

Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

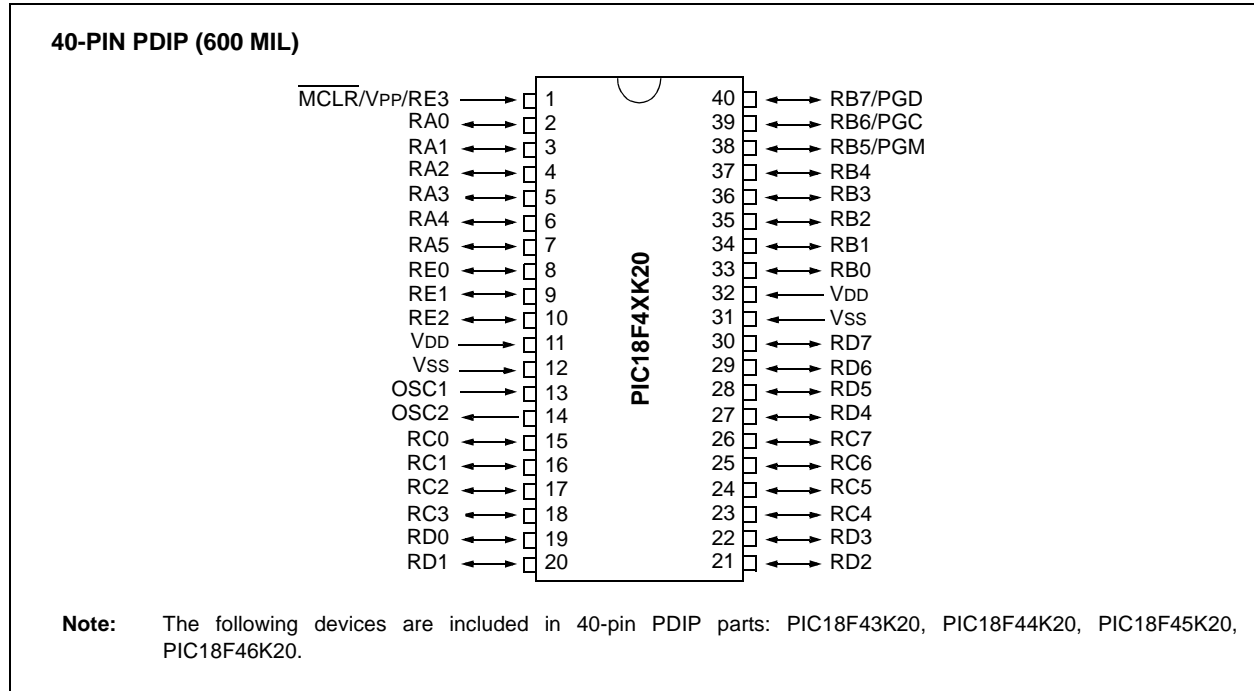
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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](#)"

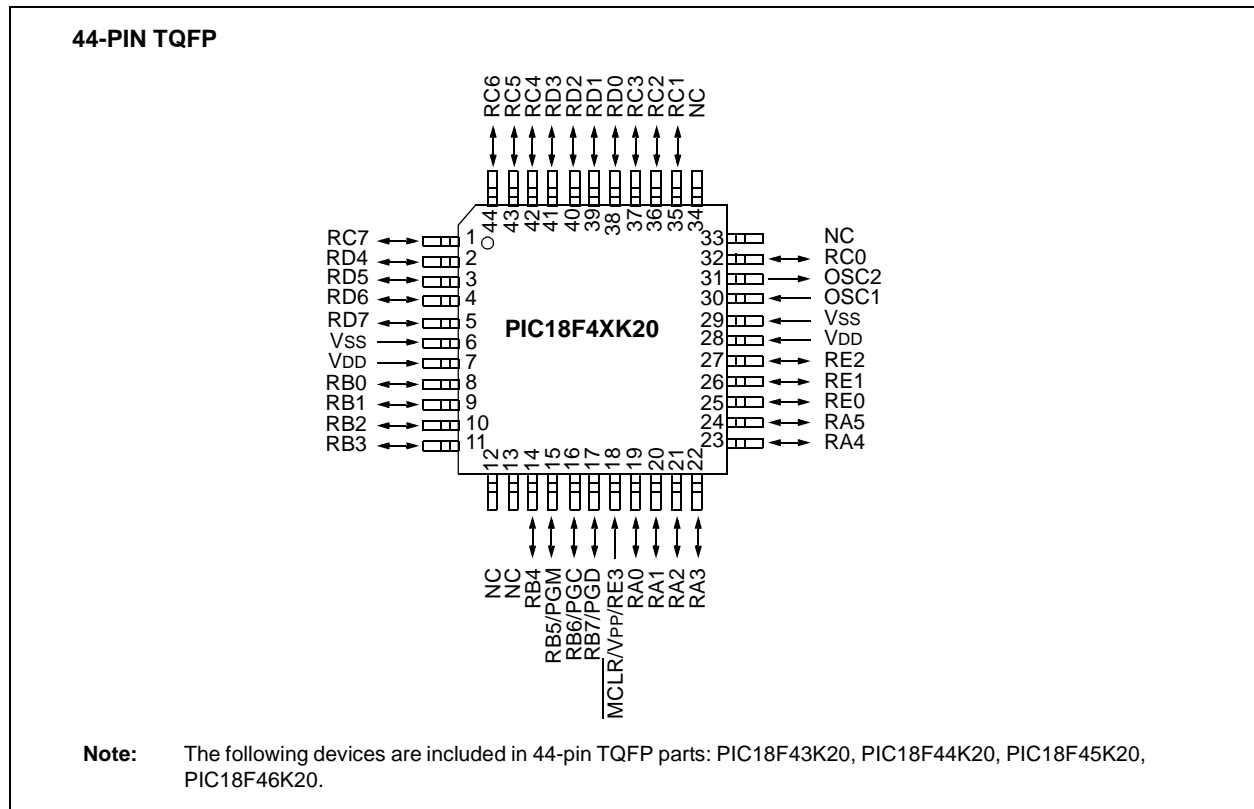
#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 14x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k20-e-mv">https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k20-e-mv</a>

