implantable para
el tratamiento de la epilepsia por medio de la estimulación del
nervio vago**

Trabajo de graduación presentado por Rodolfo José Cacacho Estrada
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,


2019

Vo.Bo.:

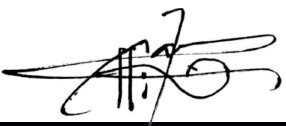
(f) 

Ing. Pablo Mazariegos

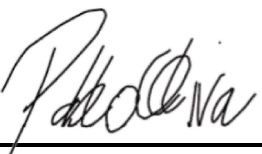
Tribunal Examinador:

(f) 

Ing. Pablo Mazariegos

(f) 

MSc. Carlos Alberto Esquit

(f) 

MSc. Pablo Roberto Oliva

Fecha de aprobación: Guatemala, 3 de diciembre de 2019.

La elaboración de esta tesis surge gracias a la petición de HUMANA con el objetivo de replicar el estimulador del nervio vago para que este pueda ser diseñado y construido en Guatemala con el fin de ayudar a muchas personas que sufren de epilepsia. Además de esto, existe el interés personal por los sistemas embebidos y la programación de microcontroladores por lo cual se realizó este proyecto con gran motivación.

El gran reto de este proyecto consistió en aplicar conocimientos vistos a lo largo de la carrera con el fin de diseñar el generador. Se utilizaron conocimientos de microcontroladores, circuitos analógicos y digitales además de conocimientos de programación. Es muy importante notar la complejidad que requiere el diseño de dispositivos embebidos, en especial cuando estos tratan con personas, puesto existen riesgos de lesión o algún mal funcionamiento.

Los fondos para la realización de este proyecto fueron totalmente por los integrantes del grupo, Andrés Girón, Hugo Elvira y mi persona. Por último, quiero agradecer a mis padres por el esfuerzo y apoyo que me brindaron durante mi carrera; a esta casa de estudios por brindarme los conocimientos que espero aplicar en mi vida profesional; agradezco a todos mis amigos que formaron parte de esta carrera y a Andrés Girón y Hugo Elvira, participantes de este proyecto.

Prefacio	v
Lista de figuras	xii
Lista de cuadros	xiii
Resumen	xv
Abstract	xvii
1. Introducción	1
2. Antecedentes	3
2.1. Modelos 101 al 105 de Cyberonics/LivaNova	3
2.1.1. Prescripción de la terapia VNS	3
3. Justificación	5
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Alcance	9
6. Marco teórico	11
6.1. Epilepsia	11
6.1.1. Síntomas	11
6.1.2. Epilepsia refractaria	12
6.2. Terapia por estimulación del nervio vago - VNS	13
6.2.1. Partes implantables de la terapia VNS	14
6.2.2. Partes no implantables de la terapia VNS	16
6.2.3. Modos de operación	17
6.3. Diseño de sistemas de baja potencia	18
6.3.1. Principales fuentes de consumo de energía	18

6.3.2.	Diseño de software	19
6.3.3.	Diseño de hardware	21
6.3.4.	Power budgeting	22
6.4.	PWM	22
6.4.1.	Resolución de tiempo del PWM	23
6.5.	Memoria EEPROM	23
6.6.	Estándar BARR-C	23
6.6.1.	Reglas generales	24
6.6.2.	Tipos de datos	24
6.6.3.	Reglas de sentencias	25
6.6.4.	Reglas de variables, procedimientos y módulos	25
6.7.	Entorno de microchip	26
6.7.1.	MPLAB X IDE	27
6.7.2.	Tecnología eXtreme Low Power XLP	28
6.8.	Serial Peripheral Interface - SPI	29
7.	Análisis y diseño del estimulador	31
7.1.	Identificación de requisitos	31
7.1.1.	Bajo consumo energético	31
7.1.2.	Generación de la señal de estimulación	32
7.1.3.	Coordinación de módulos de potencia y comunicación inalámbrica	32
7.2.	Diseño del estimulador	33
7.3.	Selección de la circuitería necesaria para replicar funcionamiento del estimulador	34
8.	Prototipo 1	37
8.1.	Reproducción de la señal de estimulación	37
8.1.1.	Frecuencia y ancho de pulso	37
8.1.2.	Tiempo de estimulación	39
8.1.3.	Tiempo de reposo de estimulación	40
8.2.	Configuración del MCU para prototipo 1	41
8.3.	Resultados del prototipo 1	42
8.3.1.	Resultados del ancho de pulso de la señal	42
8.3.2.	Resultados de la frecuencia de la señal	43
8.3.3.	Resultados del tiempo de estimulación	44
8.3.4.	Resultados del tiempo de reposo	45
8.4.	Evaluación de resultados	46
8.5.	Retos y dificultades	47
9.	Prototipo 2	49
9.1.	Modificaciones a realizar al prototipo 1	49
9.1.1.	Modificación al ancho de pulso y frecuencia de estimulación	49
9.1.2.	Calibración de los tiempos de estimulación	51
9.1.3.	Modificación a los tiempos de reposo	51
9.1.4.	Optimización y mejoras a la estructura del código	51
9.2.	Resultados de la reproducción de los parámetros de estimulación	54
9.2.1.	Resultados del ancho de pulso de la señal	54
9.2.2.	Resultados de la frecuencia de la señal	55
9.2.3.	Resultados del tiempo de estimulación	56

9.2.4. Resultados del tiempo de reposo	57
9.3. Evaluación de resultados	60
9.4. Retos y dificultades	60
10.Prototipo 3	61
10.1. Pruebas con la memoria EEPROM	61
10.2. Modificaciones a realizar al prototipo 2	62
10.3. Interconexión con el módulo de potencia	63
10.4. Evaluación de resultados	63
10.5. Retos y dificultades	64
11.Prototipo 4	65
11.1. Configuración del módulo RF	65
11.2. Modificaciones a realizar al prototipo 3	67
11.2.1. Modo estimulación	68
11.2.2. Modo reposo	68
11.2.3. Modo programación	69
11.2.4. Protocolo de comunicación	69
11.3. Evaluación de resultados	70
11.4. Retos y dificultades	70
12.Prototipo final	73
12.1. Prototipo final 1	73
12.1.1. Esquemático y placa impresa del prototipo final	74
12.2. Prototipo final 2	75
12.3. Estimación de vida útil	77
13.Conclusiones	79
14.Recomendaciones	81
15.Bibliografía	83
16.Anexos	87
16.1. Código	87
16.2. Esquemáticos	87
16.2.1. Esquemáticos prototipo final	88
16.3. Pruebas físicas	90
17.Glosario	95

Lista de figuras

1.	Estimulador del nervio vago modelo 102 Cyberonics	3
2.	Implantación del generador y colocación del electrodo	14
3.	Diagrama del circuito del generador	16
4.	Partes no implantables del tratamiento VNS	17
5.	Señal de estimulación y sus parámetros	18
6.	Señal de PWM con distintos ciclos de trabajo	23
7.	Logo de Microchip	26
8.	Interfaz de configuración de hardware de MCC	27
9.	Cálculo de T_{BE} para comparación entre deep sleep y sleep mode	29
10.	Diagrama de conexión SPI multi-esclavo	29
11.	Modos de comunicación SPI	30
12.	Primer acercamiento basado en los tres modos de operación para replicar el funcionamiento del estimulador	33
13.	Microcontrolador PIC18F46K22 en empaquetado DIP40	36
14.	Microcontrolador PIC18F27J13 en empaquetado DIP28	36
15.	PWM de 30 Hz y ciclo de trabajo de 50 %	39
16.	Diagrama de flujo para reproducir los tiempos de reposo	41
17.	Medición de ancho de pulso de 500 μs	43
18.	Medición de ancho de pulso de 1000 μs	43
19.	Medición de la frecuencia de 2 Hz	43
20.	Medición de la frecuencia de 5 Hz	43
21.	Medición de la frecuencia de 20 Hz	44
22.	Medición de la frecuencia de 30 Hz	44
23.	Medición de tiempo de estimulación de 7 s	44
24.	Medición de tiempo de estimulación de 14 s	44
25.	Medición de tiempo de estimulación de 30 s	45
26.	Medición de tiempo de estimulación de 60 s	45
27.	Medición para el tiempo de reposo de 0.2 min	45
28.	Medición para el tiempo de reposo de 1.1 min	45
29.	Medición para el tiempo de reposo de 3 min	46
30.	Medición para el tiempo de reposo de 5 min	46
31.	PIC16F1789 en empaquetado PDIP40	47
32.	Función que codifica los tiempos de estimulación	52

33.	Medición de ancho de pulso de 130 μs	55
34.	Medición de ancho de pulso de 250 μs	55
35.	Medición de ancho de pulso de 500 μs	55
36.	Medición de ancho de pulso de 1000 μs	55
37.	Medición de la frecuencia de 1 Hz	56
38.	Medición de la frecuencia de 5 Hz	56
39.	Medición de la frecuencia de 20 Hz	56
40.	Medición de la frecuencia de 30 Hz	56
41.	Medición de tiempo de estimulación de 7 s	57
42.	Medición de tiempo de estimulación de 14 s	57
43.	Medición de tiempo de estimulación de 30 s	57
44.	Medición de tiempo de estimulación de 60 s	57
45.	Medición para el tiempo de reposo de 0.2 min	58
46.	Medición para el tiempo de reposo de 1.1 min	58
47.	Medición para el tiempo de reposo de 3 min	58
48.	Medición para el tiempo de reposo de 5 min	58
49.	Señal medida para tiempos de reposo de 5, 10 y 15 minutos utilizando el Logic Pro 16	58
50.	Señal medida para tiempos de reposo de 30 y 45 minutos utilizando el Logic Pro 16	58
51.	Medición para los tiempo de reposo de 5, 10 y 15 min	59
52.	Medición para los tiempo de reposo de 30 y 45 min	59
53.	Configuración del PIC para las pruebas de la memoria EEPROM	62
54.	DIP Switch de 8 bits	62
55.	Medición de ancho de pulso de 130 μs y modulación de amplitud	63
56.	Medición de la modulación del la señal a un ancho de pulso de 130 μs	63
57.	Operación de lectura de registros del SPIRIT1	66
58.	Operación de escritura de registros del SPIRIT1	66
59.	Operación de comando del SPIRIT1	66
60.	Comunicación entre PIC16F1789 y SPSGRFC-868	67
61.	Segundo acercamiento	68
62.	Medición del tamaño del PCB lado A	74
63.	Medición del tamaño del PCB lado B	74
64.	Diseño del PCB de dos capas	75
65.	Diagrama de flujo acercamiento prototipo final	76
66.	Placa soldada del prototipo final con PIC18F27J13	77
67.	Esquemático prototipo 4 - Estimulador	87
68.	Esquemático prototipo final - Módulo de control	88
69.	Esquemático prototipo final - Módulo de potencia	89
70.	Esquemático prototipo final - Módulo RF	89
71.	Conexión de validación de memoria EEPROM	90
72.	Prototipo 4 - Estimulador	90
73.	Prototipo final - Placa impresa previo a soldar	91
74.	Prototipo final - Comparación de tamaño de componentes	92
75.	Prototipo final - Representación en 3D por software de diseño de PCBs	93

Lista de cuadros

1.	Escala de Schmidt para la intratabilidad de la epilepsia refractaria farmacológica	13
2.	Parámetros de la terapia de estimulación del nervio vago para el tratamiento de la epilepsia	14
3.	Dimensiones y peso de los modelos del generador	15
4.	Parámetros de estimulación modo normal de los modelos 102 al 105 de LivaNova	17
5.	Características de los distintos modos de operación de un MCU	20
6.	Tamaño de enteros según C99	25
7.	Características relevantes de microcontroladores	35
8.	Evaluación de microcontroladores	36
9.	Parámetros calculados de PRx para el cálculo de la frecuencia del módulo de PWM	38
10.	Parámetros calculados de $CCPRxL : CCPxCON < 5 : 4 >$ para el cálculo del ancho de pulso del PWM	38
11.	Valores mapeados de la variable cantidad al tiempo de reposo real	40
12.	Ancho de pulso teórico y generado en prototipo 1	43
13.	Frecuencia de estimulación teórica y generada en prototipo 1	44
14.	Tiempo de estimulación teórica y generado en prototipo 1	45
15.	Tiempo de reposo teórico y generado en prototipo 1	46
16.	Evaluación de logros prototipo 1	46
17.	Casos para la frecuencia de estimulación y la variable TMR1 codificada	53
18.	Casos para el tiempo de estimulación	53
19.	Casos para el ancho de pulso	53
20.	Casos para el tiempo de reposo y variable cantidad sleep	54
21.	Ancho de pulso teórico y generado en prototipo 2	55
22.	Frecuencia de estimulación teórica y generada en prototipo 2	56
23.	Tiempo de estimulación teórica y generado en prototipo 2	57
24.	Tiempo de reposo teórico y generado en prototipo 2	59
25.	Evaluación de logros prototipo 2	60
26.	Organización de los parámetros en la memoria EEPROM	62
27.	Codificación de los modos de operación	69
28.	Formato del paquete utilizado	70
29.	BOM del estimulador del nervio vago	74
30.	Corriente promedio de los módulos	78

Este proyecto consiste en diseñar e implementar un estimulador del nervio vago para el tratamiento de la epilepsia. La idea surge en conjunto con HUMANA, una asociación que trata padecimientos neurológicos en Guatemala. HUMANA plantea la realización del estimulador, pero diseñado en Guatemala, con el objetivo de reducir su costo, haciéndolo así una opción viable para muchas más personas en el país, ya que este tratamiento no es accesible para muchos.

El estimulador del nervio vago es aproximadamente del tamaño de una caja de cerillos y se implanta debajo de la piel a la altura de la clavícula. Este dispositivo emite un pulso cada cierta cantidad de tiempo a cierta intensidad y luego descansa otro periodo de tiempo. Estos parámetros son personalizables por el doctor dependiendo de la respuesta del paciente al tratamiento. En Humana se observó el modelo 102 de Cyberonics, por lo cual el diseño y la implementación se realizará según este. El modelo 102 tiene una vida útil promedio de aproximadamente 10 años y se programa mediante una varilla de programación externa.

Debido a la complejidad del proyecto, este se dividió en tres módulos, los cuales son: módulo de funcionamiento y control, módulo de potencia y módulo de comunicación inalámbrica. En este trabajo de graduación se desarrolló el módulo de funcionamiento y control. Este módulo consiste en el diseño de la circuitería capaz de emular el funcionamiento del estimulador, además de implementar un algoritmo de control que coordine los otros dos módulos y replique la señal de estimulación.

Para realizar el estimulador se aplicaron conocimientos de diseño de circuitos, programación de microcontroladores, entre otros. El estimulador se diseñó utilizando microcontroladores de 8 bits de la empresa Microchip. Se seleccionaron los microcontroladores energéticamente más eficientes para cumplir con los objetivos de un diseño de baja potencia. Además, se utilizaron varias estrategias desde la parte del software para que este consumiera menos energía.

La señal de estimulación realizada cumplió con todos los parámetros del modelo 102 de LivaNova. Para realizar la estimulación se utilizaron módulos de temporizadores, interrupciones y se aprovechó el tiempo de reposo para utilizar el modo sleep del MCU con el objetivo de reducir la potencia promedio de uso del dispositivo.

Por último, se realizó un algoritmo robusto que permite el cambio de parámetros desde la interfaz gráfica diseñada por el encargado de la comunicación inalámbrica. Para esto, se utilizó un módulo de radio frecuencia para comunicarse de manera inalámbrica con la varilla de programación.

This project's objective is to design a vagus nerve generator for the treatment of epilepsy. The idea came from HUMANA, an association that treats neurological ailments in Guatemala. HUMANA asked for the help of the Universidad del Valle de Guatemala so that the generator can be designed and built in Guatemala. The cost of the vagus nerve stimulation therapy is not affordable for the most of the population.

The generator is about the size of a match box and it is implanted under the skin at the height of the collarbone. This device generates an electric pulse that goes to the left vagus nerve. The intensity, frequency and duration of this pulse can be manually configured by the doctor. As the generator is implanted, the configuration is made wirelessly, using a programming wand that sends the data to the antenna on the generator. The life expectancy of the device is around 10 years.

Due to the complexity of the project, it was divided in 3 modules, which are the following: Operation and Control, Power and Wireless communication. The Operation and Control module are developed in this thesis. The task of this module consists in the generation of the stimulation signal, coordinating the other 2 modules and the algorithm of control of the generator.

To be able to design the generator, knowledge of circuit design, microcontroller programming amongst many others were required. The generator was designed using 8-bit microcontrollers from Micrichip. The most power-efficient MCUs were selected so that the power requirements could be fulfilled. Besides the hardware selected, some software tricks were used so that the system could save energy when not performing any useful tasks.

The stimulation signal replicated every parameter that the 102 model of LivaNova did. To make the stimulation signal, timers, interrupts and energy saving modes from the MCU were used.

Lastly, the control algorithm allowed that the parameters could be programmed using the programming wand.

En Guatemala, la asociación HUMANA ofrece el tratamiento a la epilepsia. Un tratamiento no invasivo como la cirugía del cerebro se utiliza para el tratamiento de la epilepsia refractaria farmacológica. Este tratamiento conocido como VNS consiste en mandar un pulso de corriente al nervio vago. Este pulso genera cierta inmunidad a los ataques epilépticos así reduciendo la cantidad de convulsiones y mejorando la calidad de vida de muchos que padecen este tipo de epilepsia.

El costo del tratamiento VNS hace que este sea poco accesible para muchas personas en el país, es por esta razón, que HUMANA acudió a la Universidad del Valle de Guatemala con el objetivo de replicar este dispositivo. En este trabajo se trabajó la parte de funcionamiento y control, que es la parte que se encarga en la generación de la señal de estimulación y coordinar al módulo de potencia y de comunicación inalámbrica.

Para el desarrollo de este dispositivo se utilizaron conocimientos de diseño de dispositivos de baja potencia, programación de microcontroladores, protocolos de comunicación entre otros. El estimulador tiene consumir poca energía con tal de poder aumentar la vida útil del dispositivo. Además de esto, debe ser configurable de manera remota, pues de la manera que funciona el tratamiento VNS, el médico modifica los parámetros de la estimulación según ve la respuesta en el paciente.

Por último, debido a el gran riesgo que este dispositivo puede suponer en caso de una falla, se agregaron características al algoritmo de control que limitaran la estimulación solamente a parámetros estándar de la terapia.

2.1. Modelos 101 al 105 de Cyberonics/LivaNova

Los modelos de LivaNova, previamente de Cyberonics antes de su fusión con Grupo Sorin de Italia, son dispositivos que implementan una terapia de estimulación del nervio vago, más conocida como VNS. La terapia VNS manda un pulso eléctrico a un nervio que va a el cerebro, conocido como el nervio vago. Esta terapia se utiliza para tratar a pacientes con cierto tipo de epilepsia.

Según estudios de Linanova, demuestran que el tratamiento por VNS es el número 1 a nivel mundial para las convulsiones. Además, se tiene un 81 % de satisfacción de los usuarios implantados [1] y el 73 % de los pacientes han sido reimplantados luego de cumplir la vida útil de un estimulador.



Figura 1: Estimulador del nervio vago modelo 102 Cyberonics

2.1.1. Prescripción de la terapia VNS

La terapia VNS consiste en estimular el nervio vago con un pulso de corriente a cierta frecuencia durante cierto intervalo de tiempo; luego se detiene la descarga y se deja descansar el nervio vago para evitar fibrosis en este. La terapia VNS tiene varios parámetros

modificables, entre ellos la frecuencia y el ancho del pulso de la estimulación, la intensidad de la estimulación, cuanto dura la estimulación y cuanto tiempo descansa el nervio.

El tratamiento por medio de la estimulación del nervio vago se utiliza para tratar las conocidas como Epilepsias resistentes a drogas o DRE como se conocen por sus siglas en inglés. Se estima que 1 de cada 3 pacientes que son tratados con fármacos anti-epilépticos no responderán de manera correcta al tratamiento, además en los últimos 20 años no se ha reducido el número de pacientes que no responden de manera adecuada a este tipo de fármacos. [2]. Este tipo de epilepsia tiene consecuencias que se extienden a más que solamente convulsiones [3]-[5], como lo son las siguientes:

- Lesiones relacionadas a convulsiones
- Depresión, ansiedad y disturbio del sueño
- Deficiencia en la capacidad cognitiva y de la memoria
- Efectos secundarios de las drogas anti-epilépticas
- Dificultad a la hora de la educación y el trabajo
- Dificultad de interactuar socialmente

La estimulación tiene varias maneras de reducir efectivamente las convulsiones, entre ellas desincroniza patrones EEG relacionadas a la epilepsia, altera la liberación del neurotransmisor y aumenta el flujo sanguíneo en el cerebro, específicamente en el tálamo y en la corteza cerebral.[6]-[9]

El diseño y desarrollo del estimulador del nervio vago para el tratamiento de la epilepsia tiene como fin ayudar a guatemaltecos y guatemaltecas que padecen de epilepsia y que son candidatos para optar a un tratamiento de estimulación del nervio vago y no son capaces de costear tal tratamiento. Con este tratamiento se puede mejorar la calidad de vida de dichas personas; además se estima que en Guatemala existen más de 350,000 personas que padecen de esta enfermedad, por lo cual desarrollar un dispositivo como tal sería de gran impacto y beneficio para el país.[10]

La terapia por medio de estimulación del nervio vago es preferida muchas veces ante tratamientos por medio de drogas o una operación del cerebro; esto se debe a que es menos invasiva y riesgosa que esos tratamientos. El tratamiento conocido como terapia VNS (Vagus Nerve Stimulation) por sus siglas en inglés, puede tener un costo de aproximadamente \$20K [11] en los Estados Unidos, razón por lo cual no es accesible para los miles de personas que sufren epilepsia en Guatemala. Además, se estima que en el país el 59% de la población se encuentra debajo de la línea de pobreza [12], lo cual hace más evidente que si estas personas sufren de epilepsia, estas no podrán acceder a el tratamiento.

El costo del estimulador es tan elevado no debido a la complejidad del sistema, si no por las garantías y pruebas que se han realizado a lo largo de los años a este dispositivo. Es por esto, que diseñar y fabricar un dispositivo como este en Guatemala reduciría el costo de este de gran manera. El dispositivo se puede desarrollar en fases, así realizando pruebas y mejoras hasta lograr una versión que cumpla con los estándares actuales de los estimuladores.

El módulo de funcionamiento y control es de vital importancia para el funcionamiento del estimulador, pues este se encarga de producir la señal de estimulación, modificar los parámetros y de coordinar los otros dos módulos. El módulo consiste en la circuitería y el algoritmo o programa. El diseño de la circuitería, es decir, la selección de componentes electrónicos es esencial para el dispositivo implantable, pues este define el tamaño del dispositivo e influye en la vida útil del mismo. Por otro lado, el algoritmo se encarga de controlar el funcionamiento general del dispositivo, produciendo la señal de estimulación, regulando sus parámetros y coordinando los otros dos módulos. Un diseño robusto del algoritmo reduce la probabilidad de fallos del dispositivo, desde una mala programación de los parámetros hasta la detección de un defecto de algún otro módulo del estimulador.

4.1. Objetivo general

Diseño e implementación de la circuitería e implementación del algoritmo de control del estimulador del nervio vago para realizar el prototipo del estimulador.

4.2. Objetivos específicos

- Producir la señal de estimulación bajo los parámetros estándar establecidos para el tratamiento de la epilepsia.
- Diseño robusto del algoritmo que permita la modificación de los parámetros de la señal de estimulación para que el médico actualice estos de acorde a la respuesta del paciente a el tratamiento de la epilepsia.
- Dimensionamiento de la batería y selección de los componentes del estimulador que puedan brindar una vida útil estimada del dispositivo de al menos 3 años.
- Circuito diseñado en placa impresa de tamaño no mayor al del modelo 102 de Cyberonics/LivaNova (25cm^2) para cumplir con el requisito discutido con HUMANA.

Tomando como base el modelo 102 de la marca Cyberonics, el cual fue el que se observó con Humana, se replicarán los parámetros de estimulación del nervio vago para el tratamiento de la epilepsia en un prototipo. En este prototipo solamente se incluirá el modo normal y no se realizará el modo en demanda activado por el imán. Los parámetros de la estimulación podrán ser modificados de manera remota. Debido a la complejidad del proyecto, este se dividirá en 3 módulos.

- Módulo de funcionamiento y control
- Módulo de comunicación inalámbrica
- Módulo de potencia

El enfoque de este trabajo de graduación será el módulo de funcionamiento y control, en donde se coordinarán los otros dos módulos, se controlará el comportamiento del generador y se generará la señal de estimulación.

El prototipo incorporará los módulos de comunicación inalámbrica y el de potencia que proveerá la alimentación de la señal de corriente que va hacia al electrodo; así realizando un prototipo del generador en el cual los parámetros del modo normal se logren configurar de manera inalámbrica, y produciendo la señal de corriente para ser medida de manera experimental. Este prototipo se construirá en una placa impresa de no más de 25 cm^2 con componentes de superficie en una etapa previa a la realización de pruebas con electrodos reales y pruebas en animales. No se diseñará un estuche que se pueda implantar y se dimensionará la batería a utilizar para que el estimulador tenga una vida útil de al menos 3 años.

6.1. Epilepsia

La epilepsia es un trastorno del sistema nervioso central en el que la actividad cerebral se altera, lo que provoca convulsiones o períodos de comportamiento y sensaciones inusuales, y, a veces, pérdida de la conciencia. Cualquier persona puede padecer epilepsia. La epilepsia afecta tanto a hombres como a mujeres de todas las razas, grupos étnicos y edades. Los síntomas de las convulsiones pueden variar mucho. Algunas personas con epilepsia simplemente permanecen con la mirada fija por algunos segundos durante una convulsión, mientras que otras contraen repetidamente los brazos o las piernas. El tratamiento con medicamentos o, a veces, la cirugía puede controlar las convulsiones en la mayoría de las personas con epilepsia. Algunas personas requieren tratamiento de por vida para controlar las convulsiones, pero, para otras, las convulsiones finalmente desaparecen. Algunos niños con epilepsia pueden superar la enfermedad con la edad.[13]

6.1.1. Síntomas

La epilepsia puede presentar varios síntomas, y debido a actividad anormal ocurre en el cerebro, cualquier tarea que este controla puede ser afectada. A esto se le conoce como convulsiones, y entre sus signos o síntomas más comunes se encuentran los siguientes:

- Confusión temporal
- Episodios de ausencias
- Movimientos incontrolables de brazos y/o piernas
- Pérdida del conocimiento o conciencia
- Síntomas psíquicos, como miedo, ansiedad o déjà vu

Para las personas que padecen epilepsia, por lo general, tenderán hacia un mismo tipo de episodios, por lo cual los episodios para cada paciente serán similares entre uno y otro. Los médicos clasifican a las convulsiones en función de como comienza cerebral anormal. [13]

Convulsiones focales

Las convulsiones focales son las que se localizan en solo un área del cerebro y también suelen conocerse como convulsiones parciales. Las convulsiones focales simples afectan solo una pequeña parte del cerebro. Estas pueden causar sacudidas o algún cambio de sensación, como algún sabor u olor extraño.

Las convulsiones focales complejas hacen que la persona se sienta aturdida o confundida. Por lo general la persona no podrá seguir con lo que estaba realizando por un par de minutos. Un caso más grave son las convulsiones generalizadas secundarias, que son las que empiezan en una parte del cerebro y luego se expanden hacia otras partes del cerebro. Por lo que la persona estaría experimentando primero una convulsión focal y luego una generalizada.[14]

Convulsiones generalizadas

Las convulsiones generalizadas afectan ambos hemisferios del cerebro. Entre las convulsiones generalizadas existen las de ausencia y las tonicoclónicas. Las convulsiones de ausencia son conocidas como menores, y puede que causen un parpadeo rápido o mirada fija a lo lejos por unos segundos.

El término tónico se refiere a cuando los músculos se ponen rígidos; el término clónico se refiere a los periodos cuando las partes del cuerpo tiemblan o se sacuden. Con lo anterior, podemos inferir que las convulsiones tonicoclónicas son mayores y estas pueden hacer que la persona grite, pierda el conocimiento, se caiga al piso o tenga rigidez o espasmos musculares. Algunas personas pueden sentirse cansadas luego de haber tenido una convulsión tonicoclónica. El problema de este tipo de convulsiones es que el paciente pueda golpearse al caer.[14]

6.1.2. Epilepsia refractaria

La epilepsia refractaria es un tipo de epilepsia que se describe cuando las crisis epilépticas son tan frecuentes que limitan la habilidad del paciente para vivir plenamente acorde con sus deseos y su capacidad mental y física o cuando el tratamiento anticonvulsivante no controla las crisis o sus efectos secundarios son limitantes para un desarrollo normal de la persona.[15] Este tipo de epilepsia se puede catalogar en tres categorías: Biológica, Farmacológica y Psicológica y Social.

