**MANUAL MPLAB HARMONY**

MPLAB HARMONY es una interfaz de desarrollo de software integrado que proporciona módulos flexibles para simplificar el desarrollo de características de valor agregado para dispositivos Microchip de 32 bits, este se compone de un conjunto de herramientas, dispositivos modulares, bibliotecas y aplicaciones de ejemplo que simplifican el desarrollo de software para microcontroladores y microprocesadores. A continuación se presentan las diferentes configuraciones realizadas en MPLAB HARMONY para llevar a cabo el proyecto:

* **Configuración de QEI**

En la figura 1 y figura 2 se puede observar la configuración básica de los QEI, los cuales son necesarios para realizar la comparación entre el canal 1 y canal 2 del encoder de cada uno de los motores y así establecer la velocidad en ellos. En MPLAB Harmony se realizan estos cálculos internamente, y así mismo se tiene la posibilidad de cambiar ciertos parámetros según lo requiera el usuario; en este caso se utiliza la configuración predeterminada del programa para los QEI1,QEI2 y QEI3.

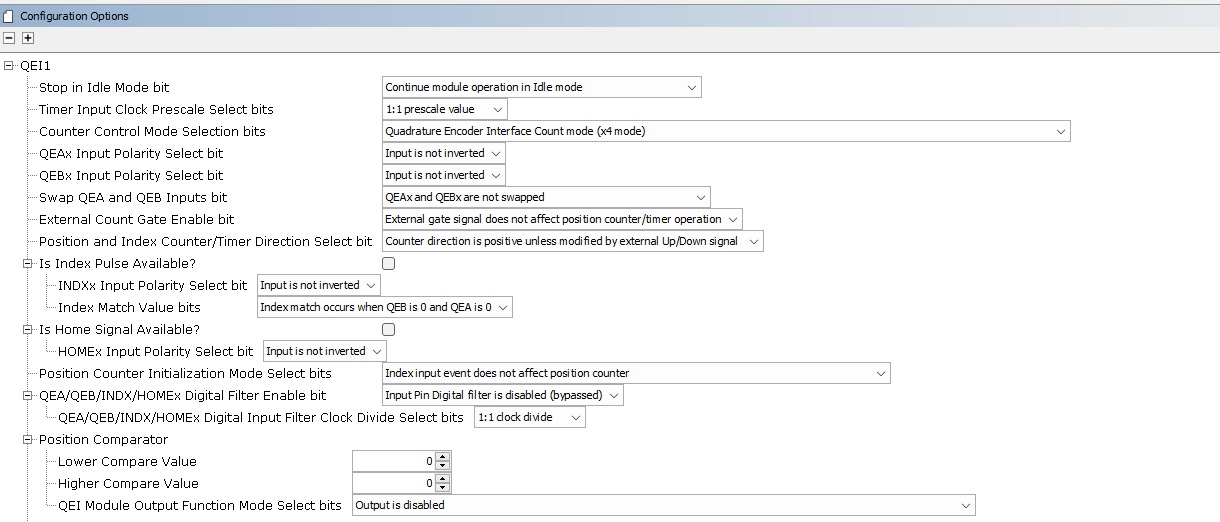
****

Figura 1. Configuración de QEI (Parte 1)

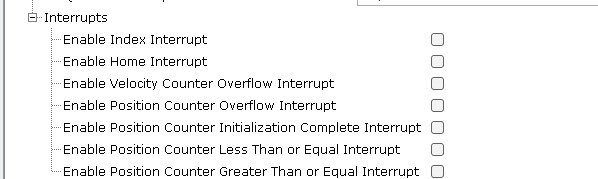


Figura 2. Configuración de QEI (Parte 2)

* **Configuración de canales PWM**

Para ingresar a la configuración de los canales PWM es necesario seleccionar MCPWM-> PWM Configurations .En la figura 3 se puede observar la configuración básica del modulador de ancho de pulso de control de motor (MCPWM), en esta sección se pueden modificar ciertos parámetros como los son el control base de tiempo primario PWM, temporizador de base, comparación de eventos especiales principal y secundario, entre otros, en este caso se utiliza la configuración predeterminada del programa para los PWM y no se modifica ningún parámetro.

