UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Diseño e Implementación del Sistema Eléctrico y Electrónico de un Dispositivo de Limpieza para IQOS 2.4

Trabajo de graduación presentado por Christian Fernando Morales López para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2019

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Diseño e Implementación del Sistema Eléctrico y Electrónico de un Dispositivo de Limpieza para IQOS 2.4

Trabajo de graduación presentado por Christian Fernando Morales López 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 Esquit

(f) _____

Ing: Luis Pedro Montenegro

Fecha de aprobación: Guatemala, 04 de Diciembre de 2019.

Prefacio

En este espacio, quisiera agradecer a todas las personas que de alguna u otra manera fueron parte este proyecto universitario y contribuyeron a hacer posible este sueño.

Primero, **a Dios**, por nunca dejar de quererme, escucharme, guiarme e iluminarme. Gracias por regalarme tantas bendiciones a lo largo de mi vida, bendiciones que no merecí y no merezco.

Segundo, **a mis padres**, por dar todo su esfuerzo para poder brindarme las oportunidades que nunca tuvieron. Gracias por siempre querer lo mejor para mi, gracias por orientar mis pasos y por darme todo su amor. Siempre tendré muy presente todos sus consejos y nunca olvidaré todo el apoyo que me han dado.

Tercero, **a mis hermanos**, por estar siempre a mi lado. Por permitirme tener una vida que sin ustedes, no sería vida.

Cuarto, **a mis amigos de la universidad**, porque sin ustedes esta etapa de mi vida no hubiese sido tan alegre como lo fue. Gracias por compartir conmigo tantos momentos, de dificultad y de gozo. Gracias por los alientos y las motivaciones. Gracias por enseñarme que la hermandad no es solo de sangre. ¡Sí se pudo!

Quinto, **a mis catedráticos de la universidad**, por dar lo mejor de ustedes para nuestra enseñanza. Gracias porque además de ser mentores, también fueron amigos.

Sexto, **a la UVG**, por confiar en mí y darme la oportunidad de estudiar en ésta, que ahora orgullosamente llamo *alma máter*.

Séptimo, **al resto de mis seres queridos, amigos y profesores**, porque todos han tenido alguna influencia en mí. Gracias por estar pendiente de mí. Porque sin ustedes, no sería quién soy hoy.

Gracias, gracias y gracias.

Índice

Lista de figuras

Lista de cuadros

Resumen

La marca IQOS, distribuida por Marlboro en el país, tiene en el mercado distintos dispositivos electrónicos libres de humo. Estos dispositivos buscan mejorar la experiencia de los consumidores de tabaco, proveyendo una solución amigable para la salud y para el medio ambiente. En estos dispositivos se utilizan cigarrillos especiales de tabaco comprimido, más cortos y con un filtro de algodón más grande, llamados Heets. El dispositivo, a diferencia de los filtros, no necesita quemar el tabaco, solo lo calienta a tal punto que pueda disfrutarse sin que exista humo.

Existe un problema con la complejidad del proceso de limpieza propuesto. Después de cierto uso, el dispositivo necesita ser limpiado exhaustivamente para poder regresar a un estado óptimo. Las paredes internas del dispositivo y la lámina que se inserta en el cigarrillo se cubren de residuos de tabaco que hace que el proceso no se lleve de forma correcta, pues no se logra transmitir la temperatura adecuada y, por lo tanto, no se puede consumir el tabaco. La empresa provee de un kit de limpieza especial y propone un método en específico. El kit cuenta con cepillos para raspar las partes internas de los IQOS e hisopos humedecidos en un líquido removedor de residuos. La parte inicial del método es girar delicadamente el cepillo en las paredes internas del IQOS, que muchas veces daña el dispositivo si no se ejecuta de manera adecuada. El daño es tan grave que arruina por completo el dispositivo y no se puede continuar con su uso. Por tanto, se ha desarrollado una solución que puede hacerlo de manera automática y segura.

Para el diseño del dispositivo limpiador fue seleccionado el IQOS 2.4, por tener mayor presencia en los consumidores. El proyecto se separa en dos grandes partes: el sistema mecánico y el sistema eléctrico y electrónico. El fin de este proyecto fue diseñar e implementar el Sistema Eléctrico y Electrónico de un Dispositivo de Limpieza para IQOS 2.4, por lo tanto, su desarrollo se compuso de diversos subsistemas, los cuales se describirán a continuación brevemente.

El Sistema Eléctrico abarca el diseño de todos los sistemas de potencia, que son la fuente de alimentación y los módulos de potencia para motores (conocidos cómo drivers de potencia). El Sistema Electrónico se refiere al microcontrolador que llevará a cabo todas las operaciones de control lógico y será quién decida cuándo, cómo y por cuánto tiempo se ejecutará un evento. Para lo cual se decidió utilizar el PIC16F1789, que en resumen, es el

encargado del movimiento de motores, despliegue de mensajes en la pantalla, reconocimiento de la interfaz de botones. Se tienen cuatro botones, uno para un modo automático, uno para limpiar únicamente del Holder, otro para limpiar únicamente el Cap y otro para resetear el microcontrolador. Se incluyeron sistemas de control digitales para los motores, los cuales funcionan a distintos voltajes: el de la bomba requiere $13.5V$, el del Holder requiere $10V$, los motores para el Cap requieren $3V$ y el Servo-Motor a $5V$. El sistema permitirá controlar el torque indirectamente, asegurando no sobre pasar el máximo permitido, garantizando que no ocurran daños.

Abstract

The IQOS brand, distributed by Marlboro in the country, has different smoke-free electronic devices on the market. These devices seek to improve the experience of tobacco consumers, providing a friendly solution for health and the environment. In these devices special compressed tobacco cigarettes are used, shorter and with a larger cotton filter, called Heets. The device, unlike filters, does not need to burn tobacco, it only heats it to a determine temperature that it can be enjoyed without smoke.

There is a problem with the complexity of the given cleaning process. After a certain use, the device needs to be thoroughly cleaned to be able to return to an optimal state. The internal walls of the device and the sheet, that is inserted into the cigarette, are covered with tobacco residues that causes the process to not be carried out properly, so that a proper temperature cannot be transmitted and therefore tobacco cannot be consumed. The company provides a special cleaning kit and proposes a specific method. The kit has brushes to scrape the internal parts of the IQOS and swabs moistened in a special waste remover liquid. The initial part of the method is to gently rotate the brush on the internal walls of the IQOS, which often damages the device if it is not executed properly. The damage is so severe that it completely ruins the device and its use cannot be continued. Therefore, a solution has been developed that can do the whole process automatically and safely.

For the design of the cleaning device, IQOS 2.4 was selected because it has a greater presence in consumers. The project is divided into two main parts: the mechanical system and the electrical and electronic system. The purpose of this project was to design and implement the Electrical and Electronic System of a Cleaning Device for IQOS 2.4, therefore, its development was composed of various subsystems, which will be briefly described below.

The Electrical System covers the design of all power systems, which are the power source and power modules for motors (known as power drivers). The Electronic System refers to the microcontroller that will carry out all the operations of logical control and it will be who decides when, how and for how long an event will be executed. For which it was decided to use the PIC16F1789, which in short, is responsible for the movement of the engines, display of messages on the screen, recognition of the button interface. There are four buttons, one for an automatic mode, one to clean only the holder, another to only clean the cap and another to reset the microcontroller. Digital control systems were included for the motors,

which operate at different voltages: the notor of the pump requires 13.5V, the motor of the holder requires 10V, the motors for the cap require 3V and the Servo-Motor requires 5V. The system will allow to control the torque indirectly, ensuring not to exceed the maximum torque allowed, guaranteeing that no damage will occur.

CAPÍTULO 1

Introducción

Este trabajo pretende resolver una muy reciente problemática de la tabacalera con sus dispositivos electrónicos libres de humo. Dicho problema ha afectado fuertemente a sus dispositivos a tal punto de afectar el mercado.

Dichos dispositivos electrónicos han sido vendidos con éxito en la comunidad guatemalteca de fumadores que desean continuar consumiendo tabaco pero que buscan disfrutarlo de manera más saludable.

El dispositivo, como tal, no presenta ningún problema de funcionamiento o utilidad. El problema surge cuando la parte interior del dispositivo requiere ser limpiada. Esta parte es la encargada de desempeñar la tarea principal, a través una delgada laminilla que se inserta dentro del cigarrillo, lo calienta al mantener presionado un botón.

Después de cierta cantidad de usos, los cigarrillos dejan residuos que forman una pequeña capa de sarro que se adhiere a las paredes interiores del dispositivo electrónico y también a la laminilla. Esto limita la funcionalidad del dispositivo pues no logra calentar adecuadamente el tabaco y por lo tanto éste no se logra disfrutar.

Cuando los usuarios identifican el problema, pues el sabor del cigarrillo cambia, saben que deben limpiar el dispositivo. La tabacalera ofrece un kit de limpieza y propone un método que puede llegar a ser un tanto tedioso y complejo para los usuarios. Quienes no guardan el cuidado suficiente, han llegado a dañar el dispositivo de manera severa. La laminilla es bastante delicada y si se aplica una fuerza excesiva sobre ella, se quiebra. Esto es un problema grave para los usuarios pues el dispositivo queda totalmente disfuncional.

El caso ha sido tan recurrente que también se ha vuelto un problema para la empresa distribuidora. Es por esto que buscan con necesidad un sistema que pueda solventar esta problemática. De manera que puedan ofrecer el servicio de limpieza en sus puntos de venta, en donde también venden los cigarrillos especiales para el dispositivo, por lo que los clientes fácilmente pueden optar por este nuevo servicio.

CAPÍTULO 2

Antecedentes

En el proceso de investigación e indagación, previo a comenzar la etapa de ideación y diseño, se encontró un único sistema de limpieza pseudo-automática (desarrollado por un tercero) para los dispositivos IQOS. En realidad, son varios los desarrolladores y distribuidores pero la idea es básicamente la misma. En esencia es un dispositivo que por medio de un motor, hace girar un cepillo que limpia la parte interna del IQOS al ser insertado en él. Normalmente estos limpiadores son recargables y se le pueden cambiar los cepillos.

Hace falta mencionar que estos limpiadores no logran cubrir todo el proceso de limpieza requerido, por lo que no se logra automatizar por completo. Implicando que los usuarios tengan que terminar el proceso manualmente, que sería la limpieza con los hisopos humedecidos en alcohol isopropílico, que suele ser utilizado en la limpieza de dispositivos electrónicos.

El más conocido de estos limpiadores automáticos es el *Vapesoon* desarrollado por *DHGate*, diseñado especialmente para el IQOS 2.4 Plus.

Figura 1: Uso de Vapesoon. [1]

CAPÍTULO 3

Justificación

La empresa distribuidora de la marca se acercó a la Universidad para discutir su problemática, sin embargo no proporcionó ningún tipo de dato relacionado con la cantidad o porcentaje de dispositivos dañados. No obstante, es su deseo encontrar una propuesta de solución a través del Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica de esta casa de estudios.

Es importante tanto para la tabacalera como para los consumidores que este problema se resuelva, puesto que un método de limpieza es siempre necesario para continuar con el uso óptimo del dispositivo. La idea es que con el dispositivo de limpieza los usuarios no tengan que preocuparse por tener que hacer un proceso exhaustivo y que además, pueda dañar su dispositivo.

Conociendo la problemática con el actual sistema de limpieza, dada por su complejidad y excesivo cuidado; es necesario desarrollar una solución que cumpla con el proceso requerido de limpieza, que sea automático, que no realice ningún tipo de daño al dispositivo y que sea eficiente en términos de tiempo. Teniendo en cuenta el papel tan determinante que juega la limpieza del dispositivo para con los usuarios y vendedores, se recalca no solo la importancia sino la urgencia que conlleva este trabajo de graduación.

Como estudiante de Ingeniería Electrónica, el proyecto permite una conexión directa con muchas de las áreas estudiadas en mi carrera, cómo: Circuitos Eléctricos, Electrónica Analógica, Electrónica Digital, Diseño de Circuitos Impresos y Sistemas de Control; que es una de las áreas más importantes de mi carrera. De esta manera, no solo tendré un acercamiento más profundo en el tema, sino que podré involucrarme en un caso que requiere de una solución real. La experiencia que ofrece el proyecto es altamente valorada para mi futuro desempeño profesional.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Diseñar e implementar el sistema eléctrico y electrónico del dispositivo de limpieza automática para el modelo de segunda generación, IQOS 2.4.

4.2. Objetivos específicos

- Diseñar un proceso automático que cumpla y se adecúe a los requerimientos de cero daños y que sea eficiente en términos de tiempo.
- Seleccionar los componentes eléctricos, electrónicos y actuadores para el control lógico del sistema.
- Desarrollar módulos de potencia para los actuadores y seleccionar la fuente de alimentación que cubra el consumo de corriente máxima.
- Implementar sistemas de control de torque a los motores DC para asegurar que no se rompa la lámina interna.
- Implementar una interfaz amigable para el usuario final a través de un sistema de botones y pantalla LCD para mensajes interactivos.

CAPÍTULO 5

Alcance

En este trabajo se busca diseñar e implementar el sistema eléctrico y electrónico del aparato de limpieza, para lograr un prototipo funcional que controle dicho aparato. Se busca que los circuitos diseñados cumplan adecuadamente su propósito y que no exista riesgo de mal funcionamiento ante las condiciones apropiadas.

Por tanto, el sistema eléctrico y electrónico podrá controlar tres motores DC y un servomotor. La fuente de alimentación tendrá la capacidad de proveer la potencia suficiente para asegurar el correcto funcionamiento de todos los sistemas en simultáneo. La pantalla deberá desplegar los mensajes pertinentes para que el usuario final pueda conocer el estatus actual de la máquina y pueda decidir entre los distintos modos de funcionamiento. Aunado a lo anterior, se contará con tres botones de control, donde cada uno será un modo distinto de funcionamiento. Se tendrá un cuarto botón para poder resetear el microcontrolador en caso se produzca una falla en la lógica, ya sea por fuentes de error humanas o eléctricas.

Se crearán tres rutinas distintas para la limpieza de las piezas del IQOS 2.4, cada una será ejecutada según el botón correspondiente. Una rutina deberá hacer el proceso de limpieza completo, las otras dos deberán hacer el proceso de limpieza por separado (Holder o Cap). Se bloquearán el resto de funciones para disminuir la posibilidad de error humano.

Rebasa del alcance e interés de este proyecto, asegurar un buen acople y funcionamiento con el sistema mecánico (que se desarrollará en paralelo por otro estudiante). Sin embargo, se tomarán las consideraciones pertinentes en los diseños para facilitar la integración a un prototipo final, que pueda combinar la circuitería con los mecanismos.

CAPÍTULO 6

Marco teórico

6.1. Dispositivos IQOS

Los IQOS son dispositivos electrónicos libres de humo desarrollados por la empresa : Phillip Morris International (PMI). El fin de estos dispositivos no es ayudar a los consumidores de tabaco a que dejen de consumirlo, sino al contrario, mejorar la experiencia de consumo. Con este dispositivo se elimina el humo que se produce en un cigarro normal, pues no se quema el tabaco sino que se calienta, gracias a la tecnología HeatControl™. Debido a esto, requieren de un nuevo cigarrillo de tabaco comprimido al cual se le inserta una lámina que lo calienta a 350°C sin quemarlo y que puede consumirse [2].

Dichos dispositivos ya se han ubicado bien en el mercado del país, por lo que se han ido desarrollando distintos modelos con distintas funcionalidades. Según el distribuidor, en Guatemala se comercializan principalmente tres modelos: IQOS 2.4 Plus, IQOS 3 y el IQOS 3 Multi.

(a) IQOS 2.4 Plus

(b) IQOS 3

(c) IQOS 3 Multi

Figura 2: Modelos de IQOS. [3]

6.1.1. Modelo IQOS 2.4 Plus

Este es el modelo menos reciente de los mencionados anteriormente, sin embargo, es el modelo con mayor presencia en los usuarios. Por esto, se ha elegido este modelo para el diseño del dispositivo limpiador. No se puede trabajar bajo un modelo genérico porque todos presentan estructuras físicas con diferencias significativas entre sí, como fue posible observar en la Figura 2. La Figura 3 expone a *grosso modo* como se compone el IQOS 2.4 Plus. De esta manera se puede tener una mejor idea del funcionamiento interno y general. Además, permite enfatizar la necesidad de un cuidado apropiado en el método de limpieza.

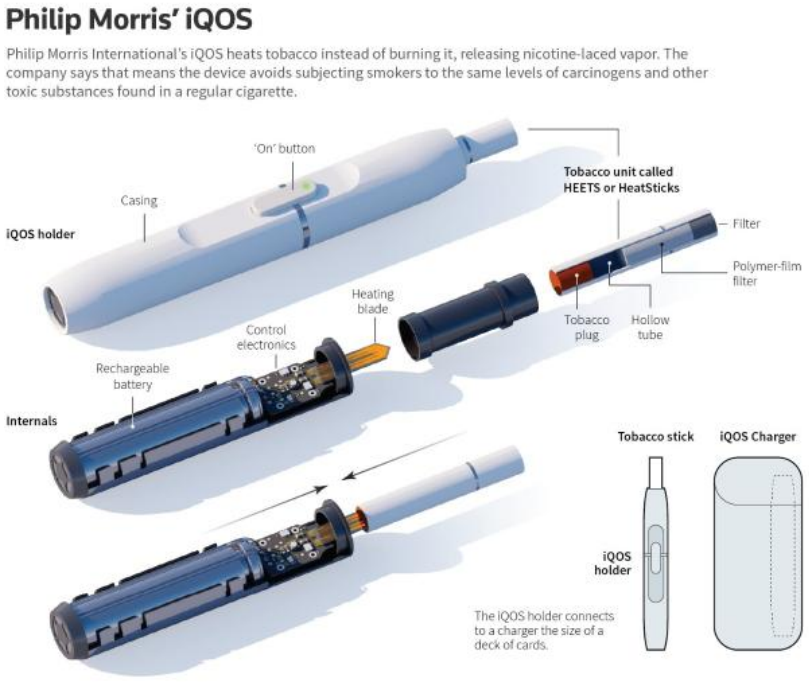


Figura 3: IQOS 2.4 Plus. [4]

El IQOS 2.4 Plus se compone esencialmente de tres partes: un Pocket Charger, un Holder y Heets. Basta con presionar el botón de encendido del Holder para calentar el Heet y comenzar a consumir. Al terminar, se devuelve el Holder al Pocket Charger para que se cargue y pueda volver a ser a utilizado. El kit también incluye cargador y la herramienta limpiadora, ver figura 4. Todos los dispositivos IQOS tienen una lámina que se inserta en el Heet luego de que este es introducido en el Holder, dicha lámina es la que suele dañarse o quebrarse por completo si el proceso de limpieza no se realiza con suficiente cuidado y con la técnica apropiada.

Figura 4: IQOS 2.4 Plus Kit. [5]

6.1.2. Método de limpieza

Como es de esperarse, después de cierto tiempo de uso, es necesario limpiar el IQOS. Sobre todo, es imprescindible limpiar el interior del Holder (donde se ubica la lámina que calienta) y el cabezal, coloquialmente denominado «*Cap*». Estos tres elementos deben ser limpiados exhaustivamente para eliminar el residuo de tabaco que se impregna en las paredes interiores. Como se ha mencionado anteriormente, se provee de un kit de limpieza. El método propuesto es completamente manual y depende mucho de la habilidad de los usuarios.

A continuación se hará una breve descripción del método usual que la mayoría de usuarios implementa.

1. Retirar el *cap* del *Holder*, para tener acceso a los tres sectores que se ensucian más: interior del *Holder*, e interiores del Cap (dos lados).

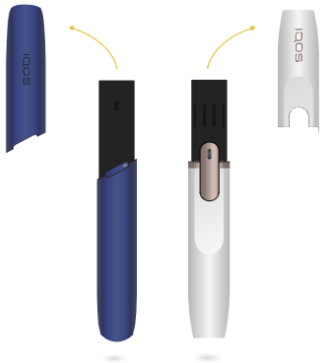


Figura 5: Holder IQOS 2.4 Plus. [6]

2. Uso de la herramienta limpiadora, que debe girarse en ambos sentidos para poder retirar bien los residuos que se adhieren a las paredes interiores y a la lámina. También suele hacerse en el Cap.

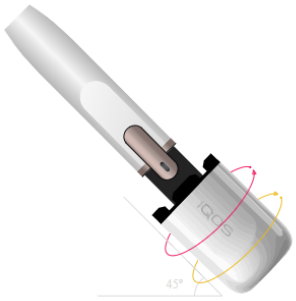
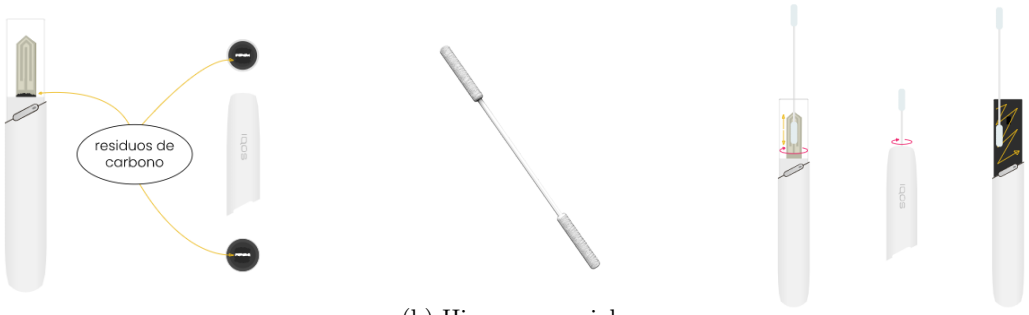


Figura 6: Herramienta limpiadora de IQOS 2.4 Plus. [6]

3. Limpieza de residuos con hisopo especial (7c). Hay que retirar las bases que se colocan para protección de la circuiteria (7a) y luego hacer la limpieza con los bastoncillos (7b). Esto último puede repetirse hasta que los bastoncillos salgan completamente limpios. También pueden usarse en el Cap.



(a) Retirar base

(b) Hisopos especiales

(c) Limpieza interior

Figura 7: Limpieza interna [6]

6.2. Herramientas de diseño

6.2.1. Diseño de circuitos impresos en Altium Designer 19

: Altium Designer (AD) es un software desarrollado por Altium LLC para el diseño y simulación de circuitos eléctricos, electrónicos y el diseño de circuitos impresos; PCBs. El software permite realizar esquemáticos de los circuitos con componentes ideales o con modelos reales (basados en componentes específicos). Así también, permite efectuar diversos análisis para la verificación del funcionamiento requerido y apropiado de los circuitos y componentes. Además, después de haber finalizado el diseño de los circuitos, se pueden obtener vistas 2D y 3D del PCB final para tener un acercamiento a cómo se verá en la realidad [7].

Cabe mencionar, que AD no solo cuenta con librerías genéricas de los componentes más utilizados, sino que permite trabajar con componentes específicos. Para esto existen tres

formas distintas, la más común es descargar las librerías previamente creadas, ya se por los proveedores o por otros usuarios, e instalarlas en el software.

