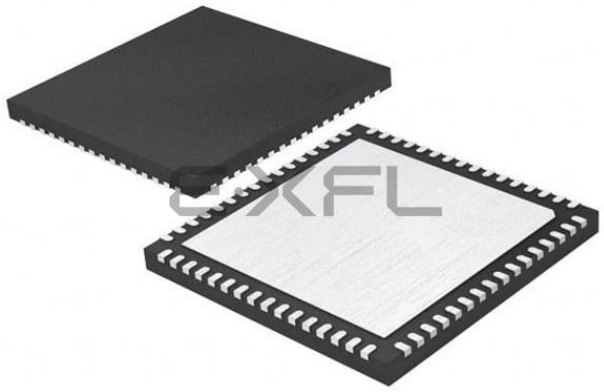


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, LVD, POR, PWM, WDT |
| Number of I/O | 53 |
| Program Memory Size | 128KB (64K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 16x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-VFQFN Exposed Pad |
| Supplier Device Package | 64-VQFN (9x9) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f67k22t-i-mrrsl |

**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.



**Trademarks**



ISBN: 978-1-61341-272-5

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

*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.*

**TABLE 1-3: PIC18F6XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name | Pin Number | Pin Type | Buffer Type | Description |
|---|---|---|---|---|
| | QFN/TQFP | | | |
| | | | | PORTD is a bidirectional I/O port. |
| RD0/PSP0/CTPLS | 58 | | | |
| RD0 | | I/O | ST | Digital I/O. |
| PSP0 | | I/O | TTL | Parallel Slave Port data. |
| CTPLS | | O | — | CTMU pulse generator output. |
| RD1/PSP1/T5CKI/T7G | 55 | | | |
| RD1 | | I/O | ST | Digital I/O. |
| PSP1 | | I/O | TTL | Parallel Slave Port. |
| T5CKI | | I | ST | Timer5 clock input. |
| T7G | | I | ST | Timer7 external clock gate input. |
| RD2/PSP2 | 54 | | | |
| RD2 | | I/O | ST | Digital I/O. |
| PSP2 | | O | TTL | Parallel Slave Port. |
| RD3/PSP3 | 53 | | | |
| RD3 | | I/O | ST | Digital I/O. |
| PSP3 | | I/O | TTL | Parallel Slave Port. |
| RD4/PSP4/SDO2 | 52 | | | |
| RD4 | | I/O | ST | Digital I/O. |
| PSP4 | | I/O | TTL | Parallel Slave Port. |
| SDO2 | | O | — | SPI data out. |
| RD5/PSP5/SDI2/SDA2 | 51 | | | |
| RD5 | | I/O | ST | Digital I/O. |
| PSP5 | | I/O | TTL | Parallel Slave Port. |
| SDI2 | | I | ST | SPI data in. |
| SDA2 | | I/O | $I^2C$ | $I^2C$™ data I/O. |
| RD6/PSP6/SCK2/SCL2 | 50 | | | |
| RD6 | | I/O | ST | Digital I/O. |
| PSP6 | | I/O | TTL | Parallel Slave Port. |
| SCK2 | | I/O | ST | Synchronous serial clock. |
| SCL2(4) | | I/O | $I^2C$ | Synchronous serial clock I/O for $I^2C$ mode. |
| RD7/PSP7/$\overline{SS2}$ | 49 | | | |
| RD7 | | I/O | ST | Digital I/O. |
| PSP7 | | I/O | TTL | Parallel Slave Port. |
| $\overline{SS2}$ | | I | TTL | SPI slave select input. |

**Legend:** TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to $V_{DD}$); $I^2C$ = $I^2C$™/SMBus

**Note 1:** Default assignment for ECCP2 when the CCP2MX Configuration bit is set.

**2:** Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.

**3:** Not available on PIC18F65K22 and PIC18F85K22 devices.

**4:** The CC6, CCP7, CCP8 and CCP9 pin placement depends on the setting of the ECCPMX Configuration bit (CONFIG3H<1>).

**TABLE 1-4: PIC18F8XK22 PINOUT I/O DESCRIPTIONS**

| Pin Name | Pin Number | Pin Type | Buffer Type | Description |
|---|---|---|---|---|
| | TQFP | | | |
| $\overline{MCLR}$/RG5 | 9 | | | Master Clear (input) or programming voltage (input). |
| RG5 | | I | ST | This pin is an active-low Reset to the device. |
| $\overline{MCLR}$ | | I | ST | General purpose, input only pin. |
| OSC1/CLKI/RA7 | 49 | | | Oscillator crystal or external clock input. |
| OSC1 | | I | CMOS | Oscillator crystal input. |
| CLKI | | I | CMOS | External clock source input. Always associated with pin function, OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.) |
| RA7 | | I/O | TTL | General purpose I/O pin. |
| OSC2/CLKO/RA6 | 50 | | | Oscillator crystal or clock output. |
| OSC2 | | O | — | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. |
| CLKO | | O | — | In certain oscillator modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| RA6 | | I/O | TTL | General purpose I/O pin. |

**Legend:** TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to $V_{DD}$); $I^2C$ = $I^2C$™/SMBus

**Note 1:** Default assignment for ECCP2 when the CCP2MX Configuration bit is set.

**2:** Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.

**3:** Not available on PIC18F65K22 and PIC18F85K22 devices.

**4:** PSP is available only in Microcontroller mode.

**5:** The CC6, CCP7, CCP8 and CCP9 pin placement depends on the setting of the ECCPMX Configuration bit (CONFIG3H<1>).

## 6.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit Program Counter that is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

The entire PIC18F87K22 family offers a range of on-chip Flash program memory sizes, from 32 Kbytes (up to 16,384 single-word instructions) to 128 Kbytes (65,536 single-word instructions).

- PIC18F65K22 and PIC18F85K22 – 32 Kbytes of Flash memory, storing up to 16,384 single-word instructions
- PIC18F66K22 and PIC18F86K22 – 64 Kbytes of Flash memory, storing up to 32,768 single-word instructions
- PIC18F67K22 and PIC18F87K22 – 128 Kbytes of Flash memory, storing up to 65,536 single-word instructions

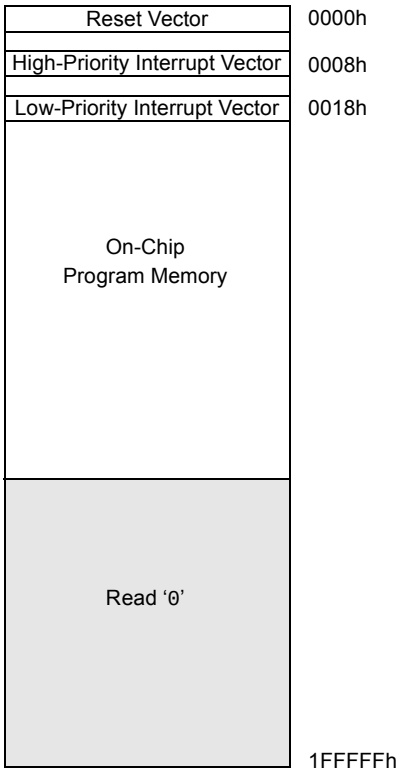
The program memory maps for individual family members are shown in Figure 6-1.

