Práctica Número 4:

"Control de Velocidad de un Motor de DC"

Objetivo

Controlar la velocidad de un motor de DC usando PWM.

Material y equipo para la práctica

1 Multímetro.
1 PC.
1 Tarjeta DEMOJM60.
1 Microcontrolador MC9S08JM60.
2 transistores NPN TIP 41.
1 resistencia de 1KΩ
2 Diodos 1N4007.
1 Motor de CD de 9 Volts.
1 Protoboard.
Cables.

Cuestionario Preliminar

1. ¿Qué tipos de motores eléctricos conoce?
2. ¿Qué es una señal PWM?
3. ¿Qué es ciclo de trabajo (Duty cicle)?
4. ¿Cómo se define el periodo de una señal?
5. ¿Cómo se define la frecuencia de una señal?

Introducción

Se tiene la impresión de que cuando se habla de motores eléctricos, se está haciendo referencia a grandes motores empleados principalmente en la industria, entonces, de acuerdo con el conocimiento que se tiene de ellos por el tipo de corriente con la que operan, se piensa que la mayoría consume corriente alterna. Si se consideran las múltiples aplicaciones que tienen los motores eléctricos, tanto en el hogar, como en la oficina y en distintas aéreas de la industria, se encontrarán que los motores de corriente directa se utilizan, con pequeñas potencias, en gran variedad de casos, por ejemplo, en juguetes, aparatos del hogar (licuadoras, batidoras, extractores, etc.), equipos de oficina y cómputo (impresoras de carro y de tipo láser, fotocopiadoras, etc.). En robótica se encuentra también un número importante de aplicaciones, y así como otros usos se tienen en medicina y en equipos dentales. En general, se puede establecer que en nuestra vida diaria usamos motores eléctricos grandes y pequeños; en particular, pequeños, los cuales se deben fabricar en gran cantidad.

Un motor de corriente directa cuenta con dos conexiones. La corriente eléctrica es proporcionada a través de estas, y por dentro fluye por cables que forman un electroimán. Este electroimán un

campo magnético que reacciona contra imanes permanentes ubicados alrededor del cable, logrando que la armadura comience a girar.

La velocidad de un motor DC puede ser controlado, gracias a una técnica llamada Modulación por Ancho de Pulso. Esto se logra prendiendo y apagando el motor de forma rápida y repetitivamente. La clave es el ciclo de trabajo (duty cycle), que es definido por el porcentaje de tiempo encendido contra el de tiempo apagado. Por ejemplo, si la corriente es proporcionada solo la mitad del tiempo, entonces el motor gira a sólo el 50% de su operación máxima. Al realizar estos cambios rápidamente el motor aparenta funcionar más lentamente sin detenerse.

Funcionamiento

Contadores y temporizadores

El microcontrolador MC09S08JM60 incorpora dos: módulos hardware para realizar las funciones de contador y temporizador estos son el Contador de Tiempo Real (Real Time Counter, RTC) y el temporizador/Modulador de Anchura de Pulsos (Timer/Pulse-Width Modulator, TPM).

Temporizador/PWM

El microcontrolador MC09S08JM60 dispone de dos temporizadores/PWM, TMP[1:2], de los cuales el TPM1 tiene 6 canales y el TPM2, poseen 2 canales CH[0:1].

Cada canal de un TPMx puede configurarse en diferentes modos:

- Modo de captura

Este modo permite obtener los momentos en los que se produce algún cambio en una determinada señal digital de entrada.

- Modo de comparación

Este modo permite generar una señal de salida digital cuyo nivel cambiará, según desee el programador, trascurrido un tiempo configurable.

- Modo de modulación por anchura de pulsos (PWM) alineados a flancos

Este modo permite generar señales digitales de periodo y ciclo de trabajo configurables. Este tipo de señal PWM generada se llama alineada a flancos, es decir, los primeros flancos de todas las señales se alinean con el comienzo del periodo.

- Modo de modulación por anchura del pulsos (PWM) centralmente alineados

También se pueden generar señales digitales con periodo y ciclos de trabajo configurables, pero ahora los centros de los periodos activos del ciclo de trabajo están con el centro de la señal de generada.

Inicialmente, se puede separar el funcionamiento de los canales de cada TPM en dos, dependiendo del estado en que se encuentre el bit 5, CPWMS, del registro de estado y control del TPMx, TPMxSC:

- ***Registro TPMxSC***

Registro de estado y control (TPMxSC): se muestra la configuración del registro de estado y control del TPM.