La segunda forma más utilizada ha sido crear las librerías de los componentes requeridos, entiéndase: diseñar el modelo 2D, diseñar el footprint, diseñar el modelo 3D (o e su defecto buscarlos y mediante el Wizard generar la librería). Para completar este tipo de librerías, es necesario relacionar el modelo generado con el código del componente para así obtener todas sus características (las mencionadas en el Datasheet). Son los principales proveedores de componentes quienes generan este código para que pueda relacionarse directamente con modelos de Altium (i.e. DigiKey o Mouser Electronics).

El otro método de trabajar con componentes específicos, es hacer uso del *: Manufacter Part Search (MPS)*, herramienta que Altium ha introducido en las últimas versiones de su software. Este Wizard permite buscar específicamente el componente deseado y con las características requeridas. Facilita y hace precisa la búsqueda de componentes y asegura que se trabaje con lo que se utilizará en la realidad. Y como añadidura, incluye los precios para poder hacer al final, un análisis de costos o bien una cotización.

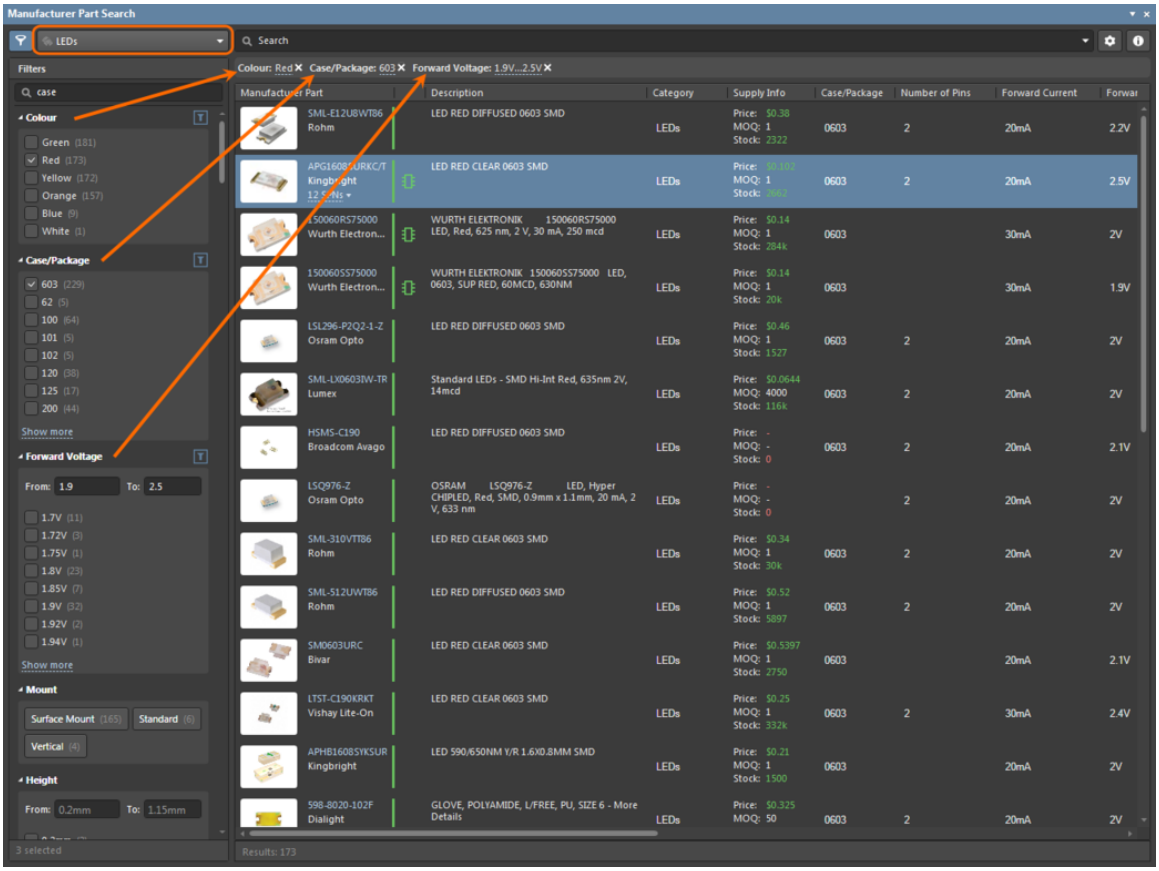


Figura 8: Manufacturer Part Search. [8]

Se ha decidido usar este programa puesto que es uno de los más utilizados a nivel profesional, empresas como: Dell, Microsoft y la NASA lo utilizan día a día. Como ya se mencionó, se empleará la versión más reciente del software, gracias a la licencia que posee

la Universidad.

6.2.2. MPLAB X Integrated Development Environment (IDE)

Este es el software por defecto utilizado para configurar y desarrollar el código de los microcontroladores, es ofrecido gratuitamente por la compañía Microchip Technology Inc., misma que fabrica los microcontroladores de la familia *PIC*. Es ideal para el desarrollo de sistemas integrados con aplicaciones que incluyan los microcontroladores de Microchip [9]. El interés de este tipo de aplicaciones es relacionar el poder computacional de un microcontrolador con otros circuitos, para lograr un control digital de bajo costo.

Algunas características esenciales que *MPLAB X IDE* otorga son:

1. Crear el diseño de alto nivel.
2. Compilar, ensamblar y relacionar el programa de la computadora al código de máquina del PIC.
3. Probar el código. El depurador (debugger) permite evaluar el código fuente.
4. «Quemar» el código en el microcontrolador y verificar que se ejecute correctamente.

MPLAB XC8

Para poder hacer el diseño en lenguaje *C* se requiere de un compilador especial. Para esto, Microchip provee el compilador XC8, con soporte para todos sus PIC de 8 bits, correspondiente al que será utilizado.

Al hacer uso de este compilador en combinación con *MPLAB X IDE*, la interfaz (totalmente gráfica) proporciona [10]:

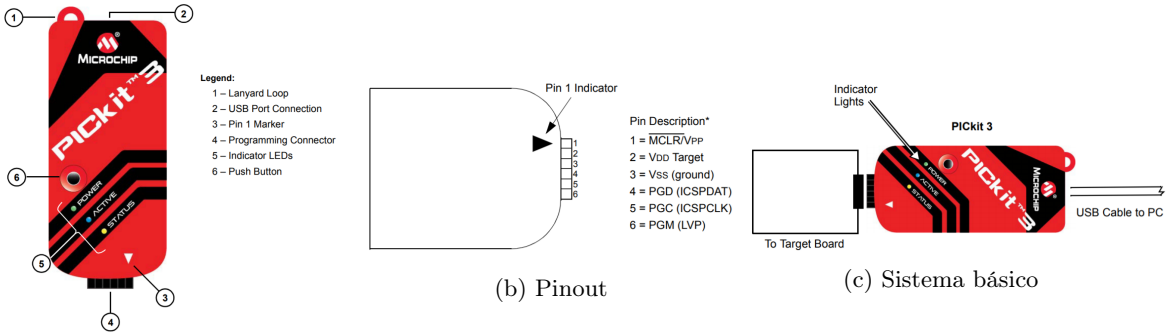
- Edición de errores (según líneas correspondientes en el código fuente).
- Paso simple a través del código fuente de *C* (inspección de variables y estructuras).
- Estructuras de datos con tipos de datos definidos.

Además, el compilador posee optimizaciones para reducir el tamaño del código y mejoras de velocidad, que se ve reflejado directamente en el proyecto en desarrollo.

PICkit™ 3 In-Circuit Debugger

Es la herramienta de desarrollo utilizada para la programación de los microcontroladores de Microchip, está diseñado para trabajar especialmente con *MPLAB X IDE* (siempre y cuando el programa corra sobre una computadora con Windows). El uso de esta herramienta es orientado a desarrollo y provee de muchas características que son muy aprovechables [11], como:

- Compatibilidad los controladores USB de Windows para gran velocidad
- Ejecución en tiempo real
- Monitoreo de cortocircuito
- Baja tensión a 5V
- LEDs de diagnóstico (potencia, activo, estado)
- Lectura, escritura y memoria de datos del microcontrolador



(a) PICkit 3

(b) Pinout

(c) Sistema básico

Figura 9: Sobre las conexiones del PICkit 3. [11]

En la figura anterior (9), se observa la composición básica de un Debugger, en este caso se muestra el *PICkit 3* porque es el elegido para programar el PIC. En la Figura 9b se observa el diagrama de pines, los cuales existen en todos los PIC, aunque no siempre se encuentran en las mismas posiciones. Es fácil notar que el PICkit 3 ofrece una solución amigable para el desarrollo de sistemas integrados con microcontroladores de Microchip. Se ha decidido utilizar este programador por la disponibilidad que tienen en el país pero sobre todo por la compatibilidad con el microcontrolador elegido.

MPLAB Code Configurator (MCC)

El MCC genera código de controlador a través de una interfaz gráfica. De esta forma, proporciona un medio amigable para la configuración de los periféricos del microcontrolador. La ventaja es que es un Plugin a *MPLAB X IDE*. También se utiliza para configurar y generar librerías que pueden ser utilizadas en cualquier PIC.

Para poder usarse, se debe crear primero un proyecto en *MPLAB X IDE*, puesto que el MCC necesita conocer específicamente que dispositivo está siendo utilizado en el proyecto. Esto último para poder tener acceso a la información específica del dispositivo, como: registros, bits y configuraciones, y para configurar la interfaz gráfica. El *MCC* genera archivos fuente y encabezados basados en las selecciones hechas en la interfaz gráfica. Dichos archivos son agregados al proyecto [12].

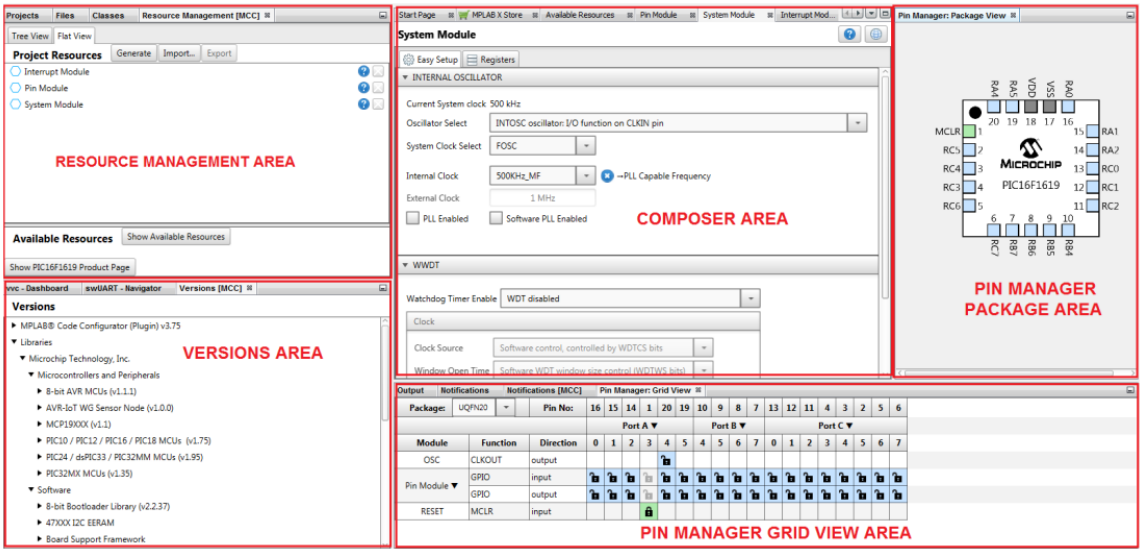


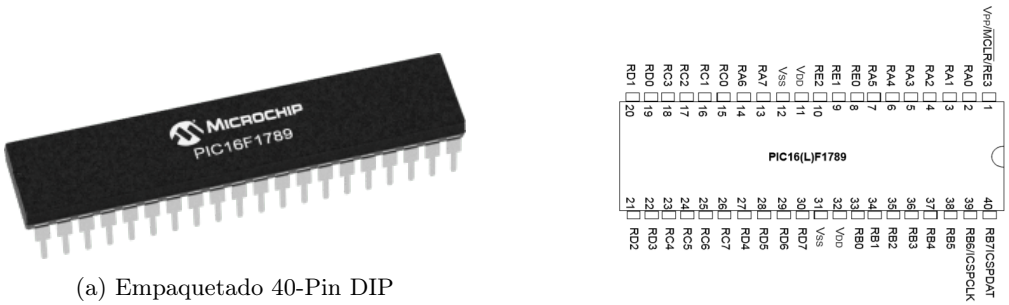
Figura 10: Áreas de operación del MCC. [12]

En la Figura 10, se tiene una captura de pantalla de la interfaz gráfica del *MCC*. Se observa el área de administración de recursos (lo que actualmente se ha configurado), el área de versiones (configuraciones disponibles), el área de configuración (donde se hace la configuración de un módulo en específico) y dos áreas de administración de pines (una basada en el empaquetado y la otra por puertos).

6.3. Microcontrolador PIC16F1789

El PIC16F1789 es un microcontrolador de la familia Microchip que posee un *CPU* con arquitectura *RISC* de alto rendimiento. Funciona con solamente 49 instrucciones, puede operar hasta una frecuencia de 32 MHz, tiene interrupciones y muchas otras cosas [13].

Ha sido seleccionado para el desarrollo de este proyecto ya que cuenta con las características suficientes para cumplir con las necesidades del sistema planteado. En la Figura 11a se observa el empaquetado que se utilizará, que es un empaquetado de 40 pines en versión «through-hole», lo que facilita las pruebas y la programación. En la Figura 11b se observa el diagrama de puertos que pueden ser utilizados como salidas y/o entradas digitales/analógicas.



(a) Empaquetado 40-Pin DIP

(b) Diagrama de Pines

Figura 11: PIC16F1789. [13]

6.3.1. Conversor DAC

Los convertidores digital-analógico (DAC por sus siglas en ingles, Digital to Analog Converter) se encargan de convertir una señal digital (dada en bits) a una señal analógica (dada en voltaje). La resolución de la señal analógica dependerá de la cantidad de bits que posea la señal digital. En la Figura 12 se puede apreciar un esquema general de un DAC. El *PIC16F1789* posee DACs que suministran una referencia de voltaje variable, radiométrica con la fuente de entrada. Los niveles de salida seleccionables dependerán de la resolución, en este caso el microcontrolador posee un DAC de 8-bits y otros tres de 5-bits. Lo que nos daría 256 (2^8) valores distintos para el primer DAC y 32 (2^5) para el resto [13].

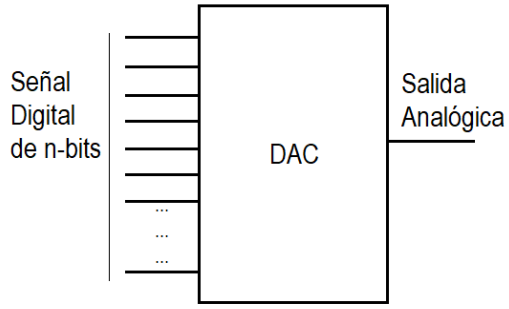


Figura 12: Esquema general de un DAC.

6.3.2. Modulación PWM

La modulación por ancho de pulso o : Pulse-Width Modulation (PWM), es una técnica muy común, utilizada para controlar el ciclo de trabajo y el período de una señal cuadrada. Es un esquema que provee potencia a una carga al cambiar rápidamente los estados de encendido o apagado, similarmente a cómo puede observarse en la Figura 13. Uno de sus usos más frecuentes en ingeniería es el control de motores, esencialmente se utiliza para controlar la posición en Servo-Motores. Puede ser utilizado en muchas otras aplicaciones como: control de intensidad en luces LED u optimización del control de ventiladores.

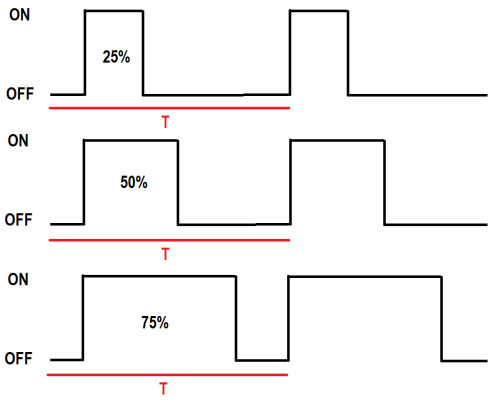


Figura 13: Diferentes señales PWM.

El *PIC16F1789* posee un controlador PWM de alto rendimiento y con una resolución de 10 bits, esto quiere decir que es posible seleccionar 2^{10} (256) valores distintos de voltaje [13]. Esta resolución es muy aprovechable para el desarrollo de este proyecto, que requiere controlar, de manera precisa, la posición de un Servo-Motor. Usualmente la frecuencia del PWM para controlar un Servo-Motor es de 50Hz, la posición angular se controla con el ancho del pulso, que varía aproximadamente de 0.5 ms (0°) a 2.5 ms (180°).

6.4. Driver de potencia

Un Driver de potencia es necesario cuando un circuito o componente no tiene la capacidad para suministrar la potencia (corriente y/o voltaje) que requiere otro componente o circuito, usualmente son necesarios cuando la etapa de control lógico no puede proveer dichos requerimientos. En la ecuación 1, se aprecia cómo la potencia se relaciona directamente con la corriente y el voltaje (tomar en cuenta que solamente aplica para cargas lineales o que pueden ser modeladas como sistemas lineales).

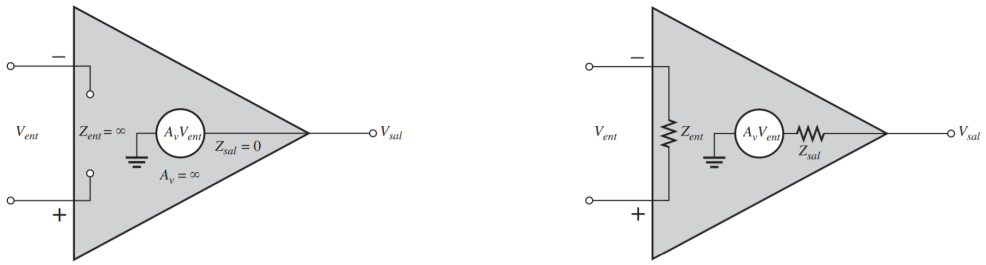
$$P = I^2V \tag{1}$$

En este caso, se requiere de un *Driver de potencia para motores DC*, dado que el microcontrolador seleccionado no puede proveer la corriente que requieren los motores. Esta etapa de potencia desempeña un papel imprescindible, sin él, se corre un riesgo muy alto de dañar toda la circuitería.

Cuando a un componente o circuito se le pide más corriente de la que este puede suministrar, él intentará darla, pero cómo le es imposible, ocurrirá un corto circuito. El corto circuito quemará a su paso varios componentes, principalmente el microcontrolador. Además, debido al gran incremento de temperatura, tanto los componentes cómo el material cercano a la circuitería, tienen una alta probabilidad de prender fuego. Por lo tanto, es imperativo no correr este riesgo, pues puede implicar perjuicios a los usuarios finales.

6.4.1. Op-Amp

Un Amplificador Operacional o : Operational Amplifier (Op-Amp), es un circuito integrado lineal que tiene como fin primordial amplificar una señal. Sin embargo, sus aplicaciones y usos van mucho más allá de ello, para lo cual hace falta adentrarse en la teoría del Op-Amp Ideal y Op-Amp Práctico. En la Figura 14 se observan los dos modelos mencionados. Cada uno representado por el símbolo del Op-Amp, con sus dos entradas: la inversora (-) y la no inversora (+), de las cuales se obtiene el voltaje de entrada V_{ent}, que es la diferencia. En la Figura 14a se puede observar el modelo básico del Op-Amp Ideal, que posee impedancia de entrada infinita, ganancia infinita e impedancia de salida cero. Esto hace que la señal de entrada se aproveche en un 100 % y que puede amplificarse hasta infinito. Por otra parte, en la Figura 14b se observa el modelo del Op-Amp Práctico, que a diferencia del Ideal, no posee ganancia ni impedancia infinita, pero sí muy altas, tampoco impedancia cero de salida pero si muy baja. Esto se debe a que en la manufactura, ningún dispositivo es ideal. Sin embargo, estas características suelen ser suficientemente aceptables como para que en los cálculos se consideren como ideales. Normalmente, la ganancia se ve limitada al voltaje de alimentación del Op-Amp [14].



(a) Op-Amp ideal

(b) Op-Amp práctico

Figura 14: Op-Amp. [14]

Retroalimentación negativa

Una de las aplicaciones más usuales de los Op-Amp es la retroalimentación negativa, esta se usa principalmente para mantener el voltaje de entrada en la salida. Esto hace que la salida no sea perturbada tan fácilmente por alguna otra señal de voltaje, sin ella, incluso el voltaje más pequeño podría llevar a Saturación el Op-Amp. La Figura 15 muestra un esquema básico de esta conexión. En el bloque de «Circuito de realimentación negativa» puede ir prácticamente cualquier circuito. Se le llama negativa ya que la retroalimentación se conecta a la entrada inversora del Op-Amp. Esta característica será de mucho provecho para el diseño del Driver de potencia.

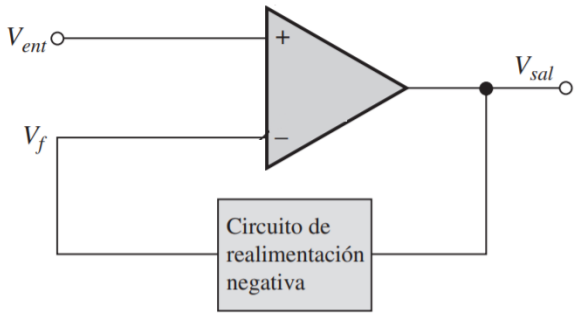


Figura 15: Op-Amp en retroalimentación negativa. [14]

6.4.2. Transistores TIP

Los transistores de la tecnología : Texas Instruments Power (TIP) son transistores BJT para aplicaciones de potencia, su característica principal es que pueden suministrar una cantidad considerable de mili-amperios. Para el Driver de potencia se utilizará la familia TIP12X, que contiene transistores darlington, con diseño especial para aplicaciones de amplificación. Son capaces de suministrar hasta 5,000 mA y disipar hasta 65 W (a temperatura ambiente, 25°C). Normalmente utilizan un empaquetado TO-220, lo que es útil para poder utilizar disipadores de calor [15]. En la Figura 16 se observa el diagrama de pines para esta serie de transistores, con el empaquetado mencionado.

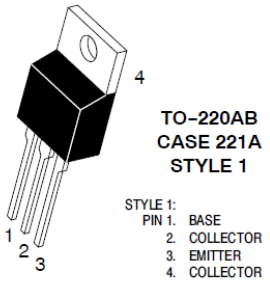


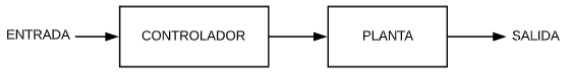
Figura 16: TIP12X PINOUT. [15]

6.5. Sistemas de control

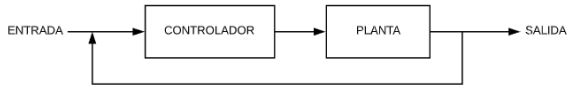
Los Sistemas de Control son utilizados para poder controlar procesos ya existentes en los cuales su respuesta no es la adecuada, donde en la mayoría de casos, estos procesos no son modificables. El Sistema de Control busca hacer que el proceso alcance la respuesta deseada a través de la modificación de una característica en especial, a definirse según los análisis matemáticos y conveniencia (viabilidad de implementación). Dichos análisis matemáticos se basan en las transformadas de Laplace, análisis vectorial y matricial.

