**Memoria de diseño**

**[Ingeniero: Jhoan Esteban León]**

**[Proyecto: Controlador de Salidas]**

El presente documento muestra los principales cálculos y procedimientos llevados a cabo para la realización del proyecto llamado **Controlador de Salidas** solicitado en la prueba técnica para el puesto de **Ingeniero de Sistemas Embebidos** en **TECREA**. Este documento no pretende ser una guía técnica acerca del diseño del sistema, sino más bien una explicación ligera (sin lenguaje demasiado técnico) de las decisiones tomadas y los componentes/procesos seleccionados.

El proyecto se organiza en 4 carpetas autodescriptivas “documentation”, “hardware”, “firmware” y “simulation”. En cada una se podrán encontrar los archivos respectivos a su categoría destacando los del proyecto EAGLE del diseño esquemático y PCB del circuito que se encuentra en la carpeta “hardware”. Este documento de memoria de diseño se encontrará en la carpeta de “documentation” junto a los datasheets de los componentes utilizados.

Adicionalmente, se utilizó el software **git** de control de versiones a nivel local para llevar un seguimiento de los cambios en todos los archivos y del avance en cada etapa del proyecto.

**Hardware:**

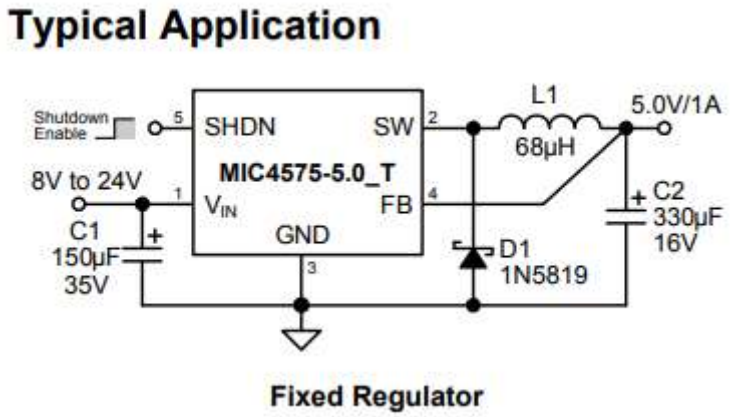
Para el diseño del HW se eligió un DAC para la generación de las señales análogas del sistema (referencia **DAC102S085** de *TI*). Este DAC posee una **resolución de 10bits** cómo lo solicitaba la especificación del proyecto. El **rango de trabajo es de 0V-5V** debido a su cualidad rail-to-rail y una corriente de salida en continuo de 11mA. Es importante destacar que el **ZERO CODE ERROR es de 5mV**. Dado que la salida de voltaje del DAC es limitado, se decidió utilizar un **OPAMP LM6211** de bajo ruido y con capacidad rail-to-rail en configuración de **amplificador no-inversor de ganancia 2** para obtener el **rango de operación de 0V-10V**. Este OPAMP puede entregar entre 20 y 30mA por lo que las cargas conectadas a las salidas análogas del sistema no deben requerir mucha corriente.

$$Resolucion\_Salidas\_Analogas = \frac{10V - 0V}{2^{10} - 1} = \frac{10V}{1023} \approx 9{,}78mV$$

Para las señales de PWM se eligieron MOSFETS canal n de potencia (referencia **IRF1010NS** de *International Rectifiers*). Estos MOSFETS pueden soportar un voltaje **VDSS de hasta 55V** y una **corriente de drain ID de hasta 85A** por lo que cumple ampliamente con la especificación de operación dada entre 0V-24V de conmutación. Adicionalmente, dado esta capacidad de los dispositivos y el empaquetado utilizado en el diseño de la PCB, las salidas de PWM pueden manejar cargas significativamente altas ($\approx 48W$). Las compuertas (gates) de los MOSFETS se conectaron a las salidas del microcontrolador mediante un divisor resistivo que garantiza el modo de saturación en la operación de los transistores que conmutan las señales de PWM.