**FIGURE 2-3: 40-PIN PDIP PIN DIAGRAMS**

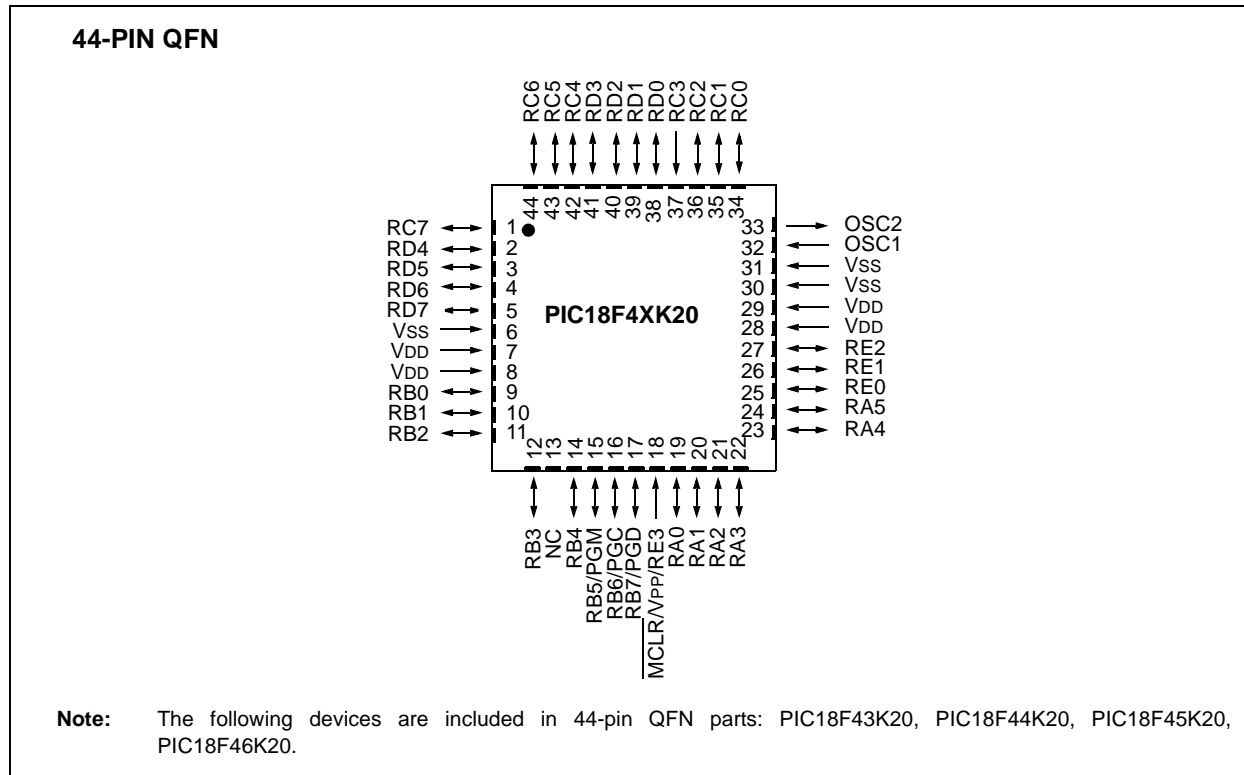


**FIGURE 2-4: 44-PIN TQFP PIN DIAGRAMS**



# PIC18F2XK20/4XK20

FIGURE 2-5: 44-PIN QFN PIN DIAGRAMS



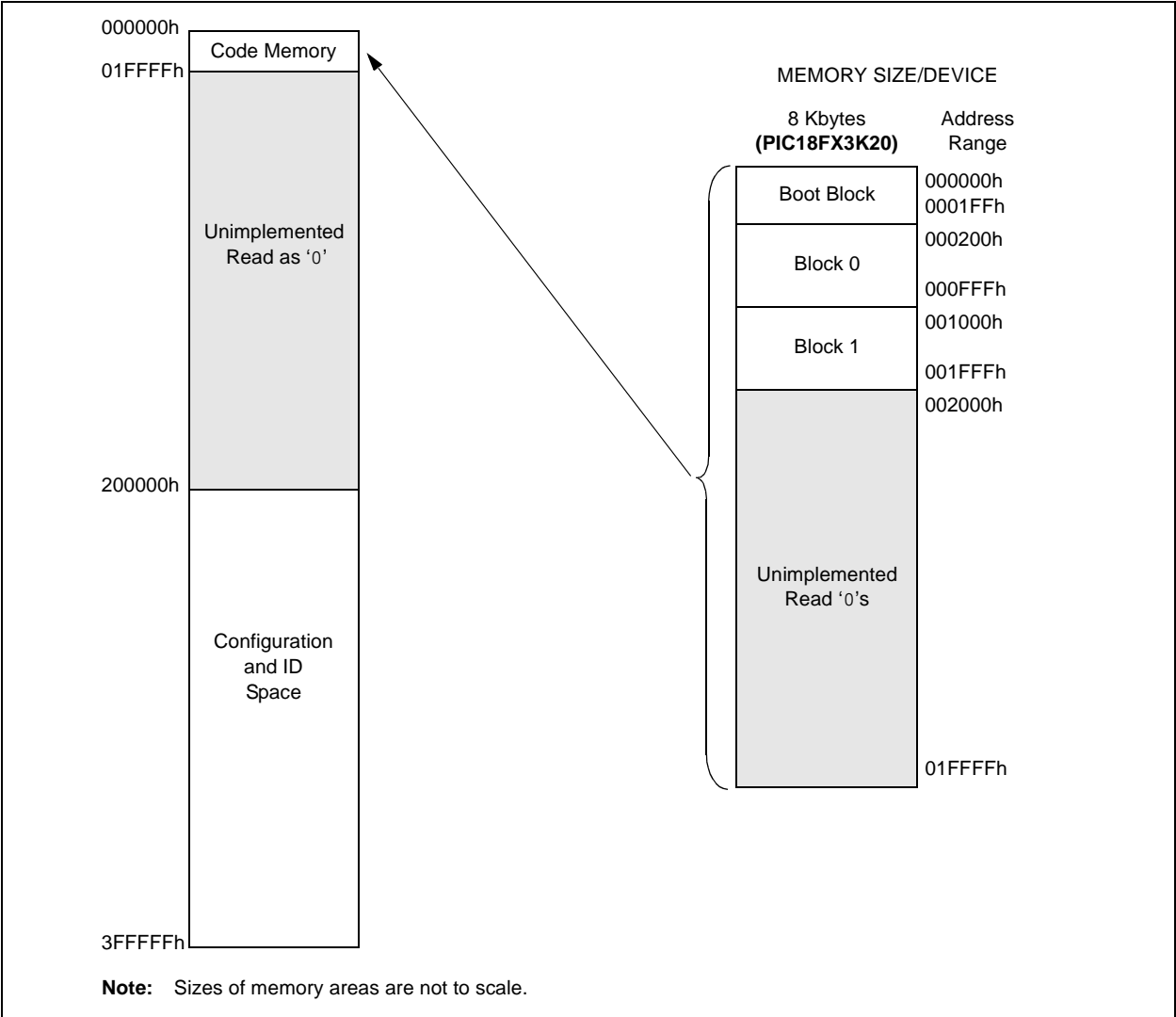
2.3 Memory Maps

For the PIC18FX3K20 devices, the code memory space extends from 0000h to 01FFFh (8 Kbytes) in two 4-Kbyte blocks. Addresses 0000h through 01FFFh, however, define a “Boot Block” region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