	Bit 7	6	5	4	3	2	1	Bit 0
Lectura Escritura	TOF 0	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
Reset	0	0	0	0	0	0	0	0

TOF

Bandera de sobre flujo del contador de 16 bits. Esta bandera se pone a "1" cuando se ha alcanzado el valor de 0x0000, superando el valor programado en el registro del módulo del contador.
Para poner el bit a cero del la bandera TOF es necesario leer el registro TPMSC y luego escribir un "0" en el bit TOF.
0: El contador del TPM no ha alcanzado el sobre flujo
1: El contador ha alcanzado un sobre flujo

TOIE

Bit para habilitar un evento de interrupción.
Cuando la bandera TOF es "1" y el bit TOIE="1", el sistema genera un evento de interrupción por sobre flujo del contador del TPM.
0: Para detectar un evento de sobre flujo es necesario hacer ***polling*** sobre TOF
1: Habilita un evento de interrupción cuando TOF = "1"

CPWMS

Habilita que todos los canales del TPM actúen como PWM alineado en el centro del período (*center align*). El objetivo es disminuir el ruido en las conmutaciones del pin de salida y el contador que trabaja en modo *up/down*.
0: No está activa la opción de alineado al centrado
1: Activa opción de PWM alineado al centro

CLKS

Selecciona la fuente de reloj del contador del TPM.
00: Módulo inactivo
01: Reloj del bus interno
10: Reloj fijo del sistema (sólo para opción con circuito PLL)
11: Reloj externo

PS

Selección del divisor de la fuente de reloj.
000: Divisor por 1
001: Divisor por 2
010: Divisor por 4
011: Divisor por 8
100: Divisor por 16
101: Divisor por 32
110: Divisor por 64
111: Divisor por 128

- ***Registro contador del TPM (TPMxCNTH:TPMxCNTL)***

Está configurado por dos registros de 8 bits. Se muestran los registros que conforman el contador de 16 bits del TPM. La acción de escribir en cualquiera de los dos registros hace que se ponga a "0" el contador de 16 bits.

TPMxCNTH (Parte Alta)

	7	6	5	4	3	2	1	0
Lectura	15	14	13	12	11	10	9	Bit 8
Escritura	Escribir un valor en este registro pone a cero el contador de 16 bits							
Reset	0	0	0	0	0	0	0	0

TPMxCNTL(Parte Baja)

	7	6	5	4	3	2	1	0
Lectura	7	6	5	4	3	2	1	Bit 0
Escritura	Escribir un valor en este registro pone a cero el contador de 16 bits							
Reset	0	0	0	0	0	0	0	0

Los bits de los registros implicados en la configuración son el bit 5, CPWMS, del registro de estado y control el TPMx, TPMxSC, que como se observa en **la tabla 1.4** determina si el modo es PWM centralmente alineado (CPWMS = 1) o cualquier otro modo (CPWMS = 0), y los bits 5 y 4, MSnB y MSnA, así como los bits 3 y 2, ELSnB y ELSnA, del registro de estado y control del canal ***n*** del TPMx, TPMXCnSC.

Tabla 1.4 Configuración de los distintos tipos PWM.

CPWMS	*MSnB:MSnA*	*ELSnB:ELSnA*	*Modo*	*Configuración*
x	xx	00	Módulo TPM deshabilitado	
0	00	01	INPUT CAPTURE	Captura en flanco de subida
		10		Captura en flanco de bajada
		11		Captura en flanco de subida o bajada
	01	01	OUTPUT COMPARE	Cambio estado de pin en OUTPUT COMPARE
		10		Poner pin en 0 en OUT COMPARE
		11		Poner pin en 1 en OUT COMPARE
	1x	10	PWM ALINEADO AL FLANCO	Comienza en alto y cae en OUT COMPARE
		X1		Comienza en bajo y sube en OUT CAMPARE
1	xx	10	PWM ALINEADO AL CENTRO	Comienza en alto y cae en OUT COMPARE
		x1		Comienza en bajo y sube en OUT CAMPARE

Obsérvese que, independientemente de los valores que se hayan configurado para los bits CPWMS, MSnB y MSnA, cuando los bits ELSnB y ELSnA son cero, el canal ***n*** que se está usando del TPMx correspondiente no puede disponer del pin de entrada/salida que tiene asociado, TPMxCHn, que quedará libre para uso como entrada/salida de propósito general del microcontrolador.

