Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

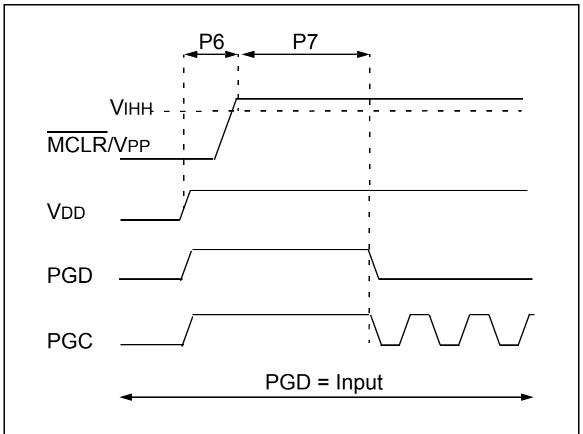
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	dsPIC
Core Size	16-Bit
Speed	30 MIPs
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, I²S, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	144KB (48K x 24)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6012at-30i-pf

5.2 Entering Enhanced ICSP Mode

The Enhanced ICSP mode is entered by holding PGC and PGD high, and then raising $\overline{\text{MCLR}}$/VPP to VIHH (high voltage), as illustrated in Figure 5-2. In this mode, the code memory, data EEPROM and Configuration bits can be efficiently programmed using the programming executive commands that are serially transferred using PGC and PGD.

FIGURE 5-2: ENTERING ENHANCED ICSP™ MODE



Note 1: The sequence that places the device into Enhanced ICSP mode places all unused I/Os in the high-impedance state.

2: Before entering Enhanced ICSP mode, clock switching must be disabled using ICSP, by programming the FCKSM<1:0> bits in the FOSC Configuration register to '11' or '10'.

3: When in Enhanced ICSP mode, the SPI output pin (SDO1) will toggle while the device is being programmed.

5.3 Chip Erase

Before a chip can be programmed, it must be erased. The Bulk Erase command (`ERASEB`) is used to perform this task. Executing this command with the MS command field set to 0x3 erases all code memory, data EEPROM and code-protect Configuration bits. The Chip Erase process sets all bits in these three memory regions to '1'.

Since non-code-protect Configuration bits cannot be erased, they must be manually set to '1' using multiple `PROGC` commands. One `PROGC` command must be sent for each Configuration register (see **Section 5.7 "Configuration Bits Programming"**).

If Advanced Security features are enabled, then individual Segment Erase operations would need to be performed, depending on which segment needs to be programmed at a given stage of system programming. The user should have the flexibility to select specific segments for programming.

Note: The Device ID registers cannot be erased. These registers remain intact after a Chip Erase is performed.

5.4 Blank Check

The term "Blank Check" means to verify that the device has been successfully erased and has no programmed memory cells. A blank or erased memory cell reads as '1'. The following memories must be blank checked:

- All implemented code memory
- All implemented data EEPROM
- All Configuration bits (for their default value)

The Device ID registers (0xFF0000:0xFF0002) can be ignored by the Blank Check since this region stores device information that cannot be erased. Additionally, all unimplemented memory space should be ignored from the Blank Check.

The `QBLANK` command is used for the Blank Check. It determines if the code memory and data EEPROM are erased by testing these memory regions. A 'BLANK' or 'NOT BLANK' response is returned. The `READD` command is used to read the Configuration registers. If it is determined that the device is not blank, it must be erased (see **Section 5.3 "Chip Erase"**) before attempting to program the chip.

5.5.3 PROGRAMMING VERIFICATION

Once code memory is programmed, the contents of memory can be verified to ensure that programming was successful. Verification requires code memory to be read back and compared against the copy held in the programmer's buffer.

The `READP` command can be used to read back all the programmed code memory.

Alternatively, you can have the programmer perform the verification once the entire device is programmed using a checksum computation, as described in **Section 6.8 "Checksum Computation"**.

5.6 Data EEPROM Programming

5.6.1 OVERVIEW

The panel architecture for the data EEPROM memory array consists of 128 rows of sixteen 16-bit data words. Each panel stores 2K words. All devices have either one or no memory panels. Devices with data EEPROM provide either 512 words, 1024 words or 2048 words of memory on the one panel (see Table 5-3).

