TECNOLÓGICO DE MONTERREY

Tema 13
CONFIGURATION REGISTERS

Special features of the PIC18

- Oscillator selector
- Resets
  - Power on
  - Power up timer
  - Oscillator startup
  - Brown ont reset
- Intrrupts
- Watchdog Timer
- Code Protection
- ID Locations
- In circuit programming

Configuration Registers

- The special features are configured on bits that are part or the configuration Registers.
- The registers are stored in the flash memory in the program space but on addresses that that our of reach of the user (300000h-3FFFFFh)
- The common practice is that the setting of these bits are in the same application code

Configuration Registers

- If included in the code, they are part of the HEX code but they can be modified on run time although the functionality will become active after reset.
- To modify on run time, you use the same SFR registers used to write to the program memory (flash), this is explained in section 7.6 of the data sheet

Configuration Registers

TABLE 26-1: CONFIGURATION BITS AND DEVICE IDs

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300000h	CONFIG1L	—	—	LS48MHZ	CPUDIV<1:0>		—	CFGPLLEN	PLLSEL	0000 0000
300001h	CONFIG1H	IESO	FCMEN	PCLKEN	—	FOSC<3:0>				0010 0101
300002h	CONFIG2L	—	$\overline{\text{LPBOR}}$	—	BORV<1:0>		BOREN<1:0>		$\overline{\text{PWRTEN}}$	0101 1111
300003h	CONFIG2H	—	—	WDTPS<3:0>				WDTEN<1:0>		0011 1111
300004h	CONFIG3L	—	—	—	—	—	—	—	—	0000 0000
300005h	CONFIG3H	MCLRE	SDOMX	—	T3CMX	—	—	PBADEN	CCP2MX	1101 0011
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	XINST	ICPRT(5)	—	—	LVP(1)	—	STRVEN	1010 0101
300007h	CONFIG4H	—	—	—	—	—	—	—	—	1111 1111
300008h	CONFIG5L	—	—	—	—	CP3(2)	CP2(2)	CP1	CP0	0000 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	1100 0000
30000Ah	CONFIG6L	—	—	—	—	WRT3(2)	WRT2(2)	WRT1	WRT0	0000 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC(3)	—	—	—	—	—	1110 0000
30000Ch	CONFIG7L	—	—	—	—	EBTR3(2)	EBTR2(2)	EBTR1	EBTR0	0000 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	0100 0000
3FFFFEh	DEVID1(4)	DEV<2:0>			REV<4:0>					qqqq qqqq
3FFFFFh	DEVID2(4)	DEV<10:3>								0101 1100

REGISTER 26-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW

U-0	U-0	R/P-0	R/P-0	R/P-0	U-0	R/P-0	R/P-0
—	—	LS48MHZ	CPUDIV<1:0>		—	CFGPLLEN	PLLSEL
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **LS48MHZ:** USB Low-Speed Clock Selection bit
Selects the clock source for Low-speed USB operation
1 = System clock is expected at 48 MHz, FS/LS USB clock divide-by is set to 8
0 = System clock is expected at 24 MHz, FS/LS USB clock divide-by is set to 4

bit 4-3 **CPUDIV<1:0>:** CPU System Clock Selection bits
11 = CPU system clock divided by 6
10 = CPU system clock divided by 3
01 = CPU system clock divided by 2
00 = No CPU system clock divide

bit 2 **Unimplemented:** Read as '0'

bit 1 **CFGPLLEN:** PLL Enable bit(1)
1 = Oscillator multiplied by 3 or 4, depending on the PLLSEL bit
0 = Oscillator used directly

bit 0 **PLLSEL:** PLL Multiplier Selection bit
1 = Output frequency is 3x the input frequency
0 = Output frequency is 4x the input frequency

REGISTER 26-2: CONFIG1H: CONFIGURATION REGISTER 1 HIGH

R/P-0	R/P-0	R/P-1	U-0	R/P-0	R/P-1	R/P-0	R/P-1
IESO	FCMEN	PCLKEN	—	FOSC<3:0>			
bit 7							bit 0

bit 7 **IESO(1):** Internal/External Oscillator Switchove
1 = Oscillator Switchover mode enabled
0 = Oscillator Switchover mode disabled

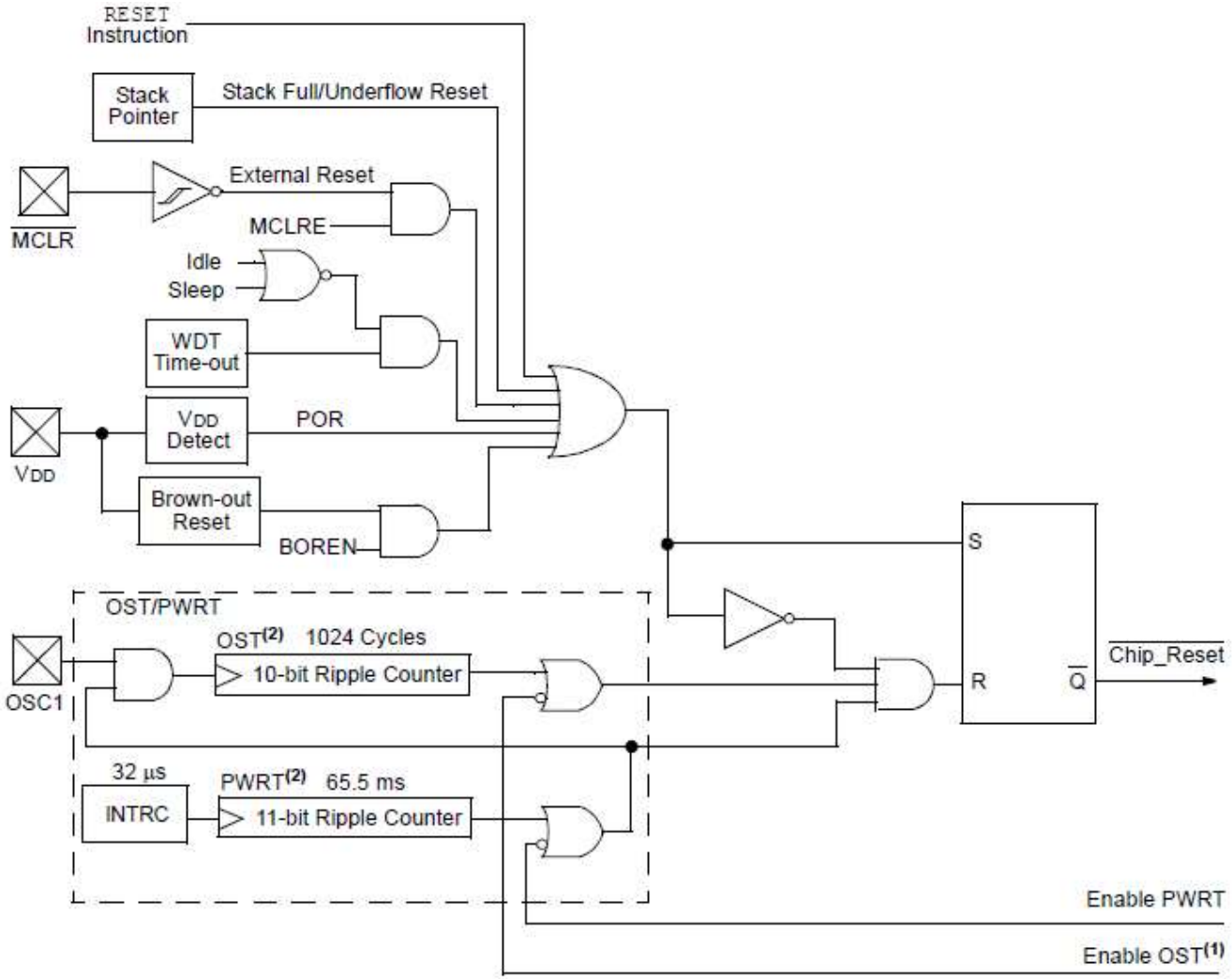
bit 6 **FCMEN(1):** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 5 **PCLKEN:** Primary Clock Enable bit
1 = Primary Clock is always enabled
0 = Primary Clock can be disabled by software