Epilepsia refractaria farmacológica - DRE

Este tipo de epilepsia es mayormente conocida como DRE por sus cifras en inglés que significan Drug Resistant Epilepsy. Esta epilepsia se puede catalogar dependiendo del grado de intratabilidad, el Cuadro 1 muestra la escala de Schmidt. Si las crisis persisten, y se encuentra dentro de las primeras 4 categorías (0 a 3) se dice que es una epilepsia insuficientemente tratada; si el diagnóstico se encuentra entre los niveles 4 al 6 se dice que la epilepsia muestra refractariedad verdadera. [16]

Persistencia de crisis	Índice de intratabilidad
Droga no de primera línea, sin importar su dosis	0
Droga de primera línea con subdosificación	1
Droga de primera línea con dosis adecuada	2
Droga de primera línea con rango sérico adecuado	3
Droga de primera línea con dosis máxima tolerable	4
2 o más drogas de primera línea a dosis máximas tolerables	5
2 o más drogas de primera línea a dosis máxima tolerable y droga de segunda línea	6

Cuadro 1: Escala de Schmidt para la intratabilidad de la epilepsia refractaria farmacológica

Otra definición dada por la Liga Internacional contra la Epilepsia es la siguiente: “Epilepsia resistente a fármacos puede definirse como aquella en la que no se ha conseguido una evolución libre de crisis a pesar de dos ensayos terapéuticos adecuados con diferentes FAE, tomados en monoterapia o asociados, siempre que se hayan seleccionado y usado de manera adecuada.” [17]

Tratamiento de epilepsia refractaria farmacológica

El problema de este tipo de epilepsia es que los pacientes llegan a estar tomando dosis mayores y medicinas cuyos efectos secundarios pueden afectar de igual manera su calidad de vida. Muchos pacientes prefieren tener una convulsión más al mes que sufrir de los efectos secundarios que llevan el consumo de estos fármacos. Entre los efectos secundarios es la falta de energía, sueño; además, su memoria se ve afectada e inclusive pueden ser inhabilitados a hablar. Otros efectos son los del metabolismo, pueden afectar el funcionamiento del hígado, entre otros. Además, estos efectos secundarios se pueden ver magnificados a la hora de combinar varias drogas.

Otra opción para los pacientes es la cirugía de cerebro, en donde se espera que entre un 60 y 70 % de los pacientes que tienen DRE con convulsiones focales lleguen a estar libres de convulsiones luego de la operación. La operación de cerebro es efectiva pero conlleva un gran riesgo, y además solamente un pequeño porcentaje de todos los pacientes que sufren de DRE son elegibles para este tratamiento. Existen tres métodos con los que se puede tratar este tipo de epilepsia actualmente aprobados por la FDA, los cuales son: estimulación del nervio vago (VNS), neuroestimulación sensible (RNS) o estimulación cerebral profunda (DBS). [18]

6.2. Terapia por estimulación del nervio vago - VNS

VNS de las siglas en inglés, Vagus Nerve Stimulation, es uno de los métodos utilizados para tratar la epilepsia que consiste en la estimulación de los pares de nervios que se encuentran ubicados a lo largo de todo el cuerpo, que se encargan del envío de mensajes entre el cerebro y el cuerpo.

La estimulación del nervio vago consiste en el uso de un dispositivo que, por medio de impulsos eléctricos, estimula el nervio. Este dispositivo es implantado en el cuerpo, debajo del pecho. Cuando se activa, el dispositivo manda impulsos eléctricos a lo largo del nervio vago izquierdo. Este sistema de protección contra la epilepsia se lleva a cabo debido a que cerca de un tercio de las personas con esta enfermedad, no responden de manera completa

a las drogas para evitar las convulsiones. De cierta manera, el VNS reduce la frecuencia de estos ataques en personas que no han logrado el control de esta enfermedad con medicación.

Este dispositivo se utiliza debido a que este procedimiento es poco invasivo, además los parámetros de la estimulación pueden ser modificados de acorde a la evolución del tratamiento del paciente. Además, esta terapia busca la reducción de drogas anti-epilépticas, ya que estas pueden producir efectos secundarios en pacientes, además de existir personas que sufren de cierto tipo de epilepsia que no puede ser controlado ni con la toma de tres tipos de drogas anti-epilépticas.

Parámetros	Unidades	Rango	Valores típicos
Corriente de estimulación	mA	0 - 3.5	1 - 2
Frecuencia de estimulación	Hz	0 - 30	20 - 30
Ancho de pulso	μs	130 - 1000	250 - 500
Tiempo de estimulación	s	7 - 60	30
Tiempo apagado de estimulación	min	0.2 - 180	5

Cuadro 2: Parámetros de la terapia de estimulación del nervio vago para el tratamiento de la epilepsia

6.2.1. Partes implantables de la terapia VNS

Las partes implantables del tratamiento VNS consisten en el generador y en los electrodos. Estas partes son implantadas en el paciente por medio de una operación poco invasiva. La colocación de estas partes se puede observar en la Figura 2. El electrodo se conecta al nervio vago, enrollándolo sobre este; para realizar este procedimiento se realiza una pequeña incisión en el cuello. Para implantar el generador, se abre en la parte superior al pecho izquierdo, debajo de la clavícula, en donde se coloca de manera subcutánea.

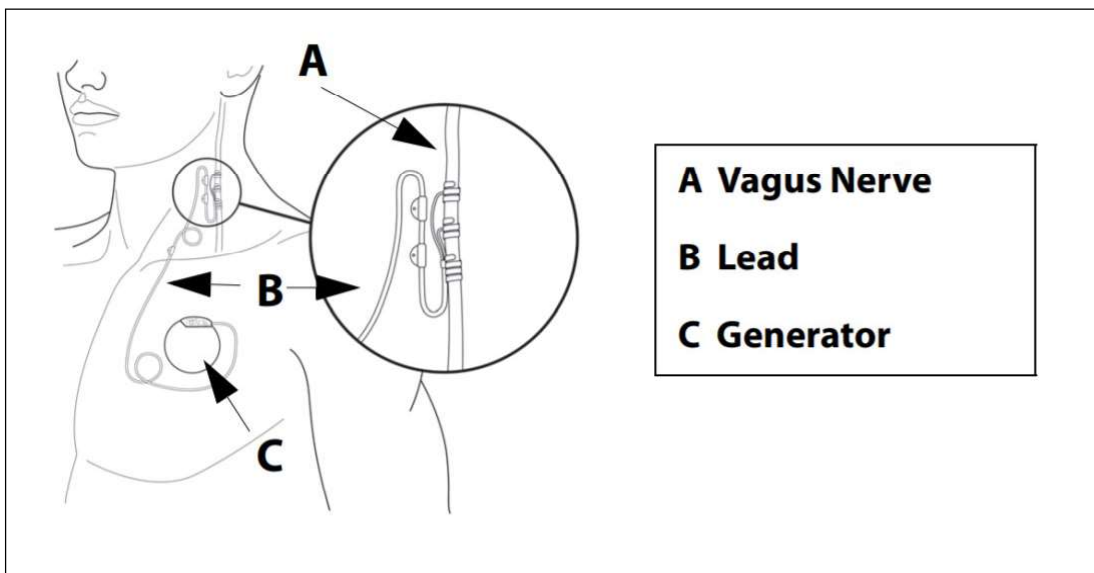


Figura 2: Implantación del generador y colocación del electrodo [19]

Generador

El generador del nervio vago es el dispositivo que se encarga de mandar el pulso eléctrico a través del electrodo conectado al nervio vago. Este se puede observar como C en la Figura 2. El estimulador además de encargarse de producir la señal eléctrica, este controla la interfaz de comunicación con la varilla de programación (Programming Wand) de la Figura 4. El generador se podría considerar la cabeza o control de la terapia VNS.

A lo largo de los años el generador ha sido mejorado, en dimensionas, vida útil y funciones. Las dimensiones de los modelos previamente mencionados se pueden observar en el Cuadro 3.

	Modelo 102/105/106	Modelo 103	Modelo 104
Dimensiones	2.0 in x 2.0 in x 0.27 in	1.8 in x 1.3 in x 0.27 in	1.8 in x 1.6 in x 0.27 in
Peso	25 g	16 g	17 g

Cuadro 3: Dimensiones y peso de los modelos del generador[20]

El generador consiste de varios componentes electrónicos los cuales se pueden observar en la Figura 3. La parte lógica y de control, mostrada como D en la Figura 3 es la encargada del funcionamiento general del estimulador, esta se encarga de la generación de la señal, manejo de tiempos de reposo, actualización de parámetros de la estimulación y la vez lleva un registro de datos pertinentes a la implantación, como historial, fecha de implantación y número de identificación del paciente. El componente que representa la parte lógica y de control es un microcontrolador. Este microcontrolador es la base del dispositivo generador, pues es el componente que dirige qué y cuándo se activa cada módulo según el firmware que este corre.

Además, el generador cuenta con la interfaz hacia los electrodos, un regulador de voltaje y su batería. La antena que se representa como E en la Figura 3 es solamente la representación de la comunicación inalámbrica, pues el generador contiene un módulo RF el cual es controlado por el microcontrolador para el envío y recepción de parámetros y datos entre el generador y el módulo de programación conocido como "Programming Wand", el cual es la interfaz externa desde donde el médico programa el generador implantado.

Por último, se tiene un Reed Switch (G), el cual es utilizado para que el generador interactúe ante la presencia del imán. Este switch es especial, y por lo general es un contacto normalmente abierto, por lo cual al acercar un campo magnético de cierta magnitud, calibrado especialmente para funcionar con el imán de la terapia de VNS, se cierra el circuito, así activando una interrupción en el microcontrolador, con la cual posteriormente este activa la estimulación "On Demand". Posteriormente al retirar el imán, el contacto vuelve a abrirse.

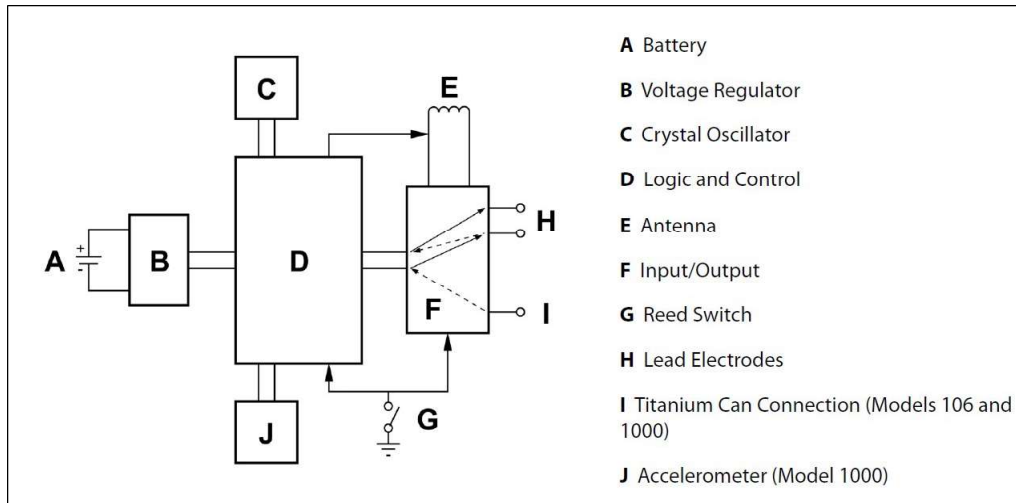


Figura 3: Diagrama del circuito del generador [20]

6.2.2. Partes no implantables de la terapia VNS

El generador implantado se complementa de otras partes externas. Las cuales se logran observar en la Figura 4. Estas partes tienen funciones específicas las cuales se explican a continuación.

Sistema de programación

El sistema de programación es la parte de la terapia VNS que se utiliza para la modificación y configuración de la terapia. A través de estos módulos, el doctor puede adaptar la terapia a la evolución de ésta en el paciente.

La computadora es utilizada para que el doctor configure los parámetros que desea que reciba el generador. Luego estos, son codificados de tal manera que el generador pueda interpretarlos. La comunicación se da de manera inalámbrica, razón por la que se necesita de una antena, la cual se encuentra incorporada en el Programming Wand.

Imán

El imán se utiliza para complementar el modo de estimulación normal. Este activa el modo “On Demand”, o más bien conocido como Modo Imán, que es el modo que le sirve al paciente a activar el estimulador al él tener un aura que le avisa que está a punto de tener una convulsión. Este imán puede ser portado como un reloj, así tener fácil acceso a este, y por lo general puede ser llevado por el mismo paciente o algún familiar. En el momento de que el paciente tenga un aura acerca de una convulsión por venir o la misma convulsión, se pase el imán sobre el estimulador encadenando así una estimulación (generalmente de mayor intensidad) que pueda, ya sea prevenir o reducir la intensidad de la convulsión.

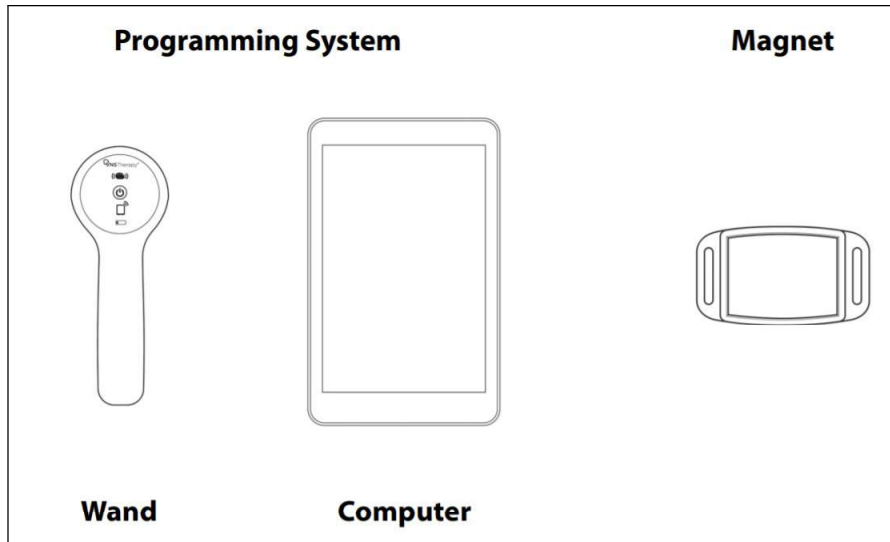


Figura 4: Partes no implantables del tratamiento VNS [20]

6.2.3. Modos de operación

Los modos de operación que presentan los modelos 101 al 105 son dos, el modo normal y el modo activado por imán. Cada modo es generalmente configurado utilizando diferentes parámetros. Estos modos son configurados por el doctor, entre ellos los parámetros de frecuencia, tiempo de estimulación e intensidad del pulso. En conjunto, los modos normal y el modo en demanda activado por el imán logran mejorar considerablemente la calidad de vida del paciente.

Modo normal

El modo de estimulación normal es el que rige como se estimula el nervio vago diariamente. Este modo es el que ayuda a reducir y prevenir las convulsiones pues está funcionando en todo momento. El modo normal se configura normalmente para que estimule durante 30 segundos cada 5 minutos.

Parámetros	Valores programables
Corriente de estimulación (mA)	0.0 a 3.5 $\pm 10\%$
Frecuencia de estimulación (Hz)	1, 2, 5, 10, 15, 20, 25, 30 $\pm 6\%$
Ancho de pulso (μs)	130, 250, 500, 750, 1000 $\pm 10\%$
Tiempo de estimulación (s)	7, 14, 21, 30, 60 $+7s/ - 15\%$
Tiempo de apagado (min)	0.2, 0.3, 0.5, 0.8, 1.1, 1.8 y 3; 5 a 60 en incrementos de 5; 60, 90, 120, 150 y 180 $(+4.4/ - 8.4s$ o $\pm 1\%)$

Cuadro 4: Parámetros de estimulación modo normal de los modelos 102 al 105 de LivaNova [20]

Los parámetros programables que se observan en el Cuadro 4 se ven ejemplificados en la gráfica mostrada en la Figura 5. Todos estos parámetros pueden ser modificados de manera

independiente, abriendo así al médico gran cantidad de ajustes adecuados para lo que crea conveniente para cada paciente.

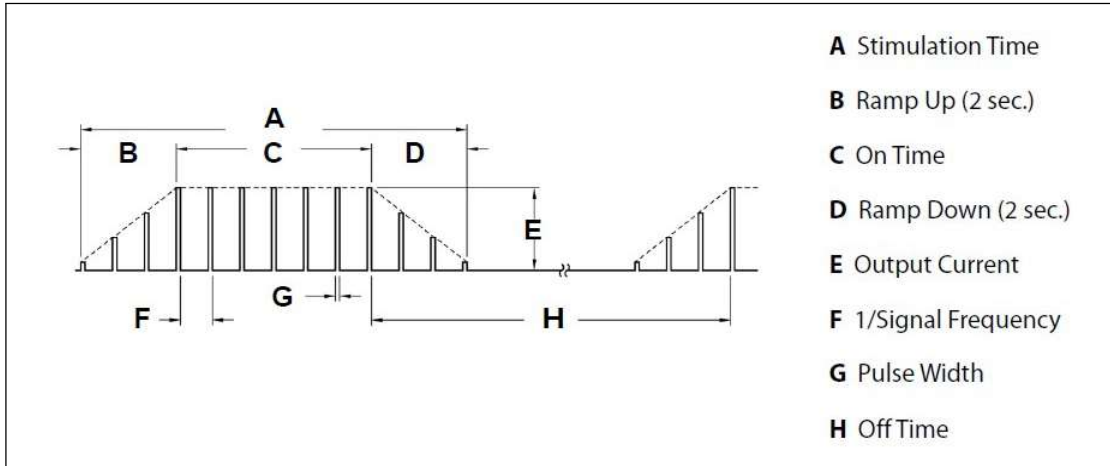


Figura 5: Señal de estimulación y sus parámetros [20]

Modo activado por imán

El modo activado por imán ayuda a que el paciente prevenga o acorte una convulsión. Hay muchos pacientes que sienten un aura que les avisa que tendrán una convulsión, por lo cual, al pasar el imán especial sobre el generador, este responde mandando un pulso de igual manera al nervio vago, el cual genera una respuesta positiva que muchas veces corta y logra prevenir los ataques. Este modo es configurado de igual manera por el doctor, y generalmente es una estimulación más intensa.

6.3. Diseño de sistemas de baja potencia

La definición de sistemas de baja potencia es muy amplia y depende de la aplicación. Por ejemplo, en algunos sistemas la energía no es escasa, pero el diseño de baja potencia se enfoca en reducir los costos de operación y maximizar la eficiencia. Para el caso del estimulador del nervio vago, la energía es limitada, pues una batería es la que alimentará al dispositivo, por lo cual el diseño de baja potencia se encarga en optimizar la vida útil del dispositivo.[21]

6.3.1. Principales fuentes de consumo de energía

En dispositivos CMOS, como lo son los microcontroladores, el consumo de energía se puede dividir en dos categorías: potencia dinámica y potencia estática.

Potencia dinámica

El consumo de potencia dinámica es la corriente que es consumida durante la operación normal de un microcontrolador. Este consumo incluye la energía debido a las transiciones de los circuitos CMOS así como la corriente de cada uno de los módulos analógicos del dispositivo, como los osciladores o los módulos de conversión analógica a digital. La siguiente ecuación define la potencia dinámica consumida por una compuerta, en donde V es el voltaje

de alimentación del sistema, f es la frecuencia de operación y C es la capacitancia de carga.

$$P = V^2 \times f \times C \quad (1)$$

Un diseñador tiene poco control sobre la capacitancia de carga del sistema, por lo cual debe controlar los otros dos parámetros que influyen al consumo de la potencia dinámica. De la fórmula anterior podemos inferir que el voltaje tiene el mayor impacto en la potencia dinámica que utiliza el sistema, por ende, al reducir el voltaje de alimentación del sistema se logra reducir de manera significativa la potencia del sistema. De igual manera, reduciendo la frecuencia de operación del sistema se reduce la potencia del sistema. Para seleccionar la frecuencia de operación hay varios factores a tomar en consideración, como lo son los requerimientos de comunicación o de muestreo, salidas PWM o de desempeño.

De lo anterior podemos concluir que un diseño eficiente energéticamente tomará ambos factores en cuenta, reduciendo el voltaje y la frecuencia al menor valor posible, así lograr un mejor consumo de potencia del dispositivo.[21]

Potencia estática

El consumo de potencia estática enmarca toda la potencia que requiere el sistema para un funcionamiento correcto mientras el sistema no está ejecutando código. Esta incluye típicamente las corrientes de polarización para circuitos analógicos, relojes de baja potencia y corrientes de fuga. Para sistemas alimentados por baterías, la potencia estática es de gran preocupación, pues los dispositivos se encuentran en modo Sleep la mayor parte de su vida.

Además, el temporizador perro guardián o más conocido como el Watchdog Timer (WDT) se toma como parte del consumo de potencia estática, pues a pesar de que este módulo está constantemente operando a frecuencia constante, este es de bajo consumo. El WDT es uno de los módulos del microcontrolador que puede despertar al dispositivo cuando este entra en modo Sleep después de cierto tiempo configurado en el programa que está corriendo.[21]

Las corrientes de fuga son causadas debido a la operación no ideal de los transistores MOSFETs usados en dispositivos que operan con lógica CMOS; estas corrientes de fuga pueden llegar a ser de varios μA de magnitud. Con la reducción del tamaño de la tecnología de los transistores, los cuales son de tamaños nanométricos, se encuentran dos grandes fuentes de consumo de potencia estática, siendo las corrientes de fuga de Sub-Threshold y de Gate. Estos factores no son controlables para el diseñador de los sistemas de baja potencia, pero este puede tomarlo como punto de partida para la selección de sus componentes. Para tecnologías mayores, la potencia dinámica es de mayor impacto, pero en tamaños menores, la estática es el problema, es por eso, que el reducir el voltaje es de gran importancia.

6.3.2. Diseño de software

El diseño de Software tiene un gran impacto a la hora del diseño de sistemas de baja potencia. Muchas veces solo se le da importancia a la selección del mejor dispositivo de baja potencia y optimizar el hardware, pero esto fácilmente se puede desperdiciar si no se utilizan las técnicas correctas a la hora de escribir el software del sistema.

Ejecución de código condicional

Una técnica normalmente utilizada es la de utilizar despertar condicionado y ejecución de código. Esto se realiza para que se ejecute el código después de ser despertado el dispositi-

tivo de cierta manera. De esta manera, el sistema puede minimizar el consumo de potencia, realizando la función que tiene que hacer lo más rápido posible, y luego regresar al estado Sleep. Además, es importante evitar que las operaciones que consumen más energía se realicen con poca frecuencia.

Adicionalmente, el sistema puede utilizar un sistema de reloj variable, así, cuando se realicen operaciones que consumen tiempos fijos, como escribir a EEPROM o conversiones A/D, se reduce la frecuencia de operación sin aumentar el tiempo de ejecución del programa.

Interrupciones contra polling

El término polling se utiliza para explicar el chequeo de variables, es decir, se chequea que el valor de cierta variable se cumpla para luego ejecutar la función definida por este valor. En cambio, las interrupciones se utilizan para que el programa siga su flujo normal, y cuando el sistema detecte automáticamente una interrupción, este detiene lo que realiza, y toma acción y ejecuta la función programada para cuando ocurra esta interrupción; luego de terminar, el programa regresa al punto donde se encontraba y continuar con la operación que realizaba previo a la interrupción.

Para evitar el uso de ciclos de espera que desperdicien energía, es importante que el diseñador evite un sistema basado en polling y utilice en cambio un sistema basado en interrupciones. Al utilizar interrupciones, se pueden utilizar modos de ahorro de energía, como lo son Sleep y Idle.

Modos de operación de un MCU

Como se menciona en la sección de potencia estática, las corrientes de fuga juegan un gran papel en la potencia estática que el MCU disipa, y estas dependen del nodo tecnológico del chip, su cantidad de transistores y de las técnicas que tenga este para la reducción de corrientes de fuga. Además, en la sección de Potencia Dinámica se menciona la ecuación 1 que relaciona la frecuencia de operación y el voltaje con la potencia dinámica del sistema. La suma de las potencias dinámica y estática nos dan el consumo total de energía del sistema. Es por esta razón que un MCU cuenta con varias técnicas y modos de operación para la reducción de potencia de si mismo. A continuación se presenta en el Cuadro 5, en donde se resumen las características de estos modos específicos a los de la marca Microchip.

Modo	CPU clock	Peripheral clock	Memoria	Retorno
Idle	Detenida	Funcionando	Retenida	Sig. instrucción
Sleep	Detenida	Detenida	Retenida	Sig. instrucción
Deep sleep	Detenida	Detenida	Retenida	Sig. instrucción

Cuadro 5: Características de los distintos modos de operación de un MCU

Modo sleep

En el modo sleep se apagan la mayoría de periféricos del MCU. Estos módulos son dependientes del microcontrolador utilizado, pues algunos no tienen el módulo deep sleep que consume menor potencia que el sleep normal. La ventaja de estos modos es que cuando no hay necesidad de hacer operaciones, esperar algún tiempo o evento, se puede entrar en sleep, apagando los periféricos y no ejecutar más código, así luego de haberse cumplido la

condición deseada, reanudar con la ejecución del programa. El modo sleep normal pueden conservar el estado de los registros de configuración, así, al despertar, no tener que configurar el dispositivo nuevamente.

Mode idle

El modo idle a diferencia del modo sleep es que este deja los módulos activos y funcionando el reloj que alimenta los periféricos, así poder seguir contando, realizar conversiones, etc. La ventaja de este es que no hay un retardo al salir de este modo. Este modo reduce el consumo de potencia dinámica del sistema. Este modo puede consume aproximadamente un cuarto de la corriente de operación normal.

Técnicas complementarias

A parte de utilizar las prácticas anteriormente mencionadas, existen otras formas de reducir el consumo energético del microcontrolador. Existen microcontroladores que cuentan la opción de deshabilitar periféricos que no se utilizan, llamado Peripheral Module Disable. La ventaja que da esta opción, es limitar el uso de energía solamente a los módulos que se necesitan, ya que por lo general, los microcontroladores tienen más módulos que no son totalmente necesarios para la aplicación.

Otra técnica que es recomendable, es que todos los puertos que no se utilizan se configuren como salidas digitales y su estado sea 0 o 1. Esto se realiza ya que si se configuraran como entradas, estas podrían tener corrientes de fuga hacia el MCU.

6.3.3. Diseño de hardware

Para el diseño de hardware eficiente se tienen que cumplir con los siguientes requerimientos:

- Maximizar la impedancia en tracks de corriente
- Minimizar impedancia en tracks de alto nivel de switcheo
- Minimizar corrientes de fuga
- Minimizar el ciclo de trabajo de operación

Si se utilizan botones, se recomienda utilizar la resistencia de pull-up del MCU, ya que esta se puede controlar en software. Además, los botones, debido a que al ser presionados por humanos, la duración de un pulso, puede durar varios cientos de milisegundos, consumiendo gran cantidad de energía. Por otro lado, es recomendable utilizar LEDs de alta intensidad y que estos sean manejados con baja corriente, 1 mA aproximadamente, o manejados por un PWM donde se reduzca el ciclo de trabajo.

Los pines IO, se inicializan como entradas, y estos consumen la menor cantidad de energía cuando se encuentran cerca de VDD o VSS, en cambio, si estos se encuentran a un voltaje entre VDD o VSS, el transistor dentro de la entrada digital se encontrará en la región lineal, consumiendo considerablemente más corriente. Como se mencionó en la sección 6.3.2, lo más práctico es dejar los pines sin conectar y colocarlos como salidas con un valor de VDD o VSS, en caso de ser configurado como entrada, se debe colocar una resistencia de 10K Ω a VSS o VDD.

Para las entradas o salidas que operan a alta velocidad, se debe reducir la capacitancia parásita, minimizando el largo del track y eliminando componentes innecesarios. Cuando se utilicen capacitores, escoger capacitores de baja corriente de fuga, generalmente estos son cerámicos u otros conocidos como film capacitors en inglés.

Debido a que el consumo de energía del sistema no solo lo representa el MCU, es importante seleccionar adecuadamente los componentes adicionales que se utilizarán. Además, para sistemas que consumen poca energía, es conveniente que el MCU controle los circuitos externos, ya sea un transreceptor, memoria EEPROM, sensores, etc. Si el dispositivo consume mucha corriente, se utiliza un mosfet el cual es controlado por el MCU.

La selección de la fuente de alimentación para un sistema de baja potencia juega un gran rol en el consumo de energía del sistema, esto ya que los reguladores lineales, PMCIs todos requieren energía para operar, y generalmente están diseñados con sistemas de alta potencia en mente. Los reguladores lineales son los más eficientes en el rango de corrientes en el que operan los sistemas de baja potencia, aunque lo mejor es evitar el uso de uno y operar al voltaje que ofrecen las baterías de LiFeS2 o LiMnO2. Al operar al voltaje que provee la batería se puede aumentar considerablemente la vida útil del sistema. Lo que se pierde al operar solamente con la batería, es que el sistema no tiene una referencia clara a lo largo del desgaste de la batería y los demás componentes deben soportar los mismos rangos de operación.

A la hora de seleccionar una batería, las de Litio, tienen una curva de descarga bastante plana, es decir, el voltaje que la batería puede ofrecer es prácticamente constante a lo largo de su vida útil.