Por otro lado, si CPWMS = 0, se puede seleccionar el canal para trabajar en:

- Modo captura

- Modo comparación
- Modo modulación por anchura de pulsos (PWM) alineados a flancos
- Modo PWM centralmente alineado

En el modo por anchura de pulsos alineado a flancos, para este modo, el bit 5, MSnB, del registro de estado y control del canal ***n*** del TPMx, TPMxCnSc, deberá estar a uno. Con la función de PWM alineado a flancos, el TPM puede generar señales digitales periódicas con una anchura de pulso en alto y bajo configurable. El periodo de la señal queda fijado por el dato escrito en el registro del módulo de 16 bits del TPMx, TPMxMOD, mientras que la anchura del pulso, y por lo tanto el ciclo de trabajo de la señal, queda fijado por el dato escrito en el registro de valor del canal ***n*** del TPMx, TPMxCnV.

Cálculos para generar una señal PWM para el control de velocidad de un motor de CD a una frecuencia de 60 Hz.

$$T = (Base\ de\ Tiempo)(Modulo) \qquad ecuación\ 1$$

Donde
T = [segundos]
Modulo = 0x0000….0xFFFF

$$Base\ de\ Tiempo = \frac{\text{Preescalador}}{\text{Frecuencia interna}} \quad ecuación\ 2$$

Donde
Preescalador = [1, 2, 4, 8, 16, 32, 64, 128]
Frecuencia interna = 8 MHz

Estableciedo los valores:
Preescalador a 128
Frecuencia interna de 8MHz
Señal con una frecuencia de 60 Hz
Periodo de 16.66 milisegundos

Y sustituyéndolos en la ecuación 2
Se tienen:

$$Base\ de\ Tiempo = \frac{128}{8\ MHz} = 16\ \mu s \qquad ecuación\ 3$$

Despejando la variable Módulo de la ecuación 1 y sustituyendo valores se tiene:

$$Módulo = \frac{16.66\ ms}{16\ \mu s} = 1041$$

El valor obtenido es el que se carga en el registro TPMxMOD, que es valor del periodo del PWM. Se debe de configurar si la señal del PWM va a estar alineada al centro, alineada al flanco y habilitar la interrupción del canal por medio del registro TPMxCnSC.

Para obtener el 50% del ciclo de trabajo de la señal se procede de la siguiente manera:

$$D.C = (1041 * .50) = 520.5$$

$$520_d = 0780_h$$

El resultado obtenido se cargará al registro TPMxMOD = 0x0208 en su forma hexadecimal. El resultado se aprecia en la figura 3.25 donde se observa la forma de la señal, su frecuencia, periodo y su ciclo de trabajo, por el pin 13 de la tarjeta DEMOJM, por los cálculos realizados anteriormente.

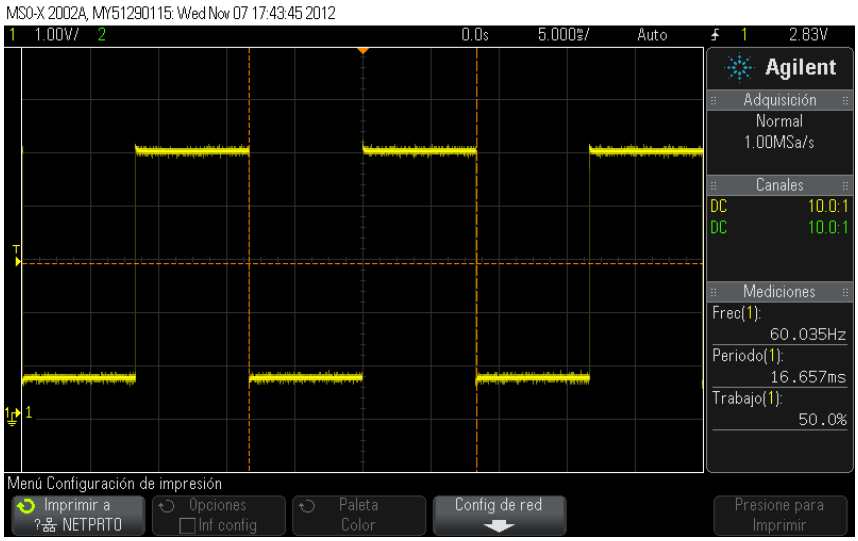


Figura 3.25 Muestra de la señal al 50% de trabajo.

