



Welcome to [E-XFL.COM](https://www.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, I <sup>2</sup> C, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, I <sup>2</sup> S, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	66KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K 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 (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/dspic30f5011t-30i-ptg">https://www.e-xfl.com/product-detail/microchip-technology/dspic30f5011t-30i-ptg</a>

# dsPIC30F Flash Programming Specification

## 5.6.3 PROGRAMMING VERIFICATION

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

Alternatively, the programmer can perform the verification once the entire device is programmed using a checksum computation, as described in [Section 6.8 "Checksum Computation"](#).

**Note:** `TBLRDL` instructions executed within a `REPEAT` loop must not be used to read from Data EEPROM. Instead, it is recommended to use PSV access.

## 5.7 Configuration Bits Programming

### 5.7.1 OVERVIEW

The dsPIC30F has Configuration bits stored in seven 16-bit registers. These bits can be set or cleared to select various device configurations. There are two types of Configuration bits: system-operation bits and code-protect bits. The system-operation bits determine the power-on settings for system-level components such as the oscillator and Watchdog Timer. The code-protect bits prevent program memory from being read and written.

The FOSC Configuration register has three different register descriptions, based on the device. The FOSC Configuration register description for the dsPIC30F2010 and dsPIC30F6010/6011/6012/6013/6014 devices are shown in [Table 5-4](#).

**Note:** If user software performs an erase operation on the configuration fuse, it must be followed by a write operation to this fuse with the desired value, even if the desired value is the same as the state of the erased fuse.

The FOSC Configuration register description for the dsPIC30F4011/4012 and dsPIC30F5011/5013 devices is shown in [Table 5-5](#).

The FOSC Configuration register description for all remaining devices (dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013, dsPIC30F3014/4013, dsPIC30F5015 and dsPIC30F6011A/6012A/6013A/6014A) is shown in [Table 5-6](#). Always use the correct register descriptions for your target processor.

The `FWDT`, `FBORPOR`, `FBS`, `FSS`, `FGS` and `FICD` Configuration registers are not device-dependent. The register descriptions for these Configuration registers are shown in [Table 5-7](#).

The Device Configuration register maps are shown in [Table 5-8](#) through [Table 5-11](#).

**TABLE 5-4: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2010 AND dsPIC30F6010/6011/6012/6013/6014**

Bit Field	Register	Description
FCKSM<1:0>	FOSC	<b>Clock Switching Mode</b> 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<1:0>	FOSC	<b>Oscillator Source Selection on POR</b> 11 = Primary Oscillator 10 = Internal Low-Power RC Oscillator 01 = Internal Fast RC Oscillator 00 = Low-Power 32 kHz Oscillator (Timer1 Oscillator)
FPR<3:0>	FOSC	<b>Primary Oscillator Mode</b> 1111 = ECIO w/PLL 16X – External Clock mode with 16X PLL. OSC2 pin is I/O 1110 = ECIO w/PLL 8X – External Clock mode with 8X PLL. OSC2 pin is I/O 1101 = ECIO w/PLL 4X – External Clock mode with 4X PLL. OSC2 pin is I/O 1100 = ECIO – External Clock mode. OSC2 pin is I/O 1011 = EC – External Clock mode. OSC2 pin is system clock output (Fosc/4) 1010 = Reserved (do not use) 1001 = ERC – External RC Oscillator mode. OSC2 pin is system clock output (Fosc/4) 1000 = ERCIO – External RC Oscillator mode. OSC2 pin is I/O 0111 = XT w/PLL 16X – XT Crystal Oscillator mode with 16X PLL 0110 = XT w/PLL 8X – XT Crystal Oscillator mode with 8X PLL 0101 = XT w/PLL 4X – XT Crystal Oscillator mode with 4X PLL 0100 = XT – XT Crystal Oscillator mode (4 MHz-10 MHz crystal) 001x = HS – HS Crystal Oscillator mode (10 MHz-25 MHz crystal) 000x = XTL – XTL Crystal Oscillator mode (200 kHz-4 MHz crystal)

# dsPIC30F Flash Programming Specification

**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	<b>Clock Switching Mode</b> 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	<b>Oscillator Source Selection on POR</b> 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	<b>Primary Oscillator Mode (when FOS&lt;2:0&gt; = 111b)</b> 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)