6.3.4. Power budgeting

Power budgeting por su nombre en inglés, se refiere a presupuestar la cantidad de energía que el sistema disipará. Esta técnica es crucial para poder elegir los componentes necesarios, programar de manera adecuada con tal de obtener la vida útil que se desea, etc. Para realizar el power budgeting es necesario calcular la corriente que consumirá el sistema durante un periodo definido, luego, se divide dentro de este periodo, así obteniendo la corriente promedio que utilizará el sistema. Con la corriente promedio, se puede proceder a utilizar la ecuación 2 para encontrar la vida útil del sistema o calcular el tamaño de batería necesario para obtener cierto tiempo deseado.

$$Life(horas) = \frac{Capacidad(mAh)}{Corriente\ del\ sistema + Auto-descarga\ Bat(mA)} \quad (2)$$

6.4. PWM

PWM o modulación por ancho de pulsos de una señal o fuente de energía es una técnica frecuentemente utilizada en ingeniería que se utiliza para modificar el ciclo de trabajo de una señal, así controlar la cantidad de energía que se transmite. El ciclo de trabajo se define como D, el cual es el porcentaje de la señal activa respecto al periodo de la señal.

$$D = \frac{\tau}{T}$$

Como se puede observar en la Figura 6, se observan varias señales de PWM con distintos ciclos de trabajo, de esta manera se transmite distinta cantidad de energía dependiendo del

ciclo de trabajo. La energía se ve representada en el voltaje que el circuito estimulado siente entre sus terminales, es decir, si tenemos un PWM de 5V de amplitud y 50% de ciclo de trabajo, el voltaje que el circuito aprecia es de 2.5V, pues es el voltaje promedio de señal.

La técnica de PWM es utilizada en gran cantidad de campos, desde controlar la intensidad del brillo de una pantalla o la velocidad de un motor.

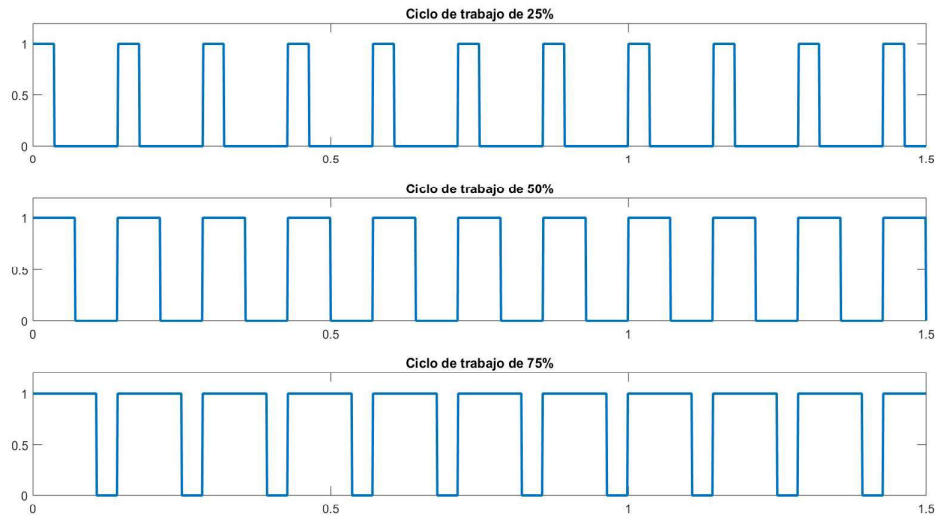


Figura 6: Señal de PWM con distintos ciclos de trabajo

6.4.1. Resolución de tiempo del PWM

Debido a que el PWM generalmente es realizado utilizando MCUs, este tiene una cantidad de pasos finitos, o en otras palabras, una cantidad finita de anchos de pulsos distintos. Además, la resolución del PWM se reduce al tener una frecuencia mayor, pues los pasos en los que varía el ancho de pulso son siempre iguales sin importar la frecuencia de la señal.

6.5. Memoria EEPROM

La memoria EEPROM por sus siglas en inglés significa “Electrically Erasable Read-Only Memory” es un tipo de memoria no volátil utilizado en computadoras, microcontroladores u otros dispositivos que permite almacenar información que puede ser reescrita de manera individual. La EEPROM se organiza como arrays de Floating-gate Mosfets. Estas memorias tienen ciclos limitados de lectura y escritura. Una memoria de este tipo en los dispositivos de la familia PIC 16F o 18F tiene como mínimo un millón de ciclos de lectura y escritura, además de poder retener la información en la memoria por 40 años.

6.6. Estándar BARR-C

El estándar BARR-C es un estándar utilizado a la hora de programar software de sistemas embebidos. Su propósito es la reducción de defectos en la programación de este tipo de sistemas. Al seguir este estándar, el programador no solamente reduce el peligro que se puede

causar al usuario y el tiempo de depuración sino facilita el mantenimiento y portabilidad del código a otros dispositivos. Este estándar no se basa en las preferencias de estilo del autor sino en la manera en la cual el estilo y forma del código pueda minimizar la cantidad de defectos en el código.

A continuación se presenta un pequeño resumen de las que se consideran las normas más relevantes mencionadas en el estándar.

6.6.1. Reglas generales

Las reglas generales son las que definen los aspectos generales del programa, y más a la redacción que a su uso.

- Cumplir con el estándar ISO C99
- No alterar o renombrar palabras claves del lenguaje con la directiva `#define`
- Ancho máximo de 80 caracteres por línea
- Brackets y paréntesis
 - Brackets siempre rodean bloques de código: `if`, `else`, etc.
 - Brackets se colocan en una línea solos
 - No confiar en reglas de precedencia de C
 - Usar paréntesis siempre que haya un `&` o `||`
- Utilizar abreviaciones o acrónimos solamente si son entendibles en el lenguaje ingenieril
- Evitar el uso de: `goto`, `auto`, `register` o `continue`
- Utilizar de palabras claves
 - Utilizar *Static* para variables o funciones que no serán vistas fuera del módulo
 - Utilizar *Const* para variables que no cambian luego de inicializadas
 - Utilizar *Volatile* para variables accesibles por ISRs o más threads

6.6.2. Tipos de datos

Al utilizar el estándar en los tipos de datos se evita problemas de overflow, comparaciones, etc. Además, ayuda a identificar que tipo de variable se utiliza y no dejarlo a la ambigüedad del compilador.

- Enteros
 - Seguir convención mostrada en el Cuadro 6
 - Campos de bits no se definen como enteros con signo
 - No se utilizan operadores de bits para manipular enteros con signo
 - No combinar enteros con y sin signo en comparaciones o expresiones
- Punto flotante
 - Evitar su uso

- Utilizar: `float_32t`, `float_64t` o `float_128t`
 - Nunca comparar igualdad o desigualdad de puntos flotantes
- Booleanos
 - Declarar como *bool*
 - Valores no booleanos deben ser convertidos a booleanos utilizando operadores de relación y no por casting

Tamaño del entero	Entero con signo	Entero sin signo
8 bits	<code>int8_t</code>	<code>uint8_t</code>
16 bits	<code>int16_t</code>	<code>uint16_t</code>
32 bits	<code>int32_t</code>	<code>uint32_t</code>
64 bits	<code>int64_t</code>	<code>uint64_t</code>

Cuadro 6: Tamaño de enteros según C99

6.6.3. Reglas de sentencias

Estas reglas se pueden observar a la hora de declarar variables, ciclos, etc. Estas reglas ayudan a que el compilador interprete correctamente lo que especificamos al programar.

- No usar coma (,) al declarar variables
- Loop
 - No utilizar números como valores iniciales o valores finales, utilizar variables
 - Cualquier loop sin código debe ser rodeado de brackets y un comentario explicando el porque no hay código
 - Loops infinitos se implementan con *for(;;)*
- Switch
 - El break para cada caso lleva la misma alineación que el caso mismo, no alineado al contenido
 - Todos los bloques switch tienen que tener un caso *default*
- No realizar asignaciones en un *if* o *else if*

6.6.4. Reglas de variables, procedimientos y módulos

Este grupo de reglas se aplica a los nombres de funciones, procedimientos así como de requisitos que deben cumplir estos.

- Variables
 - Ningún procedimiento debe tener nombre de una palabra clave en C
 - Los nombres no inician con `_`
 - Nombres deben de ser entre **3** y **31** caracteres

- No utilizar mayúsculas y separar palabras con `_`
 - Describen su propósito en su nombre
 - Variables booleanas inician con `b_` y llevan la pregunta que contestan como nombre
 - Se inicializan antes de usarse
 - Variables globales es preferible que sean declaradas al inicio todas juntas
- Procedimientos
 - No deben ser mayores a **31** caracteres
 - No se usan mayúsculas en funciones, los macros no llevan minúsculas
 - El nombre describe lo que realiza cada procedimiento
 - Cada parámetro debe ser declarado explícitamente y nombrado con sentido
- Módulos
 - Se debe mantener un estilo a lo largo del proyecto
 - Cada módulo debe ser único en sus primeros **8** caracteres
 - Todos los archivos deben tener ya sea el sufijo `.h` o `.c`
 - Headers llevan el mismo nombre que el archivo fuente
 - El header identifica cada función y variable usada
 - El source file debe de tener las siguientes partes o alguna ellas:
 - Comentario de bloque
 - Inclusiones
 - Definiciones de tipos de datos, constantes y definiciones de macros
 - Declaración de variables estáticas
 - Prototipos de funciones privadas
 - Cuerpo de las funciones públicas y/o privadas

6.7. Entorno de microchip

La empresa Microchip ofrece varias herramientas de software y hardware para el desarrollo de aplicaciones que complementan los chips que fabrican. Microchip fabrica gran variedad de microcontroladores, desde MCUs de 8 bits hasta más complejos de 32 bits. Además, ofrecen módulos analógicos, memorias y módulos de comunicación. Los microcontroladores de gama media, de 8 bits, son los de las familias 16F y 18F. Estos son superiores en características a los 10F y 12F. Microchip ofrece otras alternativas a estos los cuales son los AVR, los cuales tienen una programación más sencilla pues tienen varios acumuladores. Algo a tomar en consideración es que la IDE para programar AVR no es tan amigable con el usuario como MPLAB X.



Figura 7: Logo de Microchip

6.7.1. MPLAB X IDE

MPLAB X IDE es la plataforma que se utiliza para programar los microcontroladores de las familias PIC. Tiene bastantes características que facilitan al programador interactuar con el software antes de ser exportado al MCU, pues este cuenta con un simulador. El simulador permite al programador observar el flujo del programa observando como se modifican registros y variables, así comprobar el funcionamiento correcto del programa. Para programar los PICs se puede utilizar lenguaje *Assembler* o *C*.

MCC

MCC es una herramienta de MPLAB X IDE la cual permite al programador de manera fácil configurar el microcontrolador a utilizar. Con esta herramienta se puede configurar los módulos que se utilizaran, como operarán etc. Por ejemplo, al configurar el TIMER 1, el programador puede elegir a que frecuencia funcionará, con que reloj funcionará, el pre-escalador, el tiempo que tiene que contar, si se usarán interrupciones, etc. Además, permite configurar la función de puertos I/O, si estos son digitales o analógicos, tienen interrupciones en cambio de estado, etc.

Luego de configurar todas las opciones deseadas, el software genera el código que realiza la configuración del dispositivo. Este software generado contiene las funciones prototipo y sus definiciones, lo cual reduce el tiempo de desarrollo enormemente. En la Figura 8 se puede observar la interfaz en donde se selecciona que puertos se utilizan, si se utiliza WDT, la frecuencia del sistema, etc.

El software que genera MCC está listo para que el usuario inicie a programar su aplicación. MCC tiene ciertas desventajas, como la incompatibilidad con varios modelos de microcontroladores y que genera funciones que no serán utilizadas por el usuario. MCC genera el código en *C* para que se use con el compilador XC8.

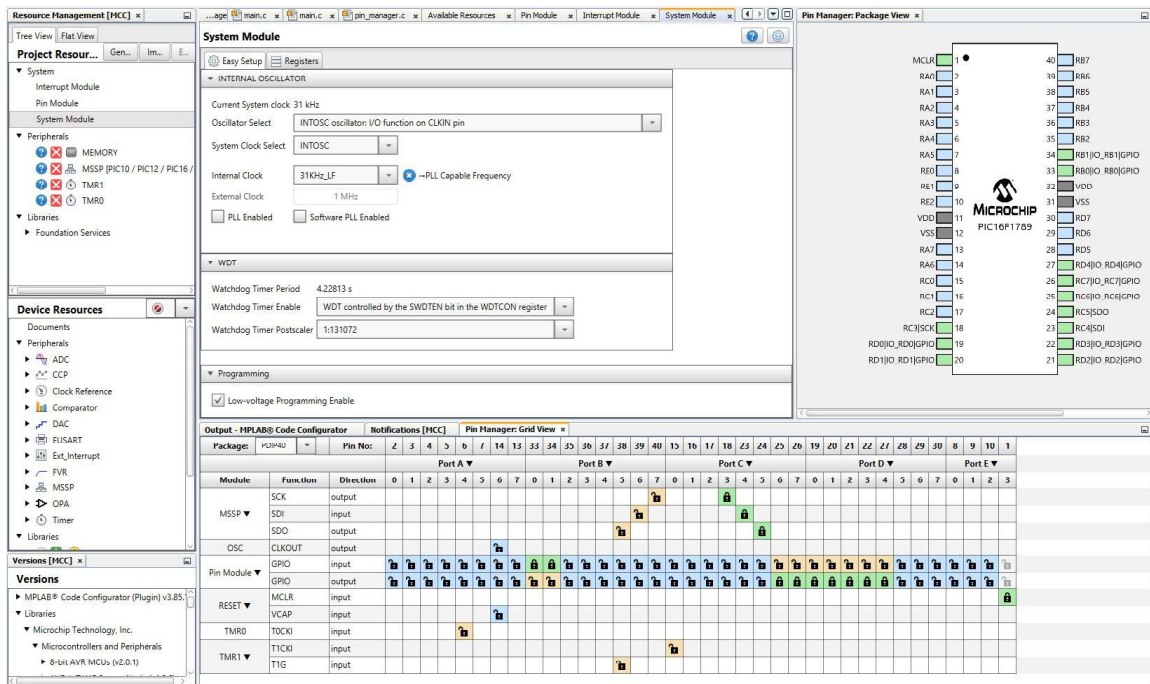


Figura 8: Interfaz de configuración de hardware de MCC

6.7.2. Tecnología eXtreme Low Power XLP

Para los microcontroladores de Microchip, la tecnología XLP es un estándar que ha evolucionado constantemente para brindar soluciones de baja potencia al desarrollador. Las características [22] que entran dentro de esta tecnología son las siguientes:

- Idle mode
- Sleep mode
- Deep Sleep mode
- Low-Voltage/Retention Sleep mode
- V_{BAT} mode
- INTOSC de alta frecuencia con PLL y postcaler programable
- WDT con time-out extendido
- ULPWU
- Oscilador secundario de 32 kHz
- BOR controlable en software

Con todas estas características [22] se logra una gran reducción de consumo de potencia. Los dispositivos catalogados como XLP cumplen con corrientes menores a:

- 100 nA para corriente de apagado
- 500 nA para corriente de WDT
- 400 nA corriente para RTCC

Microchip, con su tecnología XLP usan combinación de técnicas propias de los procesos de diseño así como de manejo de potencia para reducir la potencia en donde sea posible. Una parte clave es el uso adecuado de los modos de operación mencionados en la sección 6.3.2 donde se habla del diseño de software energéticamente eficiente.

A pesar de estas técnicas nos ayudan a la reducción de potencia. ya es responsabilidad del diseñador hacer que la aplicación diseñada sea lo más eficiente posible, es por esta razón que se evalúa cuando se debe utilizar Deep sleep mode o sleep mode. El problema de deep sleep mode es que el programa debe inicializarse desde el inicio, pudiendo tener mayor consumo de corriente que sleep mode.

Para esto se utiliza el ejemplo mostrado en la Figura 9, en donde se compara cuando es más eficiente usar cada modo. Deep sleep se selecciona cuando la duración del periodo de reposo es mayor a T_{BE} .

$$Q_{SLP} = T_{PD} \times I_{PDSL P}$$

$$Q_{DS} = (T_{INIT} \times I_{DD}) + (T_{POR} \times I_{POR}) + (T_{PD} \times I_{PDDS})$$

$$T_{BE} = T_{PD} = \frac{(T_{INIT} \times I_{DD}) + (T_{POR} \times I_{POR})}{I_{PDSL P} - I_{PDDS}}$$

Where: Q_{DS} = Total Charge Spent in Deep Sleep
 Q_{SLP} = Total Charge Spent in Sleep
 T_{BE} = Breakeven Time (interval at which $Q_{DS} = Q_{SLP}$)
 T_{INIT} = Initialization Time to Resume Full-Power Operation
 T_{PD} = Sleep or Deep Sleep Period (defined by context)
 T_{POR} = Time Required for Power-on Reset
 I_{POR} = POR Current (inrush current included)
 $I_{PDSL P}$ = Static Current in Sleep mode
 I_{PDDS} = Static Current in Deep Sleep mode

Figura 9: Cálculo de T_{BE} para comparación entre deep sleep y sleep mode [22]

6.8. Serial Peripheral Interface - SPI

Este es un estándar de comunicación serial que se usa generalmente para la comunicación de ICs en dispositivos electrónicos. Esta comunicación se da gracias a 4 señales distintas conocidas como bus SPI.

El bus SPI consiste de 4 líneas las cuales son: MOSI, MISO, SCLK y SS. Este estándar de comunicación tiene la ventaja de ser Full Dúplex, es decir, que la comunicación es bidireccional de manera simultanea. Además, este estándar permite mayores velocidades a las que ofrece I2C además de consumir menos energía que este.

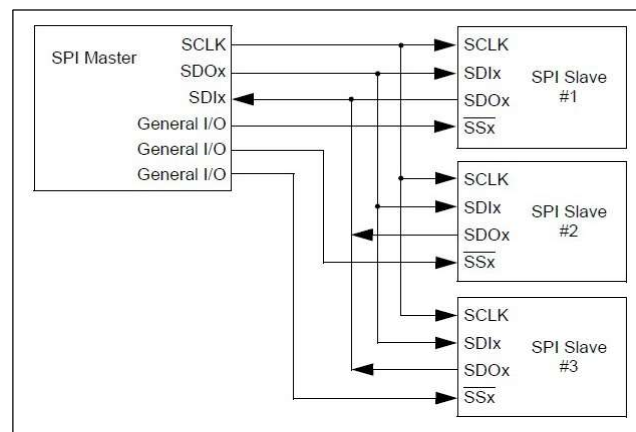


Figura 10: Diagrama de conexión SPI multi-esclavo

La comunicación SPI ocurre de la siguiente manera:

- El maestro selecciona al esclavo con el que desea comunicarse. Para esto, pone la señal SS en VSS
- El maestro manda la información a través de la línea MOSI. La información enviada está sincronizada con el SCLK

- El esclavo responde con información a través de la línea MISO

SPI se puede clasificar en 4 modos según la polaridad (CPOL) y fase (CPHA) del SCLK. Estos modos mencionados se explican en la Figura 11.¹

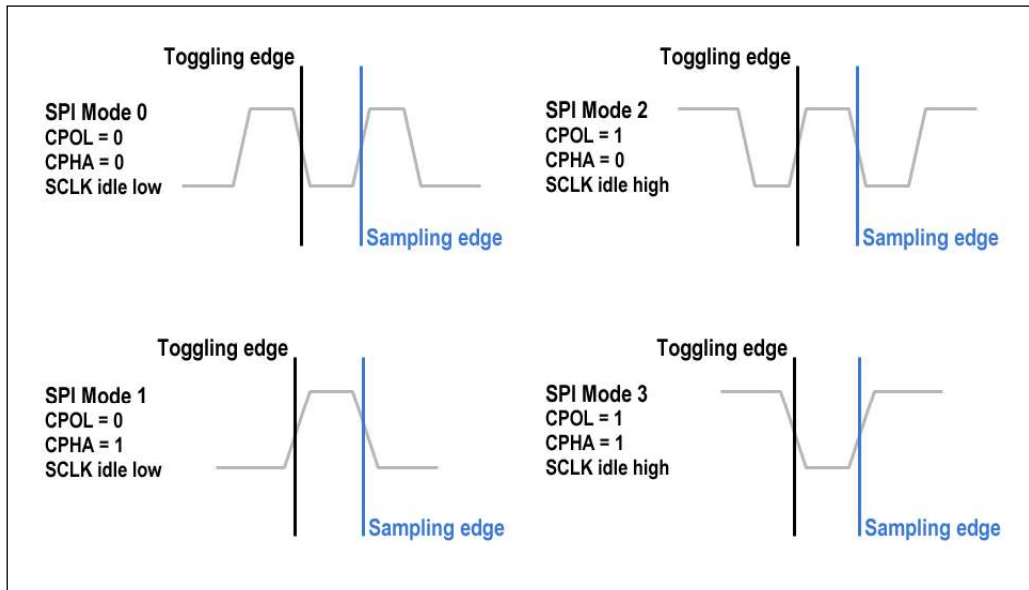


Figura 11: Modos de comunicación SPI

¹Figura obtenida de electric imp DevCenter: <https://developer.electricimp.com/resources/spi>

Para lograr una correcta implementación del Estimulador del Nervio Vago, la primera parte del trabajo consiste en el diseño del estimulador. El diseño consiste en primero identificar los objetivos y requisitos mínimos por cumplir, los cuales fueron discutidos con HUMANA. El diseño de la plataforma para la terapia VNS se basará en la utilizada por el modelo 102 de LivaNova.

Luego de haber identificado los requisitos se procederá a evaluar varios posibles micro-controladores que puedan cumplir con los requisitos. Después de esto, se diseñará un plan de trabajo el cual se basa en prototipos. Primero se realizará el primer prototipo, y en base al prototipo actual, se planteará el siguiente, así modificándolo con el fin de ir cumpliendo los requisitos. Para cada prototipo se determinará que metas se desean cumplir, para que posteriormente se evalúen sus logros.

7.1. Identificación de requisitos

Los requisitos identificados para la realización serán explicados a detalle a continuación.

- Bajo consumo energético
- Generación de la señal de estimulación
- Coordinación de módulos de Potencia y Comunicación Inalámbrica

7.1.1. Bajo consumo energético

Uno de los aspectos más importantes, sino tal vez el más importante, es que el sistema consuma la menor cantidad de energía posible. El estimulador tiene que cumplir todas sus funciones, pero hacerlo de manera eficiente. Para realizar esto, se debe considerar los aspectos observados en las secciones 6.3.2 y 6.3.3, desde un buen manejo de software, utilizando interrupciones, seleccionando de manera correcta la frecuencia de operación y a la vez utilizar modos de ahorro de energía.

Además, para la comunicación inalámbrica se debe seleccionar un módulo de baja potencia que preferiblemente cuente con interfaz de comunicación SPI, pues esta interfaz, puede

utilizar velocidades mayores que las que EUSART proporciona, así poder terminar las transacciones lo más rápido posible con el objetivo de regresar a los modos de ahorro de energía. El estudiante Andrés Girón desarrollará el Programming Wand y la interfaz de programación con la que se comunicará con el módulo RF implantado; además de esto, se encargará de seleccionar y configurar los módulos para la comunicación RF cumpliendo con los requisitos de potencia anteriormente mencionados.

En conclusión, para cumplir con el bajo consumo de potencia del sistema, se deben seleccionar componentes electrónicos (MCU, módulo RF, etc) capaces de operar en rangos de voltaje de baterías de estándar médico. Para el MCU es conveniente que este tenga varios osciladores internos para poder cambiarlos a la hora de realizar distintas funciones, por ejemplo una frecuencia alta para la interfaz SPI con el módulo RF y una frecuencia baja para la ejecución del programa de estimulación.

7.1.2. Generación de la señal de estimulación

Para la generación de la señal de estimulación se tienen que cumplir con todos los parámetros especificados en el Cuadro 4, pues el estimulador se basa en el modelo 102 de LivaNova. Se replicarán todos los parámetros, a excepción de la corriente de estimulación, pues este parámetro será trabajado en el trabajo del Módulo de Potencia del estimulador realizado por el estudiante Hugo Elvira. Para poder reproducir los parámetros de la señal de estimulación se necesita de un MCU tenga como mínimo 2 módulos de temporizador como mínimo, aunque es preferible que el MCU posea más en caso de ser necesarios para otra función o algún imprevisto. La señal de estimulación puede ser generada por un PWM en caso de que este cumpla con todos los parámetros de tiempo de la señal, por lo cual se requiere de al menos 1 módulo PWM. Es posible que además del PWM anterior se necesite otro para la interconexión con el módulo de potencia.

Además, para la modificación de parámetros, será necesario tomar en cuenta el módulo de comunicación inalámbrica, que será externo y será controlado por el MCU. También hay que considerar que el estimulador tiene que modificar los parámetros de forma segura, pues este luego de ser configurado realizará la estimulación según los parámetros programados y hasta la siguiente cita con el médico se revisarán los parámetros y posiblemente se modificarán. Para realizar lo anterior, es importante que los parámetros de estimulación pertenezcan a un universo finito, en donde solamente estos sean los posibles que puedan estimular al paciente.

Por último, en caso de alguna falla o reseteo del dispositivo es importante que el MCU cuente con memoria no volátil, para que éste continúe estimulando con los parámetros previamente configurados.

7.1.3. Coordinación de módulos de potencia y comunicación inalámbrica

Para la coordinación entre módulos, se necesita por lo menos un periférico SPI para comunicarse con el módulo RF seleccionado. Las características del módulo RF, como lo son su frecuencia de operación, ancho de banda, etc, son explicadas en el trabajo de graduación Andrés Girón, en donde él es el encargado de seleccionar y configurar el módulo RF. Lo que se realizará en este trabajo es brindar el manejo y operación del módulo ya configurado, en donde se leerán los paquetes recibidos por éste. Todo esto será trabajado por medio de interrupciones para evitar el chequeo de banderas o “polling” como fue mencionado en la sección 6.3.2.

El manejo de la amplitud de la señal, es decir la corriente de estimulación, será trabajado

en el módulo de potencia realizado por Hugo Elvira. En este trabajo se reproducirán los parámetros de tiempo de la señal de estimulación, es decir su frecuencia, ancho de pulso, cuando y por cuanto tiempo estimulará. La interacción que se tendrá con el módulo de potencia es la de brindar la señal al módulo de potencia para que este module su amplitud según el parámetro configurado. El parámetro de la amplitud de la corriente será manejado como los otros parámetros de tiempo y se almacenará en memoria o no volátil.

Además de brindar la señal, parámetros como la impedancia del electrodo calculado por medio del módulo de potencia serán almacenados en la memoria del dispositivo para posteriormente ser enviados al módulo externo para realizar chequeos del estado del electrodo.

7.2. Diseño del estimulador

Debido a los objetivos planteados y a los requisitos de diseño identificados del Estimulador, se plantea un acercamiento basado en modos de operación. Los tres modos de operación identificados son: Modo de Estimulación, Modo de Reposo y Modo de Programación. En la Figura 12 se muestra el primer acercamiento para la replicar el funcionamiento del estimulador.

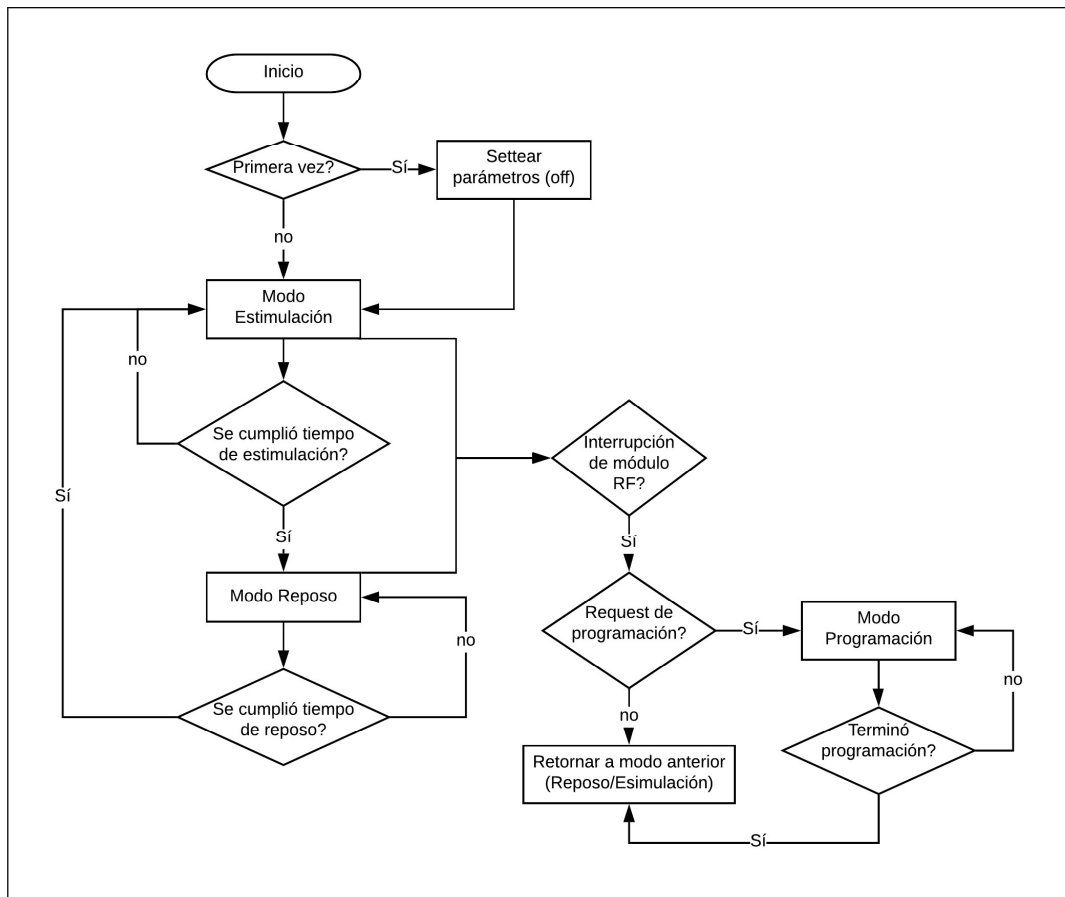


Figura 12: Primer acercamiento basado en los tres modos de operación para replicar el funcionamiento del estimulador

En el modo de estimulación se generará la señal de estimulación, en donde se controlarán

los parámetros de ancho de pulso, frecuencia de estimulación y tiempo de estimulación. Para la realización de este modo se necesita calibrar los parámetros para hacerlos lo más cercano a los teóricos o dentro del rango del modelo 102 de LivaNova.