bit 4 **Unimplemented:** Read as '0'

bit 3-0 **FOSC<3:0>:** Oscillator Selection bits
1111 = Reserved
1110 = Reserved
1101 = EC oscillator **(low power, <4 MHz)**
1100 = EC oscillator, CLKO function on OSC2 **(low power, <4 MHz)**
1011 = EC oscillator **(medium power, 4 MHz - 16 MHz)**
1010 = EC oscillator, CLKO function on OSC2 **(medium power, 4 MHz - 16 MHz)**
1001 = Internal oscillator block, CLKO function on OSC2
1000 = Internal oscillator block
0111 = External RC oscillator
0110 = External RC oscillator, CLKO function on OSC2
0101 = EC oscillator **(high power, 16 MHz - 48 MHz)**
0100 = EC oscillator, CLKO function on OSC2 **(high power, 16 MHz - 48 MHz)**
0011= HS oscillator **(medium power, 4 MHz - 16 MHz)**
0010= HS oscillator **(high power, 16 MHz - 25 MHz)**
0001= XT oscillator
0000= LP oscillator

Reset


RESET
Instruction
Stack Pointer
Stack Full/Underflow Reset
External Reset
MCLR
MCLRE
Idle
Sleep
WDT
Time-out
VDD
VDD Detect
POR
Brown-out Reset
BOREN
S
R
Q
Chip_Reset
OST/PWRT
OST(2) 1024 Cycles
OSC1
10-bit Ripple Counter
32 µs
PWRT(2) 65.5 ms
INTRC
11-bit Ripple Counter
Enable PWRT
Enable OST(1)

REGISTER 26-3: CONFIG2L: CONFIGURATION REGISTER 2 LOW

U-0	R/P-1	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	$\overline{\text{LPBOR}}$	—	BORV<1:0>(1)		BOREN<1:0>(2)		$\overline{\text{PWRTEN}}$(2)
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **$\overline{\text{LPBOR}}$:** Low-Power Brown-out Reset Enable bits
1 = Low-Power Brown-out Reset disabled
0 = Low-Power Brown-out Reset enabled

bit 5 **Unimplemented:** Read as '0'

bit 4- **BORV<1:0>:** Brown-out Reset Voltage bits(1)
11 = V_{BOR} set to 1.9V nominal
10 = V_{BOR} set to 2.2V nominal
01 = V_{BOR} set to 2.5V nominal
00 = V_{BOR} set to 2.85V nominal

bit 2- **BOREN<1:0>:** Brown-out Reset Enable bits(2)
11 = Brown-out Reset enabled in hardware only (SBOREN is disabled)
10 = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled)
01 = Brown-out Reset enabled and controlled by software (SBOREN is enabled)
00 = Brown-out Reset disabled in hardware and software

bit 0 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit(2)
1 = PWRT disabled
0 = PWRT enabled

REGISTER 24-3: CONFIG2H: CONFIGURATION REGISTER 2 HIGH

U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	WDTPS<3:0>				WDTEN<1:0>	
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **WDTPS<3:0>:** Watchdog Timer Postscale Select bits

1111 = 1:32,768	0111 = 1:128
1110 = 1:16,384	0110 = 1:64
1101 = 1:8,192	0101 = 1:32
1100 = 1:4,096	0100 = 1:16
1011 = 1:2,048	0011 = 1:8
1010 = 1:1,024	0010 = 1:4
1001 = 1:512	0001 = 1:2
1000 = 1:256	0000 = 1:1

bit 1-0 **WDTEN<1:0>:** Watchdog Timer Enable bits

11 = WDT enabled in hardware; SWDTEN bit disabled

10 = WDT controlled by the SWDTEN bit

01 = WDT enabled when device is active, disabled when device is in Sleep; SWDTEN bit disabled

00 = WDT disabled in hardware; SWDTEN bit disabled

REGISTER 26-5: CONFIG3H: CONFIGURATION REGISTER 3 HIGH

R/P-1	R/P-1	U-0	R/P-1	U-0	U-0	R/P-1	R/P-1
MCLRE	SDOMX	—	T3CMX	—	—	PBADEN	CCP2MX
bit 7							bit 0

bit 7 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit
1 = $\overline{\text{MCLR}}$ pin enabled; RE3 input pin disabled
0 = RE3 input pin enabled; $\overline{\text{MCLR}}$ disabled

bit 6 **SDOMX:** SDO Output MUX bit
1 = SDO is on RB3
0 = SDO is on RC7

bit 5 **Unimplemented:** Read as '0'

bit 4 **T3CMX:** Timer3 Clock Input MUX bit
1 = T3CKI is on RC0
0 = T3CKI is on RB5