### 6.1.1 HARD MEMORY VECTORS

All PIC18 devices have a total of three hard-coded return vectors in their program memory space. The Reset vector address is the default value to which the Program Counter returns on all device Resets; it is located at 0000h.

PIC18 devices also have two interrupt vector addresses for handling high-priority and low-priority interrupts. The high-priority interrupt vector is located at 0008h and the low-priority interrupt vector is at 0018h. The locations of these vectors are shown, in relation to the program memory map, in Figure 6-2.

**FIGURE 6-2: HARD VECTOR FOR PIC18F87K22 FAMILY DEVICES**




**Legend:** (Top of Memory) represents upper boundary of on-chip program memory space (see Figure 6-1 for device-specific values). Shaded area represents unimplemented memory. Areas are not shown to scale.

### 6.1.2 PROGRAM COUNTER

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide and contained in three separate 8-bit registers.

The low byte, known as the PCL register, is both readable and writable. The high byte, or PCH register, contains the PC<15:8> bits and is not directly readable or writable. Updates to the PCH register are performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits; it is also not directly readable or writable. Updates to the PCU register are performed through the PCLATU register.

The contents of PCLATH and PCLATU are transferred to the Program Counter by any operation that writes PCL. Similarly, the upper two bytes of the Program Counter are transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 6.1.5.1 "Computed GOTO"**).

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the Least Significant bit of PCL is fixed to a value of '0'. The PC increments by two to address sequential instructions in the program memory.

The CALL, RCALL, GOTO and program branch instructions write to the Program Counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the Program Counter.

### 6.1.3 RETURN ADDRESS STACK

The return address stack enables execution of any combination of up to 31 program calls and interrupts. The PC is pushed onto the stack when a CALL or RCALL instruction is executed or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction. The value also is pulled off the stack on ADDULNK and SUBULNK instructions, if the extended instruction set is enabled. PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, STKPTR. The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the Top-of-Stack Special Function Registers. Data can also be pushed to, or popped from, the stack using these registers.

A CALL type instruction causes a push onto the stack. The Stack Pointer is first incremented and the location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction following the CALL). A RETURN type instruction causes a pop from the stack. The contents of the location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

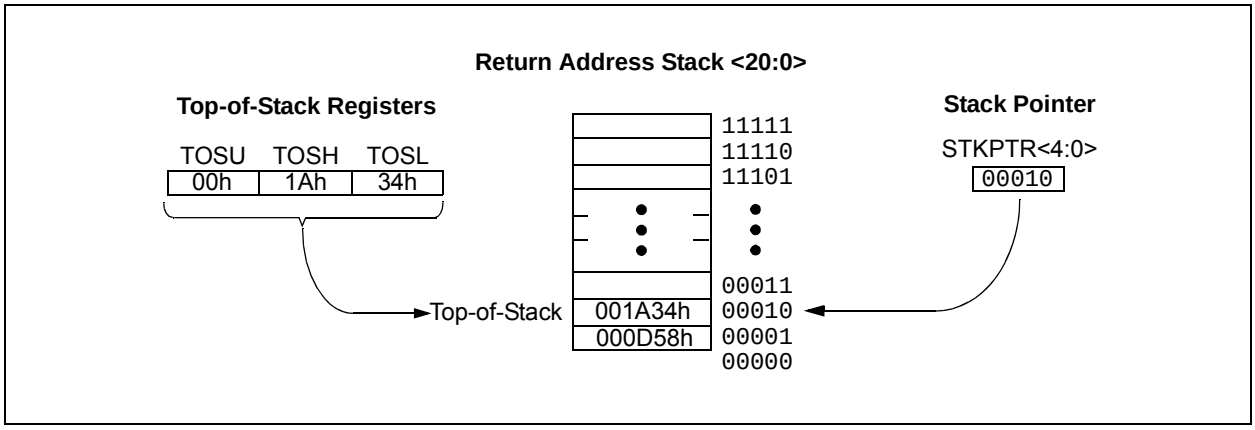
The Stack Pointer is initialized to '00000' after all Resets. There is no RAM associated with the location corresponding to a Stack Pointer value of '00000'; this is only a Reset value. Status bits indicate if the stack is full, has overflowed or has underflowed.

#### 6.1.3.1 Top-of-Stack Access

Only the top of the return address stack (TOS) is readable and writable. A set of three registers, TOSU:TOSH:TOSL, holds the contents of the stack location pointed to by the STKPTR register (Figure 6-3). This allows users to implement a software stack, if necessary. After a CALL, RCALL or interrupt (or ADDULNK and SUBULNK instructions, if the extended instruction set is enabled), the software can read the pushed value by reading the TOSU:TOSH:TOSL registers. These values can be placed on a user-defined software stack. At return time, the software can return these values to TOSU:TOSH:TOSL and do a return.

While accessing the stack, users must disable the Global Interrupt Enable bits to prevent inadvertent stack corruption.

**FIGURE 6-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS**



**REGISTER 7-1: EECON1: EEPROM CONTROL REGISTER 1**

| R/W-x | R/W-x | U-0 | R/W-0 | R/W-x | R/W-0 | R/S-0 | R/S-0 |
|---|---|---|---|---|---|---|---|
| EEPGD | CFGS | — | FREE | WRERR(1) | WREN | WR | RD |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | S = Settable bit | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit
- 1 = Access Flash program memory
- 0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
- 1 = Access Configuration registers
- 0 = Access Flash program or data EEPROM memory

bit 5 **Unimplemented:** Read as '0'

bit 4 **FREE:** Flash Row Erase Enable bit
- 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
- 0 = Perform write-only

bit 3 **WRERR:** Flash Program/Data EEPROM Error Flag bit(1)
- 1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation or an improper write attempt)
- 0 = The write operation completed

bit 2 **WREN:** Flash Program/Data EEPROM Write Enable bit
- 1 = Allows write cycles to Flash program/data EEPROM
- 0 = Inhibits write cycles to Flash program/data EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle (The operation is self-timed and the bit is cleared by hardware once the write is complete. The WR bit can only be set (not cleared) in software.)
- 0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
- 1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. The RD bit cannot be set when EEPGD = 1 or CFGS = 1.)
- 0 = Does not initiate an EEPROM read

**Note 1:** When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows tracing of the error condition.

## 8.8 Operation in Power-Managed Modes

In alternate, power-managed Run modes, the external bus continues to operate normally. If a clock source with a lower speed is selected, bus operations will run at that speed. In these cases, excessive access times for the external memory may result if Wait states have been enabled and added to external memory operations. If operations in a lower power Run mode are anticipated, users should provide in their applications for adjusting memory access times at the lower clock speeds.

