

Lucrarea . Biți de configurare

Objective

Înțelegerea rolului bitilor de configurare și a modului în care aceștia pot fi setați:

Înainte de a începe programarea unui microcontroller, este necesar să se stabilească condițiile de funcționare ale microcontrolerului. Printre aceste condiții se pot enumera: definirea frecvenței semnalului de ceas, viteza de execuție a instrucțiunilor, sursa de ceas, modul de lucru pentru pini cu anumite funcționalități specifice (programare/depanare, comunicație), protecția împotriva executării unor părți greșite de program, etc.

După cum se poate observa în schema electrică din Fig.1, placa folosită la laborator a fost proiectată astfel încât programatorul PIC KIT 3 să comunice cu microcontrolerul prin cei doi pini PGD1 și PGC1.

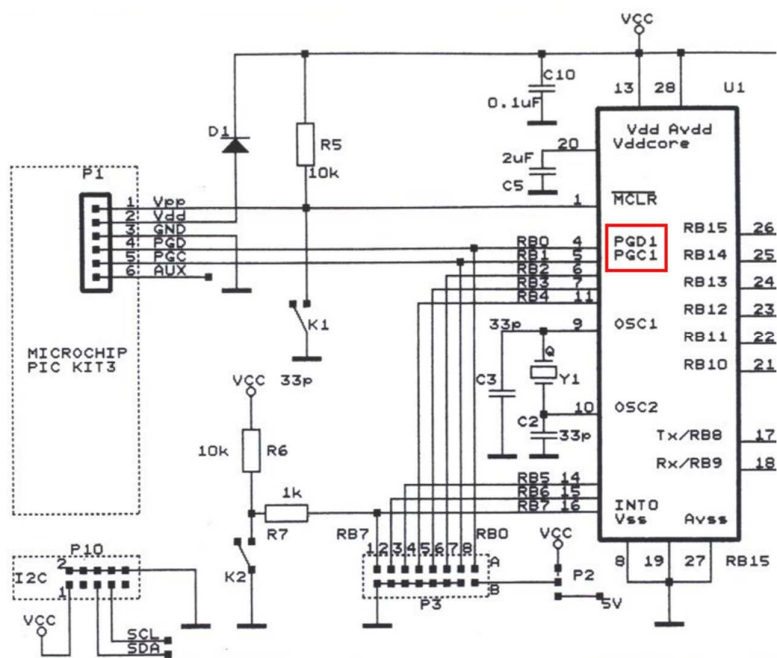


Fig.1: Schema electrică a microcontrolerului folosit la laborator

Microcontroler- ul poate fi însă programat pentru ca PIC KIT 3 să comunice prin una din cele 3 perechi de pini disponibile pe microcontroler, cele marcate cu roșu în Fig.2.

28-Pin SDIP, SOIC

SDIP, SOIC		Pin		Function	
MCLR	1	28	AVDD		
AN0/VREF-/CN2/RA0	2	27	AVss		
AN1/VREF-/CN3/RA1	3	26	PWM1L1/RP15 ⁽¹⁾ /CN11/PMCS1/RB15		
PGED1/AN2/C2IN-/RP0 ⁽¹⁾ /CN4/RB0	4	25	PWM1H1/RTCC/RP14 ⁽¹⁾ /CN12/PMWR/RB14		
PGEC1/AN3/C2IN+/RP1 ⁽¹⁾ /CN5/RB1	5	24	PWM1L2/RP13 ⁽¹⁾ /CN13/PMRD/RB13		
AN4/C1IN-/RP2 ⁽¹⁾ /CN6/RB2	6	23	PWM1H2/RP12 ⁽¹⁾ /CN14/PMD0/RB12		
AN5/C1IN+/RP3 ⁽¹⁾ /CN7/RB3	7	22	PGEC2/TMS/PWM1L3/RP11 ⁽¹⁾ /CN15/PMD1/RB11		
Vss	8	21	PGED2/TDI/PWM1H3/RP10 ⁽¹⁾ /CN16/PMD2/RB10		
OSC1/CLKI/CN30/RA2	9	20	VCAP/VDDCORE		
OSC2/CLKO/CN29/PMA0/RA3	10	19	Vss		
SOSCI/RP4 ⁽¹⁾ /CN1/PMBE/RB4	11	18	TDO/PWM2L1/SDA1/RP9 ⁽¹⁾ /CN21/PMD3/RB9		
SOSCO/T1CK/CN0/PMA1/RA4	12	17	TCK/PWM2H1/SC1/RP8 ⁽¹⁾ /CN22/PMD4/RB8		
VDD	13	16	INT0/RP7 ⁽¹⁾ /CN23/PMD5/RB7		
PGED3/ASDA1/RP5 ⁽¹⁾ /CN27/PMD7/RB5	14	15	PGEC3/ASCL1/RP6 ⁽¹⁾ /CN24/PMD6/RB6		