TABLE 2-2: IMPLEMENTATION OF CODE MEMORY

Device	Code Memory Size (Bytes)
PIC18F23K20	000000h-001FFFh (8K)
PIC18F43K20	

FIGURE 2-6: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX3K20 DEVICES



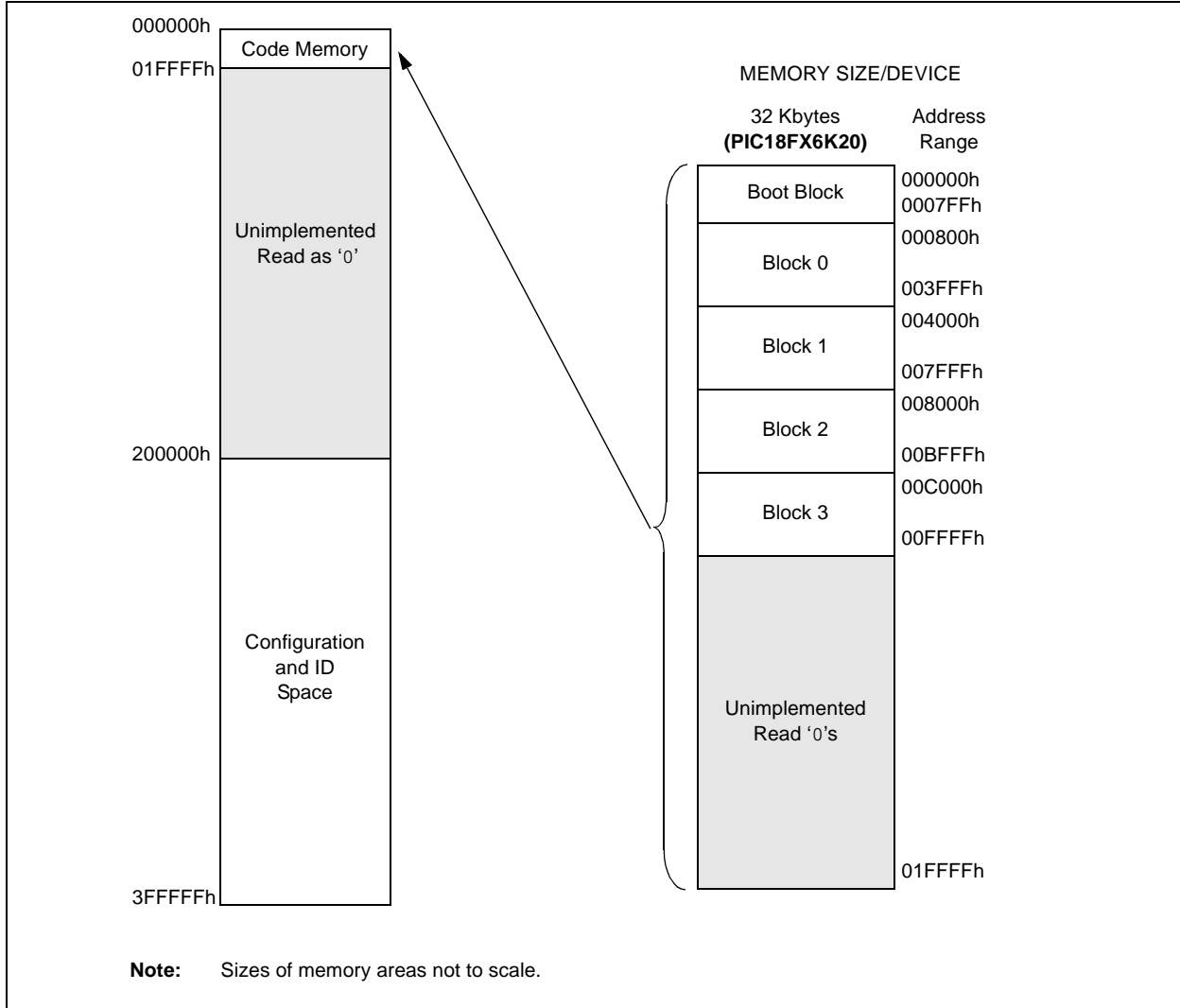
# PIC18F2XK20/4XK20

For PIC18FX6K20 devices, the code memory space extends from 000000h to 00FFFFh (64 Kbytes) in four 16-Kbyte blocks. Addresses 000000h through 0007FFh, however, define a “Boot Block” region that is treated separately from Block 0. All of these blocks define code protection boundaries within the code memory space.

**TABLE 2-5: IMPLEMENTATION OF CODE MEMORY**

Device	Code Memory Size (Bytes)
PIC18F26K20	000000h-00FFFFh (64K)
PIC18F46K20	

**FIGURE 2-9: MEMORY MAP AND THE CODE MEMORY SPACE FOR PIC18FX6K20 DEVICES**



In addition to the code memory space, there are three blocks in the configuration and ID space that are accessible to the user through table reads and table writes. Their locations in the memory map are shown in Figure 2-10.

Users may store identification information (ID) in eight ID registers. These ID registers are mapped in addresses 200000h through 200007h. The ID locations read out normally, even after code protection is applied.

Locations 300000h through 30000Dh are reserved for the Configuration bits. These bits select various device options and are described in **Section 5.0 “Configuration Word”**. These Configuration bits read out normally, even after code protection.

Locations 3FFFEh and 3FFFFh are reserved for the device ID bits. These bits may be used by the programmer to identify what device type is being programmed and are described in **Section 5.0 “Configuration Word”**. These device ID bits read out normally, even after code protection.

## 2.3.1 MEMORY ADDRESS POINTER

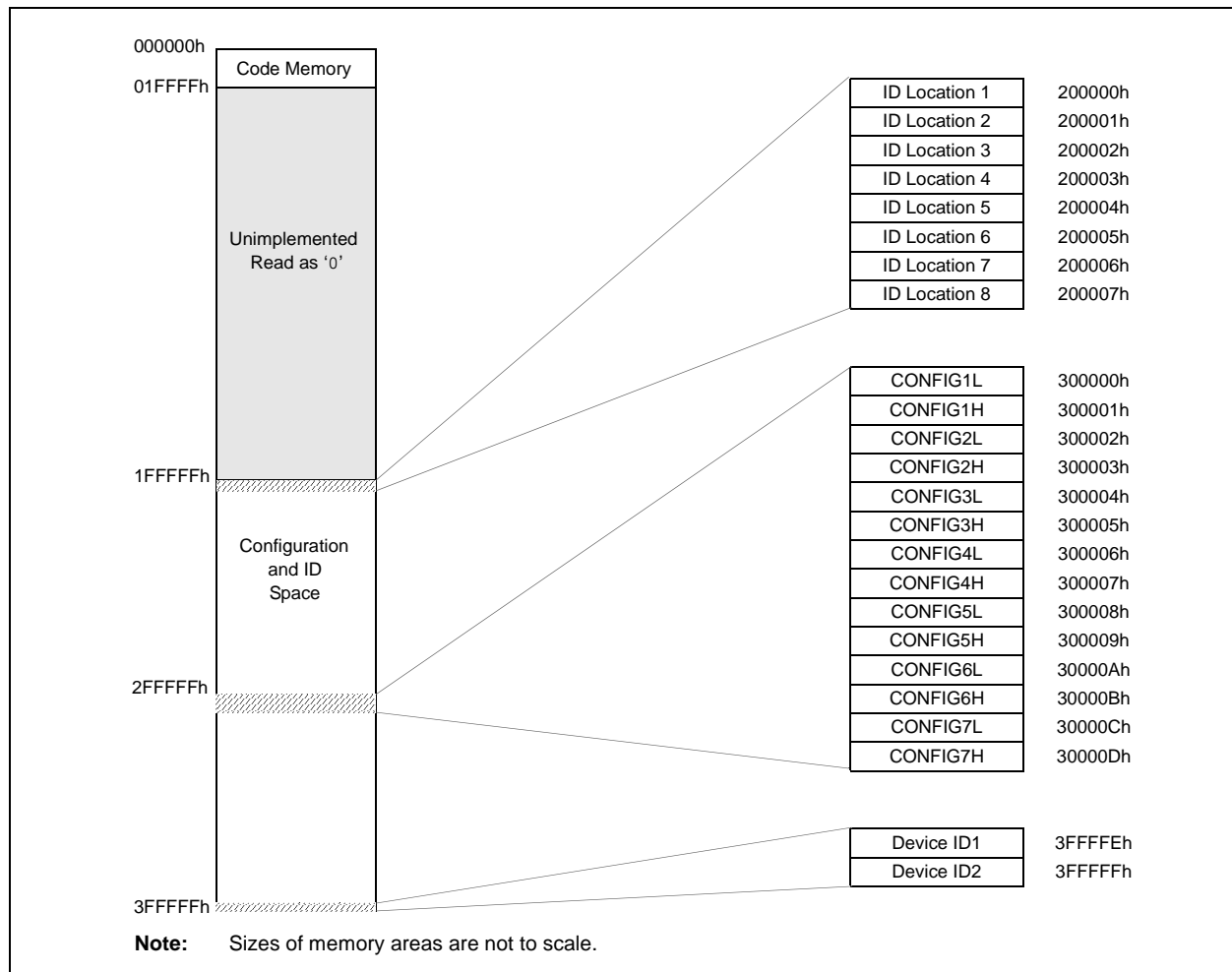
Memory in the address space, 000000h to 3FFFFFFh, is addressed via the Table Pointer register, which is comprised of three Pointer registers:

- TBLPTRU, at RAM address 0FF8h
- TBLPTRH, at RAM address 0FF7h
- TBLPTRL, at RAM address 0FF6h

TBLPTRU	TBLPTRH	TBLPTRL
Addr[21:16]	Addr[15:8]	Addr[7:0]

The 4-bit command, '0000' (core instruction), is used to load the Table Pointer prior to using any read or write operations.