Para realizar el cálculo del ciclo de trabajo del 75% se procede de la siguiente manera:

$$D.C = (1041 * .75) = 780.75$$

$$780_d = 030C_h$$

Se carga el valor nuevo al registro TPMxMOD = 0x030C en forma hexadecimal, pero se debe de considerar en la tarjeta DEMOJM, que en ella se obtiene a su salida del canal CH[0] un valor inverso a la señal, por lo tanto, se debe de plantear la siguiente ecuación para que dé el valor correspondiente a 75% del ciclo de trabajo de la señal.

$$D.C = (1041 * .25) = 260.25$$

$$260_d = 104_h$$

Cargado el valor correspondiente al 25% en el registro TPMxMOD, dará el resultado correcto como se muestra en la figura 3.26

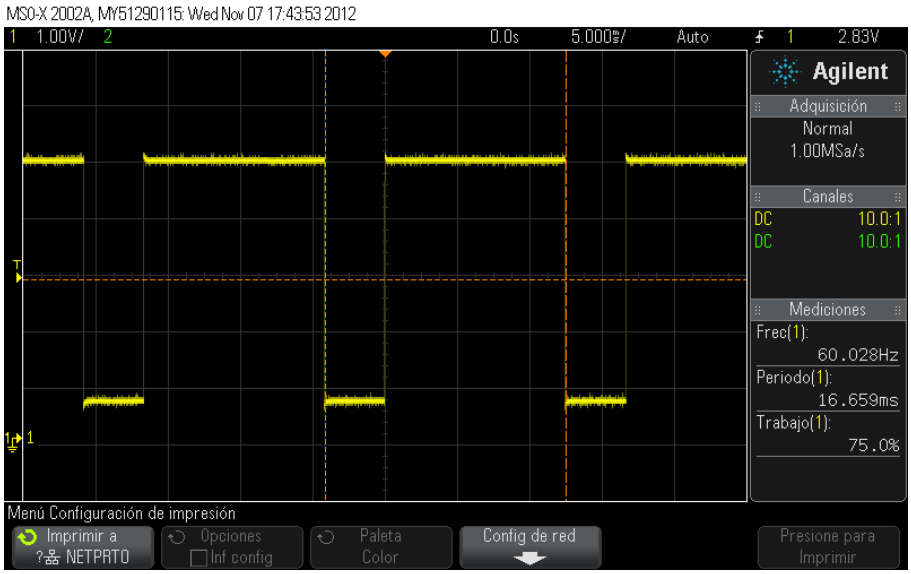


Figura 3.26 Señal al 75% de trabajo.

Desarrollo

El circuito de la figura 3.27 detalla la aplicación a implementar, para el controlar la velocidad de un motor de DC.

