

## PIC Configuration:

Configuration bits are a collection of specialized bits that can only be modified at program time. Configuration bits are read during reset and enable or disable hardware features in the microcontroller. The features controlled by the configuration bits include, but are not limited to, the clock source, the Watchdog Timer (WDT), the Brown-Out Detector (BOD), and the Memory Read protection. Configuration bits are not executable code. Essentially, they are fuses located in the program memory space.

Each PIC microcontroller has its own set of configuration bits.

## PIC16F877A Configuration Bits

Configuration Bits are used to enable or disable different hardware features of the microcontroller. These are read during startup/reset and enable or disable different hardware features based on the values of bits. Configuration bits in pic16f877a can be used to control the hardware features of pic such as clock source, Power-up Timer, Watchdog Timer, Memory Code Protection, etc. The configuration bits are mapped to the program memory location 2007h. it belongs to the configuration memory space, not the user program memory space and hence it can be programmed only during the programming time.

The configuration bits are set using the directives written in the software. The configuration bit syntax can be understood from the compiler manual. The user can easily create the configuration code using the IDE such as MPLAB X IDE. The configuration bit window in the IDE can be used to generate suitable directives according to the selected options. The tutorial discusses different configuration bits in PIC16F877A and the method to generate desired directives using MPLAB X IDE.

## Configuration bits in PIC16F877A

### Oscillator Configurations

The Oscillator provides the required clock signal for the operation of the microcontroller. The pic16f877a has four oscillator modes namely LP, XT, HS, and RC. The modes LP, XT, HS utilizes a crystal or a ceramic resonator and it is connected to the clock input pins to generate the required oscillations. The LP mode is designed to drive 32KHZ Crystal. The XT mode is selected for the crystals in the range of 4MHZ or lower and in the case of HS mode the crystal range is above 4MHZ. The RC Mode is selected when the RC network is used to create the clock input for the MCU. The oscillation frequency accuracy is less in RC Network. It can be used in time-insensitive applications for cost-saving.

```
#pragma config FOSC = EXTRC      // Oscillator Selection bits (RC  
oscillator)
```

```
#pragma config FOSC = HS          // Oscillator Selection bits (HS
oscillator)
#pragma config FOSC = XT          // Oscillator Selection bits (XT
oscillator)
#pragma config FOSC = LP          // Oscillator Selection bits (LP
oscillator)
```

## Watchdog Timer (WDT)

Watchdog timer detects when the microcontroller is stuck in an endless loop. It is a free-running counter that generates devices reset when the WDT Time out occurs. In the sleep mode operation, WDT time out wake up the device to continue its operation. The program writes zero every time. Watchdog timer detects the software malfunctioning and resets the controller. watchdog timer counts from an initial value set by the controller to the zero value. The embedded software restarts the WDT Before reaching zero. In the software malfunctioning events, the counter reaches zero and it generates the reset signal.

```
#pragma config WDTE = ON          // Watchdog Timer Enable bit (WDT enabled)
#pragma config WDTE = OFF        // Watchdog Timer Enable bit (WDT disabled)
```

## Power-up Timer (PWRT)

The power-up timer introduces a small delay after power-up reset or brown-out reset. The quick start of the microcontroller after the rest can cause problems. The power-up timer provides a startup delay of 72 ms which allows VDD to rise to an acceptable level. The reset state of the chip is retained as long as the PWRT is active. The PWRT ensures the stability of the supply voltage before the clock gets started.

```
#pragma config PWRTE = ON        // Power-up Timer Enable bit (PWRT enabled)
#pragma config PWRTE = OFF        // Power-up Timer Enable bit (PWRT
disabled)
```

## Brown-out Reset (BOR)

Brown out reset allows the microcontroller to reset when the supply voltage drops below a specific voltage. IF the VDD falls below VBOR(~4V) For a duration of TBOR (100  $\mu$ S) it reset the device. The device will remain in a reset state in the BROWN out reset case until the VDD again rises above VBOR and after that, the Power-up Timer keeps the device in reset for a time duration of TPWRT ( 72 mS). The device which falls below VBOR In the TPWRT initiates BROWN out restart when VDD rises above VBOR with the Power-up Timer Reset. Regardless of the state of the PWRT configuration bit, the power-up timer is enabled with the Brown-out Reset activation.

```
#pragma config BOREN = ON        // Brown-out Reset Enable bit (BOR enabled)
#pragma config BOREN = OFF        // Brown-out Reset Enable bit (BOR
disabled)
```

## Low Voltage (SINGLE SUPPLY) ICSP

The LVP bit is for enabling low voltage programming mode. The advantage of low voltage programming is that the device can be programmed with ordinary logic levels whereas, in high voltage programming, the MCLR pin should be connected to the voltage level more than Vdd. In low voltage programming mode, it uses only a single supply. To enter programming mode, PGM bit should be connected to Vdd, so the bit PGM (RB5) will not be available for I/O operation.

```
#pragma config LVP = ON           /*Low-Voltage (Single-Supply) In-Circuit
Serial Programming Enable bit    (RB3/PGM pin has PGM function; low-voltage
programming enabled)*/
#pragma config LVP = OFF         /*Low-Voltage (Single-Supply) In-Circuit
Serial Programming Enable bit    (RB3 is digital I/O, HV on MCLR must be
used for programming)*/
```

## Data EEPROM Memory Code Protection bit (CPD)

The two bits CPD and WRTD protects the EEPROM Data from external reads and writes. CPD controls internal and external reads and writes whereas WRTD controls internal reads and writes.

```
#pragma config CPD = ON           // Data EEPROM Memory Code Protection bit
(Data EEPROM code-protected)
#pragma config CPD = OFF          // Data EEPROM Memory Code Protection bit
(Data EEPROM code protection off)
```

## Flash Program Memory Write Enable bit

These bits allow us to select the sectors in the flash programming memory for recording of the data or for the In-circuit Serial Programming. WE can write to the flash memory by the use of EECON. The firmware can be used to write data into the flash memory.

```
#pragma config WRT = OFF          // Flash Program Memory Write Enable bits
(Write protection off; all program memory may be written to by EECON
control)
#pragma config WRT = 256          // Flash Program Memory Write Enable bits
(0000h to 00FFh write-protected; 0100h to 1FFFh may be written to by EECON
control)
#pragma config WRT = 1FOURTH     // Flash Program Memory Write Enable bits
(0000h to 07FFh write-protected; 0800h to 1FFFh may be written to by EECON
control)
#pragma config WRT = HALF         // Flash Program Memory Write Enable bits
(0000h to 0FFFh write-protected; 1000h to 1FFFh may be written to by EECON
control)
```