**FIGURE 2-10: CONFIGURATION AND ID LOCATIONS FOR PIC18F2XK20/4XK20 DEVICES**

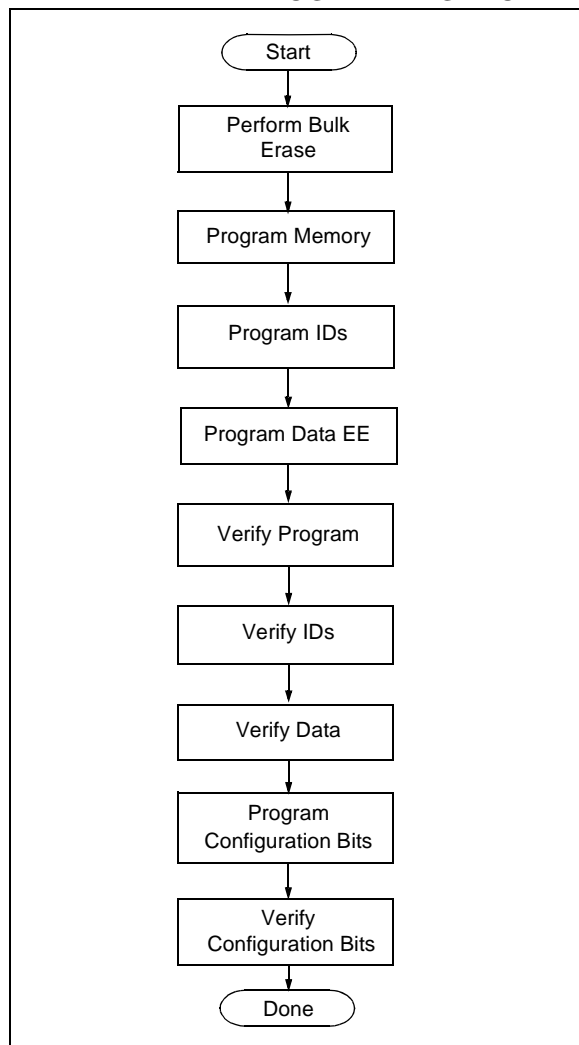


# PIC18F2XK20/4XK20

## 2.4 High-Level Overview of the Programming Process

Figure 2-11 shows the high-level overview of the programming process. First, a Bulk Erase is performed. Next, the code memory, ID locations and data EEPROM are programmed. These memories are then verified to ensure that programming was successful. If no errors are detected, the Configuration bits are then programmed and verified.

**FIGURE 2-11: HIGH-LEVEL PROGRAMMING FLOW**

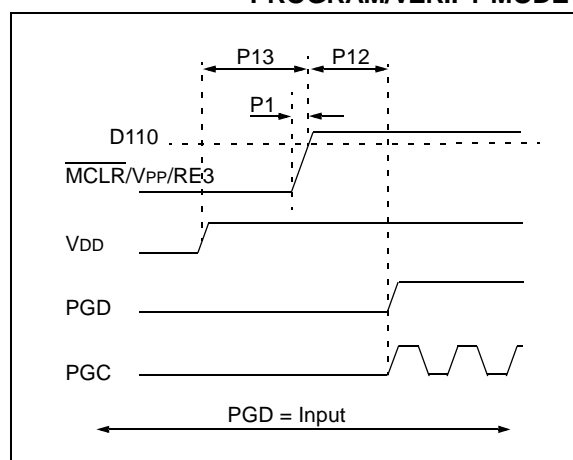


## 2.5 Entering and Exiting High-Voltage ICSP Program/Verify Mode

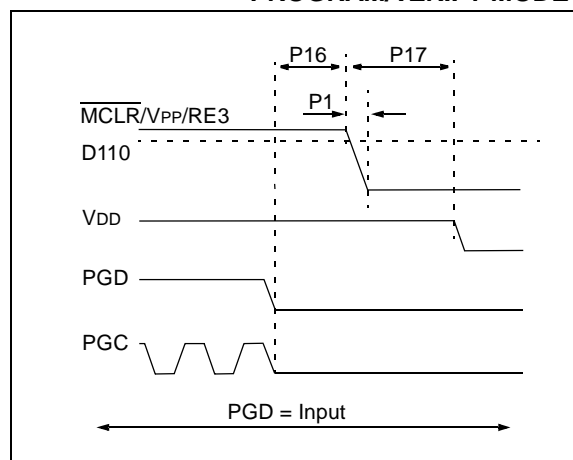
As shown in Figure 2-12, the High-Voltage ICSP Program/Verify mode is entered by holding PGC and PGD low and then raising  $\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$  to  $V_{\text{IH}}^{\text{HH}}$  (high voltage). Once in this mode, the code memory, data EEPROM, ID locations and Configuration bits can be accessed and programmed in serial fashion. Figure 2-13 shows the exit sequence.

The sequence that enters the device into the Program/Verify mode places all unused I/Os in the high-impedance state.

**FIGURE 2-12: ENTERING HIGH-VOLTAGE PROGRAM/VERIFY MODE**



**FIGURE 2-13: EXITING HIGH-VOLTAGE PROGRAM/VERIFY MODE**

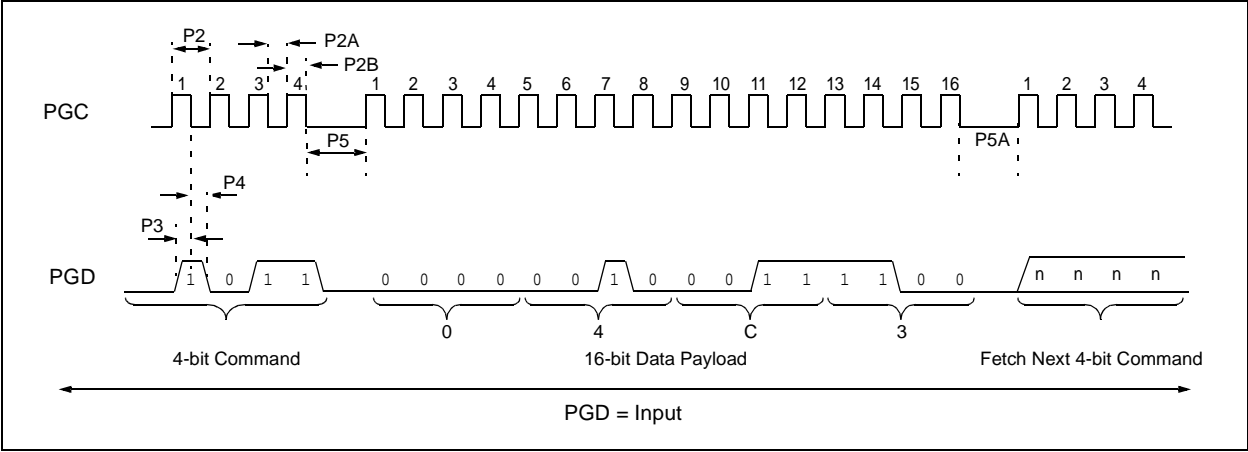


# PIC18F2XK20/4XK20

TABLE 2-7: SAMPLE COMMAND SEQUENCE

4-Bit Command	Data Payload	Core Instruction
1101	3C 40	Table Write, post-increment by 2