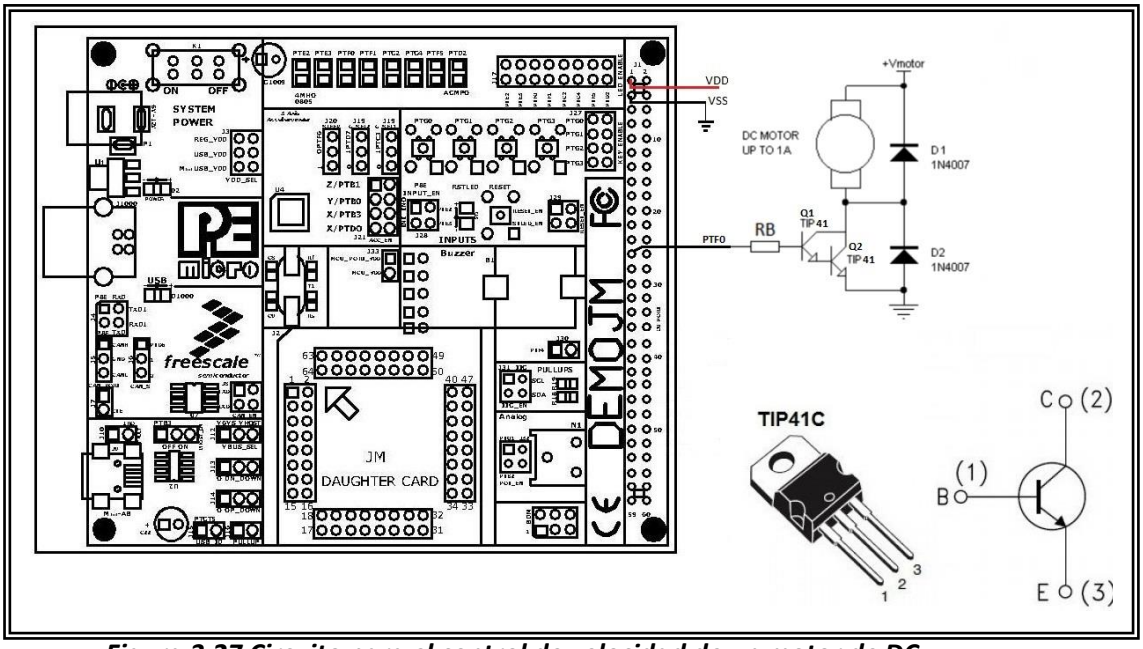
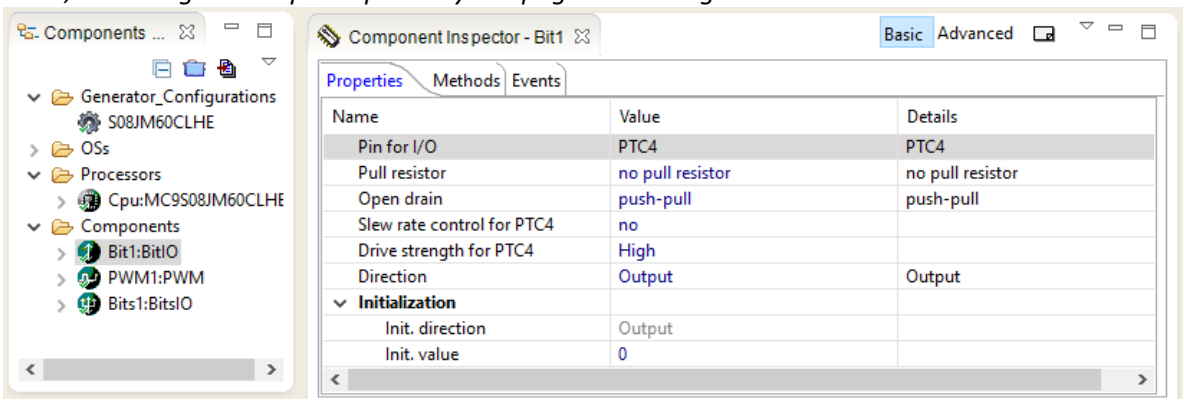


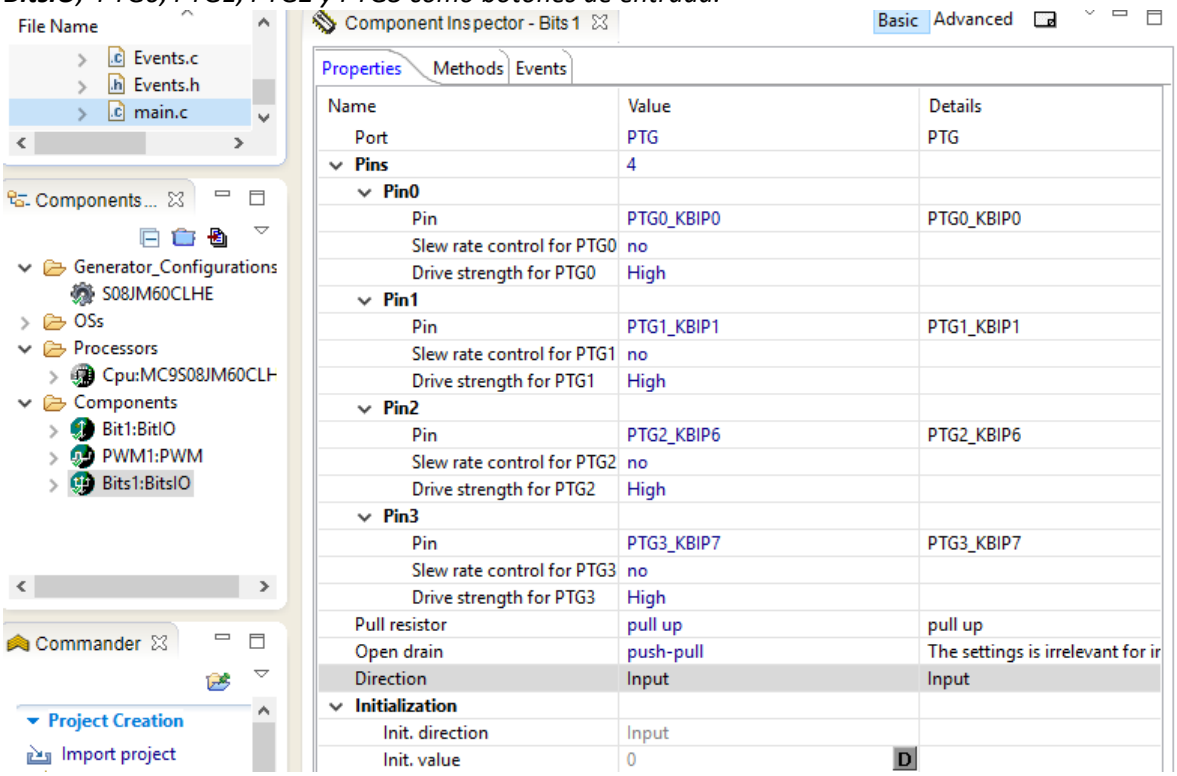
Figura 3.27 Circuito para el control de velocidad de un motor de DC.