Los Sistemas de Control tienen una gran división, están los *Sistemas de Control de lazo abierto* y los *Sistemas de Control de lazo cerrado*. Los Sistemas de Control de lazo abierto

normalmente no son lo suficientemente robustos y no aseguran estabilidad ante ningún tipo de perturbación. Por contraparte, un Sistema de Control de lazo cerrado si asegura estas dos cosas, características muy deseables para cualquier Sistema de Control. Esto es posible ya que este tipo de sistemas tienen una retroalimentación, la cuál es analizada y corregida para mejorar la salida. A diferencia del otro, que la salida depende solamente de la entrada. En la Figura 21 se observan los diagramas de bloques de los dos tipos de Sistemas de Control. Al sistema a controlar (circuito o componente), se le denomina planta. Los diagramas de bloques son la representación más común de los sistemas de control.



(a) Sistema de Control de lazo abierto



(b) Sistema de Control de lazo cerrado

Figura 17: Tipos básicos de Sistemas de Control.

Para esta aplicación, solamente interesan los Sistemas de Control con una salida y una entrada, conocidos como sistemas SISO. Debido a esto, no es necesario aplicar la Teoría de Control Moderna.

6.5.1. Teoría de control clásica

En los circuitos eléctricos y electrónicos, existen distintos controladores y compensadores. Cada una de ellos obedece a una teoría previamente desarrollada. Se pueden definir dos grandes ramas: Teoría de Control Clásica y Teoría de Control Moderna.

En la Teoría de Control Clásica se ha desarrollado y utilizado principalmente el *Controlador PID*. En esta teoría, los Sistemas de Control se clasifican bajo las siguientes propiedades:

1. *Causalidad*: los sistemas pueden ser causales o no causales, esto depende de la existencia de una relación entre los valores futuros y los valores actuales.

2. *Linealidad*: los sistemas pueden ser lineales o no lineales, esto en concordancia con la ecuación diferencial que modela al sistema.

3. *Función de tiempo*: los sistemas pueden ser de tiempo discreto o de tiempo continuo, esto depende si se modela bajo una ecuación diferencial o una ecuación de diferencias.

4. *Evolución de las variables*: los sistemas pueden ser estacionarios o no estacionarios, esto depende si sus variables son valores constantes o no constantes.

5. *Respuesta del sistema*: el sistema puede ser estable o inestable, esto dependenderá si la salida está acotada cómo lo está su entrada.

6. *Predicción del comportamiento del sistema*: el sistema puede ser determinístico o estocástico, esto dependerá si su salida es predecible o no.

El comportamiento de las plantas suele describirse a través de *Funciones de transferencia*, las cuales tiene la forma general presentada en la ec. 2. Estas funciones se obtienen a través de los sistemas de ecuaciones diferenciales que modelan a la planta.

$$G(s) = \frac{b_m s^m + ... + b_1 s + b_0}{s^{m+1} + a_m s^m + ... + a_1 s + a_0} \tag{2}$$

Controlador PID

Es uno de los controladores más comunes, busca modificar proporcionalmente (error presente), integralmente (error pasado) y diferencialmente (error futuro). Existen dos maneras de implementarlo, la primera es completamente analógica, utilizando circuitos físicos. La segunda es una implementación digital y permite ser implementado en cualquier microcontrolador, puesto que las funciones de transferencia, pueden ser modeladas como ecuaciones de diferencias.

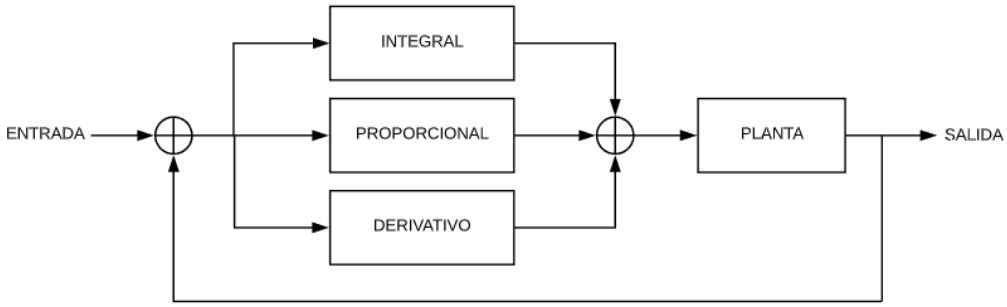


Figura 18: Diagrama de bloques de un Sistema de Control con controlador PID.

Básicamente, cada uno de los bloques que conforman el Controlador PID, son un coeficiente que ajusta de la manera deseada la salida.

Ecuaciones de diferencias

Una ecuación de diferencias es una representación discreta de una ecuación diferencial, generalmente se transforma del dominio de la frecuencia (s) al dominio discreto (z). Esto es necesario para implementaciones digitales ya que las distintas herramientas computacionales son meramente discretas y no son capaces de manejar funciones continuas. Una ecuación de diferencias tiene como forma general la ec. 3.

$$u(n) = b_0 f(n) + b_1 f(n-1) + ... + b_k f(n-k) - a_1 u(n-1) - a_2(n-2) - ... - a_3(n-k) \tag{3}$$

$$\forall \text{ n } \exists \text{ en Z}$$

Donde:

$u(n-k)$ son las salidas anteriores,
$f(n-k)$ las entradas anteriores y
$f(n)$ la entrada actual.

La k será equivalente al grado de la Función de transferencia, donde en la ec. 2 sería m.

Controlador discreto

Un controlador discreto, es una versión de un controlador pero en su versión discreta. Existen diferentes métodos de discretización, los más comúnes son: Tustin, Zero-Pole Matching, Zero-Order Hole, Backward Euler, entre otros. Estos controladores son representados a través de ecuaciones de diferencias, las cuáles pueden ser implementadas en un microcontrolador. Los controladores discretos son necesarios, pues un controlador continuo no puede ser implementado en un microcontrolador. Sin embargo, el diseño del controlador nace como continuo por la naturaleza de la planta. De ahí surge la necesidad de discretizar el controlador. Ambas representaciones son equivalentes, por lo que los resultados a obtener deberían ser muy similares. De hecho, es mucho más sencillo ajustar un controlador discreto que uno analógico, pues en el caso de un Controlador PID Analógico, esto significaría cambiar valores de resistencia (o en su defecto capacitancias) para tener un efecto en cada uno de los coeficientes. En un Controlador PID Discreto significaría solamente modificar un número vía software, este número es el coeficiente como tal.

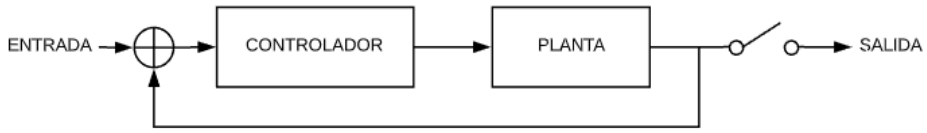


Figura 19: Diagrama de bloques de un Sistema de Control con controlador discreto.

En la Figura 19 se observa una diferencia al diagrama de bloques de la figura 18, y es el interruptor. Este nuevo elemento, indica la discretización que conlleva el sistema, se ha perdido la continuidad en el tiempo. Es necesario aclarar que el controlador también es distinto, pues en este nuevo sistema se tiene el equivalente ya discretizado.

6.6. Matlab R2018b

Matlab es un pograma especializado para la matemática que combina un entorno favorable para el análisis iterativo y el proceso de diseño. Tiene su propio lenguaje en el cual se pueden expresar muchas cosas de manera directa, tal como: vectores, matrices, funciones de transferencia, entre muchas otras cosas. Da la posibilidad de generar un «scripts» que pueden ser reutilizados. Todas las herramientas de Matlab han sido diseñadas a un nivel profesional, por lo tanto, es un programa muy confiable y es utilizado mucho en áreas académicas, de investigación e industriales.

Se ha decidido utilizar la versión R2018b por la disponibilidad de licencia en la Universidad.

6.6.1. Simulink

Es una herramienta de Matlab, completamente gráfica, que permite realizar simulaciones fieles a la realidad. La facilidad está en la simulación de sistemas dinámicos, pues Simulink ofrece todo lo que en un laboratorio sería necesario para experimentar. En el caso de los circuitos, se cuenta con distintas fuentes de voltajes, medidores, osciloscopios y los principales elementos que componen los circuitos.

Además, es posible trabajar con Matlab y Simulink en conjunto, se combina la programación textual con la programación visual para diseñar los sistemas deseados en un entorno de simulación. En los sistemas de control, se pueden modelar las dinámicas de la planta y ajustar y revisar los controladores [16]. Es necesario para verificar el diseño de los controladores, antes de hacer una implementación real, hay que simular.

Otra gran ventaja, es que a partir de modelos realizados en simulink, es posible obtener funciones de transferencia que pueden ser utilizadas en el entorno de Matlab.

En la Figura 20 se puede apreciar un sistema de control genérico, lo interesante es observar el uso de funciones de transferencia (que fueron obtenidas a través de Matlab) para efectuar la simulación.

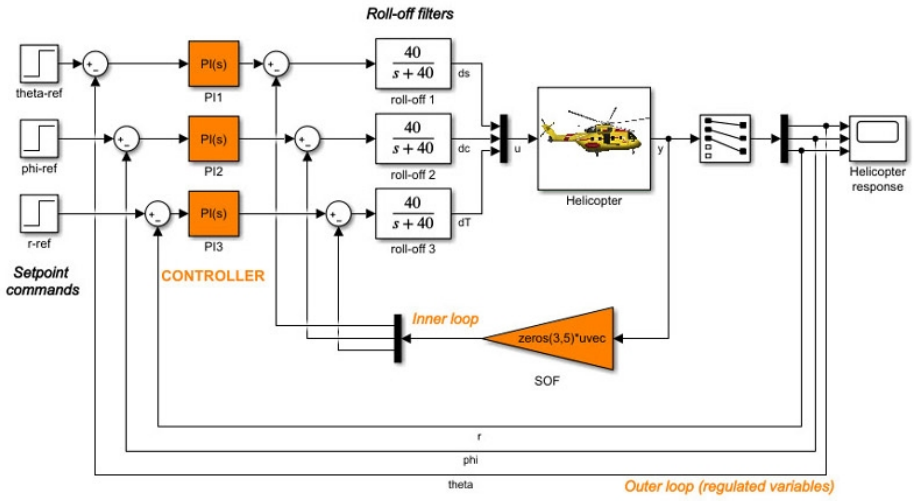


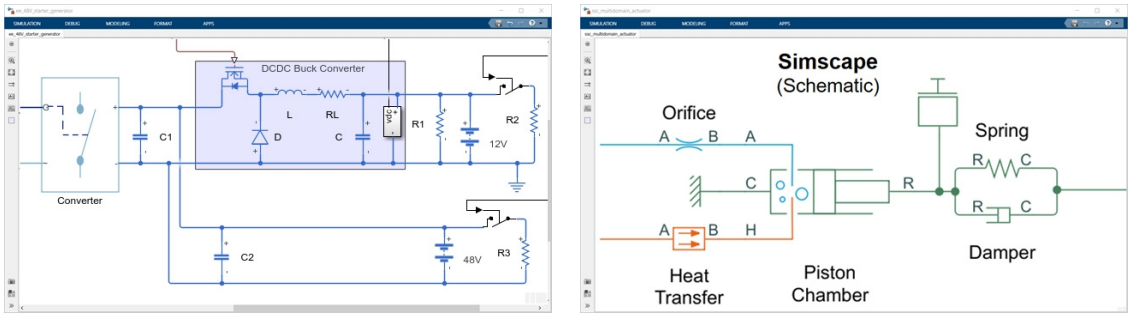
Figura 20: Simulink - Sistema de Control. [16]

6.6.2. Simscape

Simscape es una herramienta de Matlab que permite representar sistemas físicos en el entorno de Simulink. Simscape simula únicamente componentes físicos, especialemte los

principales utilizados en los sistemás físicos como lo pueden ser los sistemas mecánicos o los circuitos eléctricos, algunos ejemplos son: motores eléctricos, actuadores hidráulicos, componentes eléctricos pasivos, etc. A pesar de lo anterior, es posible integrar estos componente físicos con diagramas de bloques y otros modelos del entorno de Simulink. Debido a esto, es muy aprovechable utilizar los componentes de Simscape, porque asegura un diseño limpio.

Hace falta mencionar que Simscape es ideal para el desarrollo de Sistemas de Control y para evaluar el desempeño de los mismos. Ofrece componentes que pueden modelarse como ideales o prácticos, esta es una gran ventaja que permite obtener resultados completamente certeros y precisos. Además, estos componentes también pueden ser parametrizados con variables y expresiones de Matlab [17].



(a) Convertidor DC-DC (b) Sistema Mecánico e Hidráulico

Figura 21: Ejemplos de sistemas físicos utilizando Simscape. [16]

CAPÍTULO 7

Análisis y diseño preliminar

En el inicio del proyecto, todavía no se había podido conocer a detalle la problemática, las necesidades ni los requerimientos de la tabacalera. Por tanto, comenzar a desarrollar el proyecto era un tanto complicado pues no era posible trazar una dirección. Sin embargo, el proceso de indagación y recopilación de información básica si se pudo realizar.

Se pudo formar una idea previa de la forma en que se pensaba abordar el proyecto, se pudieron preseleccionar algunos componentes y circuitos pero nada definitivo. Tal es el caso del PIC16F887, que en el protocolo aparece como parte del Marco Teórico y que para este trabajo final, ha sido cambiado por el PIC16F1789. El reemplazo fue necesario al ir conociendo las nuevas necesidades que fueron surgiendo sobre la marcha. De manera similar sucedió con otros componentes y circuitos, todos los reemplazos conllevaron una mejora significativa pero presentaron nuevos retos.

7.1. Identificación de requisitos

Gracias a una primera reunión con la tabacalera, en donde se visitó un centro de ventas, fue posible conocer los distintos modelos de IQOS disponibles en el mercado del país. Se pudo definir en conjunto una idea del funcionamiento general de la máquina limpiadora. Pero más importante, se pudieron definir los requisitos principales y se conocieron las necesidades básicas. Cada una de ellas se describirá a continuación. Además se conoció la intención final de la máquina y cómo sería utilizada, la máquina está destinada a ser operada por un persona encargada y será un servicio que la tabacalera desea prestar.

Limpieza automática

Se requiere de un sistema que sea capaz de realizar todo el proceso de limpieza de manera completamente automática. Se busca que la intervención manual sea la más mínima posible. De manera que, el usuario final no deba preocuparse por llevar a cabo ningún paso del método de limpieza, sino que solamente sea responsable de accionar la máquina limpiadora.

Proceso de limpieza completo

Es muy importante que la máquina pueda llevar todo el proceso de limpieza por completo. Una máquina que solamente pueda realizar parcialmente este proceso no es funcional, por lo tanto es necesario que cada paso del método de limpieza sea efectuado por la máquina. Para asegurar una limpieza apropiada y que los clientes puedan continuar con el uso óptimo de sus dispositivos.

Sin daños a los dispositivos

Es de suma importancia que la máquina de limpieza no provoque ningún daño a los dispositivos, puesto que de ahí surge también la urgencia de crear una máquina que lleve a cabo el proceso de limpieza. Este requerimiento no es flexible, es una de las principales prioridades de diseño. No servirá de nada una máquina que cumpla con los demás requerimientos si no cumple con este. No se coloca como el primero en importancia porque para cuestiones de desarrollo, primero es necesario analizar y diseñar los requerimientos previos.

Menor tiempo posible

Cómo se desea que la máquina limpiadora sea un servicio que la tabacalera pueda prestar en sus centros de distribución, hace falta que el tiempo que lleve el proceso de limpieza sea el menor posible. Esto porque no se quiere que los clientes tenga que esperar un tiempo excesivo, esto le restaría éxito al servicio. Además, se comentó que diariamente llegan al menos 50 clientes, por lo que se les debe poder atender a todos.

Amigable para el usuario final

Cómo se desea que la máquina limpiadora sea operada por un colaborador de la tabacalera, el diseño y la forma de uso puede ser un tanto más compleja que para un usuario común de IQOS. De cualquier manera, esto no permite que la complejidad sea muy alta. Por lo tanto, es necesario que la forma de controlar la máquina sea completamente intuitiva y que su diseño sea tal, que el usuario final no se confunda con su funcionamiento.

7.2. Restricciones

A pesar de tener ya un direccionamiento adecuado de cómo debía dirigirse el proyecto, no se contaba con la información técnica suficiente ni con la retroalimentación necesaria para que el desarrollo fuera fluido. Esto debido a que no fue posible entablar una buena relación de comunicación con la tabacalera, de quienes no fue posible obtener más información ni comunicación luego del primer acercamiento. El acceso a información técnica de los IQOS y a los dispositivos como tal era de suma importancia para poder llevar a cabo con fidelidad el desarrollo del proyecto. Tampoco se pudieron organizar más reuniones para revisiones eventuales, la retroalimentación de la tabacalera juega un papel importante para saber si las necesidades y requerimientos se están cumpliendo o si bien, pueden mejorarse o deben cambiarse.

Sin embargo, esto no era motivo suficiente para imposibilitar el proyecto. Se procedió a continuar con el desarrollo del mismo con la información general, a la que todo público tiene acceso. Fue necesario conseguir un IQOS 2.4 Plus para poder tomar las mediciones y consideraciones necesarias para los mecanismos de la máquina de limpieza, puesto que a partir de ellos debían surgir los diseños de este trabajo de graduación.

7.3. Sistemas mecánicos y sistemas eléctricos y electrónicos

Como anteriormente se mencionó, el proyecto general fue dividido en dos grandes módulos. Siendo los *Sistemas Mecánicos* y los *Sistemas Eléctricos y Electrónicos*. Es importante mencionar que en el inicio del proyecto, se trabajó muy en conjunto con el desarrollador del Sistema Mecánico, puesto que era necesario definir en conjunto los actuadores para los mecanismos, ya que serían controlados de manera lógica desde el microcontrolador.

Cabe mencionar que siempre se mantuvo una comunicación fluida con la contraparte mecánica, para poder ir definiendo y cambiando los diseños acorde a los cambios que surgieron en los mecanismos. A pesar de que ambos módulos se desarrollaron completamente independientes uno del otro, se aprovecho esta buena comunicación para poder darle ese valor agregado, de alcanzar un prototipo final funcional.

7.4. Módulos de desarrollo

Para llevar a cabo de forma correcta y ordenada el desarrollo de este proyecto, es necesario separar y delimitar bien los módulos en los que se dividirá. Esto permitirá que cada uno de los módulos pueda trabajarse independientemente e incluso, en paralelo. Sin embargo, como todo es parte de un conjunto, siempre existirán dependencias entre los módulos.

Por lo tanto, se ha decidido hacer la separación en tres distintos módulos, los cuales se describirán en seguida:

- Módulo de potencia

- Módulo de control
- Módulo de sistemas de protección

7.4.1. Módulos de potencia

Los módulos de potencia se conforman por todos los circuitos encargados de suministrar la potencia que requiere el resto de sistemas, como lo es el control de motores, el control lógico y por supuesto, la alimentación general. Estos circuitos serán los primeros en ser diseñados, pero también serán los que más cambios tengan, pues se sabe por adelantado que la posibilidad de tener que cambiar componentes es muy alta.

7.4.2. Módulos de control

El módulo de control, como lo dice su nombre, será el encargado de llevar todo el control lógico del funcionamiento de la máquina. La dificultad acá, recae en la combinación de la electrónica digital con la electrónica analógica. Es necesario que todo el control lógico se pueda llevar de forma precisa, además que sea fácil seleccionar los modos de funcionamiento.

7.4.3. Módulo de sistemas de protección

Los sistemas de protección serán los últimos en ser diseñados e implementados, esto porque se requiere que los anteriores módulos ya estén en versión final para poder implementarlos. Los sistemas de protección buscan dar un valor agregado a todos los sistemas, para aumentar la confiabilidad y certeza de un buen funcionamiento y prevenir un mal funcionamiento o daño.

Esencialmente los sistemas de protección son dos: protección de alimentación y protección de control. El primero es para crear una barrera entre la máquina limpiadora y la toma de energía. La segunda, es básicamente el diseño e implementación de sistemas de control para los motores DC, para garantizar que su comportamiento este dentro de los límites permitidos. De esta forma se protege tanto el IQOS (al evitar la posibilidad de daños internos) y se protege el circuito de alimentación, pues se consolida un consumo de potencia ideal.

CAPÍTULO 8

Análisis y diseño de sistemas

8.1. Circuitos de potencia

Antes de comenzar los diseño, fue necesario seleccionar los actuadores que cumplieran con los requerimientos. Por lo tanto, se hicieron pruebas de funcionamiento con distintos motores, simulando las cargas con las que serían finalmente utilizados.

De ahí se obtuvo lo siguiente:

Propósito	Detalle	Voltaje Máximo (V)	Corriente Máxima (mA)	Cantidad
Bomba	Motor DC	13.5	1500	1
Cepillo (Holder)	Motor DC	10	500	1
Cepillo (Cap)	Mini DC	2.5	200	2
Mecanismo Cap	Mini Servo 9g	5	200	1

Cuadro 1: Actuadores seleccionados.

Es importante mencionar que en ningún momento se utilizarán todos los motores en simultáneo. Se ha decidido esto por dos razones, primero que no es realmente necesario y segundo, que al hacerlo de esta forma, se reduce la corriente de operación máxima de todo el circuito. Esto deja un consumo máximo de corriente de $1.5A$ en actuadores.

8.1.1. Circuito de alimentación

El circuito de alimentación, conocido como fuente de alimentación, será el encargado de suministrar toda la potencia que requiere la máquina de limpieza para funcionar. Para poder

diseñarla, fue necesario hacer pruebas individuales del resto de módulos y componentes, para conocer los consumos de corriente y los voltajes de operación y así saber cuál es la corriente máxima de operación.

De ahí se obtuvo que la corriente máxima de operación es de $2A$ ($1.5A$ del actuador principal y $0.5A$ del resto de circuitos)y que se requieren tres voltajes distintos: $15V$, $12V$ y $5V$. La alimentación de $15V$ será para el control de los motores de $10V$ y $13.5V$, la alimentación de $12V$ servirá para un ventilador (necesario por la cantidad de calor a disipar) y la alimentación de $5V$ para el microcontrolador, Servo-Motor y LCD.

