

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Obsolete
Core Processor	dsPIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	132KB (44K x 24)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	6K 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/dspic30f6011t-20i-pf

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

2.2 Pins Used During Programming

The pins identified in Table 2-1 are used for device programming. Refer to the appropriate device data sheet for complete pin descriptions.

TABLE 2-1:dsPIC30F PIN DESCRIPTIONSDURING PROGRAMMING

Pin Name	Pin Type	Pin Description
MCLR/VPP	Р	Programming Enable
Vdd	Р	Power Supply
Vss	Р	Ground
PGC	I	Serial Clock
PGD	I/O	Serial Data

Legend: I = Input, O = Output, P = Power

2.3 Program Memory Map

The program memory space extends from 0x0 to 0xFFFFFE. Code storage is located at the base of the memory map and supports up to 144 Kbytes (48K instruction words). Code is stored in three, 48 Kbyte memory panels that reside on-chip. Table 2-2 shows the location and program memory size of each device.

Locations 0x800000 through 0x8005BE are reserved for executive code memory. This region stores either the programming executive or debugging executive. The programming executive is used for device programming, while the debug executive is used for incircuit debugging. This region of memory cannot be used to store user code.

Locations 0xF80000 through 0xF8000E are reserved for the Configuration registers. The bits in these registers may be set to select various device options, and are described in Section 5.7 "Configuration Bits Programming".

Locations 0xFF0000 and 0xFF0002 are reserved for the Device ID registers. These bits can be used by the programmer to identify what device type is being programmed and are described in **Section 10.0 "Device ID"**. The device ID reads out normally, even after code protection is applied.

Figure 2-2 illustrates the memory map for the dsPIC30F devices.

2.4 Data EEPROM Memory

The Data EEPROM array supports up to 4 Kbytes of data and is located in one memory panel. It is mapped in program memory space, residing at the end of User Memory Space (see Figure 2-2). Table 2-2 shows the location and size of data EEPROM in each device.

TABLE 2-2: CODE MEMORY AND DATA EEPROM MAP AND SIZE

Device	Code Memory map (Size in Instruction Words)	Data EEPROM Memory Map (Size in Bytes)
dsPIC30F2010	0x000000-0x001FFE (4K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F2011	0x000000-0x001FFE (4K)	None (0K)
dsPIC30F2012	0x000000-0x001FFE (4K)	None (0K)
dsPIC30F3010	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3011	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3012	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3013	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F3014	0x000000-0x003FFE (8K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F4011	0x000000-0x007FFE (16K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F4012	0x000000-0x007FFE (16K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F4013	0x000000-0x007FFE (16K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5011	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5013	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5015	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F5016	0x000000-0x00AFFE (22K)	0x7FFC00-0x7FFFFE (1K)
dsPIC30F6010	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6010A	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFF (4K)
dsPIC30F6011	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6011A	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6012	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6012A	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6013	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6013A	0x000000-0x015FFE (44K)	0x7FF800-0x7FFFFE (2K)
dsPIC30F6014	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6014A	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)
dsPIC30F6015	0x000000-0x017FFE (48K)	0x7FF000-0x7FFFFE (4K)

5.0 DEVICE PROGRAMMING

5.1 Overview of the Programming Process

Once the programming executive has been verified in memory (or loaded if not present), the dsPIC30F can be programmed using the command set shown in Table 5-1. A detailed description for each command is provided in Section 8.0 "Programming Executive Commands".

Command	Description
SCHECK	Sanity check
READD	Read data EEPROM, Configuration registers and device ID
READP	Read code memory
PROGD	Program one row of data EEPROM and verify
PROGP	Program one row of code memory and verify
PROGC	Program Configuration bits and verify
ERASEB	Bulk Erase, or erase by segment
ERASED	Erase data EEPROM
ERASEP	Erase code memory
QBLANK	Query if the code memory and data EEPROM are blank
QVER	Query the software version

TABLE 5-1: COMMAND SET SUMMARY

A high-level overview of the programming process is illustrated in Figure 5-1. The process begins by entering Enhanced ICSP mode. The chip is then bulk erased, which clears all memory to '1' and allows the device to be programmed. The Chip Erase is verified before programming begins. Next, the code memory, data Flash and Configuration bits are programmed. As these memories are programmed, they are each verified to ensure that programming was successful. If no errors are detected, the programming is complete and Enhanced ICSP mode is exited. If any of the verifications fail, the procedure should be repeated, starting from the Chip Erase. If Advanced Security features are enabled, then individual Segment Erase operations need to be performed, based on user selections (i.e., based on the specific needs of the user application). The specific operations that are used typically depend on the order in which various segments need to be programmed for a given application or system.