1. Crear un nuevo proyecto en CodeWarrior (Eclipse) seleccionando el dispositivo MC9S08JM60 para ser conectado con P&E USB Multilink y seleccionar **Lenguaje C**, y seleccionar **Processor Expert**.

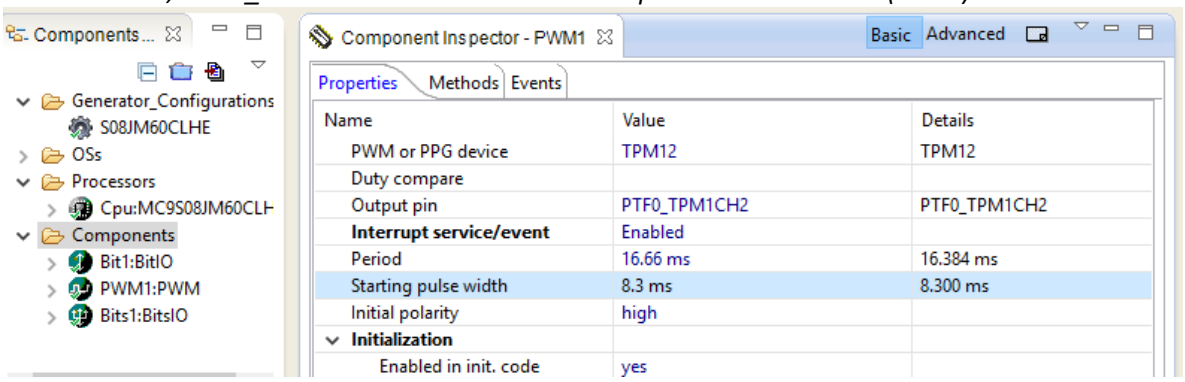
2. Agregar los siguientes componentes:
   ***BitIO**, Led testigo PTC4 que se prende y se apaga cada 2 segundos.*