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **PBADEN:** PORTB A/D Enable bit
1 = ANSELB<5:0> resets to 1, PORTB<5:0> pins are configured as analog inputs on Reset
0 = ANSELB<5:0> resets to 0, PORTB<4:0> pins are configured as digital I/O on Reset

bit 0 **CCP2MX:** CCP2 MUX bit
1 = CCP2 input/output is multiplexed with RC1
0 = CCP2 input/output is multiplexed with RB3

REGISTER 26-6: CONFIG4L: CONFIGURATION REGISTER 4 LOW

R/P-1	R/P-0	R/P-1	U-0	U-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$(2)	XINST	ICPRT(3)	—	—	LVP(1)	—	STVREN
bit 7							bit 0

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit(2)
1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins
0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug

bit 6 **XINST:** Extended Instruction Set Enable bit
1 = Instruction set extension and Indexed Addressing mode enabled
0 = Instruction set extension and Indexed Addressing mode disabled (Legacy mode)

bit 5 **ICPRT:** Dedicated In-Circuit (ICD) Port Enable bit(3)
1 = ICPORT enabled (ICD function on dedicated ICD pins)
0 = ICPORT disabled (ICD function on default ICD pins, RB6/7)

bit 4- **Unimplemented:** Read as '0'

bit 2 **LVP:** Single-Supply ICSP Enable bit
1 = Single-Supply ICSP enabled
0 = Single-Supply ICSP disabled

bit 1 **Unimplemented:** Read as '0'

bit 0 **STVREN:** Stack Full/Underflow Reset Enable bit
1 = Stack full/underflow will cause Reset
0 = Stack full/underflow will not cause Reset

27.2 Extended Instruction Set

Note 1: Can only be changed by a programmer in high-voltage programming mode.
2: The $\overline{\text{DEBUG}}$ bit is managed automatically by device development tools including debuggers and programmers. For normal device operations, this bit should be maintained as a '1'.
3: Available only on 44-pin TQFP package devices. Program this bit clear on all other devices.

REGISTER 26-6: CONFIG4L: CONFIGURATION REGISTER 4 LOW

R/P-1	R/P-0	R/P-1	U-0	U-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$(2)	XINST	ICPRT(3)	—	—	LVP(1)	—	STVREN
bit 7							bit 0

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit(2)
1 = Background debugger disabled, RB6 and RB7 configured as general purpose I/O pins
0 = Background debugger enabled, RB6 and RB7 are dedicated to In-Circuit Debug

bit 6 **XINST:** Extended Instruction Set Enable bit
1 = Instruction set extension and Indexed Addressing mode enabled
0 = Instruction set extension and Indexed Addressing mode disabled (Legacy mode)

bit 5-3 **Unimplemented:** Read as '0'

bit 2 **LVP:** Single-Supply ICSP Enable bit
1 = Single-Supply ICSP enabled
0 = Single-Supply ICSP disabled

bit 1 **Unimplemented:** Read as '0'

bit 0 **STVREN:** Stack Full/Underflow Reset Enable bit
1 = Stack full/underflow will cause Reset
0 = Stack full/underflow will not cause Reset

REGISTER 26-7: CONFIG5L: CONFIGURATION REGISTER 5 LOW

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	CP3(1)	CP2(1)	CP1	CP0
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CP3:** Code Protection bit(1)
1 = Block 3 not code-protected
0 = Block 3 code-protected

bit 2 **CP2:** Code Protection bit(1)
1 = Block 2 not code-protected
0 = Block 2 code-protected

bit 1 **CP1:** Code Protection bit
1 = Block 1 not code-protected
0 = Block 1 code-protected

bit 0 **CP0:** Code Protection bit
1 = Block 0 not code-protected
0 = Block 0 code-protected

REGISTER 26-8: CONFIG5H: CONFIGURATION REGISTER 5 HIGH

R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
CPD	CPB	—	—	—	—	—	—
bit 7							bit 0

bit 7 **CPD:** Data EEPROM Code Protection bit
1 = Data EEPROM not code-protected
0 = Data EEPROM code-protected

bit 6 **CPB:** Boot Block Code Protection bit
1 = Boot Block not code-protected
0 = Boot Block code-protected

bit 5-0 **Unimplemented:** Read as '0'

REGISTER 26-9: CONFIG6L: CONFIGURATION REGISTER 6 LOW

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	WRT3(1)	WRT2(1)	WRT1	WRT0
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **WRT3:** Write Protection bit(1)
1 = Block 3 not write-protected
0 = Block 3 write-protected

bit 2 **WRT2:** Write Protection bit(1)
1 = Block 2 not write-protected
0 = Block 2 write-protected

bit 1 **WRT1:** Write Protection bit
1 = Block 1 not write-protected
0 = Block 1 write-protected

bit 0 **WRT0:** Write Protection bit
1 = Block 0 not write-protected
0 = Block 0 write-protected

REGISTER 26-10: CONFIG6H: CONFIGURATION REGISTER 6 HIGH

R/C-1	R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0
WRTD	WRTB	WRTC(1)	—	—	—	—	—
bit 7							bit 0

bit 7 **WRTD:** Data EEPROM Write Protection bit
1 = Data EEPROM not write-protected
0 = Data EEPROM write-protected

bit 6 **WRTB:** Boot Block Write Protection bit
1 = Boot Block not write-protected
0 = Boot Block write-protected

bit 5 **WRTC:** Configuration Register Write Protection bit(1)
1 = Configuration registers not write-protected
0 = Configuration registers write-protected

bit 4-0 **Unimplemented:** Read as '0'

REGISTER 26-11: CONFIG7L: CONFIGURATION REGISTER 7 LOW

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	EBTR3(1)	EBTR2(1)	EBTR1	EBTR0
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **EBTR3:** Table Read Protection bit(1)
1 = Block 3 not protected from table reads executed in other blocks
0 = Block 3 protected from table reads executed in other blocks

bit 2 **EBTR2**: Table Read Protection bit(1)
1 = Block 2 not protected from table reads executed in other blocks
0 = Block 2 protected from table reads executed in other blocks

bit 1 **EBTR1:** Table Read Protection bit
1 = Block 1 not protected from table reads executed in other blocks
0 = Block 1 protected from table reads executed in other blocks

bit 0 **EBTR0:** Table Read Protection bit
1 = Block 0 not protected from table reads executed in other blocks
0 = Block 0 protected from table reads executed in other blocks

REGISTER 26-12: CONFIG7H: CONFIGURATION REGISTER 7 HIGH

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **EBTRB:** Boot Block Table Read Protection bit
1 = Boot Block not protected from table reads executed in other blocks
0 = Boot Block protected from table reads executed in other blocks

bit 5-0 **Unimplemented:** Read as '0'

REGISTER 26-13: DEVID1: DEVICE ID REGISTER 1

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

bit 7-5 **DEV<2:0>:** Device ID bits
These bits, together with DEV<10:3> in DEVID2, determine the device ID.
See Table 26-2 for complete Device ID list.

bit 4-0 **REV<4:0>:** Revision ID bits
These bits indicate the device revision.