Section 5.2 "Entering Enhanced ICSP Mode" through Section 5.8 "Exiting Enhanced ICSP Mode" describe the programming process in detail.

FIGURE 5-1: PROGRAMMING FLOW



5.5 Code Memory Programming

5.5.1 OVERVIEW

The Flash code memory array consists of 512 rows of thirty-two, 24-bit instructions. Each panel stores 16K instruction words, and each dsPIC30F device has either 1, 2 or 3 memory panels (see Table 5-2).

	Code Size	Number	Number
Device	(24-bit	of	of
	Words)	Rows	Panels
dsPIC30F2010	4K	128	1
dsPIC30F2011	4K	128	1
dsPIC30F2012	4K	128	1
dsPIC30F3010	8K	256	1
dsPIC30F3011	8K	256	1
dsPIC30F3012	8K	256	1
dsPIC30F3013	8K	256	1
dsPIC30F3014	8K	256	1
dsPIC30F4011	16K	512	1
dsPIC30F4012	16K	512	1
dsPIC30F4013	16K	512	1
dsPIC30F5011	22K	704	2
dsPIC30F5013	22K	704	2
dsPIC30F5015	22K	704	2
dsPIC30F5016	22K	704	2
dsPIC30F6010	48K	1536	3
dsPIC30F6010A	48K	1536	3
dsPIC30F6011	44K	1408	3
dsPIC30F6011A	44K	1408	3
dsPIC30F6012	48K	1536	3
dsPIC30F6012A	48K	1536	3
dsPIC30F6013	44K	1408	3
dsPIC30F6013A	44K	1408	3
dsPIC30F6014	48K	1536	3
dsPIC30F6014A	48K	1536	3
dsPIC30F6015	48K	1536	3

TABLE 5-2: DEVICE CODE MEMORY SIZE

5.5.2 PROGRAMMING METHODOLOGY

Code memory is programmed with the PROGP command. PROGP programs one row of code memory to the memory address specified in the command. The number of PROGP commands required to program a device depends on the number of rows that must be programmed in the device.

A flowchart for programming of code memory is illustrated in Figure 5-3. In this example, all 48K instruction words of a dsPIC30F6014A device are programmed. First, the number of commands to send (called 'RemainingCmds' in the flowchart) is set to 1536 and the destination address (called 'BaseAddress') is set to '0'. Next, one row in the device is programmed with a PROGP command. Each PROGP command contains data for one row of code memory of the dsPIC30F6014A. After the first command is processed successfully, 'RemainingCmds' is decremented by 1 and compared to 0. Since there are more PROGP commands to send, 'BaseAddress' is incremented by 0x40 to point to the next row of memory.

On the second PROGP command, the second row of each memory panel is programmed. This process is repeated until the entire device is programmed. No special handling must be performed when a panel boundary is crossed.



FLOWCHART FOR PROGRAMMING dsPIC30F6014A CODE MEMORY



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



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 0x0001FF] 011 = No Boot Segment 010 = High security; Small-sized Boot Program Flash [Boot Segment starts after BS and ends at 0x0003FF] 011 = High security; Medium-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] 001 = High security; Large-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 0x000FFF]
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] 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] 01 = Large-sized Secure Data EEPROM [(512 – N) bytes of 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/6011A/6013A, and (2048 – N) bytes in dsPIC30F6010A/6012A/6014A/6015]

TABLE 5-7: CONFIGURATION BITS DESCRIPTION (CONTINUED)

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)		
•		
•		
Comn	nand 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	
ls	w1	
MSB2	MSB1	
lsw2		

Iswx: 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 Isw2			
	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 Command Descriptions

All commands that are supported by the programming executive are described in Section 8.5.1 "SCHECK Command" through Section 8.5.11 "QVER Command".

8.5.1 SCHECK COMMAND

15	12	11	0
	Opcode	Length	

Field	Description
Opcode	0x0
Length	0x1

The SCHECK command instructs the programming executive to do nothing, but generate a response. This command is used as a "sanity check" to verify that the programming executive is operational.