En el Modo de Reposo se replicará el parámetro de tiempo de reposo, además, este tiempo se optimizará para utilizar la menor cantidad de energía posible, utilizando modos de ahorro de energía. Este tiempo de reposo será calibrado de igual manera para ser lo más cercano al teórico o dentro del rango del modelo 102.

Por último, en el modo de programación se comunicará el MCU con el módulo RF en donde se recibirán los nuevos parámetros a configurar. Además de configurar los nuevos parámetros, el MCU se encargará de enviar los parámetros actualmente configurados así como datos de pertinentes al médico, como fecha de la implantación, ID del paciente, la impedancia calculada por el módulo de potencia, entre otros.

7.3. Selección de la circuitería necesaria para replicar funcionamiento del estimulador

Lo primero por hacer es seleccionar la unidad lógica y control mostrada en la Figura 3, la cual claramente es un MCU. El MCU a seleccionar tiene que cumplir con las siguientes características mínimas:

- Diseño de baja potencia
- Modos de ahorro de energía
- Varios osciladores internos
- Como mínimo 2 módulos independientes de PWM
- Como mínimo 3 temporizadores
- Por lo menos 1 periférico SPI
- Por lo menos 1 periférico I2C
- Memoria no volátil, EEPROM

Para la selección del microcontrolador se presentarán varios MCUs que cumplan con los requisitos mínimos mencionados. Luego, se utilizará cierta métrica con la cual se evaluarán características claves del microcontrolador. En esta métrica se le dará más peso a MCUs de la marca Microchip pues ya se tiene conocimiento previo de estos, así como a los que menor consumo energético tengan.

Por último, en Guatemala es difícil encontrar gran variedad en cuanto a componentes electrónicos, en especial cuando se necesitan modelos tan específicos, ya que generalmente las electrónicas del país solo trabajan con los modelos más comerciales.

En el Cuadro 7 se observa una comparación de 6 microcontroladores distintos y sus características más relevantes que podrían ser útiles para la elaboración de prototipos.

	PIC16F1789	PIC16F1783	PIC18F46K22	PIC18F47J13	PIC18F27J13	MSP430FR2476
Fabricante	Microchip					TI
Program Memory	16384	4096	32768	65536	65536	65536
EEPROM	256	256	1024	0	0	0
SRAM	2048	512	3896	3760	3760	8192
I/Os	36	25	36	34	23	43
Timers 8-bit	2	2	3	4	4	0
Timers 16-bit	1	1	4	4	4	5
10-bit ADC	0	0	2	1	1	0
12-bit ADC	1	1	0	1	1	1
5-bit DAC	3	0	1	0	0	1
8-bit DAC	1	1	0	0	0	0
CCP/ECCP	3	2	4	10	10	10
EUSART	1	1	2	2	2	2
MSSP (I2C/SPI)	1	1	2	2	2	2
Empaquetado PDIP	Y	Y	Y	N	Y	N
Sleep Current	50 nA	50 nA	20 nA	9 nA	9 nA	37 nA
PMD	N	N	N	Y	Y	N

Cuadro 7: Características relevantes de microcontroladores

Luego de realizar la comparación entre los 6 MCU distintos, podemos proceder a evaluarlos para así decidir cual de estos es el elegido para la aplicación. Para realizar la evaluación que se muestra en el Cuadro 8 se dio puntaje de 0 a 10 en las categorías más relevantes. Luego, se obtuvo el promedio de la suma de cada una de estas categorías.

Hay categorías que son más importantes que otras, como lo son la del fabricante, el empaquetado PDIP y el módulo de PMD. Se le dio bastante importancia a que el MCU tenga empaquetado PDIP pues reduce el tiempo de la elaboración de prototipos considerablemente, ya que se puede utilizar placa de pruebas para los prototipos iniciales. Posteriormente, ya cuando se tenga el diseño o prototipo final, se fabrica la placa impresa con la versión SMD del microcontrolador. Por esta razón, si el modelo del MCU tiene empaquetado PDIP, se dio una puntuación de 10, si no de 0.

De igual manera, se considera que la característica de PMD es clave, pues solamente los periféricos deseados consumirán energía, por esto, si posee este módulo se dio una calificación de 10, y si no, de 2. Esto va combinado con la el puntaje de sleep current, y se evaluó con un 10 al de menor corriente, y se fue reduciendo la calificación conforme más corriente consumía el MCU.

Por último, la categoría de fabricante se considera igual de importante, pues debido a los conocimientos previos del curso de Microcontroladores Aplicados a la Industria, se trabajó con los MCU de Microchip. Por este motivo, ya se conoce bastante bien la plataforma de desarrollo, sus dispositivos, etc. Si el fabricante del microcontrolador es Microchip se dio un 10, si no un 5, pues las otras plataformas de desarrollo son igual de capaces que la

ofrecida por Microchip.

	PIC16F1789	PIC16F1783	PIC18F46K22	PIC18F47J13	PIC18F27J13	MSP430FR2476
Fabricante	10	10	10	10	10	5
Memoria no volátil	10	10	10	6	6	6
PMD	2	2	2	10	10	2
Sleep Current	4	4	8	10	10	6
Timers	5	5	8	10	10	7
I/Os	10	7	10	10	8	10
SPI	7	7	10	10	10	10
PDIP	10	10	10	0	10	0
Total	7.3	6.9	8.5	8.3	9.3	5.8

Cuadro 8: Evaluación de microcontroladores

Con los resultados del Cuadro 8 se observa que el MCU a elegir es el PIC18F27J13. Este tiene la ventaja de tener un empaquetado PDIP así como es el microcontrolador que menor corriente de sleep consume. Luego de este, se considera el PIC18F46K22 y el PIC16F1789. El PIC18F47J13 no se toma en cuenta pues es de la misma familia que el PIC18F27J13, con la diferencia de no ofrecer empaquetado PDIP.

El PIC18F46K22 es el modelo ideal para los primeros prototipos, pues ofrece memoria EEPROM además de tener más I/Os que el 27J13, pues estos pines extras facilitan su capacidad de depurar el software. En último caso, se podría usar el PIC16F1789 en caso de existir algún problema con el K22.

Los modelos anteriormente mencionados son buenos para realizar los primeros prototipos, pero el **PIC18F27J13** es el que debe usarse para hacer el diseño final.



Figura 13: Microcontrolador PIC18F46K22 en empaquetado DIP40



Figura 14: Microcontrolador PIC18F27J13 en empaquetado DIP28

En el prototipo 1 se utilizará el microcontrolador PIC18F46K22, esto debido a la facilidad de conseguirse en Guatemala además de tener empaquetado PDIP, el cual es compatible para la utilización en placas de pruebas o Protoboard. Además, el PIC18F46K22 posee más I/Os que el PIC18F27J13, por lo cual es más útil para debuggear en prototipos iniciales. El objetivo de este prototipo es generar la señal de estimulación y cumplir con parámetros que se establecen en el Cuadro 4.

8.1. Reproducción de la señal de estimulación

Debido a la importancia del bajo consumo, se utilizará el oscilador interno con la frecuencia más baja que tiene el PIC18F46K22, la cual es de 31 kHz. Según esta frecuencia de oscilación se calcularán los valores respectivos para reproducir la señal de estimulación.

8.1.1. Frecuencia y ancho de pulso

Para reproducir la señal de estimulación se utilizará el módulo de PWM del microcontrolador, pues la modulación por ancho de pulso modula la cantidad de energía con la que se estimulará el nervio vago.

Según la Frecuencia de oscilación (FOSC) seleccionada, se calculan los valores necesarios del módulo PWM para generar la frecuencia de estimulación; la ecuación 3 es la que se utiliza para este cálculo. El módulo PWM utiliza como base un temporizador que tiene tres posibles pre-escaladores. Los posibles valores del pre-escalador son 1, 4 y 16. Debido a que la frecuencia más baja que se debe generar es de 1 Hz, se selecciona el pre-escalador (*TMRx Prescale Value*) con un valor de 16. El módulo ECCP junto con un temporizador generan la señal PWM.

$$FREQ_{PWM} = \frac{FOSC}{4 \cdot (PRx + 1) \cdot (TMRxPrescaler)} \quad (3)$$

Al utilizar la ecuación 3 se generó el Cuadro 9, en donde se puede observar que para poder realizar la frecuencia de 1 Hz se necesitan 9 bits, por lo cual esta es la única frecuencia que no se puede realizar con este módulo, ya que el registro PRx es solamente de 8 bits.

Frecuencia (Hz)	Valor de PRx	Bits necesarios
1	483	9
2	241	8
5	96	7
10	47	6
15	31	5
20	23	5
25	18	4
30	15	4

Cuadro 9: Parámetros calculados de PRx para el cálculo de la frecuencia del módulo de PWM

Luego de calcular los valores necesarios para cada una de las frecuencias de estimulación se procede a calcular los valores necesarios para generar el ancho de pulso del PWM utilizando la ecuación 4. La ventaja de utilizar el módulo PWM es que el ancho de pulso se controla con un valor de 10 bits sin la necesidad de utilizar otro timer.

En la ecuación 4 el parámetro $CCPRxL : CCPxCON < 5 : 4 >$ es el valor de 10 bits que se modifica para variar el ciclo de trabajo del PWM. El valor de $TMRx Prescale Value$ es el mismo que fue seleccionado previamente, 16.

$$Ancho\ de\ Pulso = \frac{(CCPRxL : CCPxCON < 5 : 4 >) \cdot (TMRx\ Prescale\ Value)}{FOSC} \quad (4)$$

El Cuadro 10 se generó utilizando la ecuación 4, en donde se redondean los valores del parámetro $CCPRxL : CCPxCON < 5 : 4 >$ al entero más cercano. Se puede observar que no existe valor tal de $CCPRxL : CCPxCON < 5 : 4 >$ que logre generar un ancho de pulso de $130\mu s$, $250\mu s$ y de $750\mu s$.

Ancho de pulso(μs)	CCPRxL:CCPxCON<5:4>
130	0
250	0
500	1
750	1
1000	2

Cuadro 10: Parámetros calculados de $CCPRxL : CCPxCON < 5 : 4 >$ para el cálculo del ancho de pulso del PWM

Para evaluar la realización del PWM se procedió a medir los resultados en un osciloscopio, utilizando frecuencias 2, 5, 10, 15, 20 25 y 30 Hz. Además, se tiene que cumplir con los valores que se muestran en el Cuadro 4 y brindar un porcentaje de error similar o menor, el cual es de un $\pm 6\%$.

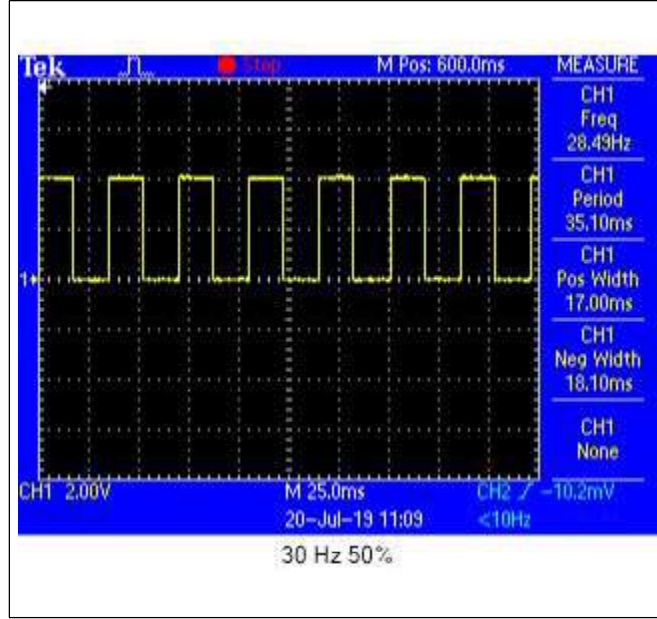


Figura 15: PWM de 30 Hz y ciclo de trabajo de 50 %

Como se puede observar en la Figura 15, el PWM generado tiene una frecuencia de 28.49, con lo cual el porcentaje de error es menor al 6 % presentado por el modelo 102. Se utilizó un ciclo de trabajo de 50 % por facilidad.

8.1.2. Tiempo de estimulación

Luego de generar la señal de estimulación (Frecuencia y Ancho de Pulso) se procedió a limitar ésta solamente durante el periodo mostrado en el Cuadro 4 como tiempo de estimulación. Para realizar esto se utilizó un módulo de timer. El módulo seleccionado del MCU es el Timer 0 que es un timer de 8 o 16 bits. Primero se seleccionó la fuente de reloj del timer, la cual fue FOSC, que tiene un valor de 31kHz.

Luego se selecciona la cantidad de tiempo que contará este módulo, replicando así el tiempo de estimulación. Al observar los parámetros de tiempo de estimulación posibles, se eligió que el Timer 0 cuente en periodos de 1 segundo, así repetir este periodo n veces hasta llegar al valor configurado. Para calcular los valores necesarios para cargar en los registros de configuración del módulo, se utiliza la ecuación 5. En la ecuación se selecciona el valor del pre-escalador, PRE , el cual es de 8 bits, lo que quiere decir que puede dividir la frecuencia de entrada 2^8 veces; cualquier valor de PRE mayor o igual a 32 puede hacer que el Timer 0 cuente 1 segundo.

Luego de realizar esto, calculamos el valor de $TMR0$ con el cual hacemos que T_{out} se igual a 1 segundo. Seleccionando PRE igual a 64, el valor más cercano de $TMR0$ para hacer el tiempo de un segundo es de 134. Con los valores anteriores seleccionados se calculó que el módulo Timer 0 cuente intervalos de 0.999 segundos.

$$T_{out} = \frac{4 \cdot PRE \cdot (256 - TMR0)}{FOSC} \quad (5)$$

8.1.3. Tiempo de reposo de estimulación

Debido a la forma de la terapia de estimulación vagal, el estimulador no realiza ninguna función especial durante buena parte de tiempo. Este tiempo en donde las funciones son nulas, se puede aprovechar para que el MCU entre en modo de ahorro de energía.

El PIC18F46K22 al entrar en modo Sleep guarda el estado actual de los periféricos y la memoria RAM del dispositivo, pero deteniendo la ejecución del código. Para salir de este modo existen varias maneras, como por ejemplo, al expirar el contador del WDT o cuando se activa alguna otra interrupción. Se podría utilizar un módulo de reloj externo o un RTCC pero debido a que este consumiría más potencia se rechazó esta opción. Además, como el tiempo de reposo de la estimulación es programable por el médico, se decidió utilizar el WDT para que este saque al MCU del modo Sleep.

$$T_{Sleep} = 4ms \cdot (PRE) \quad (6)$$

$$T_{Sleep} = 4ms \cdot 2^{11} \simeq 8.2s \quad (7)$$

La manera en la que funciona el WDT es que tiene periodos fijos de expiración, los cuales son programados utilizando un pre-escalador de 16 bits. En las ecuaciones 6 y 7 se muestra el procedimiento utilizado para el cálculo del tiempo de vencimiento del WDT. El tiempo seleccionado es de 8.2 segundos aproximadamente; para contar tiempos mayores, se repetirá este procedimiento n veces, como se muestra en la Figura 16 hasta cumplir con los tiempos de reposo de estimulación.

En el diagrama de flujo, la variable cantidad será la que se modificará cuando el médico programe el tiempo de reposo según los valores del Cuadro 4. El valor de la variable cantidad representa los tiempos mostrados en el Cuadro 11. Los valores para los tiempos de 5 a 60 son múltiplos del valor de la variable cantidad que genera el tiempo de reposo de 5 minutos; de igual manera con los de 60 a 180 minutos.

Tiempo de reposo	Cantidad
0.2	1
0.3	2
0.5	4
0.8	6
1.1	8
1.8	13
3	22
5	37
5-60	37-444
60-180	444-1332

Cuadro 11: Valores mapeados de la variable cantidad al tiempo de reposo real

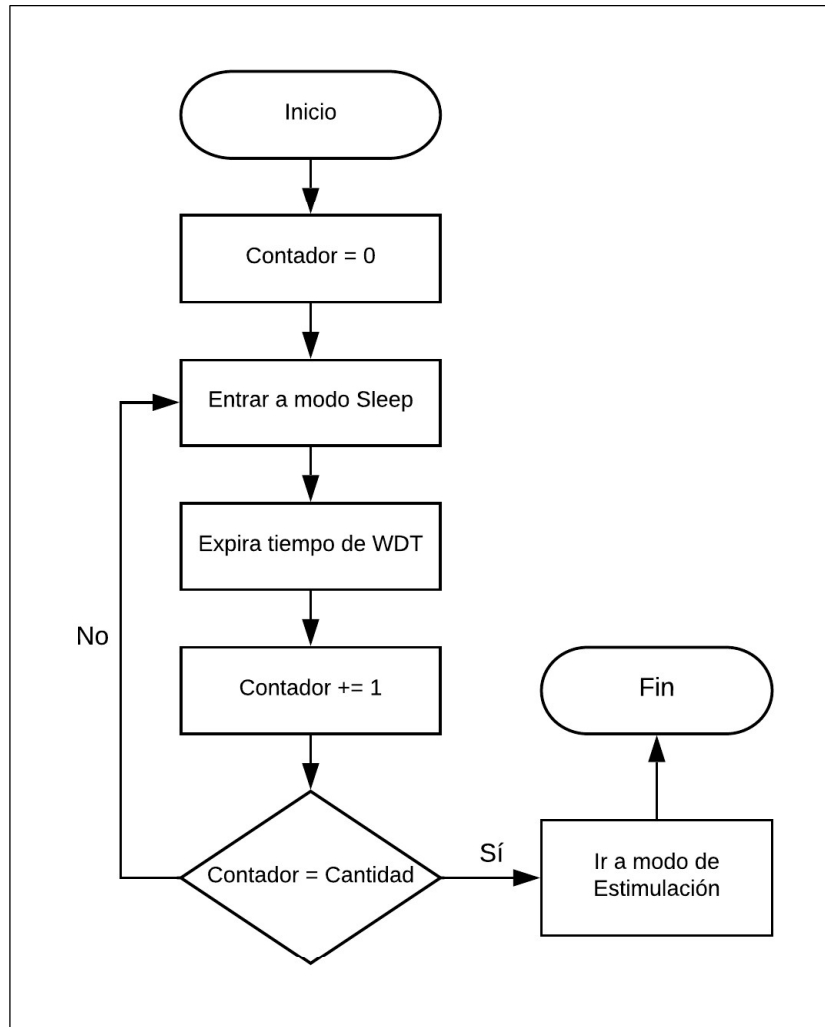


Figura 16: Diagrama de flujo para reproducir los tiempos de reposo

8.2. Configuración del MCU para prototipo 1

La implementación del código que realiza lo anteriormente descrito se realizó utilizando el plugin MCC de MPLAB X IDE. Esta herramienta permite una selección fácil de los módulos que se utilizarán así como la configuración de los pines de entrada y de salida. Las configuraciones realizadas son las siguientes:

1. WDT
 - a) Pre-escalador: 2048
 - b) Configurable en software
2. Timer 0
 - a) Pre-escalador: 16
 - b) Frecuencia de oscilación: 31 kHz
 - c) Modo de operación: Temporizador

3. Timer 2

a) Pre-escalador: 16

4. ECCP 1

a) Modo: PWM

b) Utilizar Timer 2

5. Puertos de entrada/salida

a) RD0: Salida de PWM

b) RD1: Salida indica tiempo de estimulación

Luego de realizar la configuración en MCC se generaron las funciones estándar para interactuar con los módulos seleccionados, como cambio de frecuencia, cambio de periodo del timer, iniciar timer, etc. Con las funciones generadas por MCC, se procedió a implementar y darle forma al código con tal de realizar el ciclo en el que operará el generador. Para esto se utilizaron banderas o variables para indicar en que modo se encuentra. Se identificaron dos modos de operación por el momento, modo de estimulación y modo de reposo. Dentro del modo de estimulación se activa el Timer 0 que es el encargado de controlar el tiempo de estimulación. Luego de esto se activa el timer vinculado al módulo ECCP y se habilita la salida de la señal PWM.

Para evitar el polleo, se configuraron las interrupciones para el temporizador del tiempo de estimulación. Luego de haberse cumplido el tiempo de estimulación, se apagan los temporizadores y la salida del módulo PWM, también se apaga la salida que indica que el estimulador se encuentra en modo de estimulación y posteriormente se entra al modo de reposo. En el modo de reposo se ejecuta el procedimiento mostrado en la Figura 16. Luego de haberse cumplido el tiempo, se vuelve a estimular.

8.3. Resultados del prototipo 1

Para la toma de resultados, se evaluaron todos los casos de cada una de las variables que se pueden programar. Además, se tomaron cinco mediciones para cada parámetro para sacar un promedio del valor el cual es reportado en las secciones siguientes.

8.3.1. Resultados del ancho de pulso de la señal

Para la medición del ancho de pulso se midió la salida del PWM y se midió el tiempo en alto de la señal como se puede observar en las Figuras 17 y 18. Para ello se midió la salida del PWM en el pin RD0.

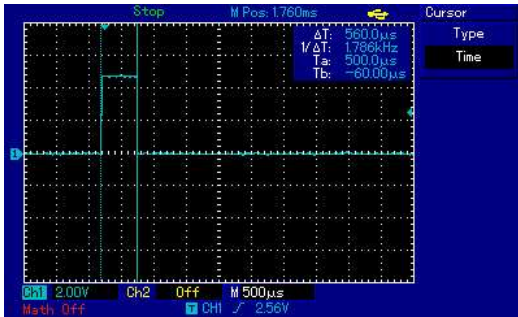


Figura 17: Medición de ancho de pulso de $500 \mu s$

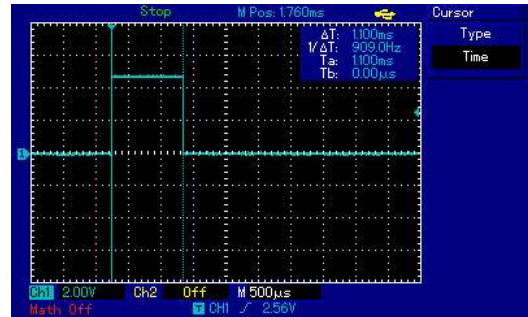


Figura 18: Medición de ancho de pulso de $1000 \mu s$

Ancho de pulso (μs)			
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?
130	—	—	No
250	—	—	No
500	558.5	11.7 %	Sí
750	558.5 ^a	25.3 %	No
1000	1102.2	10.22 %	Sí

Nota: — indica que el valor no se logró replicar.

^a Se reportó el mismo valor que para $500 \mu s$ ya que es el más cercano que se pudo generar.

Cuadro 12: Ancho de pulso teórico y generado en prototipo 1

8.3.2. Resultados de la frecuencia de la señal

Para realizar la medición de la frecuencia de la señal se midió el tiempo entre cada pulso en alto de la señal del glspwm en el pin RD0. En el osciloscopio se puede mostrar el periodo o la frecuencia que de un intervalo de tiempo. En las Figuras 19 a 22 se muestra como se realizaron las mediciones para cuatro frecuencias distintas.

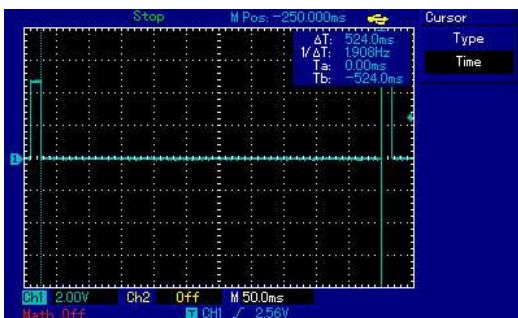


Figura 19: Medición de la frecuencia de 2 Hz

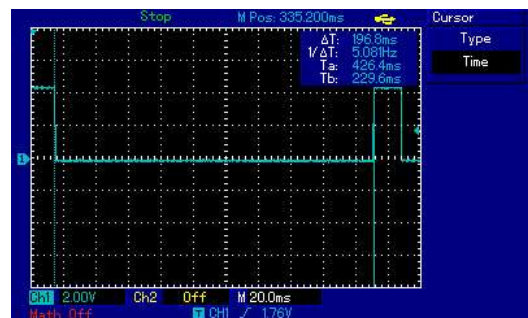


Figura 20: Medición de la frecuencia de 5 Hz

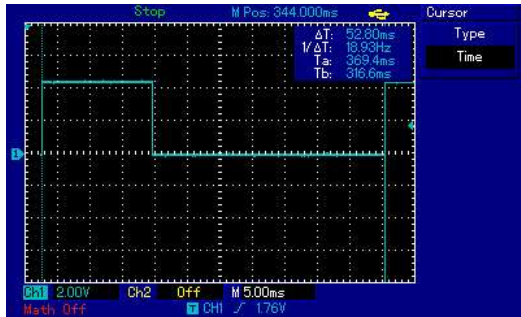


Figura 21: Medición de la frecuencia de 20 Hz

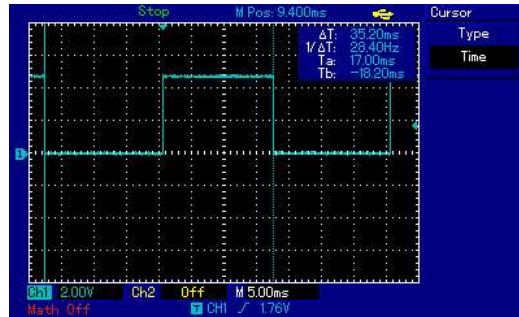


Figura 22: Medición de la frecuencia de 30 Hz

Frecuencia de estimulación (Hz)			
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?
1	—	—	No
2	1.92	4.00 %	Sí
5	5.11	2.20 %	Sí
10	10.09	0.91 %	Sí
15	15.14	0.91 %	Sí
20	18.98	5.10 %	Sí
25	24.49	2.04 %	Sí
30	28.63	4.57 %	Sí

Nota: — indica que el valor no se logró replicar.

Cuadro 13: Frecuencia de estimulación teórica y generada en prototipo 1

8.3.3. Resultados del tiempo de estimulación

Para la medición de los tiempos de estimulación se midió el puerto RD1, que está en alto (VDD) cuando se encuentra en modo estimulación, y bajo (VSS) cuando es modo de reposo. En las Figuras 23 a la 26 se muestran ejemplos de como se realizaron las medidas en donde se muestran cuatro tiempos de estimulación distintos.

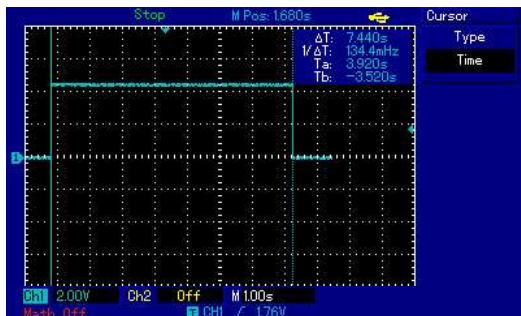


Figura 23: Medición de tiempo de estimulación de 7 s

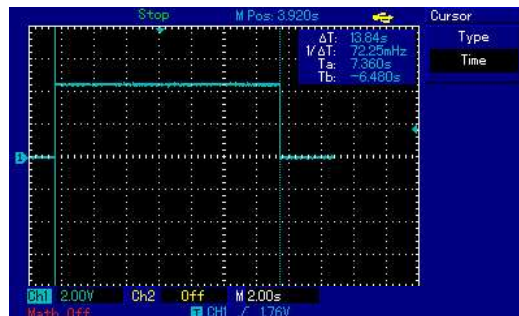


Figura 24: Medición de tiempo de estimulación de 14 s

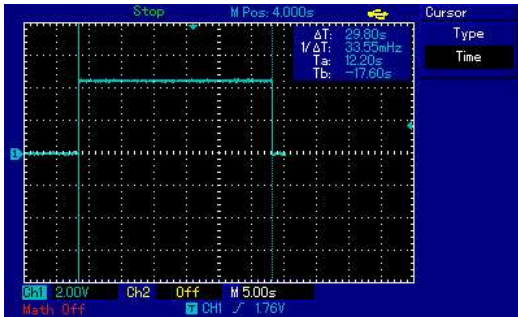


Figura 25: Medición de tiempo de estimulación de 30 s

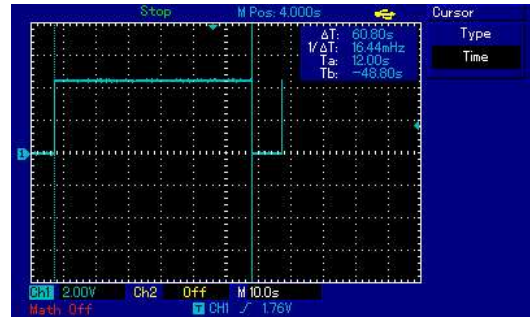


Figura 26: Medición de tiempo de estimulación de 60 s

Tiempo de estimulación (s)			
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?
7	7.43	6.14 %	Sí
14	13.86	1.00 %	Sí
21	21.13	0.62 %	Sí
30	29.81	0.63 %	Sí
60	60.73	1.22 %	Sí

Cuadro 14: Tiempo de estimulación teórico y generado en prototipo 1

8.3.4. Resultados del tiempo de reposo

Para los tiempos de reposo se midió el tiempo bajo (VSS) en el pin RD1, que es cuando el MCU está en modo reposo. Se muestran 4 mediciones distintas de tiempos de reposo mostradas en las Figuras 27 a la 30.