TABLE 5-3: DATA EEPROM SIZE

Device	Data EEPROM Size (Words)	Number of Rows
dsPIC30F2010	512	32
dsPIC30F2011	0	0
dsPIC30F2012	0	0
dsPIC30F3010	512	32
dsPIC30F3011	512	32
dsPIC30F3012	512	32
dsPIC30F3013	512	32
dsPIC30F3014	512	32
dsPIC30F4011	512	32
dsPIC30F4012	512	32
dsPIC30F4013	512	32
dsPIC30F5011	512	32
dsPIC30F5013	512	32
dsPIC30F5015	512	32
dsPIC30F5016	512	32
dsPIC30F6010	2048	128
dsPIC30F6010A	2048	128
dsPIC30F6011	1024	64
dsPIC30F6011A	1024	64
dsPIC30F6012	2048	128
dsPIC30F6012A	2048	128
dsPIC30F6013	1024	64
dsPIC30F6013A	1024	64
dsPIC30F6014	2048	128
dsPIC30F6014A	2048	128
dsPIC30F6015	2048	128

5.6.2 PROGRAMMING METHODOLOGY

The programming executive uses the `PROGD` command to program the data EEPROM. Figure 5-4 illustrates the flowchart of the process. Firstly, the number of rows to program (RemainingRows) is based on the device size, and the destination address (DestAddress) is set to '0'. In this example, 128 rows (2048 words) of data EEPROM will be programmed.

The first `PROGD` command programs the first row of data EEPROM. Once the command completes successfully, 'RemainingRows' is decremented by 1 and compared with 0. Since there are 127 more rows to program, 'BaseAddress' is incremented by 0x20 to point to the next row of data EEPROM. This process is then repeated until all 128 rows of data EEPROM are programmed.

FIGURE 5-4: FLOWCHART FOR PROGRAMMING dsPIC30F6014A DATA EEPROM

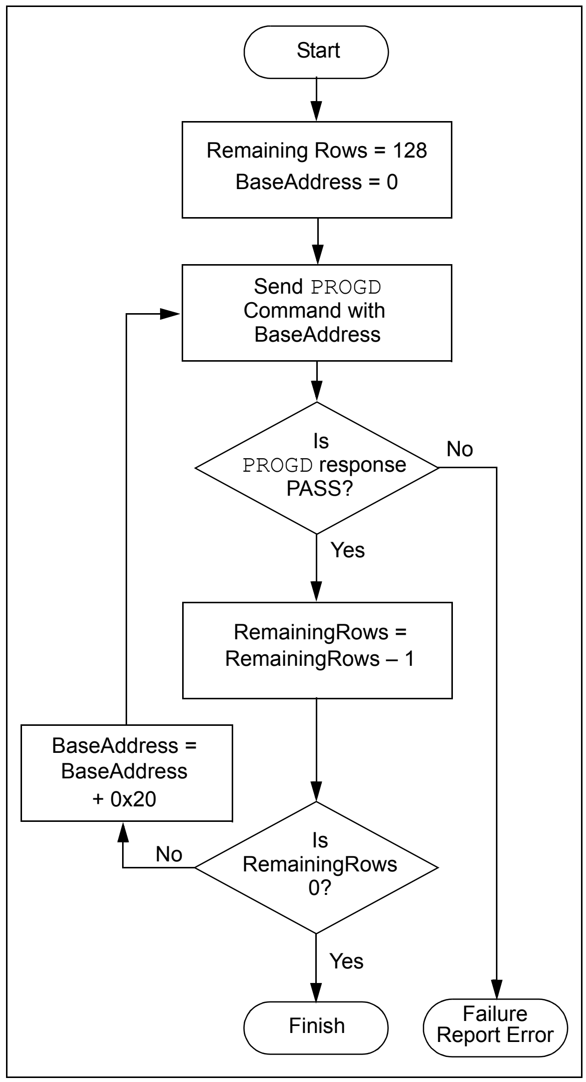


TABLE 5-6: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013/3014, dsPIC30F4013, dsPIC30F5015/5016, dsPIC30F6010A/6011A/6012A/6013A/6014A AND dsPIC30F6015