REGISTER 26-14: DEVID2: DEVICE ID REGISTER 2

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

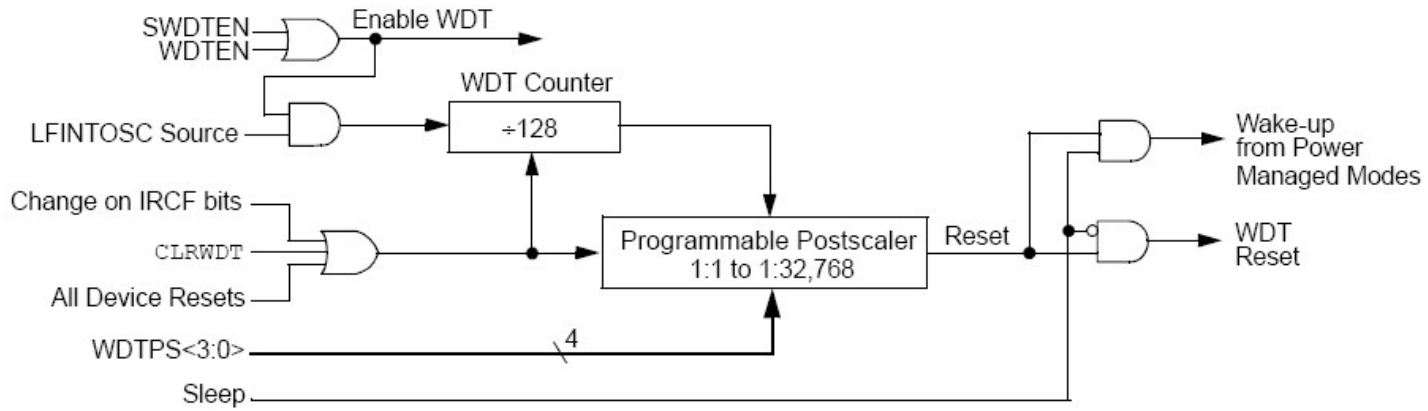
bit 7-0 **DEV<10:3>:** Device ID bits
These bits, together with DEV<2:0> in DEVID1, determine the device ID.
See Table 26-2 for complete Device ID list.

TABLE 26-2: DEVICE ID TABLE FOR THE PIC18(L)F2X/45K50 FAMILY

DEV<10:3>	DEV<2:0>	Part Number
0101 1100	000	PIC18F45K50
	001	PIC18F25K50
	011	PIC18F24K50
	100	PIC18LF45K50
	101	PIC18LF25K50
	111	PIC18LF24K50

Watchdog timer

- Its base time is given by LFINTOSC
- The timeout goea from 4ms to 131 seconds
- The WDT counter is cleared (kicked) when you execute the CLRWDT and SLEEP instructions



Watchdog timer

REGISTER 26-15: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN(1)
bit 7							bit 0

bit 7-1 **Unimplemented:** Read as '0'

bit 0 **SWDTEN:** Software Enable or Disable the Watchdog Timer bit(1)

1 = WDT is turned on

0 = WDT is turned off (Reset value)

Note 1: This bit has no effect if the Configuration bit, WDTEN, is enabled.

Code Protection

- The program memory is divided from 3 to 5 blocks depending on the device.
- The memory size per block can be from 0.5K to 2K bytes
- Each block has 3 protection bits associated with them :
  - Code-Protect bit (Cpn)
  - Write-Protect bit (WTRn)
  - External Block Table Read bit (EBTRn)

MEMORY SIZE/DEVICE				Block Code Protection Controlled By:
8 Kbytes (PIC18(L)FX3K22)	16 Kbytes (PIC18(L)FX4K22)	32 Kbytes (PIC18(L)FX5K22)	64 Kbytes (PIC18(L)FX6K22)	
Boot Block (000h-1FFh)	Boot Block (000h-7FFh)	Boot Block (000h-7FFh)	Boot Block (000h-7FFh)	CPB, WRTB, EBTRB
Block 0 (200h-FFFh)	Block 0 (800h-1FFFh)	Block 0 (800h-1FFFh)	Block 0 (800h-3FFFh)	CP0, WRT0, EBTR0
Block 1 (1000h-1FFFh)	Block 1 (2000h-3FFFh)	Block 1 (2000h-3FFFh)	Block 1 (4000h-7FFFh)	CP1, WRT1, EBTR1
		Block 2 (4000h-5FFFh)	Block 2 (8000h-BFFFh)	CP2, WRT2, EBTR2
		Block 3 (6000h-7FFFh)	Block 3 (C000h-FFFFh)	CP3, WRT3, EBTR3
Unimplemented Read '0's (2000h-1FFFFFh)	Unimplemented Read '0's (4000h-1FFFFFh)	Unimplemented Read '0's (8000h-1FFFFFh)	Unimplemented Read '0's (10000h-1FFFFFh)	(Unimplemented Memory Space)