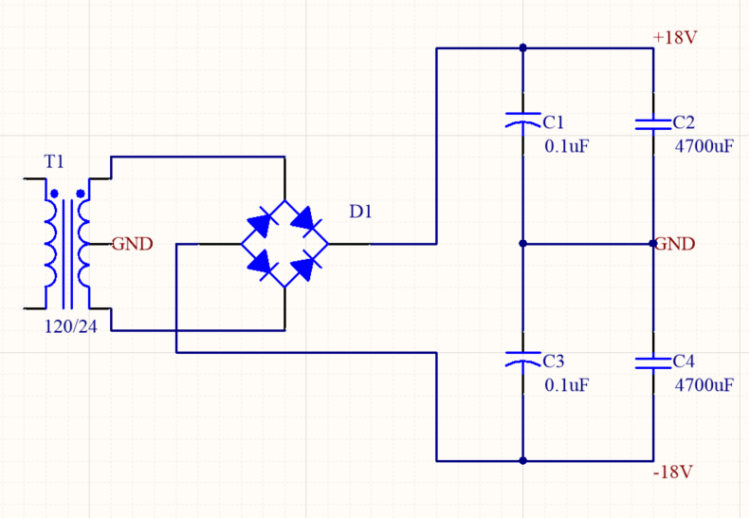


Figura 22: Rectificador de onda completa AC-DC

Como se desea que el dispositivo limpiador pueda ser implementado con facilidad, debe tomarse en cuenta la alimentación primaria. Esta, por motivos de generalización, debe ser la alimentación proveniente de cualquier tomacorriente. En Guatemala, la alimentación más común es de 120V@60Hz. Por lo tanto, la primera consideración es la conversión de voltaje AC a DC.

En la Figura 22 se observa el circuito rectificador de la señal de alimentación primaria, que será el voltaje proveniente del tomacorriente $120V$ AC. El transformador se encarga de reducir el voltaje, a su salida se encuentra una señal de $24V$ AC (RMS). El puente de diodos es en realidad el encargado de rectificar la señal sinusoidal, como es un puente completo, se aprovecha la onda completa. En la Figura 23 se observa la transformación de la onda. En la Figura 8.2.a está la onda completa (la de 24V RMS), en la Figura 8.2.b está la onda después del rectificador y en la Figura 8.2.c está la onda completamente corregida, después de los capacitores. Ya puede ser considerada una señal constante en el tiempo (DC).

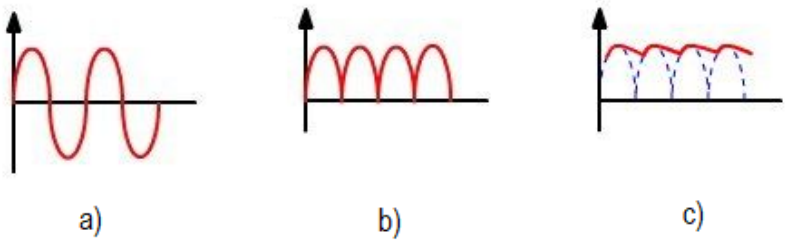


Figura 23: Onda rectificada [18]

Ahora bien, ésta es solamente la fuente primaria, la encargada de suministrar toda la corriente que requieran todos los sistemas en conjunto (max. $2A$). Para esto, se hace la selección de los componentes adecuados. En el Cuadro 2 se puede visualizar. Se puede notar que el transformador elegido tiene una capacidad de hasta $3A$, esto para dejar un margen y no sobrecalentar el transformador por la potencia que tenga que disipar.

Componente	Detalle	Cantidad
Transformador 120/24 V	3A	1
Rectificador Completo	10A	1
Capacitor 0.1uF	25V	2
Capacitor 4700uF	25V	2

Cuadro 2: Componentes requeridos para fuente primaria.

Una vez ya se tenga el voltaje DC deseado, se procede a la selección y diseño de los reguladores. En este caso, ya se conoce qué voltajes se requieren, por lo tanto se requiere:

Componente	Código	Detalle	Cantidad
Regulador +15V	LM7815	1A	3
Regulador -15V	LM7915	1A	3
Regulador +12V	LM7812	1A	1
Regulador +5V	LM7805	1A	1
Diodo	1N4001	-	6
Capacitor Electrolítico	1uF	25V	6
Capacitor Cerámico	0.1uF	25V	2
Capacitor Cerámico	0.33uF	25V	2

Cuadro 3: Componentes requeridos para fuentes secundarias.

A continuación, las figuras 24 a la 27 muestran los circuitos de conexión para poder obtener los voltajes deseados. Cabe recalcar que cada uno de estos circuitos formará parte de lo que será la fuente de alimentación.

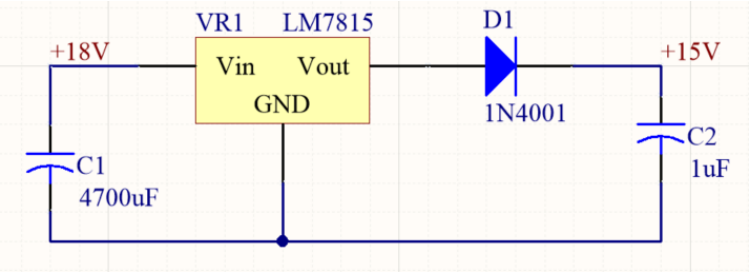


Figura 24: Regulador de +15V

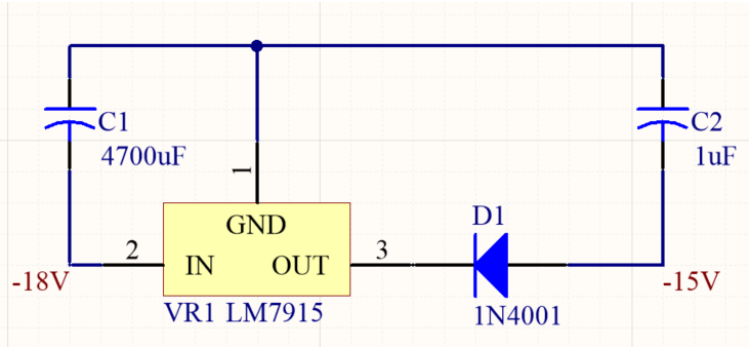


Figura 25: Regulador de -15V

Como se pudo observar en las figuras anteriores, estos reguladores llevan un diodo en serie a su salida de voltaje. Esto porque, debido a que no es posible conseguir reguladores de $3A$ en Guatemala, se deben conectar tres en paralelo. El diodo entonces, evita corrientes de retorno hacia el regulador. Esta corriente de retorno podría provocar un corto circuito en los reguladores, además sufrirían de sobrecalentamiento y estarían muy propensos a dañarse.

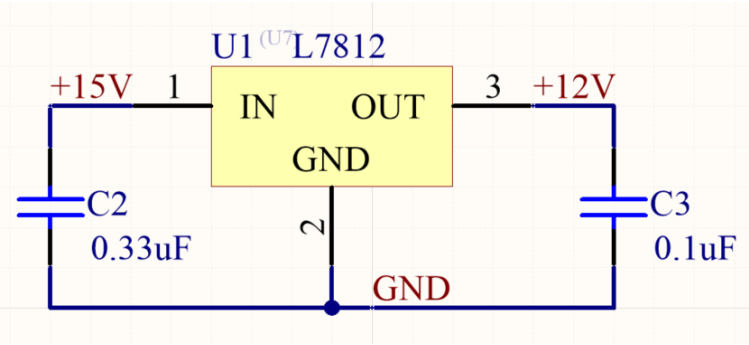


Figura 26: Regulador de -12V

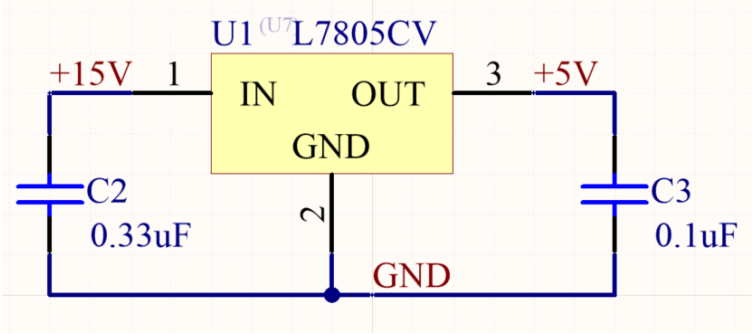


Figura 27: Regulador de +5V

8.1.2. Drivers de potencia

Los Drivers ha diseñar son una garantía de que los actuadores podrán tener la potencia requerida sin que el microcontrolador sufra daños. Todos los drivers de potencia tendrán un voltaje constante, del cual su magnitud dependerá directamente del microcontrolador, pero serán de corriente variable, según la carga lo solicite.

Motor DC

En el caso de los motores DC, se diseñarán dos tipos diferentes. El primero es un Drivers de potencia con ganancia unitaria y el segundo con ganancia variable (puesto que los voltajes de operación de ciertos motores superan el voltaje máximo que puede suministrar el microcontrolador.

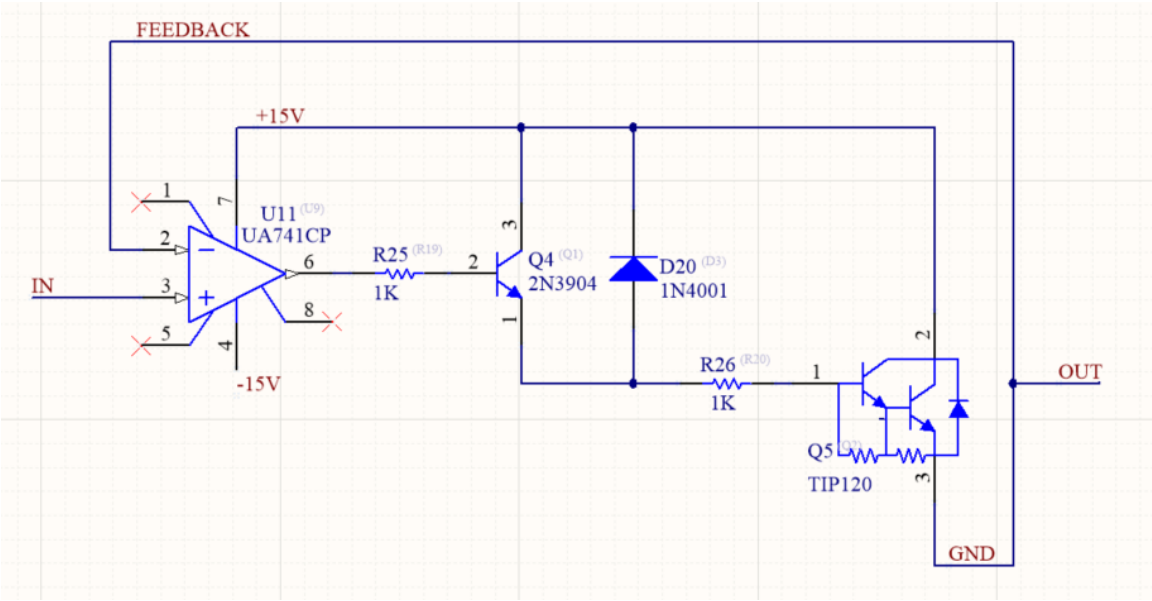


Figura 28: Driver de potencia con ganancia unitaria

En el caso del circuito de la Figura 28, el Op-Amp funciona como un seguidor de voltaje, por la retroalimentación que tiene, garantiza que el voltaje en el nodo de salida será el mismo que el voltaje en el nodo de entrada. Ahora bien, los transistores funcionan como amplificadores de corriente. Esto debido a que los circuitos integrados utilizados (tanto el microcontrolador como el Op-Amp) no pueden suministrar la corriente requerida por los motores. Se necesita una etapa doble de transistores por la cantidad de corriente que consumirán los motores (max. $1.5A$). El primer transistor le hará driving al segundo transistor, que es un transistor de potencia. Cabe mencionar que debe incluirse un diodo en paralelo al primer transistor para evitar corrientes de retorno, no se coloca otro en el segundo transistor porque este ya cuenta con uno interno.

Este segundo driver, tiene añadido una etapa extra al inicio, esta etapa es de ganancia. Esto significa que aumentará el voltaje obedeciendo a la siguiente ecuación:

$$V_{out} = V_{in} * (\frac{R2}{R1} + 1) \tag{4}$$

Donde $R2$ y $R1$ son las resistencias internas del potenciómetro, siendo en realidad un divisor de voltaje. Como es un potenciómetro, esta relación puede cambiarse y por eso se le denomina de Ganancia Variable.

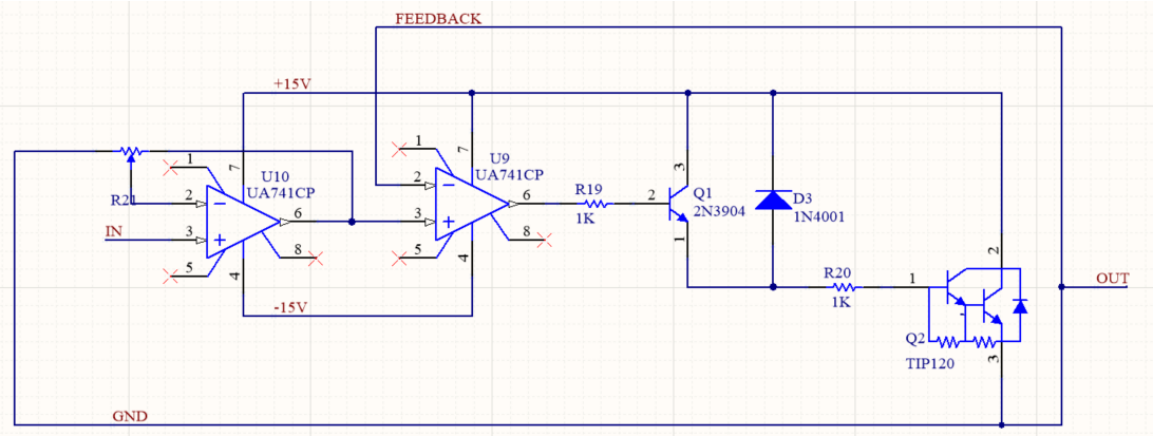


Figura 29: Driver de potencia con ganancia variable

Servo Motor

A pesar de que el microcontrolador si puede suministrar la corriente necesaria para hacer funcionar el Servo, se utilizará un seguidor de voltaje no inversor (para mantener polaridad del voltaje). Este circuito también se incluyó en los anteriores drivers de potencia y en este caso se incluye para que sea el Op-Amp quien suministre la corriente y no el microcontrolador. Todas estas consideraciones son para darle protección al microcontrolador y para asegurar un correcto funcionamiento.

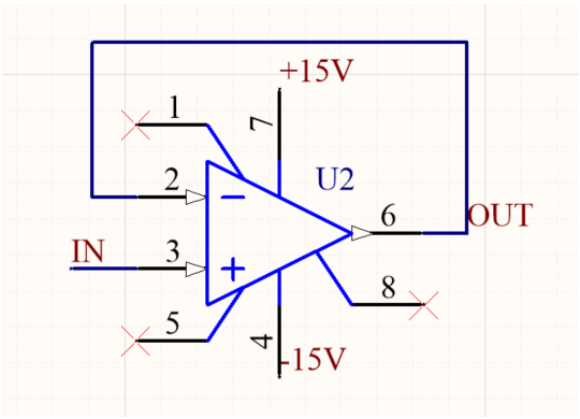


Figura 30: Seguidor de voltaje no inversor para PWM

Por tanto, se hace la selección de los componentes para los drivers de potencia:

Componente	Código	Detalle	Cantidad
Op-Amp	UA741	-	8
Transistor BJT NPN	2N3904	-	4
Transistor Darlington NPN	TIP122	-	4
Resistencia	1 kOhm	1/2 W	8
Diodo	1N4001	-	4

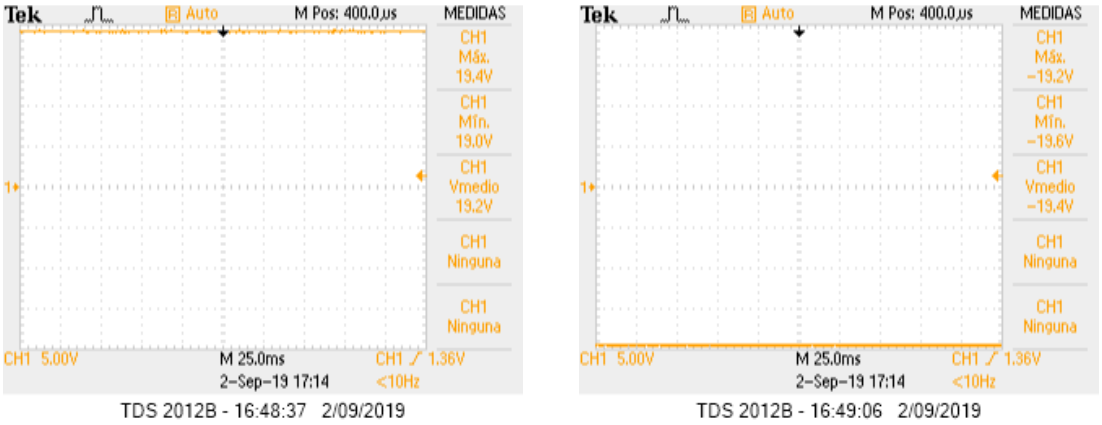
Cuadro 4: Componentes requeridos para drivers de potencia.

CAPÍTULO 9

Pruebas de los sistemas

9.1. Fuente de alimentación

Luego de las simulaciones, se procedió a la construcción de los circuitos en físico. Para lo cuál, se comenzó con la fuente de alimentación para poder observar y analizar el comportamiento en la realidad. Las pruebas a continuación fueron para el circuito sin ninguna carga.



(a) Voltaje en rectificador (terminal positiva)

(b) Voltaje en rectificador (terminal negativa)

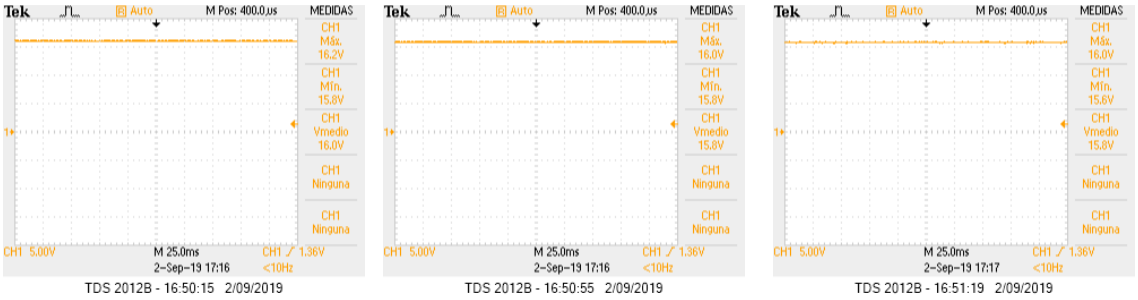
Figura 31: Voltajes en el rectificador

La primera etapa del circuito es puramente eléctrica, como se observa en el circuito, por

tanto es necesario conocer el voltaje DC que saldrá del nodo del Rectificador. Es necesario recordar que en este mismo nodo están en paralelo dos capacitores, por lo que se espera que el voltaje de salida sea mayor al voltaje RMS del transformador.

Como se observa en la Figura 31a el voltaje medio alcanzado es de $19.2V$ en la terminal positiva, en la Figura 31b se observa la terminal negativa con un voltaje medio de $-19.4V$. La diferencia de voltaje no es significativa por lo que puede ser despreciada. Dicha diferencia se puede deber a la manufactura no ideal de los componentes, en especial de los diodos (del Rectificador, que seguramente tienen un consumo levemente diferente de voltaje.

El voltaje en este nodo será el voltaje de entrada para los reguladores de voltaje de $15V$. En la Figura 32 se observan los voltajes de las salidas de los tres reguladores utilizados. En la Figura 32a se observa que el voltaje medio es de $16V$, o sea $0.2mV$ mayor a los otros dos reguladores. Esta diferencia también es despreciable, el mero asunto recae en que los reguladores están prácticamente $1V$ por encima del voltaje esperado ($15V$). Esto se debe al capacitor que va en paralelo a la salida de cada regulador, sin embargo no afecta para los fines del proyecto. Es más, se espera una leve caída de voltaje cuando la fuente de alimentación tenga carga completa.



(a) Voltaje en regulador de 15V 1 (b) Voltaje en regulador de 15V 2 (c) Voltaje en regulador de 15V 3

Figura 32: Voltajes en reguladores de 15V

La Figura 33 muestra el voltaje de salida definitivo para la alimentación de $15V$. Nuevamente se aprecia una caída de voltaje, que se debe a los diodos en serie a la salida de cada uno de los reguladores. A pesar de eso, el voltaje nominal es de $15.4V$, muy cercano a lo deseado.

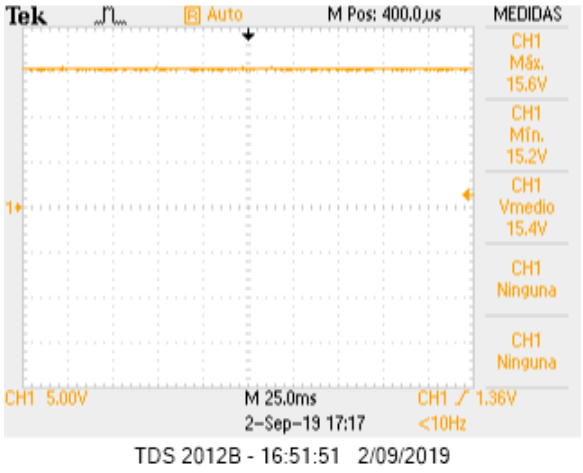
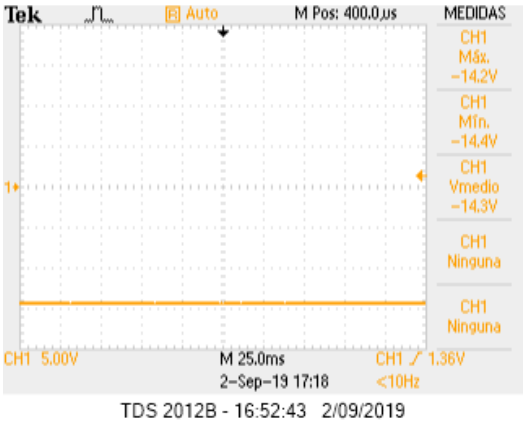
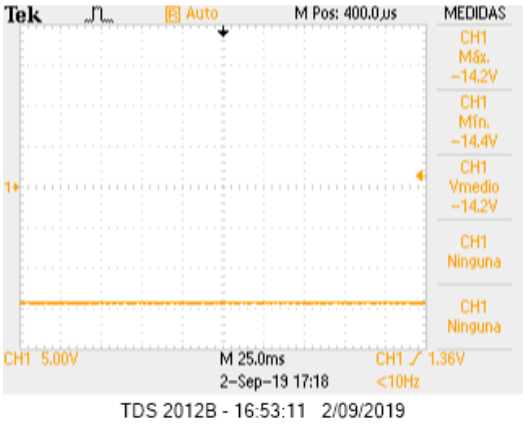


Figura 33: Voltaje en nodo común de reguladores de 15V

El caso de la alimentación de $-15V$ es similar a lo explicado anteriormente, aunque existen dos grandes diferencias. La primera es que solamente se utilizaron dos reguladores, debido a que la carga de este lado de la fuente será mucho menor, entonces no se requiere suministrar la misma cantidad de corriente. La otra diferencia es que el voltaje es menor al esperado (en magnitud). Esto se debe a que a pesar del diseño, la fuente no logra ser completamente simétrica y habiendo mencionado lo anterior, para los fines que se le dará, no afecta en lo absoluto.