Bit Field	Register	Description
FCKSM<1:0>	FOSC	**Clock Switching Mode** `1x` = Clock switching is disabled, Fail-Safe Clock Monitor is disabled `01` = Clock switching is enabled, Fail-Safe Clock Monitor is disabled `00` = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
FOS<2:0>	FOSC	**Oscillator Source Selection on POR** `111` = Primary Oscillator `110` = Reserved `101` = Reserved `100` = Reserved `011` = Reserved `010` = Internal Low-Power RC Oscillator `001` = Internal Fast RC Oscillator (no PLL) `000` = Low-Power 32 kHz Oscillator (Timer1 Oscillator)
FPR<4:0>	FOSC	**Primary Oscillator Mode (when FOS<2:0> = `111b`)** `11xxx` = Reserved (do not use) `10111` = HS/3 w/PLL 16X – HS/3 crystal oscillator with 16X PLL (10 MHz-25 MHz crystal) `10110` = HS/3 w/PLL 8X – HS/3 crystal oscillator with 8X PLL (10 MHz-25 MHz crystal) `10101` = HS/3 w/PLL 4X – HS/3 crystal oscillator with 4X PLL (10 MHz-25 MHz crystal) `10100` = Reserved (do not use) `10011` = HS/2 w/PLL 16X – HS/2 crystal oscillator with 16X PLL (10 MHz-25 MHz crystal) `10010` = HS/2 w/PLL 8X – HS/2 crystal oscillator with 8X PLL (10 MHz-25 MHz crystal `10001` = HS/2 w/PLL 4X – HS/2 crystal oscillator with 4X PLL (10 MHz-25 MHz crystal) `10000` = Reserved (do not use) `01111` = ECIO w/PLL 16x – External clock with 16x PLL. OSC2 pin is I/O `01110` = ECIO w/PLL 8x – External clock with 8x PLL. OSC2 pin is I/O `01101` = ECIO w/PLL 4x – External clock with 4x PLL. OSC2 pin is I/O `01100` = Reserved (do not use) `01011` = Reserved (do not use) `01010` = FRC w/PLL 8x – Internal fast RC oscillator with 8x PLL. OSC2 pin is I/O `01001` = Reserved (do not use) `01000` = Reserved (do not use) `00111` = XT w/PLL 16X – XT crystal oscillator with 16X PLL `00110` = XT w/PLL 8X – XT crystal oscillator with 8X PLL `00101` = XT w/PLL 4X – XT crystal oscillator with 4X PLL `00100` = Reserved (do not use) `00011` = FRC w/PLL 16x – Internal fast RC oscillator with 8x PLL. OSC2 pin is I/O `00010` = Reserved (do not use) `00001` = FRC w/PLL 4x – Internal fast RC oscillator with 4x PLL. OSC2 pin is I/O `00000` = Reserved (do not use)

TABLE 5-7: CONFIGURATION BITS DESCRIPTION

Bit Field	Register	Description
FWPSA<1:0>	FWDT	**Watchdog Timer Prescaler A** 11 = 1:512 10 = 1:64 01 = 1:8 00 = 1:1
FWPSB<3:0>	FWDT	**Watchdog Timer Prescaler B** 1111 = 1:16 1110 = 1:15 • • • 0001 = 1:2 0000 = 1:1
FWDTEN	FWDT	**Watchdog Enable** 1 = Watchdog enabled (LPRC oscillator cannot be disabled. Clearing the SWDTEN bit in the RCON register will have no effect) 0 = Watchdog disabled (LPRC oscillator can be disabled by clearing the SWDTEN bit in the RCON register)
MCLREN	FBORPOR	**Master Clear Enable** 1 = Master Clear pin ($\overline{MCLR}$) is enabled 0 = $\overline{MCLR}$ pin is disabled
PWMPIN	FBORPOR	**Motor Control PWM Module Pin Mode** 1 = PWM module pins controlled by PORT register at device Reset (tri-stated) 0 = PWM module pins controlled by PWM module at device Reset (configured as output pins)
HPOL	FBORPOR	**Motor Control PWM Module High-Side Polarity** 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	FBORPOR	**Motor Control PWM Module Low-Side Polarity** 1 = PWM module low-side output pins have active-high output polarity 0 = PWM module low-side output pins have active-low output polarity
BOREN	FBORPOR	**PBOR Enable** 1 = PBOR enabled 0 = PBOR disabled
BORV<1:0>	FBORPOR	**Brown-out Voltage Select** 11 = 2.0V (not a valid operating selection) 10 = 2.7V 01 = 4.2V 00 = 4.5V
FPWRT<1:0>	FBORPOR	**Power-on Reset Timer Value Select** 11 = PWRT = 64 ms 10 = PWRT = 16 ms 01 = PWRT = 4 ms 00 = Power-up Timer disabled
RBS<1:0>	FBS	**Boot Segment Data RAM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 11 = No Data RAM is reserved for Boot Segment 10 = Small-sized Boot RAM [128 bytes of RAM are reserved for Boot Segment] 01 = Medium-sized Boot RAM [256 bytes of RAM are reserved for Boot Segment] 00 = Large-sized Boot RAM [512 bytes of RAM are reserved for Boot Segment in dsPIC30F5011/5013, and 1024 bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015]