■ = Pins are up to 5V tolerant

Fig.2: Capsula microcontrolerului folosit la laborator

Pentru a alege perechea de pini prin care comunică programatorul, se poate utiliza bitul de configurare dedicat, ICS, așa cum apare și în Tabelul 28-2.

Un alt exemplu de configurare necesară se referă la interfața de comunicație serială I2C. Aceasta este o interfață folosită în sistemele încorporate pentru a realiza comunicații între microcontroller și senzori.

Interfața de comunicație serială I2C se bazează pe două semnale, un semnal de cceas și un semnal de date transmise prin pinii predefiniți SDA și SCL sau prin pinii alternativi ASDA și ASCL marcați cu albastru în Fig. 1. Pentru a alege perechea de pini alternativi, se folosește bitul de configurare ALTI2C, după cum se explică și în Tabelul 28-2.

TABLE 28-2: dsPIC33F CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
ALTI2C	FPOR	Alternate I ² C™ pins 1 = I ² C mapped to SDA1/SCL1 pins 0 = I ² C mapped to ASDA1/ASCL1 pins
JTAGEN	FICD	JTAG Enable bit 1 = JTAG enabled 0 = JTAG disabled
ICS<1:0>	FICD	ICD Communication Channel Select bits 11 = Communicate on PGEC1 and PGED1 10 = Communicate on PGEC2 and PGED2 01 = Communicate on PGEC3 and PGED3 00 = Reserved, do not use

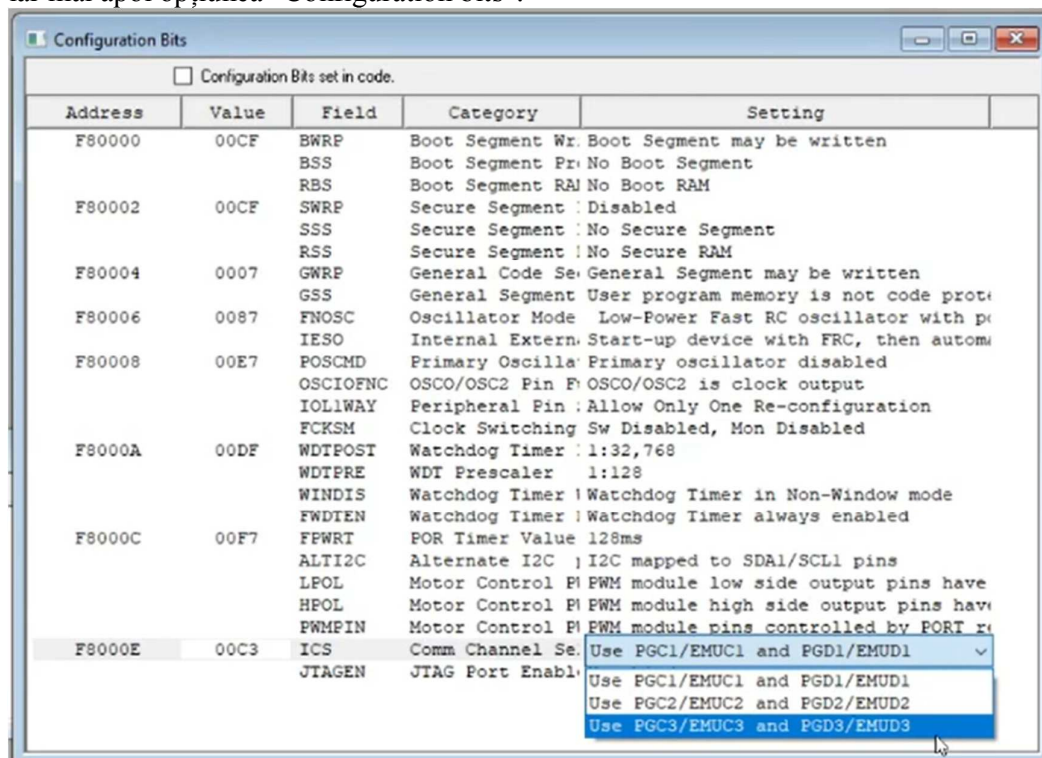
Configurarea microcontroller-ului DSPIC33FJ32MC302