# dsPIC30F Flash Programming Specification

**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 (CONTINUED)**

Bit Field	Register	Description
FPR<4:0>	FOSC	<b>Alternate Oscillator Mode (when FOS&lt;2:0&gt; = 011b)</b> 1xxxx = Reserved (do not use) 0111x = Reserved (do not use) 01101 = Reserved (do not use) 01100 = ECIO – External clock. OSC2 pin is I/O 01011 = EC – External clock. OSC2 pin is system clock output (Fosc/4) 01010 = Reserved (do not use) 01001 = ERC – External RC oscillator. OSC2 pin is system clock output (Fosc/4) 01000 = ERCIO – External RC oscillator. OSC2 pin is I/O 00111 = Reserved (do not use) 00110 = Reserved (do not use) 00101 = Reserved (do not use) 00100 = XT – XT crystal oscillator (4 MHz-10 MHz crystal) 00010 = HS – HS crystal oscillator (10 MHz-25 MHz crystal) 00001 = Reserved (do not use) 00000 = XTL – XTL crystal oscillator (200 kHz-4 MHz crystal)

# dsPIC30F Flash Programming Specification

**TABLE 5-7: CONFIGURATION BITS DESCRIPTION (CONTINUED)**

Bit Field	Register	Description
SSS<2:0>	FSS	<b>Secure Segment Program Memory Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)</b> 111 = No Secure Segment 110 = Standard security; Small-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x001FFF] 101 = Standard security; Medium-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x003FFF] 100 = Standard security; Large-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x007FFF] 011 = No Secure Segment 010 = High security; Small-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x001FFF] 001 = High security; Medium-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x003FFF] 000 = High security; Large-sized Secure Program Flash [Secure Segment starts after BS and ends at 0x007FFF]
SWRP	FSS	<b>Secure Segment Program Memory Write Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)</b> 1 = Secure Segment program memory is not write-protected 0 = Secure program memory is write-protected
GSS<1:0>	FGS	<b>General Segment Program Memory Code Protection (only present in dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)</b> 11 = Code protection is disabled 10 = Standard security code protection is enabled 0x = High security code protection is enabled
GCP	FGS	<b>General Segment Program Memory Code Protection (present in all devices except dsPIC30F5011/5013/6010A/6011A/6012A/6013A/6014A/6015)</b> 1 = General Segment program memory is not code-protected 0 = General Segment program memory is code-protected
GWRP	FGS	<b>General Segment Program Memory Write Protection</b> 1 = General Segment program memory is not write-protected 0 = General Segment program memory is write-protected
BKBUG	FICD	<b>Debugger/Emulator Enable</b> 1 = Device will reset into Operational mode 0 = Device will reset into Debug/Emulation mode
COE	FICD	<b>Debugger/Emulator Enable</b> 1 = Device will reset into Operational mode 0 = Device will reset into Clip-on Emulation mode
ICS<1:0>	FICD	<b>ICD Communication Channel Select</b> 11 = Communicate on PGC/EMUC and PGD/EMUD 10 = Communicate on EMUC1 and EMUD1 01 = Communicate on EMUC2 and EMUD2 00 = Communicate on EMUC3 and EMUD3
RESERVED	FBS, FSS, FGS	<b>Reserved (read as '1', write as '1')</b>
—	All	<b>Unimplemented (read as '0', write as '0')</b>

**TABLE 5-8: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F2010, dsPIC30F4011/4012 AND dsPIC30F6010/ 6011/6012/6013/ 6014)**

Address	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FOSC	FCKSM<1:0>		—	—	—	—	FOS<1:0>		—	—	—	—	FPR<3:0>			
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN <sup>(1)</sup>	HPOL <sup>(1)</sup>	LPOL <sup>(1)</sup>	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	Reserved <sup>(2)</sup>		—	—	—	Reserved <sup>(2)</sup>	—	—	—	—	Reserved <sup>(2)</sup>			
0xF80008	FSS	—	—	Reserved <sup>(2)</sup>		—	—	Reserved <sup>(2)</sup>		—	—	—	—	Reserved <sup>(2)</sup>			
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved <sup>(2)</sup>	GCP	GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