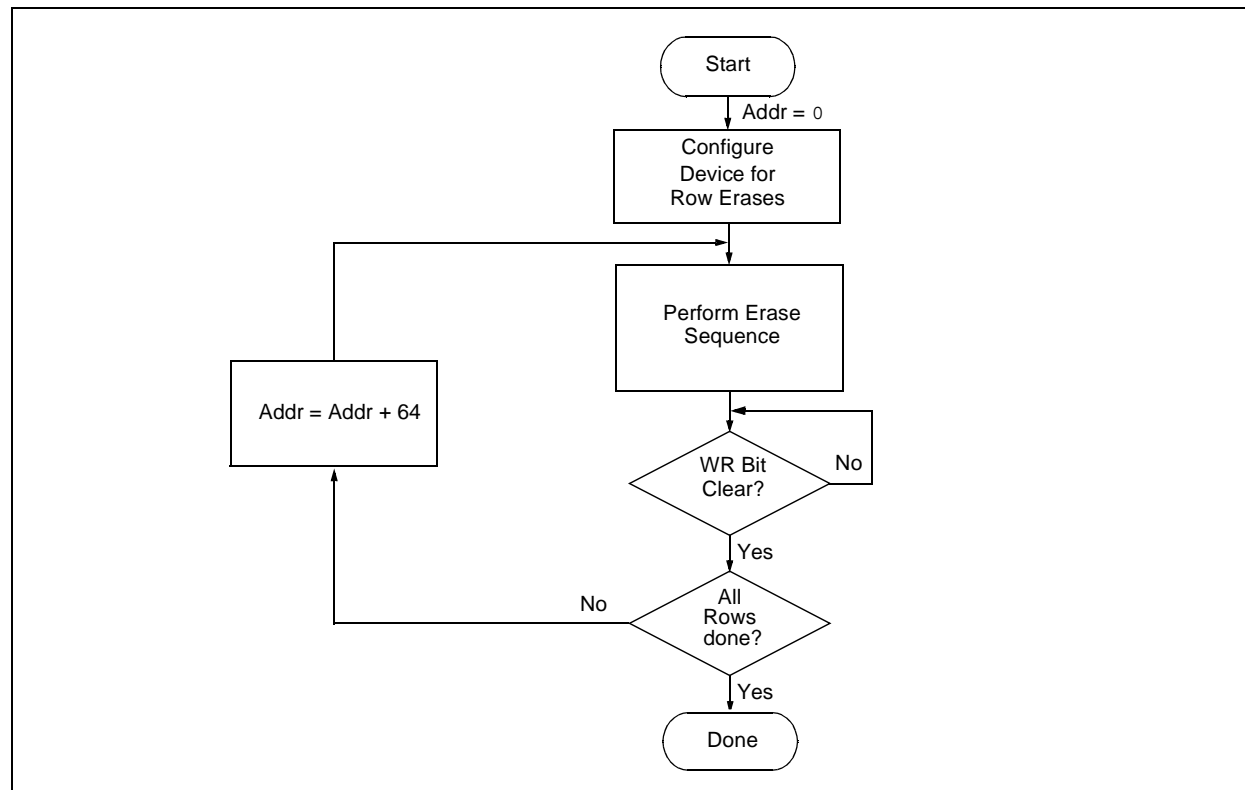
FIGURE 2-16: TABLE WRITE, POST-INCREMENT TIMING DIAGRAM (1101)





# PIC18F2XK20/4XK20

FIGURE 3-3: SINGLE ROW ERASE CODE MEMORY FLOW



## 3.2 Code Memory Programming

Programming code memory is accomplished by first loading data into the write buffer and then initiating a programming sequence. The write and erase buffer sizes shown in Table 3-4 can be mapped to any location of the same size beginning at 000000h. The actual memory write sequence takes the contents of this buffer and programs the proper amount of code memory that contains the Table Pointer.

The programming duration is externally timed and is controlled by PGC. After a Start Programming command is issued (4-bit command, '1111'), a NOP is issued, where the 4th PGC is held high for the duration of the programming time, P9.

After PGC is brought low, the programming sequence is terminated. PGC must be held low for the time specified by parameter P10 to allow high-voltage discharge of the memory array.

The code sequence to program a PIC18F2XK20/4XK20 device is shown in Table 3-5. The flowchart shown in Figure 3-4 depicts the logic necessary to completely write a PIC18F2XK20/4XK20 device. The timing diagram that details the Start Programming command and parameters P9 and P10 is shown in Figure 3-5.

**Note:** The TBLPTR register must point to the same region when initiating the programming sequence as it did when the write buffers were loaded.

**TABLE 3-4: WRITE AND ERASE BUFFER SIZES**

Devices (Arranged by Family)	Write Buffer Size (bytes)	Erase Size (bytes)
PIC18F26K20, PIC18F46K20	64	64
PIC18F24K20, PIC18F25K20, PIC18F44K20, PIC18F45K20	32	64
PIC18F23K20, PIC18F43K20	16	64

**TABLE 3-5: WRITE CODE MEMORY CODE SEQUENCE**

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
0000	84 A6	BSF EECON1, WREN
Step 2: Point to row to write.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[15:8]>	MOVLW <Addr[15:8]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 3: Load write buffer. Repeat for all but the last two bytes.		
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
Step 4: Load write buffer for last two bytes and start programming.		
1111	<MSB><LSB>	Write 2 bytes and start programming.
0000	00 00	NOP - hold PGC high for time P9 and low for time P10.
To continue writing data, repeat steps 2 through 4, where the Address Pointer is incremented by 2 at each iteration of the loop.		

