

BITS DE CONFIGURACION(FUSES)

Los bits de configuración hacen parte de las características especiales del CPU, que establecen los parámetros de configuración del MCU, son una colección de bits especiales que solo pueden ser modificados durante la programación. Los bits de configuración son leídos durante el reset y habilitan o deshabitan característica del hardware en el microcontrolador (MCU). Algunas de las características controladas por los bits de configuración incluyen , fuente de reloj , temporizador [WatchDog \[1\]](#), detección [brownout](#) y protección de lectura. Los bits de configuración no son código ejecutable, esencialmente son fusibles localizados el espacio de la memoria del programa [2] . Los registros de configuración para el caso del [PIC18f8722](#), están en la sección **25.0 SPECIAL FEATURES OF THE CPU** del datasheet.



Registros de configuración especial PIC18F8722

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	00-- 0111
300002h CONFIG2L	—	—	—	BORV1	BORV0	BOREN1	BOREN0	PWRTEN	---1 1111
300003h CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	---1 1111
300004h CONFIG3L ⁽⁵⁾	WAIT	BW	ABW1	ABW0	—	—	PM1	PM0	1111 --11
300005h CONFIG3H	MCLRE	—	—	—	—	LPT1OSC	ECCPMX ⁽⁵⁾	CCP2MX	1--- -011
300006h CONFIG4L	DEBUG	XINST	BBSIZ1	BBSIZ0	—	LVP	—	STVREN	1000 -1-1
300008h CONFIG5L	CP7 ⁽¹⁾	CP6 ⁽¹⁾	CP5 ⁽²⁾	CP4 ⁽²⁾	CP3 ⁽³⁾	CP2	CP1	CP0	1111 1111
300009h CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah CONFIG6L	WRT7 ⁽¹⁾	WRT6 ⁽¹⁾	WRT5 ⁽²⁾	WRT4 ⁽²⁾	WRT3 ⁽³⁾	WRT2	WRT1	WRT0	1111 1111
30000Bh CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch CONFIG7L	EBRT7 ⁽¹⁾	EBRT6 ⁽¹⁾	EBTR5 ⁽²⁾	EBTR4 ⁽²⁾	EBTR3 ⁽³⁾	EBTR2	EBTR1	EBTR0	1111 1111
30000Dh CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh DEVID1 ⁽⁴⁾	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh DEVID2 ⁽⁴⁾	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	xxxx xxxx

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition.
Shaded cells are unimplemented, read as '0'.

Note 1: Unimplemented in PIC18F6527/6622/6627/8527/8622/8627 devices.
2: Unimplemented in PIC18F6527/6622/8527/8622 devices.
3: Unimplemented in PIC18F6527/8527 devices.
4: See Register 25-13 for DEVID1 values. DEVID registers are read-only and cannot be programmed by the user.
5: Unimplemented in PIC18F6527/6622/6627/6722 devices.

[1]. Section 9. Watchdog Timer (WDT), PIC24F <http://www1.microchip.com/downloads/en/DeviceDoc/39697b.pdf>

[2] View and Set Configuration Bits <http://microchip.wikidot.com/mplabx:view-and-set-configuration-bits>

La configuración de estos registros en Mplab se hace mediante la directiva `#Pragma config`, para agilizar el proceso de selección de estos bits, Mplab cuenta con la herramienta Configuration Bits, se encuentra en `RUN>Set Configuration Bits`, en ella se configuran los registros y se genera el código respectivo. Este debe incluirse en el proyecto, al ser un directiva pre-procesamiento puede ser ubicada en cualquier parte del código.

Registro **CONFIG1H** : Registro de configuración 1 Alto (byte de dirección 300001h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IESO	FCMEN	--	--	FOSC3	FOSC2	FOSC1	FOSC

IESO: Internal external switching over bit, Two-speed Start up Enabled; Permite al microcontrolador cambiar entre fuente interna y externa. Esta característica se usa para acelerar el inicio después de estar en modo SLEEP.

Para el proyecto *Ejemplo_000_BlinkingLed.X* estará deshabilitado

```
#pragma config IESO = OFF
```

FCMEN: Fail-Safe Clock Monitor Enable bit, Permite cambiar automáticamente a uno de los modos internos de reloj si una fuente externa de reloj falla.