**Note 1:** On the 6011, 6012, 6013 and 6014, these bits are reserved (read as '1' and must be programmed as '1').

**Note 2:** Reserved bits read as '1' and must be programmed as '1'.

**TABLE 5-9: dsPIC30F CONFIGURATION REGISTERS (FOR dsPIC30F5011/5013)**

Address	Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FOSC	FCKSM<1:0>		—	—	—	—	FOS<1:0>		—	—	—	—	FPR<3:0>			
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	Reserved <sup>(1)</sup>			BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	RBS<1:0>		—	—	—	EBS	—	—	—	—	BSS<2:0>			BWRP
0xF80008	FSS	—	—	RSS<1:0>		—	—	ESS<1:0>		—	—	—	—	SSS<2:0>			SWRP
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	GSS<1:0>		GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

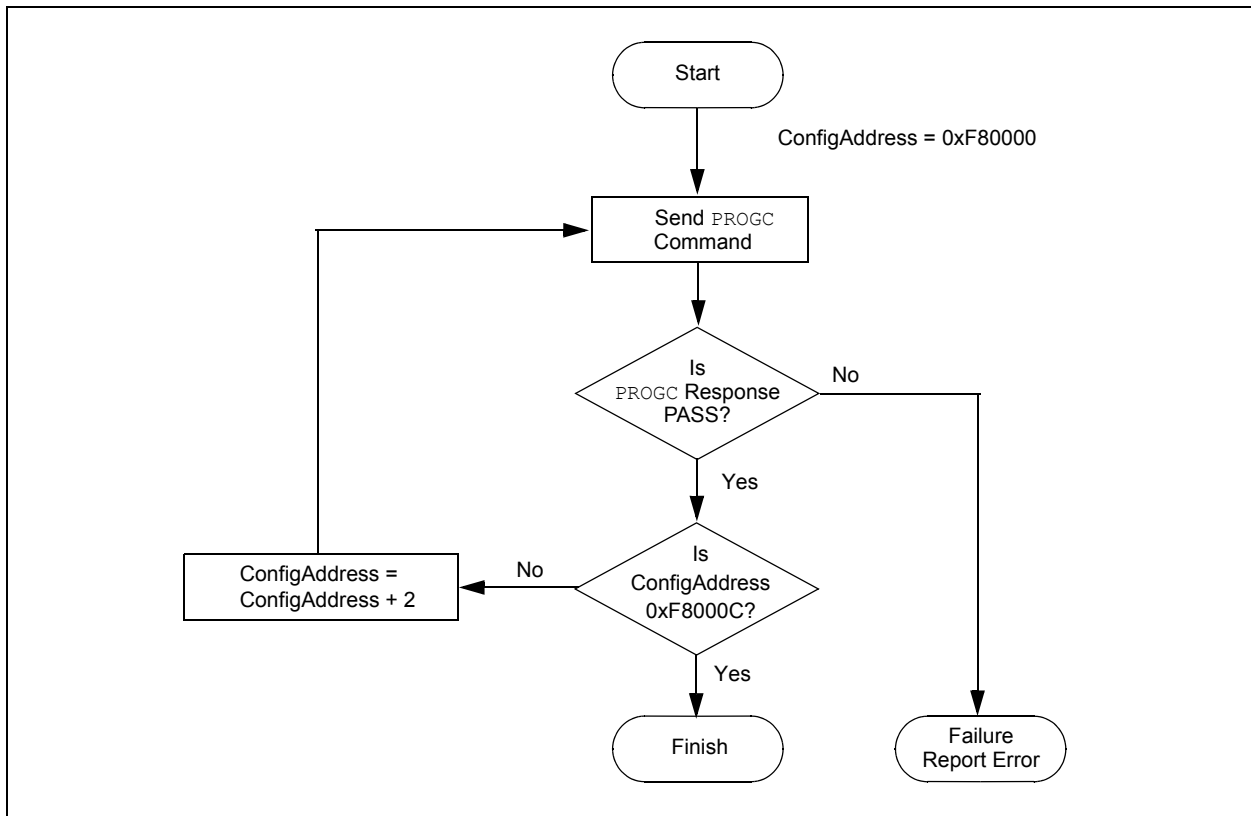
**Note 1:** Reserved bits read as '1' and must be programmed as '1'.

# dsPIC30F Flash Programming Specification

## 5.8 Exiting Enhanced ICSP Mode

The Enhanced ICSP mode is exited by removing power from the device or bringing MCLR to V<sub>IL</sub>. When normal user mode is next entered, the program that was stored using Enhanced ICSP will execute.