# PIC18F2XK20/4XK20

FIGURE 3-4: PROGRAM CODE MEMORY FLOW

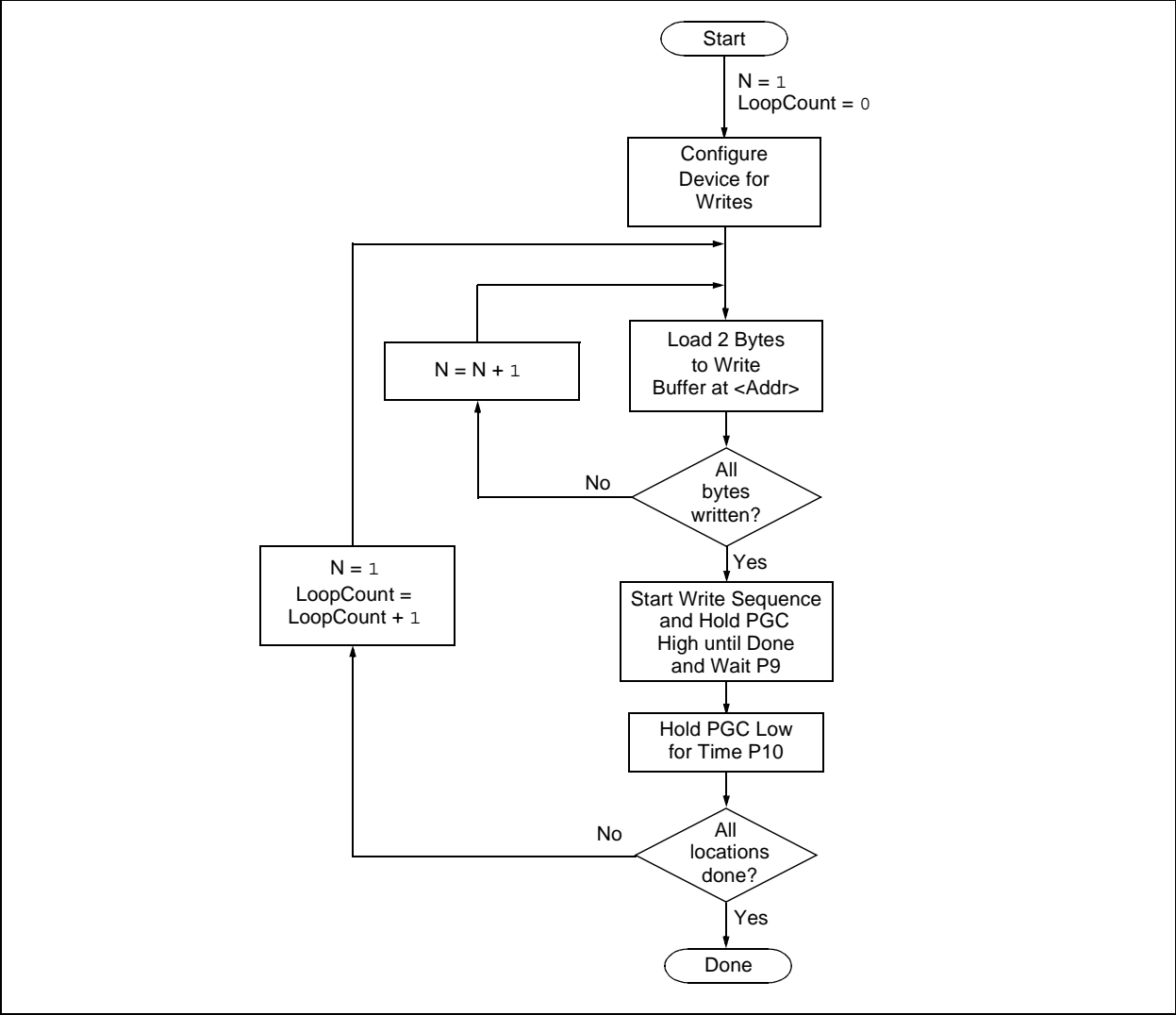
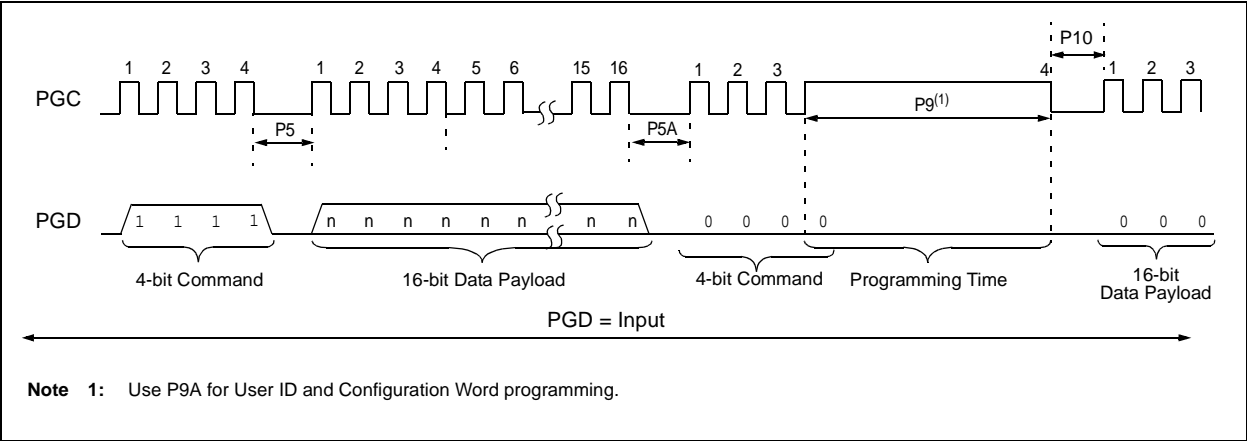


FIGURE 3-5: TABLE WRITE AND START PROGRAMMING INSTRUCTION TIMING DIAGRAM (1111)



## 3.2.1 MODIFYING CODE MEMORY

The previous programming example assumed that the device has been Bulk Erased prior to programming (see **Section 3.1.1 “High-Voltage ICSP Bulk Erase”**). It may be the case, however, that the user wishes to modify only a section of an already programmed device.

The appropriate number of bytes required for the erase buffer must be read out of code memory (as described in **Section 4.2 “Verify Code Memory and ID Locations”**) and buffered. Modifications can be made on this buffer. Then, the block of code memory that was read out must be erased and rewritten with the modified data.

The WREN bit must be set if the WR bit in EECON1 is used to initiate a write sequence.

**TABLE 3-6: MODIFYING CODE MEMORY**

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to code memory.		
0000	8E A6	BSF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 2: Read code memory into buffer ( <b>Section 4.1 “Read Code Memory, ID Locations and Configuration Bits”</b> ).		
Step 3: Set the Table Pointer for the block to be erased.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[8:15]>	MOVLW <Addr[8:15]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 4: Enable memory writes and setup an erase.		
0000	84 A6	BSF EECON1, WREN
0000	88 A6	BSF EECON1, FREE
Step 5: Initiate erase.		
0000	88 A6	BSF EECON1, FREE
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP
0000	00 00	NOP Erase starts on the 4th clock of this instruction
Step 6: Poll WR bit. Repeat until bit is clear.		
0000	50 A6	MOVF EECON1, W, 0
0000	6E F5	MOVWF TABLAT
0000	00 00	NOP
0000	<MSB><LSB>	Shift out data <sup>(1)</sup>
Step 7: Load write buffer. The correct bytes will be selected based on the Table Pointer.		
0000	0E <Addr[21:16]>	MOVLW <Addr[21:16]>
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[8:15]>	MOVLW <Addr[8:15]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
1101	<MSB><LSB>	Write 2 bytes and post-increment address by 2.
.	.	
.	.	Repeat as many times as necessary to fill the write buffer
.	.	Write 2 bytes and start programming.
1111	<MSB><LSB>	NOP - hold PGC high for time P9 and low for time P10.
0000	00 00	
To continue modifying data, repeat Steps 2 through 6, where the Address Pointer is incremented by the appropriate number of bytes (see Table 3-4) at each iteration of the loop. The write cycle must be repeated enough times to completely rewrite the contents of the erase buffer.		
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN

**TABLE 3-7: PROGRAMMING DATA MEMORY**

4-bit Command	Data Payload	Core Instruction
Step 1: Direct access to data EEPROM.		
0000	9E A6	BCF EECON1, EEPGD
0000	9C A6	BCF EECON1, CFGS
Step 2: Set the data EEPROM Address Pointer.		
0000	0E <Addr>	MOVLW <Addr>
0000	6E A9	MOVWF EEADR
0000	0E <AddrH>	MOVLW <AddrH>
0000	6E AA	MOVWF EEADRH
Step 3: Load the data to be written.		
0000	0E <Data>	MOVLW <Data>
0000	6E A8	MOVWF EEDATA
Step 4: Enable memory writes.		
0000	84 A6	BSF EECON1, WREN
Step 5: Initiate write.		
0000	82 A6	BSF EECON1, WR
0000	00 00	NOP
0000	00 00	NOP ;write starts on 4th clock of this instruction
Step 6: Poll WR bit, repeat until the bit is clear.		
0000	50 A6	MOVF EECON1, W, 0
0000	6E F5	MOVWF TABLAT
0000	00 00	NOP
0010	<MSB><LSB>	Shift out data <sup>(1)</sup>
Step 7: Hold PGC low for time P10.		
Step 8: Disable writes.		
0000	94 A6	BCF EECON1, WREN
Repeat steps 2 through 8 to write more data.		

**Note 1:** See Figure 4-4 for details on shift out data timing.

# PIC18F2XK20/4XK20

## 4.0 READING THE DEVICE

### 4.1 Read Code Memory, ID Locations and Configuration Bits

Code memory is accessed one byte at a time via the 4-bit command, '1001' (table read, post-increment). The contents of memory pointed to by the Table Pointer (TBLPTRU:TBLPTRH:TBLPTRL) are serially output on PGD.

The 4-bit command is shifted in LSb first. The read is executed during the next 8 clocks, then shifted out on PGD during the last 8 clocks, LSb to MSb. A delay of P6 must be introduced after the falling edge of the 8th

PGC of the operand to allow PGD to transition from an input to an output. During this time, PGC must be held low (see Figure 4-1). This operation also increments the Table Pointer by one, pointing to the next byte in code memory for the next read.

