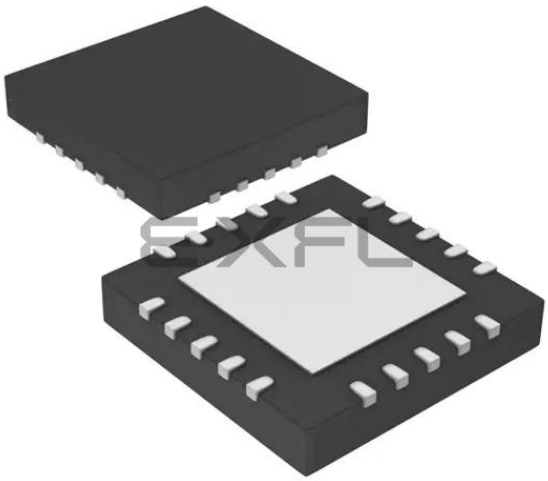


Welcome to E-XFL.COM

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	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VQFN Exposed Pad
Supplier Device Package	20-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f08ka101-i-mq

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC24F08KA101
- PIC24F16KA101
- PIC24F08KA102
- PIC24F16KA102

The PIC24F16KA102 family introduces a new line of extreme low-power Microchip devices: a 16-bit microcontroller family with a broad peripheral feature set and enhanced computational performance. It also offers a new migration option for those high-performance applications, which may be outgrowing their 8-bit platforms, but do not require the numerical processing power of a digital signal processor.

1.1 Core Features

1.1.1 16-BIT ARCHITECTURE

Central to all PIC24F devices is the 16-bit modified Harvard architecture, first introduced with Microchip's dsPIC® digital signal controllers. The PIC24F CPU core offers a wide range of enhancements, such as:

- 16-bit data and 24-bit address paths with the ability to move information between data and memory spaces
- Linear addressing of up to 12 Mbytes (program space) and 64 Kbytes (data)
- A 16-element working register array with built-in software stack support
- A 17 x 17 hardware multiplier with support for integer math
- Hardware support for 32-bit by 16-bit division
- An instruction set that supports multiple addressing modes and is optimized for high-level languages, such as C
- Operational performance up to 16 MIPS

1.1.2 POWER-SAVING TECHNOLOGY

All of the devices in the PIC24F16KA102 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **On-the-Fly Clock Switching:** The device clock can be changed under software control to the Timer1 source or the internal, low-power RC oscillator during operation, allowing users to incorporate power-saving ideas into their software designs.
- **Doze Mode Operation:** When timing-sensitive applications, such as serial communications, require the uninterrupted operation of peripherals, the CPU clock speed can be selectively reduced, allowing incremental power savings without missing a beat.
- **Instruction-Based Power-Saving Modes:** There are three instruction-based power-saving modes:
  - Idle Mode: The core is shut down while leaving the peripherals active.
  - Sleep Mode: The core and peripherals that require the system clock are shut down, leaving the peripherals that use their own clock, or the clock from other devices, active.
  - Deep Sleep Mode: The core, peripherals (except RTCC and DSWDT), Flash and SRAM are shut down.

1.1.3 OSCILLATOR OPTIONS AND FEATURES

The PIC24F16KA102 family offers five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes using crystals or ceramic resonators.
- Two External Clock modes offering the option of a divide-by-2 clock output.
- Two Fast Internal Oscillators (FRCs): One with a nominal 8 MHz output and the other with a nominal 500 kHz output. These outputs can also be divided under software control to provide clock speed as low as 31 kHz or 2 kHz.
- A Phase Locked Loop (PLL) frequency multiplier, available to the External Oscillator modes and the 8 MHz FRC oscillator, which allows clock speeds of up to 32 MHz.
- A separate Internal RC oscillator (LPRC) with a fixed 31 kHz output, which provides a low-power option for timing-insensitive applications.