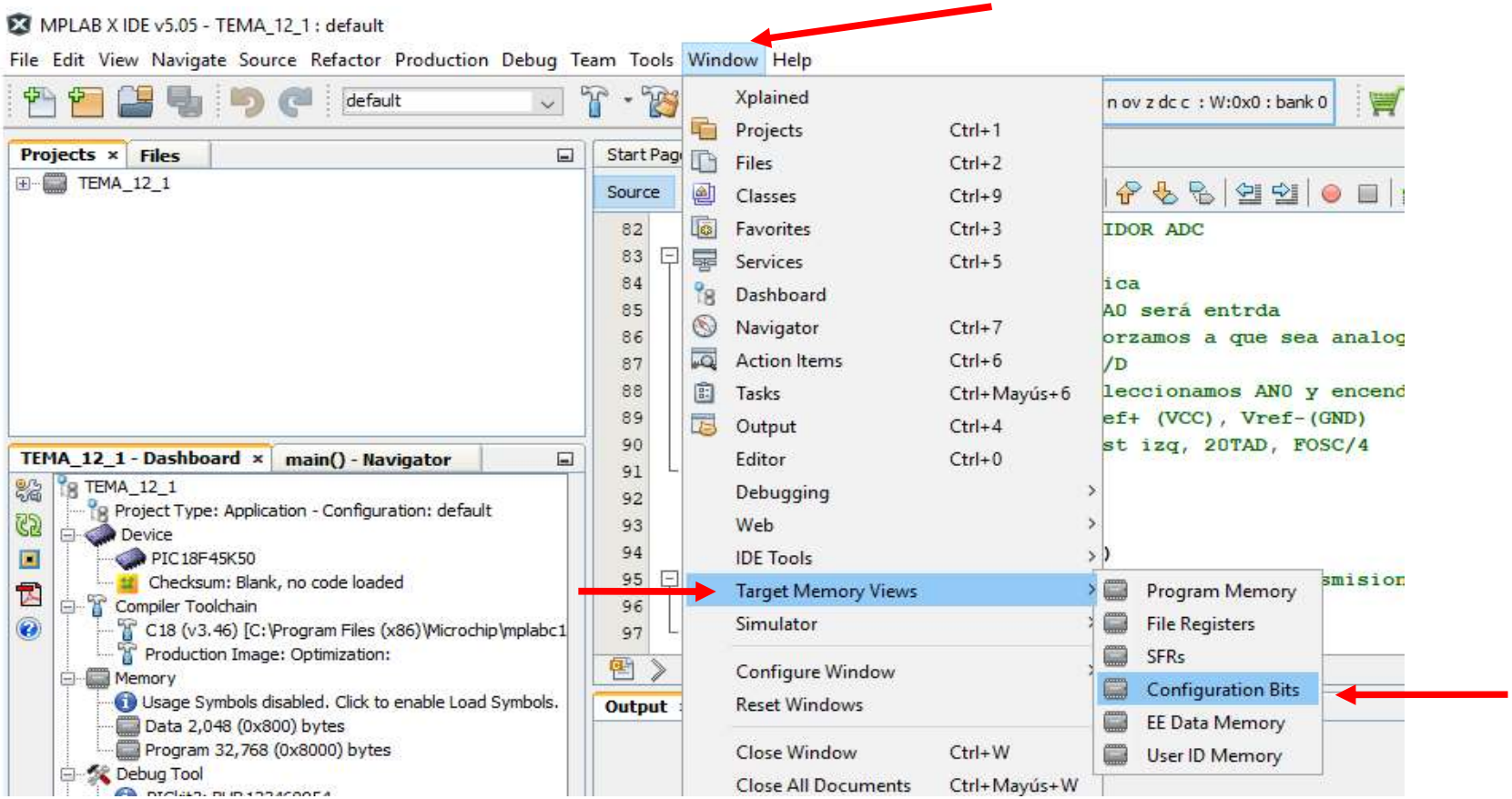
Programming the configuration bits

- Using special directives in the source code, the configuration bits can be made part of the application hex file.
- The values can also be changed using SFR[s related to writing and reading from the program flash memory (section 6.6 of the PIC manual)

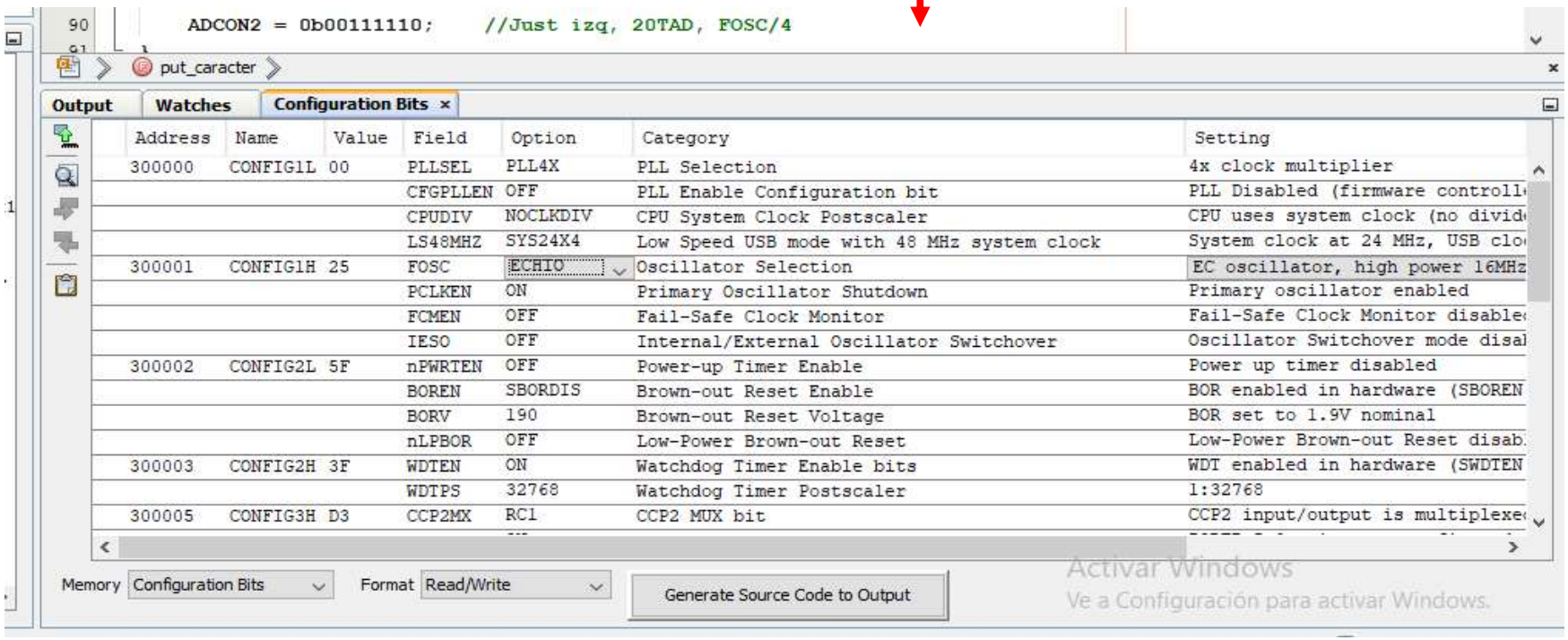
Setting the configuration bits in the source code

- We use the following pragma
  - #pragma config parameters
- We can use the MPLABx to define the setting.