(a) Voltaje en regulador de -15V 1



(b) Voltaje en regulador de -15V 2

Figura 34: Voltajes en el reguladores de -15V

La Figura 35 muestra el voltaje nominal de salida para la alimentación de $-15V$ y presenta un valor real de $-13.3V$. Se da otra caída de voltaje por los diodos en serie a la salida de los reguladores.

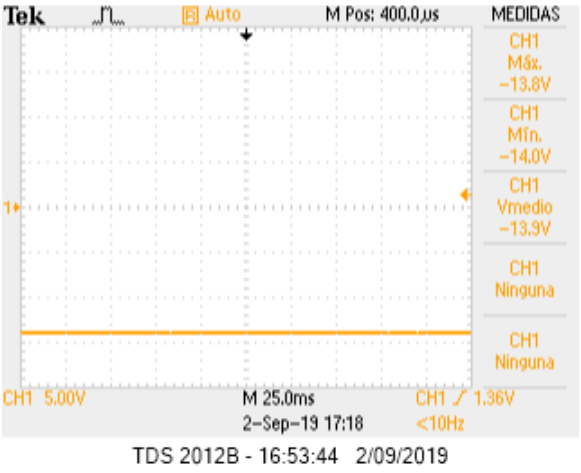


Figura 35: Voltaje en nodo común de reguladores de -15V

Pese a que los diodos puestos en serie a la salida de los reguladores provocan una caída de voltaje, no se puede prescindir de ellos. Se realizaron pruebas sin los diodos y los componentes se calentaron (sin carga), incluso el transformador. Esto debido a las corrientes de retorno, cuestión que los diodos evitan. También se hicieron pruebas con diodos en la conexión al nodo de referencia de los reguladores pero la caída de voltaje era mayor y las posibilidades de corrientes de retorno en el nodo de referencia no era muy altas. Por tanto, se optó por no colocarlos.

La Figura 36 muestra el voltaje en el nodo de salida del regulador de $5V$, el valor de voltaje medio es de $5.67V$.

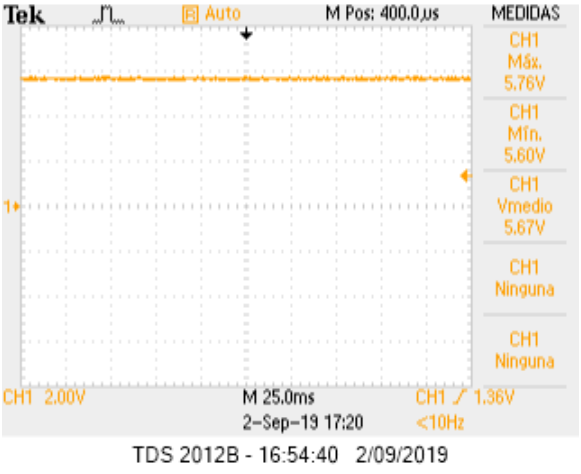


Figura 36: Voltaje en regulador de 5V

La Figura 37 muestra el voltaje en el nodo de salida del regulador de $12V$, el valor de

voltaje medio es de $13V$.

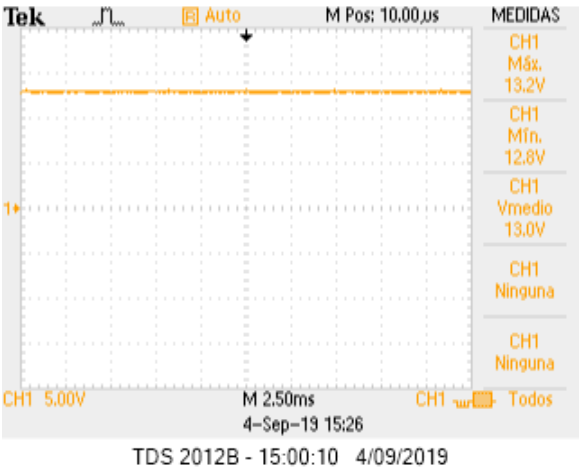


Figura 37: Voltaje en regulador de 12V

De la misma manera que en el resto de los reguladores, el voltaje en el nodo de salida de la alimentación, también fue más alto de lo esperado. Esto debido a los capacitores colocados en paralelo a la salida de los reguladores, sin embargo, tenga la carga completa, habrá una caída de voltaje en cada nodo de salida. Esto por la compensación con la corriente que tendrá que suministrar, obedeciendo a la ecuación:

$$V = IR \tag{5}$$

La Figura 38 muestra el diagrama completo de conexión de toda la fuente de alimentación, esto incluye tanto la fuente primaria como cada una de las fuentes secundarias (los reguladores de distintos voltajes).

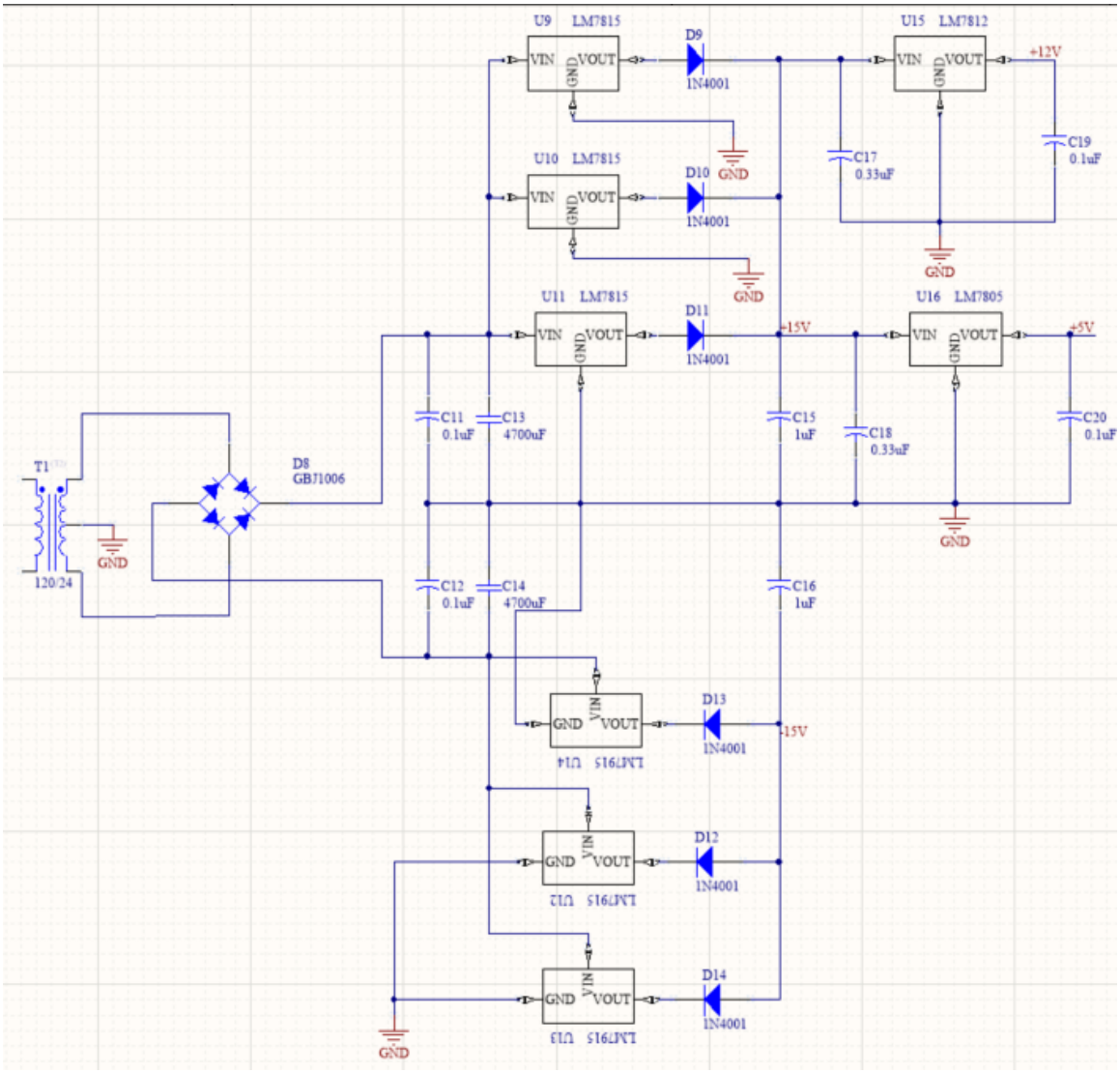


Figura 38: Diagrama de conexión de la fuente de alimentación

9.2. Pruebas de Drivers para motores DC

En la Figura 39 se observa el comportamiento del driver de potencia con un motor DC. La señal de entrada fue un divisor de voltaje dado por un potenciómetro. Se observa que al llegar al voltaje máximo, el ruido se incrementa. Esto ruido esta dado por la inductancia parásita que genera el motor.

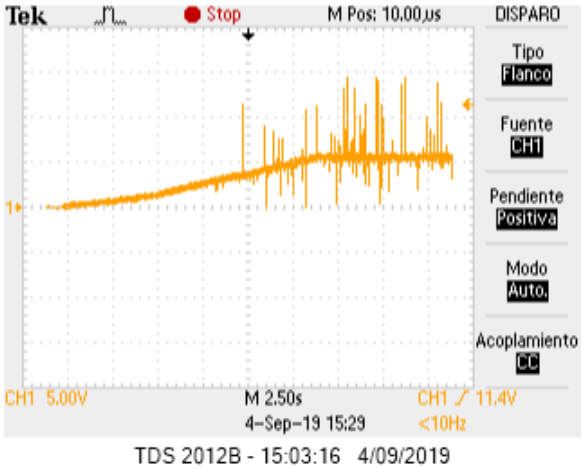


Figura 39: Voltaje de salida de Driver DC con motor

Para poder solventar el problema anterior, fue necesario contrarrestar la inductancia parásita introduciendo capacitancias. Para esto, se colocaron dos filtros: un filtro electrolítico de $1uF$ y un filtro cerámico de $0.1uF$. Luego de eso, se repitió la prueba y se obtuvo la señal de la Figura 40.

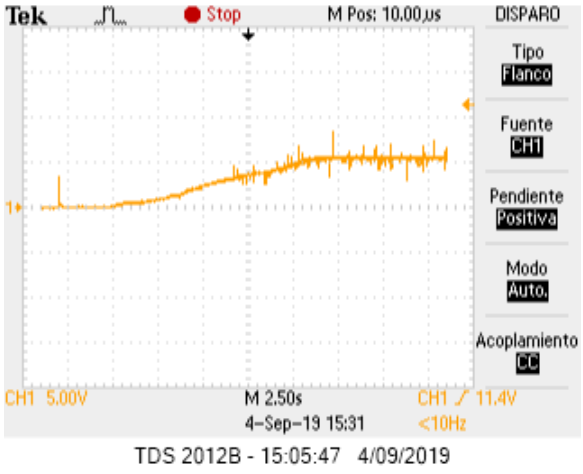


Figura 40: Voltaje de salida de Driver DC con motor con carga y con filtros

9.3. Pruebas con DAC

Para continuar con el desarrollo, era necesario hacer pruebas con el DAC del PIC16F1789. Se programó un diente de sierra en el DAC de 8-bits. En la Figura 41 se observa la señal obtenida.

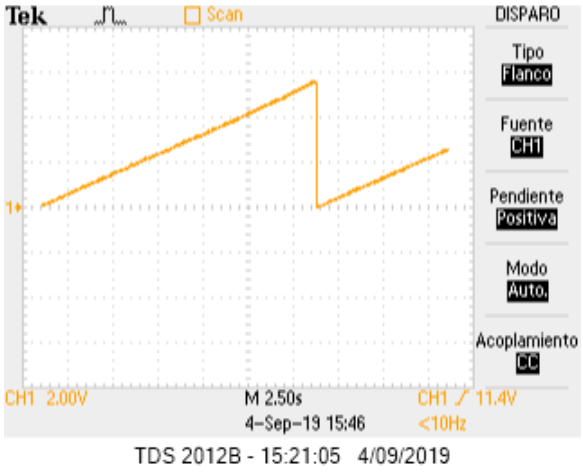


Figura 41: Diente de sierra con DAC del PIC16F1789

Posterior a eso, era necesario efectuar la prueba del DAC con el driver de potencia. Se programó nuevamente un diente de sierra. En la Figura 42 se observa la señal del diente de sierra pero con deformación paulatina. Cada vez que se repite el diente, el valor máximo de voltaje alcanzado es menor. También es notable el ruido de la señal.

El problema ahora es que el ruido afecta a la alimentación del microcontrolador, puesto que las referencias están interconectadas.

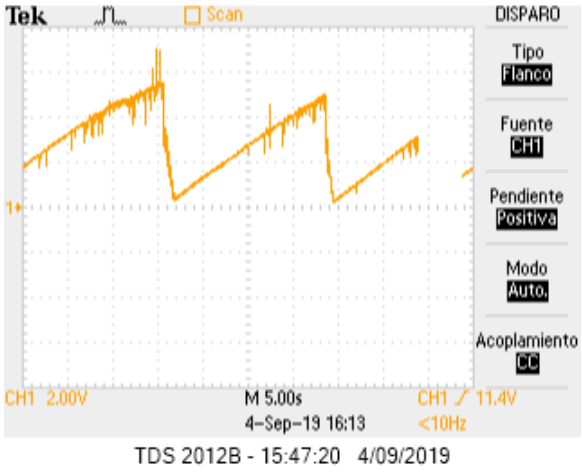


Figura 42: Diente de sierra con DAC del PIC16F1789 con Driver de potencia con motor

Para poder solventar el problema de ruido, fue necesario agregar filtros capacitivos a la alimentación del microcontrolador y un inductor del orden de mH en serie a la salida de la alimentación de $5V$ (que es la del microcontrolador). Tal como se muestra en la siguiente Figura (43). Además, fue necesario incluir dos capacitores en paralelo a la salida del driver

de potencia, uno de $1uF$ y otro de $0.1uF$. Por lo tanto, será necesario agregarlos en cada driver de potencia que utilice motores DC.

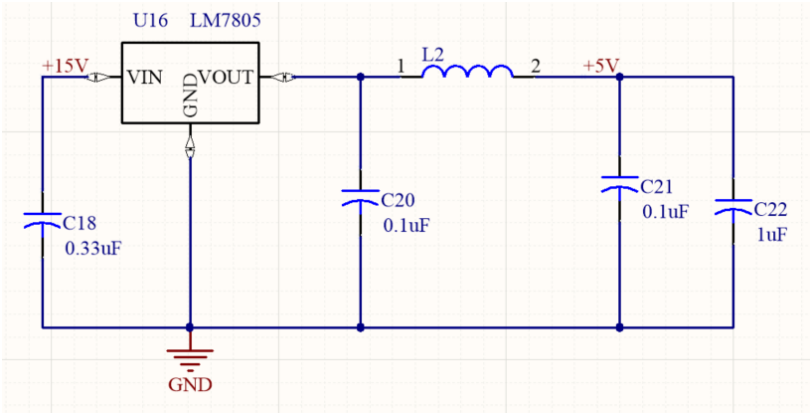


Figura 43: Nuevo diagrama de fuente secundaria de 5V con filtros

En la Figura 44 se observa la señal resultante. Se muestran varios periodos del diente de sierra, sin ruido y con carga.

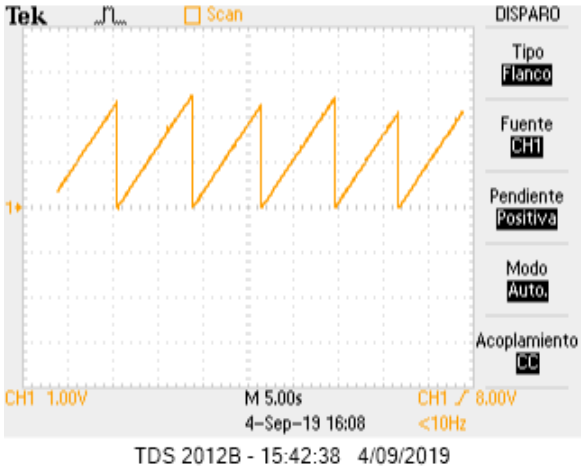


Figura 44: Diente de sierra con DAC del PIC16F1789 con Driver de potencia con motor con filtros

El Cuadro 5 presenta un resumen de todos los componentes seleccionados para la construcción de todos los sistemas de potencia. En ella, también se incluyen los disipadores de calor, los filtros para eliminación de ruido y las bases para los Op-Amp.

Componente	Código	Detalle	Cantidad
Transformador 120/24 V	-	3A	1
Rectificador completo	-	10A	1
Capacitor 0.1uF	-	25V	6
Capacitor 1uF	-	25V	4
Capacitor 4700uF	-	25V	2
Inductor		10mH	1
Regulador +15V	LM7815	1A	3
Regulador -15V	LM7915	1A	3
Regulador +12V	LM7812	1A	1
Regulador +5V	LM7805	1A	1
Diodo	1N4001	-	14
Capacitor electrolítico	1uF	25V	6
Capacitor cerámico	0.1uF	25V	2
Capacitor cerámico	0.33uF	25V	2
Op-Amp	UA741	-	8
Transistor BJT NPN	2N3904	-	4
Transistor Darlington NPN	TIP122	-	4
Resistencia	1 kOhm	1/2 W	8
Base 8DIP	-	-	8
Disipador	TO-220	-	8

Cuadro 5: Resumen de componentes seleccionados para etapa de potencia.

CAPÍTULO 10

Sistema de Control

10.1. Descripción del sistema

Los sistemas de control a diseñar, buscarán controlar indirectamente el torque de los distintos motores. Para alcanzar esto, se realizaron pruebas físicas de los motores, con cerdas, simulando el proceso de limpieza en el Holder, en el Cap y en la bomba. De ahí se obtuvo la siguiente tabla:

Motor	Voltaje (V)
Bomba	13.5
Holder	10
Cap	2

Cuadro 6: Voltajes para funcionamiento óptimo.

Es importante mencionar que el Cap utilizará dos motores idénticos. Los voltajes de la tabla anterior son los voltajes que aseguran un correcto funcionamiento. En otras palabras, garantizan la integridad de las piezas del IQOS y su buena limpieza. Por tanto, los sistemas de control a diseñar buscarán mantener constantes estos voltajes.

Por motivos de generalización, solamente habrán dos sistemas de control. Esto porque solo se van a tomar en cuenta dos plantas distintas. La primera será un driver de potencia con ganancia unitaria (el utilizado para los motores del Cap) y el otro será un driver de potencia con ganancia 4 (el utilizado para los motores de la bomba y del Holder). Además, se consideró otra simplificación, no se tomó en cuenta el transistor darlington (TIP122) pues no se encuentra dentro de los componentes de Simscape. Es válido hacer esta simplificación puesto que este transitor no tiene ningún efecto en el voltaje final a suministrar, solamente

en la corriente.

10.1.1. Driver de potencia ganancia unitaria

En la Figura 45 se observa el circuito de Simulink, herramienta que además de permitir simular el comportamiento del circuito, permite obtener modelos del comportamiento del circuito sin necesidad de tener que obtener datos experimentales. Se utilizaron los valores ya seleccionados anteriormente.

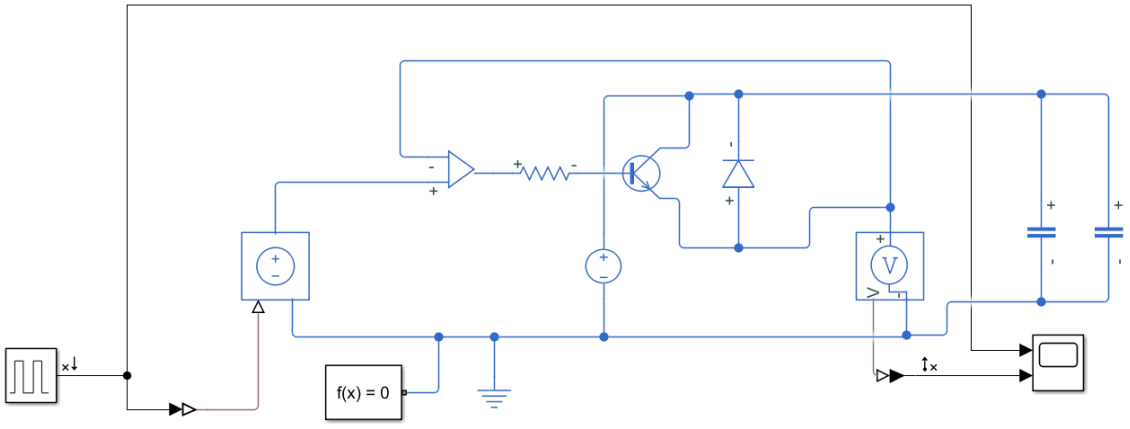


Figura 45: Driver de potencia con ganancia unitaria en Simulink

A través del uso del Linear Analysis Tool se pudo obtener el modelo linealizado de la planta en cuestión. La ventaja de esta herramienta es que permite exportar la variable de Simulink para que pueda ser manipulada en el entorno de Matlab. El modelo fue basado en la respuesta al escalón de la planta. Como era de esperarse, dicha respuesta debe ser igual a la ganancia, o sea, uno (1).

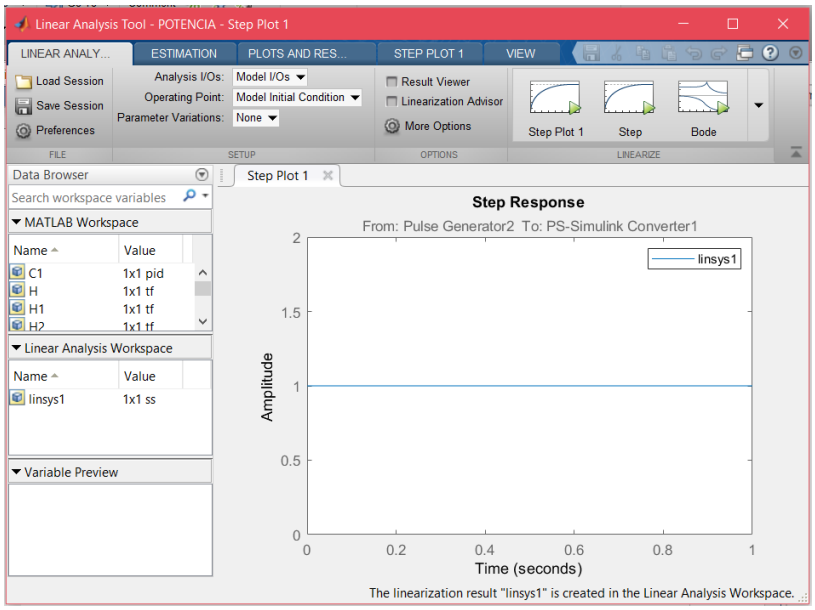


Figura 46: Linear Analysis Tool

En el entorno de Matlab, se requiere convertir la variable exportada de Simulink a una función de transferencia. Como se muestra en la Figura 47, es necesario utilizar el comando **tf()**. Para poder hacer el diseño de un controlador PID, se debe llamar a la herramienta *PID Tuner*. La figura también muestra el comando de invocación de la herramienta. Se puede observar que puede recibir como parámetro la función de transferencia de la planta que se desea controlar. Sin embargo, si no se ingresa ningún parámetro, la herramienta permite exportar la variable del entorno de Matlab.

```
H1 = tf(potenciaG)
pidTuner(H1)
```

Figura 47: Código para obtención de la función de transferencia

En la Figura 48 se observa el comportamiento de la planta sin ningún controlador diseñado. Es importante hacer mención que de por sí, el sistema ya tiene implementado un sistema de control y es básicamente la retroalimentación que tiene el circuito. Es por esto que se visualiza una respuesta muy buena. Sin embargo, como el sistema está propenso a perturbaciones eléctricas, originadas por los motores, es importante agregar un controlador para que el microcontrolador pueda conocer el valor actual de voltaje y modificar el valor de salida si se requiere.