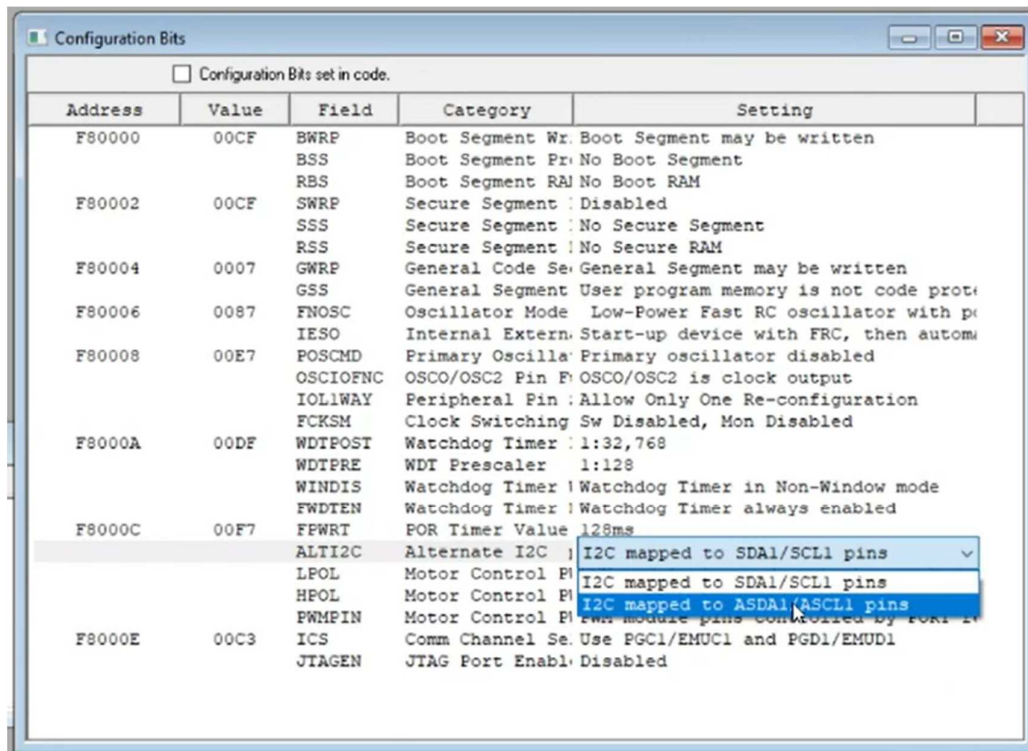
Configurarea biților se poate face în 2 moduri:

- *direct din softul MPLAB IDE*

Exemple de configurare a biților direct din softul MATLAB IDE:

OBS: Pentru a putea ajunge la panoul de configurare a biților, se selectează “Configure” din meniu, iar mai apoi opțiunea “Configuration bits”.





- *în codul sursă*

Exemplu de configurare a biților în codul sursă:

```

1  #include "p33Fxxx.h"
2
3  // Select Internal FRC at POR
4  _FOSCSEL(FNOSC_FRC);
5  // Enable Clock Switching and Configure
6  _FOSC(FCKSM_CSECMD & OSCIOFNC_OFF);
7
8  void delay(unsigned int t)
9  {
10     unsigned int x;
11     unsigned int i;
12     for(i=0;i<650;i++)
13     {
14         for(x=0;x<t;x++);
15     }
16
17     void initPLL(void)
18     {

```

Avantajul configurării în codul sursă este acela că informația se păstrează, programul putând fi rulat de pe orice calculator, iar dezavantajul acestei metode este înțelegerea, interpretarea mai dificilă a instrucțiunilor.

Avantajul configurării direct în softul MPLAB IDE este ușurința interpretării și setării, iar dezavantajul este faptul că acei biți sunt setați în fișierul proiect, fiind nevoie să se refacă această configurare a biților atunci când se dorește copierea fișierelor sursă sau să fie necesară copierea întregului proiect.