TABLE 5-7: CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Register	Description
EBS	FBS	**Boot Segment Data EEPROM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 1 = No Data EEPROM is reserved for Boot Segment 0 = 128 bytes of Data EEPROM are reserved for Boot Segment in dsPIC30F5011/5013, and 256 bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015
BSS<2:0>	FBS	**Boot Segment Program Memory Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 111 = No Boot Segment 110 = Standard security; Small-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x0003FF] 101 = Standard security; Medium-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x000FFF] 100 = Standard security; Large-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x001FF] 011 = No Boot Segment 010 = High security; Small-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x0003FF] 001 = High security; Medium-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x000FFF] 000 = High security; Large-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x001FFF]
BWRP	FBS	**Boot Segment Program Memory Write Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 1 = Boot Segment program memory is not write-protected 0 = Boot Segment program memory is write-protected
RSS<1:0>	FSS	**Secure Segment Data RAM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 11 = No Data RAM is reserved for Secure Segment 10 = Small-sized Secure RAM [(256 – N) bytes of RAM are reserved for Secure Segment] 01 = Medium-sized Secure RAM [(768 – N) bytes of RAM are reserved for Secure Segment in dsPIC30F5011/5013, and (2048 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] 00 = Large-sized Secure RAM [(1024 – N) bytes of RAM are reserved for Secure Segment in dsPIC30F5011/5013, and (4096 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] where N = Number of bytes of RAM reserved for Boot Sector.
ESS<1:0>	FSS	**Secure Segment Data EEPROM Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)** 11 = No Data EEPROM is reserved for Secure Segment 10 = Small-sized Secure Data EEPROM [(128 – N) bytes of Data EEPROM are reserved for Secure Segment in dsPIC30F5011/5013, and (256 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] 01 = Medium-sized Secure Data EEPROM [(256 – N) bytes of Data EEPROM are reserved for Secure Segment in dsPIC30F5011/5013, and (512 – N) bytes in dsPIC30F6010A/6011A/6012A/6013A/6014A/6015] 00 = Large-sized Secure Data EEPROM [(512 – N) bytes of Data EEPROM are reserved for Secure Segment in dsPIC30F5011/5013, (1024 – N) bytes in dsPIC30F6011A/6013A, and (2048 – N) bytes in dsPIC30F6010A/6012A/6014A/6015] where N = Number of bytes of Data EEPROM reserved for Boot Sector.

8.0 PROGRAMMING EXECUTIVE COMMANDS

8.1 Command Set

The programming executive command set is shown in Table 8-1. This table contains the opcode, mnemonic, length, time out and description for each command. Functional details on each command are provided in the command descriptions (see **Section 8.5 "Command Descriptions"**).

8.2 Command Format

All programming executive commands have a general format consisting of a 16-bit header and any required data for the command (see Figure 8-1). The 16-bit header consists of a 4-bit opcode field, which is used to identify the command, followed by a 12-bit command length field.

FIGURE 8-1: COMMAND FORMAT

15 ... 12	11 ... 0
Opcode	Length
Command Data First Word (if required)	
•	
•	
Command Data Last Word (if required)	

The command opcode must match one of those in the command set. Any command that is received which does not match the list in Table 8-1 will return a "NACK" response (see **Section 9.2.1 "`Opcode` Field"**).

The command length is represented in 16-bit words since the SPI operates in 16-bit mode. The programming executive uses the Command Length field to determine the number of words to read from the SPI port. If the value of this field is incorrect, the command will not be properly received by the programming executive.

8.3 Packed Data Format

When 24-bit instruction words are transferred across the 16-bit SPI interface, they are packed to conserve space using the format shown in Figure 8-2. This format minimizes traffic over the SPI and provides the programming executive with data that is properly aligned for performing table write operations.

FIGURE 8-2: PACKED INSTRUCTION WORD FORMAT

15 ... 8	7 ... 0
lsw1	
MSB2	MSB1
lsw2	

lswx: Least significant 16 bits of instruction word
MSBx: Most Significant Byte of instruction word

> **Note:** When the number of instruction words transferred is odd, MSB2 is zero and lsw2 cannot be transmitted.

8.4 Programming Executive Error Handling

The programming executive will "NACK" all unsupported commands. Additionally, due to the memory constraints of the programming executive, no checking is performed on the data contained in the Programmer command. It is the responsibility of the programmer to command the programming executive with valid command arguments, or the programming operation may fail. Additional information on error handling is provided in **Section 9.2.3 "QE_Code Field"**.

8.5.5 PROGP COMMAND

15 … 12	11 … 8	7 … 0
Opcode	Length	
Reserved		Addr_MSB
Addr_LS		
D_1		
D_2		
...		
D_N		

Field	Description
Opcode	0x5
Length	0x33
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	LS 16 bits of 24-bit destination address
D_1	16-bit data word 1
D_2	16-bit data word 2
...	16-bit data word 3 through 47
D_48	16-bit data word 48

The PROGP command instructs the programming executive to program one row of code memory (32 instruction words) to the specified memory address. Programming begins with the row address specified in the command. The destination address should be a multiple of 0x40.

The data to program to memory, located in command words D_1 through D_48, must be arranged using the packed instruction word format shown in Figure 8-2.

After all data has been programmed to code memory, the programming executive verifies the programmed data against the data in the command.