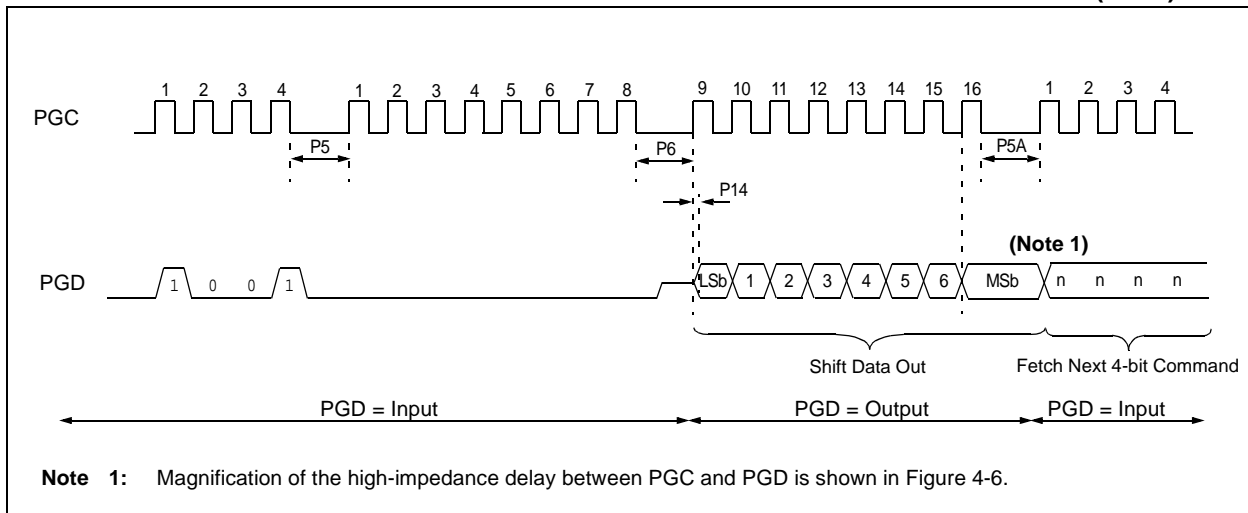
This technique will work to read any memory in the 000000h to 3FFFFFFh address space, so it also applies to the reading of the ID and Configuration registers.

**Note:** When table read protection is enabled, the first read access to a protected block should be discarded and the read repeated to retrieve valid data. Subsequent reads of the same block can be performed normally.

**TABLE 4-1: READ CODE MEMORY SEQUENCE**

4-bit Command	Data Payload	Core Instruction
Step 1: Set Table Pointer		
0000	0E <Addr[21:16]>	MOVLW Addr[21:16]
0000	6E F8	MOVWF TBLPTRU
0000	0E <Addr[15:8]>	MOVLW <Addr[15:8]>
0000	6E F7	MOVWF TBLPTRH
0000	0E <Addr[7:0]>	MOVLW <Addr[7:0]>
0000	6E F6	MOVWF TBLPTRL
Step 2: Read memory and then shift out on PGD, LSb to MSb		
1001	00 00	TBLRD *+

**FIGURE 4-1: TABLE READ POST-INCREMENT INSTRUCTION TIMING DIAGRAM (1001)**



# PIC18F2XK20/4XK20

## 5.0 CONFIGURATION WORD

The PIC18F2XK20/4XK20 devices have several Configuration Words. These bits can be set or cleared to select various device configurations. All other memory areas should be programmed and verified prior to setting Configuration Words. These bits may be read out normally, even after read or code protection. See Table 5-1 for a list of Configuration bits and device IDs and Table 5-3 for the Configuration bit descriptions.

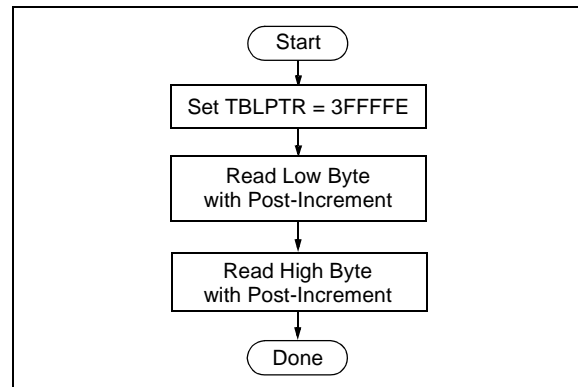
### 5.1 User ID Locations

A user may store identification information (ID) in eight ID locations mapped in 200000h:200007h. It is recommended that the Most Significant nibble of each ID be Fh. In doing so, if the user code inadvertently tries to execute from the ID space, the ID data will execute as a NOP.

## 5.2 Device ID Word

The device ID word for the PIC18F2XK20/4XK20 devices is located at 3FFFFEh:3FFFFFh. These bits may be used by the programmer to identify what device type is being programmed and read out normally, even after code or read protection. See Table 5-2 for a complete list of device ID values.

**FIGURE 5-1: READ DEVICE ID WORD FLOW**



**TABLE 5-1: CONFIGURATION BITS AND DEVICE IDs**

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	00-- 0111
300002h CONFIG2L	—	—	—	BORV1	BORV0	BOREN1	BOREN0	PWRTEN	---1 1111
300003h CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	---1 1111
300005h CONFIG3H	MCLRE	—	—	—	HFOFST	LPT1OSC	PBADEN	CCP2MX	1--- 1011
300006h CONFIG4L	DEBUG	XINST	—	—	—	LVP	—	STVREN	10-- -1-1
300008h CONFIG5L	—	—	—	—	CP3 <sup>(1)</sup>	CP2 <sup>(1)</sup>	CP1	CP0	---- 1111
300009h CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah CONFIG6L	—	—	—	—	WRT3 <sup>(1)</sup>	WRT2 <sup>(1)</sup>	WRT1	WRT0	---- 1111
30000Bh CONFIG6H	WRD	WRB	WRTC	—	—	—	—	—	111- ----
30000Ch CONFIG7L	—	—	—	—	EBTR3 <sup>(1)</sup>	EBTR2 <sup>(1)</sup>	EBTR1	EBTR0	---- 1111
30000Dh CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh DEVID1 <sup>(2)</sup>	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	See Table 5-2
3FFFFFh DEVID2 <sup>(2)</sup>	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	See Table 5-2

**Legend:** x = unknown, u = unchanged, — = unimplemented. Shaded cells are unimplemented, read as '0'.

**Note 1:** These bits are only implemented on specific devices. Refer to **Section 2.3 “Memory Maps”** to determine which bits apply based on available memory.