NOTES:

TABLE 4-6: TIMER REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TMR1	0100	Timer1 Register																0000
PR1	0102	Timer1 Period Register																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																0000
TMR3	010A	Timer3 Register																0000
PR2	010C	Timer2 Period Register																FFFF
PR3	010E	Timer3 Period Register																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-7: INPUT CAPTURE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
IC1BUF	0140	Input Capture 1 Register																FFFF
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-8: OUTPUT COMPARE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
OC1RS	0180	Output Compare 1 Secondary Register																FFFF
OC1R	0182	Output Compare 1 Register																FFFF
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM2	OCM1	OCM0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-15: A/D REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADC1BUF0	0300	A/D Data Buffer 0																xxxx
ADC1BUF1	0302	A/D Data Buffer 1																xxxx
ADC1BUF2	0304	A/D Data Buffer 2																xxxx
ADC1BUF3	0306	A/D Data Buffer 3																xxxx
ADC1BUF4	0308	A/D Data Buffer 4																xxxx
ADC1BUF5	030A	A/D Data Buffer 5																xxxx
ADC1BUF6	030C	A/D Data Buffer 6																xxxx
ADC1BUF7	030E	A/D Data Buffer 7																xxxx
ADC1BUF8	0310	A/D Data Buffer 8																xxxx
ADC1BUF9	0312	A/D Data Buffer 9																xxxx
ADC1BUFA	0314	A/D Data Buffer 10																xxxx
ADC1BUFB	0316	A/D Data Buffer 11																xxxx
ADC1BUFC	0318	A/D Data Buffer 12																xxxx
ADC1BUFD	031A	A/D Data Buffer 13																xxxx
ADC1BUFE	031C	A/D Data Buffer 14																xxxx
ADC1BUFF	031E	A/D Data Buffer 15																xxxx
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	OFFCAL	—	CSCNA	—	—	BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	—	—	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	—	—	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0328	CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1PCFG	032C	—	—	—	PCFG12	PCFG11	PCFG10	—	—	—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	—	—	—	CSSL12	CSSL11	CSSL10	—	—	—	—	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-16: CTMU REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

5.5.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of Flash program memory at a time by erasing the programmable row. The general process is:

1. Read a row of program memory (32 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase a row (see Example 5-1):
   a) Set the NVMOP bits (NVMCON<5:0>) to '011000' to configure for row erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
   b) Write the starting address of the block to be erased into the TBLPAG and W registers.
   c) Write 55h to NVMKEY.
   d) Write AAh to NVMKEY.
   e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 32 instructions from data RAM into the program memory buffers (see Example 5-1).
5. Write the program block to Flash memory:
   a) Set the NVMOP bits to '000100' to configure for row programming. Clear the ERASE bit and set the WREN bit.
   b) Write 55h to NVMKEY.
   c) Write AAh to NVMKEY.
   d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs, as displayed in Example 5-5.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY ROW – ASSEMBLY LANGUAGE CODE

```
; Set up NVMCON for row erase operation
        MOV     #0x4058, W0                 ;
        MOV     W0, NVMCON                  ; Initialize NVMCON
; Init pointer to row to be ERASED
        MOV     #tblpage(PROG_ADDR), W0     ;
        MOV     W0, TBLPAG                  ; Initialize PM Page Boundary SFR
        MOV     #tbloffset(PROG_ADDR), W0   ; Initialize in-page EA[15:0] pointer
        TBLWTL  W0, [W0]                    ; Set base address of erase block
        DISI    #5                          ; Block all interrupts
                                              for next 5 instructions
        MOV     #0x55, W0
        MOV     W0, NVMKEY                  ; Write the 55 key
        MOV     #0xAA, W1                   ;
        MOV     W1, NVMKEY                  ; Write the AA key
        BSET    NVMCON, #WR                 ; Start the erase sequence
        NOP                                 ; Insert two NOPs after the erase
        NOP                                 ; command is asserted
```

EXAMPLE 5-2: ERASING A PROGRAM MEMORY ROW – 'C' LANGUAGE CODE

```
// C example using MPLAB C30

int __attribute__ ((space(auto_psv))) progAddr = 0x1234;    // Variable located in Pgm Memory, declared as a
                                                            // global variable
    unsigned int offset;

//Set up pointer to the first memory location to be written

    TBLPAG = __builtin_tblpage(&progAddr);                  // Initialize PM Page Boundary SFR
    offset = __builtin_tbloffset(&progAddr);                // Initialize lower word of address

    __builtin_tblwtl(offset, 0x0000);                       // Set base address of erase block
                                                            // with dummy latch write

    NVMCON = 0x4058;                                        // Initialize NVMCON

    asm("DISI #5");                                         // Block all interrupts for next 5 instructions
    __builtin_write_NVM();                                  // C30 function to perform unlock
                                                            // sequence and set WR
```