In Sleep and Idle modes, the microcontroller core does not need to access data; bus operations are suspended. The state of the external bus is frozen, with the address/data pins, and most of the control pins, holding at the same state they were in when the mode was invoked. The only potential changes are to the $\overline{CE}$, $\overline{LB}$ and $\overline{UB}$ pins, which are held at logic high.

**TABLE 8-3: REGISTERS ASSOCIATED WITH THE EXTERNAL MEMORY BUS**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| MEMCON(1) | EBDIS | — | WAIT1 | WAIT0 | — | — | WM1 | WM0 |
| PADCFG1 | RDPU | REPU | RJPU(1) | — | — | RTSECSEL1 | RTSECSEL0 | — |
| PMD1 | PSPMD | CTMUMD | RTCCMD | TMR4MD | TMR3MD | TMR2MD | TMR1MD | EMBMD |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used during External Memory Bus access.

**Note 1:** Unimplemented in 64-pin devices (PIC18F6XK22), read as '0'.

## 12.11 Parallel Slave Port

PORTD can function as an 8-bit-wide Parallel Slave Port (PSP), or microprocessor port, when control bit, PSPMODE (PSPCON<4>), is set. The port is asynchronously readable and writable by the external world through the $\overline{RD}$ control input pin (RE0/P2D/$\overline{RD}$/AD8) and $\overline{WR}$ control input pin (RE1/P2C/$\overline{WR}$/AD9).

| **Note:** | The Parallel Slave Port is available only in Microcontroller mode. |
|---|---|

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an eight-bit latch.

Setting bit, PSPMODE, enables port pin, RE0/P2D/$\overline{RD}$/AD8, to be the $\overline{RD}$ input, RE1/P2C/$\overline{WR}$/AD9 to be the $\overline{WR}$ input and RE2/P2B/CCP10/$\overline{CS}$/AD10 to be the $\overline{CS}$ (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (= 111).

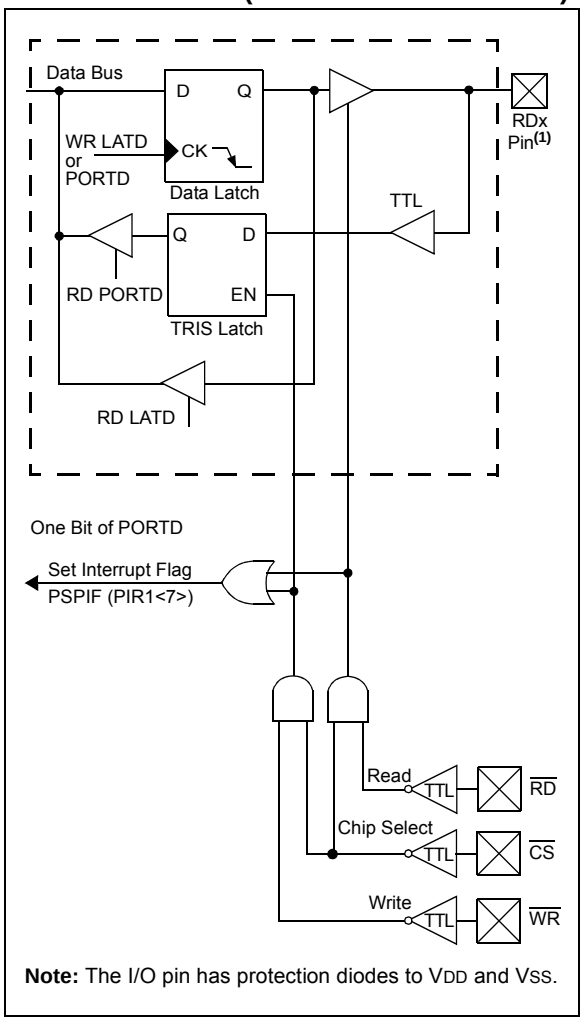
A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low and ends when either are detected high. The PSPIF and IBF flag bits (PIR1<7> and PSPCON<7>, respectively) are set when the write ends.

A read from the PSP occurs when both the $\overline{CS}$ and $\overline{RD}$ lines are first detected low. The data in PORTD is read out and the OBF bit (PSPCON<6>) is set. If the user writes new data to PORTD to set OBF, the data is immediately read out, but the OBF bit is not set.

When either the $\overline{CS}$ or $\overline{RD}$ line is detected high, the PORTD pins return to the input state and the PSPIF bit is set. User applications should wait for PSPIF to be set before servicing the PSP. When this happens, the IBF and OBF bits can be polled and the appropriate action taken.

The timing for the control signals in Write and Read modes is shown in Figure 12-4 and Figure 12-5, respectively.

**FIGURE 12-3: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**




**Note:** The I/O pin has protection diodes to VDD and VSS.

### 18.1.4 RTCEN BIT WRITE

RTCWREN (RTCCFG<5>) must be set before a write to RTCEN can take place. Any write to the RTCEN bit, while RTCWREN = 0, will be ignored.

Like the RTCEN bit, the RTCVALH and RTCVALL registers can only be written to when RTCWREN = 1. A write to these registers, while RTCWREN = 0, will be ignored.

## 18.2 Operation

### 18.2.1 REGISTER INTERFACE

The register interface for the RTCC and alarm values is implemented using the Binary Coded Decimal (BCD) format. This simplifies the firmware when using the module, as each of the digits is contained within its own 4-bit value (see Figure 18-2 and Figure 18-3).