Expected Response (2 words):
0x1500
0x0002

Note: Refer to Table 5-2 for code memory size information.

8.5.6 PROGC COMMAND

15 … 12	11 … 8	7 … 0
Opcode	Length	
Reserved		Addr_MSB
Addr_LS		
Data		

Field	Description
Opcode	0x6
Length	0x4
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	LS 16 bits of 24-bit destination address
Data	Data to program

The PROGC command programs data to the specified Configuration register and verifies the programming. Configuration registers are 16 bits wide, and this command allows one Configuration register to be programmed.

Expected Response (2 words):
0x1600
0x0002

Note: This command can only be used for programming Configuration registers.

8.5.7 ERASEB COMMAND

15 12	11 2	0
Opcode	Length	
Reserved		MS

Field	Description
Opcode	0x7
Length	0x2
Reserved	0x0
MS	Select memory to erase: 0x0 = All Code in General Segment 0x1 = All Data EEPROM in General Segment 0x2 = All Code and Data EEPROM in General Segment, interrupt vectors and FGS Configuration register 0x3 = Full Chip Erase 0x4 = All Code and Data EEPROM in Boot, Secure and General Segments, and FBS, FSS and FGS Configuration registers 0x5 = All Code and Data EEPROM in Secure and General Segments, and FSS and FGS Configuration registers 0x6 = All Data EEPROM in Boot Segment 0x7 = All Data EEPROM in Secure Segment

The ERASEB command performs a Bulk Erase. The MS field selects the memory to be bulk erased, with options for erasing Code and/or Data EEPROM in individual memory segments.

When Full Chip Erase is selected, the following memory regions are erased:

- All code memory (even if code-protected)
- All data EEPROM
- All code-protect Configuration registers

Only the executive code memory, Unit ID, device ID and Configuration registers that are not code-protected remain intact after a Chip Erase.

Expected Response (2 words):
0x1700
0x0002

Note: A Full Chip Erase cannot be performed in low-voltage programming systems (VDD less than 4.5 volts). ERASED and ERASEP must be used to erase code memory, executive memory and data memory. Alternatively, individual Segment Erase operations may be performed.

8.5.8 ERASED COMMAND

15 12	11 8	7 0
Opcode	Length	
Num_Rows		Addr_MSB
Addr_LS		

Field	Description
Opcode	0x8
Length	0x3
Num_Rows	Number of rows to erase (max of 128)
Addr_MSB	MSB of 24-bit base address
Addr_LS	LS 16 bits of 24-bit base address

The ERASED command erases the specified number of rows of data EEPROM from the specified base address. The specified base address must be a multiple of 0x20. Since the data EEPROM is mapped to program space, a 24-bit base address must be specified.

After the erase is performed, all targeted bytes of data EEPROM will contain 0xFF.

Expected Response (2 words):
0x1800
0x0002

Note: The ERASED command cannot be used to erase the Configuration registers or device ID. Code-protect Configuration registers can only be erased with the ERASEB command, while the device ID is read-only.

10.0 DEVICE ID

The device ID region is 2 x 16 bits and can be read using the READD command. This region of memory is read-only and can also be read when code protection is enabled.

Table 10-1 shows the device ID for each device, Table 10-2 shows the device ID registers and Table 10-3 describes the bit field of each register.

TABLE 10-1: DEVICE IDS

Device	DEVID	Silicon Revision							
		A0	A1	A2	A3	A4	B0	B1	B2
dsPIC30F2010	0x0040	0x1000	0x1001	0x1002	0x1003	0x1004	—	—	—
dsPIC30F2011	0x0240	—	0x1001	—	—	—	—	—	—
dsPIC30F2012	0x0241	—	0x1001	—	—	—	—	—	—
dsPIC30F3010	0x01C0	0x1000	0x1001	0x1002	—	—	—	—	—
dsPIC30F3011	0x01C1	0x1000	0x1001	0x1002	—	—	—	—	—
dsPIC30F3012	0x00C1	—	—	—	—	—	0x1040	0x1041	—
dsPIC30F3013	0x00C3	—	—	—	—	—	0x1040	0x1041	—
dsPIC30F3014	0x0160	—	0x1001	0x1002	—	—	—	—	—
dsPIC30F4011	0x0101	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F4012	0x0100	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F4013	0x0141	—	0x1001	0x1002	—	—	—	—	—
dsPIC30F5011	0x0080	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F5013	0x0081	—	0x1001	0x1002	0x1003	0x1003	—	—	—
dsPIC30F5015	0x0200	0x1000	—	—	—	—	—	—	—
dsPIC30F5016	0x0201	0x1000	—	—	—	—	—	—	—
dsPIC30F6010	0x0188	—	—	—	—	—	—	0x1040	0x1042
dsPIC30F6010A	0x0281	—	—	0x1002	0x1003	0x1004	—	—	—
dsPIC30F6011	0x0192	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6011A	0x02C0	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6012	0x0193	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6012A	0x02C2	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6013	0x0197	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6013A	0x02C1	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6014	0x0198	—	—	—	0x1003	—	—	0x1040	0x1042
dsPIC30F6014A	0x02C3	—	—	0x1002	—	—	0x1040	0x1041	—
dsPIC30F6015	0x0280	—	—	0x1002	0x1003	0x1004	—	—	—