Expected Response (2 words):

0x1000 0x0002

Note: This instruction is not required for programming, but is provided for development purposes only.

8.5.2 READD COMMAND

15	12	11	8	7	0	
Орс	ode			Length		
Reser	ved0	N				
Reserved1				Addr_MSB		
	Addr_LS					

Field	Description
Opcode	0x1
Length	0x4
Reserved0	0x0
Ν	Number of 16-bit words to read (max of 2048)
Reserved1	0x0
Addr_MSB	MSB of 24-bit source address
Addr_LS	LS 16 bits of 24-bit source address

The READD command instructs the programming executive to read N 16-bit words of memory starting from the 24-bit address specified by Addr_MSB and Addr_LS. This command can only be used to read 16-bit data. It can be used to read data EEPROM, Configuration registers and the device ID.

Expected Response (2+N words):

0x1100 N + 2 Data word 1

Data word N

Note: Reading unimplemented memory will cause the programming executive to reset.

dsPIC30F Flash Programming Specification

8.5.7 ERASEB COMMAND

15 12	11	2	0
Opcode	Length		
	Reserved	MS	3

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 $0x2 = All Code and Data EEPROM inGeneral Segment, interrupt vectors andFGS Configuration register0x3 = Full Chip Erase0x4 = All Code and Data EEPROM inBoot, Secure and General Segments,and FBS, FSS and FGS Configurationregisters0x5 = All Code and Data EEPROM inSecure and General Segments, andFSS and FGS Configuration registers0x6 = All Data EEPROM in BootSegment0x7 = All Data EEPROM in SecureSegment$

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	
Оро	code			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 2×16 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 10-1shows the device ID for each device,Table 10-2shows the device ID registers and Table 10-33describes the bit field of each register.

Davias		Silicon Revision							
Device	DEVID	A0	A1	A2	A3	A4	В0	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	_	_	_

TARI	F 10)_1·	DEVICE IDS
		/- .	

TABLE 10-2: dsPIC30F DEVICE ID REGISTERS

Addroop	Nama								В	it							
Address	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF0000	DEVID		DEVID<15:0>														
0xFF0002	DEVREV	PROC<3:0>			REV<5:0>					DOT<5:0>							

Bit Field	Register	Description			
DEVID<15:0>	DEVID	Encodes the device ID.			
PROC<3:0>	DEVREV	Encodes the process of the device (always read as 0x001).			
REV<5:0>	DEVREV	Encodes the major revision number of the device. 000000 = A			
		000001 = B 000010 = C			
DOT<5:0>	DEVREV	Encodes the minor revision number of the device.			
		000000 = 0			
		000001 = 1			
		000010 = 2			
Examples:					
Rev A.1 = 0000 000	0 0000 0001				
Rev A.2 = 0000 000	0 0000 0010				
Rev B.0 = 0000 000	0 0100 0000				
This formula applies to	o all dsPIC30F devices	s, with the exception of the following:			
• dsPIC30F6010					
 dsPIC30F6011 					
• dsPIC30F6012					
• dsPIC30F6013					
• dsPIC30F6014					
Refer to Table 10-1 for the actual revision IDs.					

TABLE 10-3: DEVICE ID BITS DESCRIPTION

11.0 ICSP™ MODE

11.1 ICSP Mode

ICSP mode is a special programming protocol that allows you to read and write to the dsPIC30F programming executive. The ICSP mode is the second (and slower) method used to program the device. This mode also has the ability to read the contents of executive memory to determine whether the programming executive is present. This capability is accomplished by applying control codes and instructions serially to the device using pins PGC and PGD.

In ICSP mode, the system clock is taken from the PGC pin, regardless of the device's oscillator Configuration bits. All instructions are first shifted serially into an internal buffer, then loaded into the Instruction register and executed. No program fetching occurs from internal memory. Instructions are fed in 24 bits at a time. PGD is used to shift data in and PGC is used as both the serial shift clock and the CPU execution clock.

Data is transmitted on the rising edge and latched on the falling edge of PGC. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

Note 1: During ICSP operation, the operating frequency of PGC must not exceed 5 MHz.
2: Because ICSP is slower, it is recommended that only Enhanced ICSP (E-ICSP) mode be used for device programming, as described in Section 5.1 "Overview of the Programming Process".