**FIGURE 18-2: TIMER DIGIT FORMAT**

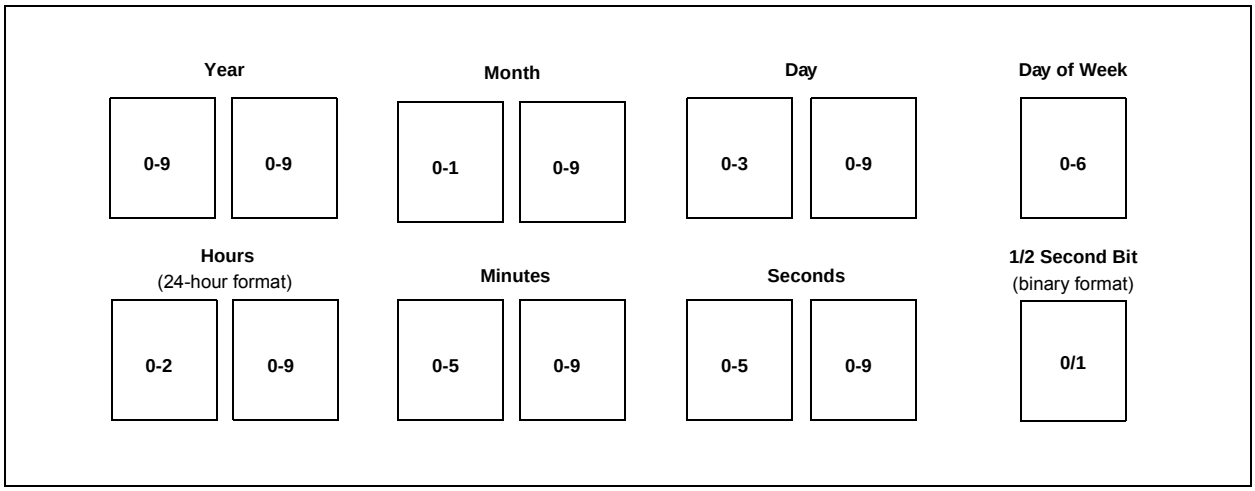



**FIGURE 18-3: ALARM DIGIT FORMAT**

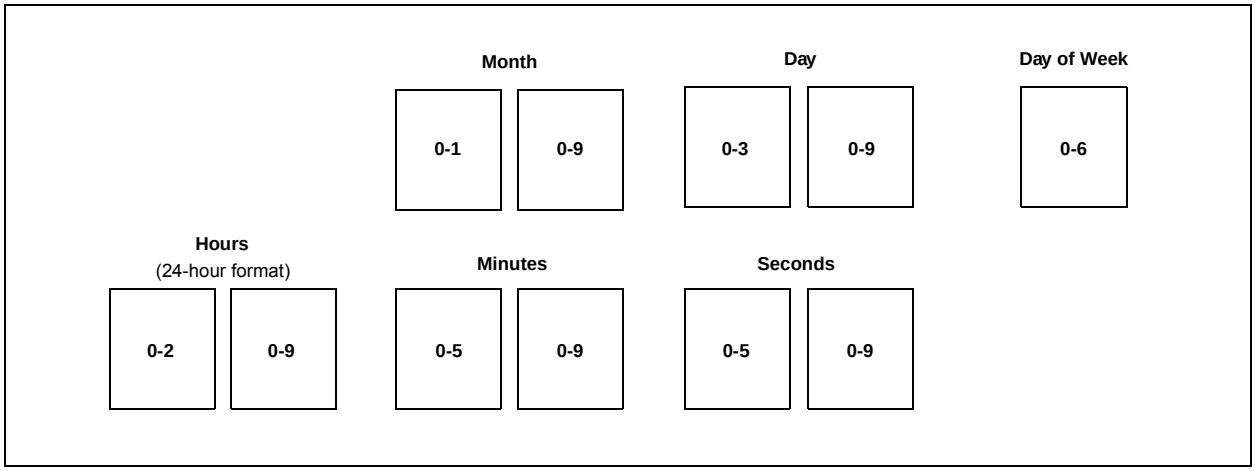


**TABLE 19-7: REGISTERS ASSOCIATED WITH PWM AND TIMERS (CONTINUED)**

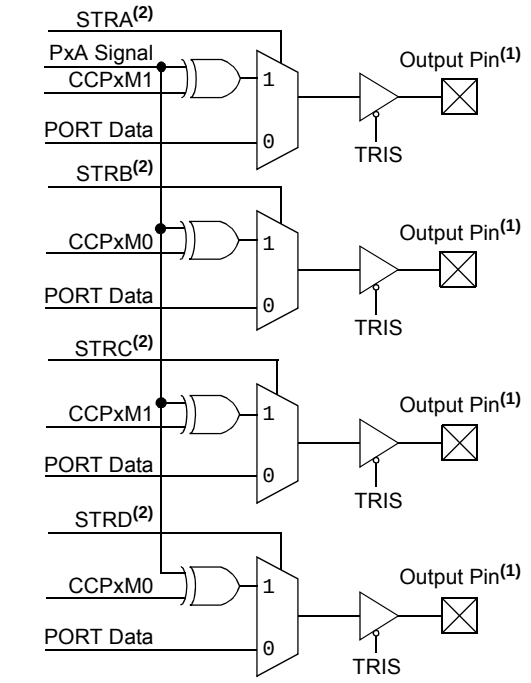
| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| TRISH(2) | TRISH7 | TRISH6 | TRISH5 | TRISH4 | TRISH3 | TRISH2 | TRISH1 | TRISH0 |
| TMR2 | Timer2 Register | | | | | | | |
| TMR4 | Timer4 Register | | | | | | | |
| TMR6 | Timer6 Register | | | | | | | |
| TMR8 | Timer8 Register | | | | | | | |
| PR2 | Timer2 Period Register | | | | | | | |
| PR4 | Timer4 Period Register | | | | | | | |
| PR6 | Timer6 Period Register | | | | | | | |
| PR8 | Timer8 Period Register | | | | | | | |
| T2CON | — | T2OUTPS3 | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| T4CON | — | T4OUTPS3 | T4OUTPS2 | T4OUTPS1 | T4OUTPS0 | TMR4ON | T4CKPS1 | T4CKPS0 |
| T6CON | — | T6OUTPS3 | T6OUTPS2 | T6OUTPS1 | T6OUTPS0 | TMR6ON | T6CKPS1 | T6CKPS0 |
| T8CON | — | T8OUTPS3 | T8OUTPS2 | T8OUTPS1 | T8OUTPS0 | TMR8ON | T8CKPS1 | T8CKPS0 |
| CCPR4L | Capture/Compare/PWM Register 4 Low Byte | | | | | | | |
| CCPR4H | Capture/Compare/PWM Register 4 High Byte | | | | | | | |
| CCPR5L | Capture/Compare/PWM Register 5 Low Byte | | | | | | | |
| CCPR5H | Capture/Compare/PWM Register 5 High Byte | | | | | | | |
| CCPR6L | Capture/Compare/PWM Register 6 Low Byte | | | | | | | |
| CCPR6H | Capture/Compare/PWM Register 6 High Byte | | | | | | | |
| CCPR7L | Capture/Compare/PWM Register 7 Low Byte | | | | | | | |
| CCPR7H | Capture/Compare/PWM Register 7 High Byte | | | | | | | |
| CCPR8L | Capture/Compare/PWM Register 8 Low Byte | | | | | | | |
| CCPR8H | Capture/Compare/PWM Register 8 High Byte | | | | | | | |
| CCPR9L(1) | Capture/Compare/PWM Register 9 Low Byte | | | | | | | |
| CCPR9H(1) | Capture/Compare/PWM Register 9 High Byte | | | | | | | |
| CCPR10L(1) | Capture/Compare/PWM Register 10 Low Byte | | | | | | | |
| CCPR10H(1) | Capture/Compare/PWM Register 10 High Byte | | | | | | | |
| CCP4CON | — | — | DC4B1 | DC4B0 | CCP4M3 | CCP4M2 | CCP4M1 | CCP4M0 |
| CCP5CON | — | — | DC5B1 | DC5B0 | CCP5M3 | CCP5M2 | CCP5M1 | CCP5M0 |
| CCP6CON | — | — | DC6B1 | DC6B0 | CCP6M3 | CCP6M2 | CCP6M1 | CCP6M0 |
| CCP7CON | — | — | DC7B1 | DC7B0 | CCP7M3 | CCP7M2 | CCP7M1 | CCP7M0 |
| CCP8CON | — | — | DC8B1 | DC8B0 | CCP8M3 | CCP8M2 | CCP8M1 | CCP8M0 |
| CCP9CON(1) | — | — | DC9B1 | DC9B0 | CCP9M3 | CCP9M2 | CCP9M1 | CCP9M0 |
| CCP10CON(1) | — | — | DC10B1 | DC10B0 | CCP10M3 | CCP10M2 | CCP10M1 | CCP10M0 |
| CCPTMRS1 | C7TSEL1 | C7TSEL0 | — | C6TSEL0 | — | C5TSEL0 | C4TSEL1 | C4TSEL0 |
| CCPTMRS2 | — | — | — | C10TSEL0(1) | — | C9TSEL0(1) | C8TSEL1 | C8TSEL0 |
| PMD3 | CCP10MD(1) | CCP9MD(1) | CCP8MD | CCP7MD | CCP6MD | CCP5MD | CCP4MD | TMR12MD(1) |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PWM or Timer2/4/6/8.