**FIGURE 5-5: CONFIGURATION BIT PROGRAMMING FLOW**



# dsPIC30F Flash Programming Specification

## 6.6 Configuration Information in the Hexadecimal File

To allow portability of code, the programmer must read the Configuration register locations from the hexadecimal file. If configuration information is not present in the hexadecimal file, a simple warning message should be issued by the programmer. Similarly, while saving a hexadecimal file, all configuration information must be included. An option to not include the configuration information can be provided.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

## 6.7 Unit ID

The dsPIC30F devices contain 32 instructions of Unit ID. These are located at addresses 0x8005C0 through 0x8005FF. The Unit ID can be used for storing product information such as serial numbers, system manufacturing dates, manufacturing lot numbers and other such application-specific information.

A Bulk Erase does not erase the Unit ID locations. Instead, erase all executive memory using steps 1-4 as shown in [Table 12-1](#), and program the Unit ID along with the programming executive. Alternately, use a Row Erase to erase the row containing the Unit ID locations.

## 6.8 Checksum Computation

Checksums for the dsPIC30F are 16 bits in size. The checksum is to total sum of the following:

- Contents of code memory locations
- Contents of Configuration registers

[Table A-1](#) describes how to calculate the checksum for each device. All memory locations are summed one byte at a time, using only their native data size. More specifically, Configuration and device ID registers are summed by adding the lower two bytes of these locations (the upper byte is ignored), while code memory is summed by adding all three bytes of code memory.

**Note:** The checksum calculation differs depending on the code-protect setting. [Table A-1](#) describes how to compute the checksum for an unprotected device and a read-protected device. Regardless of the code-protect setting, the Configuration registers can always be read.

## 7.0 PROGRAMMER – PROGRAMMING EXECUTIVE COMMUNICATION

### 7.1 Communication Overview

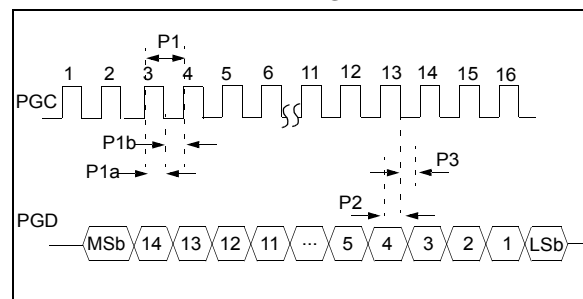
The programmer and programming executive have a master-slave relationship, where the programmer is the master programming device and the programming executive is the slave.

All communication is initiated by the programmer in the form of a command. Only one command at a time can be sent to the programming executive. In turn, the programming executive only sends one response to the programmer after receiving and processing a command. The programming executive command set is described in [Section 8.0 “Programming Executive Commands”](#). The response set is described in [Section 9.0 “Programming Executive Responses”](#).

### 7.2 Communication Interface and Protocol

The Enhanced ICSP interface is a 2-wire SPI interface implemented using the PGC and PGD pins. The PGC pin is used as a clock input pin, and the clock source must be provided by the programmer. The PGD pin is used for sending command data to, and receiving response data from, the programming executive. All serial data is transmitted on the falling edge of PGC and latched on the rising edge of PGD. All data transmissions are sent Most Significant bit (MSb) first, using 16-bit mode (see [Figure 7-1](#)).

**FIGURE 7-1: PROGRAMMING EXECUTIVE SERIAL TIMING**



Since a 2-wire SPI interface is used, and data transmissions are bidirectional, a simple protocol is used to control the direction of PGD. When the programmer completes a command transmission, it releases the PGD line and allows the programming executive to drive this line high. The programming executive keeps the PGD line high to indicate that it is processing the command.

After the programming executive has processed the command, it brings PGD low for 15  $\mu$ sec to indicate to the programmer that the response is available to be



