

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

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

Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	38
Program Memory Size	16KB (5.5K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 22x10b/12b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f16km204-e-ml

Email: info@E-XFL.COM

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

TABLE 4-4: ICN 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
CNPD1	56h	CN15PDE ^(1,2)	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE(2)	CN9PDE ^(1,2)	-	CN7PDE ^(1,2)	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	58h	CN31PDE ⁽²⁾	CN30PDE	CN29PDE	CN28PDE ⁽²⁾	CN27PDE ^(1,2)	CN26PDE ⁽²⁾	CN25PDE ⁽²⁾	CN24PDE ^(1,2)	CN23PDE	CN22PDE	CN21PDE	CN20PDE(2)	CN19PDE ⁽²⁾	CN18PDE ⁽²⁾	CN17PDE ⁽²⁾	CN16PDE ^(1,2)	0000
CNPD3	5Ah	_	_	_	_	_	_	_	_	_	_	_	CN36PDE ⁽²⁾	CN35PDE ⁽²⁾	CN34PDE ⁽²⁾	CN33PDE ⁽²⁾	CN32PDE ⁽²⁾	0000
CNEN1	62h	CN15IE ^(1,2)	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE ⁽²⁾	CN9IE ^(1,2)	_	CN7IE ^(1,2)	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	64h	CN31IE ⁽²⁾	CN30IE	CN29IE	CN28IE ⁽²⁾	CN27IE ^(1,2)	CN26IE ⁽²⁾	CN25IE ⁽²⁾	CN24IE ^(1,2)	CN23IE	CN22IE	CN21IE	CN20IE ⁽²⁾	CN19IE ⁽²⁾	CN18IE ⁽²⁾	CN17IE ⁽²⁾	CN16IE ^(1,2)	0000
CNEN3	66h	-	_	—	_	_	_	—	—	—	_	_	CN36IE ⁽²⁾	CN35IE ⁽²⁾	CN34IE ⁽²⁾	CN33IE ⁽²⁾	CN32IE ⁽²⁾	0000
CNPU1	6Eh	CN15PUE ^(1,2)	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE ⁽²⁾	CN9PUE ^(1,2)	—	CN7PUE ^(1,2)	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	70h	CN31PUE ⁽²⁾	CN30PUE	CN29PUE	CN28PUE(2)	CN27PUE ^(1,2)	CN26PUE ⁽²⁾	CN25PUE ⁽²⁾	CN24PUE ^(1,2)	CN23PUE	CN22PUE	CN21PUE	CN20PUE ⁽²⁾	CN19PUE ⁽²⁾	CN18PUE ⁽²⁾	CN17PUE ⁽²⁾	CN16PUE ^(1,2)	0000
CNPU3	72h	_	_	_		_	_	_	_	_	_	_	CN36PUE ⁽²⁾	CN35PUE ⁽²⁾	CN34PUE ⁽²⁾	CN33PUE ⁽²⁾	CN32PUE ⁽²⁾	0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.

Note 1: These bits are available only on 28-pin devices

2: These bits are available only on 44-pin devices

EXAMPLE 5-3: LOADING THE WRITE BUFFERS – ASSEMBLY LANGUAGE CODE

;	Set up NVMCO	N for row programming operation	ıs	
	MOV	#0x4004, W0	;	
	MOV	W0, NVMCON	;	Initialize NVMCON
;	Set up a poir	nter to the first program memor	сy	location to be written
;	program memo:	ry selected, and writes enabled	1	
	MOV	#0x0000, W0	;	
	MOV	W0, TBLPAG	;	Initialize PM Page Boundary SFR
	MOV	#0x1500, W0	;	An example program memory address
;	Perform the '	TBLWT instructions to write the	2	latches
;	0th_program_	word		
	MOV	#LOW_WORD_0, W2	;	
	MOV	#HIGH_BYTE_0, W3	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0++]	;	Write PM high byte into program latch
;	lst_program_v	word		
	MOV	#LOW_WORD_1, W2	;	
	MOV	#HIGH_BYTE_1, W3	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0++]	;	Write PM high byte into program latch
;	2nd_program_v	word		
	MOV	#LOW_WORD_2, W2	;	
	MOV	#HIGH_BYTE_2, W3	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0++]	;	Write PM high byte into program latch
	•			
	•			
	•			
;	32nd_program	_word		
	MOV	#LOW_WORD_31, W2	;	
	MOV	<pre>#HIGH_BYTE_31, W3</pre>	;	
	TBLWTL	W2, [WU]	;	Write PM low word into program latch
	TBLWTH	W3, [W0]	;	Write PM high byte into program latch