REGISTER 8-7: IFS3: INTERRUPT FLAG STATUS REGISTER 3

U-0	R/W-0, HS	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIF	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:	HS = Hardware 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 15 **Unimplemented:** Read as ‘0’

bit 14 **RTCIF:** Real-Time Clock and Calendar Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13-0 **Unimplemented:** Read as ‘0’

REGISTER 8-13: IPC0: INTERRUPT PRIORITY CONTROL REGISTER 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
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 15 **Unimplemented:** Read as ‘0’

bit 14-12 **T1IP<2:0>:** Timer1 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **OC1IP<2:0>:** Output Compare Channel 1 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **IC1IP<2:0>:** Input Capture Channel 1 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **INT0IP<2:0>:** External Interrupt 0 Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 7 **CLKLOCK:** Clock Selection Lock Enabled bit
If FSCM is enabled (FCKSM1 = 1):
1 = Clock and PLL selections are locked
0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit
If FSCM is disabled (FCKSM1 = 0):
Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.

bit 6 **Unimplemented:** Read as '0'

bit 5 **LOCK:** PLL Lock Status bit(2)
1 = PLL module is in lock or PLL module start-up timer is satisfied
0 = PLL module is out of lock, PLL start-up timer is running or PLL is disabled

bit 4 **Unimplemented:** Read as '0'

bit 3 **CF:** Clock Fail Detect bit
1 = FSCM has detected a clock failure
0 = No clock failure has been detected

bit 2 **Unimplemented:** Read as '0'

bit 1 **SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit
1 = Enable secondary oscillator
0 = Disable secondary oscillator

bit 0 **OSWEN:** Oscillator Switch Enable bit
1 = Initiate an oscillator switch to clock source specified by NOSC<2:0> bits
0 = Oscillator switch is complete

Note 1: Reset values for these bits are determined by the FNOSC Configuration bits.
2: Also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.

REGISTER 16-1: SPI1STAT: SPI1 STATUS AND CONTROL REGISTER (CONTINUED)

bit 1 **SPITBF:** SPI1 Transmit Buffer Full Status bit

1 = Transmit has not yet started, SPI1TXB is full
0 = Transmit has started, SPI1TXB is empty

In Standard Buffer mode:
Automatically set in hardware when the CPU writes to the SPITBF location, loading SPITBF.
Automatically cleared in hardware when the SPI1 module transfers data from SPI1TXB to SPIRBF.

In Enhanced Buffer mode:
Automatically set in hardware when CPU writes to the SPI1BUF location, loading the last available buffer location.
Automatically cleared in hardware when a buffer location is available for a CPU write.

bit 0 **SPIRBF:** SPI1 Receive Buffer Full Status bit

1 = Receive is complete; SPI1RXB is full
0 = Receive is not complete; SPI1RXB is empty

In Standard Buffer mode:
Automatically set in hardware when SPI1 transfers data from SPIRBF to SPIRBF.
Automatically cleared in hardware when the core reads the SPI1BUF location, reading SPIRBF.

In Enhanced Buffer mode:
Automatically set in hardware when SPI1 transfers data from SPI1SR to buffer, filling the last unread buffer location.
Automatically cleared in hardware when a buffer location is available for a transfer from SPI1SR.

19.2 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

19.2.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through corresponding register pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTR bits (RCFGCAL<9:8>) to select the desired Timer register pair (see Table 19-1).

By writing the RTCVALH byte, the RTCC Pointer value, the RTCPTR<1:0> bits decrement by one until they reach '00'. Once they reach '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 19-1: RTCVAL REGISTER MAPPING

RTCPTR<1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 19-2).

By writing the ALRMVALH byte, the Alarm Pointer value bits (ALRMPTR<1:0>) decrement by one until they reach '00'. Once they reach '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

TABLE 19-2: ALRMVAL REGISTER MAPPING

ALRMPTR <1:0>	Alarm Value Register Window	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL bytes, the ALRMPTR<1:0> value will be decremented. The same applies to the RTCVALH or RTCVALL bytes with the RTCPTR<1:0> being decremented.