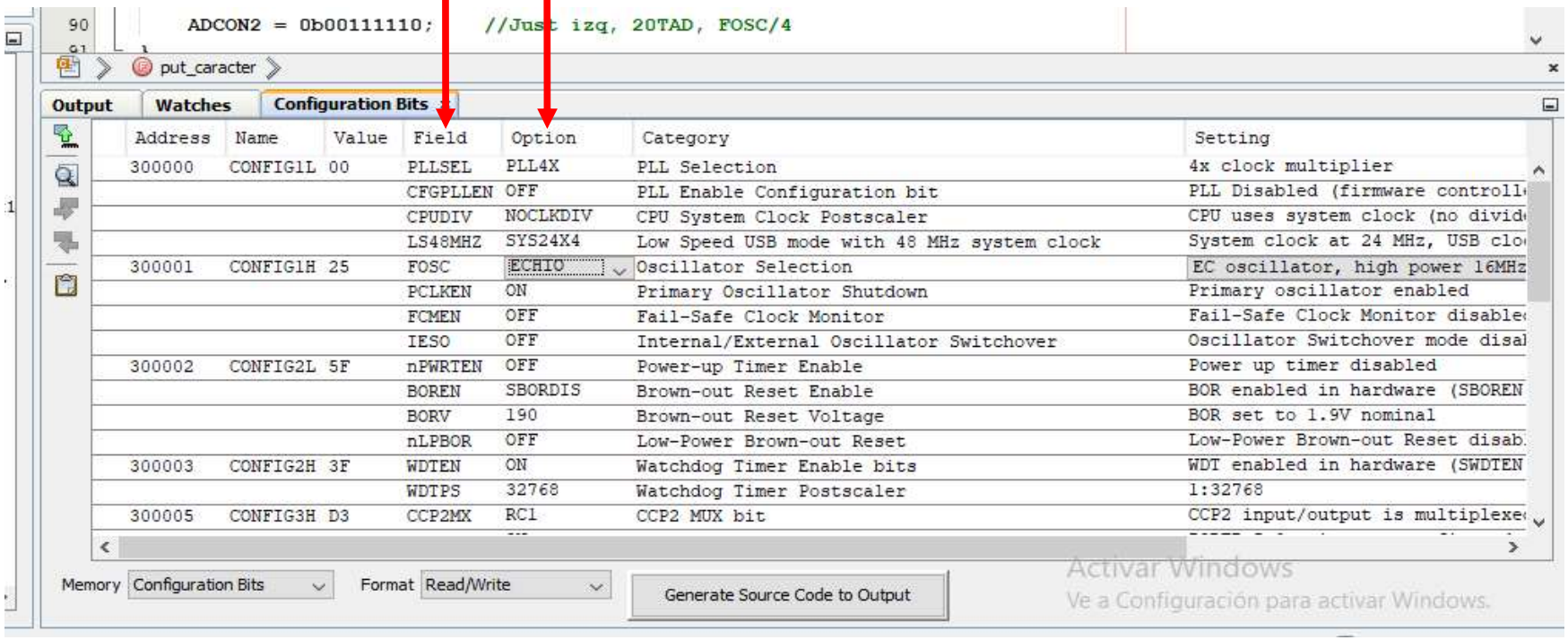
Using the MPLAB define the bits



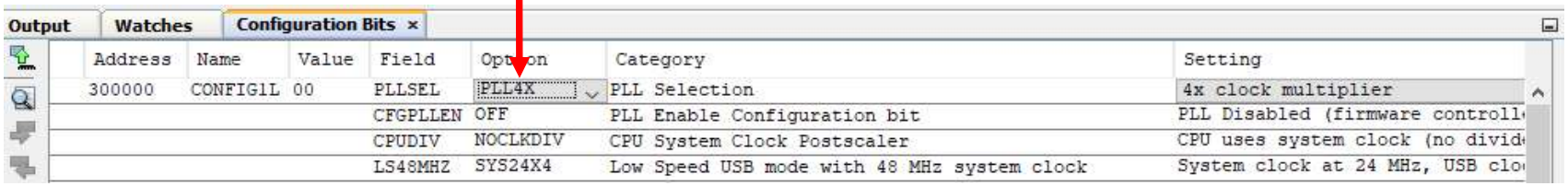
After: Window/Target memory views/Configuration bits a wintow will appear with the registers, the configuration bits and a menu of options for each bit



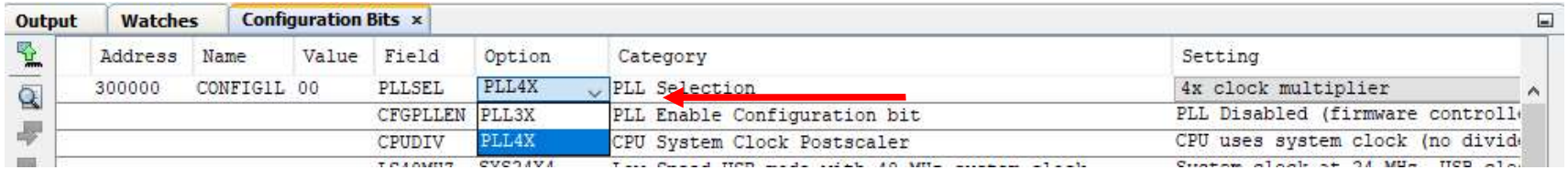
"Field" shows the name(s) of the bit(s), "Option" the mode it can have, there you select the one you want