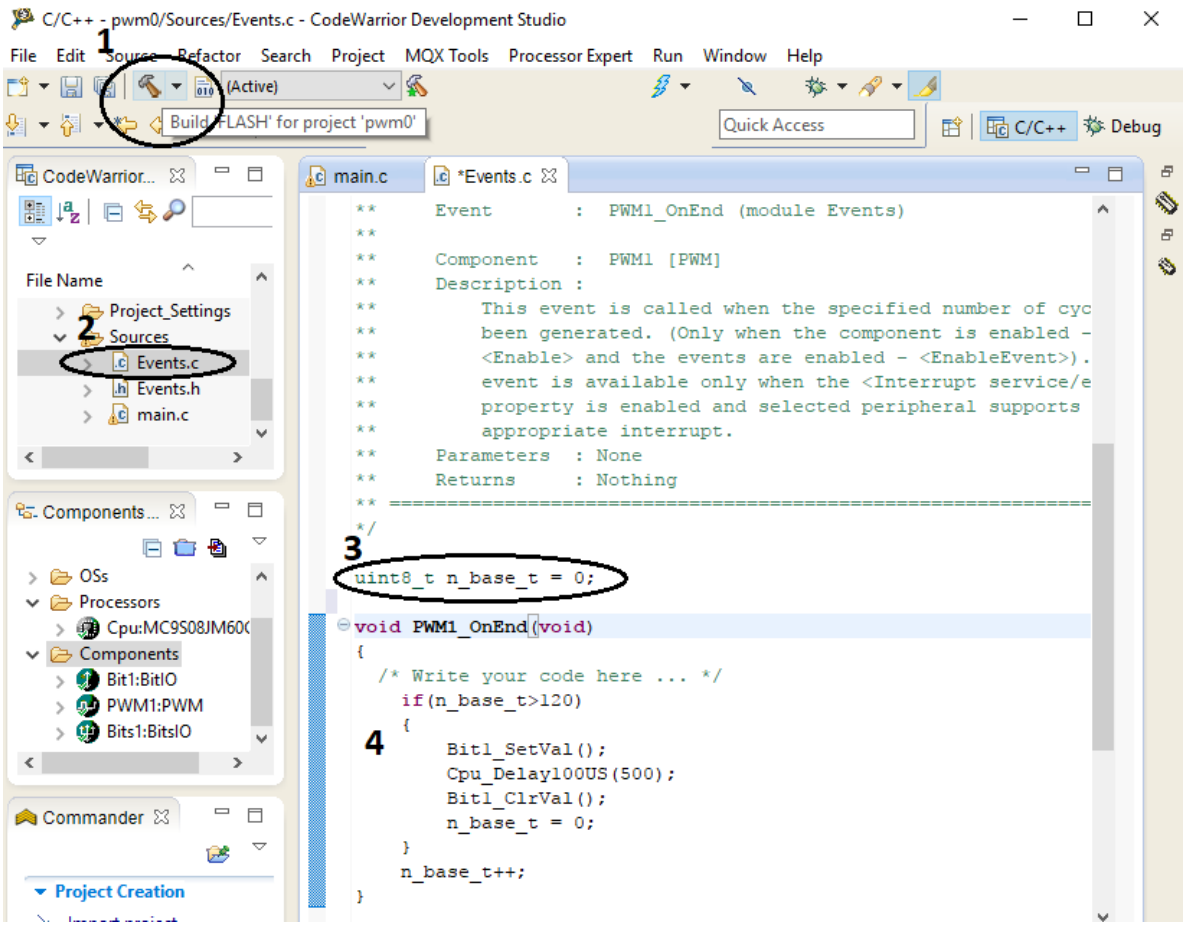
***BitsIO**, PTG0, PTG1, PTG2 y PTG3 como botones de entrada.*



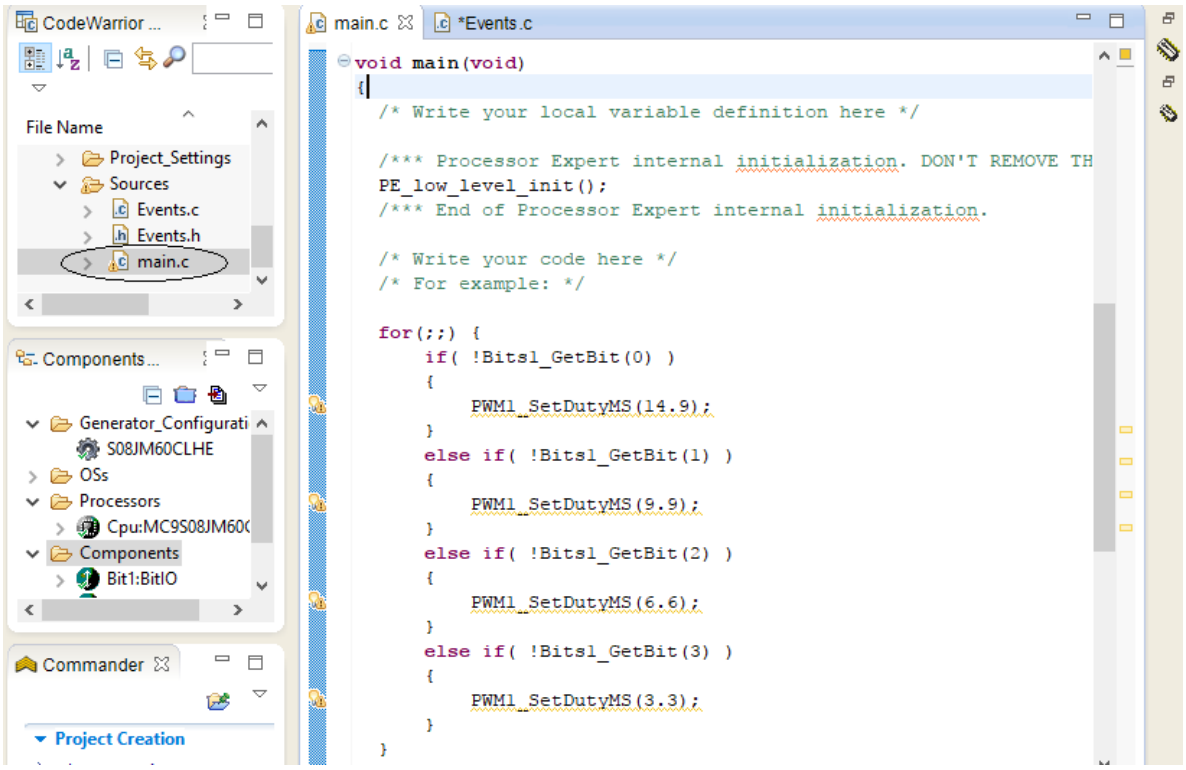
***PWM**, PTF0_TPM1CH2 como timer base con periodo de 16.66 ms (60 Hz).*