Note:	This only applies to read operations and not write operations.

19.2.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RCFGCAL<13>) must be set (refer to Example 19-1).

Note:	To avoid accidental writes to the timer, it is recommended that the RTCWREN bit (RCFGCAL<13>) is kept clear at any other time. For the RTCWREN bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN; therefore, it is recommended that code follow the procedure in Example 19-1.

19.2.3 SELECTING RTCC CLOCK SOURCE

The clock source for the RTCC module can be selected using the RTCOSC (FDS<5>) bit. When the bit is set to '1', the Secondary Oscillator (SOSC) is used as the reference clock and when the bit is '0', LPRC is used as the reference clock.

EXAMPLE 19-1: SETTING THE RTCWREN BIT

```
    asm volatile ("push w7")                ;
    asm volatile ("push w8")                ;
    asm volatile ("disi #5")                ;
    asm volatile ("mov #0x55, w7")          ;
    asm volatile ("mov w7, _NVMKEY")        ;
    asm volatile ("mov #0xAA, w8")          ;
    asm volatile ("mov w8, _NVMKEY")        ;
    asm volatile ("bset _RCFGCAL, #13")     ; //set the RTCWREN bit
    asm volatile ("pop w8")                 ;
    asm volatile ("pop w7");
```

REGISTER 20-2: CRCXOR: CRC XOR POLYNOMIAL 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
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X7	X6	X5	X4	X3	X2	X1	—
bit 7							bit 0

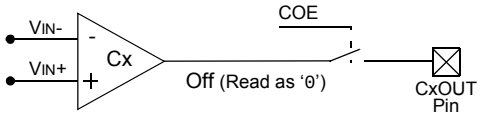
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-1 **X<15:1>:** XOR of Polynomial Term X^n Enable bits

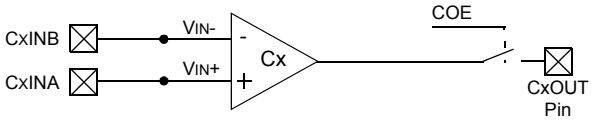
bit 0 **Unimplemented:** Read as ‘0’

FIGURE 23-2: INDIVIDUAL COMPARATOR CONFIGURATIONS

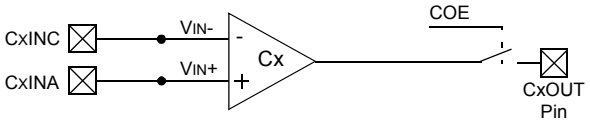
Comparator Off
CON = 0, CREF = x, CCH<1:0> = xx



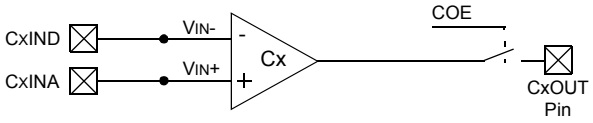
Comparator CxINB < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 00



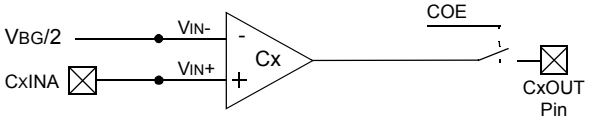
Comparator CxINC < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 01



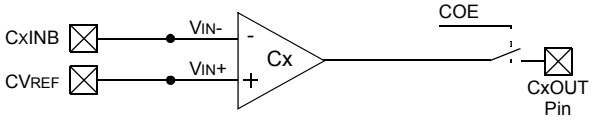
Comparator CxIND < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 10



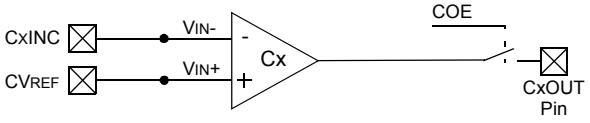
Comparator VBG/2 < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 11



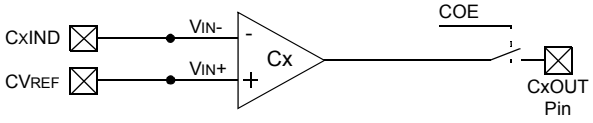
Comparator CxINB < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 00



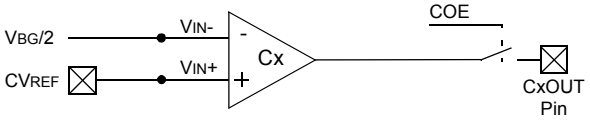
Comparator CxINC < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 01



Comparator CxIND < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 10



Comparator VBG/2 < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 11



27.11 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.