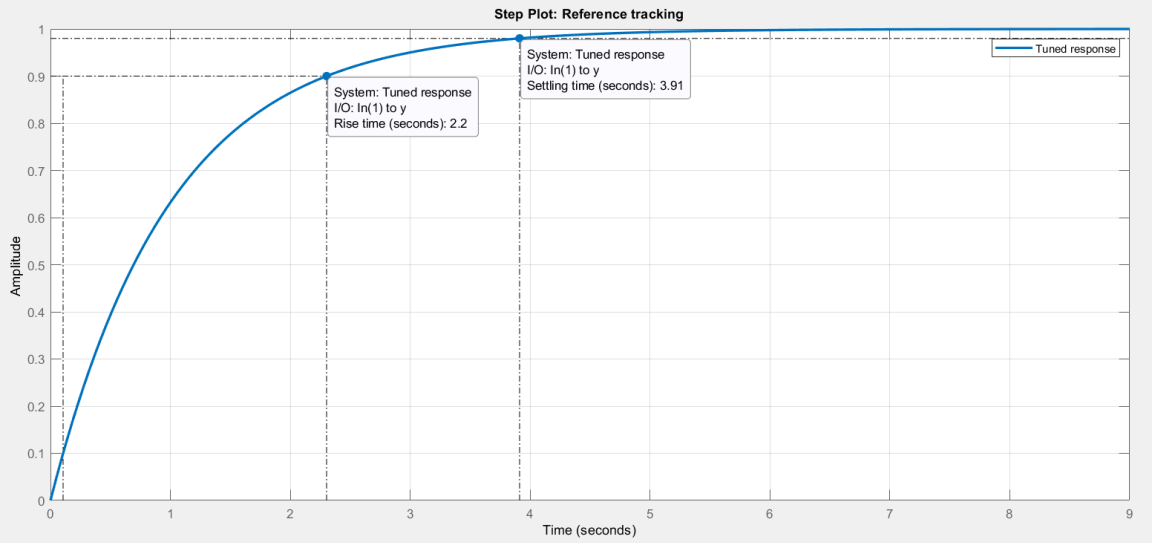


Figura 48: Sistema sin control

Por tanto, se ha decidido diseñar un controlador PID, sin embargo, la implementación solamente será de un controlador PI, pues como se observa en el Cuadro 7, no se obtuvieron valores para el control diferencial. En la Figura 49 se observa el cambio en el comportamiento del sistema, con el controlador diseñado su respuesta es mucho más rápida y por tanto será menos vulnerable a perturbaciones externas. Ahora la planta alcanzará el valor deseado en menos de 1 segundo.

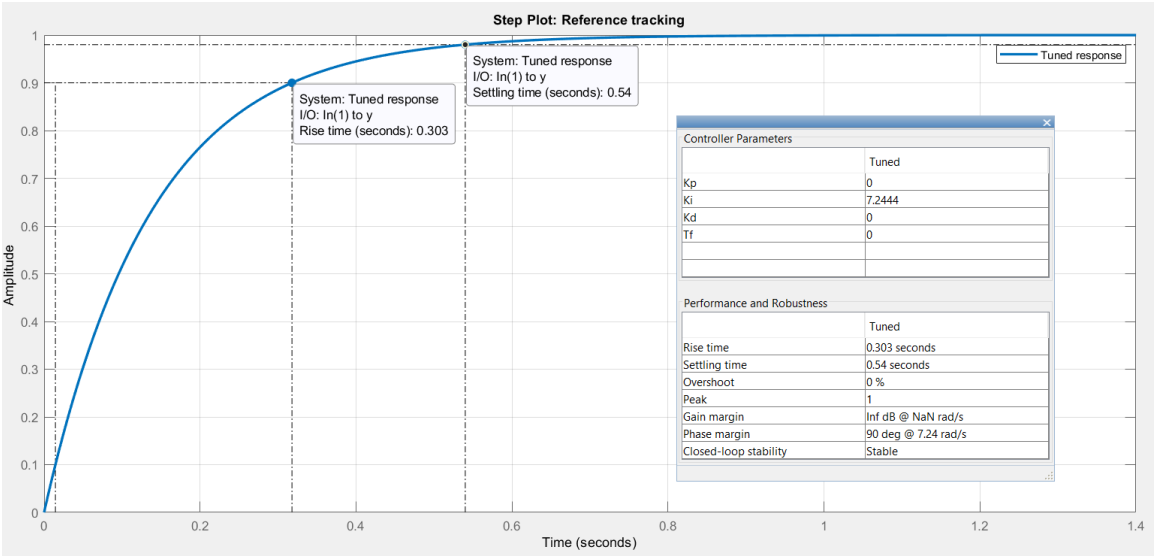


Figura 49: Sistema con control

Constante	Valor
Kp1	0
Ki1	7.2444
Kd1	0

Cuadro 7: Valores de las constantes del controlador PID diseñado.

10.1.2. Driver de potencia con ganancia 4

Como el diseño del controlador anterior, se siguió un proceso similar. En la Figura 50 se observa el circuito utilizado. En este caso se observa una etapa de ganancia extra, tal como es el driver de potencia para los motores de la bomba y del Holder.

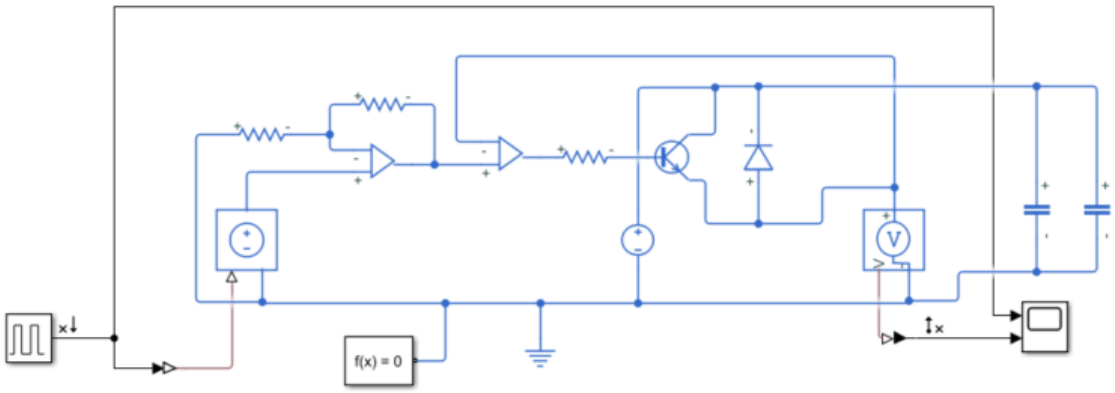


Figura 50: Driver de potencia con ganancia 4 en Simulink

Del mismo modo, utilizando el Linear Analysis Tool y los comandos de Matlab, se obtiene el comportamiento del sistema sin controlador, como se observa en la Figura 51.

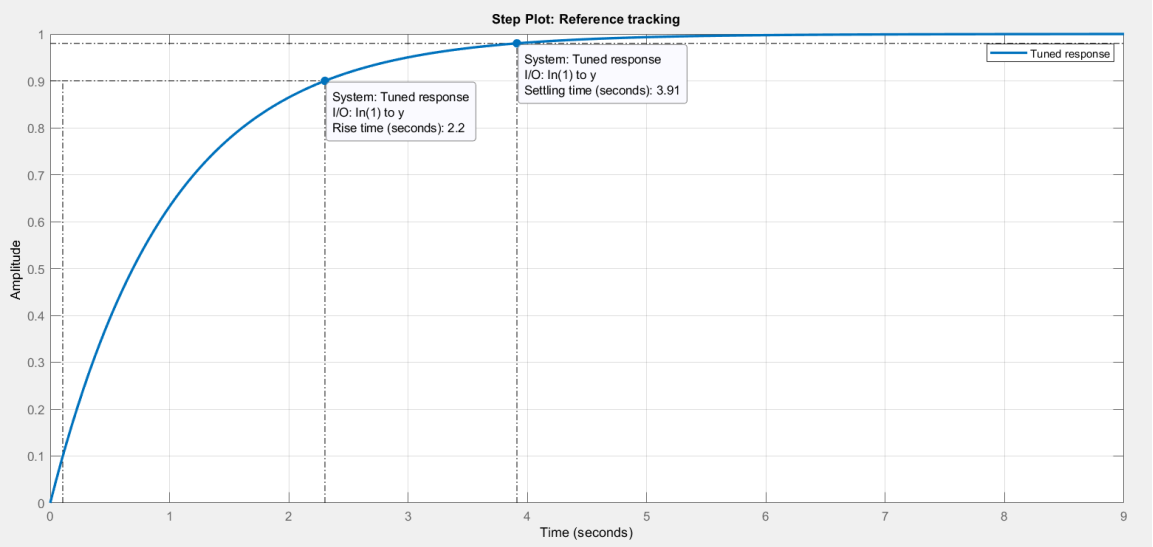


Figura 51: Sistema sin control

También se diseñó un controlador PID, pero nuevamente no se obtuvieron valores para el controlador diferencial (ver Cuadro 8). También se buscó que la respuesta fuera más rápida, alcanzando el valor óptimo en menos de un segundo.

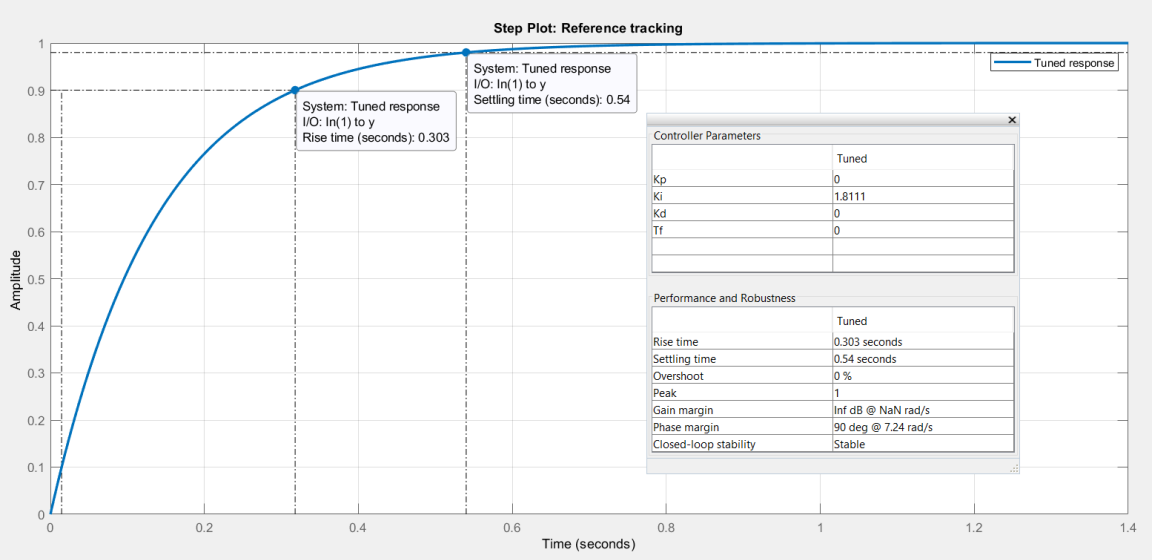


Figura 52: Sistema con control

10.1.3. Controladores

Matlab ofrece la obtención de la ecuación del controlador, es necesario utilizar el comando **pid(Kp,Ki,Kd,Tf)**. En este caso, es fácil reconocer las tres constantes obtenidas para los controladores PID diseñados anteriormente. De ahí se obtienen las siguientes ecuaciones en

Constante	Valor
Kp2	0
Ki2	1.8111
Kd2	0

Cuadro 8: Valores de las constantes del controlador PID diseñado.

el dominio de la frecuencia:

$$C1 = \frac{Ki1}{s} \tag{6}$$

$$C2 = \frac{Ki2}{s} \tag{7}$$

Donde $Ki1 = 7.2444$ y $Ki2 = 1.8111$ (ver tablas 7 y 8).

Entonces (6) es la ecuación del controlador para el driver de potencia con ganancia unitaria y (7) es la ecuación del controlador para el driver de potencia con ganancia 4.

10.2. Implementación digital de los controladores

La forma más sencilla de implementar estos controladores es de manera digital, hacerlo de forma analógica no solo implica la construcción de más circuitos, sino que no garantiza un control fiel al diseño. Para poder implementar un controlador en un microcontrolador, primero es necesario discretizarlo.

10.2.1. Discretización de los controladores

Existen diferentes métodos de discretización, Matlab ofrece seis distintos métodos de discretización para los controladores PID. Para estos fines, se evaluarán tres. Para efectuar las discretización se debe utilizar el comando **c2d(sys,Ts,method)**. Donde **sys** es la variable tipo *controlador PID*, **Ts** es la frecuencia de muestreo deseada y **method** es el método de discretización deseado. Para todos los métodos se utilizará una frecuencia de muestreo de aproximadamente $Ts = 30us$.

Discretización Tustin

Utilizando el comando **c2d(C1,Ts,'Tustin')**, se obtiene el controlador discreto:

$$C1 = Ki1 * \frac{Ts(z+1)}{2(z-1)} \tag{8}$$

Utilizando el comando **c2d(C2,Ts,'Tustin')**, se obtiene el controlador discreto:

$$C1 = Ki2 * \frac{Ts(z+1)}{2(z-1)} \tag{9}$$

Donde $Ki1 = 7.2444$ y $Ki2 = 1.8111$ (ver tablas 7 y 8).

Discretización Zero-Pole Matched

Utilizando el comando **c2d(C1,Ts,'matched')**, se obtiene el controlador discreto:

$$C1 = Ki1 * \frac{Ts}{z-1} \tag{10}$$

Utilizando el comando **c2d(C2,Ts,'matched')**, se obtiene el controlador discreto:

$$C1 = Ki2 * \frac{Ts}{z-1} \tag{11}$$

Donde $Ki1 = 7.2444$ y $Ki2 = 1.8111$ (ver tablas 7 y 8).

Discretización Zero-Order Hole

Utilizando el comando **c2d(C1,Ts,'zoh')**, se obtiene el controlador discreto:

$$C1 = Ki1 * \frac{Ts}{z-1} \tag{12}$$

Utilizando el comando **c2d(C2,Ts,'zoh')**, se obtiene el controlador discreto:

$$C1 = Ki2 * \frac{Ts}{z-1} \tag{13}$$

Donde $Ki1 = 7.2444$ y $Ki2 = 1.8111$ (ver tablas 7 y 8).

Tras pruebas realizadas, se obtuvo que la mejor respuesta fue dada por los controladores discretizados a través de los métodos de *ZOH* y *ZPM*. Por tanto, se requiere llevar la ecuación del controlador discreto a su forma más expandida y con potencias negativas.

$$C1 = \frac{Ki1 * Ts}{1 - z^{-1}} \tag{14}$$

$$C2 = \frac{Ki2 * Ts}{1 - z^{-1}} \tag{15}$$

Obteniéndose la forma general siguiente:

$$C = \frac{b_0}{1 - a_1 * z^{-1}} \tag{16}$$

De ahí las siguiente tabla:

Controlador	$b_0(x10^{-5})$	a_1
C1	22.3	1
C2	5.57	1

Cuadro 9: Valores de las constantes del controladores discretizados.

10.2.2. Algoritmo de implementación

Para poder implementar los controladores discretizados en el microcontrolador, se requiere de un algoritmo que pueda actualizar los valores del controlador automáticamente. Entonces, se tiene:

$$e = r - y;$$
$$u = b_0 * e - a_1 * u_1;$$
$$u_1 = u;$$

Donde r es el valor de referencia (en este caso un valor constante), y la retroalimentación de la planta a controlar hacia el microcontrolador y u el valor controlado.

Cabe mencionar que el microcontrolador puede leer valores de hasta $5V$, por lo que será necesario implementar una etapa de ganancia menor a 1, para asegurar que el valor de retroalimentación no supere el voltaje máximo (como es el caso de los motores con driver de potencia de ganancia 4). Ver Cuadro 9 para los valores reales de implementación.

Además, es importante mencionar que los controladores a implementar no deben ser ejecutados todo el tiempo, sino solamente cada vez que uno de los motores requiera ser utilizado.

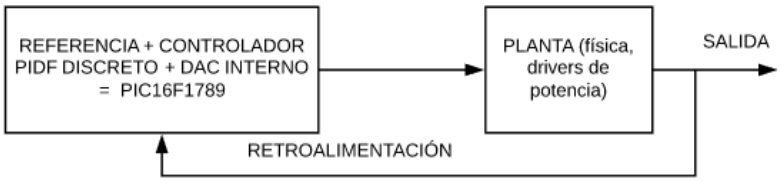


Figura 53: Diagrama de bloques de sistema de control discreto

CAPÍTULO 11

Control lógico

A pesar de que la programación del microcontrolador se inició desde las primeras etapas del proyecto, se ha dejado hasta este punto porque es acá dónde se realiza la unión y la coordinación de cada uno de los sistemas previamente desarrollados. La etapa de pruebas había requerido el uso del microcontrolador, sin embargo eran meramente primeros acercamientos. A diferencia de este punto que ya presenta los módulos definitivos.

Cómo ya se conoce, el microcontrolador seleccionado es el PIC16F1789. Para la entrega del protocolo, se había pre-seleccionado el microcontrolador PIC16F887, principalmente por motivos de disponibilidad en el país. No obstante, con la selección definitiva de los motores a utilizar y habiendo investigado otros posibles microcontroladores. Se optó por sustituir el PIC166F887 por el PIC16F1789, que es en realidad, la generación actualizada. Esta nueva elección incrementó el nivel de dificultad, por ser un microcontrolador relativamente nuevo, se cuenta con muy poca documentación y material de soporte. A pesar de ello, tiene características que permiten un incremento en la optimización del diseño final de todos los sistemas en desarrollo. Sin mencionar que se tuvo la posibilidad de poder comprarlo a través de internet.

En el Cuadro 10 se muestran los componentes seleccionados para llevar a cabo la etapa de control lógico.

Componente	Código	Detalle	Cantidad
Microcontrolador	PIC16F1789	40DIP	1
LCD 16x2	-	5V	1
Push-Buttons	-	-	4
Base 40DIP	-	-	1

Cuadro 10: Resumen de componentes seleccionados para etapa de control lógico.

11.1. Módulos

Para poder desarrollar ordenadamente e independientemente cada uno de las partes del control lógico, se ha decidido separar por módulos. Cada uno de ellos se describirá a continuación.

11.1.1. DAC

Cómo ya se sabe, el PIC16F1789 posee cuatro convertidores digital-analógico (DACs). Esto permite llevar el control de cualquier componente, elemento o circuito que requiera de una señal analógica (variable en el tiempo) como su entrada. Tal es el caso de los motores DC. Por tanto, los DACs permiten controlar, con cierta precisión, dichos elementos. Los DACs serán utilizados para ajustar el valor deseado de voltaje.

El primer DAC, de resolución de 8 bits, se utilizará para controlar el motor encargado de la limpieza del Holder. El resto de los DACs son de resolución de 5 bits y se utilizarán para el resto de motores. Como se mencionó mucho más arriba, la programación de estos módulos se hará a través del MCC.

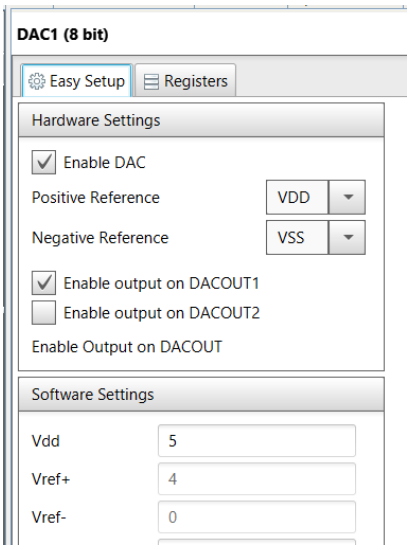


Figura 54: Configuración DAC vía MCC

Como se observa en la Figura 54, la alimentación del DAC y su referencia están dadas por valores internos del mismo microcontrolador. Si no, implicaría la implementación de un nuevo voltaje de referencia.

11.1.2. PWM

Para este módulo, fue necesario bajar la frecuencia de operación del microcontrolador ha $31kHz$, pues es la única frecuencia a la cuál se puede conseguir un período de $20ms$ en la señal PWM. Para el cálculo de este valor se utilizaron la siguientes ecuaciones:

$$PWM_{period} = [(PR2) + 1] * 4 * T_{osc} * (TMR2PrescaleValue) = 20ms \tag{17}$$

$$PulseWidth = (CCPRxL : CCPxCON < 5 : 4 >) * T_{osc} * (TMR2PrescaleValue) = variable \tag{18}$$

$$DutyCycleRatio = \frac{(CCPRxL : CCPxCON < 5 : 4 >)}{4 * ((PR2) + 1)} = variable \tag{19}$$

Nota: $T_{osc} = 1/F_{osc}$ y $F_{osc} = 31kHz$

Es posible observar, que el período de la señal PWM a diseñar, depende directamente de la frecuencia de funcionamiento del microcontrolador y de la configuración del TMR2. Inicialmente, todas las pruebas se habían realizado con una frecuencia de 8 MHz. Por lo tanto, aun probando distintos valores de *prescaler* no era posible conseguir un período de 20 ms. La figura 58 muestra la relación de los registros del microcontrolador con la señal PWM a generar.