TABLE 10-2: dsPIC30F DEVICE ID REGISTERS

<table>
<tr><th rowspan="2">Address</th><th rowspan="2">Name</th><th colspan="16">Bit</th></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0xFF0000</td><td>DEVID</td><td colspan="16">DEVID<15:0></td></tr>
<tr><td>0xFF0002</td><td>DEVREV</td><td colspan="4">PROC<3:0></td><td colspan="6">REV<5:0></td><td colspan="6">DOT<5:0></td></tr>
</table>

11.3 Entering ICSP Mode

The ICSP mode is entered by holding PGC and PGD low, raising $\overline{MCLR}/V_{PP}$ to V_{IHH} (high voltage), and then performing additional steps as illustrated in Figure 11-4.

> **Note 1:** The sequence that places the device into ICSP mode places all unused I/O pins to the high-impedance state.
>
> **2:** Once ICSP mode is entered, the PC is set to 0x0 (the Reset vector).
>
> **3:** Before leaving the Reset vector, execute two `GOTO` instructions, followed by a single `NOP` instruction must be executed.

FIGURE 11-4: ENTERING ICSP™ MODE

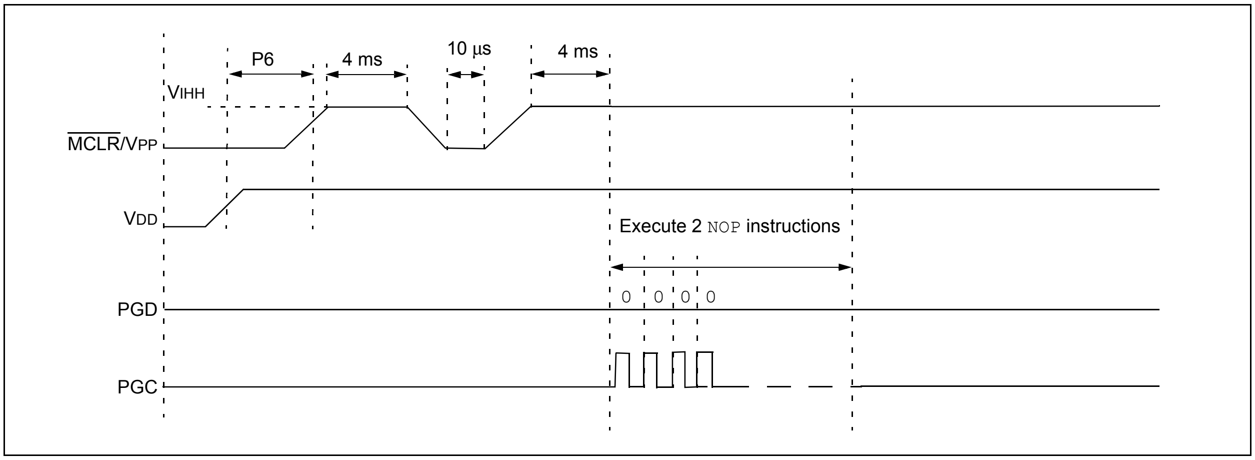


TABLE 11-4: SERIAL INSTRUCTION EXECUTION FOR BULK ERASING PROGRAM MEMORY (ONLY IN NORMAL-VOLTAGE SYSTEMS) (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
0000	200558	`MOV #0x55, W8`
0000	883B38	`MOV W8, NVMKEY`
0000	200AA9	`MOV #0xAA, W9`
0000	883B39	`MOV W9, NVMKEY`
Step 11: Initiate the erase cycle.		
0000	A8E761	`BSET NVMCON, #WR`
0000	000000	`NOP`
0000	000000	`NOP`
—	—	`Externally time 'P13a' ms` (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
0000	000000	`NOP`
0000	000000	`NOP`
0000	A9E761	`BCLR NVMCON, #WR`
0000	000000	`NOP`
0000	000000	`NOP`

Note 1: Steps 2-8 are only required for the dsPIC30F5011/5013 devices. These steps may be skipped for all other devices in the dsPIC30F family.

TABLE 11-7: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION REGISTERS (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 6: Write the Configuration register data to the write latch and increment the write pointer.		