Printre regiștrii importanți în configurarea microcontrolerului putem enumera:

- **FBS (Boot Code Segment Configuration Register)**
- **FGS (General Code Segment Configuration Register)**
- **FOSCSEL (Oscillator Source Selection Register)**

- FOSC (Oscillator Selection Configuration Bits Register)
- FWDT (Watchdog Timer Configuration Register)
- FPOR (POR Configuration Register)
- FICD (In-Circuit Debugger Configuration Register)

TABLE 28-1: DEVICE CONFIGURATION REGISTER MAP

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FBS	RBS<1:0>		—	—	BSS<2:0>			BWRP
0xF80002	FSS ⁽¹⁾	RSS<1:0>		—	—	SSS<2:0>			SWRP
0xF80004	FGS	—	—	—	—	—	GSS<1:0>		GWRP
0xF80006	FOSCSEL	IESO	—	—	—		FNOSC<2:0>		
0xF80008	FOSC	FCKSM<1:0>		IOL1WAY	—	—	OSCIOFNC	POSCMD<1:0>	
0xF8000A	FWDT	FWDTEN	WINDIS	—	WDTPRE	WDTPOST<3:0>			
0xF8000C	FPOR	PWMPIN	HPOL	LPOL	ALT12C	—	FPWRT<2:0>		
0xF8000E	FICD	Reserved ⁽²⁾		JTAGEN	—	—	—	ICS<1:0>	

```

** Only one invocation of FICD should appear in a project,
** at the top of a C source file (outside of any function).
**
** The following constants can be used to set FICD.
** Multiple options may be combined, as shown:
**
** _FICD( OPT1_ON & OPT2_OFF & OPT3_PLL )
**
** JTAG Enable Bit:
**     JTAGEN_OFF           JTAG is disabled
**     JTAGEN_ON            JTAG is enabled
**
** ICD communication channel select bits:
**     ICS_NONE             Reserved
**     ICS_PGD3             communicate on PGC3/EMUC3 and PGD3/EMUD3
**     ICS_PGD2             communicate on PGC2/EMUC2 and PGD2/EMUD2
**     ICS_PGD1             communicate on PGC1/EMUC1 and PGD1/EMUD1
**
** /

#define JTAGEN_OFF           0xFFDF
#define JTAGEN_ON           0xFFFF

#define ICS_NONE            0xFFFC
#define ICS_PGD3            0xFFFD
#define ICS_PGD2            0xFFFE
#define ICS_PGD1            0xFFFF

```

a. Registrul FBS

Cu ajutorul acestui registru putem configura mărimea memoriei program, mărimea memoriei RAM, dar și securitatea memoriei program.

BWRB - Memoria program este:

- 1 = Neprotejată (poate și scrisă).
- 0 = Protejată (asemenea unui boot loader).

BSS<2:0> - Mărimea memoriei program:

X11 = Fără memorie program.

- Mărimea este de 1K.

110 = Securitate standard; memoria program se termină la adresa 0x0007FE.

010 = Securitate ridicată; memoria program se termină la adresa 0x0007FE.

- Mărimea este de 4K.

101 = Securitate standard; memoria program se termină la adresa 0x001FFE.

001 = Securitate ridicată; memoria program se termină la adresa 0x001FFE.

- Mărimea este de 8K.

100 = Securitate standard; memoria program se termină la adresa 0x003FFE.

000 = Securitate ridicată; memoria program se termină la adresa 0x003FFE.

RBS<1:0> - Mărimea memoriei RAM.

11 = Nu este definită.

10 = Memorie RAM de 128 bytes.

01 = Memorie RAM de 256 bytes.

00 = Memorie RAM de 1024 bytes.

b. Registrul FGS

Acest registru poate fi configurat pentru a oferi protecție la scrierea programului.

GSS<1:0> - Protecția segmentului general de cod

11 = Fără protecția programului.

10 = Securitate standard.

0x = Securitate ridicată.

GWRP - Protecția programului la scriere

1 = Programul este neprotejat la scriere.

0 = Programul este protejat la scriere.

c. Registrul FOSCSEL

Acest registru poate fi configurat pentru a alege tipul oscilatorului (intern, extern, secundar).

IESO - Bitul de configurare și selecție al oscilatorului sursă

1 = Pornire cu FRC, apoi trecere automată pe oscilatorul selectat.

0 = Pornire directă cu oscilatorul selectat.

FNOSC<2:0> - Selecția oscilatorului implicit

111 = Oscilator intern Fast RC (FRC) cu postscaler.

110 = Oscilator intern Fast RC (FRC) cu frecvența cristalului de quartz /16.

101 = Oscilator LPRC (low power).

100 = Oscilator secundar LP.