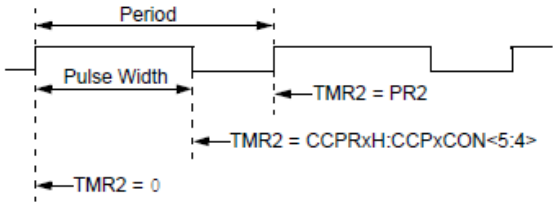
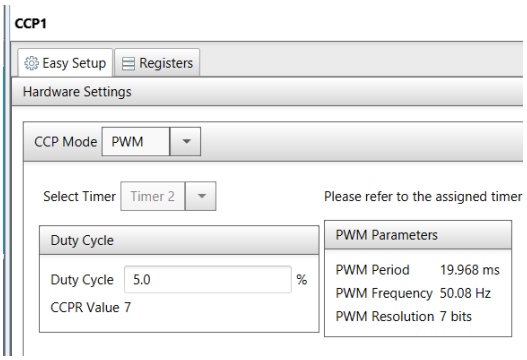


Figura 55: CCP PWM señal de salida

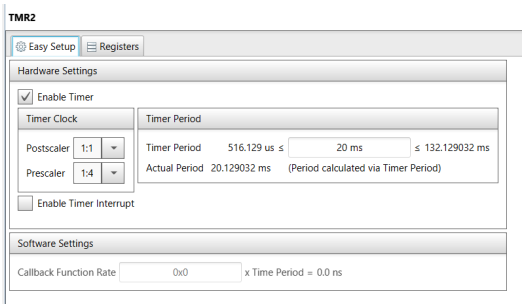
Como es de esperar, la resolución de

$$Resolution = \frac{log[4 * (PR2 + 1)]}{log(2)} = 7bits \tag{20}$$

Fue hasta que se cambió a la frecuencia más baja que se pudo recalcular todo de forma sencilla utilizando el MCC. La Figura 56a muestra la configuración para el manejo de servo motores. Se puede observar en la Figura 56b que el período del TMR2 será el período de la señal PWM.



(a) Configuración PWM vía MCC



(b) Configuración TMR2 vía MCC

Figura 56: Configuración para PWM

11.1.3. Botones

Para asegurar un correcto funcionamiento, se programaron interrupciones para los tres botones de control de funcionamiento. Esto, para evitar efectos de rebote. La Figura 57 ejemplifica el efecto rebote. Este efecto se da en casi todos los botones mecánicos y básicamente es que al momento de accionar el switch (pulsar el botón), se producen micro-pulsaciones que ocurren muy rápidamente. Para los humanos es casi imperceptible, pero para el microcontrolador no, por lo que él reconoce un tren de pulsaciones.

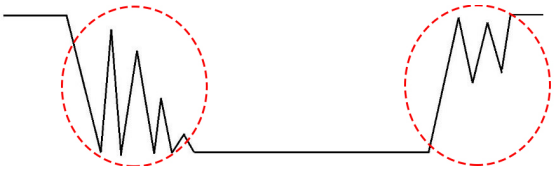


Figura 57: Efecto rebote en botones mecánicos [19]

La interrupción hace que el microcontrolador se quede solamente con la primera pulsación (se puede decidir el flanco, si de subida o de bajada). Esto asegura que no hayan lecturas erróneas. Además, otra ventaja de las interrupciones es que se ejecutan en paralelo con el resto de la programación, por lo que no importa en qué momento se haga la pulsación, el microcontrolador la reconocerá.

El fin principal de los botones, es levantar banderas, estas banderas serán las encargadas de ejecutar ciertos pedazos de código, correspondientes a las rutinas de limpieza.

11.1.4. LCD

Lastimosamente el MCC no cuenta con soporte para pantallas LCD genéricas, por lo tanto, fue necesario encontrar una librería que pudiera funcionar adecuadamente con el microcontrolador. Después de una búsqueda exhaustiva, se encontró la librería idónea.

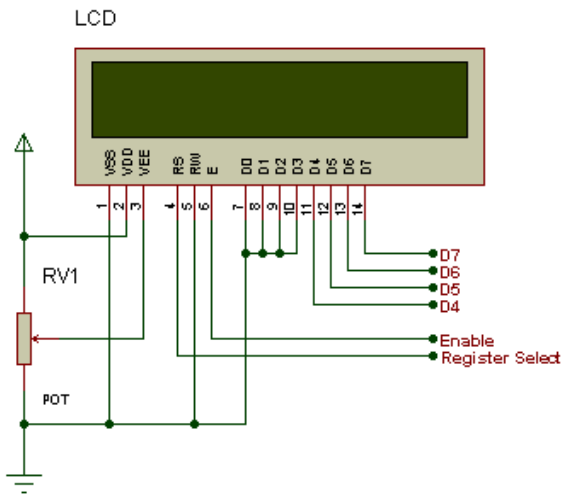


Figura 58: Diagrama de conexión de LCD 16x2 [20]

En este caso, no se utilizará la conexión completa, la librería permite manejar la LCD con la mitad de los pines de control. Esto es una ventaja puesto que disminuye el cableado en la implementación, además deja disponible puertos que pueden utilizarse para otros usos.

La LCD estará encargada de desplegar mensajes del estatus actual del dispositivo de limpieza, por ejemplo:

- Encendido y preparándose
- Lista para uso
- En modo de limpieza automática
- En modo de limpieza programada para Holder
- En modo de limpieza programada para Cap
- Tiempo estimado de finalización, entre otros

11.2. Rutinas

Es necesario definir rutinas de código, las cuales ejecutarán un proceso determinado de limpieza. En sí, definen una secuencia de control específica. Hace falta mencionar que cada secuencia irá acompañada de mensajes pertinentes que se desplegarán en la pantalla LCD. A continuación se hará una descripción breve de la secuencia que cada rutina seguirá, no se caerá en especificaciones de funcionamiento o programación.

11.2.1. Modo automático

El modo *automático* realizará todo el proceso de limpieza por completo. Por lo tanto sigue la secuencia:

1. Activación de servo-motor para asegurar el Cap
2. Activación de bomba para humedecer las cerdas de los cepillos
3. Desactivación de bomba
4. Activación de motor de limpieza de holder
5. Desactivación de motor de limpieza de holder
6. Activación de motor de limpieza de Cap
7. Desactivación de motor de limpieza de Cap
8. Desactivación de servo-motor para liberar Cap
9. Vuelta al menú principal

11.2.2. Limpieza Holder

El modo *Limpieza Holder* realizará todo el proceso de limpieza por completo. Por lo tanto sigue la secuencia:

1. Activación de bomba para humedecer las cerdas de los cepillos
2. Desactivación de bomba
3. Activación de motor de limpieza de holder
4. Desactivación de motor de limpieza de Cap
5. Vuelta al menú principal

11.2.3. Limpieza Cap

El modo *Limpieza Cap* realizará todo el proceso de limpieza por completo. Por lo tanto sigue la secuencia:

1. Activación de servo-motor para asegurar el Cap
2. Activación de bomba para humedecer las cerdas de los cepillos
3. Desactivación de bomba

4. Activación de motor de limpieza de Cap
5. Desactivación de motor de limpieza de Cap
6. Desactivación de servo-motor para liberar Cap
7. Vuelta al menú principal

CAPÍTULO 12

Prototipo

12.1. Diseño de placa de alimentación

12.1.1. Diseño de circuito (Esquemático)

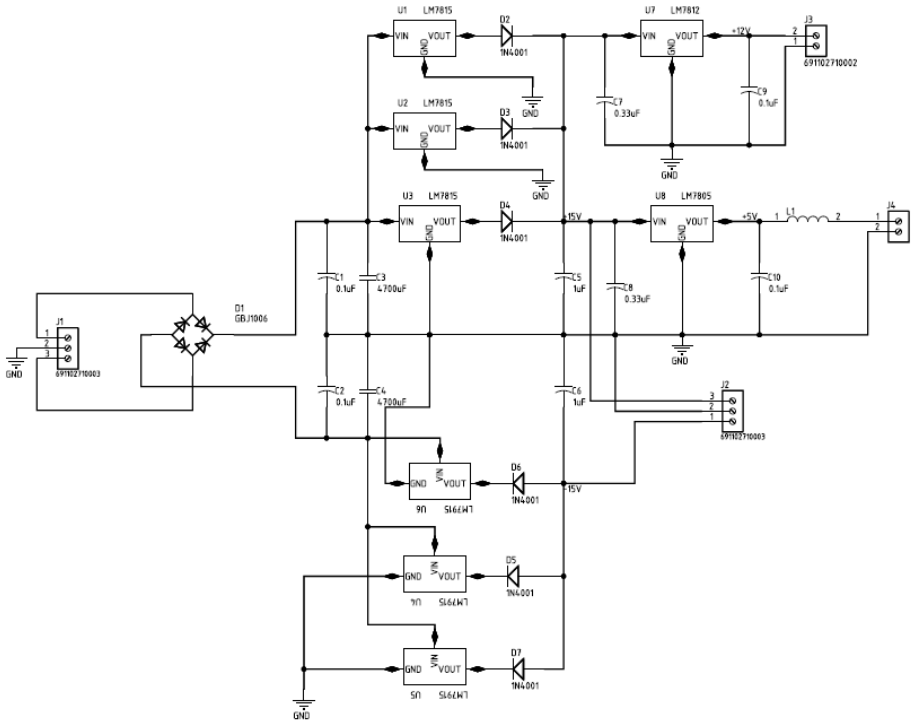


Figura 59: Esquemático de la fuente para placa

Para el diseño del esquemático (diagrama de todos los circuitos en conjunto 59), solamente se requirió añadir las borneras adecuadas para facilitar la interconexión. En este caso, no se incluye el transformador ya que no estará integrado en el PCB.

12.1.2. Diseño de PCB

Cómo se sabe, para el diseño de las placas, fue necesario buscar los modelos exactos de los componentes enlistados. Puesto que dichos modelos poseen los footprints según las características físicas de cada componente.

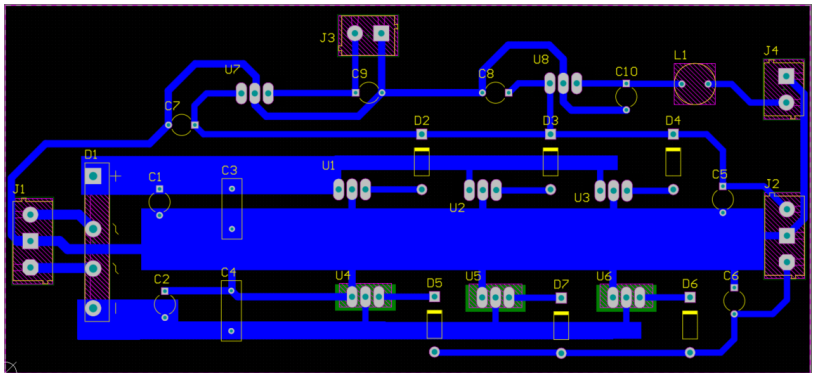


Figura 60: Diseño de placa de alimentación en vista 2D

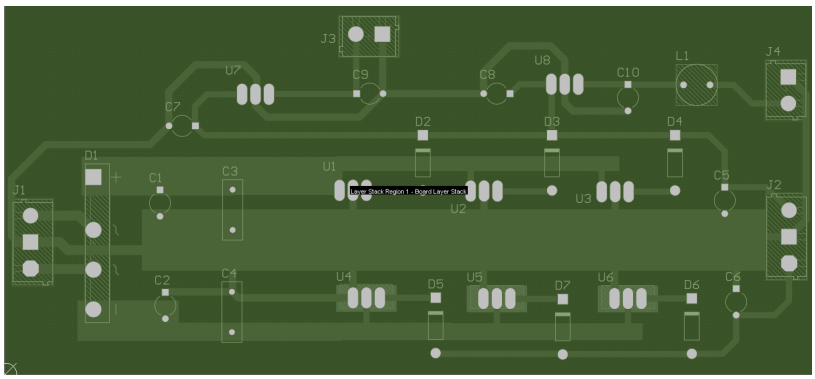


Figura 61: Diseño de placa de alimentación en vista multicapa

En el caso de la Figura 62 y 63, se observa un modelo 3D de la placa, sería más bien como una vista previa. Cabe mencionar que dentro de la búsqueda de los modelos de los componentes, no se priorizó el modelo 3D, es por eso que en algunos casos no se tiene una vista 3D. Generar este modelo permite ver rápidamente algún error en la organización del espacio y además otorga un nivel de profesionalismo más alto al diseño de la placa.

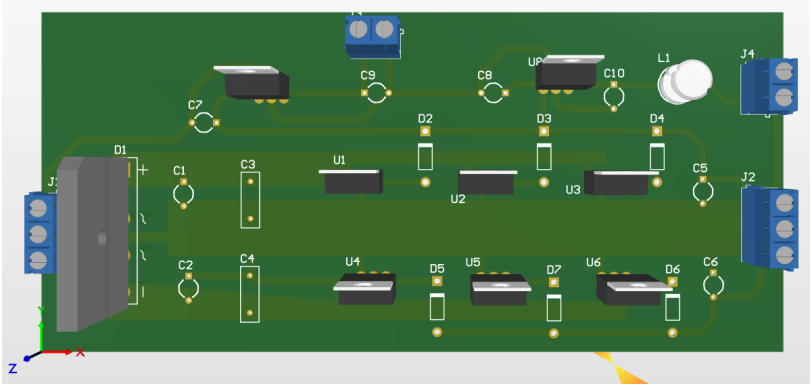


Figura 62: Placa de alimentación en vista 3D desde vista de planta

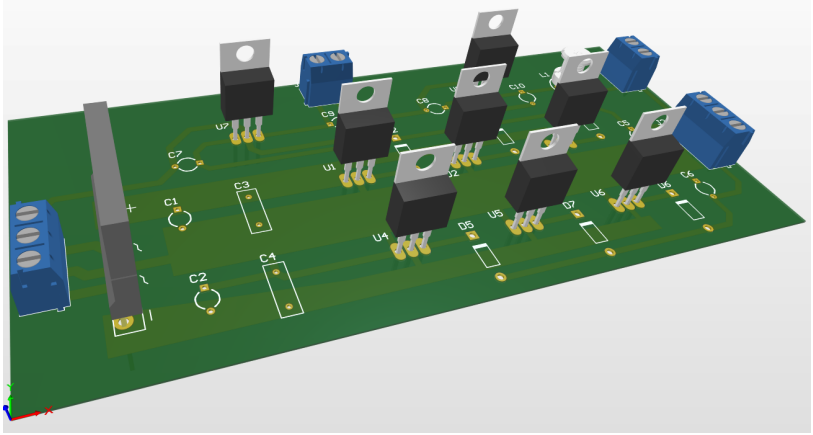


Figura 63: Placa de alimentación en vista 3D desde vista angular

Es importante mencionar que cada uno de los reguladores que se observan en la Figura 63 tendrán un disipador de calor como el de la Figura 64.

Figura 64: Disipador de calor para empaquetado TO-220

12.2. Diseño de placa de control

12.2.1. Diseño del esquemático

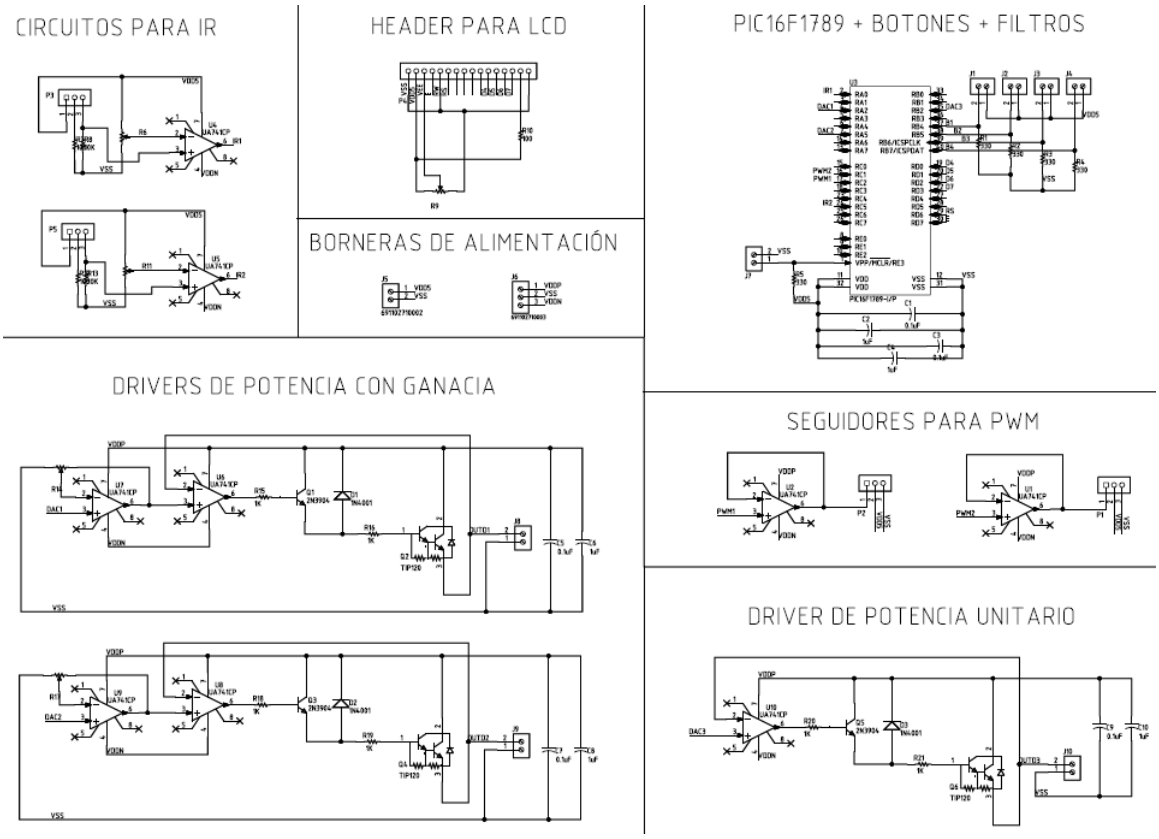


Figura 65: Esquemático de la fuente para placa

La Figura 65 nos indica de entrada que el circuito de esta placa será mucho más grande. Como se podrá visualizar en la Figura 66, se observa ahora un color rojo. La combinación de color azul y rojo indica que el diseño es de una placa de doble cara, donde los tracks rojos son la parte de arriba de la placa y los tracks azules son las conexiones en la parte de abajo de la placa. La Figura 67 permite visualizar todos los niveles y los puntos de interconexión de los componentes con los tracks de la placa. Permite tener una mejor entendimiento de cómo será el funcionamiento eléctrico de este PCB. Se observa una mayoría de color azul en el diseño, esto porque se empleó una técnica de polígonos para aumentar y facilitar el tamaño e interconexión de un nodo en específico.

12.2.2. Diseño del PCB

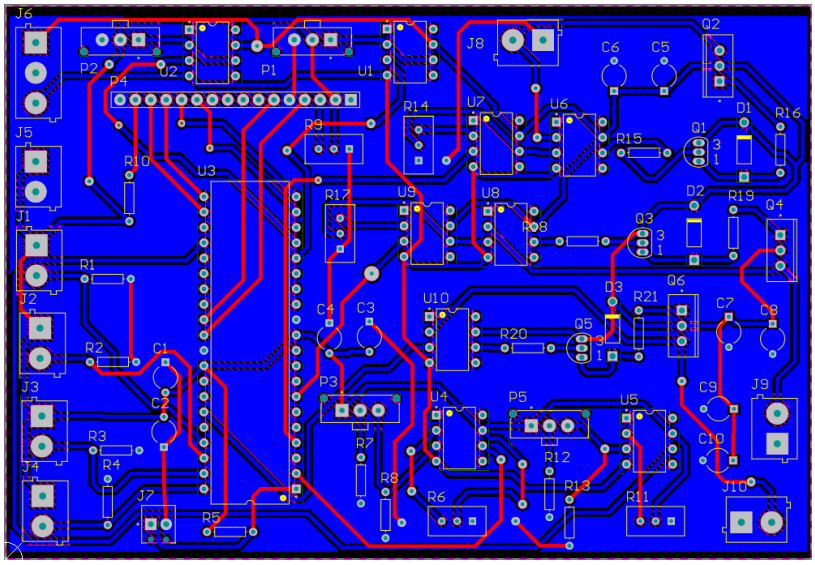


Figura 66: Diseño de placa de control en vista 2D

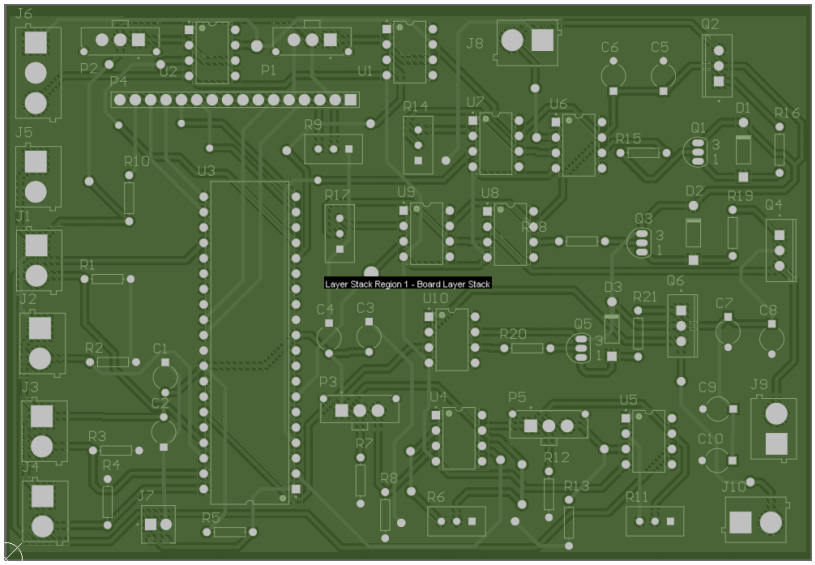


Figura 67: Diseño de placa de control en vista multicapa

Para el caso de los circuitos integrados, que tienen un precio mayor (en especial el PIC16F1789), no se soldará el componente como tal. Se soldarán bases especiales para cada uno de los empaquetados. De tal manera que los circuitos integrados puedan ponerse después de que la placa este finalizada. Esto da dos ventajas, la primera es que no se sobre-calentará innecesariamente al componente. La segunda es que permitirá reemplazar estos componentes de forma mucho más sencilla. Ya sea por el mismo modelo o por otro modelo que cumpla con el empaquetado.