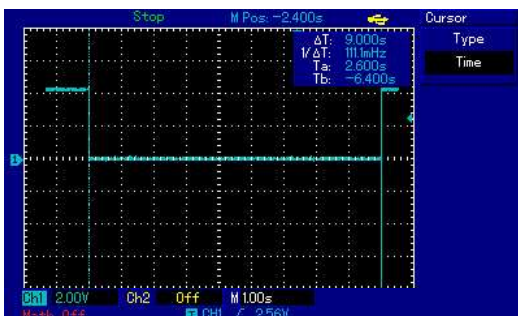


Figura 27: Medición para el tiempo de reposo de 0.2 min

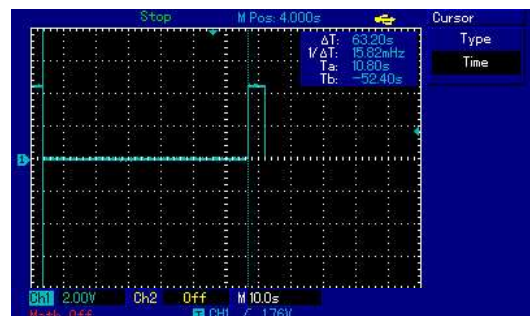


Figura 28: Medición para el tiempo de reposo de 1.1 min

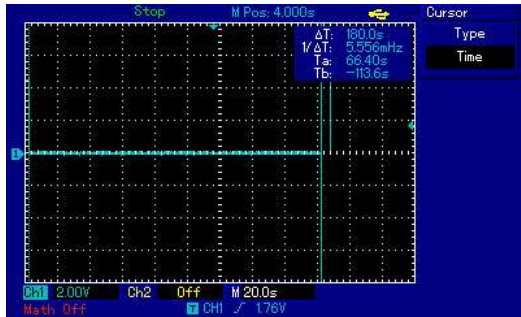


Figura 29: Medición para el tiempo de reposo de 3 min

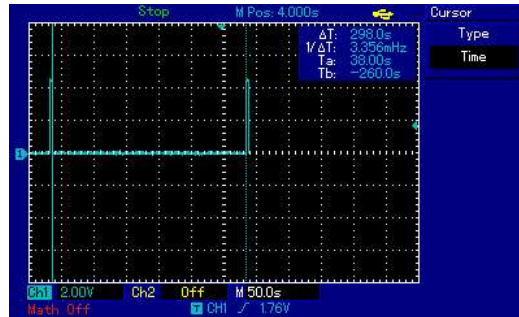


Figura 30: Medición para el tiempo de reposo de 5 min

Tiempo de reposo (min)			
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?
0.2	0.15	25.00 %	No
0.3	0.30	0.56 %	Sí
0.5	0.45	9.33 %	Sí
0.8	0.75	5.83 %	Sí
1.1	1.05	4.24 %	Sí
1.8	1.81	0.28 %	Sí
3	3.04	1.28 %	Sí
5	4.98	0.45 %	Sí
5-60	—	—	No
60-180	—	—	No

Nota: — indica que el valor no se logró medir.

Cuadro 15: Tiempo de reposo teórico y generado en prototipo 1

8.4. Evaluación de resultados

Luego de realizar las pruebas para cada uno de los parámetros de la señal de estimulación se evalúan según los logros obtenidos; en el Cuadro 16 se muestra el porcentaje de logro para cada parámetro. De todos los parámetros, solamente se logró cumplir completamente con el tiempo de estimulación. Para los demás parámetros configurables, a excepción del ancho de pulso, se logró cumplir con la mayoría de los parámetros.

Parámetro	Porcentaje de replicación	¿Se cumplió?
Ancho de pulso	50 %	No
Frecuencia de estimulación	87.5 %	No
Tiempo de estimulación	100 %	Sí
Tiempo de reposo	70 %	No

Cuadro 16: Evaluación de logros prototipo 1

8.5. Retos y dificultades

Por ser primera vez que se trabajó con la función de sleep, se realizó una lectura extensiva del manual del PIC18F46K22. Además, el microcontrolador usado tenía cierta falla, por lo cual, al alimentar el dispositivo este parecía no funcionar. Después de varios intentos, se descubrió que por alguna extraña razón había que colocar un cable en el puerto RA7 de éste. Debido a que este MCU era el único que se tenía se tuvo que proceder con este problema.

Paralelamente, se procedió a encargar el PIC16F1789, pues en conjunto con Andrés Girón, encargado del módulo RF, se decidió que éste se usaría como alternativa en caso de que el K22 fallara. El PIC16F1789 fue encargado con tiempo de anticipación ya que al no haber en Guatemala, se mandó a traer a los Estados Unidos, y su tiempo de envío es de aproximadamente dos semanas. Se optó por éste ya que este MCU es compatible con el plugin MCC.

Luego, al realizar las mediciones de los parámetros de la señal de estimulación se encontró con la dificultad que cada vez que se deseaba cambiar algún parámetro había que recalcular éste, por lo cual el trabajo se hizo bastante extenso.

Con los parámetros de tiempo de reposo, no se pudieron medir tiempos mayores a 5 minutos (10 a 180 minutos) debido a que el osciloscopio tiene la limitante de solamente poder mostrar aproximadamente señales de hasta 580 segundos. Por esta razón solamente se calibraron los tiempos hasta 5 minutos.

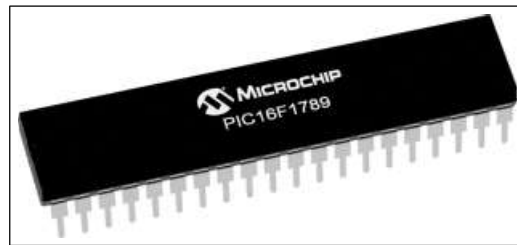


Figura 31: PIC16F1789 en empaquetado PDIP40

Para la realización del prototipo 2 se utilizará el mismo MCU (PIC 18F46K22). A pesar de la falla encontrada en el prototipo 1, este modelo es el más similar al PIC18F27J13. En este prototipo se modificarán los aspectos necesarios para lograr cumplir con todos los parámetros de la señal de estimulación y a la vez se optimizará el código para que cumpla con el estándar observado en la sección 6.6, de esta manera reducir la posible cantidad de errores y poder depurar de manera más fácil del programa. Además, este estándar es de gran importancia pues en el prototipo posterior se unirá el código que se encarga de la comunicación con el módulo RF.

9.1. Modificaciones a realizar al prototipo 1

La modificación a realizar al prototipo 1 es cumplir con todos los parámetros que se muestran en el Cuadro 4. Además de esto, simplificar la estructura del código para poder realizar un fácil cambio de parámetros, previendo el cambio que se tendrá que realizar posteriormente a la hora de conectarse con el módulo RF para poder modificar los parámetros de manera inalámbrica.

Además, se tiene que adaptar el código en los siguientes aspectos: declaración de variables, utilización de los archivos headers y normas de estilo. Estos cambios se realizan para cumplir con lo estipulado por el estándar BARR-C.

9.1.1. Modificación al ancho de pulso y frecuencia de estimulación

Debido a que los los parámetros de ancho de pulso y el de frecuencia de estimulación fueron los de menor logro, como se observa en el Cuadro 16. Por esta razón, se plantean tres nuevos acercamientos con el objetivo de cumplir estos objetivos, los cuales se explican a continuación.

Modificación a la frecuencia de oscilación

Al aumentar la frecuencia de oscilación del dispositivo, con el módulo PWM podemos obtener anchos de pulsos más pequeños, pero como contra parte, no podemos generar fre-

cuencias más pequeñas, por ende, no podríamos generar la frecuencia de de 1 Hz requerida. Hay que resaltar, que a pesar de aumentar la frecuencia de oscilación, con el módulo ECCP en modoPWM solamente se pueden realizar 1024 anchos de pulso distintos de nuestra frecuencia de oscilación mínima, es decir, el ciclo de trabajo mínimo que se puede realizar está dado por:

$$\frac{100}{1024} \% = 0.098 \%$$

Con lo observación anterior, no se puede cumplir con el ciclo de trabajo más pequeño dado según los parámetros para la terapia VNS, ya que:

$$100 \% \frac{PW_{min}}{1} = 100 \% \frac{130 \times 10^{-6} s}{1s} = 0.013 \% < 0.098 \% \%$$

Además, el aumento en la frecuencia de oscilación del sistema conllevaría un mayor uso de potencia, como se puede observar en la ecuación 1.

Utilizar un nuevo MCU

Analizando las limitantes del módulo de PWM del PIC 18F46K22 y de su módulo PWM se podría utilizar un nuevo MCU que tenga un PWM con una mayor resolución. La resolución mínima que el módulo PWM tendría que tener está dada por la siguiente ecuación, en donde R es la resolución en bits del módulo.

$$R = \log_2 \left(\frac{1.0000 s}{130 \times 10^{-6} s} \right) = 12.91bits$$

Con lo demostrado anteriormente se debe de buscar un MCU que tenga un módulo PWM con resolución mínima de 13 bits. Esto implicaría que el PIC 18F27J13 ya no se podría usar, ya que este es el microcontrolador objetivo para el prototipo final. El modelo PIC16F1789 posee un módulo de PWM de 16 bits pero el problema de cambiar a este es que no se podría portar el código a otro con menor consumo de potencia. Se buscó un modelo similar al PIC18F27J13 que tenga características similares de bajo consumo de potencia y se encontró que la alternativa sería usar un MCU de la familia PIC18FXXQ43. El problema de esta familia es que aún se encuentra en su etapa de diseño, por lo cual se descarta esta opción.

A pesar de esto, si deseamos cambiar el modelo para el prototipo final para que sea el PIC16F1789 estaríamos sacrificando vida útil del dispositivo, pues se considera una prioridad el ahorro de potencia.

Dejar de utilizar el módulo PWM

Luego de analizar las dos opciones relacionadas al uso de un módulo PWM, se procedió a pensar una alternativa que pudiera cumplir con los parámetros de estimulación y a la vez que fuera portable al PIC18F27J13.

El periodo entre instrucción e instrucción está dado por: $\frac{FOSC}{4}$, por lo que el tiempo entre instrucciones es de aproximadamente $129 \times 10^{-6} s$. Este tiempo se encuentra dentro del porcentaje de error del Ancho de Pulso que se muestra en el Cuadro 4. Debido a lo cercano, si prendiese y apagase el puerto, se reproduciría el ancho de pulso requerido. Tomando esto como base, y observando que los tiempos de ancho de pulso son múltiplos de este, se puede agregar una instrucción entre prender y apagar el tiempo, así aumentando el tiempo en un

factor de dos. Con esta solución se lograría realizar los valores de ancho de pulso sin cambiar el microcontrolador ni aumentar la frecuencia de oscilación del sistema.

Ya resuelto el problema del ancho de pulso, se procede a replicar la frecuencia de estimulación. Para esto se utiliza un timer del MCU para contar el periodo de la frecuencia respectiva. La configuración del timer a realizar se hará de manera similar a la hecha en la sección 8.1.2. A diferencia de la configuración anterior, aquí se modificará el periodo de interrupción dependiendo de la frecuencia seleccionada. Otra ventaja de reproducir la frecuencia de estimulación de esta manera es que podemos realizar la frecuencia de 1 Hz que no fue lograda en el prototipo 1, como se observa en el Cuadro 13.

Luego de explorar las tres alternativas planteadas, se elige dejar de utilizar el módulo de PWM y realizar el ancho de pulso y frecuencia de estimulación de manera manual. Para la frecuencia de estimulación se utilizará el Timer 1 y para el ancho de pulso se aprovechará el periodo de instrucción de aproximadamente $129\mu s$.

9.1.2. Calibración de los tiempos de estimulación

Debido a que el oscilador interno puede variar, y debido a que se utilizan interrupciones para la realización del tiempo de estimulación, se deben calibrar los tiempos obtenidos en el prototipo 1 así reducir el porcentaje de error. Para eso se calcularon los tiempos utilizando la ecuación 5 y luego fueron modificados de tal manera de reducir el error de este, ya que los tiempos teóricos cambian debido al tiempo entre la interrupción y la acción que se ejecuta debido a esta.

9.1.3. Modificación a los tiempos de reposo

Para los tiempos de reposo, como se observa en el Cuadro 15, no se cumplió al cien por ciento, pues no se logró replicar el parámetro de 0.2 minutos. Con el objetivo de reproducir este parámetro se procedió a modificar el pre-escalador del WDT de tal manera que ahora el periodo de expiración de este fuera de aproximadamente 4.1 segundos, previamente siendo alrededor de 9 segundos tomando el tiempo de reactivación del MCU. Debido a este cambio, se modificaron los valores de la variable contador del Cuadro 11. Como se observó en la ecuación 7, ahora se utilizará un pre-escalador con un valor de $2^{10} = 1024$ produciendo un tiempo teórico de 4.096 s.

9.1.4. Optimización y mejoras a la estructura del código

Para el prototipo 1, se obvió la utilización del estándar BARR-C, por lo cual, todas las variables utilizadas y funciones que fueron utilizadas se tienen que redefinir utilizando este estándar. También se tienen que aplicar las reglas de sangría y de llaves ($\{, \}$) de los ciclos for, while, switch, etc con el objetivo de reducir la cantidad de bugs.

Además de los cambios estéticos se simplificó el funcionamiento con funciones que seleccionen los parámetros dentro de valores finitos para evitar la estimulación con algún parámetro fuera del rango de la terapia VNS. Con las funciones creadas, se prepara la base para cuando haya que modificar los parámetros a través de los parámetros recibidos por medio del módulo RF. En la Figura 32 se muestra un ejemplo de como se realizó la función de codificación para el parámetro de tiempo de estimulación.

Ya que se codificaron los casos para cada uno de los parámetros, en los Cuadros 17 al 20 se muestran los casos y el valor del parámetro asociado. En el Cuadro 17 se muestran como

se codificó el valor a cargar en el registro del TRM1. Este valor representa cierta frecuencia de estimulación. Estos valores fueron calibrados con tal de obtener un menor porcentaje de error al obtenido en el prototipo 1.

```
void sel_time_stim(uint8_t time_var)
{
    switch(time_var)
    {
        // 7 s
        case 1:
            cantidad_on = 7;
            break;

        // 14 s Hz
        case 2:
            cantidad_on = 14;
            break;

        // 21 s
        case 3:
            cantidad_on = 21;
            break;

        // 30 s
        case 4:
            cantidad_on = 30;
            break;

        // 60 s
        case 5:
            cantidad_on = 60;
            break;

        default:
            cantidad_on = 30;
            break;
    }
}
```

Figura 32: Función que codifica los tiempos de estimulación

En el Cuadro 18 se muestran los casos codificados para el tiempo de estimulación. Debido a que el tiempo de estimulación se basa en repetir n veces 1 segundo, solamente se calibró el tiempo del temporizador para que este se acercara lo más posible a 1 segundo, así lograr replicar los tiempos de estimulación estipulados al repetir este tiempo las veces adecuadas para cada valor.

Caso	Frecuencia (Hz)	TMR1
0	1	0x8CF0
1	2	0xC711
2	5	0xE976
3	10	0xF4FC
4	15	0xF8F8
5	20	0xFABA
6	25	0xFBFF
7	30	0xFC9A
Default	1	0x8CF0

Cuadro 17: Casos para la frecuencia de estimulación y la variable TMR1 codificada

Para los valores del tiempo de reposo, se recalcularon utilizando el nuevo periodo del WDT. Estos valores para la variable cantidad se muestran en el Cuadro 11. Los tiempos mayores a 5 minutos se estimaron basándose en un tiempo de referencia de 5 minutos.

Caso	Tiempo de estimulación (s)
0	7
1	14
2	21
3	30
4	60
Default	30

Cuadro 18: Casos para el tiempo de estimulación

Para los valores del ancho de pulso se realizó una codificación similar los mostrados anteriormente, pero con la diferencia que en el caso del switch se ejecuta la acción de prender y apagar el puerto de la señal. Para variar el ancho de pulso se agregaron instrucciones NOP, las cuales solamente agregan tiempo entre el encendido y apagado del puerto. Por ejemplo, para el ancho de pulso de $1000\mu s$, se agregaron siete instrucciones NOP las cuales consumen aproximadamente $910\mu s$, por lo cual, contando el tiempo de la siguiente instrucción, se calcula un valor teórico de $1040\mu s$, que se encuentra dentro del rango de porcentaje de error del modelo 102 de LivaNova.

Caso	Ancho de pulso (μs)
0	130
1	250
2	500
3	750
4	1000
Default	0

Cuadro 19: Casos para el ancho de pulso

Caso	Tiempo de reposo (min)	Cantidad sleep
0	0.2	3
1	0.3	4
2	0.5	7
3	0.8	11
4	1.1	15
5	1.8	24
6	3	40
7	5	67
8	10	134
9	15	201
10	20	268
11	25	335
12	30	402
13	35	469
14	40	536
15	45	603
16	50	670
17	55	737
18	60	804
19	90	1206
20	120	1608
21	150	2010
22	180	2412
Default	5	67

Cuadro 20: Casos para el tiempo de reposo y variable cantidad sleep

9.2. Resultados de la reproducción de los parámetros de estimulación

Para la toma de resultados, se evaluaron todos los casos de cada una de las variables que se pueden programar, de igual manera que se realizó con el prototipo 1. Antes de realizar las muestras, utilizando el osciloscopio se calibraron las variables con las cuales se producen los parámetros con el objetivo de que fueran lo más cercanos a los del modelo 102 de LivaNova (ver Cuadro 4). Además, se tomaron cinco mediciones de cada parámetro para sacar un promedio del valor el cual es reportado en las secciones siguientes.

9.2.1. Resultados del ancho de pulso de la señal

En la sección siguiente se muestran tomas del osciloscopio del ancho de pulso de la señal de estimulación generadas por el prototipo, en donde se evidencian los resultados conseguidos utilizando los parámetros mencionados en la sección anterior. Además se hace una evaluación de estos para ver si estos cumplen con una replica de los generados por el modelo 102 de LivaNova.

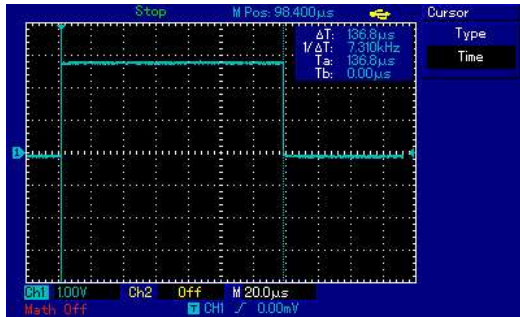


Figura 33: Medición de ancho de pulso de $130 \mu s$

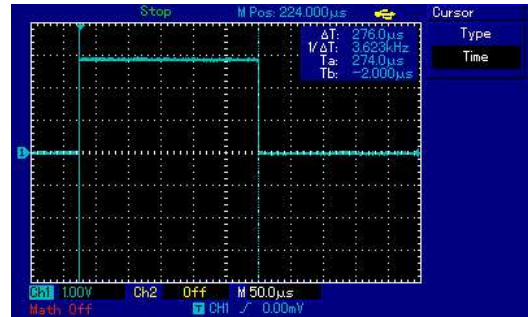


Figura 34: Medición de ancho de pulso de $250 \mu s$

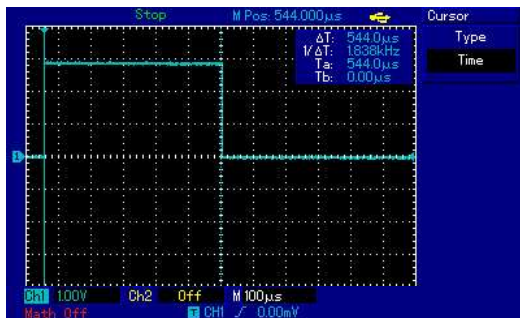


Figura 35: Medición de ancho de pulso de $500 \mu s$

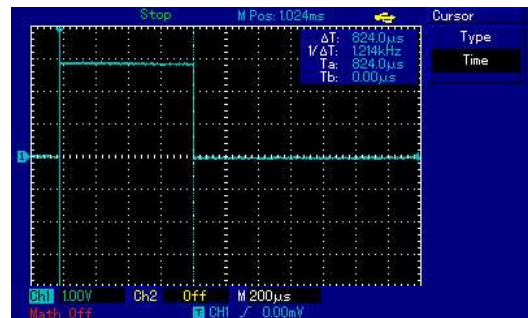


Figura 36: Medición de ancho de pulso de $1000 \mu s$

Ancho de pulso (μs)			
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?
130	137.4	5.69 %	Sí
250	273.3	9.33 %	Sí
500	544.0	8.80 %	Sí
750	822.0	9.60 %	Sí
1000	1100.0	10.00 %	Sí

Cuadro 21: Ancho de pulso teórico y generado en prototipo 2

9.2.2. Resultados de la frecuencia de la señal

En la sección siguiente se muestran tomas del osciloscopio de la frecuencia la señal de estimulación generadas por el prototipo, en donde se evidencian los resultados conseguidos utilizando los parámetros mencionados en la sección anterior. Además se hace una evaluación de estos para ver si estos cumplen con una replica de los generados por el modelo 102 de LivaNova.

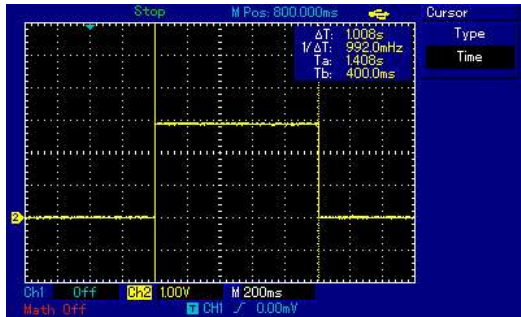


Figura 37: Medición de la frecuencia de 1 Hz

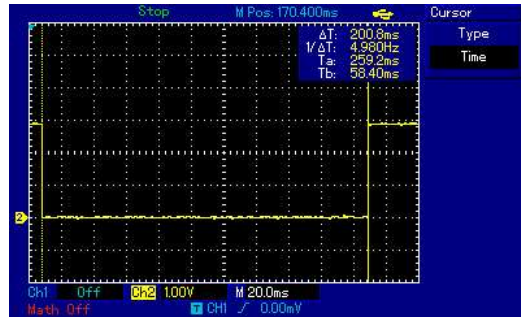


Figura 38: Medición de la frecuencia de 5 Hz

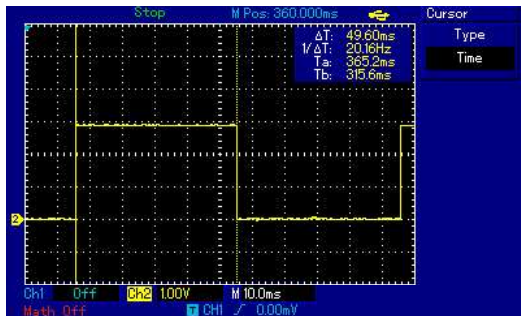


Figura 39: Medición de la frecuencia de 20 Hz

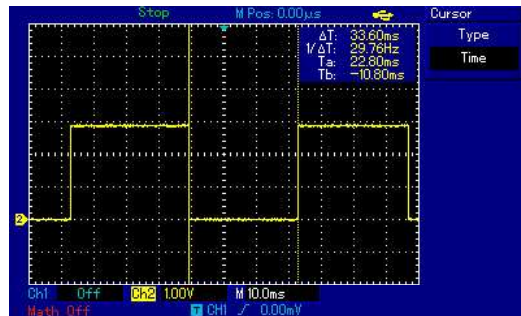


Figura 40: Medición de la frecuencia de 30 Hz

Frecuencia de estimulación (Hz)				
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?	
1	0.984	1.58 %	Sí	
2	1.996	0.20 %	Sí	
5	4.946	1.08 %	Sí	
10	9.991	0.09 %	Sí	
15	15.105	0.70 %	Sí	
20	20.000	0.00 %	Sí	
25	25.065	0.26 %	Sí	
30	29.625	1.25 %	Sí	

Cuadro 22: Frecuencia de estimulación teórica y generada en prototipo 2

9.2.3. Resultados del tiempo de estimulación

En la sección siguiente se muestran tomas del osciloscopio del tiempo de estimulación de la señal generadas por el prototipo, en donde se evidencian los resultados conseguidos utilizando los parámetros mencionados en la sección anterior. Además se hace una evaluación de estos para ver si estos cumplen con una replica de los generados por el modelo 102 de Liva.Nova.

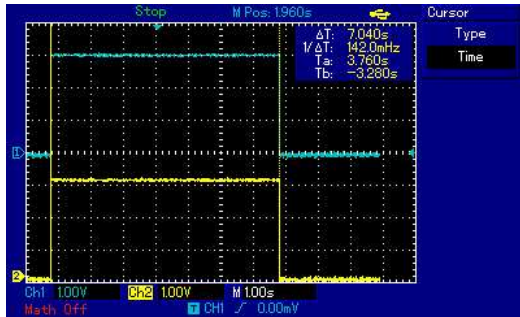


Figura 41: Medición de tiempo de estimulación de 7 s

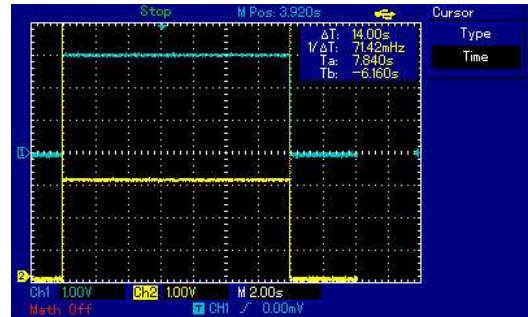


Figura 42: Medición de tiempo de estimulación de 14 s

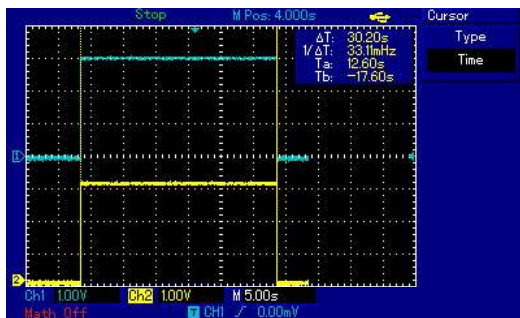


Figura 43: Medición de tiempo de estimulación de 30 s

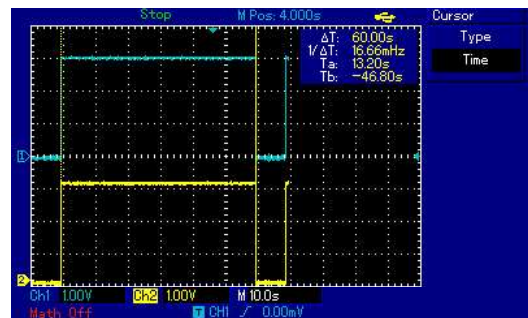


Figura 44: Medición de tiempo de estimulación de 60 s

Tiempo de estimulación (s)			
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?
7	7.04	0.57 %	Sí
14	14.00	0.00 %	Sí
21	21.04	0.19 %	Sí
30	30.20	0.67 %	Sí
60	60.20	0.33 %	Sí

Cuadro 23: Tiempo de estimulación teórico y generado en prototipo 2

9.2.4. Resultados del tiempo de reposo

En la sección siguiente se muestran tomas del osciloscopio para los tiempos de reposo generados por el prototipo, en donde se evidencian los resultados conseguidos utilizando los parámetros mencionados en la sección anterior. Además se hace una evaluación de estos para ver si estos cumplen con una replica de los generados por el modelo 102 de LivaNova. Para los tiempos mayores a 5 minutos, se utilizó un Logic Pro 16 que permite analizar señales por más tiempo con el objetivo de validar los resultados que no se lograron validar en el prototipo 1.