EXAMPLE 5-4: LOADING THE WRITE BUFFERS – 'C' LANGUAGE CODE

```
// C example using MPLAB C30
  #define NUM_INSTRUCTION_PER_ROW 64
int __attribute__ ((space(auto_psv))) progAddr = 0x1234 // Variable located in Pgm Memory
  unsigned int offset;
  unsigned int i;
  unsigned int progData[2*NUM_INSTRUCTION_PER_ROW];
                                                            // Buffer of data to write
  //Set up NVMCON for row programming
  NVMCON = 0 \times 4004;
                                                            // Initialize NVMCON
  //Set up pointer to the first memory location to be written
  TBLPAG = __builtin_tblpage(&progAddr);
                                                           // Initialize PM Page Boundary SFR
                                                            // Initialize lower word of address
  offset = __builtin_tbloffset(&progAddr);
  //Perform TBLWT instructions to write necessary number of latches
  for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)</pre>
  {
                                                          // Write to address low word
      __builtin_tblwtl(offset, progData[i++]);
       __builtin_tblwth(offset, progData[i]);
                                                            // Write to upper byte
      offset = offset + 2;
                                                            // Increment address
  }
```

REGISTER 8-14: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0	
—	—	—	—	—	—	CCT5IE	—	
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:								
R = Readabl	e bit	W = Writable	bit	U = Unimplemented bit, read as '0'				
-n = Value at POR		'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unknown		
bit 15-10	Unimplemen	ted: Read as '	כי					
hit O		turo/Compore	E Timor Interru	nt Enchla hit				

bit 9	CCT5IE: Capture/Compare 5 Timer Interrupt Enable bit
	1 = Interrupt request is enabled
	0 = Interrupt request is not enabled

bit 8-0 Unimplemented: Read as '0'

REGISTER 8-15: IEC3: INTERRUPT ENABLE CONTROL REGISTER 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
_	RTCIE	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	BCL2IE	SSP2IE	—
bit 7 bit							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	d 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	RTCIE: Real-Time Clock and Calendar Interrupt Enable bit
	1 = Interrupt request is enabled0 = Interrupt request is not enabled
bit 13-3	Unimplemented: Read as '0'
bit 2	BCL2IE: MSSP2 I ² C [™] Bus Collision Interrupt Enable bit
	1 = Interrupt request is enabled0 = Interrupt request is not enabled
bit 1	SSP2IE: MSSP2 SPI/I ² C Event Interrupt Enable bit
	1 = Interrupt request is enabled0 = Interrupt request is not enabled
bit 0	Unimplemented: Read as '0'

9.1 CPU Clocking Scheme

The system clock source can be provided by one of four sources:

- Primary Oscillator (POSC) on the OSCI and OSCO pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins

The PIC24FXXXXX family devices consist of two types of secondary oscillator:

- High-Power Secondary Oscillator
- Low-Power Secondary Oscillator

These can be selected by using the SOSCSEL (FOSC<5>) bit.

- Fast Internal RC (FRC) Oscillator:
 - 8 MHz FRC Oscillator
 - 500 kHz Lower Power FRC Oscillator
- Low-Power Internal RC (LPRC) Oscillator with two modes:
 - High-Power/High-Accuracy mode
 - Low-Power/Low-Accuracy mode

The Primary Oscillator and 8 MHz FRC sources have the option of using the internal 4x PLL. The frequency of the FRC clock source can optionally be reduced by the programmable clock divider. The selected clock source generates the processor and peripheral clock sources.

The processor clock source is divided by two to produce the internal instruction cycle clock, Fcy. In this document, the instruction cycle clock is also denoted by Fosc/2. The internal instruction cycle clock, Fosc/2, can be provided on the OSCO I/O pin for some operating modes of the Primary Oscillator.