27.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} 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.

27.13 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.

TABLE 29-7: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Idle Current (IIDLE): Core Off, Clock on Base Current, PMD Bits are Set(2)						
DC40	48	100	μA	-40°C	1.8V	0.5 MIPS, FOSC = 1 MHz
DC40a		100		+25°C		
DC40b		100		+60°C		
DC40c		100		+85°C		
DC40d		100		+125°C		
DC40e	106	215	μA	-40°C	3.3V	
DC40f		215		+25°C		
DC40g		215		+60°C		
DC40h		215		+85°C		
DC40i		450		+125°C		
DC42	94	200	μA	-40°C	1.8V	1 MIPS, FOSC = 2 MHz
DC42a		200		+25°C		
DC42b		200		+60°C		
DC42c		200		+85°C		
DC42d		300		+125°C		
DC42e	160	395	μA	-40°C	3.3V	
DC42f		395		+25°C		
DC42g		395		+60°C		
DC42h		395		+85°C		
DC42i		600		+125°C		
DC43	3.1	6.0	mA	-40°C	3.3V	16 MIPS, FOSC = 32 MHz
DC43a		6.0		+25°C		
DC43b		6.0		+60°C		
DC43c		6.0		+85°C		
DC43d		6.0		+125°C		
DC44	0.56	0.74	mA	-40°C	1.8V	FRC (4 MIPS), FOSC = 8 MHz
DC44a		0.74		+25°C		
DC44b		0.74		+60°C		
DC44c		0.74		+85°C		
DC44d		0.74		+125°C		
DC44e	0.95	1.50	mA	-40°C	3.3V	
DC44f		1.50		+25°C		
DC44g		1.50		+60°C		
DC44h		1.50		+85°C		
DC44i		1.50		+125°C		

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Operating Parameters:
- Core is off
- EC mode with the clock input driven with a square wave rail-to-rail
- I/Os are configured as outputs, driven low
- $\overline{MCLR}$ – VDD
- WDT FSCM are disabled
- SRAM, program and data memory are active
- All PMD bits are set except for the modules being measured

TABLE 29-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD) (CONTINUED)

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended		
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**	
Power-Down Current (IPD): PMD Bits are Set, PMSLP Bit is '0'(2)					
DC62	0.450	0.650	μA	-40°C	1.8V, Timer1 w/32 kHz Crystal: T132 (SOSC – LP)(3)
DC62a		0.650		+25°C	
DC62b		0.650		+60°C	
DC62c		0.650		+85°C	
DC62d		—		+125°C	
DC62e	0.730	0.980	μA	-40°C	3.3V, Timer1 w/32 kHz Crystal: T132 (SOSC – LP)(3)
DC62f		0.980		+25°C	
DC62g		0.980		+60°C	
DC62h		0.980		+85°C	
DC62i		—		+125°C	
DC64	5.5	7.10	μA	-40°C	1.8V, HLVD(3,4)
DC64a		7.10		+25°C	
DC64b		7.80		+60°C	
DC64c		8.30		+85°C	
DC64d		10.00		+125°C	
DC64e	6.2	7.10	μA	-40°C	3.3V, HLVD(3,4)
DC64f		7.10		+25°C	
DC64g		7.80		+60°C	
DC64h		8.30		+85°C	
DC64i		9.00		+125°C	
DC63	4.5	6.60	μA	-40°C	3.3V, BOR(3,4)
DC63a		6.60		+25°C	
DC63b		6.60		+60°C	
DC63c		6.60		+85°C	
DC63d		9.00		+125°C	
DC62	0.49	0.65	μA	-40°C	1.8V, RTCC(3,5)
DC62a		0.65		+25°C	
DC62b		0.65		+60°C	
DC62c		0.65		+85°C	
DC62d		0.98		+125°C	
DC62e	0.80	0.98	μA	-40°C	3.3V, RTCC(3,5)
DC62f		0.98		+25°C	
DC62g		0.98		+60°C	
DC62h		0.98		+85°C	
DC62i		0.98		+125°C	

Note 1: Data in the "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as outputs and set low. WDT, etc., are all switched off.

