

16F628A - The Configuration Word

© Jon Wilder

25 de agosto de 2011.

Tradução e edição Prof. esp eng° Luiz Antonio Vargas Pinto
22 de Fevereiro de 2015

A palavra de configuração (word config) está entre os itens que mais assustam os novatos em PIC.
A config word em toda a família PIC16F é uma palavra de 14 bits onde cada um deles liga/desliga alguma função específica do PIC.

São estas as funções:

1. OSC selection - Habilita o tipo de oscilador (interno, externo RC ou cristal externo)
2. Reset - Habilita MCLR entre External Master Clear ou Entrada digital
3. Power-on Reset (POR)
4. Power-up Timer (PWRT)
5. Oscillator Start-Up Timer (OST)
6. Brown-out Reset (BOR)
7. Interrupts
8. Watchdog Timer (WDT)
9. Sleep
10. Code protection (impede a recuperação dos códigos de programa via hardware)
11. ID Locations (impede a recuperação dos dados da EEPROM via hardware)
12. In-Circuit Serial Programming™ (ICSP™)
13. Low Voltage Programming

Uma descrição completa da config word assim como a própria config word pode ser encontrada no Datasheet na seção "Special Features of the CPU". Essas características dependem do PIC em uso e devem ser respeitadas. Alguns PICs tem mais itens que outros, e isto significa que existirão mais bits para configurar. Em alguns casos, existem até duas config words.

A config word em todos os PICs é somente gravável em tempo de programação, isto é, após o programa ser gravado em flashrom não podem ser acessados. Em outras palavras, esses bits não podem ser acessados por programação em uso, razão pelo que, as vezes se referem a esses bits como "fusíveis".

Uma vez escritos só é possível alterá-los regravando o PIC.

Em muitos PICs da família 16F, a config word residem-no endereço `0x2007` da ROM. No caso de PICs que possuem duas Config words tal como o 16F887 isto reside nos endereços `0x2007` e `0x2008`.

No 16F628A, a config word é descrita como segue:

| | | | | | | | | | | | | | | |
|-----|------------------------|----|----|----|---|-------------------------|-----|-------|--------|-------|---|------|-------|-------|
| | $\overline{\text{CP}}$ | x | x | x | x | $\overline{\text{CPD}}$ | LVP | BOREN | MCLRRE | FOSC2 | $\overline{\text{PWRT}}\overline{\text{E}}$ | WDTE | FOSC1 | FOSC0 |
| Bit | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Considere o seguinte caso:

| | | | | | | | | | | | | | | |
|-----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Bit | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Por default, todos os bits de configuração são inicializados em 1 assim somente precisamos configurar os 0s.

No 16F628A, os bits de 9 a 12 não estão implementados e são lidos como 0. (Entretanto, no 16F887, todos os bits não implementados são lidos como 1.

Para Low Voltage Programming (LVP), Brown Out Reset (BOREN), External Master Clear (MCLR) e o Watchdog Timer (WDTE), colocar esses bits para 0 desligará as funções e ligará o MCLR como entrada digital.

Para Program Code Protection (CP), Data Code Protection (CPD) e Power Up Timer (PWRTE), colocar em 0 habilitará essas funções.

O tipo de oscilador é definido pelos bits 1, 2 e 5 (FOSC2, FOSC1 e FOSC0). Há um total de oito possíveis configurações de oscilador. Escrevendo os seguintes padrões com 3 bits teremos:

| FOSC2 | FOSC1 | FOSC0 | |
|-------|-------|-------|--|
| 1 | 1 | 1 | Oscilador externo RC: CLKOUT função no pino RA6/OSC2/CLKOUT, Resistor e Capacitor em RA7/OSC1/CLKIN |
| 1 | 1 | 0 | Oscilador externo RC: I/O função no pino RA6/OSC2/CLKOUT, Resistor e Capacitor em RA7/OSC1/CLKIN |
| 1 | 0 | 1 | INTOSC Oscilador: CLKOUT função no pino RA6/OSC2/CLKOUT, I/O função em RA7/OSC1/CLKIN |
| 1 | 0 | 0 | INTOSC Oscilador: I/O função no pino RA6/OSC2/CLKOUT, I/O função em RA7/OSC1/CLKIN |
| 0 | 1 | 1 | EC (External Clock): I/O função no pino RA6/OSC2/CLKOUT, CLKIN em RA7/OSC2/CLKIN |
| 0 | 1 | 0 | HS Oscilador: High-speed cristal/ressonador (8MHz - 20MHz) em RA6/OSC2/CLKOUT e RA7/OSC1/CLKIN |
| 0 | 0 | 1 | XT Oscilador: Cristal/ressonador (100kHz - 4MHz) em RA6/OSC2/CLKOUT e RA7/OSC1/CLKIN |
| 0 | 0 | 0 | LP Oscilador: Low-power cristal/ressonador (32kHz - 200kHz) em RA6/OSC2/CLKOUT e RA7/OSC1/CLKIN |

Descrição:

bit 13: CP: Flash Program Memory Code Protection bit⁽²⁾

(PIC16F648A)

1 = Code protection off

0 = 0000h to 0FFFh code-protected

(PIC16F628A)