$$V_{GS\ activo} = 5V * \frac{R2}{R1 + R2} = 5V * \frac{100K}{110K} \approx 4{,}54V$$

Finalmente, el sistema se diseñó de tal manera que el uso de fuentes de alimentación internas para regulación se viera disminuido (alimentando los principales circuitos a 24V directamente) solo requiriendo una fuente tipo **buck** (también llamadas **step-down**) de 5V@1A basada en el regulador **MIC4575** de *Microchip*. Esta fuente toma como entrada la especificación dada de 24V@2A y la transforma en un nivel para alimentar los circuitos lógicos del sistema que son principalmente el microcontrolador PIC16F1939 y el DAC. La configuración, los componentes y sus valores se tomaron de la recomendación dada por el fabricante en su hoja de datos (ver imagen a continuación). Adicionalmente, se usaron capacitores de filtrado a la entrada y salida del circuito de alimentación.



**Firmware:**

El firmware para el microcontrolador PIC16F1939 se desarrolló en el IDE *CCS C Compiler* y se simuló (en gran parte, destacando la comunicación serial) con el software *Proteus 8 Professional*. El diseño es una aplicación en **bare-metal**, es decir, sin ningún **RTOS**. Únicamente se utiliza la capacidad del procesador y las interrupciones para una ejecución **foreground-background**.

Así pues, el primer proceso de configuración detallado es la generación de la base tiempo para las señales de PWM. Aquí se tenía la especificación de poder generar señales entre 0Hz-50Hz. Para esto se decidió primero un reloj de cristal externo de **20MHz**. Después, se decidió utilizar el TIMER2 del microcontrolador para contar con precisión tiempos de **15,25us**. Esta base de tiempo se decidió teniendo en cuenta los siguientes parámetros:

- Rango máximo de valores para contar con 16bits: $2^{16} - 1 = 65535$
- Frecuencia y Periodo de PWM mínimos (aparte de 0Hz): $f_{min} = 1Hz$ , $T_{max} = 1seg$

$$Base\ de\ tiempo = \frac{1seg}{65535} = 15{,}25us$$

Una vez seleccionada la base de tiempo se procedió a configurar con el TIMER2 con el valor del periodo calculado con la fórmula del datasheet que es la siguiente:

$$PRx = \frac{T_{base}}{4 * Preescaler * T_{osc}} - 1$$

Teniendo en cuenta los parámetros de la ecuación, se eligió el Preescaler de 1 quedando:

$$PRx = \frac{15{,}25us}{4 * 1 * \frac{1}{20MHz}} - 1 = 75{,}25 \cong 75$$

Así se configuró este valor en el registro de periodo del TIMER2 para generar una base de tiempo real de **15,20us**. Con esta base, se puede contar la cantidad de ciclos necesarios (dentro del rango de 0-65535) para generar con precisión señales de PWM con frecuencias entre 0Hz-50Hz como estaba especificado.

La configuración de los periféricos de comunicación (UART y SPI) se manejó de manera estándar sin parámetros especiales. El funcionamiento del DAC es mediante SPI por una sola transacción de 16bits por canal de acuerdo a la indicación del datasheet (ver imagen a continuación). De esta manera, los comandos recibidos por el serial se transmiten directamente (en 10bits) al DAC dependiendo del canal que se quiera seleccionar cambiando los primeros 4bits de la transmisión.

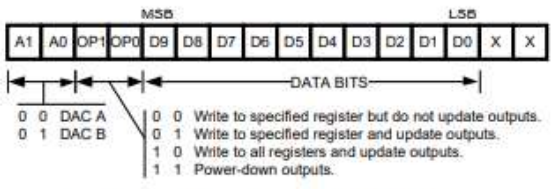



**Figure 32. Input Register Contents**

***Nota:** Los 2 LEDs del sistema se utilizaron para debugging o visualización durante el desarrollo.