**Note 1:** Unimplemented on devices with a program memory of 32 Kbytes (PIC18F65K22 and PIC18F85K22).

**2:** Unimplemented on 64-pin devices (PIC18F6XK22), read as '0'.

**FIGURE 20-16: SIMPLIFIED STEERING BLOCK DIAGRAM**




**Note 1:** Port outputs are configured as displayed when the CCPxCON register bits, PxM<1:0> = 00 and CCP1M<3:2> = 11.

**2:** Single PWM output requires setting at least one of the STRx bits.

### 20.4.7.1 Steering Synchronization

The STRSYNC bit of the PSTRxCON register gives the user two choices for when the steering event will happen. When the STRSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTRxCON register. In this case, the output signal at the Px<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STRSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 20-17 and 20-18 illustrate the timing diagrams of the PWM steering depending on the STRSYNC setting.

**FIGURE 20-17: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRSYNC = 0)**

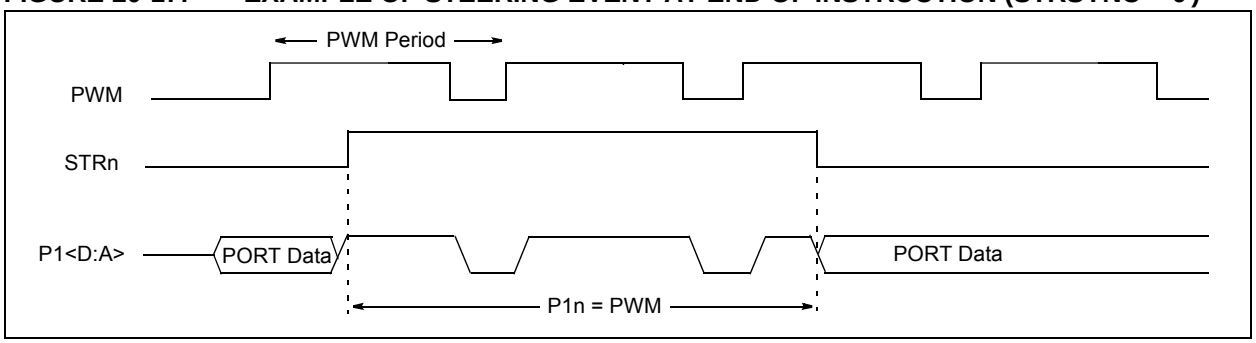
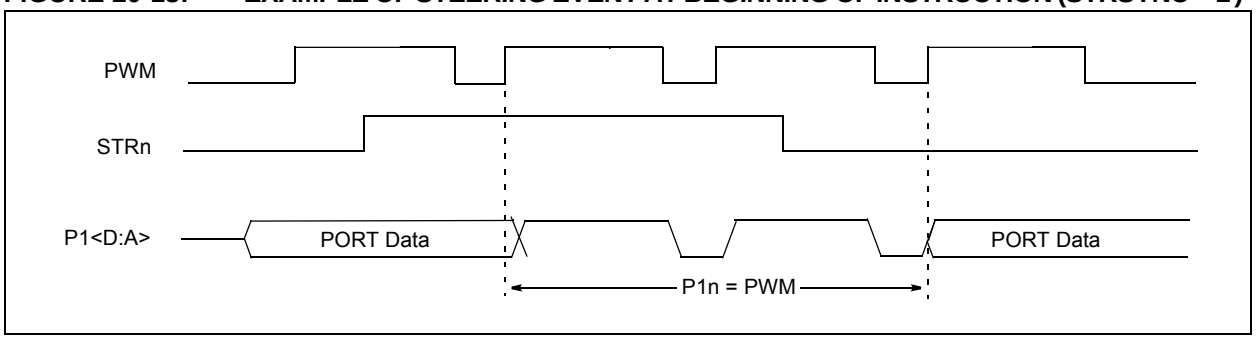



**FIGURE 20-18: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRSYNC = 1)**



### 20.4.8 OPERATION IN POWER-MANAGED MODES

In Sleep mode, all clock sources are disabled. Timer2/4/6/8 will not increment and the state of the module will not change. If the ECCPx pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency from HF-INTOSC and the postscaler may not be immediately stable.

In PRI_IDLE mode, the primary clock will continue to clock the ECCPx module without change.

#### 20.4.8.1 Operation with Fail-Safe Clock Monitor (FSCM)

If the Fail-Safe Clock Monitor (FSCM) is enabled, a clock failure will force the device into the power-managed RC_RUN mode and the OSCFIF bit of the PIR2 register will be set. The ECCPx will then be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

### 20.4.9 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the ECCP registers to their Reset states. This forces the ECCP module to reset to a state compatible with previous, non-Enhanced CCP modules used on other PIC18 and PIC16 devices.