Para el proyecto *Ejemplo_000_BlinkingLed.X* estará deshabilitado

```
#pragma config FCMEN = OFF
```

FOSC<3:0>: Oscillator Selection bits, Permite seleccionar la fuente de reloj para la CPU, se tienen 12 opciones, de ellas se selecciona la 8, oscilador interno, con funciones de puerto en los pines RA6 y RA7,

1000 = Internal oscillator block, port function on RA6 and RA7

Para el proyecto *Ejemplo_000_BlinkingLed.X* será,

```
#pragma config OSC = INTIO67
```

La configuración final del registro **CONFIG1H** es,

```
// CONFIG1H
```

```
#pragma config OSC = INTIO67
```

```
#pragma config FCMEN = OFF
```

```
#pragma config IESO = OFF
```

Registro **CONFIG2L** : Registro de configuración 1 bajo (byte de dirección 300002h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
--	--	--	BORV1	BORV0	BOREN1	BOREN0	~PWRTEN

D005	VBOR	Brown-out Reset Voltage					
		BORV<1:0> = 11	2.00	2.05	2.16	V	PIC18LF6627/6722/8627/8722
		BORV<1:0> = 11	2.00	2.11	2.22	V	PIC18LF6527/6622/8527/8622
		BORV<1:0> = 10	2.65	2.79	2.93	V	PIC18LF6X27/6X22/8X27/8X22
		BORV<1:0> = 01 ⁽²⁾	4.11	4.33	4.55	V	All devices
		BORV<1:0> = 00	4.36	4.59	4.82	V	All devices

BORV<1:0> : Brown-out Reset Voltage bits

Para el proyecto ***Ejemplo_000_BlinkingLed.X*** será 3, para un reset automático solo cuando el voltaje sea menor a ~2V

`#pragma config BORV = 3`

BOREN<1:0>: Brown-out Reset Enable bits, Habilita el reset por Brown-out, se deshabilita por ello la los bits BORV, no tendrán efecto.

00 = Brown-out Reset disabled in hardware and software

Para el proyecto ***Ejemplo_000_BlinkingLed.X*** será

`#pragma config BOREN = OFF`

~PWRTEN: Power-up Timer Enable bit, mantiene al microcontrolador en reset por X ms cuando el voltaje de alimentación es aplicado, esto le da tiempo a la fuente de alimentación para estabilizarse.

Para el proyecto ***Ejemplo_000_BlinkingLed.X*** será

`#pragma config PWRT = OFF`

La configuración final del registro **CONFIG2L** es,

`// CONFIG2L`

`#pragma config PWRT = OFF`

`#pragma config BOREN = OFF`

`#pragma config BORV = 3`

Registro **CONFIG2H** : Registro de configuración 2 alto (byte de dirección 300003h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
--	--	--	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN

WDTPS<3:0>: Watchdog Timer Postscale Select bits , selección del postcaler para el Watch Dog timer.

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

```
#pragma config WDTPS = 32768
```

Relación 1:32768

WDTEN: Watchdog Timer Enable bit, Bit de habilitación del WDT

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

```
#pragma config WDT = OFF
```

WDT deshabilitado

La configuración final del registro **CONFIG2H** es,

```
// CONFIG2H
```

```
#pragma config WDT = OFF
```

```
#pragma config WDTPS = 32768
```

Registro **CONFIG3L** : Registro de configuración 2 alto (byte de dirección 300004h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
<u>WAIT</u>	BW	ABW1	ABW0	---	---	PM1	PM0

WAIT:External bus data wait enable:

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config WAIT = OFF`

BW: Data Bus width select:

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config DATABW = DATA16BIT`

ABW<1:0>:Address Bus Width Select bits,

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config ADDR20BIT`

PM<1:0>:Processor Data Memory Mode Select bits

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config MODE = MC`

La configuración final del registro **CONFIG3L** es,

```
// CONFIG3L
```

```
#pragma config MODE = MC
```

```
#pragma config ADDR20BIT
```

```
#pragma config DATABW = DATA16BIT
```

```
#pragma config WAIT = OFF
```

Registro **CONFIG3H** : Registro de configuración 3 alto (byte de dirección 300005h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
~MCLRE	---	---	---	---	LPT1OSC	ECCPMX	CCP2MX

~MCLRE: Master Clear Enable Bit, Habilita o deshabilita el pin de reset, si se deshabilita, el pin que corresponde a MCLRE en este caso RG5 queda como entrada digital, solo entrada.

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config MCLRE = ON`

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config LPT1OSC = OFF`

ECCPMX : ECCP MUX bit , multiplexa las conexión del módulo ECCP en RE<6:3> o RH<7:4>

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config ECCPMX = PORTE`

CCP2MX: ECCP MUX bit, multiplexa las conexión del módulo CCP2 con RC1 o RB3

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

`#pragma config CCP2MX = PORTC`

La configuración final del registro **CONFIG3H** es,

`// CONFIG3H`

`#pragma config CCP2MX = PORTC`

`#pragma config ECCPMX = PORTE`

`#pragma config MCLRE = ON`

`#pragma config LPT1OSC = OFF`

Registro **CONFIG4L** : Registro de configuración 4 bajo (byte de dirección 300006h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
~DEBUG	XINST	BBSIZ1	BBSIZ0	---	LVP	---	STVREN

~DEBUG: Background debugger Enable Bit, Habilitación para background DEBUG , los pines RB6 y RB7 son dedicados al DEBUG en el circuito.

Este bit no es recomendable manipularlo, el IDE lo hace automáticamente.