3. Construir la función **`PWM1_OnEnd`** para que cada 2 segundo (16.66 * 120 = 2000 ms) se encienda y apague el Led PTC4.
   1. Hacer clic en Build.
   2. Hacer doble clic en Events.c
   3. Declara la variable ***n_base_t*** de tipo *uint8_t*
   4. Escribir el código.



4. Hacer en ***main.c*** un ciclo infinito, que verifique el valor de los botones PTGx (0-3) y cambie el Duty de la señal PWM a 90%, 60%, 40% y 20%, respectivamente.



Se muestra el diagrama de flujo el programa principal para variar la velocidad de un motor de dc:

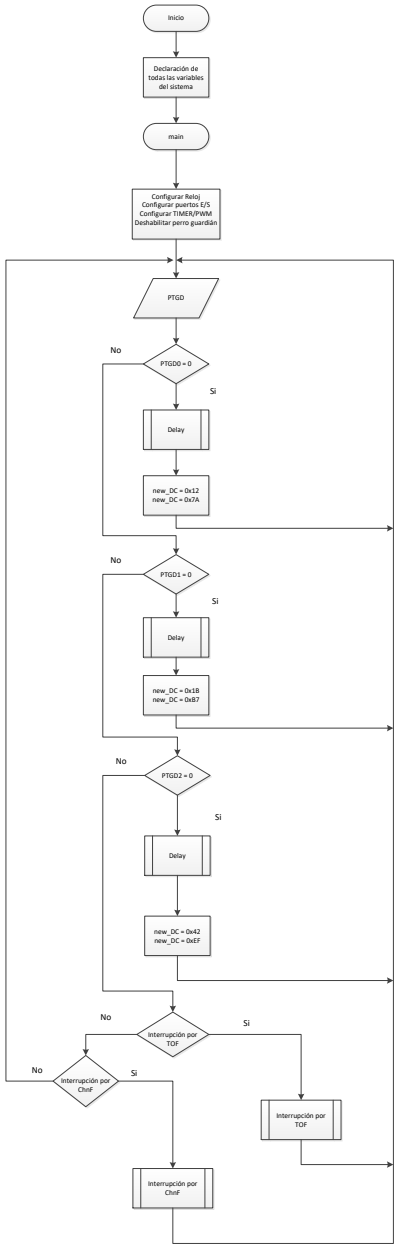


Figura 3.28 Configuración de módulos y periféricos para variar la velocidad del motor.

Se muestra el diagrama de flujo para cuando es invocada la interrupción por desbordamiento por TIMER:

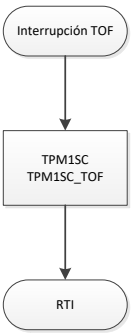


Figura 3.29 Atención a la interrupción por TIMER.

Se muestra el diagrama de flujo para cuando es invocada la interrupción por TIMER_PWM:

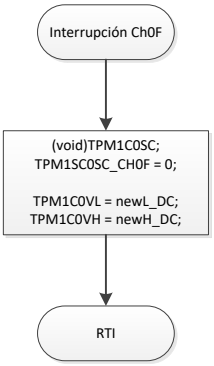


Figura 3.30 Atención a la interrupción por TIMER_PWM