**TABLE 20-4: REGISTERS ASSOCIATED WITH ECCP1/2/3 MODULE AND TIMER1/2/3/4/6/8/10/12**

| File Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF |
| RCON | IPEN | SBOREN | $\overline{CM}$ | $\overline{RI}$ | $\overline{TO}$ | $\overline{PD}$ | $\overline{POR}$ | $\overline{BOR}$ |
| PIR3 | TMR5GIF | — | RC2IF | TX2IF | CTMUIF | CCP2IF | CCP1IF | RTCCIF |
| PIR4 | CCP10IF(1) | CCP9IF(1) | CCP8IF | CCP7IF | CCP6IF | CCP5IF | CCP4IF | CCP3IF |
| PIE3 | TMR5GIE | — | RC2IE | TX2IE | CTMUIE | CCP2IE | CCP1IE | RTCCIE |
| PIE4 | CCP10IE(1) | CCP9IE(1) | CCP8IE | CCP7IE | CCP6IE | CCP5IE | CCP4IE | CCP3IE |
| IPR3 | TMR5GIP | — | RC2IP | TX2IP | CTMUIP | CCP2IP | CCP1IP | RTCCIP |
| IPR4 | CCP10IP(1) | CCP9IP(1) | CCP8IP | CCP7IP | CCP6IP | CCP5IP | CCP4IP | CCP3IP |
| TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| TRISE | TRISE7 | TRISE6 | TRISE5 | TRISE4 | TRISE3 | TRISE2 | TRISE1 | TRISE0 |
| TRISH(2) | TRISH7 | TRISH6 | TRISH5 | TRISH4 | TRISH3 | TRISH2 | TRISH1 | TRISH0 |
| TMR1H | Timer1 Register High Byte | | | | | | | |
| TMR1L | Timer1 Register Low Byte | | | | | | | |
| TMR2 | Timer2 Register | | | | | | | |
| TMR3H | Timer3 Register High Byte | | | | | | | |
| TMR3L | Timer3 Register Low Byte | | | | | | | |
| TMR4 | Timer4 Register | | | | | | | |
| TMR6 | Timer6 Register | | | | | | | |
| TMR8 | Timer8 Register | | | | | | | |
| TMR10(1) | TMR10 Register | | | | | | | |
| TMR12(1) | TMR10 Register | | | | | | | |
| PR2 | Timer2 Period Register | | | | | | | |
| PR4 | Timer4 Period Register | | | | | | | |
| PR6 | Timer6 Period Register | | | | | | | |
| PR8 | Timer8 Period Register | | | | | | | |
| PR10(1) | Timer10 Period Register | | | | | | | |
| PR12(1) | Timer12 Period Register | | | | | | | |

**Note 1:** Unimplemented on devices with a program memory of 32 Kbytes (PIC18F65K22 and PIC18F85K22).
**2:** Unimplemented on 64-pin devices (PIC18F6XK22), read as '0'.

**REGISTER 22-1: TXSTAx: TRANSMIT STATUS AND CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CSRC | TX9 | TXEN(1) | SYNC | SENDB | BRGH | TRMT | TX9D |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care.
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-Bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit is enabled
0 = Transmit is disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission has completed
Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR is empty
0 = TSR is full

bit 0 **TX9D:** 9th bit of Transmit Data
Can be address/data bit or a parity bit.

**Note 1:** SREN/CREN overrides TXEN in Sync mode.

## 25.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 32-tap resistor ladder network that provides a selectable reference voltage. Although its primary purpose is to provide a reference for the analog comparators, it may also be used independently of them.

A block diagram of the module is shown in Figure 25-1. The resistor ladder is segmented to provide a range of CVREF values and has a power-down function to conserve power when the reference is not being used. The module's supply reference can be provided from either device VDD/VSS or an external voltage reference.

### 25.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCON register (Register 25-1). The comparator voltage reference provides a range of output voltage with 32 levels.

The CVR<4:0> selection bits (CVRCON<4:0>) offer a range of output voltages. Equation 25-1 shows the how the comparator voltage reference is computed.

**EQUATION 25-1:**

If CVRSS = 1:

$$CVREF = (VREF-) + (CVR<4:0>/32) \bullet (VREF+ - VREF-)$$

If CVRSS = 0:

$$CVREF = (AVSS) + (CVR<4:0>/32) \bullet (AVDD - AVSS)$$

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF- that are multiplexed with RA3 and RA2. The voltage source is selected by the CVRSS bit (CVRCON<5>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output (see Table 31-2 in **Section 31.0 "Electrical Characteristics"**).

**REGISTER 25-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CVREN | CVROE | CVRSS | CVR4 | CVR3 | CVR2 | CVR1 | CVR0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7 **CVREN:** Comparator Voltage Reference Enable bit
1 = CVREF circuit is powered on
0 = CVREF circuit is powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit
1 = CVREF voltage level is output on CVREF pin
0 = CVREF voltage level is disconnected from CVREF pin

bit 5 **CVRSS:** Comparator VREF Source Selection bit
1 = Comparator reference source, CVRSRC = VREF+ – VREF-
0 = Comparator reference source, CVRSRC = AVDD – AVSS

bit 4-0 **CVR<4:0>:** Comparator VREF Value Selection 0 ≤ CVR<4:0> ≤ 31 bits
When CVRSS = 1:
CVREF = (VREF-) + (CVR<4:0>/32) • (VREF+ – VREF-)
When CVRSS = 0:
CVREF = (AVSS) + (CVR<4:0>/32) • (AVDD – AVSS)

**EXAMPLE 27-3: CAPACITANCE CALIBRATION ROUTINE**

```
#include "p18cxxx.h"

#define COUNT 25                        //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5                 //time in uS
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023                    //for unsigned conversion 10 sig bits
#define ADREF 3.3                       //Vdd connected to A/D Vr+
#define RCAL .027                       //R value is 4200000 (4.2M)
                                        //scaled so that result is in
                                        //1/100th of uA

int main(void)
{
    int i;
    int j = 0;                          //index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

//assume CTMU and A/D have been setup correctly
//see Example 25-1 for CTMU & A/D setup
setup();

CTMUCONHbits.CTMUEN = 1;                //Enable the CTMU
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1;       //drain charge on the circuit
        DELAY;                          //wait 125us
        CTMUCONHbits.IDISSEN = 0;       //end drain of circuit

        CTMUCONLbits.EDG1STAT = 1;      //Begin charging the circuit
                                        //using CTMU current source
        DELAY;                          //wait for 125us
        CTMUCONLbits.EDG1STAT = 0;      //Stop charging circuit

        PIR1bits.ADIF = 0;              //make sure A/D Int not set
        ADCON0bits.GO=1;                //and begin A/D conv.
        while(!PIR1bits.ADIF);          //Wait for A/D convert complete


        Vread = ADRES;                  //Get the value from the A/D
        PIR1bits.ADIF = 0;              //Clear A/D Interrupt Flag
        VTot += Vread;                  //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);        //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;               //CTMUISrc is in 1/100ths of uA
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
}
```

## 27.9 Operation During Sleep/Idle Modes

### 27.9.1 SLEEP MODE

When the device enters any Sleep mode, the CTMU module current source is always disabled. If the CTMU is performing an operation that depends on the current source when Sleep mode is invoked, the operation may not terminate correctly. Capacitance and time measurements may return erroneous values.