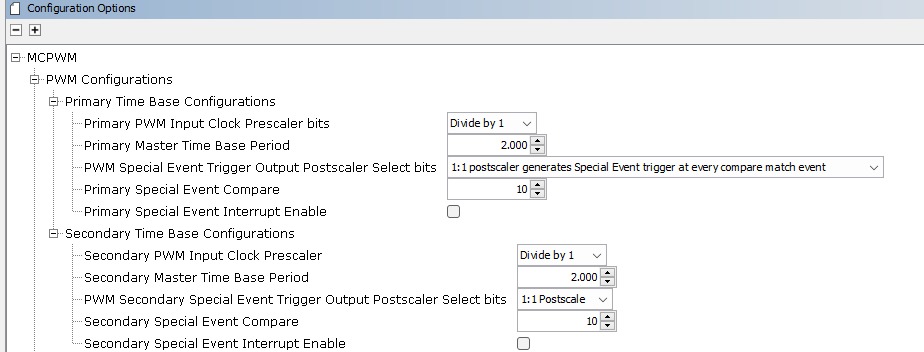
****

Figura 3. Configuración del modulador de ancho de pulso de control de motor (MCPWM)

En la figura 4, se presentan las configuraciones detalladas del canal PWM, en este caso es necesario habilitar el enable de los canales que se utilizan, para ello se selecciona el Channel 2, Channel 6 y Channel 8 (Figura 5); cabe recalcar que no se modifica ningún otro parámetro.

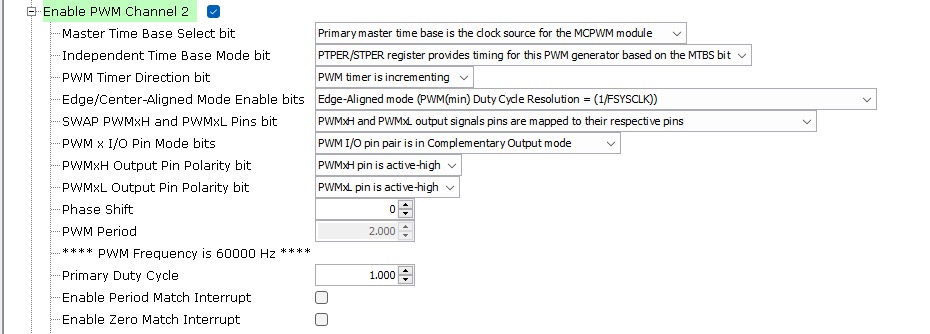


Figura 4. Configuraciones detalladas del enable PWM del canal 2.

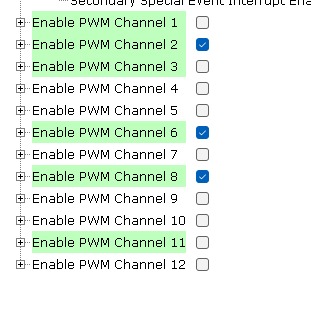


Figura 5. Selección del enable PWM del Channel 2, Channel 6 y Channel 8.

* **Configuración básica del CoreTimer**

En la figura 6 se observa la configuración del CoreTimer, este corresponde a un temporizador central, el cual incrementa a medida que se ejecutan las instrucciones, en este caso no se modifica ningún parámetro y se utiliza la configuración predeterminada.

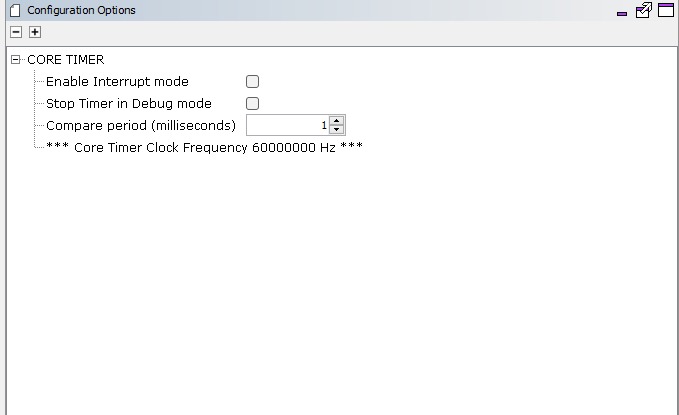
****

Figura 6. Configuración del CoreTimer.