9.2 Initial Configuration on POR

The oscillator source (and operating mode) that is used at a device Power-on Reset (POR) event is selected using Configuration bit settings. The Oscillator Configuration bit settings are located in the Configuration registers in the program memory (for more information, see Section 25.1 "Configuration Bits"). The Primary Oscillator Configuration bits, POSCMD<1:0> (FOSC<1:0>), and the Initial Oscillator Select Configuration bits, FNOSC<2:0> (FOSCSEL<2:0>), select the oscillator source that is used at a POR. The FRC Primary Oscillator with Postscaler (FRCDIV) is the default (unprogrammed) selection. The Secondary Oscillator, or one of the internal oscillators, may be chosen by programming these bit locations. The EC mode Frequency Range Configuration bits, POSCFREQ<1:0> (FOSC<4:3>), optimize power consumption when running in EC mode. The default configuration is "frequency range is greater than 8 MHz".

The Configuration bits allow users to choose between the various clock modes, shown in Table 9-1.

9.2.1 CLOCK SWITCHING MODE CONFIGURATION BITS

The FCKSM<1:0> Configuration bits (FOSC<7:6>) are used jointly to configure device clock switching and the FSCM. Clock switching is enabled only when FCKSM1 is programmed ('0'). The FSCM is enabled only when FCKSM<1:0> are both programmed ('00').

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FNOSC<2:0>	Notes
8 MHz FRC Oscillator with Postscaler (FRCDIV)	Internal	11	111	1, 2
500 kHz FRC Oscillator with Postscaler (LPFRCDIV)	Internal	11	110	1
Low-Power RC Oscillator (LPRC)	Internal	11	101	1
Secondary (Timer1) Oscillator (SOSC)	Secondary	00	100	1
Primary Oscillator (HS) with PLL Module (HSPLL)	Primary	10	011	
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	
Primary Oscillator (HS)	Primary	10	010	
Primary Oscillator (XT)	Primary	01	010	
Primary Oscillator (EC)	Primary	00	010	
8 MHz FRC Oscillator with PLL Module (FRCPLL)	Internal	11	001	1
8 MHz FRC Oscillator (FRC)	Internal	11	000	1

TABLE 9-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Note 1: The OSCO pin function is determined by the OSCIOFNC Configuration bit.

2: This is the default oscillator mode for an unprogrammed (erased) device.

9.4 Clock Switching Operation

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switching process.

Note: The Primary Oscillator mode has three different submodes (XT, HS and EC), which are determined by the POSCMDx Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

9.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in the FOSC Configuration register must be programmed to '0'. (Refer to **Section 25.0** "**Special Features**" for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), the clock switching function and FSCM function are disabled; this is the default setting.

The NOSCx control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSCx bits (OSCCON<14:12>) will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN control bit (OSCCON<0>) has no effect when clock switching is disabled; it is held at '0' at all times.

9.4.2 OSCILLATOR SWITCHING SEQUENCE

At a minimum, performing a clock switch requires this basic sequence:

- 1. If desired, read the COSCx bits (OSCCON<14:12>) to determine the current oscillator source.
- 2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
- 3. Write the appropriate value to the NOSCx bits (OSCCON<10:8>) for the new oscillator source.
- 4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
- 5. Set the OSWEN bit to initiate the oscillator switch.

Once the basic sequence is completed, the system clock hardware responds automatically, as follows:

- 1. The clock switching hardware compares the COSCx bits with the new value of the NOSCx bits. If they are the same, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
- If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and CF (OSCCON<3>) bits are cleared.
- The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware will wait until the OST expires. If the new source is using the PLL, then the hardware waits until a PLL lock is detected (LOCK = 1).
- 4. The hardware waits for 10 clock cycles from the new clock source and then performs the clock switch.
- 5. The hardware clears the OSWEN bit to indicate a successful clock transition. In addition, the NOSCx bits value is transferred to the COSCx bits.
- The old clock source is turned off at this time, with the exception of LPRC (if WDT, FSCM or RTCC with LPRC as a clock source is enabled) or SOSC (if SOSCEN remains enabled).