11.2 ICSP Operation

Upon entry into ICSP mode, the CPU is idle. Execution of the CPU is governed by an internal state machine. A 4-bit control code is clocked in using PGC and PGD, and this control code is used to command the CPU (see Table 11-1).

The SIX control code is used to send instructions to the CPU for execution, while the REGOUT control code is used to read data out of the device via the VISI register. The operation details of ICSP mode are provided in Section 11.2.1 "SIX Serial Instruction Execution" and Section 11.2.2 "REGOUT Serial Instruction Execution".

TABLE 11-1:CPU CONTROL CODES IN
ICSP™ MODE

4-bit Control Code	Mnemonic	Description
0000b	SIX	Shift in 24-bit instruction and execute.
0001b	REGOUT	Shift out the VISI register.
0010b-1111b	N/A	Reserved.

11.2.1 SIX SERIAL INSTRUCTION EXECUTION

The SIX control code allows execution of dsPIC30F assembly instructions. When the SIX code is received, the CPU is suspended for 24 clock cycles as the instruction is then clocked into the internal buffer. Once the instruction is shifted in, the state machine allows it to be executed over the next four clock cycles. While the received instruction is executed, the state machine simultaneously shifts in the next 4-bit command (see Figure 11-2).

- Note 1: Coming out of the ICSP entry sequence, the first 4-bit control code is always forced to SIX and a forced NOP instruction is executed by the CPU. Five additional PGC clocks are needed on startup, thereby resulting in a 9-bit SIX command instead of the normal 4-bit SIX command. After the forced SIX is clocked in, ICSP operation resumes as normal (the next 24 clock cycles load the first instruction word to the CPU). See Figure 11-1 for details.
 - 2: TBLRDH, TBLRDL, TBLWTH and TBLWTL instructions must be followed by a NOP instruction.

11.7 Writing Configuration Memory

The FOSC, FWDT, FBORPOR and FICD registers are not erasable. It is recommended that all Configuration registers be set to a default value after erasing program memory. The FWDT, FBORPOR and FICD registers can be set to a default all '1's value by programming 0xFFFF to each register. Since these registers contain unimplemented bits that read as '0' the default values shown in Table 11-6 will be read instead of 0xFFFF. The recommended default FOSC value is 0xC100, which selects the FRC clock oscillator setting.

The FGS, FBS and FSS Configuration registers are special since they enable code protection for the device. For security purposes, once any bit in these registers is programmed to '0' (to enable some code protection feature), it can only be set back to '1' by performing a Bulk Erase or Segment Erase as described in Section 11.5 "Erasing Program Memory in Normal-Voltage Systems". Programming these bits from a '0' to '1' is not possible, but they may be programmed from a '1' to a '0' to enable code protection.

Table 11-7 shows the ICSP programming details for clearing the Configuration registers. In Step 1, the Reset vector is exited. In Step 2, the write pointer (W7) is loaded with 0x0000, which is the original destination address (in TBLPAG 0xF8 of program memory). In Step 3, the NVMCON is set to program one Configura-

tion register. In Step 4, the TBLPAG register is initialized, to 0xF8, for writing to the Configuration registers. In Step 5, the value to write to the each Configuration register (0xFFFF) is loaded to W6. In Step 6, the Configuration register data is written to the write latch using the TBLWTL instruction. In Steps 7 and 8, the NVMCON is unlocked for programming and the programming cycle is initiated, as described in Section 11.4 "Flash Memory Programming in ICSP Mode". In Step 9, the internal PC is set to 0x100 as a safety measure to prevent the PC from incrementing into unimplemented memory. Lastly, Steps 3-9 are repeated six times until all seven Configuration registers are cleared.

TABLE 11-6:	DEFAULT CONFIGURATION
	REGISTER VALUES

Address	Register	Default Value		
0xF80000	FOSC	0xC100		
0xF80002	FWDT	0x803F		
0xF80004	FBORPOR	0x87B3		
0xF80006	FBS	0x310F		
0xF80008	FSS	0x330F		
0xF8000A	FGS	0x0007		
0xF8000C	FICD	0xC003		