* **Configuración básica del TMR2**

Se presenta en la figura 7 la configuración del TMR2, el cual se utiliza para realizar el proceso de interrupción, en esta sección se establece el tiempo máximo en el Timer interno que corresponde a 1.092283 ms y se establecen como predeterminados los demás parámetros.

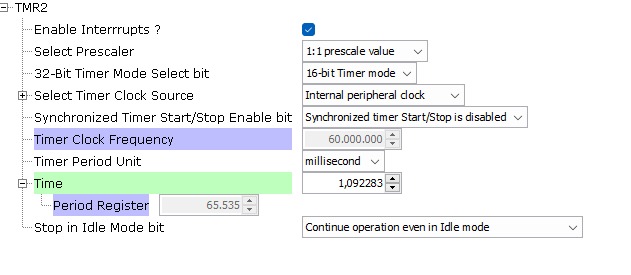


Figura 7. Configuración básica del TMR2

* **Configuración del PIC32MK1024MCM100**

Para realizar la configuración del PIC32MK1024MCM100 es necesario seleccionar System->Device & Project-> PIC32MK1024MCM100 Device Configuration->DEVCFG0, en esta sección se habilita la opción **Background Debugger Enable bits (DEBUG)**, en el parámetro In-Circuit Emulator/Debugger Communication Channel Select bits (ICESEL) se selecciona **ICS\_PGx2** y por último en la opción DEbug Mode CPU Access Permission (DEGEPER) se seleccionan los permisos de acceso **PG\_2\_0.** Esta configuración se presenta en la Figura 8.

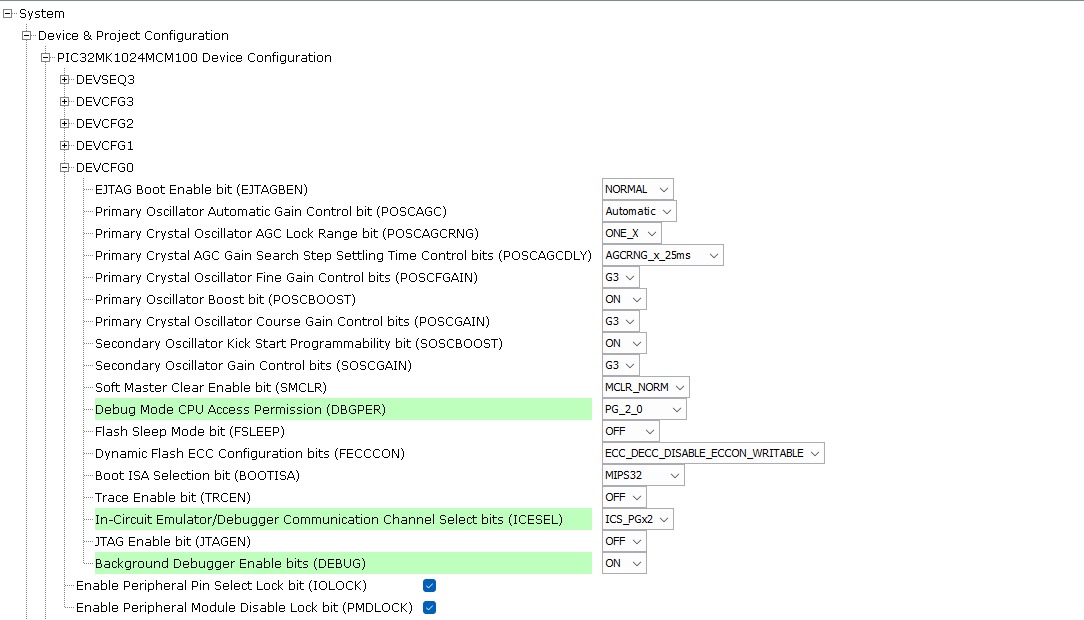


Figura 8. Configuración del PIC32MK1024MCM100.