# dsPIC30F Flash Programming Specification

TABLE 8-1: PROGRAMMING EXECUTIVE COMMAND SET

Opcode	Mnemonic	Length (16-bit words)	Time Out	Description
0x0	SCHECK	1	1 ms	Sanity check.
0x1	READD	4	1 ms/row	Read N 16-bit words of data EEPROM, Configuration registers or device ID starting from specified address.
0x2	READP	4	1 ms/row	Read N 24-bit instruction words of code memory starting from specified address.
0x3	Reserved	N/A	N/A	This command is reserved. It will return a NACK.
0x4	PROGD <sup>(2)</sup>	19	5 ms	Program one row of data EEPROM at the specified address, then verify.
0x5	PROGP <sup>(1)</sup>	51	5 ms	Program one row of code memory at the specified address, then verify.
0x6	PROGC	4	5 ms	Write byte or 16-bit word to specified Configuration register.
0x7	ERASEB	2	5 ms	Bulk Erase (entire code memory or data EEPROM), or erase by segment.
0x8	ERASED <sup>(2)</sup>	3	5 ms/row	Erase rows of data EEPROM from specified address.
0x9	ERASEP <sup>(1)</sup>	3	5 ms/row	Erase rows of code memory from specified address.
0xA	QBLANK	3	300 ms	Query if the code memory and data EEPROM are blank.
0xB	QVER	1	1 ms	Query the programming executive software version.

**Note 1:** One row of code memory consists of (32) 24-bit words. Refer to [Table 5-2](#) for device-specific information.

**Note 2:** One row of data EEPROM consists of (16) 16-bit words. Refer to [Table 5-3](#) for device-specific information.

# dsPIC30F Flash Programming Specification

## 8.5.11 QVER COMMAND

15	12	11	0
Opcode	Length		

Field	Description
Opcode	0xB
Length	0x1

The `QVER` command queries the version of the programming executive software stored in test memory. The “version.revision” information is returned in the response’s `QE_Code` using a single byte with the following format: main version in upper nibble and revision in the lower nibble (i.e., 0x23 is version 2.3 of programming executive software).

### Expected Response (2 words):

0x1BMN (where “MN” stands for version M.N)  
0x0002

## 9.0 PROGRAMMING EXECUTIVE RESPONSES

### 9.1 Overview

The programming executive sends a response to the programmer for each command that it receives. The response indicates if the command was processed correctly, and includes any required response or error data.

The programming executive response set is shown in [Table 9-1](#). This table contains the opcode, mnemonic and description for each response. The response format is described in [Section 9.2 “Response Format”](#).

**TABLE 9-1: PROGRAMMING EXECUTIVE RESPONSE SET**

Opcode	Mnemonic	Description
0x1	PASS	Command successfully processed.
0x2	FAIL	Command unsuccessfully processed.
0x3	NACK	Command not known.

## 9.2 Response Format

As shown in [Example 9-1](#), all programming executive responses have a general format consisting of a two word header and any required data for the command. [Table 9-2](#) lists the fields and their descriptions.

**EXAMPLE 9-1: FORMAT**

15	12	11	8	7	0
Opcode	Last_Cmd		QE_Code		
Length					
D_1 (if applicable)					
...					
D_N (if applicable)					

**TABLE 9-2: FIELDS AND DESCRIPTIONS**

Field	Description
Opcode	Response opcode.
Last_Cmd	Programmer command that generated the response.
QE_Code	Query code or Error code.
Length	Response length in 16-bit words (includes 2 header words.)
D_1	First 16-bit data word (if applicable).
D_N	Last 16-bit data word (if applicable).

### 9.2.1 Opcode FIELD

The Opcode is a 4-bit field in the first word of the response. The Opcode indicates how the command was processed (see [Table 9-1](#)). If the command is processed successfully, the response opcode is PASS. If there is an error in processing the command, the response opcode is FAIL, and the `QE_Code` indicates the reason for the failure. If the command sent to the programming executive is not identified, the programming executive returns a NACK response.

### 9.2.2 Last\_Cmd FIELD

The Last\_Cmd is a 4-bit field in the first word of the response and indicates the command that the programming executive processed. Since the programming executive can only process one command at a time, this field is technically not required. However, it can be used to verify whether the programming executive correctly received the command that the programmer transmitted.

# dsPIC30F Flash Programming Specification

## 11.4 Flash Memory Programming in ICSP Mode

Programming in ICSP mode is described in [Section 11.4.1 “Programming Operations”](#) through [Section 11.4.3 “Starting and Stopping a Programming Cycle”](#). Step-by-step procedures are described in [Section 11.5 “Erasing Program Memory in Normal-Voltage Systems”](#) through [Section 11.13 “Reading the Application ID Word”](#). All programming operations must use serial execution, as described in [Section 11.2 “ICSP Operation”](#).