3: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

4: Current applies to Sleep only.

5: Current applies to Sleep and Deep Sleep.

6: Current applies to Deep Sleep only.

FIGURE 29-5: EXTERNAL CLOCK TIMING

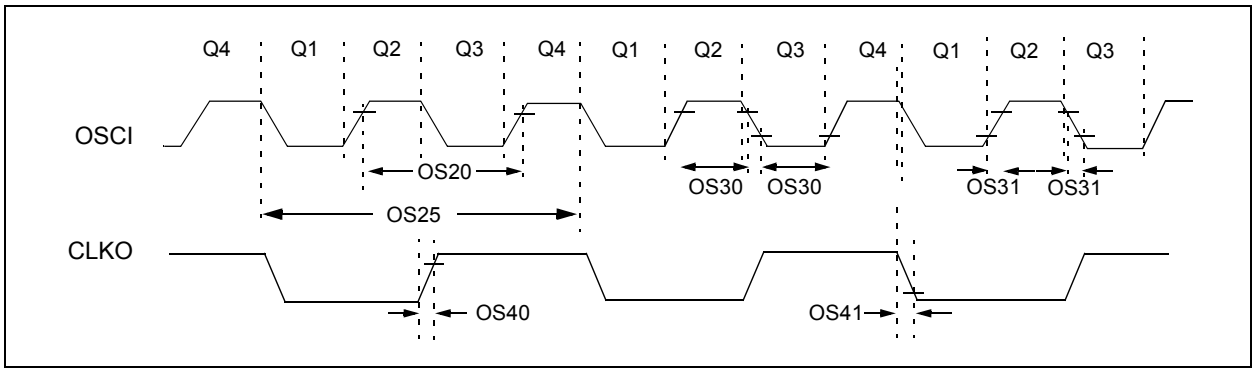


TABLE 29-19: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 1.8 to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
OS10	FOSC	External CLKI Frequency (External clocks allowed only in EC mode)(2)	DC 4	— —	32 8	MHz MHz	EC ECPLL
		Oscillator Frequency(2)	0.2 4 4 31	— — — —	4 25 8 33	MHz MHz MHz kHz	XT HS HSPLL SOSC
OS20	TOSC	TOSC = 1/FOSC	—	—	—	—	See Parameter OS10 for FOSC value
OS25	TCY	Instruction Cycle Time(3)	62.5	—	DC	ns	
OS30	TosL, TosH	External Clock in (OSCI) High or Low Time	0.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	External Clock in (OSCI) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKO Rise Time(4)	—	6	10	ns	
OS41	TckF	CLKO Fall Time(4)	—	6	10	ns	

Note 1: Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Refer to Figure 29-1 for the minimum voltage at a given frequency.

3: Instruction cycle period (TCY) equals two times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “Min.” values with an external clock applied to the OSCI/CLKI pin. When an external clock input is used, the “Max.” cycle time limit is “DC” (no clock) for all devices.

4: Measurements are taken in EC mode. The CLKO signal is measured on the OSCO pin. CLKO is low for the Q1-Q2 period (1/2 TCY) and high for the Q3-Q4 period (1/2 TCY).