Figura 45: Medición para el tiempo de reposo de 0.2 min

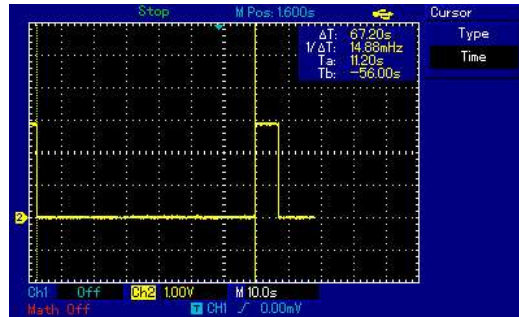


Figura 46: Medición para el tiempo de reposo de 1.1 min



Figura 47: Medición para el tiempo de reposo de 3 min

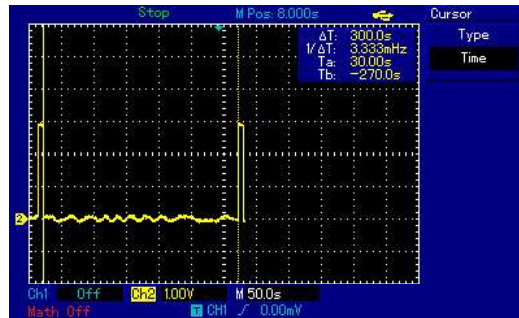


Figura 48: Medición para el tiempo de reposo de 5 min

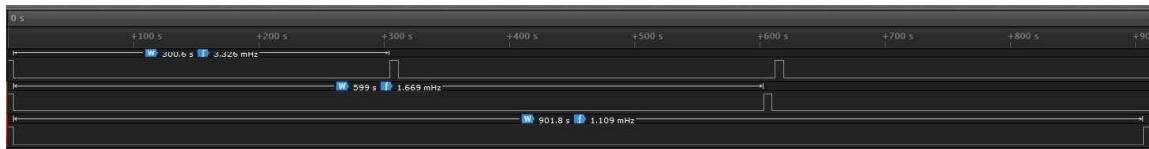


Figura 49: Señal medida para tiempos de reposo de 5, 10 y 15 minutos utilizando el Logic Pro 16

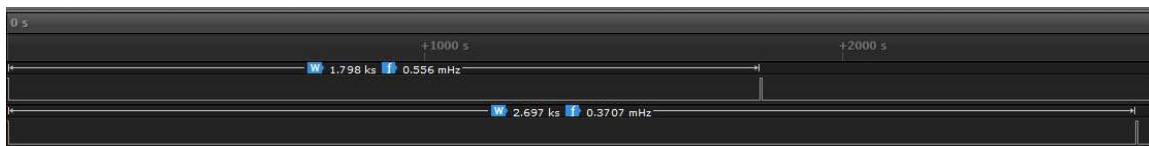


Figura 50: Señal medida para tiempos de reposo de 30 y 45 minutos utilizando el Logic Pro 16

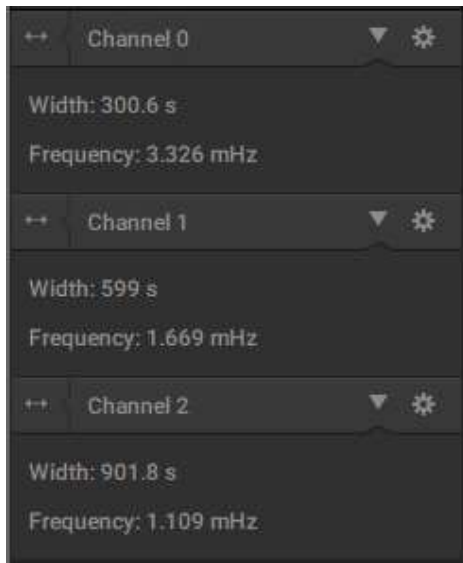


Figura 51: Medición para los tiempo de reposo de 5, 10 y 15 min

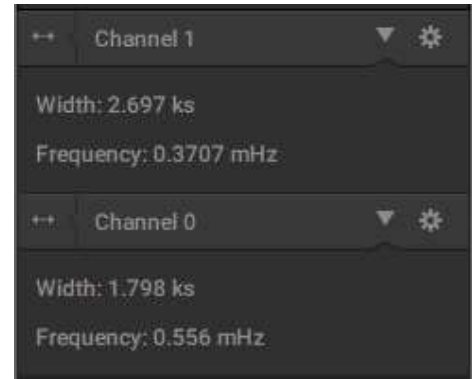


Figura 52: Medición para los tiempo de reposo de 30 y 45 min

Tiempo de reposo (min)			
Valor teórico	Valor experimental	Porcentaje de error	¿Se cumplió?
0.2	0.23	12.67 %	Sí
0.3	0.30	0.00 %	Sí
0.5	0.52	4.67 %	Sí
0.8	0.82	2.92 %	Sí
1.1	1.12	1.82 %	Sí
1.8	1.80	0.00 %	Sí
3	3.00	0.00 %	Sí
5	5.01	0.20 %	Sí
10	9.99	0.17 %	Sí
15	15.03	0.20 %	Sí
20	19.98	0.10 %	Sí
25	25.01	0.04 %	Sí
30	29.97	0.11 %	Sí
35	35.03	0.10 %	Sí
40	40.02	0.05 %	Sí
45	44.95	0.11 %	Sí
50	50.02	0.04 %	Sí
55	54.96	0.07 %	Sí
60	60.05	0.08 %	Sí
90	90.04	0.04 %	Sí
120	119.98	0.02 %	Sí
150	150.04	0.03 %	Sí
180	179.98	0.01 %	Sí

Cuadro 24: Tiempo de reposo teórico y generado en prototipo 2

9.3. Evaluación de resultados

Luego del cambio de estrategia para la realización de la frecuencia y el ancho de pulso se logró cumplir con todos los parámetros de estimulación (ver Cuadros 21 y 22). Para los tiempos de reposo mayores a 5 minutos se utilizó el Logic Pro 16 con el objetivo de no estar limitados por la capacidad del osciloscopio y de su ventana de muestreo de 600 segundos. De esta manera, se cumplen con todos los parámetros como se observa en el Cuadro 25.

Parámetro	Porcentaje de replicación	¿Se cumplió?
Ancho de pulso	100 %	Sí
Frecuencia de estimulación	100 %	Sí
Tiempo de estimulación	100 %	Sí
Tiempo de reposo	100 %	Sí

Cuadro 25: Evaluación de logros prototipo 2

Además de cumplir con todos los parámetros se simplificó el funcionamiento del programa y se adaptó para que este cumpliera con el estándar BARR-C.

9.4. Retos y dificultades

El PIC18F46K22 presentó la misma falla, lo que hizo que la toma de mediciones tomara más tiempo. Además, debido a que se midieron tiempos de hasta tres horas, la toma de mediciones se realizó a lo largo de dos semanas.

El prototipo 3 se enfocará en utilizar la memoria EEPROM. Además de trabajar con la memoria EEPROM, se realizará la primera interconexión con el módulo de potencia para observar que este módulo modula la corriente de la señal de estimulación. Este prototipo será trabajado con el PIC18F46K22.

10.1. Pruebas con la memoria EEPROM

Antes de modificar y agregar la opción de guardar en la memoria EEPROM al programa del prototipo 2 se procedió a probar esto por separado. Primero se investigó en el manual de XC8 como se pueden precargar valores en la memoria, pues de otra forma, no se puede asegurar que los valores dentro de la memoria son ceros. Además, si se quisiera iniciar ciertos parámetros como default para el estimulador no se podría hacer hasta modificarlos de manera remota.

Con lo leído en el manual de XC8, se encontró la directiva `__EEPROM_DATA` que permite al usuario precargar valores en la memoria EEPROM. La configuración del MCU utilizado para realizar la prueba se puede observar en la Figura 53, en donde los pines del puerto A (OUTPUTx) son configurados como salida y los pines del puerto C (INPUTx) son configurados como entradas. En el pin RB0 se coloca un botón para interactuar con el MCU.

Lo realizado para probar el funcionamiento de la memoria fue precargar cierto valor en el primer espacio de memoria. Luego, en el ciclo principal del programa este valor de 8 bits es leído y cargado al puerto A. En el puerto A se conectaron LEDs para mostrar el valor actual en la memoria y comprobarlo de manera visual.

Luego, para modificar el valor, se conectó un DIP Switch de 8 bits en el puerto C como el que se observa en la Figura 54. Al presionar el botón en RB0, el valor que se coloca en el DIP Switch es guardado en la memoria. El nuevo valor es cargado en el puerto A reflejando el nuevo valor en memoria.

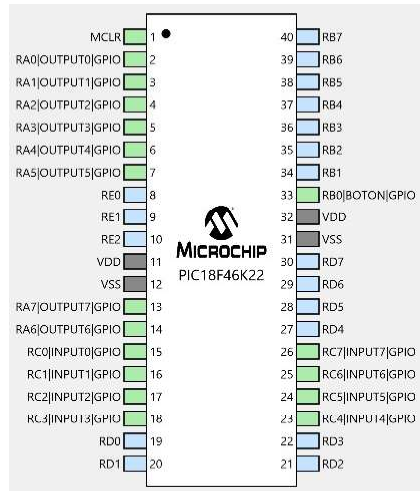


Figura 53: Configuración del PIC para las pruebas de la memoria EEPROM

Para comprobar que el valor fue efectivamente guardado en la memoria no volátil se desconectó la fuente de alimentación y se volvió a conectar. El valor cargado por medio del botón y leído en el puerto C efectivamente es el nuevo valor reflejado en los LEDs conectados al puerto A.

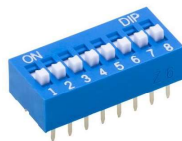


Figura 54: DIP Switch de 8 bits

10.2. Modificaciones a realizar al prototipo 2

Luego de haber comprobado como funciona la memoria EEPROM se pudo proceder a modificar el código del estimulador. Para este prototipo se agregó la parte de inicialización mostrada en la Figura 12. Al agregar esta parte al programa, el dispositivo carga la configuración en memoria cada vez que inicia el programa.

Ya con las funciones de cambios de parámetros programadas en el prototipo 2, se tiene que agregar una función que guarde los valores en la memoria y otra que cargue estos valores al inicio del programa. Se necesitan 4 bytes de memoria EEPROM para guardar los parámetros de estimulación. Los datos fueron organizados en la memoria como se muestra en el Cuadro 26.

Dirección en memoria EEPROM	Parámetro
0x00	Ancho de pulso
0x01	Tiempo de estimulación
0x02	Frecuencia de estimulación
0x03	Tiempo de reposo

Cuadro 26: Organización de los parámetros en la memoria EEPROM

Ya que aún no se tenía conexión con el módulo RF, los parámetros no pueden ser modificados fácilmente, motivo por lo cual se agregó un botón conectado a RB0. Este botón tenía la finalidad de modificar los parámetros a ciertos valores predefinidos.

Para esto se configuró la interrupción de este puerto, así cuando esta se detectara, se activara una bandera. Esta bandera permite que durante el ciclo principal del programa se modifiquen los parámetros. Los parámetros modificados son luego guardados en los mismos espacios de memoria que los mostrados en el Cuadro 26.

Luego de haber presionado el botón, se verifica el cambio en la estimulación, verificando si los valores seteados por el botón son los activos. Luego de comprobar el cambio correcto, desconectamos y volvemos a conectar la fuente para comprobar que los nuevos parámetros sigan activos.

10.3. Interconexión con el módulo de potencia

Luego de haber agregado el funcionamiento de la memoria no volátil al programa del estimulador se procedió a interconectar este con el módulo de potencia. Como se mencionó previamente, este módulo es trabajado por Hugo Elvira en su trabajo de graduación.

Para la interconexión de los módulos se conecta la salida de la señal de estimulación que se encuentra en el pin RD0. Esta señal es modulada completamente por el circuito del otro módulo. En las Figuras 55 y 56 se observa como se ha modulado la señal gracias al módulo de potencia. En las Figuras se puede observar que el ancho de pulso de la señal es el mismo y no se observa sobre-elevamiento debido al control efectuado. La señal se muestra en forma de voltaje ya que el osciloscopio solamente puede medir señales de voltaje.

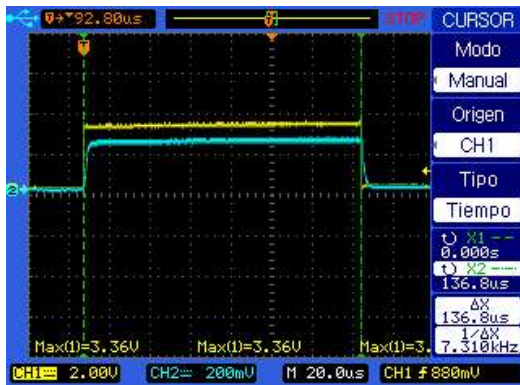


Figura 55: Medición de ancho de pulso de 130 μ s y modulación de amplitud

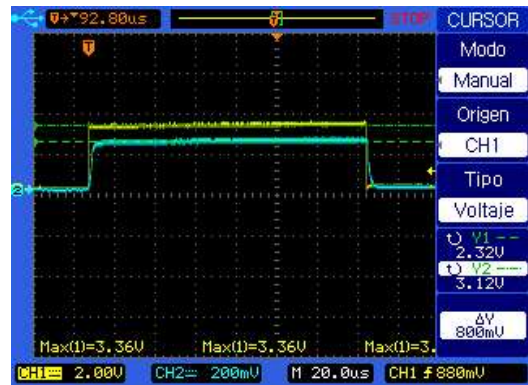


Figura 56: Medición de la modulación de la señal a un ancho de pulso de 130 μ s

10.4. Evaluación de resultados

El prototipo 3 se considera exitoso, pues deja sentada la base para realizar la configuración a través del módulo RF.

Además de cumplir haber logrado la lectura y escritura en EEPROM se mantuvo la estructura del código para que siga cumpliendo con el estándar BARR-C.

10.5. Retos y dificultades

Se tuvo que trabajar con un nuevo módulo del MCU con el que no se había trabajado previamente. El haber realizado la prueba descrita en la sección 10.1 facilitó agregar la parte de la memoria no volátil al código del estimulador. Esta prueba salió con bastante facilidad.

Tomando como base lo logrado con la memoria EEPROM en el prototipo anterior, se tiene como objetivo lograr la comunicación con el módulo RF para así poder modificar los parámetros de estimulación de manera remota. El PIC18F46K22 se usará para realizar este prototipo.

11.1. Configuración del módulo RF

Como primer paso de este prototipo se procedió a configurar el módulo RF seleccionado por Andrés Girón. El módulo seleccionado es un **SPSGRFC-868**, el cual implementa al **SPIRIT1**. La ventaja del SPSGRFC-868 es que ya contiene el circuito de adaptación de impedancias para la antena. Este módulo es un transreceptor, lo que quiere decir que funciona tanto como receptor y receptor. Este módulo puede ser configurado para funcionar a varios rangos de frecuencias portadoras; la frecuencia central portadora seleccionada por Andrés Girón es de 868 MHz.

Para poder configurar este módulo se utiliza el bus serial de comunicación SPI. Ya que Andrés Girón es el encargado de programar el programming wand, el trabajó con el PIC16F1789. Se utilizó este MCU ya que se contaba con tres de estos microcontroladores, lo que facilitaba la realización de pruebas.

El modo de la comunicación SPI del SPSGRFC-868 es 0, por lo cual se configuró de esta manera la interfaz SPI en el PIC. Además, se seleccionó un reloj de transmisión de 1 MHz, esto con el objetivo de realizar estas transacciones lo más rápido posible. Se necesita que las transacciones SPI ocurran lo más rápido posible para que el MCU pueda regresar lo antes posible en algún modo de ahorro de energía, así reduciendo la potencia promedio del dispositivo.

Luego, se crearon varias funciones para hacer la transferencia de datos de acorde a como funciona el módulo, aprovechando la ventaja del *burst mode* del SPIRIT1. El *burst mode* se aplica que mientras mantenga la señal CS en VSS, puedo seguir escribiendo o leyendo los registros de la siguiente dirección a la solicitada. En las Figuras 57 a la 59 se puede observar como se ejecutan las operaciones de lectura y escritura a los registros del SPIRIT1 así como se le dan comandos a este.

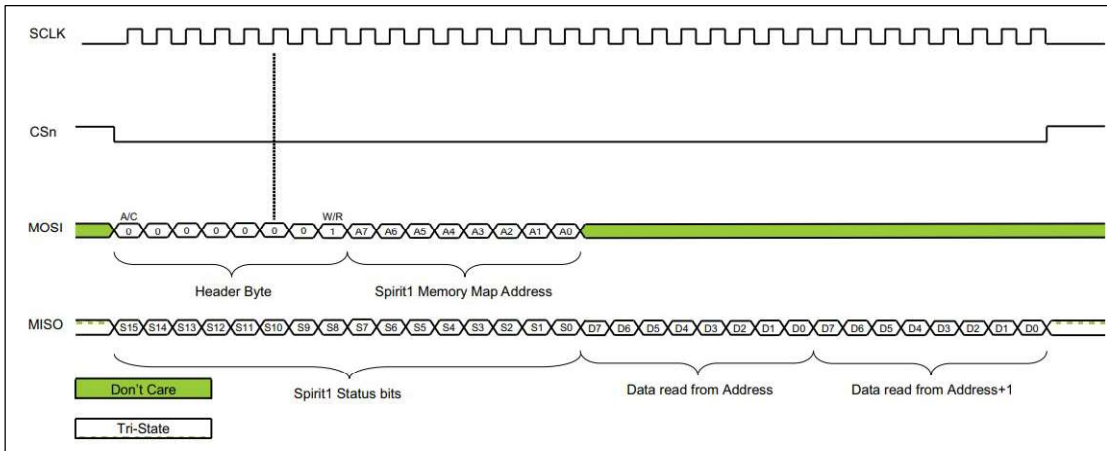


Figura 57: Operación de lectura de registros del SPIRIT1

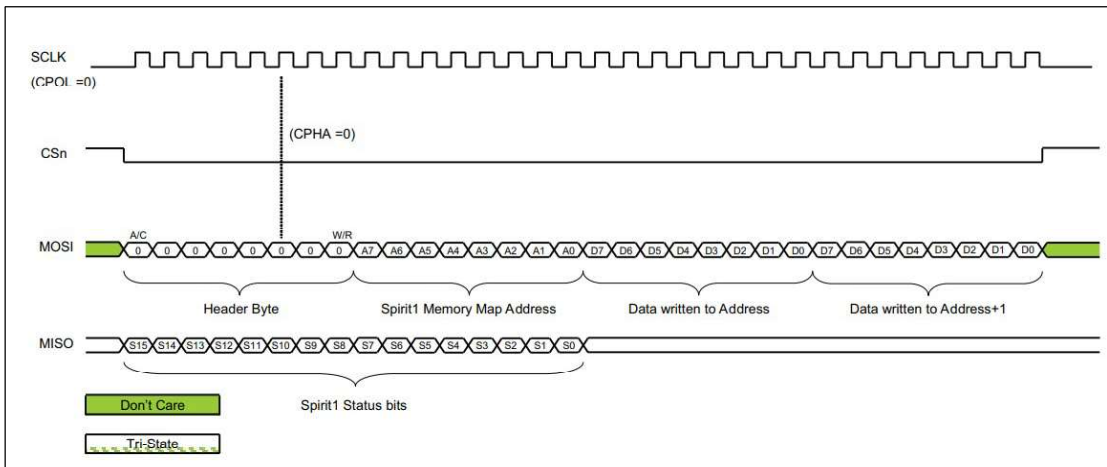


Figura 58: Operación de escritura de registros del SPIRIT1

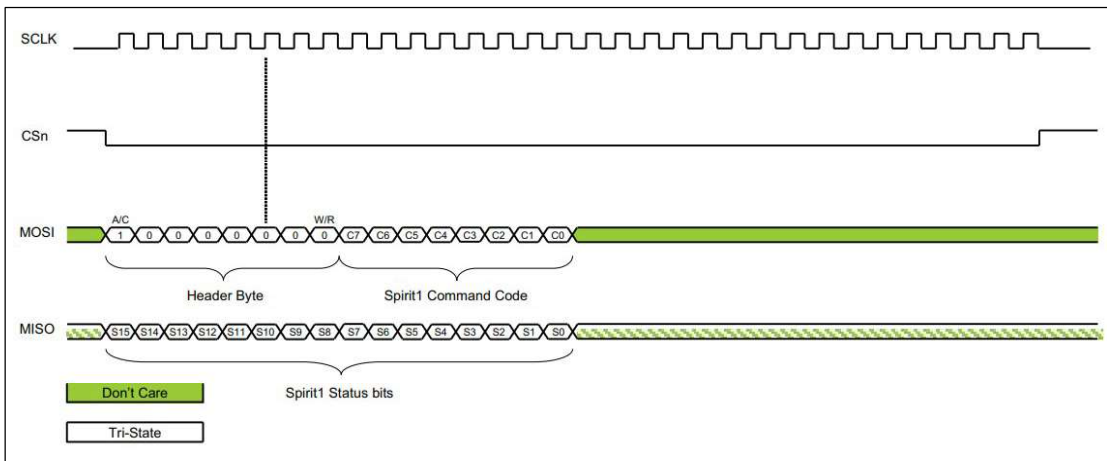


Figura 59: Operación de comando del SPIRIT1

El encargado de la comunicación inalámbrica se encargó de escoger la configuración del

módulo, para que se procediera a configurarlo según las características que él cree son las más convenientes para la aplicación. Algunos de los parámetros configurados del SPSGRFC son los siguientes:

- Frecuencia central de 868 MHz
- Filtrado por CRC y dirección
- Potencia de transmisión de -20dbm
- Modulación 2-FSK
- Data rate de 38.5 kbps

Ya con la tabla de registros y sus valores, se procedió a configurar el módulo. Para comprobar el funcionamiento de la comunicación primero se procedió a leer los registros que brindarían la información del dispositivo, como su número de parte y su versión. Luego de confirmar que la comunicación SPI con el SPIRIT1 era exitosa y que las funciones lograban transmitir la información correctamente, se procedió a configurar el módulo.

Luego de ya haber configurado el módulo se realizó la prueba en conjunto con Andrés Girón de comunicar 2 SPSGRFC, esto con el objetivo de probar la comunicación antes de incorporarlo al código del estimulador.

La prueba realizada para comunicar dos módulos SPSGRFC fue configurar uno como receptor y el otro como emisor. En el emisor, se lee el DIP Switch conectado en el puerto C del PIC16F1789. Luego, al presionar el botón este envía el paquete.

En el receptor, se agregó un botón que da el comando al SPSGRFC de entrar en RX. Luego, se configuraron interrupciones en el pin RB1 del MCU. En la configuración del SPSGRFC se agregó que el GPIO3 de este módulo notifique por medio de interrupciones al microcontrolador cuando este haya recibido un mensaje exitoso. Al momento que el MCU conectado receptor detectaba la interrupción, este procedía a leer el RX FIFO, el cual contiene la información que el emisor envió. Esta información es luego mostrada a través del puerto A por medio de LEDs en donde se verifica que la información recibida es la que se colocó en el DIP Switch conectado al PIC con la función de emisor.

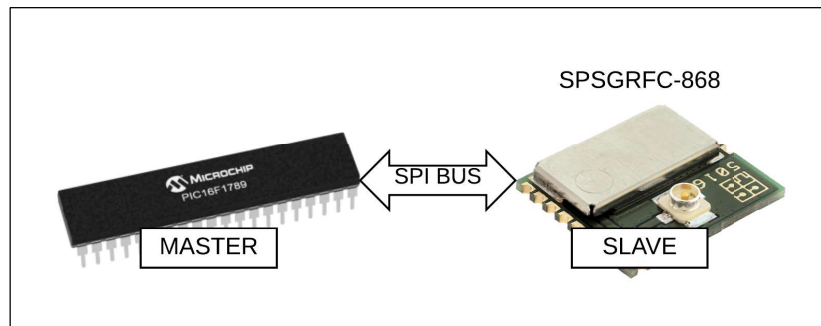


Figura 60: Comunicación entre PIC16F1789 y SPSGRFC-868

11.2. Modificaciones a realizar al prototipo 3

Luego de haber logrado comunicación exitosa entre dos módulos RF distintos se debe proceder a juntar el código del estimulador y el de comunicación RF. Primero se modificará el

primer acercamiento mostrado en la Figura 12. El nuevo acercamiento (ver Figura 61) tendrá tres modos de operación de igual manera que el acercamiento anterior, pero a diferencia que el modo estimulación incorporará todo lo necesario para generar la señal de estimulación. Estos modos son los que se pueden observar en el Cuadro 27. La función de cada modo se explica a continuación.

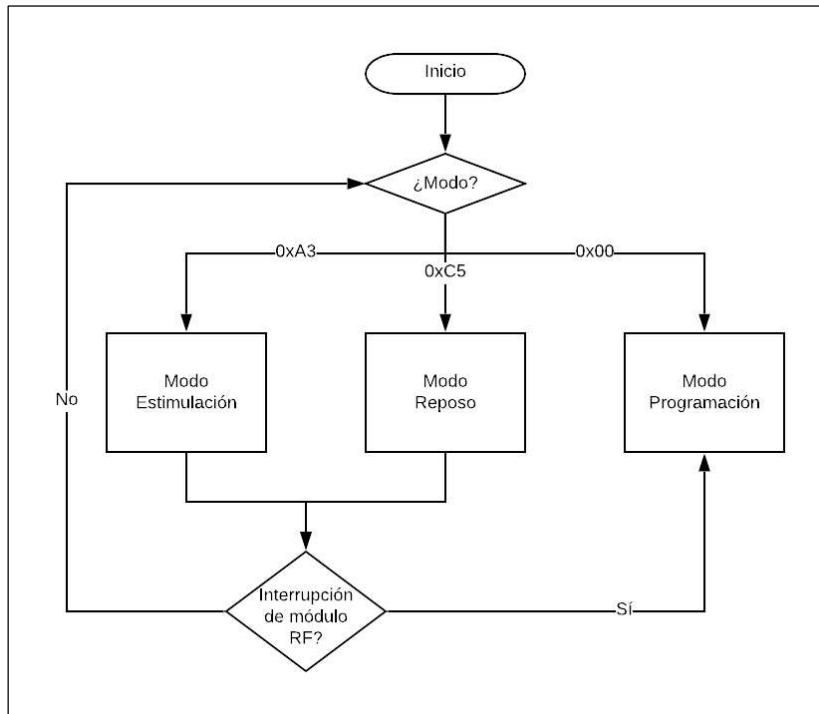


Figura 61: Segundo acercamiento

11.2.1. Modo estimulación

El modo estimulación replicará todo lo mostrado en la Figura 12, en donde se cumple con la terapia, que consiste en estimular y un tiempo de reposo. La ventaja de realizarlo de esta manera es se puede programar para que el estimulador no estimule por completo. En este modo se incorporarán los modos de ahorro de energía discutidos previamente y su funcionamiento para producir los tiempos de reposo será igual a como se han generado en los prototipos anteriores.

La única manera para salir de este modo es mediante una bandera que es activada cuando se presiona el botón conectado a RB0. Al presionar este botón, el MCU entra en el modo de programación.

11.2.2. Modo reposo

El modo de reposo consistirá en un modo en donde el dispositivo este en constante ciclo de ahorro de energía. Este modo es útil para cuando aún no se activado o configurado los parámetros para la terapia. El estimulador, al encontrarse en este modo de operación, su consumo de energía es el más bajo posible, pues solamente está esperando a ser programado.

La manera en que se realizo este modo de operación es un ciclo infinito en donde el MCU entra en modo de dormir y es despertado cuando el tiempo de expiración del WDT expira.

La única manera de salir de este modo, es presionando el botón conectado a RB0, que es lo que permite entrar al dispositivo en modo de programación.

11.2.3. Modo programación

Este modo es el encargado de cambiar los parámetros del estimulador. Para entrar a este modo, se tiene que venir de algún otro modo (estimulación o reposo) previo. Al entrar a este modo el MCU cambia su reloj para que este opera a 4 MHz. El motivo de esto es que las transacciones SPI que se realizan con el módulo SPSGRFC ocurran lo más rápido posible, esto por motivos de potencia. Se cambia la frecuencia para que la comunicación del SPI se lleve a cabo a 1 MHz, como fue realizado en las pruebas anteriores.

Se programó el módulo RF para que genere ciertas interrupciones. Cuando cambia el estado del pin RB1, el PIC procede a leer los registros del SPIRIT1 para determinar que interrupción fue la que activó el cambio en RB1. En caso de leer que ha recibido un paquete exitosamente, el PIC procede a leer este para luego modificar los parámetros de estimulación y el modo que ha sido programado.

Los modos de operación fueron codificados para que así pudieran ser programados de manera remota. Al recibir un valor de 0xC5, el estimulador entra en el modo de reposo, consumiendo la menor cantidad de energía. Al recibir un 0xA3, este estimula utilizando los parámetros recibidos. Antes de entrar a alguno de los otros módulos, se cambia el reloj represándolo a 31kHz.

Modo	Codificación
Reposo	0xC5
Estimulación	0xA3
Programación	0x00

Cuadro 27: Codificación de los modos de operación

11.2.4. Protocolo de comunicación

Para el envío y recepción de información se utilizó el protocolo definido por el paquete observada en el Cuadro 28. Esta paquete consiste de tres bytes, que contienen la información de los parámetros y modo de operación codificada.

En el primer byte se recibe el modo de operación. Los valores que se pueden recibir son el de estimulación (0xA3) o el de reposo (0xC5). Al recibir un valor incorrecto se descarta y se vuelve a esperar un nuevo paquete a ser recibido.

En el segundo byte, se recibe los valores del ancho de pulso (PW) y el de tiempo de estimulación (ST). Los valores que representan estos tres bits se pueden decodificar utilizando los Cuadros 19 y 18 respectivamente.

En el tercer byte se reciben los valores de la frecuencia de estimulación (FREQ) y el de tiempo de reposo (SLEEP TIME). Para codificar que valor representan estos valores codificados se utilizan los Cuadros 17 y 20 respectivamente.