0000
0000
0000
	BB1B96
	000000
	000000
		`TBLWTL W6, [W7++]`
		`NOP`
		`NOP`

Step 7: Unlock the NVMCON for programming.		

0000
0000
0000
0000
	200558
	883B38
	200AA9
	883B39
		`MOV #0x55, W8`
		`MOV W8, NVMKEY`
		`MOV #0xAA, W9`
		`MOV W9, NVMKEY`

Step 8: Initiate the write cycle.		

0000
0000
0000
—
 	
0000
0000
0000
0000
0000
	A8E761
	000000
	000000
	—
	
	000000
	000000
	A9E761
	000000
	000000
		`BSET NVMCON, #WR`
		`NOP`
		`NOP`
		`Externally time 'P12a' ms` (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
		`NOP`
		`NOP`
		`BCLR NVMCON, #WR`
		`NOP`
		`NOP`

Step 9: Reset device internal PC.		
0000 0000	040100 000000	`GOTO 0x100` `NOP`
Step 10: Repeat steps 3-9 until all 7 Configuration registers are cleared.		

11.9 Writing Data EEPROM

The procedure for writing data EEPROM is very similar to the procedure for writing code memory, except that fewer words are programmed in each operation. When writing data EEPROM, one row of data EEPROM is programmed at a time. Each row consists of sixteen 16-bit data words. Since fewer words are programmed during each operation, only working registers W0:W3 are used as temporary holding registers for the data to be programmed.

Table 11-9 shows the ICSP programming details for writing data EEPROM. Note that a different NVMCON value is required to write to data EEPROM, and that the TBLPAG register is hard-coded to 0x7F (the upper byte address of all locations of data EEPROM).

TABLE 11-9: SERIAL INSTRUCTION EXECUTION FOR WRITING DATA EEPROM

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000 0000 0000	040100 040100 000000	GOTO 0x100 GOTO 0x100 NOP
Step 2: Set the NVMCON to write 16 data words.		
0000 0000	24005A 883B0A	MOV #0x4005, W10 MOV W10, NVMCON
Step 3: Initialize the write pointer (W7) for TBLWT instruction.		

0000
0000
0000
	2007F0
	880190
	2xxxx7
		MOV #0x7F, W0
		MOV W0, TBLPAG
		MOV #<DestinationAddress15:0>, W7

Step 4: Load W0:W3 with the next 4 data words to program.		

0000
0000
0000
0000
	2xxxx0
	2xxxx1
	2xxxx2
	2xxxx3
		MOV #<WORD0>, W0
		MOV #<WORD1>, W1
		MOV #<WORD2>, W2
		MOV #<WORD3>, W3

Step 5: Set the read pointer (W6) and load the (next set of) write latches.		
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	EB0300 000000 BB1BB6 000000 000000 BB1BB6 000000 000000 BB1BB6 000000 000000 BB1BB6 000000 000000	CLR W6 NOP TBLWTL [W6++], [W7++] NOP NOP TBLWTL [W6++], [W7++] NOP NOP TBLWTL [W6++], [W7++] NOP NOP TBLWTL [W6++], [W7++] NOP NOP
Step 6: Repeat steps 4-5 four times to load the write latches for 16 data words.		

TABLE 11-9: SERIAL INSTRUCTION EXECUTION FOR WRITING DATA EEPROM (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 7: Unlock the NVMCON for writing.		

0000
0000
0000
0000
	200558
	883B38
	200AA9
	883B39
		`MOV #0x55, W8`
		`MOV W8, NVMKEY`
		`MOV #0xAA, W9`
		`MOV W9, NVMKEY`

Step 8: Initiate the write cycle.		

0000
0000
0000
—
 	
0000
0000
0000
0000
0000
	A8E761
	000000
	000000
	—
	
	000000
	000000
	A9E761
	000000
	000000
		`BSET NVMCON, #WR`
		`NOP`
		`NOP`
		`Externally time 'P12a' ms` (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
		`NOP`
		`NOP`
		`BCLR NVMCON, #WR`
		`NOP`
		`NOP`

Step 9: Reset device internal PC.		
0000 0000	040100 000000	`GOTO 0x100` `NOP`
Step 10: Repeat steps 2-9 until all data memory is programmed.		

11.13 Reading the Application ID Word

The application ID word is stored at address 0x8005BE in executive code memory. To read this memory location, you must use the SIX control code to move this program memory location to the VISI register. The REGOUT control code must then be used to clock the contents of the VISI register out of the device. The corresponding control and instruction codes that must be serially transmitted to the device to perform this operation are shown in Table 11-13.

Once the programmer has clocked-out the application ID word, it must be inspected. If the application ID has the value 0xBB, the programming executive is resident in memory and the device can be programmed using the mechanism described in **Section 5.0 "Device Programming"**. However, if the application ID has any other value, the programming executive is not resident in memory. It must be loaded to memory before the device can be programmed. The procedure for loading the programming executive to the memory is described in **Section 12.0 "Programming the Programming Executive to Memory"**.