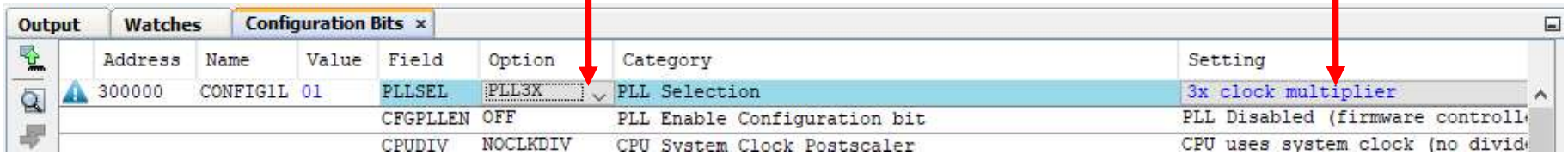
For example, select the value of PLLSEL



The following options will appear PLL3X y PLL4X

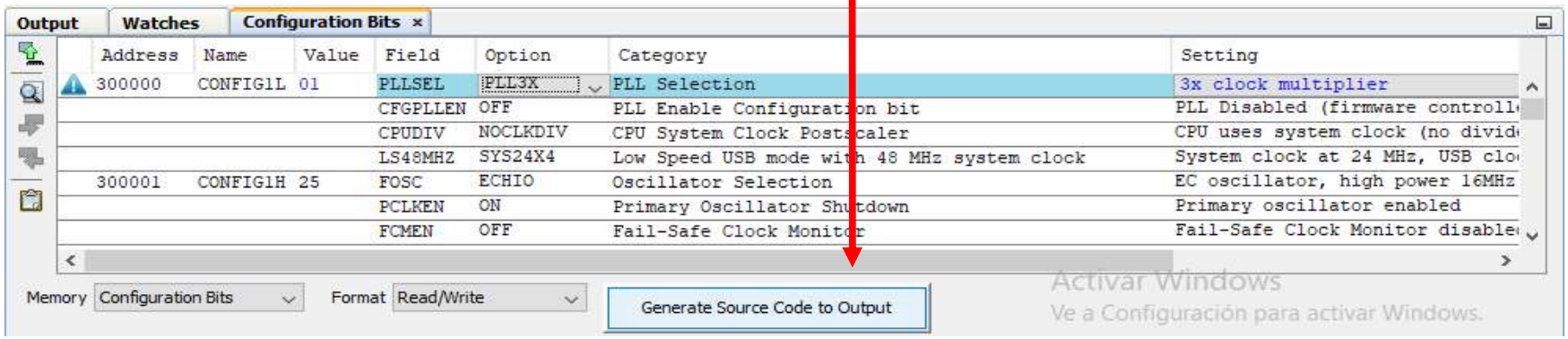


Selection a new option, “setting” will show the description :

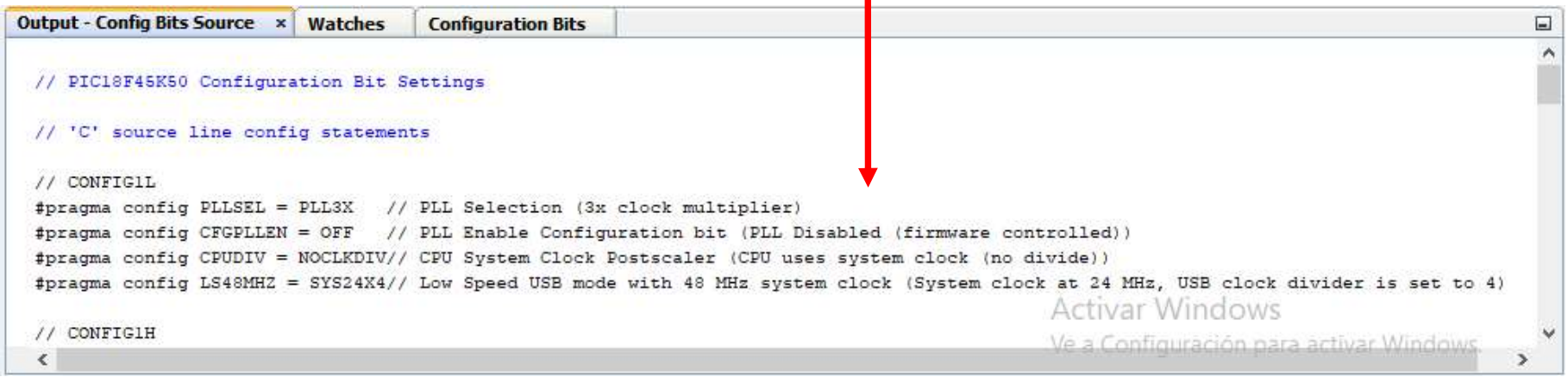


You repeat the process for each bit(s) you want to configure

When you finishg, click on "Generate Source Code Output"