1 = Code protection off

0 = 0000h to 07FFh code-protected

(PIC16F627A)

1 = Code protection off

0 = 0000h to 03FFh code-protected

bit 12-9: Unimplemented: Read as '0'

bit 8: CPD: Data Code Protection bit⁽³⁾

1 = Data memory code protection off

0 = Data memory code-protected on

bit 7: LVP: Low-Voltage Programming Enable bit

1 = RB4/PGM pino tem função PGM, low-voltage programming ligado

0 = RB4/PGM is digital I/O, HV on MCLR must be used for programming

bit 6: BOREN: Brown-out Reset Enable bit ⁽¹⁾

1 = BOR Reset enabled

0 = BOR Reset disabled

bit 5: MCLRE: RA5/MCLR/VPP Pin Function Select bit

1 = RA5/MCLR/VPP pin function is MCLR

0 = RA5/MCLR/VPP pin function is digital Input, MCLR internally tied to VDD

bit 3: PWRTE: Power-up Timer Enable bit ⁽¹⁾

1 = PWRT disabled

0 = PWRT enabled

bit 2: WDTE: Watchdog Timer Enable bit

1 = WDT enabled

0 = WDT disabled

Para gravar a config word existe duas formas:

1. Alguns softwares de programação permitirão a você configurar o config fuse settings no programa por alguma ferramenta tal como checkbox (caixa de escolha).
2. O mais comum é usar a diretiva "__config" no topo de seu código de programa (observe ainda que se trata de um duplo underscore precedendo o config).

O uso mais popular e simplificado do config word do 16F628A é __0x3D18. Vamos avaliar isto sob o prisma da figura seguinte:

| | | | | | | | | | | | | | | |
|-----|------------------------|----|----|----|---|-------------------------|-----|-------|-------|-------|---------------------------|------|-------|-------|
| | $\overline{\text{CP}}$ | x | x | x | x | $\overline{\text{CPD}}$ | LVP | BOREN | MCLRE | FOSC2 | $\overline{\text{PWRTE}}$ | WDTE | FOSC1 | FOSC0 |
| Bit | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

O qual interpretaremos da seguinte forma:

1. Sem proteção de código
2. EEPROM sem proteção
3. O pino RB4/PGM é usado para I/O
4. Brown-Out reset desligado
5. RA5 será Input digital – MCLR off
6. Oscilador Interno
7. Power-UP desligado
8. Watch Dog Timer desligado

bit 4, 1-0: FOSC<2:0>: Oscillator Selection bits⁽⁴⁾

- 111 = RC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, Resistor and Capacitor on RA7/OSC1/CLKIN
- 110 = RC oscillator: I/O function on RA6/OSC2/CLKOUT pin, Resistor and Capacitor on RA7/OSC1/CLKIN
- 101 = INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
- 100 = INTOSC oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
- 011 = EC: I/O function on RA6/OSC2/CLKOUT pin, CLKIN on RA7/OSC1/CLKIN
- 010 = HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
- 001 = XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
- 000 = LP oscillator: Low-power crystal on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN

Note

1. Habilitar Brown-out Reset não habilita automaticamente o Power-up Timer (PWRT) – aplicável aos PIC16F627/628.
2. O esquema de proteção de código (code protection) foi alterado no código usado para os PIC16F627/628. O programa inteiro na Flash precisa ser apagado para setar o bit CP, desligando o code protection. Veja “PIC16F627A/628A/648A EEPROM Memory Programming Specification” (DS41196) para mais detalhes.
3. Todos os dados na EEPROM precisam ser apagado para setar o bit CPD, desligando o code protection. Veja “PIC16F627A/628A/648A EEPROM Memory Programming Specification” (DS41196) para mais detalhes.
4. Quando o MCLR é ativado no modo INTOSC, o oscilador de cloxk interno é desabilitado.

Legend:

R = bit legível W = bit gravável U = não implementado, lido como '0'
 -n = valor em POR '1' = bit is set '0' = bit is cleared x = bit desconhecido

MPASM traz arquivos de inclusão(include) para cada PIC, e nestes arquivos, eles tem labels iguais para cada bit do config word. Desde que os fusíveis são inicializados para 1 por default, nós somente precisamos usar os labels para os fusíveis que nós necessitamos por em 0. O método de label é preferido porque todos os PICs usam basicamente os mesmo labels e isto provê portabilidade entre códigos de PICs

Exemplos:

```
include    <P16F628A.INC>
__config  _LVP_OFF & _BOREN_OFF & _PWRT_ON and _WDT_OFF & _HS_OSC
```

Sugestão

| | CP | x | x | x | x | CPD | LVP | BOREN | MCLRE | FOSC2 | PWRT | WDTE | FOSC1 | FOSC0 |
|-----|----|----|----|----|---|-----|-----|-------|-------|-------|------|------|-------|-------|
| Bit | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

Ou seja 0x3F09

O qual interpretamos da seguinte forma:

1. Sem proteção de código
2. EEPROM sem proteção
3. O pino RB4/PGM é usado para I/O
4. Brown-Out reset desligado
5. RA5 será Input digital – MCLR off
6. Oscilador Externo de 4 MHz em RA6 e RA7
7. Power-UP desligado
8. Watch Dog Timer desligado