### 11.4.1 PROGRAMMING OPERATIONS

Flash memory write and erase operations are controlled by the NVMCON register. Programming is performed by setting NVMCON to select the type of erase operation ([Table 11-2](#)) or write operation ([Table 11-3](#)), writing a key sequence to enable the programming and initiating the programming by setting the WR control bit, NVMCON<15>.

In ICSP mode, all programming operations are externally timed. An external 2 ms delay must be used between setting the WR control bit and clearing the WR control bit to complete the programming operation.

**TABLE 11-2: NVMCON ERASE OPERATIONS**

NVMCON Value	Erase Operation
0x407F	Erase all code memory, data memory (does not erase UNIT ID).
0x4075	Erase 1 row (16 words) of data EEPROM.
0x4074	Erase 1 word of data EEPROM.
0x4072	Erase all executive memory.
0x4071	Erase 1 row (32 instruction words) from 1 panel of code memory.
0x406E	Erase Boot Secure and General Segments, then erase FBS, FSS and FGS configuration registers.
0x4066	Erase all Data EEPROM allocated to Boot Segment.
0x405E	Erase Secure and General Segments, then erase FSS and FGS configuration registers.
0x4056	Erase all Data EEPROM allocated to Secure Segment.
0x404E	Erase General Segment, then erase FGS configuration register.
0x4046	Erase all Data EEPROM allocated to General Segment.

**TABLE 11-3: NVMCON WRITE OPERATIONS**

NVMCON Value	Write Operation
0x4008	Write 1 word to configuration memory.
0x4005	Write 1 row (16 words) to data memory.
0x4004	Write 1 word to data memory.
0x4001	Write 1 row (32 instruction words) into 1 panel of program memory.

### 11.4.2 UNLOCKING NVMCON FOR PROGRAMMING

Writes to the WR bit (NVMCON<15>) are locked to prevent accidental programming from taking place. Writing a key sequence to the NVMKEY register unlocks the WR bit and allows it to be written to. The unlock sequence is performed as follows:

```
MOV    #0x55, W8
MOV    W8, NVMKEY
MOV    #0xAA, W9
MOV    W9, NVMKEY
```

**Note:** Any working register, or working register pair, can be used to write the unlock sequence.

### 11.4.3 STARTING AND STOPPING A PROGRAMMING CYCLE

Once the unlock key sequence has been written to the NVMKEY register, the WR bit (NVMCON<15>) is used to start and stop an erase or write cycle. Setting the WR bit initiates the programming cycle. Clearing the WR bit terminates the programming cycle.

All erase and write cycles must be externally timed. An external delay must be used between setting and clearing the WR bit. Starting and stopping a programming cycle is performed as follows:

```
BSET    NVMCON, #WR
<Wait 2 ms>
BCLR    NVMCON, #WR
```

## 11.5 Erasing Program Memory in Normal-Voltage Systems

The procedure for erasing program memory (all code memory, data memory, executive memory and code-protect bits) consists of setting NVMCON to 0x407F, unlocking NVMCON for erasing and then executing the programming cycle. This method of bulk erasing program memory only works for systems where VDD is between 4.5 volts and 5.5 volts. The method for erasing program memory for systems with a lower VDD (3.0 volts–4.5 volts) is described in [Section 6.1 “Erasing Memory”](#).

# dsPIC30F Flash Programming Specification

## 11.6 Erasing Program Memory in Low-Voltage Systems

The procedure for erasing program memory (all code memory and data memory) in low-voltage systems (with VDD between 2.5 volts and 4.5 volts) is quite different than the procedure for erasing program memory in normal-voltage systems. Instead of using a Bulk Erase operation, each region of memory must be individually erased by row. Namely, all of the code memory, executive memory and data memory must be erased one row at a time. This procedure is detailed in [Table 11-5](#).

Due to security restrictions, the FBS, FSS and FGS register cannot be erased in low-voltage systems. Once any bits in the FGS register are programmed to '0', they can only be set back to '1' by performing a Bulk Erase in a normal-voltage system. Alternatively, a Segment Erase operation can be performed instead of a Bulk Erase.

Normal-voltage systems can also be used to erase program memory as shown in [Table 11-5](#). However, since this method is more time-consuming and does not clear the code-protect bits, it is not recommended.