Los valores se codificaron con el objetivo de mandar la información (modo de operación y 4 parámetros de estimulación) en la menor cantidad de bytes necesaria.

Byte 1								Byte 2								Byte 3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MODO								0	PW			0	ST			FREQ				SLEEP TIME			

Cuadro 28: Formato del paquete utilizado

11.3. Evaluación de resultados

Para evaluar el prototipo, Andrés Girón realizó la interfaz de programación para que desde la computadora se pudieran seleccionar los parámetros y modo de operación que el médico desee seleccionar. El estimulador se programó para que iniciara en los dos modos distintos, así, observar el cambio fácilmente al recibir los nuevos parámetros.

Se utilizaron varios pines como salida con LEDs conectados para fácilmente entender en que modo se encontraba. La prueba para comprobar el funcionamiento del prototipo es la siguiente:

- Programar el estimulador con ciertos parámetros y modo inicial
- Presionar botón para entrar a modo de programación
- Mandar parámetros distintos a los programados desde la computadora
- Observar el cambio de modo y de los parámetros en caso de entrar a modo de estimulación
- Reiniciar el MCU y observar si este inicia en el modo programado desde la computadora
- Se vuelve a repetir la prueba sin reprogramar el MCU

La prueba mencionada se repitió extensamente, con lo cual se logró comprobar que el estimulador funciona de manera correcta al interconectarse con el módulo de comunicación inalámbrica. Con este prototipo se puede validar el cambio de parámetros, el uso de memoria no volátil y la generación de la señal de estimulación según los parámetros del modelo 102 de LivaNova.

11.4. Retos y dificultades

El trabajar con el módulo SPSGRFC resultó más complicado y tardado de lo que se esperaba. Primero que todo, el módulo cuenta con aproximadamente 150 registros de configuración y bastantes opciones que se pueden configurar. Luego de esto, se procedió a programar las rutinas para el intercambio de información mediante SPI. Estas rutinas fueron diseñadas utilizando la hoja de datos del SPIRIT1, que es el transreceptor que tiene el SPSGRFC. El SPSGRFC está basado en un SPIRIT1. En la hoja de datos, se indicaba que el SPIRIT1 funciona en el modo 1 de SPI.

Luego de una semana de pruebas y no lograr enviar y recibir los datos deseados en el SPSGRFC se utilizó el Logic Pro 16 utilizado para medir los tiempos de reposo en el prototipo 2 ya que este tiene la capacidad de analizar diferentes protocolos de comunicación, entre ellos SPI. Se comprobó que la información se mandaba de manera correcta pero el SPIRIT1 no recibía la información de manera correcta. Por esta razón, se descubrió que este

módulo en verdad funcionaba utilizando el modo 0 de SPI. Al ya cambiar la configuración del SPI en el PIC16F1789 se logró configurar de manera correcta y luego se logró comunicar dos módulos SPSGRFC.

Posteriormente, se procedió a portar el código que realiza la comunicación con el módulo SPSGRFC del PIC16F1789 al PIC18F46K22. Este proceso debería de ser fácil, pues ambos MCUs poseen los mismos módulos, solamente cambia la ubicación del módulo en los puertos del microcontrolador. Al ya haber portado el código, se observó que funcionó una vez, luego, el programa por más que fuera cargado no se lograba configurar el módulo SPSGRFC. Se utilizó el Logic Pro 16 y se observó que la información se envía de manera correcta pero el módulo RF no funcionaba de la manera esperada.

Luego, se hicieron pruebas cambiando el PIC16F1789 y este si configuraba de manera correcta al SPSGRFC, por lo cual, de alguna manera, el PIC18F46K22 tenía un defecto que no permitía la comunicación. Por lo cual, se procedió a utilizar un PIC18F45K22, el MCU de la misma familia pero con menor memoria RAM y de programa. El resultado fue el mismo. Debido a limitantes de tiempo, se procedió a exportar el código del prototipo 3 al PIC16F1789. Este implicó que se tuvieron que modificar ciertas funciones pues los microcontroladores, a pesar de ser similares, ciertos módulos no son exactamente iguales.

Después de haber portado el código, se logró configurar el módulo RF exitosamente para luego lograr la prueba realizada haciendo que estimulador funcionara según la configuración enviada a través de la computadora.

El problema más grande de haber portado el código es que los tiempos de la señal de estimulación fueron descalibrados.

Este capítulo trata acerca de la recompilación de los tres distintos módulos que componen el estimulador del nervio vago, en un prototipo de placa impresa. En este capítulo se realizará la cantidad de prototipos en placa impresa que sean necesarios para cumplir con los objetivos siguientes:

1. Comunicación bidireccional entre el estimulador y la varilla de programación.
2. Funcionamiento correcto de la estimulación y sus parámetros.
3. La placa impresa no tenga ningún arreglo después de fabricada.

Para todos los prototipos finales realizados se utilizarán los siguientes componentes como mínimo, pues estos son los seleccionados para cumplir con los requerimientos de funcionamiento y potencia.

- PIC18F27J13
- Módulo SPSGRF-868
- Módulo de potencia según esquemático presentado en la Figura 69
- Interfaz de conexión para voltajes de operación y de programación

Además, todas las placas tienen que cumplir con el objetivo de utilizar componentes SMD y que esta no exceda un área superficial mayor a 25 centímetros cuadrados. A continuación se presentarán todos los diseños diseñados y elaborados con tal de cumplir con los objetivos anteriores, tanto como de parte física hasta la parte lógica de programación.

12.1. Prototipo final 1

Como prototipo final se utilizó el PIC16F1789 debido a que no se consiguió el PIC18F27J13. Para el prototipo final se diseñó una placa impresa con componentes SMD. Además, se calculó el tiempo de vida útil que este dispositivo tendría ya incorporando los tres módulos que componen al estimulador.

12.1.1. Esquemático y placa impresa del prototipo final

Para la realización del estimulador se procedió a realizar el diseño que incluye a los tres módulos que componen a este. Cada integrante brindo su esquemático y fue unido para poder integrar el esquemático final del estimulador. Los componentes utilizados se muestran en el BOM en el Cuadro 29. Además, se calculó un costo total de los componentes el cual no asciende a más de Q200.00.

Parte	Descripción	Cant.	Precio U.	Precio T.
PIC16F1789	Microcontrolador	1	Q 30.00	Q 30.00
SPSGRFC-868	RF Transceiver	1	Q 130.00	Q 130.00
Header 1x6		1	Q 1.00	Q 1.00
Header 1x5		1	Q 1.00	Q 1.00
2N3904	NPN General Purpose Amplifier	1	Q 3.00	Q 3.00
ADTL082ARMZ	2x OpAmps TL082	1	Q 21.00	Q 21.00
Resistor	Res Smd 1k Ohm 0.1% 1/2w 0603	5	Q 0.25	Q 1.25
Switch SMD	Switch SMD 12V	1	Q 3.50	Q 3.50
Total				Q 190.75

Cuadro 29: BOM del estimulador del nervio vago

Luego de haber unido los tres módulos en un solo esquemático, (se muestra en la Figuras 68 a la 70 en los anexos), se procedió a diseñar la placa que se muestra en la Figura 64 la cual mide 45 mm x 45 mm, cumpliendo así con la medida requerida y especificada en los objetivos del trabajo de graduación. Esta placa se realizó de dos capas debido a su complejidad. El tamaño y diseño de la placa se puede observar con claridad en las Figuras 62 y 63.

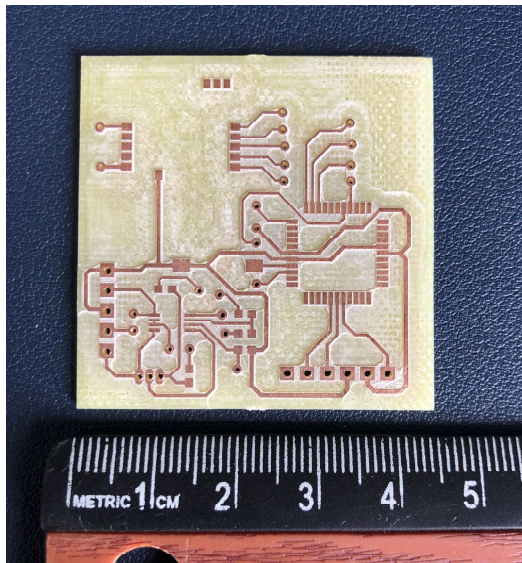


Figura 62: Medición del tamaño del PCB lado A

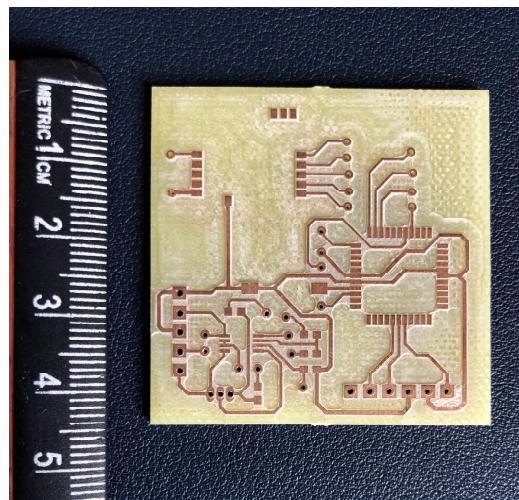


Figura 63: Medición del tamaño del PCB lado B

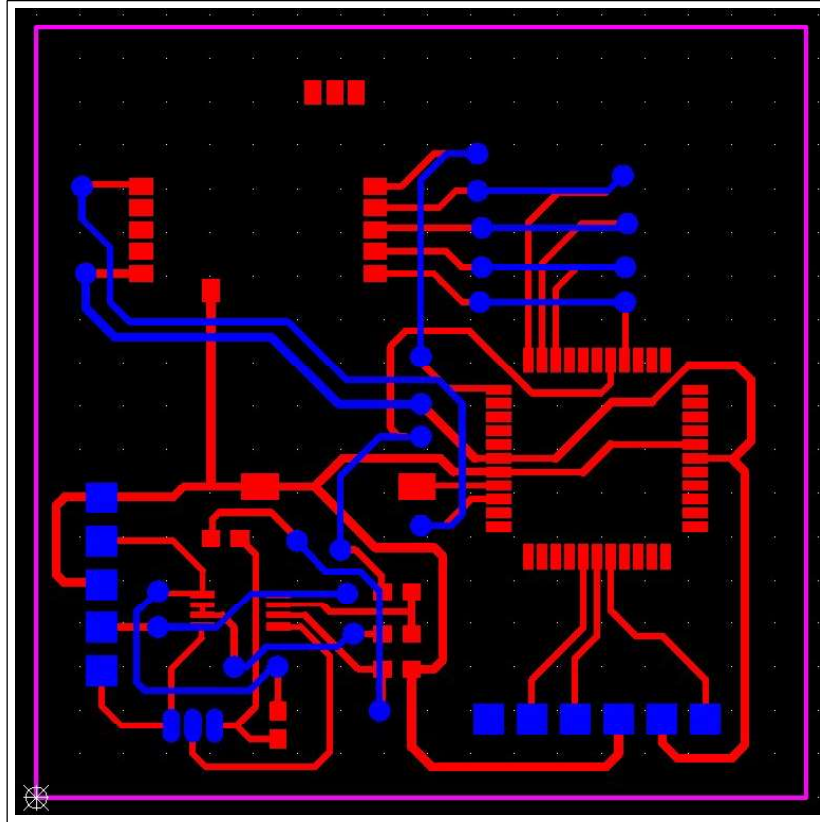


Figura 64: Diseño del PCB de dos capas

12.2. Prototipo final 2

Para el prototipo final 2 se tomó como base el prototipo final 1 y se trabajó sobre este, mejorando tanto la parte de la lógica, que es el software y la parte física que es el hardware. Las mejoras que quedaron pendientes del prototipo anterior fueron las siguientes:

- Comunicación inalámbrica automática (sin botón)
- Comunicación bidireccional
- Utilizar PIC18F27J13
- Más indicadores para ver estado de funcionamiento
- Mejorar efectividad de comunicación

La mejora principal de este prototipo es agregar la comunicación bidireccional, así de esta manera poder confirmar que el estimulador recibió y configuró de manera correcta los parámetros y su modo de operación. Para realizar esto se realizaron pruebas intensivas junto con Andrés Girón hasta lograr este objetivo. Se realizaron pruebas con los distintos parámetros de estimulación y modos de funcionamiento. Para mejorar la efectividad de la comunicación se cambió la lógica de funcionamiento, en donde el estimulador se reinicia cada vez que recibe nuevos parámetros. La razón por la que se realizó esto de esa manera

es por defectos en el módulo de comunicación, que cuando recibe un mensaje este queda en un ciclo en el cual que no puede volver a recibir otro mensaje nuevamente.

Los cambios que se realizaron al flujo del dispositivo se pueden observar en la Figura 65. En esta se observa el cambio realizado a la hora del funcionamiento. Además, se cambió para que el dispositivo solamente pudiera ser programado cuando se encuentra en modo de reposo o en modo sleep, de esta manera, se evita que la llegada de un mensaje interrumpa la estimulación al paciente. Además de esto, se realizó para que el módulo de comunicación solamente estuviera activo cada 4 segundos dentro del periodo de reposo. Con este cambio, se reduce la cantidad de tiempo que esta activo el módulo RF, así se reduce la potencia promedio del dispositivo.

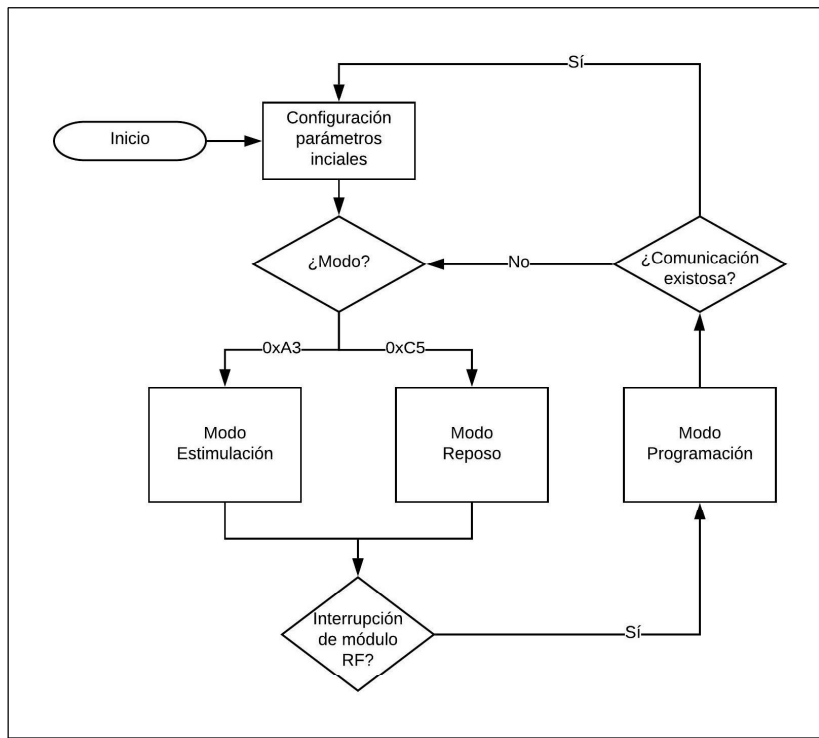


Figura 65: Diagrama de flujo acercamiento prototipo final

Con las modificaciones anteriores se mejoro significativamente la comunicación del dispositivo y además se agregó la comunicación bidireccional. El porcentaje de efectividad pasó de ser 20 % a más de 95 %. Además, otra mejora substancial respecto al prototipo anterior fue hacer que la comunicación inalámbrica fuera automática, ya que no se logró en el prototipo anterior. Los nuevos ajustes hacen que sin necesidad de oprimir un botón se pueda programar el dispositivo. Con esta mejora, se lograron cumplir los objetivos para la comunicación inalámbrica

En los cambios a la placa, en la parte del hardware se implementó el dispositivo con el PIC18F27J13 el cual tiene mejores funciones que optimizan su consumo de energía, siendo menor al del PIC16F1789. Además de este cambio, se agregaron switches físicos para poder modificar fácilmente el modo de funcionamiento físico, ya sea pruebas o estimulación, además se agregaron varios LEDs (4 en total) para ayudar a visualizar el estado actual del estimulador y de manera visual ver el funcionamiento del mismo, así poder hacer una depuración de manera más fácil.

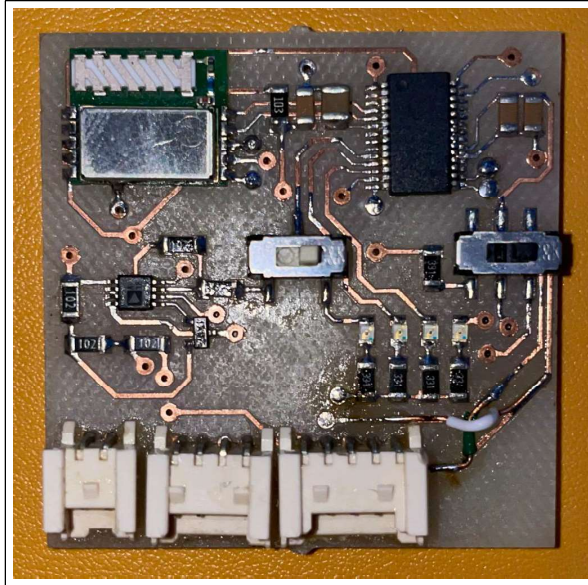


Figura 66: Placa soldada del prototipo final con PIC18F27J13

12.3. Estimación de vida útil

Para el cálculo de la vida útil del estimulador primero se debe calcular la corriente promedio que consume cada módulo. Para calcular este promedio, esta corriente se promedia en el tiempo, esto quiere decir que se calcula la corriente promedio que consume el módulo durante un periodo de tiempo. Lo anteriormente mencionado se realizó debido a que los módulos no siempre consumen la misma corriente, pues esta depende de la función que está realizando el módulo.

Para el cálculo de la corriente se tomaron los parámetros siguientes:

- Corriente de estimulación de 3 mA
- Tiempo de estimulación de 30 s
- Tiempo de reposo de 5 min
- Frecuencia de estimulación de 30 Hz
- Ancho de pulso de 1000 μs
- Cambio de parámetros semanal

Con los parámetros anteriores se procedió a calcular la corriente promedio de cada módulo. Para la corriente promedio del módulo de potencia, los valores de estas corrientes fueron calculados por Hugo Elvira, encargado de este módulo. Las corrientes calculadas fueron las de estimulación y las de no estimulación. Los valores fueron calculados para una corriente de 3 mA de estimulación. Para calcular la corriente promedio del módulo, se calculó el tiempo que este módulo estaría activado utilizando la frecuencia de 30 Hz y el ancho de pulso de 1000 μs . Se calculó que el tiempo que estaba el módulo encendido era de 900 μs y el tiempo apagado era de 29.9991 s. Luego, este valor se promedia con el tiempo totalmente apagado, que es cuando el todo el módulo está apagado, que son los 5 minutos.

Para el módulo de control, se calculó la corriente promedio del módulo utilizando la hoja de datos del PIC16F1789. Luego se tomaron los módulos que se utilizan del MCU para cuando este estaba estimulando y para cuando entraba en modo Sleep. Luego, se promedió la corriente en este tiempo. Para el módulo RF se realizó algo similar. Estos valores fueron calculados por Andrés Girón, encargado de este módulo. Se calculó este valor de la corriente promediando una recepción de una duración de 4 segundos y un cambio de parámetros semanal. Con los cálculos, se realizó el Cuadro 30.

Módulo	Corriente promedio (mA)	Porcentaje
Potencia	0.37732	99.63 %
Control	0.00141	0.37 %
RF	0.00067	0.16 %
Corriente promedio	0.379	100.0 %

Cuadro 30: Corriente promedio de los módulos

Como se puede observar en el Cuadro 30 el módulo de potencia es el que más corriente consume, por lo cual es el que limita la vida útil que podría tener el dispositivo. Según los datos brindados por Hugo Elvira, el circuito consume 4.15 mA cuando no estimula, lo cual es aproximadamente 2000 veces más grande que los otros dos módulos. Para aumentar la vida útil del dispositivo se debe reducir la corriente del módulo de potencia, pues este módulo consume aproximadamente el 99.63 % del estimulador.

Para el cálculo de la batería, se utiliza la ecuación 2. Se tomaron como referencia las baterías del fabricante Eagle-Picher. Esta empresa se dedica a la fabricación de baterías de estándar médico como lo son sus modelos LTC-15 MC-S7. Este modelo tiene una capacidad de 1500 mAh y una dimensión aceptable para el tamaño del neuro-estimulador. Además, esta empresa fabrica baterías de tamaño y forma hechas a la medida.

Realizando los cálculos, una batería que pueda ofrecer al dispositivo de una vida útil de tres años, es decir unas 26 mil horas de vida, debe de ser de un tamaño aproximado de 10 Ah. Debido a limitantes de tamaño del estimulador se recomendaría utilizar cuatro celdas de 1500 mAh cada una, ofreciendo una capacidad de 6000 mAh que le daría al estimulador una vida útil de aproximadamente dos años.

Las conclusiones son:

1. Se replicaron todos los parámetros de la señal de estimulación que ofrece el modelo 102 de LivaNova dentro de los rangos de porcentaje de error según el parámetro correspondiente. Para el ancho de pulso el porcentaje de error es de 10 %; para la frecuencia de estimulación es de 1.58 %; para el tiempo de estimulación es de 0.67 % y para el tiempo de reposo la desviación máxima es de ± 2 segundos.
2. Se implementó un algoritmo en el estimulador que permite modificar los parámetros actuales de estimulación de manera confiable. Además, el algoritmo no permite que se estimule al paciente con parámetros fuera del universo de los parámetros de la estimulación previamente definidos por el modelo 102 de LivaNova.
3. El algoritmo del estimulador presenta un diseño robusto pues se utilizaron características del MCU como memoria no volátil con el objetivo de garantizar que en caso de alguna falla externa, el estimulador continúe la estimulación bajo los parámetros previamente configurados por el médico.
4. Se seleccionó una batería de 6000 mAh de la marca EaglePicher con estándar médico para implantación que brinda una vida útil al estimulador de aproximadamente 2 años. El consumo de corriente promedio del estimulador es de $379 \mu A$, consumiendo el módulo de potencia el 99.75 % de esta corriente, siendo un factor limitante para que el estimulador alcance la vida útil de 3 años.
5. Se diseñó una placa impresa de un tamaño de un área de 20.25 cm^2 con componentes SMD lo cual cumple con el tamaño máximo planteado con HUMANA, teniendo así la placa un tamaño similar al del modelo 102 de LivaNova.
6. Se logró una comunicación bidireccional entre el estimulador y la varilla de programación con más de un 90 % de efectividad.

Luego de haber diseñado el estimulador del nervio vago que produce la señal cumpliendo todos los parámetros se recomienda exportar este código al PIC18F27J13, pues este tiene un menor consumo de energía comparado al microcontrolador utilizado en el prototipo final.

Además, se recomienda utilizar otro módulo RF que sea de menor potencia para que este hardware sea igualmente eficiente al MCU. Además, se recomienda el uso de un regulador de voltaje con el objetivo de garantizar un voltaje estable de alimentación para los sistemas del estimulador.

Del mismo modo, se recomienda agregar el modo imán al prototipo, pues su implementación lo vuelve equiparable al modelo 102 de LivaNova. Para la realización de este modo se tiene que agregar un Reed Switch como el que se observa en la figura 3, luego, configurar interrupciones en algún puerto, para que cuando este switch se cierre, genere una interrupción que pueda ser interpretada por microcontrolador para activar el modo imán.

Como última recomendación, se recomienda mejorar la eficiencia del módulo de potencia ya que este módulo es el que limita la vida útil del estimulador. A pesar de esto, se recomienda que se utilice el PIC18F27J13 el cual ofrece un manejo de consumo de energía superior al que ofrece el PIC16F1789.

-
- [1] V. S. Wasade, L. Schultz, K. Mohanarangan, A. Gaddam, J. M. Schwalb y M. Spanaki-Varelas, «Long-term seizure and psychosocial outcomes of vagus nerve stimulation for intractable epilepsy», *Epilepsy & Behavior*, vol. 53, págs. 31-36, dic. de 2015. DOI: 10.1016/j.yebeh.2015.09.031. dirección: <https://doi.org/10.1016/j.yebeh.2015.09.031>.
- [2] R. Mohanraj y M. J. Brodie, «Diagnosing refractory epilepsy: response to sequential treatment schedules», *European Journal of Neurology*, vol. 13, n.º 3, págs. 277-282, mar. de 2006. DOI: 10.1111/j.1468-1331.2006.01215.x. dirección: <https://doi.org/10.1111/j.1468-1331.2006.01215.x>.
- [3] J. W. Wheless, «Intractable epilepsy: A survey of patients and caregivers», *Epilepsy & Behavior*, vol. 8, n.º 4, págs. 756-764, jun. de 2006. DOI: 10.1016/j.yebeh.2006.03.010. dirección: <https://doi.org/10.1016/j.yebeh.2006.03.010>.
- [4] R. S. Fisher, B. G. Vickrey, P. Gibson, B. Hermann, P. Penovich, A. Scherer y S. Walker, «The impact of epilepsy from the patient's perspective I. Descriptions and subjective perceptions», *Epilepsy Res.*, vol. 41, n.º 1, págs. 39-51, ago. de 2000.
- [5] S. L. Helters, M. S. Duh, A. Guérin, S. P. Sarda, T. M. Samuelson, M. T. Bunker, B. D. Olin, S. D. Jackson y E. Faught, «Clinical and economic impact of vagus nerve stimulation therapy in patients with drug-resistant epilepsy», *Epilepsy & Behavior*, vol. 22, n.º 2, págs. 370-375, oct. de 2011. DOI: 10.1016/j.yebeh.2011.07.020. dirección: <https://doi.org/10.1016/j.yebeh.2011.07.020>.
- [6] B. Koo, «EEG Changes With Vagus Nerve Stimulation», *Journal of Clinical Neurophysiology*, vol. 18, n.º 5, págs. 434-441, sep. de 2001. DOI: 10.1097/00004691-200109000-00008. dirección: <https://doi.org/10.1097/00004691-200109000-00008>.
- [7] R. W. Roosevelt, D. C. Smith, R. W. Clough, R. A. Jensen y R. A. Browning, «Increased extracellular concentrations of norepinephrine in cortex and hippocampus following vagus nerve stimulation in the rat», *Brain Research*, vol. 1119, n.º 1, págs. 124-132, nov. de 2006. DOI: 10.1016/j.brainres.2006.08.048. dirección: <https://doi.org/10.1016/j.brainres.2006.08.048>.

- [8] P. Boon, R. Raedt, V. de Herdt, T. Wyckhuys y K. Vonck, «Electrical stimulation for the treatment of epilepsy», *Neurotherapeutics*, vol. 6, n.º 2, págs. 218-227, abr. de 2009, ISSN: 1878-7479. DOI: 10.1016/j.nurt.2008.12.003. dirección: <https://doi.org/10.1016/j.nurt.2008.12.003>.
- [9] K. Vonck, V. D. Herdt, T. Bosman, S. Dedeurwaerdere, K. V. Laere y P. Boon, «Thalamic and limbic involvement in the mechanism of action of vagus nerve stimulation, a SPECT study», *Seizure*, vol. 17, n.º 8, págs. 699-706, dic. de 2008. DOI: 10.1016/j.seizure.2008.05.001. dirección: <https://doi.org/10.1016/j.seizure.2008.05.001>.
- [10] Humana, Centro de epilepsia y neurocirugía funcional, *Historia Humana*, Accedido por última vez el 31 de julio de 2019, dirección: <https://www.humanagt.org/historia.html>.
- [11] Chicago Institute of Epilepsy, *Vagus Nerve Stimulation: Cost of treatment*, Accedido por última vez el 31 de julio de 2019, 2016. dirección: <https://epilepsy-chicago.org/epilepsy/treatment/vagus-nerve-stimulation/cost-of-treatment/>.
- [12] M. Candia, *Guatemala: Declaración al término de la misión sobre la Consulta del Artículo IV de 2019*, Accedido por última vez el 31 de julio de 2019, 2019. dirección: <https://www.imf.org/es/News/Articles/2019/05/06/mcs050619-guatemala-staff-concluding-statement-of-the-2019-article-iv-mission>.
- [13] M. Sandhya Pruthi, *Epilepsia: Síntomas y causas*, Accedido por última vez el 16 de agosto de 2019, 2019. dirección: <https://www.mayoclinic.org/es-es/diseases-conditions/epilepsy/symptoms-causes/syc-20350093>.
- [14] CDC, *Epilepsia: Tipos de convulsiones*, Accedido por última vez el 17 de agosto de 2019, 2018. dirección: <https://www.cdc.gov/epilepsy/spanish/basicos/convulsiones.html>.
- [15] O. Devinsky, «Patients with Refractory Seizures», *New England Journal of Medicine*, vol. 340, n.º 20, págs. 1565-1570, mayo de 1999. DOI: 10.1056/nejm199905203402008. dirección: <https://doi.org/10.1056/nejm199905203402008>.
- [16] J. Aicardi, «CLINICAL APPROACH TO THE MANAGEMENT OF INTRACTABLE EPILEPSY», *Developmental Medicine & Child Neurology*, vol. 30, n.º 4, págs. 429-440, nov. de 2008. DOI: 10.1111/j.1469-8749.1988.tb04769.x. dirección: <https://doi.org/10.1111/j.1469-8749.1988.tb04769.x>.
- [17] P. Kwan, A. Arzimanoglou, A. T. Berg, M. J. Brodie, W. A. Hauser, G. Mathern, S. L. Moshé, E. Perucca, S. Wiebe y J. French, «Definition of drug resistant epilepsy: Consensus proposal by the ad hoc Task Force of the ILAE Commission on Therapeutic Strategies», *Epilepsia*, vol. 51, n.º 6, págs. 1069-1077, nov. de 2009. DOI: 10.1111/j.1528-1167.2009.02397.x. dirección: <https://doi.org/10.1111/j.1528-1167.2009.02397.x>.
- [18] M. Wood, *New treatment options for people with drug-resistant epilepsy*, Accedido por última vez el 17 de agosto de 2019, 2019. dirección: <https://www.uchicagomedicine.org/forefront/neurosciences-articles/2019/may/new-treatment-options-for-people-with-drug-resistant-epilepsy>.
- [19] Livanova, *Physician's Manual VNS Therapy Programming Software*, 11.^a ed., US Version, Livanova, 100 Cyberonics Boulevard, Houston, Texas 77058, USA, jun. de 2019.