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

Command (Binary)	Data (Hexadecimal)	Description
Step 1: Exit th	e Reset vector.	
0000 0000 0000	040100 040100 000000	GOTO 0x100 GOTO 0x100 NOP
Step 2: Initiali	ze the write pointer (W7) for the TBLWT instruction.
0000	200007	MOV #0x0000, W7
Step 3: Set th	e NVMCON to progr	am 1 Configuration register.
0000	24008A 883B0A	MOV #0x4008, W10 MOV W10, NVMCON
Step 4: Initialize the TBLPAG register.		
0000	200F80 880190	MOV #0xF8, W0 MOV W0, TBLPAG
Step 5: Load the Configuration register data to W6.		
0000	2xxxx0 000000	MOV # <config_value>, W0 NOP</config_value>

11.10 Reading Code Memory

Reading from code memory is performed by executing a series of TBLRD instructions and clocking out the data using the REGOUT command. To ensure efficient execution and facilitate verification on the programmer, four instruction words are read from the device at a time.

Table 11-10 shows the ICSP programming details for reading code memory. In Step 1, the Reset vector is exited. In Step 2, the 24-bit starting source address for reading is loaded into the TBLPAG and W6 registers. The upper byte of the starting source address is stored to TBLPAG, while the lower 16 bits of the source address are stored to W6.

To minimize the reading time, the packed instruction word format that was utilized for writing is also used for reading (see Figure 11-5). In Step 3, the write pointer W7 is initialized, and four instruction words are read from code memory and stored to working registers W0:W5. In Step 4, the four instruction words are clocked out of the device from the VISI register using the REGOUT command. In Step 5, the internal PC is reset to 0x100, as a precautionary measure, to prevent the PC from incrementing into unimplemented memory when large devices are being read. Lastly, in Step 6, Steps 3-5 are repeated until the desired amount of code memory is read.

TABLE 11-10:	SERIAL INSTRUCTION EXECUTION FOR READING CODE MEMORY

Command (Binary)	Data (Hexadecimal)	Description		
Step 1: Exit th	ne Reset vector.			
0000	040100	GOTO 0x100		
0000	040100	GOTO 0x100		
0000	000000	NOP		
Step 2: Initiali	ze TBLPAG and	the read point	er (W6) for TBLRD instruction.	
0000	200xx0	MOV	# <sourceaddress23:16>, W0</sourceaddress23:16>	
0000	880190	MOV	W0, TBLPAG	
0000	2xxxx6	MOV	# <sourceaddress15:0>, W6</sourceaddress15:0>	
Step 3: Initiali	ze the write point	er (W7) and s	tore the next four locations of code memory to W0:W5.	
0000	EB0380	CLR	W7	
0000	000000	NOP		
0000	BA1B96	TBLRDL	[W6], [W7++]	
0000	000000	NOP		
0000	000000	NOP		
0000	BADBB6	TBLRDH.B	[W6++], [W7++]	
0000	000000	NOP		
0000	000000	NOP		
0000	BADBD6	TBLRDH.B	[++W6], [W7++]	
0000	000000	NOP		
0000	000000	NOP		
0000	BA1BB6	TBLRDL	[W6++], [W7++]	
0000	000000	NOP		
0000	000000	NOP		
0000	BA1B96	TBLRDL	[W6], [W7++]	
0000	000000	NOP		
0000	000000	NOP		
0000	BADBB6	TBLRDH.B	[W6++], [W7++]	
0000	000000	NOP		
0000	000000	NOP		
0000	BADBD6	'T'BLRDH.B	[++W6], [W//++]	
0000	000000	NOP		
0000	000000	NOP		
0000	BAUBB6	TRTKDT	[W0++], [W/]	
0000	000000	NOP		
0000	000000	NOP		

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

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

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 programming executive to the memory is described in Section 12.0 "Programming the 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 VIL. Programming can then take place by following the procedure outlined in **Section 5.0 "Device Programming"**.

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

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

Command (Binary)	Data (Hexadecimal)	Description		
Step 4: Output	it W0:W5 using th	ne VISI register and REGOUT command.		
0000	883C20	MOV WO, 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		
Step 5: Reset the device internal PC.				
0000	040100	GOTO 0x100		
0000	000000	NOP		
Step 6: Repeat Steps 3-5 until all 736 instruction words of executive memory are read.				