11.14 Exiting ICSP Mode

After confirming that the programming executive is resident in memory, or loading the programming executive, ICSP mode is exited by removing power to the device or bringing $\overline{MCLR}$ to V_{IL}. Programming can then take place by following the procedure outlined in **Section 5.0 "Device Programming"**.

TABLE 11-13: SERIAL INSTRUCTION EXECUTION FOR READING THE APPLICATION ID WORD

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit the Reset vector.		
0000	040100	`GOTO 0x100`
0000	040100	`GOTO 0x100`
0000	000000	`NOP`
Step 2: Initialize TBLPAG and the read pointer (W0) for `TBLRD` instruction.		
0000	200800	`MOV #0x80, W0`
0000	880190	`MOV W0, TBLPAG`
0000	205BE0	`MOV #0x5BE, W0`
0000	207841	`MOV VISI, W1`
0000	000000	`NOP`
0000	BA0890	`TBLRDL [W0], [W1]`
0000	000000	`NOP`
0000	000000	`NOP`
Step 3: Output the VISI register using the REGOUT command.		
0001	<VISI>	`Clock out contents of the VISI register`
0000	000000	`NOP`

TABLE 12-1: PROGRAMMING THE PROGRAMMING EXECUTIVE (CONTINUED)

Command (Binary)	Data (Hexadecimal)	Description
Step 8: Set the read pointer (W6) and load the (next four write) latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BBEBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BBEBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
Step 9: Repeat Steps 7-8 eight times to load the write latches for the 32 instructions.		
Step 10: Unlock the NVMCON for programming.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
Step 11: Initiate the programming cycle.		
0000	A8E761	BSET NVMCON, #15
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see **Section 13.0 "AC/DC Characteristics and Timing Requirements"**)
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #15
0000	000000	NOP
0000	000000	NOP
Step 12: Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
Step 13: Repeat Steps 7-12 until all 23 rows of executive memory are programmed.		

TABLE 13-1: AC/DC CHARACTERISTICS (CONTINUED)

AC/DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating Temperature: 25° C is recommended			
Param. No.	**Sym**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
P9b	TDLY5	Delay between PGD ↓by programming executive to PGD released by programming executive	15	—	µs	—
P10	TDLY6	Delay between PGD released by programming executive to first PGC ↑ of response	5	—	µs	—
P11	TDLY7	Delay between clocking out response words	10	—	µs	—
P12a	TPROG	Row Programming cycle time	1	4	ms	ICSP mode
P12b	TPROG	Row Programming cycle time	0.8	2.6	ms	Enhanced ICSP mode
P13a	TERA	Bulk/Row Erase cycle time	1	4	ms	ICSP mode
P13b	TERA	Bulk/Row Erase cycle time	0.8	2.6	ms	Enhanced ICSP mode

APPENDIX C: REVISION HISTORY

Note:	Revision histories were not recorded for revisions A through H. The previous revision (J), was published in August 2007.

Revision K (November 2010)

This version of the document includes the following updates:

- Added Note three to **Section 5.2 "Entering Enhanced ICSP Mode"**
- Updated the first paragraph of **Section 10.0 "Device ID"**
- Updated Table 10-1: Device IDs
- Removed the VARIANT bit and updated the bit definition for the DEVID register in Table 10-2: dsPIC30F Device ID Registers
- Removed the VARIANT bit and updated the bit field definition and description for the DEVID register in Table 10-3: Device ID Bits Description
- Updated Note 3 in **Section 11.3 "Entering ICSP Mode"**
- Updated Step 11 in Table 11-4: Serial Instruction Execution for BUlk Erasing Program Memory (Only in Normal-voltage Systems)
- Updated Steps 5, 12 and 19 in Table 11-5: Serial Instruction Execution for Erasing Program Memory (Either in Low-voltage or Normal-voltage Systems)
- Updated Steps 5, 6 and 8 in Table 11-7: Serial Instruction Execution for Writing Configuration Registers
- Updated Steps 6 and 8 in Table 11-8: Serial Instruction Execution for Writing Code Memory
- Updated Steps 6 and 8 in Table 11-9: Serial Instruction Execution for Writing Data EEPROM
- Updated Entering ICSP™ Mode (see Figure 11-4)
- Updated Steps 4 and 11 in Table 12-1: Programming the Programming Executive
- Renamed parameters: P12 to P12a and P13 to P13a, and added parameters P12b and P13b in Table 13-1: AC/DC Characteristics

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Trademarks

Printed on recycled paper.

ISBN: 978-1-60932-636-4

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.