- [20] —, *VNS Therapy System Epilepsy Physician's Manual*, 1.^a ed., US Version, Livanova, 100 Cyberonics Boulevard, Houston, Texas 77058, USA, abr. de 2019.
- [21] B. Ivey, *AN1416: Low-Power Design Guide*, 1.^a ed., Microchip Technology Inc., 2355 West Chandler Blvd. Chandler, Arizona, USA 85224-6199, nov. de 2011.
- [22] A. B. Brant Ivey Rajat Hedge, *eXtreme Low-Power (XLP) PIC Microcontrollers: An Introduction to Microchip's Low-Power Devices*, 2.^a ed., Microchip Technology Inc., 2355 West Chandler Blvd. Chandler, Arizona, USA 85224-6199, nov. de 2016.

16.1. Código

Para acceder al código que implementa el Estimulador del nervio vago acceder al repositorio en el siguiente link: <https://github.com/roldolfocacacho/VNS>

16.2. Esquemáticos

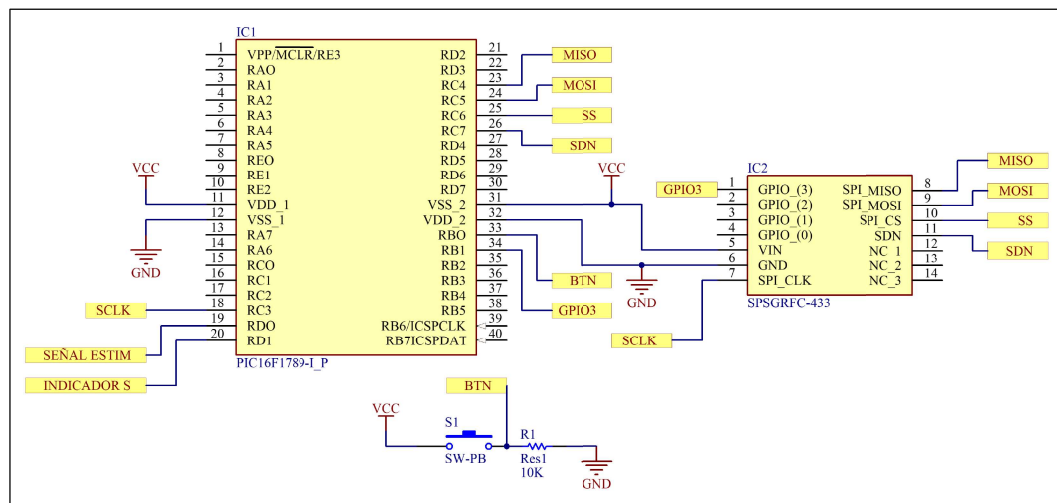


Figura 67: Esquemático prototipo 4 - Estimulador

16.2.1. Esquemáticos prototipo final

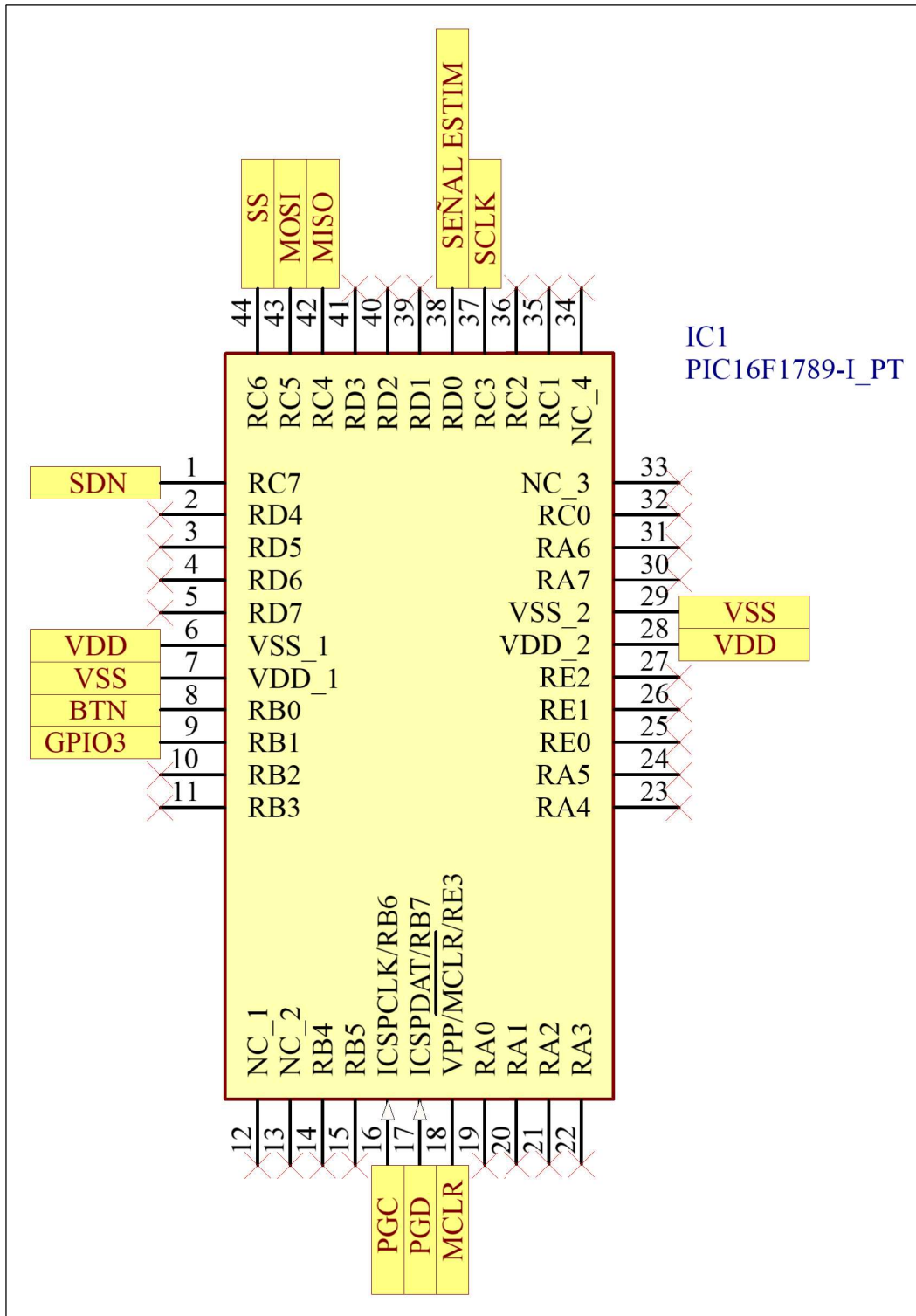


Figura 68: Esquemático prototipo final - Módulo de control

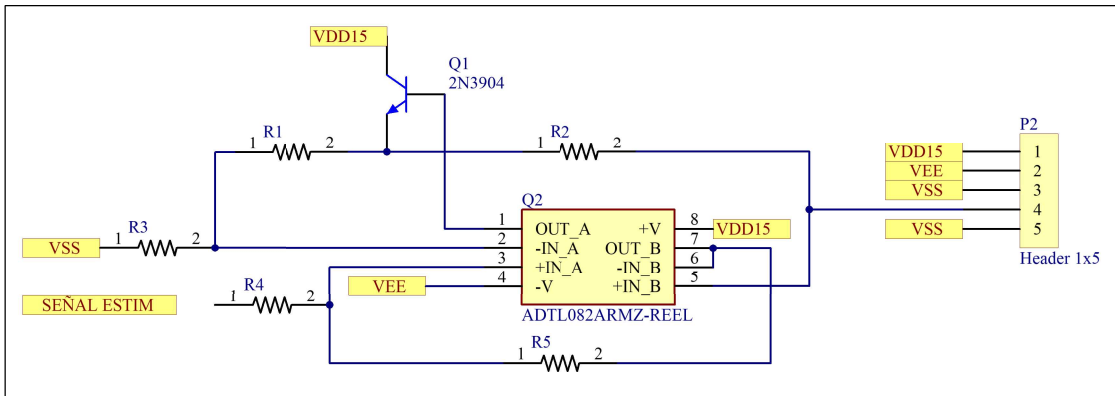


Figura 69: Esquemático prototipo final - Módulo de potencia

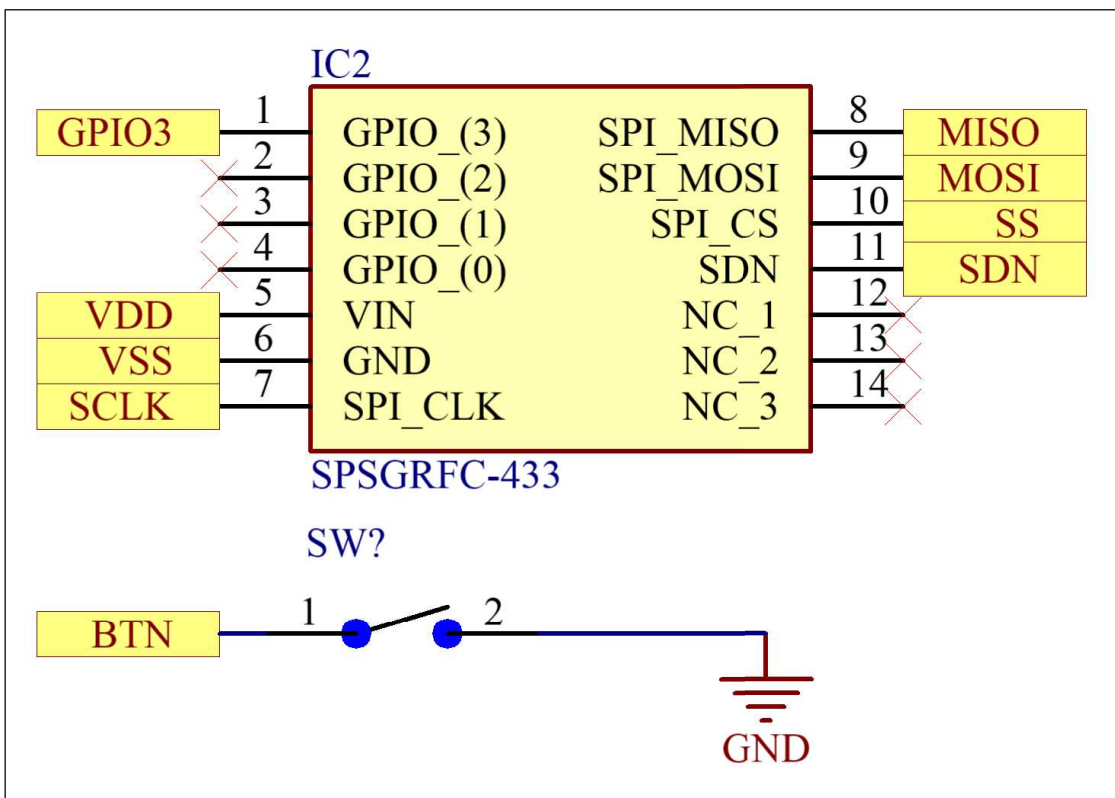


Figura 70: Esquemático prototipo final - Módulo RF

16.3. Pruebas físicas

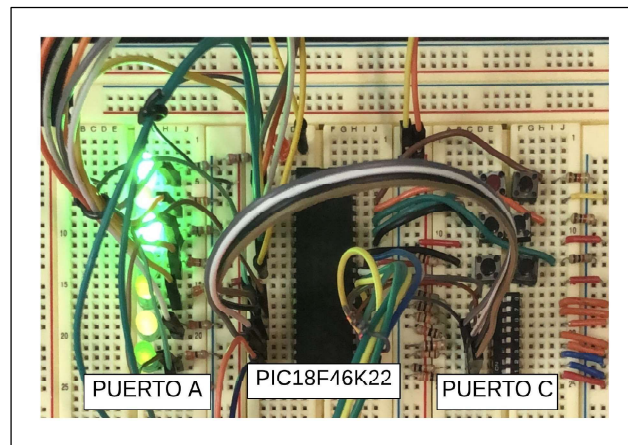


Figura 71: Conexión de validación de memoria EEPROM

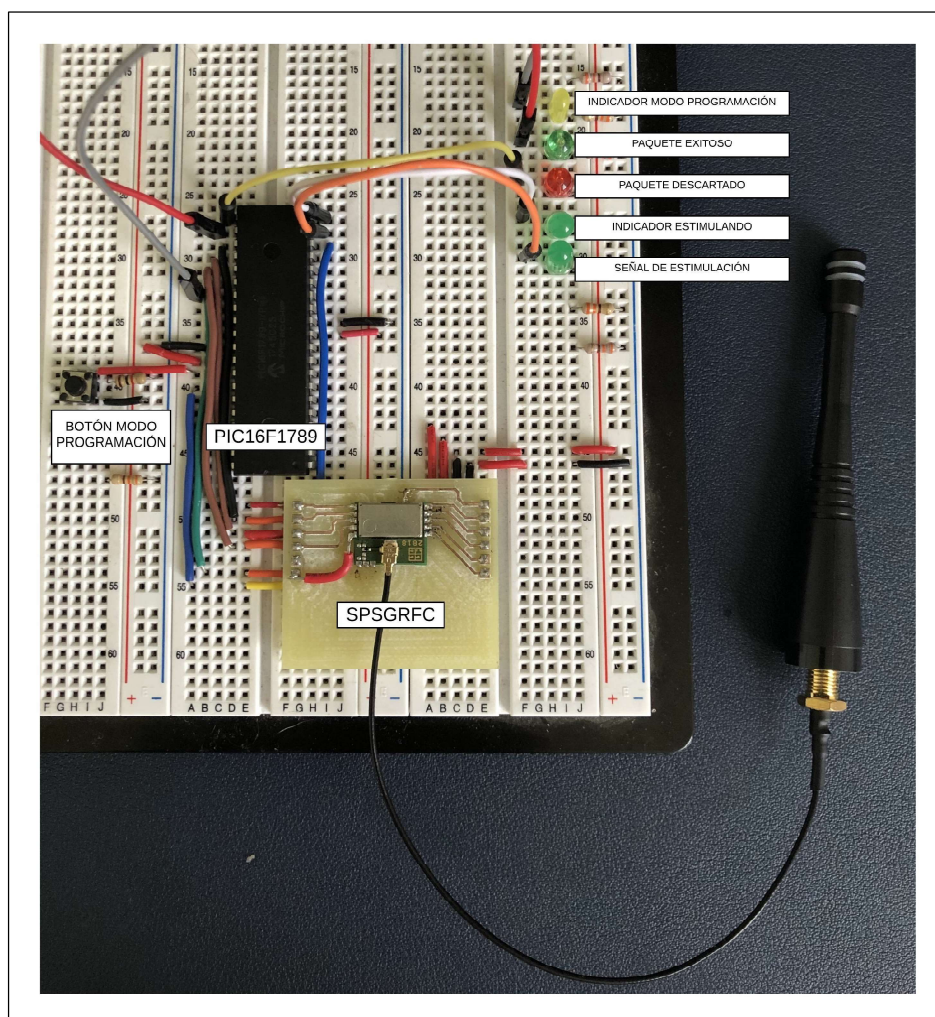


Figura 72: Prototipo 4 - Estimulador

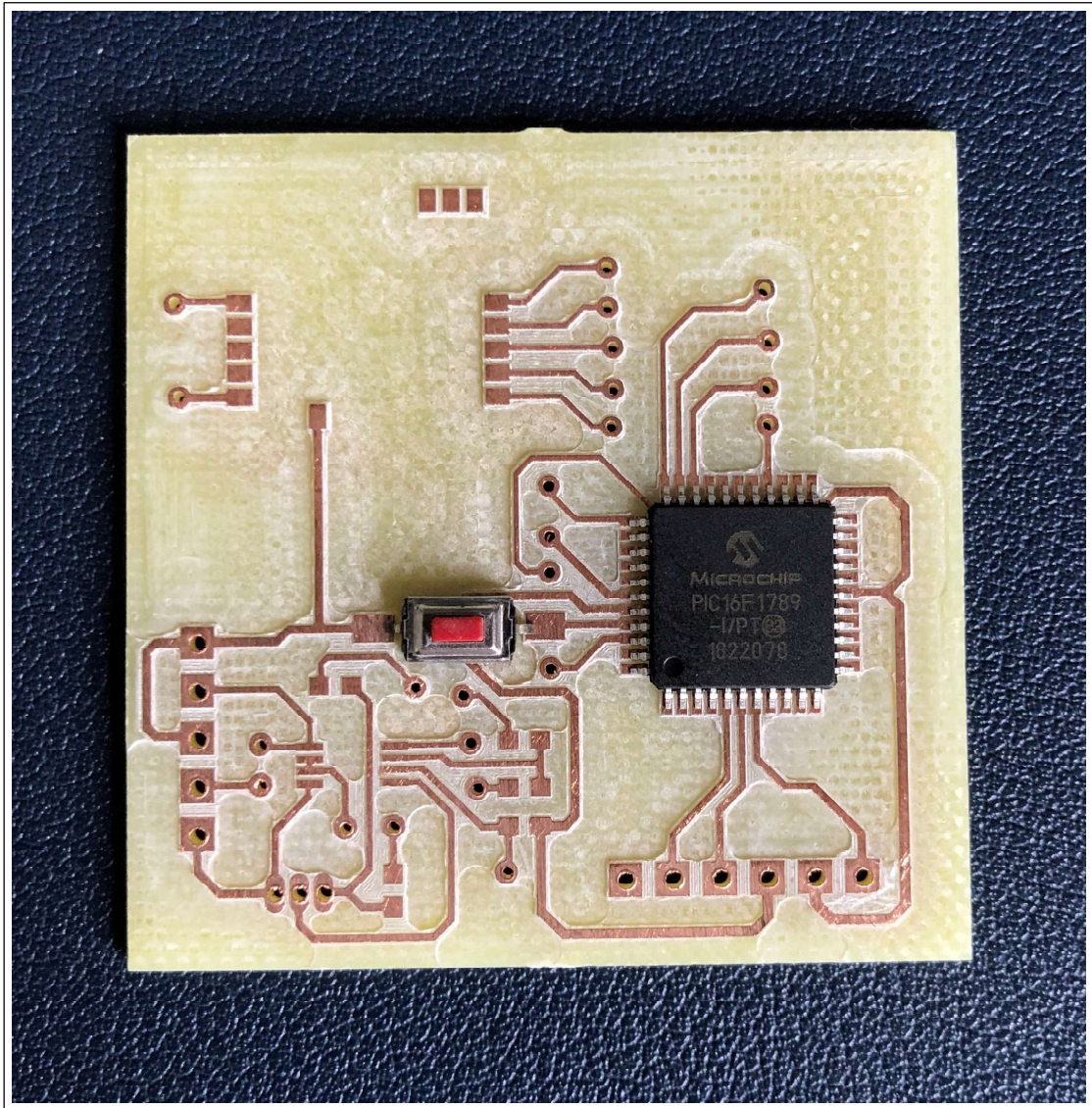


Figura 73: Prototipo final - Placa impresa previo a soldar

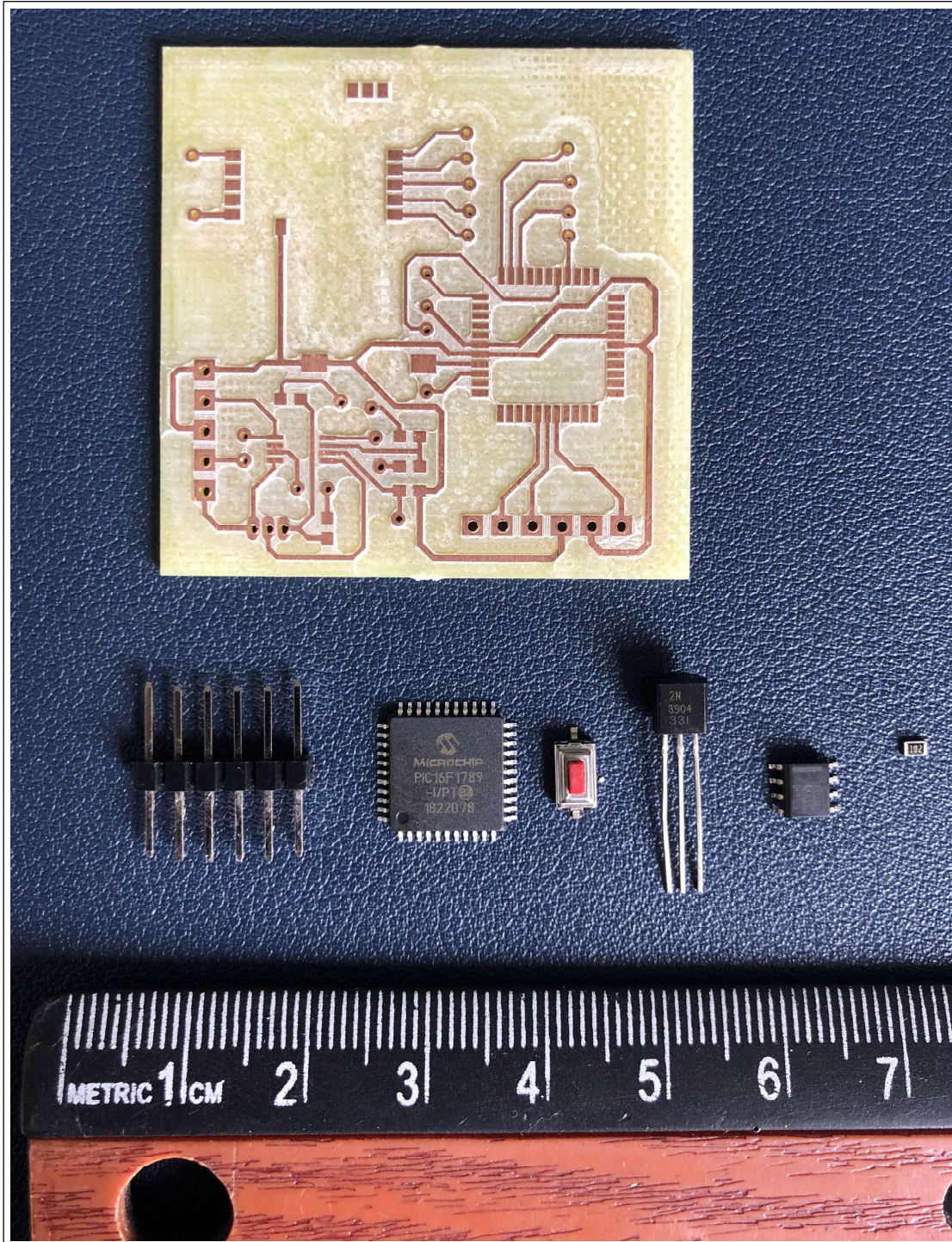


Figura 74: Prototipo final - Comparación de tamaño de componentes

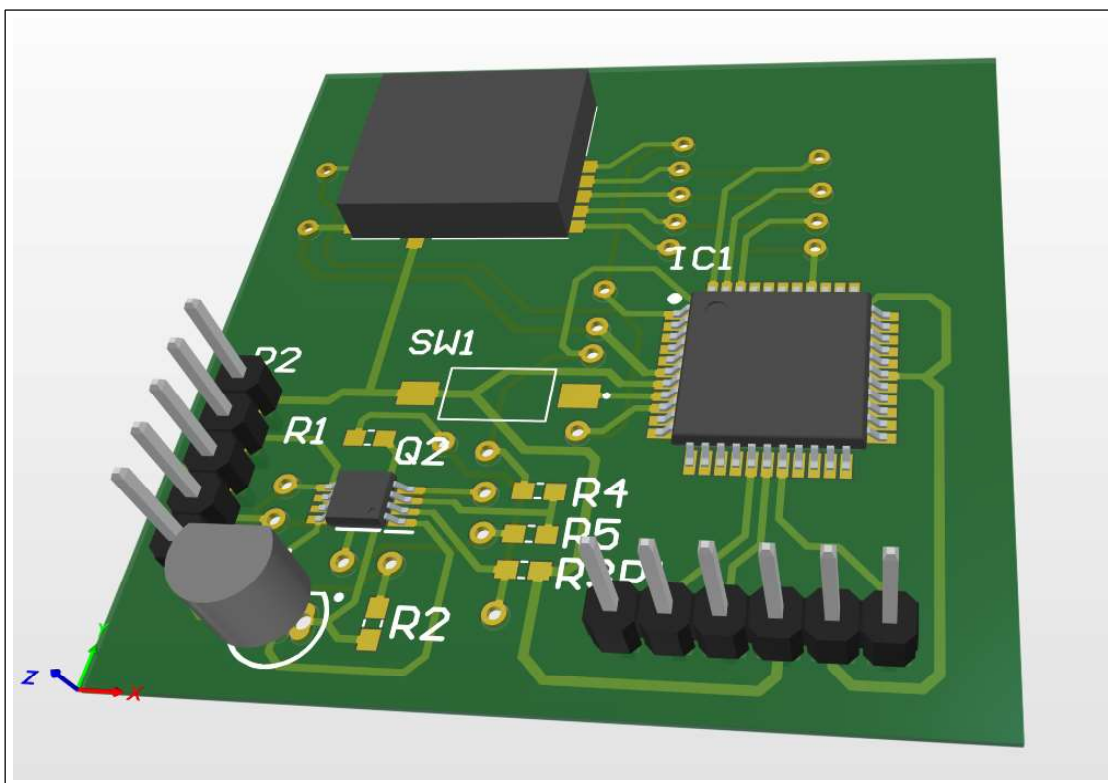


Figura 75: Prototipo final - Representación en 3D por software de diseño de PCBs

BOM Bill of Materials. 74

bug Es un problema en el software que desencadena un resultado indeseado plural. 51

CRC Cyclic Redundancy Check. Es un código para detección de errores de los datos en bruto. 67

DBS Deep Brain Stimulation. 13

EEPROM Electrically Erasable Programmable Read-only Memory. Tipo de memoria no volátil. 20, 22, 23, 33, 34, 61–63, 65

I2C Inter-Integrated Circuit. Es un estándar de comunicación serial desarrollado por Philips Semiconductors en 1982. Este es síncrono y funciona con un dispositivo como Maestro y otro como Esclavo de similar manera al SPI. A diferencia de SPI solamente utiliza dos líneas de señal, SCK y SDA. 29, 34

IC Integrated Circuit. 29

kbps kilobits per second. 67

MCC MPLAB Code Configurator. 41, 42, 47

MCU Microcontroller Unit. 20–23, 27, 32, 34–36, 39, 40, 47, 49–51, 61, 64, 65, 67–71, 78, 79, 81

MISO Master Input Slave Output. 29, 30

MOSI Master Output Slave Input. 29

PCB Printed Circuit Board. 74, 75

PDIP Empaquetado compatible con placas de pruebas, Plastic Dual in-line Package. 35–37

- PMD** Peripheral Module Disable. 35
- PWM** Modulación por ancho de pulso, en inglés Pulse Width Modulation. 22, 23, 32, 34, 37–39, 42, 49–51
- RNS** Responsive Neurostimulation. 13
- RTCC** Real Time Calendar/Clock. 28, 40
- SCLK** Serial Clock. 29, 30
- SMD** Surface-mount device. 35, 73, 79
- SPI** Serial Peripheral Interface. Es un estándar de comunicación serial síncrono que funciona con un dispositivo como Maestro y otro como Esclavo. Utiliza 4 señales para la comunicación, MISO, MOSI, SCLK y SS. 29–32, 34, 65, 67, 69–71
- SS** Slave Select. 29
- VNS** Terapia de estimulación del nervio vago, más reconocida como VNS por sus siglas en inglés, Vagus Nerve Stimulation. Esta terapia se utiliza para el el tratamiento de epilepsias no operables o imposibles de tratarpor medio de drogas antiepilépticas. 1, 3, 13, 15, 31, 50, 51, 73
- WDT** Watchdog Timer o perro guardián en español es un módulo de los microcontroladores que al expirar cierto tiempo programado, hace que el microcontrolador entre en un estado de Reset. Este módulo tiene el fin de evitar que el microcontrolador entre en un estado desconocido. 27, 28, 40, 51, 53, 68
- XC8** Compilar de MPLAB para microcontroladores de 8 bits. 61