TABLE 12-2: READING EXECUTIVE MEMORY (CONTINUED)

13.0 AC/DC CHARACTERISTICS AND TIMING REQUIREMENTS

TABLE 13-1: AC/DC CHARACTERISTICS

AC/DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating Temperature: 25° C is recommended			
Param. No.	Sym	Characteristic	Min	Мах	Units	Conditions
D110	Vінн	High Programming Voltage on MCLR/VPP	9.00	13.25	V	
D112	IPP	Programming Current on MCLR/VPP	_	300	μA	_
D113	IDDP	Supply Current during programming	_	30	mA	Row Erase Program memory
				30	mA	Row Erase Data EEPROM
			—	30	mA	Bulk Erase
D001	Vdd	Supply voltage	2.5	5.5	V	
D002	Vddbulk	Supply voltage for Bulk Erase programming	4.5	5.5	V	—
D031	VIL	Input Low Voltage	Vss	0.2 Vss	V	
D041	VIH	Input High Voltage	0.8 VDD	Vdd	V	
D080	Vol	Output Low Voltage	—	0.6	V	IOL = 8.5 mA
D090	Voн	Output High Voltage	Vdd - 0.7	—	V	IOH = -3.0 mA
D012	Сю	Capacitive Loading on I/O Pin (PGD)	—	50	pF	To meet AC specifications
P1	TSCLK	Serial Clock (PGC) period	50	_	ns	ICSP™ mode
			1	—	μs	Enhanced ICSP mode
P1a	TSCLKL	Serial Clock (PGC) low time	20	—	ns	ICSP mode
			400	—	ns	Enhanced ICSP mode
P1b	TSCLKH	Serial Clock (PGC) high time	20	—	ns	ICSP mode
			400	—	ns	Enhanced ICSP mode
P2	TSET1	Input Data Setup Timer to PGC \downarrow	15	_	ns	_
P3	THLD1	Input Data Hold Time from PGC \downarrow	15	—	ns	
P4	TDLY1	Delay between 4-bit command and command operand	20	—	ns	—
P4a	TDLY1a	Delay between 4-bit command operand and next 4-bit command	20	—	ns	_
P5	TDLY2	Delay between last PGC ↓of command to first PGC ↑ of VISI output	20	—	ns	_
P6	TSET2	VDD ↑ setup time to MCLR/VPP	100	—	ns	_
P7	THLD2	Input data hold time from MCLR/VPP 1	2	_	μs	ICSP mode
			5	_	ms	Enhanced ICSP mode
P8	TDLY3	Delay between last PGC ↓ of command word to PGD driven ↑ by programming executive	20	—	μs	_
P9a	TDLY4	Programming Executive Command processing time	10	_	μs	_

AC/DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating Temperature: 25° C is recommended			
Param. No.	Sym	Characteristic	Min	Мах	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

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

APPENDIX B: HEX FILE FORMAT

Flash programmers process the standard HEX format used by the Microchip development tools. The format supported is the Intel[®] HEX 32 Format (INHX32). Please refer to Appendix A in the "*MPASM User's Guide*" (DS33014) for more information about hex file formats.

The basic format of the hex file is:

:ВВААААТТНННН...ННННСС

Each data record begins with a 9-character prefix and always ends with a 2-character checksum. All records begin with ':' regardless of the format. The individual elements are described below.

- BB is a two-digit hexadecimal byte count representing the number of data bytes that appear on the line. Divide this number by two to get the number of words per line.
- AAAA is a four-digit hexadecimal address representing the starting address of the data record. Format is high byte first followed by low byte. The address is doubled because this format only supports 8-bits. Divide the value by two to find the real device address.
- TT is a two-digit record type that will be '00' for data records, '01' for end-of-file records and '04' for extended-address record.
- HHHH is a four-digit hexadecimal data word. Format is low byte followed by high byte. There will be BB/2 data words following TT.
- CC is a two-digit hexadecimal checksum that is the two's complement of the sum of all the preceding bytes in the line record.

Because the Intel hex file format is byte-oriented, and the 16-bit program counter is not, program memory sections require special treatment. Each 24-bit program word is extended to 32 bits by inserting a socalled "phantom byte". Each program memory address is multiplied by 2 to yield a byte address.

As an example, a section that is located at 0x100 in program memory will be represented in the hex file as 0x200.

The hex file will be produced with the following contents:

:020000040000fa

:040200003322110096

:0000001FF

Notice that the data record (line 2) has a load address of 0200, while the source code specified address 0x100. Note also that the data is represented in "littleendian" format, meaning the Least Significant Byte (LSB) appears first. The phantom byte appears last, just before the checksum.

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 BUIk 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