### 27.9.2 IDLE MODE

The behavior of the CTMU in Idle mode is determined by the CTMUSIDL bit (CTMUCONH<5>). If CTMUSIDL is cleared, the module will continue to operate in Idle mode. If CTMUSIDL is set, the module's current source is disabled when the device enters Idle mode. In this case, if the module is performing an operation when Idle mode is invoked, the results will be similar to those with Sleep mode.

## 27.10 Effects of a Reset on CTMU

Upon Reset, all registers of the CTMU are cleared. This disables the CTMU module, turns off its current source and returns all configuration options to their default settings. The module needs to be re-initialized following any Reset.

If the CTMU is in the process of taking a measurement at the time of Reset, the measurement will be lost. A partial charge may exist on the circuit that was being measured, which should be properly discharged before the CTMU makes subsequent attempts to make a measurement. The circuit is discharged by setting and clearing the IDISSEN bit (CTMUCONH<1>) while the A/D Converter is connected to the appropriate channel.

**TABLE 27-1: REGISTERS ASSOCIATED WITH CTMU MODULE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| CTMUCONH | CTMUEN | — | CTMUSIDL | TGEN | EDGEN | EDGSEQEN | IDISSEN | CTTRIG |
| CTMUCONL | EDG2POL | EDG2SEL1 | EDG2SEL0 | EDG1POL | EDG1SEL1 | EDG1SEL0 | EDG2STAT | EDG1STAT |
| CTMUICON | ITRIM5 | ITRIM4 | ITRIM3 | ITRIM2 | ITRIM1 | ITRIM0 | IRNG1 | IRNG0 |
| PIR3 | TMR5GIF | — | RC2IF | TX2IF | CTMUIF | CCP2IF | CCP1IF | RTCCIF |
| PIE3 | TMR5GIE | — | RC2IE | TX2IE | CTMUIE | CCP2IE | CCP1IE | RTCCIE |
| IPR3 | TMR5GIP | — | RC2IP | TX2IP | CTMUIP | CCP2IP | CCP1IP | RTCCIP |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used during ECCP operation.

**TABLE 28-1: 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 300000h | CONFIG1L | — | XINST | — | SOSCSEL1 | SOSCSEL0 | INTOSCSEL | — | $\overline{\text{RETEN}}$ | -1-1 1--1 |
| 300001h | CONFIG1H | IESO | FCMEN | — | PLLCFG | FOSC3 | FOSC2 | FOSC1 | FOSC0 | 00-0 1000 |
| 300002h | CONFIG2L | — | BORPWR1 | BORWPR0 | BORV1 | BORV0 | BOREN1 | BOREN0 | $\overline{\text{PWRTEN}}$ | -111 1111 |
| 300003h | CONFIG2H | — | WDTPS4 | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | WDTEN1 | WDTEN0 | -111 1111 |
| 300004h | CONFIG3L | WAIT(2) | BW(2) | ABW1(2) | ABW0(2) | EASHFT(2) | — | — | RTCOSC | ---- ---1 |
| 300005h | CONFIG3H | MCLRE | — | — | — | MSSPMSK | — | ECCPMX(2) | CCP2MX | 1--- 1-11 |
| 300006h | CONFIG4L | $\overline{\text{DEBUG}}$ | — | — | BBSIZ0 | — | — | — | STVREN | 1--1 ---1 |
| 300008h | CONFIG5L | CP7(1) | CP6(1) | CP5(1) | CP4(1) | CP3 | CP2 | CP1 | CP0 | 1111 1111 |
| 300009h | CONFIG5H | CPD | CPB | — | — | — | — | — | — | 11-- ---- |
| 30000Ah | CONFIG6L | WRT7(1) | WRT6(1) | WRT5(1) | WRT4(1) | WRT3 | WRT2 | WRT1 | WRT0 | 1111 1111 |
| 30000Bh | CONFIG6H | WRTD | WRTB | WRTC | — | — | — | — | — | 111- ---- |
| 30000Ch | CONFIG7L | EBTR7(1) | EBTR6(1) | EBTR5(1) | EBTR4(1) | EBTR3 | EBTR2 | EBTR1 | EBTR0 | 1111 1111 |
| 30000Dh | CONFIG7H | — | EBTRB | — | — | — | — | — | — | -1-- ---- |
| 3FFFFEh | DEVID1(3) | DEV2 | DEV1 | DEV0 | REV4 | REV3 | REV2 | REV1 | REV0 | xxxx xxxx |
| 3FFFFFh | DEVID2(3) | DEV10 | DEV9 | DEV8 | DEV7 | DEV6 | DEV5 | DEV4 | DEV3 | xxxx xxxx |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. Shaded cells are unimplemented, read as '0'.