011 = Oscilator primar cu PLL (XT, HS, EC).

010 = Oscilator primar (XT, HS, EC).

001 = Oscilator intern Fast RC (FRC) cu PLL.

000 = Oscilator FRC.

Oscilatorul intern FRC funcționează normal cu frecvența cristalului de quartz la 7.37MHz. Aceasta poate fi divizată pe o scală de la 1:2 la 1:256 folosind bitul **FRCDIV**.

Oscilatorul primar poate folosi următoarele surse de ceas:

- Cristal de quartz (XT) cu frecvența între 3-7 MHz conectat la pinii OSC1 și OSC2. Oscilatorul

XT este conceput pentru a oferi un compromis între o performanță la frecvență ridicată și un consum modest de energie.

- Cristal de quartz (HS) cu frecvență între 10-40MHz conectat la pinii OSC1 și OSC2. Oscilatorul HS este conceput pentru a oferi beneficiu maxim. Este aproximativ de 5 ori mai profitabil decât un oscilator XT din punct de vedere al performanțelor.

- Sursa de ceas externă legată direct la pinul OSC1.

Oscilatorul secundar LP (Low Power) este conceput pentru puteri mici și folosește un cristal de quartz de 32.768 kHz. Este conectat la pinii SOSCI și SOSCO.

d. Registrul FOSC

FCKSM<1:0> - Modul comutării ceasului

1X = Comutarea ceasului dezactivată, Fail-Safe Clock, Monitor dezactivat.

01 = Comutarea ceasului activată Fail-Safe Clock, Monitor dezactivat.

00 = Comutarea ceasului activată Fail-Safe Clock, Monitor activat.

IOL1WAY - Bit configurare periferice

1 = Permite o singură reconfigurare.

0 = Permite mai multe reconfigurări.

OSCIOFNC - Funcția pinului OSC2 (excepție în modurile XT și HS)

1 = OSC2 clock-output.

0 = cu scop genera digital I/O.

POSCMD<1:0> - Modul oscilatorului primar

11 = Oscilator primar dezactivat.

10 = Mod oscillator HS.

01 = Mod oscillator XT.

00 = Mod EC (external clock).

e. Registrul FWDT

FWDTEN - Bit de activare watchdog timer

1 = watchdog timer mereu activat.

0 = watchdog timer activat sau dezactivat prin soft.

WINDIS - Bit de activare a watchdog timer window

1 = Mod de funcționare în non-window.

0 = Mod de funcționare în window.

WDTPRE - Watchdog timer prescaler

1 = 1:128

0 = 1:32

WDTPOST<3:0> - Watchdog timer postscaler

1111 = 1:32,768

1110 = 1:16,384

0001 = 1:2

0000 = 1:1

f. Registrul FPOR

PWMPIN: Pin al modului PWM de control al motorului

1 = Pini modului PWM controlați de registrul PORT la resetarea dispozitivului (tri-stated)

0 = Pini modului PWM controlați de modulul PWM la resetarea dispozitivului (configurați ca pini de ieșire)

HPOL: Motor Control PWM High-Side Polarity bit

1 = PWM module high-side output pins have active-high output polarity

0 = PWM module high-side output pins have active-low output polarity

LPOL: Motor Control PWM Low-Side Polarity bit

1 = PWM module low-side output pins have active-high output polarity

0 = PWM module low-side output pins have active-low output polarity

ALT12C: Bit alternativ de pini I2C

1 = I2C mapat la pini SDA1/SCL1

0 = I2C mapat la pini ASDA1/ASCL1

BOREN: Brown-out Reset Enable bit

1 = BOR este activat

0 = BOR este dezactivat

FPWRT<2:0>: Biți de selectare a valorii temporizatorului de resetare la pornire

111 = PWRT = 128 ms

110 = PWRT = 64 ms

101 = PWRT = 32 ms

100 = PWRT = 16 ms

011 = PWRT = 8 ms

010 = PWRT = 4 ms

001 = PWRT = 2 ms

000 = PWRT = Dezactivat

g. Registrul FICD

JTAGEN: Bit de activare JTAG

1 = JTAG este activat

0 = JTAG este dezactivat

ICS<1:0>: Biți de activare pentru selectarea canalului de comunicare ICD

11 = Comunicație pe PGC1/EMUC1 și PGD1/EMUD1

10 = Comunicație pe PGC2/EMUC2 și PGD2/EMUD2

01 = Comunicație pe PGC3/EM