FIGURE 29-14: INPUT CAPTURE TIMINGS

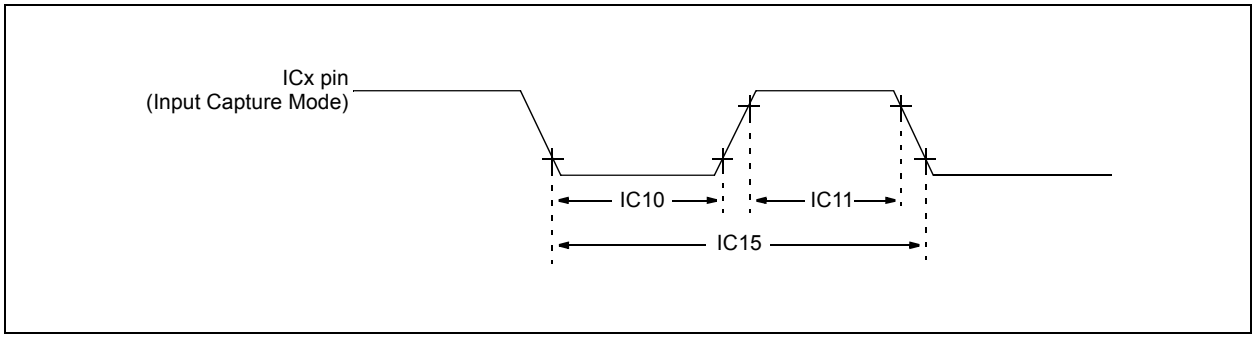


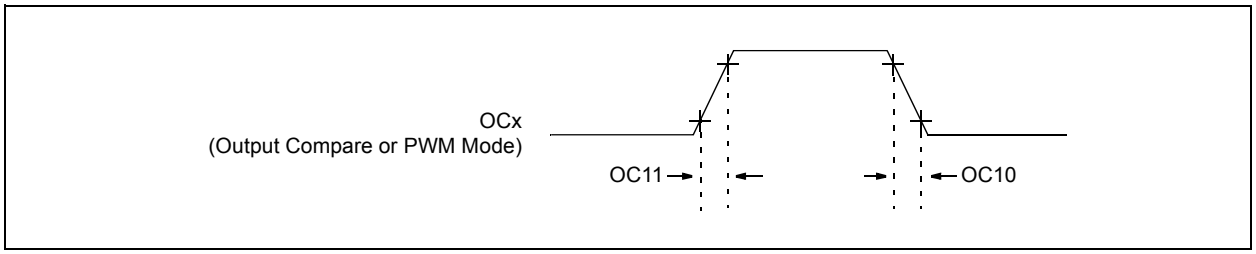
TABLE 29-33: INPUT CAPTURE

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
IC10	TccL	ICx Input Low Time – Synchronous Timer	No Prescaler	TCY + 20	—	ns	Must also meet Parameter IC15
			With Prescaler	20	—	ns	
IC11	TccH	ICx Input Low Time – Synchronous Timer	No Prescaler	TCY + 20	—	ns	Must also meet Parameter IC15
			With Prescaler	20	—	ns	
IC15	TccP	ICx Input Period – Synchronous Timer		$\frac{2 * T_{CY} + 40}{N}$	—	ns	N = prescale value (1, 4, 16)

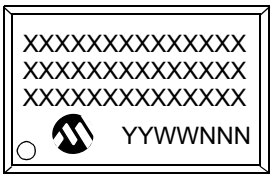
TABLE 29-34: OUTPUT CAPTURE

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
OC11	TccR	OC1 Output Rise Time	—	10	ns	
			—	—	ns	
OC10	TccF	OC1 Output Fall Time	—	10	ns	
			—	—	ns	

FIGURE 29-15: OUTPUT COMPARE TIMINGS

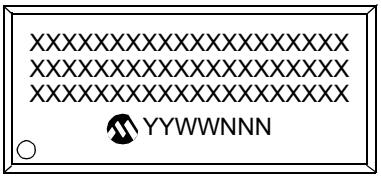


20-Lead SOIC (.300”)

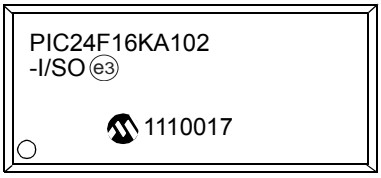


Example

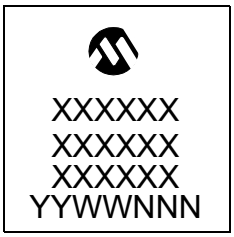
28-Lead SOIC (.300”)



Example

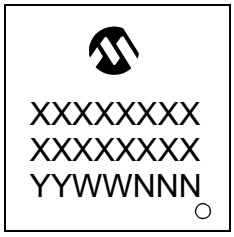


20-Lead QFN



Example

28-Lead QFN



Example

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___ Y ___ N

Device: PIC24F16KA102 Family Literature Number: DS39927C

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?