**Note:** Program memory must be erased before writing any data to program memory.

**TABLE 11-5: SERIAL INSTRUCTION EXECUTION FOR ERASING PROGRAM MEMORY (EITHER IN LOW-VOLTAGE OR NORMAL-VOLTAGE SYSTEMS)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Initialize NVMADR and NVMADRU to erase code memory and initialize W7 for row address updates.</b>		
0000	EB0300	CLR W6
0000	883B16	MOV W6, NVMADR
0000	883B26	MOV W6, NVMADRU
0000	200407	MOV #0x40, W7
<b>Step 3: Set NVMCON to erase 1 row of code memory.</b>		
0000	24071A	MOV #0x4071, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 4: Unlock the NVMCON to erase 1 row of code memory.</b>		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 5: Initiate the erase cycle.</b>		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P13a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP

# dsPIC30F Flash Programming Specification

## 11.8 Writing Code Memory

The procedure for writing code memory is similar to the procedure for clearing the Configuration registers, except that 32 instruction words are programmed at a time. To facilitate this operation, working registers W0:W5 are used as temporary holding registers for the data to be programmed.

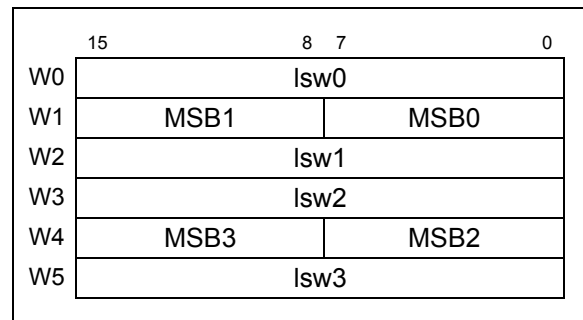
Table 11-8 shows the ICSP programming details, including the serial pattern with the ICSP command code, which must be transmitted Least Significant bit first using the PGC and PGD pins (see Figure 11-2). In Step 1, the Reset vector is exited. In Step 2, the NVMCON register is initialized for single-panel programming of code memory. In Step 3, the 24-bit starting destination address for programming is loaded into the TBLPAG register and W7 register. The upper byte of the starting destination address is stored to TBLPAG, while the lower 16 bits of the destination address are stored to W7.

To minimize the programming time, the same packed instruction format that the programming executive uses is utilized (Figure 8-2). In Step 4, four packed instruction words are stored to working registers W0:W5 using the MOV instruction and the read pointer W6 is initialized. The contents of W0:W5 holding the packed instruction word data is shown in Figure 11-4.

In Step 5, eight TBLWT instructions are used to copy the data from W0:W5 to the write latches of code memory. Since code memory is programmed 32 instruction words at a time, Steps 4 and 5 are repeated eight times to load all the write latches (Step 6).

After the write latches are loaded, programming is initiated by writing to the NVMKEY and NVMCON registers in Steps 7 and 8. In Step 9, the internal PC is reset to 0x100. This is a precautionary measure to prevent the PC from incrementing into unimplemented memory when large devices are being programmed. Lastly, in Step 10, Steps 2-9 are repeated until all of code memory is programmed.

**FIGURE 11-5: PACKED INSTRUCTION WORDS IN W0:W5**



**TABLE 11-8: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Set the NVMCON to program 32 instruction words.</b>		
0000	24001A	MOV #0x4001, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 3: Initialize the write pointer (W7) for TBLWT instruction.</b>		
0000	200xx0	MOV #<DestinationAddress23:16>, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx7	MOV #<DestinationAddress15:0>, W7
<b>Step 4: Initialize the read pointer (W6) and load W0:W5 with the next 4 instruction words to program.</b>		
0000	2xxxx0	MOV #<LSW0>, W0
0000	2xxxx1	MOV #<MSB1:MSB0>, W1
0000	2xxxx2	MOV #<LSW1>, W2
0000	2xxxx3	MOV #<LSW2>, W3
0000	2xxxx4	MOV #<MSB3:MSB2>, W4
0000	2xxxx5	MOV #<LSW3>, W5

# dsPIC30F Flash Programming Specification

## 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
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Set the NVMCON to write 16 data words.</b>		
0000	24005A	MOV #0x4005, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 3: Initialize the write pointer (W7) for TBLWT instruction.</b>		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx7	MOV #<DestinationAddress15:0>, W7
<b>Step 4: Load W0:W3 with the next 4 data words to program.</b>		
0000	2xxxx0	MOV #<WORD0>, W0
0000	2xxxx1	MOV #<WORD1>, W1
0000	2xxxx2	MOV #<WORD2>, W2
0000	2xxxx3	MOV #<WORD3>, W3
<b>Step 5: Set the read pointer (W6) and load the (next set of) write latches.</b>		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 6: Repeat steps 4-5 four times to load the write latches for 16 data words.</b>		