**Note 1:** Implemented only on the PIC18F67K22 and PIC18F87K22 devices.

**2:** Implemented only on the 80-pin devices (PIC18F8XK22).

**3:** See Register 28-14 for DEVID1 values. DEVID registers are read-only and cannot be programmed by the user.

## 30.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

## 30.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

## 30.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

## 31.2 DC Characteristics: Power-Down and Supply Current PIC18F87K22 Family (Industrial/Extended) (Continued)

| PIC18F87K22 Family (Industrial/Extended) | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Supply Current (IDD) Cont.(2,3)** | | | | | | |
| | All devices | 2.1 | 5.5 | µA | -40°C | VDD = 1.8V(4) Regulator Disabled | FOSC = 31 kHz (**RC_IDLE** mode, LF-INTOSC) |
| | | 2.1 | 5.7 | µA | +25°C | | |
| | | 2.2 | 6.0 | µA | +85°C | | |
| | | 10 | 20 | µA | +125ºC | | |
| | All devices | 3.7 | 7.5 | µA | -40°C | VDD = 3.3V(4) Regulator Disabled | |
| | | 3.9 | 7.8 | µA | +25°C | | |
| | | 3.9 | 8.5 | µA | +85°C | | |
| | | 12 | 24 | µA | +125ºC | | |
| | All devices | 70 | 180 | µA | -40°C | VDD = 5V(5) Regulator Enabled | |
| | | 80 | 190 | µA | +25°C | | |
| | | 80 | 200 | µA | +85°C | | |
| | | 200 | 420 | µA | +125ºC | | |
| | All devices | 330 | 650 | µA | -40°C | VDD = 1.8V(4) Regulator Disabled | FOSC = 1 MHz (**RC_IDLE** mode, HF-INTOSC) |
| | | 330 | 640 | µA | +25°C | | |
| | | 330 | 630 | µA | +85°C | | |
| | | 500 | 850 | µA | +125ºC | | |
| | All devices | 520 | 850 | µA | -40°C | VDD = 3.3V(4) Regulator Disabled | |
| | | 520 | 900 | µA | +25°C | | |
| | | 520 | 850 | µA | +85°C | | |
| | | 800 | 1200 | µA | +125ºC | | |
| | All devices | 590 | 940 | µA | -40°C | VDD = 5V(5) Regulator Enabled | |
| | | 600 | 960 | µA | +25°C | | |
| | | 620 | 990 | µA | +85°C | | |
| | | 1000 | 1400 | µA | +125ºC | | |
| | All devices | 470 | 770 | µA | -40°C | VDD = 1.8V(4) Regulator Disabled | FOSC = 4 MHz (**RC_IDLE** mode, internal HF-INTOSC) |
| | | 470 | 770 | µA | +25°C | | |
| | | 460 | 760 | µA | +85°C | | |
| | | 700 | 1000 | µA | +125ºC | | |
| | All devices | 800 | 1400 | µA | -40°C | VDD = 3.3V(4) Regulator Disabled | |
| | | 800 | 1350 | µA | +25°C | | |
| | | 790 | 1300 | µA | +85°C | | |
| | | 1100 | 1400 | µA | +125ºC | | |
| | All devices | 880 | 1600 | µA | -40°C | VDD = 5V(5) Regulator Enabled | |
| | | 890 | 1700 | µA | +25°C | | |
| | | 910 | 1800 | µA | +85°C | | |
| | | 1200 | 2200 | µA | +125ºC | | |

**Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in a high-impedance state and tied to VDD or VSS, and all features that add delta current are disabled (such as WDT, SOSC oscillator, BOR, etc.).

**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in Active Operation mode are:
OSC1 = External square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
MCLR = VDD; WDT enabled/disabled as specified.

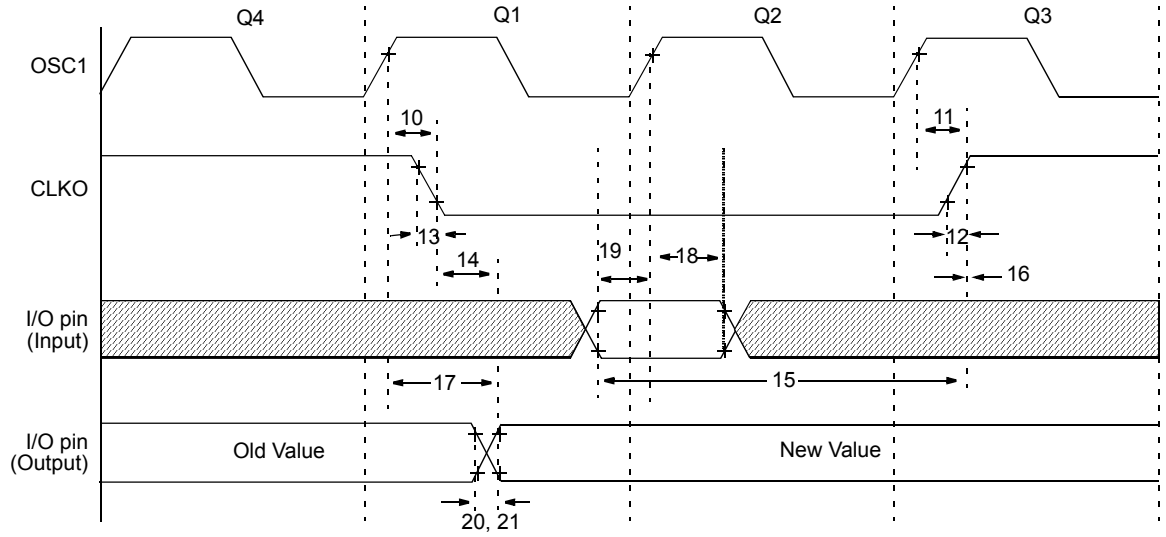
**3:** Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

**4:** Voltage regulator disabled (ENVREG = 0, tied to VSS, $\overline{\text{RETEN}}$ (CONFIG1L<0>) = 1).

**5:** Voltage regulator enabled (ENVREG = 1, tied to VDD, SRETEN (WDTCON<4>) = 1 and $\overline{\text{RETEN}}$ (CONFIG1L<0>) = 0).

**6:** 48 MHz, maximum frequency at +125ºC.

**FIGURE 31-5: CLKO AND I/O TIMING**




**Note:** Refer to Figure 31-3 for load conditions.

**TABLE 31-9: CLKO AND I/O TIMING REQUIREMENTS**

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 10 | TOSH2CKL | OSC1 ↑ to CLKO ↓ | — | 75 | 200 | ns | **(Note 1)** |
| 11 | TOSH2CKH | OSC1 ↑ to CLKO ↑ | — | 75 | 200 | ns | **(Note 1)** |
| 12 | TCKR | CLKO Rise Time | — | 15 | 30 | ns | **(Note 1)** |
| 13 | TCKF | CLKO Fall Time | — | 15 | 30 | ns | **(Note 1)** |
| 14 | TCKL2IOV | CLKO ↓ to Port Out Valid | — | — | 0.5 TCY + 20 | ns | |
| 15 | TIOV2CKH | Port In Valid before CLKO ↑ | 0.25 TCY + 25 | — | — | ns | |
| 16 | TCKH2IOI | Port In Hold after CLKO ↑ | 0 | — | — | ns | |
| 17 | TOSH2IOV | OSC1 ↑ (Q1 cycle) to Port Out Valid | — | 50 | 150 | ns | |
| 18 | TOSH2IOI | OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time) | 100 | — | — | ns | |
| 19 | TIOV2OSH | Port Input Valid to OSC1 ↑ (I/O in setup time) | 0 | — | — | ns | |
| 20 | TIOR | Port Output Rise Time | — | 10 | 25 | ns | |
| 21 | TIOF | Port Output Fall Time | — | 10 | 25 | ns | |
| 22† | TINP | INTx pin High or Low Time | 20 | — | — | ns | |
| 23† | TRBP | RB<7:4> Change INTx High or Low Time | TCY | — | — | ns | |

† These parameters are asynchronous events not related to any internal clock edges.

**Note 1:** Measurements are taken in EC mode, where CLKO output is 4 x TOSC.