A window will appear with the code where all the definitions are set. Copy this code preferably at the end of your source code, or make a header file (*.h) and include the file in the Project and also invoke the file at the beginning of your code



```
// PIC18F45K50 Configuration Bit Settings

// 'C' source line config statements

// CONFIG1L
#pragma config PLLSEL = PLL3X   // PLL Selection (3x clock multiplier)
#pragma config CFGPLLEN = OFF   // PLL Enable Configuration bit (PLL Disabled (firmware controlled))
#pragma config CPUDIV = NOCLKDIV// CPU System Clock Postscaler (CPU uses system clock (no divide))
#pragma config LS48MHZ = SYS24X4// Low Speed USB mode with 48 MHz system clock (System clock at 24 MHz, USB clock divider is set to 4)

// CONFIG1H
#pragma config FOSC = ECHIO     // Oscillator Selection (EC oscillator, high power 16MHz to 48MHz)
#pragma config PCLKEN = ON      // Primary Oscillator Shutdown (Primary oscillator enabled)
#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF       // Internal/External Oscillator Switchover (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config nPWRTEN = OFF    // Power-up Timer Enable (Power up timer disabled)
#pragma config BOREN = SBORDIS  // Brown-out Reset Enable (BOR enabled in hardware (SBOREN is ignored))
#pragma config BORV = 190       // Brown-out Reset Voltage (BOR set to 1.9V nominal)
#pragma config nLPBOR = OFF     // Low-Power Brown-out Reset (Low-Power Brown-out Reset disabled)

// CONFIG2H
#pragma config WDTEN = ON       // Watchdog Timer Enable bits (WDT enabled in hardware (SWDTEN ignored))
#pragma config WDTPS = 32768    // Watchdog Timer Postscaler (1:32768)

// CONFIG3H
#pragma config CCP2MX = RC1     // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON      // PORTB A/D Enable bit (PORTB<5:0> pins are configured as analog input channels on Reset)
#pragma config T3CMX = RC0      // Timer3 Clock Input MUX bit (T3CKI function is on RC0)
#pragma config SDOMX = RB3      // SDO Output MUX bit (SDO function is on RB3)
#pragma config MCLRE = ON       // Master Clear Reset Pin Enable (MCLR pin enabled; RE3 input disabled)

// CONFIG4L
#pragma config STVREN = ON      // Stack Full/Underflow Reset (Stack full/underflow will cause Reset)
#pragma config LVP = ON         // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled if MCLRE is also 1)
#pragma config ICPRT = OFF      // Dedicated In-Circuit Debug/Programming Port Enable (ICPORT disabled)
#pragma config XINST = OFF      // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode disabled)

// CONFIG5L
#pragma config CP0 = OFF        // Block 0 Code Protect (Block 0 is not code-protected)
#pragma config CP1 = OFF        // Block 1 Code Protect (Block 1 is not code-protected)
#pragma config CP2 = OFF        // Block 2 Code Protect (Block 2 is not code-protected)
#pragma config CP3 = OFF        // Block 3 Code Protect (Block 3 is not code-protected)
```

```
// CONFIG5H
#pragma config CPB = OFF        // Boot Block Code Protect (Boot block is not code-protected)
#pragma config CPD = OFF        // Data EEPROM Code Protect (Data EEPROM is not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF       // Block 0 Write Protect (Block 0 (0800-1FFFh) is not write-protected)
#pragma config WRT1 = OFF       // Block 1 Write Protect (Block 1 (2000-3FFFh) is not write-protected)
#pragma config WRT2 = OFF       // Block 2 Write Protect (Block 2 (04000-5FFFh) is not write-protected)
#pragma config WRT3 = OFF       // Block 3 Write Protect (Block 3 (06000-7FFFh) is not write-protected)

// CONFIG6H
#pragma config WRTC = OFF       // Configuration Registers Write Protect (Configuration registers (300000-3000FFh) are not write-protected)
#pragma config WRTB = OFF       // Boot Block Write Protect (Boot block (0000-7FFh) is not write-protected)
#pragma config WRTD = OFF       // Data EEPROM Write Protect (Data EEPROM is not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF      // Block 0 Table Read Protect (Block 0 is not protected from table reads executed in other blocks)
#pragma config EBTR1 = OFF      // Block 1 Table Read Protect (Block 1 is not protected from table reads executed in other blocks)
#pragma config EBTR2 = OFF      // Block 2 Table Read Protect (Block 2 is not protected from table reads executed in other blocks)
#pragma config EBTR3 = OFF      // Block 3 Table Read Protect (Block 3 is not protected from table reads executed in other blocks)

// CONFIG7H
#pragma config EBTRB = OFF      // Boot Block Table Read Protect (Boot block is not protected from table reads executed in other blocks)

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#include <xc.h>
```

```
#include<p18f45k50.h>       //Contiene las definiciones uc

//De los bits de configuracion estos son los que refieren al watch dog timer
#pragma config WDTEN = SWON         //WDT is controlled by SWDTEN bit of the WDTCON register
#pragma config WDTPS = 32           //WDT Presscaler 1:32

main(void){
    //Configura el oscilador interno a 16Mhz
    OSCCON = 0b01111100;            //OSC INTERNO DE 16MHZ

    //El Watch dog se configuró en pragma con prescala de 32 o sea que
    //Reiniciará al micro en aprox 32 * 4msec =128 msec si no le pegamos

    //Aquí habilitamos por software el WDT (esta activo ya pero no habilitado)
    //Si no le pegamos en 128msec, se va a reiniciar el micro

    WDTCONbits.SWDTEN = 1;

    //Aqui va mi codigo

    //Aqui le pego al perro o me resetea, lo tengo que hacer períodicamene antes de
    //Que se cumplan los 128msec
    ClrWdt();

}
```

How the configuration bits are transferred to the PIC ?

- In the *.hex file after register:
- 020000040030CA

```
:0600000063EF00F01200A6
:020006000000F8
:08000800060EF66E000EF76E05
:10001000000EF86E00010900F550656F0900F550FB
:10002000666F03E1656701D03DD00900F550606F50
:10003000090 0F550616F0900F550626F0900090071
:10004000F550E96E0900F550EA6E09000900090053
:10005000F550636F0900F550646F09000900F6CF91
:1000600067F0F7CF68F0F8CF69F060C0F6FF61C0C5
:10007000F7FF62C0F8FF0001635302E1645307E039
:100080000900F550EE6E6307F8E26407F9D767C020
:10009000F6FF68C0F7FF69C0F8FF00016507000EB2
:0600A000665BBFD71200F1
:0A00A600000EF36E00EE00F0060EEF
:1000B00001D81200EA6002D0EE6AFCD7F350E96082
:0600C0001200EE6AFCD7FD
:0A00C60015EE00F025EE00F0F86AD8
:1000D000019C04EC00F071EC00F072EC00F0FBD736
:0200E00012000C
:0200E20012000A
:0200E400120008
:020000040030CA
:0100010021DD
:010002001DE0
:0100030016E6
:00000001FF
```


http://www.kanda.com/blog/microcontrollers/pic-microcontrollers/pic-hex-file-format/

Non volatile memory EEPROM

- The PIC18F45K50 has 256 bytes of EEPROM
- Data stored in this memory are maintained if you remove the energy.
- This area can be used to store user configuration parameters, calibration, set-points, etc.

How to use this

- Section 8.8
- Registers involved are:

TABLE 8-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	IOCIE	TMR0IF	INT0IF	IOCIF	120
EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	—
EEDATA	EEPROM Data Register								—
EECON2	EEPROM Control Register 2 (not a physical register)								—
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	110
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	130
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	124
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	127

Legend: — = unimplemented, read as '0'. Shaded bits are not used during EEPROM access.

Example

EXAMPLE 8-3: DATA EEPROM REFRESH ROUTINE

```
        CLRF    EEADR               ; Start at address 0
        BCF     EECON1, CFGS        ; Set for memory
        BCF     EECON1, EEPGD       ; Set for Data EEPROM
        BCF     INTCON, GIE         ; Disable interrupts
        BSF     EECON1, WREN        ; Enable writes
Loop                                ; Loop to refresh array
        BSF     EECON1, RD          ; Read current address
        MOVLW   55h                 ;
        MOVWF   EECON2              ; Write 55h
        MOVLW   0AAh                ;
        MOVWF   EECON2              ; Write 0AAh
        BSF     EECON1, WR          ; Set WR bit to begin write
        BTFSC   EECON1, WR          ; Wait for write to complete
        BRA     $-2
        INCFSZ  EEADR, F            ; Increment address
        BRA     LOOP                ; Not zero, do it again

        BCF     EECON1, WREN        ; Disable writes
        BSF     INTCON, GIE         ; Enable interrupts
```

EEPROM specifications

Data EEPROM Memory					
Byte Endurance	100K	—	—	E/W	-40°C to +85°C
VDD for Read/Write	VDDMIN	—	VDDMAX	V	Using EECON to read/ write
Erase/Write Cycle Time	—	3	4	ms	
Characteristic Retention	—	40	—	Year	Provided no other specifications are violated
Number of Total Erase/Write Cycles before Refresh(2)	1M	10M	—	E/W	-40°C to +85°C

http://www.kanda.com/blog/microcontrollers/pic-microcontrollers/pic-hex-file-format/