# dsPIC30F Flash Programming Specification

**TABLE 11-10: SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 4: Output W0:W5 using the VISI register and REGOUT command.</b>		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C24	MOV W4, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C25	MOV W5, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
<b>Step 5: Reset the device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 6: Repeat steps 3-5 until all desired code memory is read.</b>		

# dsPIC30F Flash Programming Specification

## 11.11 Reading Configuration Memory

The procedure for reading configuration memory is similar to the procedure for reading code memory, except that 16-bit data words are read instead of 24-bit words. Since there are seven Configuration registers, they are read one register at a time.

Table 11-11 shows the ICSP programming details for reading all of the configuration memory. Note that the TBLPAG register is hard-coded to 0xF8 (the upper byte address of configuration memory), and the read pointer W6 is initialized to 0x0000.

**TABLE 11-11: SERIAL INSTRUCTION EXECUTION FOR READING ALL CONFIGURATION MEMORY**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Initialize TBLPAG, and the read pointer (W6) and the write pointer (W7) for TBLRD instruction.</b>		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
0000	EB0300	CLR W6
0000	EB0380	CLR W7
0000	000000	NOP
<b>Step 3: Read the Configuration register and write it to the VISI register (located at 0x784).</b>		
0000	BA0BB6	TBLRDL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	883C20	MOV W0, VISI
0000	000000	NOP
<b>Step 4: Output the VISI register using the REGOUT command.</b>		
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
<b>Step 5: Reset device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 6: Repeat steps 3-5 six times to read all of configuration memory.</b>		



# dsPIC30F Flash Programming Specification

## 11.12 Reading Data Memory

The procedure for reading data memory is similar to that of reading code memory, except that 16-bit data words are read instead of 24-bit words. Since less data is read in each operation, only working registers W0:W3 are used as temporary holding registers for the data to be read.

Table 11-12 shows the ICSP programming details for reading data memory. Note that the TBLPAG register is hard-coded to 0x7F (the upper byte address of all locations of data memory).

**TABLE 11-12: SERIAL INSTRUCTION EXECUTION FOR READING DATA MEMORY**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Initialize TBLPAG and the read pointer (W6) for TBLRD instruction.</b>		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx6	MOV #<SourceAddress15:0>, W6
<b>Step 3: Initialize the write pointer (W7) and store the next four locations of code memory to W0:W5.</b>		
0000	EB0380	CLR W7
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 4: Output W0:W5 using the VISI register and REGOUT command.</b>		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
<b>Step 5: Reset device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 6: Repeat steps 3-5 until all desired data memory is read.</b>		

# dsPIC30F Flash Programming Specification

## 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 MCLR to V<sub>IL</sub>. 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
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Initialize TBLPAG and the read pointer (W0) for TBLRD instruction.</b>		
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
<b>Step 3: Output the VISI register using the REGOUT command.</b>		
0001	<VISI>	Clock out contents of the VISI register
0000	000000	NOP

# dsPIC30F Flash Programming Specification

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

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 8:</b> 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	BEBBB6	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	BEBBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 9:</b> Repeat Steps 7-8 eight times to load the write latches for the 32 instructions.		
<b>Step 10:</b> 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
<b>Step 11:</b> Initiate the programming cycle.		
0000	A8E761	BSET NVMCON, #15
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #15
0000	000000	NOP
0000	000000	NOP
<b>Step 12:</b> Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 13:</b> Repeat Steps 7-12 until all 23 rows of executive memory are programmed.		

# dsPIC30F Flash Programming Specification

**TABLE 12-2: READING EXECUTIVE MEMORY (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 4:</b> Output W0:W5 using the VISI register and REGOUT command.		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C24	MOV W4, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
0000	883C25	MOV W5, VISI
0000	000000	NOP
0001	—	Clock out contents of VISI register
<b>Step 5:</b> Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 6:</b> Repeat Steps 3-5 until all 736 instruction words of executive memory are read.		

---

**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.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscent Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICTail, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 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.*

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