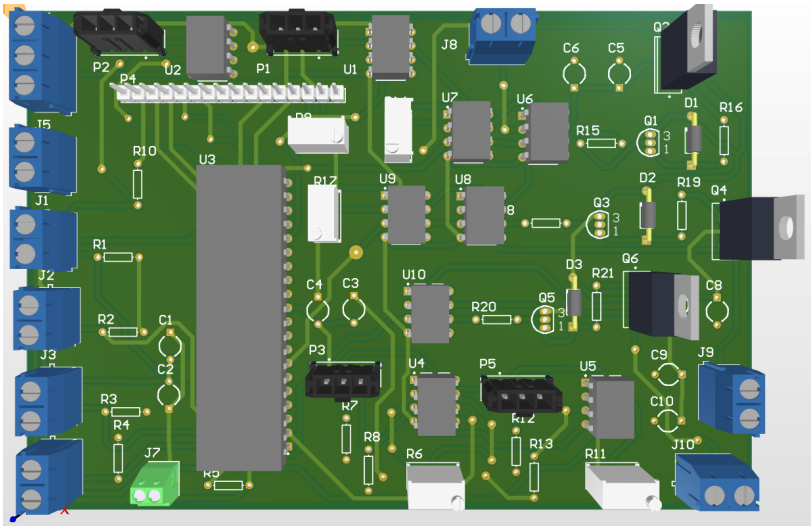


Figura 68: Placa de control en vista 3D desde vista de planta

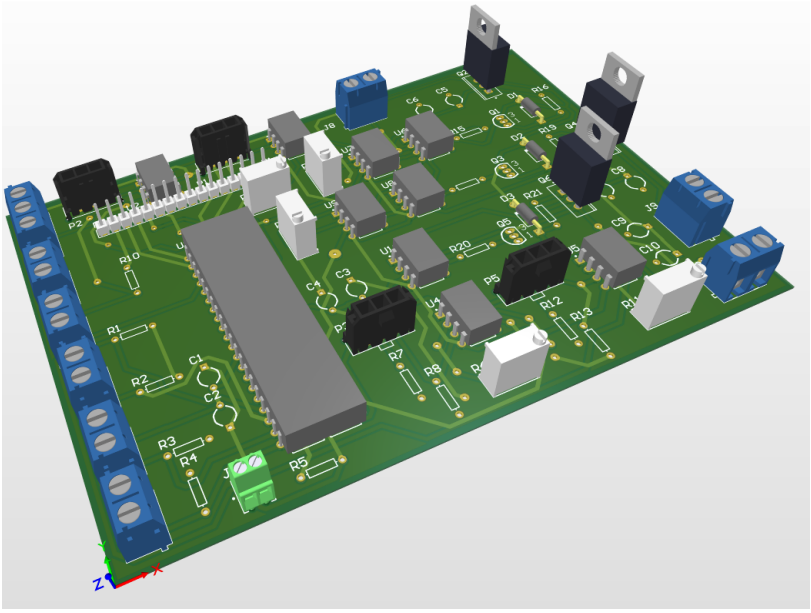


Figura 69: Placa de control en vista 3D desde vista angular

Cabe mencionar que los transistores de potencia (que también son empaquetado TO-220 como los reguladores), tendrán disipador de calor como el de la Figura 64.

Como es de suponer, las placas están destinadas a estar encerradas en la máquina como tal. Todo este calor provocado por la potencia disipada podría afectar a los componentes. A pesar de que los disipadores son suficientes, se agregará un ventilador para crear un flujo de aire que enfríe a los disipadores, a los componentes y al transformador. Este irá conectado al fuente de +12V (ver Figura 26).

CAPÍTULO 13

Conclusiones

1. Se diseñó una fuente de alimentación capaz de entregar distintos voltajes: 15V, -15V, 12V y 5V. En total, la fuente tiene la capacidad de suministrar hasta 3A. Con lo cuál se pueden alimentar todos los motores, incluido un ventilador para enfriamiento de los sistemas de potencia.

2. Se implementó exitosamente una interfaz gráfica a través de una pantalla LCD y botones, que permite al usuario manejar con mayor facilidad el dispositivo de limpieza y conocer el estatus actual del dispositivo.

3. Se implementaron adecuadamente distintos sistemas de control: un controlador PIDF discreto implementado en el PIC16F1789, retroalimentación en los circuito analógicos de potencia y el manejo digital del voltaje óptimo vía el DAC del microcontrolador. Estos sistemas garantizan que no ocurran daños en ninguna de las partes del IQOS.

4. El diseño del sistema automatiza por completo el proceso de limpieza que exige el IQOS 2.4, además requiere de una intervención mínima por parte del usuario final.

5. Se logró que el proceso automatizado cumpliera con el objetivo de efectividad de tiempo, siendo lo suficientemente rápido para el usuario final, alcanzando una mejora de 10 veces en comparación con el método tradicional.

6. Se logró un diseño e implementación óptimo y funcional del sistema eléctrico y electrónico para el dispositivo de limpieza.

CAPÍTULO 14

Recomendaciones

1. Entablar una buena comunicación con la tabacalera para que el desarrollo del proyecto sea el adecuado y puede direccionarse apropiadamente.
2. Reconsiderar el diseño de la fuente de alimentación para una segunda versión del prototipo final, de tal manera que su construcción ocupe un menor espacio o ver la posibilidad de seleccionar una fuente ya existente.
3. Rediseñar los circuitos para que la alimentación no requiera de una fuente simétrica, para facilitar el diseño y/o selección de la fuente de alimentación.
4. Rediseñar los PCBs con componentes de superficie (SMD/SMT), de tal manera que pueda reducirse significativamente el tamaño de la placa de control.

CAPÍTULO 15

Bibliografía

[1] D. Gate, *Uso de Vapesoon con IQOS*. dirección: https://es.dhgate.com/product/vapesoon-automatic-cleaner-with-battery-350mah/424530857.html.

[2] P. M. International, *IQOS 2.4 Plus*. dirección: https://gt.iqos.com/es/producto/iqos/2-4plus-kit.

[3] ——, *IQOS*. dirección: https://gt.iqos.com/es.

[4] ——, *Servicio y Reparación de IQOS 2.4 Plus*. dirección: http://tekhnetshop.mercadoshops.com.ar/servicio-reparacion-de-iqos-24-30-limpieza-repuestos-1052809248xJM.

[5] ——, *IQOS 2.4 Plus Kit*. dirección: https://www.newcigarette.store/p/iqos-device-2-4-plus-white-with-bluetooth-kit.

[6] E. T. S. Combustión, *Como limpiar el IQOS*. dirección: https://eltabacosincombustion.es/como-limpiar-el-iqos/.

[7] A. LLC, *Altium Designer Documentation 19.0*. dirección: https://www.altium.com/documentation/19.0/display/ADES/Altium+Designer+Documentation.

[8] ——, *Manufacter Part Search A19*. dirección: https://www.altium.com/documentation/19.0/display/ADES/IntegratedLibrary_Pnl-ManufacturerPartSearch((Manufacturer+Part+Search))_AD.

[9] M. T. Inc., "MPLAB® X IDE User's Guide", pág. 330, 2015.

[10] ——, "MPLAB XC8 C Compiler User's Guide for PIC MCU", pág. 455, 2018.

[11] ——, "PICkit™ 3 In-Circuit Debugger/Programmer User's Guide", pág. 94, 2013.

[12] ——, "MPLAB Code Configurator v3.xx User's Guide", pág. 37, 2019.

[13] ——, "PIC16F1788/9: 28-Pin 8-bit Advanced Analog Flash Microcontroller", pág. 474, 2015.

[14] T. L. Floyd, "Dispositivos Electrónicos", en *Dispositivos Electrónicos*, Pearson, 2008, pág. 1010.

[15] O. Semiconductor, "TIP12X: Plastic Medium-Power Complementary Silicon Transistors", n.º TIP120/D, pág. 7, 2014.

[16] Mathworks, *Simulink - Simulation and Model-based Design*. dirección: https://www.mathworks.com/products/simulink.html.

[17] ——, *Simscape - Model and simulate multidomain physical sistems*. dirección: https://www.mathworks.com/products/simulink.html.

[18] A. Tecnología, *Fuente de Alimentación*. dirección: https://www.areatecnologia.com/electronica/fuente-alimentacion.html.

[19] F. Martínez, *Tutorial Arduino: Entradas (2): Botones*. dirección: https://openwebinars.net/blog/tutorial-arduino-entradas-2-botones/.

[20] torres.electronico, *tengo un display de 16x2 y no visualizo lectura*. dirección: https://www.yoreparo.com/es/electronica/electronica-digital/preguntas/936654/tengo-un-display-de-16x2-y-no-visualizo-lectura.

CAPÍTULO 16

Anexos

16.1. Programación de PIC16F1789

16.1.1. MCC

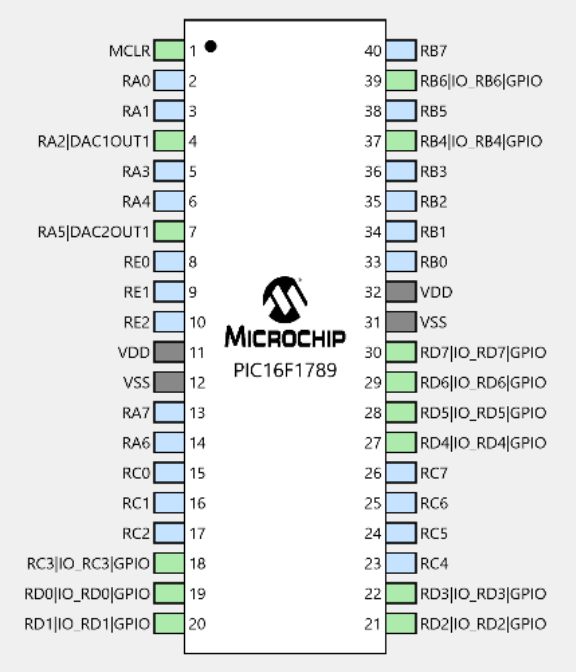


Figura 70: Diagrama de pines en uso del PIC16F1789 en MCC

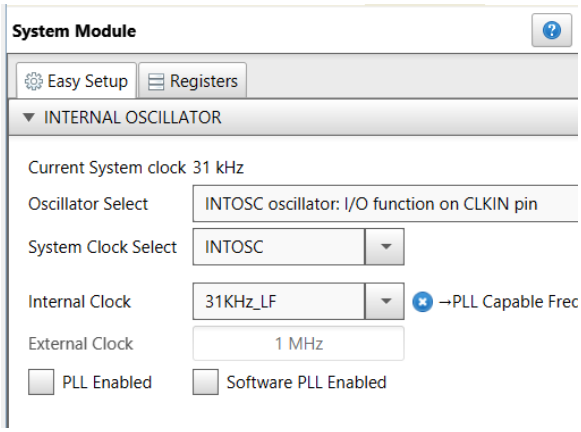


Figura 71: Configuración del reloj interno del PIC16F1789 en MCC

16.1.2. DAC

```c
void DAC1_Initialize(void)
{
    // DAC1EN enabled; DAC1NSS VSS; DAC1PSS VDD; DAC1OE1 enabled; DAC1OE2 disabled;
    DAC1CON0 = 0xA0;
    // DAC1R 255;
    DAC1CON1 = 0xFF;
}

void DAC1_SetOutput(uint8_t inputData)
{
    DAC1CON1  = inputData;
}

uint8_t DAC1_GetOutput(void)
{
    return DAC1CON1;
}
```

Figura 72: Programación de módulo DAC en C

16.1.3. PWM

```c
void PWM1_Initialize(void)
{
    // Set the PWM1 to the options selected in the User Interface

    // CCP1M PWM; DC1B 3;
    CCP1CON = 0x3C;

    // CCPR1L 1;
    CCPR1L = 0x01;

    // CCPR1H 0;
    CCPR1H = 0x00;


}

void PWM1_LoadDutyValue(uint16_t dutyValue)
{
   // Writing to 8 MSBs of pwm duty cycle in CCPRL register
    CCPR1L = ((dutyValue & 0x03FC)>>2);

   // Writing to 2 LSBs of pwm duty cycle in CCPCON register
    CCP1CON = ((uint8_t)(CCP1CON & 0xCF) | ((dutyValue & 0x0003)<<4));
}
```

Figura 73: Programación de módulo PWM en C

16.1.4. Interrupciones

```
void PIN_MANAGER_IOC(void)
{
    // interrupt on change for pin IOCBF4
    if(IOCBFbits.IOCBF4 == 1)
    {
        IOCBF4_ISR();
    }
    // interrupt on change for pin IOCBF6
    if(IOCBFbits.IOCBF6 == 1)
    {
        IOCBF6_ISR();
    }
}
```

```
void IOCBF6_ISR(void) {

    // Add custom IOCBF6 code

    // Call the interrupt handler for the callback registered at runtime
    if(IOCBF6_InterruptHandler)
    {
        IOCBF6_InterruptHandler();
    }
    IOCBFbits.IOCBF6 = 0;
}

/**
  Allows selecting an interrupt handler for IOCBF6 at application runtime
*/
void IOCBF6_SetInterruptHandler(void (* InterruptHandler)(void)){
    IOCBF6_InterruptHandler = InterruptHandler;
}

/**
  Default interrupt handler for IOCBF6
*/
void IOCBF6_DefaultInterruptHandler(void){
    // add your IOCBF6 interrupt custom code
    // or set custom function using IOCBF6_SetInterruptHandler()
}
```

Figura 74: Vector de manejo de interrupciones en C

16.1.5. LCD

```
#ifndef _LCD_XC_H
#define _LCD_XC_H

/*...3 lines */
//#include "mcc_generated_files/mcc.h"

#include <stdint.h>
#include <stdbool.h>

#ifdef __cplusplus
extern "C" {
#endif

/*...3 lines */
// set up the timing for the LCD delays
#define LCD_delay           5     // ~5mS
#define LCD_Startup         15    // ~15mS

// Command set for Hitachi 44780U LCD display controller
#define LCD_CLEAR           0x01
#define LCD_HOME            0x02
#define LCD_CURSOR_BACK     0x10
#define LCD_CURSOR_FWD      0x14
#define LCD_PAN_LEFT        0x18
#define LCD_PAN_RIGHT       0x1C
#define LCD_CURSOR_OFF      0x0C
#define LCD_CURSOR_ON       0x0E
#define LCD_CURSOR_BLINK    0x0F
#define LCD_CURSOR_LINE2    0xC0
```

```
// display controller setup commands from page 46 of Hitachi datasheet
#define FUNCTION_SET        0x28                        // 4 bit interface, 2 lines, 5x8 font
#define ENTRY_MODE          0x06                        // increment mode
#define DISPLAY_SETUP       0x0C                        // display on, cursor off, blink offd

#define LCDLine1()          LCDPutCmd(LCD_HOME)         // legacy support
#define LCDLine2()          LCDPutCmd(LCD_CURSOR_LINE2) // legacy support
#define shift_cursor()      LCDPutCmd(LCD_CURSOR_FWD)   // legacy support
#define cursor_on()         LCDPutCmd(LCD_CURSOR_ON)    // legacy support
#define DisplayClr()        LCDPutCmd(LCD_CLEAR)        // Legacy support


//...6 lines
#define instr       0
#define data        1

// These #defines create the pin connections to the LCD in case they are changed on a future demo board
#define LCD_PORT    PORTD
#define LCD_PWR     PORTDbits.RD4                       // LCD power pin
#define LCD_EN      PORTDbits.RD7                       // LCD enable
#define LCD_RW      PORTDbits.RD5                       // LCD read/write line
#define LCD_RS      PORTDbits.RD6                       // LCD register select line

#define NB_LINES    2                                   // Number of display lines
#define NB_COL      16                                  // Number of characters per line

// Function prototypes
/*...17 lines */
void LCD_Initialize(void);
```

```
// Function prototypes
/*...17 lines */
void LCD_Initialize(void);

 /*...16 lines */
void LCDPutChar(uint8_t ch);

/*...16 lines */
void LCDPutCmd(uint8_t ch);

/*...16 lines */
void LCDPutStr(const char *);

/*...16 lines */
void LCDWriteNibble(uint8_t ch,uint8_t rs);

/*...20 lines */
void LCDGoto(uint8_t pos, uint8_t ln);

#ifdef __cplusplus
}
#endif

#endif  /* _LCD_XC_H */
```

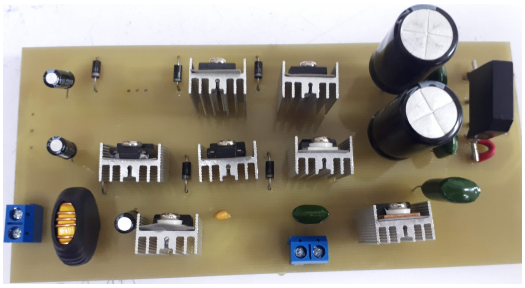
Figura 75: Header File para manejo de LCD

```c
#define _XTAL_FREQ 8000000

#include <xc.h>
#include "lcd.h"

void LCD_Initialize()
{
    // clear latches before enabling TRIS bits
    LCD_PORT = 0;
    TRISD = 0x00;
    // power up the LCD
    LCD_PWR = 1;
    // required by display controller to allow power to stabilize
    __delay_ms(LCD_Startup);
    // required by display initialization
    LCDPutCmd(0x32);
    // set interface size, # of lines and font
    LCDPutCmd(FUNCTION_SET);
    // turn on display and sets up cursor
    LCDPutCmd(DISPLAY_SETUP);
    DisplayClr();
    // set cursor movement direction
    LCDPutCmd(ENTRY_MODE);
}

void LCDWriteNibble(uint8_t ch,uint8_t rs)
{
    // always send the upper nibble
    ch = (ch >> 4);
    // mask off the nibble to be transmitted
    ch = (ch & 0x0F);
    // clear the lower half of LCD_PORT
    LCD_PORT = (LCD_PORT & 0xF0);
    // move the nibble onto LCD_PORT
    LCD_PORT = (LCD_PORT | ch);
    // set data/instr bit to 0 = insructions; 1 = data
    LCD_RS = rs;
    // RW - set write mode
    LCD_RW = 0;
    // set up enable before writing nibble
    LCD_EN = 1;
    // turn off enable after write of nibble
    LCD_EN = 0;
}

void LCDPutChar(uint8_t ch)
{
    __delay_ms(LCD_delay);
    //Send higher nibble first
    LCDWriteNibble(ch,data);
    //get the lower nibble
    ch = (ch << 4);
    // Now send the low nibble
    LCDWriteNibble(ch,data);
}


void LCDPutCmd(uint8_t ch)
{
    __delay_ms(LCD_delay);
    //Send the higher nibble
    LCDWriteNibble(ch,instr);
    //get the lower nibble
    ch = (ch << 4);
    __delay_ms(1);
    //Now send the lower nibble
    LCDWriteNibble(ch,instr);
}


void LCDPutStr(const char *str)
{
    uint8_t i=0;
     // While string has not been fully traveresed
    while (str[i])
    {
        // Go display current char
        LCDPutChar(str[i++]);
    }

}

void LCDGoto(uint8_t pos,uint8_t ln)
{
    // if incorrect line or column
    if ((ln > (NB_LINES-1)) || (pos > (NB_COL-1)))
    {
        // Just do nothing
        return;
    }
    // LCD_Goto command
    LCDPutCmd((ln == 1) ? (0xC0 | pos) : (0x80 | pos));
    // Wait for the LCD to finish
    __delay_ms(LCD_delay);
}
```

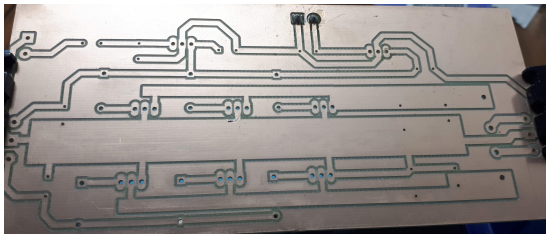

Figura 76: Source File para manejo de LCD

16.2. Prototipo final

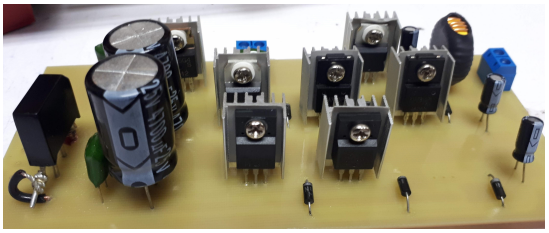
16.2.1. Placa final



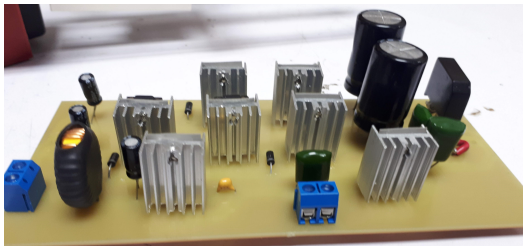
(a) Vista superior



(b) Vista de abajo



(c) Vista frontal



(d) Vista trasera

Figura 77: Vistas del PCB final de alimentación

CAPÍTULO 17

Glosario

AD : Altium Designer. 14

BJT : Bipolar Junction Transistor. 22

Cap : Nombre que coloquialmente se ha denominado para designar el cabezal del Holder. 9, 13, 14, 33, 51, 65–67

CPU : Central Processing Unit. 18

Datasheet : El datasheet es la hoja de datos que describe todas las características físicas y de operación de un objeto en especial. 15

Debugger : Un *debugger* es esencialmente un depurador que permite identificar y remover errores. A la acción y efecto de hacerlo se le conoce cómo *debugging*. 17

DIP : Dual In-line Package. 19

Driver : Circuito o componente que se encarga de controlar otro circuito o componente para obtener una característica deseada. 20–22, 37

Función de transferencia : Una función de transferencia es una función que modela la salida de un sistema de ecuaciones diferenciales. Generalmente se escribe en el dominio de la frecuencia (s) ya que se hace uso de la *Transformada de Laplace*, pero también puede escribirse en el dominio del tiempo (t). En circuitos, la función de transferencia representa el comportamiento del sistema. 24–26

Heet : Cigarrillo especial para dispositivos IQOS, más corto y con tabaco comprimido. 12

Holder : El IQOS como tal, el dispositivo en el cuál se inserta el Heet y que calienta el tabaco en él. 9, 12, 13, 51, 55, 62, 65, 87

MCC : MPLAB Code Configurator. 17

MPS : Manufacter Part Search. 15

Op-Amp : Operational Amplifier. 21, 22

PCB : Printed Circuit Boards, por sus siglas en inglés. Son las tarjetas de circuitos impresos, comúnmente conocidas como *placas*. 14

Periférico : Son dispositivos externos que pueden ser controlados a través de un microcontrolador o computadora y de los cuales se obtiene una entrada, se modifica su comportamiento o ambas cosas. 17

PIC : Programmable Interrupt Controller. 16, 17

Plugin : En programas de computadora, un plugin es un complemento al programa original que permite añadir una nueva característica. Usualmente esta característica añadida. 17

PMI : Phillip Morris International. 11

Pocket Charger : Cargador del IQOS 2.4 Plus. 12

PWM : Pulse-Width Modulation. 19

Rectificador : En circuitos eléctricos, un rectificador es un puente de diodos que rectifica una señal sinusoidal por completo y la la vuelve una señal constante (DC). Este rectificador de onda, aprovecha también la parte negativa. El valor de voltaje constante será un tanto mayo al voltaje RMS del transformador en cuestión. 42

RISC : Reduced Instruction Set Computing. 18

Saturación : En Op-Amps y transistores, el modo saturación se refiere a llevar a los límites al componente, esto se traduce a que la salida tendrá la misma magnitud que el voltaje de alimentación. 21

Script : En programación, un «script» es un documento en donde se ha escrito, en un lenguaje determinado, una serie de ejecuciones que tienen un fin específico. Suele tener la extensión correspondiente al lenguaje elegido. 25

SISO : Por sus siglas en inglés, Single Input Single Output. Se utiliza para denominar a los sistemas de control que solo requieren de una entrada para controlar la salida. 23

TIP : Texas Instruments Power. 22

Wizard : En programas de computadora, el wizard es un asistente para configurar determinadas opciones. 15