XINST: Extended Instruction Set Enable bit, habilita la extensión de instrucciones, [XC8](#) no soporta esta característica, la instrucciones adicionales están en el datashet en la sección [26.2 “Extended Instruction Set”](#)

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

```
#pragma config XINST = OFF
```

BBSIZ1: Boot Block Size Select bits, esta opción determina el tamaño de los bloques de memoria en modo microprocesador.

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

```
#pragma config BBSIZ = BB2K
```

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

```
#pragma config LVP = OFF
```

STVREN: StakFull /Underflow reset enable bit, Permite el reset debido a casos como el desbordamiento en el stack o perdida de la dirección de retorno.

Para el proyecto *Ejemplo_000_BlinkingLed.X* será

```
#pragma config STVREN = ON
```

La configuración final del registro **CONFIG4L** es,

```
// CONFIG4L
```

```
#pragma config STVREN = ON
```

```
#pragma config LVP = OFF
```

```
#pragma config BBSIZ = BB2K
```

```
#pragma config XINST = OFF
```

Registro **CONFIG5L** : Registro de configuración 5 bajo (byte de dirección 300008h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CP7	CP6	CP5	CP4	CP3	CP2	CP1	CP0

Bits de protección de código, determinado por bloques de memoria, para el proyecto *Ejemplo_000_BlinkingLed.X* no se protegerá.

La configuración final del registro **CONFIG5L** es,

```
// CONFIG5L
```

```
#pragma config CP0 = OFF
```

```
#pragma config CP1 = OFF
```

```
#pragma config CP2 = OFF
```

```
#pragma config CP3 = OFF
```

```
#pragma config CP4 = OFF
```

```
#pragma config CP5 = OFF
```

```
#pragma config CP6 = OFF
```

```
#pragma config CP7 = OFF
```

Registro **CONFIG5H** : Registro de configuración 5 alto (byte de dirección 300009h)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CPD	CPB	---	---	---	---	---	---

CPD: Protección de la EEPROM

CPB: Protección de los bloques de boot

La configuración final del registro **CONFIG5H** es,

```
// CONFIG5H
```

```
#pragma config CPB = OFF
```

```
#pragma config CPD = OFF
```


Registro **CONFIG6L** : Registro de configuración 6 bajo (byte de dirección 30000Ah)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
WRT7	WRT6	WRT5	WRT4	WRT3	WRT2	WRT1	WRT0

WRT: Registro de protección contra escritura, la protección está determinado por bloques de memoria, del bloque 0 al 7.

La configuración final del registro CONFIG6L es,

```
// CONFIG6L
```

```
#pragma config WRT0 = OFF
```

```
#pragma config WRT1 = OFF
```

```
#pragma config WRT2 = OFF
```

```
#pragma config WRT3 = OFF
```

```
#pragma config WRT4 = OFF
```

```
#pragma config WRT5 = OFF
```

```
#pragma config WRT6 = OFF
```

```
#pragma config WRT7 = OFF
```

Registro **CONFIG6H** : Registro de configuración 6 alto (byte de dirección 30000Bh)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
WRTD	WRTB	WRTC	---	---	---	---	---

WRTD: Data EEPROM Write Protection bit, para el proyecto *Ejemplo_000_BlinkingLed.X* no se protegerá.

WRTB : Boot Block Write Protection bit, para el proyecto *Ejemplo_000_BlinkingLed.X* no se protegerá.

WRTC: Configuration Register Write Protection bit, para el proyecto *Ejemplo_000_BlinkingLed.X* no se protegerá.

La configuración final del registro CONFIG6H es,

```
// CONFIG6H
```

```
#pragma config WRTC = OFF
```

```
#pragma config WRTB = OFF
```

```
#pragma config WRTD = OFF
```

Registro **CONFIG7L** : Registro de configuración 7 bajo (byte de dirección 30000Ch)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
EBTR7	EBTR6	EBTR5	EBTR4	EBTR3	EBTR2	EBTR1	EBTR0

EBTR7: Table Read Protection bit , protección contra lectura de bloques, para el proyecto *Ejemplo_000_BlinkingLed.X* no se protegerá.

La configuración final del registro CONFIG7L es,

```
// CONFIG7L
```

```
#pragma config EBTR0 = OFF
```

```
#pragma config EBTR1 = OFF
```

```
#pragma config EBTR2 = OFF
```

```
#pragma config EBTR3 = OFF
```

```
#pragma config EBTR4 = OFF
```

```
#pragma config EBTR5 = OFF
```

```
#pragma config EBTR6 = OFF
```

```
#pragma config EBTR7 = OFF
```

Registro **CONFIG7H** : Registro de configuración 7 alto (byte de dirección 30000Dh)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
---	EBTRB	---	---	---	---	---	---

EBTRB: Boot Block Table Read Protection bit, para el proyecto *Ejemplo_000_BlinkingLed.X* no se protegerá.

La configuración final del registro CONFIG7H es,

```
// CONFIG7H
```

```
#pragma config EBTRB = OFF
```