**2:** DEVID registers are read-only and cannot be programmed by the user.

**TABLE 5-2:     DEVICE ID VALUE**

Device	Device ID Value	
	DEVID2	DEVID1
PIC18F23K20	20h	111x xxxx
PIC18F24K20	20h	101x xxxx
PIC18F25K20	20h	011x xxxx
PIC18F26K20	20h	001x xxxx
PIC18F43K20	20h	110x xxxx
PIC18F44K20	20h	100x xxxx
PIC18F45K20	20h	010x xxxx
PIC18F46K20	20h	000x xxxx

**Note:**     The 'x's in DEVID1 contain the device revision code.



# PIC18F2XK20/4XK20

**TABLE 5-3: PIC18F2XK20/4XK20 BIT DESCRIPTIONS**

Bit Name	Configuration Words	Description
IESO	CONFIG1H	Internal External Switchover bit 1 = Internal External Switchover mode enabled 0 = Internal External Switchover mode disabled
FCMEN	CONFIG1H	Fail-Safe Clock Monitor Enable bit 1 = Fail-Safe Clock Monitor enabled 0 = Fail-Safe Clock Monitor disabled
FOSC<3:0>	CONFIG1H	Oscillator Selection bits 11xx = External RC oscillator, CLKOUT function on RA6 101x = External RC oscillator, CLKOUT function on RA6 1001 = HFINTOSC, CLKOUT function on RA6, port function on RA7 1000 = HFINTOSC, port function on RA6, port function on RA7 0111 = External RC oscillator, port function on RA6 0110 = HS oscillator, PLL enabled (clock frequency = 4 x FOSC1) 0101 = EC oscillator, port function on RA6 0100 = EC oscillator, CLKOUT function on RA6 0011 = External RC oscillator, CLKOUT function on RA6 0010 = HS oscillator 0001 = XT oscillator 0000 = LP oscillator
BORV<1:0>	CONFIG2L	Brown-out Reset Voltage bits 11 = VBOR set to 1.8V 10 = VBOR set to 2.2V 01 = VBOR set to 2.7V 00 = VBOR set to 3.0V
BOREN<1:0>	CONFIG2L	Brown-out Reset Enable bits 11 = Brown-out Reset enabled in hardware only (SBOREN is disabled) 10 = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled) 01 = Brown-out Reset enabled and controlled by software (SBOREN is enabled) 00 = Brown-out Reset disabled in hardware and software
PWRTEN	CONFIG2L	Power-up Timer Enable bit 1 = PWRT disabled 0 = PWRT enabled
WDPS<3:0>	CONFIG2H	Watchdog Timer Postscaler Select bits 1111 = 1:32,768 1110 = 1:16,384 1101 = 1:8,192 1100 = 1:4,096 1011 = 1:2,048 1010 = 1:1,024 1001 = 1:512 1000 = 1:256 0111 = 1:128 0110 = 1:64 0101 = 1:32 0100 = 1:16 0011 = 1:8 0010 = 1:4 0001 = 1:2 0000 = 1:1

# PIC18F2XK20/4XK20

**TABLE 5-3: PIC18F2XK20/4XK20 BIT DESCRIPTIONS (CONTINUED)**

Bit Name	Configuration Words	Description
CP3	CONFIG5L	Code Protection bits (Block 3 code memory area) 1 = Block 3 is not code-protected 0 = Block 3 is code-protected
CP2	CONFIG5L	Code Protection bits (Block 2 code memory area) 1 = Block 2 is not code-protected 0 = Block 2 is code-protected
CP1	CONFIG5L	Code Protection bits (Block 1 code memory area) 1 = Block 1 is not code-protected 0 = Block 1 is code-protected
CP0	CONFIG5L	Code Protection bits (Block 0 code memory area) 1 = Block 0 is not code-protected 0 = Block 0 is code-protected
CPD	CONFIG5H	Code Protection bits (Data EEPROM) 1 = Data EEPROM is not code-protected 0 = Data EEPROM is code-protected
CPB	CONFIG5H	Code Protection bits (Boot Block memory area) 1 = Boot Block is not code-protected 0 = Boot Block is code-protected
WRT3	CONFIG6L	Write Protection bits (Block 3 code memory area) 1 = Block 3 is not write-protected 0 = Block 3 is write-protected
WRT2	CONFIG6L	Write Protection bits (Block 2 code memory area) 1 = Block 2 is not write-protected 0 = Block 2 is write-protected
WRT1	CONFIG6L	Write Protection bits (Block 1 code memory area) 1 = Block 1 is not write-protected 0 = Block 1 is write-protected
WRT0	CONFIG6L	Write Protection bits (Block 0 code memory area) 1 = Block 0 is not write-protected 0 = Block 0 is write-protected
WRTD	CONFIG6H	Write Protection bit (Data EEPROM) 1 = Data EEPROM is not write-protected 0 = Data EEPROM is write-protected
WRTB	CONFIG6H	Write Protection bit (Boot Block memory area) 1 = Boot Block is not write-protected 0 = Boot Block is write-protected
WRTC	CONFIG6H	Write Protection bit (Configuration registers) 1 = Configuration registers are not write-protected 0 = Configuration registers are write-protected

# PIC18F2XK20/4XK20

**TABLE 5-4: CHECKSUM COMPUTATION (CONTINUED)**

Device	Code-Protect	Checksum	Blank Value	0xAA at 0 and Max Address
PIC18FX5K20	None	SUM[0000:07FF]+SUM[0800:1FFF]+SUM[2000:3FFF]+SUM[4000:5FFF]+SUM[6000:7FFF]+(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+(CONFIG7H & 40h)	8362h	82B8h
	Boot Block	SUM[0800:1FFF]+SUM[2000:3FFF]+SUM[4000:5FFF]+SUM[6000:7FFF]+(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+(CONFIG7H & 40h)+SUM_ID	8B35h	8AEAh
	Boot/Block 0/Block 1	SUM[4000:5FFF]+SUM[6000:7FFF]+(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+(CONFIG7H & 40h)+SUM_ID	C332h	C2E7h
	All	(CONFIG1L & 00h)+(CONFIG1H & CFh)+(CONFIG2L & 1Fh)+(CONFIG2H & 1F)+(CONFIG3L & 00h)+(CONFIG3H & 8Fh)+(CONFIG4L & C5h)+(CONFIG4H & 00h)+(CONFIG5L & 0Fh)+(CONFIG5H & C0h)+(CONFIG6L & 0Fh)+(CONFIG6H & E0h)+(CONFIG7L & 0Fh)+(CONFIG7H & 40h)+SUM_ID	0326h	0330h

**Legend:**

Item	Description
CONFIGx	Configuration Word
SUM[a:b]	Sum of locations, a to b inclusive
SUM_ID	Byte-wise sum of lower four bits of all customer ID locations
+	Addition
&	Bit-wise AND

## 6.0 AC/DC CHARACTERISTICS TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE (CONTINUED)

Standard Operating Conditions Operating Temperature: 25°C is recommended						
Param No.	Sym.	Characteristic	Min.	Max.	Units	Conditions
P12	THLD2	Input Data Hold Time from $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \uparrow$	2	—	$\mu\text{s}$	
P13	TSET2	$\text{VDD} \uparrow$ Setup Time to $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \uparrow$	100	—	ns	
P14	TVALID	Data Out Valid from PGC $\uparrow$	10	—	ns	
P15	TSET3	PGM $\uparrow$ Setup Time to $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \uparrow$	2	—	$\mu\text{s}$	
P16	TDLY8	Delay between Last PGC $\downarrow$ and $\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \downarrow$	0	—	s	
P17	THLD3	$\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \downarrow$ to $\text{VDD} \downarrow$	—	100	ns	
P18	THLD4	$\overline{\text{MCLR}}/\text{VPP}/\text{RE3} \downarrow$ to PGM $\downarrow$	0	—	s	
P19	THIZ	Delay from PGC $\uparrow$ to PGD High-Z	3	10	nS	
P20	TPPDP	Hold time after VPP changes	5	—	$\mu\text{s}$	

**Note 1:** Do not allow excess time when transitioning  $\overline{\text{MCLR}}$  between  $\text{VIL}$  and  $\text{VIHH}$ ; this can cause spurious program executions to occur. The maximum transition time is:  
 $1 \text{ Tcy} + \text{TPWRT}$  (if enabled) +  $1024 \text{ TOSC}$  (for LP, HS, HS/PLL and XT modes only) +  $2 \text{ ms}$  (for HS/PLL mode only) +  $1.5 \mu\text{s}$  (for EC mode only) where  $\text{Tcy}$  is the instruction cycle time,  $\text{TPWRT}$  is the Power-up Timer period and  $\text{TOSC}$  is the oscillator period. For specific values, refer to the Electrical Characteristics section of the device data sheet for the particular device.

---

**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, rPIC 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, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, PIC<sup>32</sup> logo, 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.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.



Printed on recycled paper.

*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.*