Note 1: The processor will continue to execute code throughout the clock switching sequence. Timing-sensitive code should not be executed during this time.

2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.

12.0 TIMER1

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on timers, refer to the "PIC24F Family Reference Manual", "Timers" (DS39704).

The Timer1 module is a 16-bit timer which can serve as the time counter for the Real-Time Clock (RTC) or operate as a free-running, interval timer/counter. Timer1 can operate in three modes:

- 16-Bit Timer
- 16-Bit Synchronous Counter
- 16-Bit Asynchronous Counter

Timer1 also supports these features:

- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation During CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

Figure 12-1 illustrates a block diagram of the 16-bit Timer1 module.

To configure Timer1 for operation:

- 1. Set the TON bit (= 1).
- 2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
- 3. Set the Clock and Gating modes using the TCS and TGATE bits.
- 4. Set or clear the TSYNC bit to configure synchronous or asynchronous operation.
- 5. Load the timer period value into the PR1 register.
- 6. If interrupts are required, set the Timer1 Interrupt Enable bit, T1IE. Use the Timer1 Interrupt Priority bits, T1IP<2:0>, to set the interrupt priority.



13.0 CAPTURE/COMPARE/PWM/ TIMER MODULES (MCCP AND SCCP)

Note:	This data sheet summarizes the features
	of this group of PIC24F devices. It is not
	intended to be a comprehensive refer-
	ence source. For more information on the
	MCCP/SCCP modules, refer to the
	"PIC24F Family Reference Manual".

PIC24FV16KM204 family devices include several Capture/Compare/PWM/Timer base modules, which provide the functionality of three different peripherals of earlier PIC24F devices. The module can operate in one of three major modes:

- General Purpose Timer
- Input Capture
- Output Compare/PWM

The module is provided in two different forms, distinguished by the number of PWM outputs that the module can generate. Single output modules (SCCPs) provide only one PWM output. Multiple output modules (MCCPs) can provide up to six outputs and an extended range of power control features, depending on the pin count of the particular device. All other features of the modules are identical.

The SCCP and MCCP modules can be operated only in one of the three major modes at any time. The other modes are not available unless the module is reconfigured for the new mode. A conceptual block diagram for the module is shown in Figure 13-1. All three modes share a time base generator and a common Timer register pair (CCPxTMRH/L); other shared hardware components are added as a particular mode requires.

Each module has a total of seven control and status registers:

- CCPxCON1L (Register 13-1)
- CCPxCON1H (Register 13-2)
- CCPxCON2L (Register 13-3)
- CCPxCON2H (Register 13-4)
- CCPxCON3L (Register 13-5)
- CCPxCON3H (Register 13-6)
- CCPxSTATL (Register 13-7)

Each module also includes eight buffer/counter registers that serve as Timer Value registers or data holding buffers:

- CCPxTMRH/CCPxTMRL (Timer High/Low Counters)
- CCPxPRH/CCPxPRL (Timer Period High/Low)
- CCPxRA (Primary Output Compare Data Buffer)
- CCPxRB (Secondary Output Compare Data Buffer)
- CCPxBUFH/CCPxBUFL (Input Capture High/Low Buffers)

FIGURE 13-1: MCCPx/SCCPx CONCEPTUAL BLOCK DIAGRAM



13.2 General Purpose Timer

Timer mode is selected when CCSEL = 0 and MOD<3:0> = 0000. The timer can function as a 32-bit timer or a dual 16-bit timer, depending on the setting of the T32 bit (Table 13-2).

T32 (CCPxCON1L<5>)	Operating Mode
0	Dual Timer Mode (16-bit)
1	Timer Mode (32-bit)

TABLE 13-2: TIMER OPERATION MODE

Dual 16-Bit Timer mode provides a simple timer function with two independent 16-bit timer/counters. The primary timer uses CCPxTMRL and CCPxPRL. Only the primary timer can interact with other modules on the device. It generates the MCCPx Sync out signals for use by other MCCP modules. It can also use the SYNC<4:0> bits signal generated by other modules.

The secondary timer uses CCPxTMRH and CCPxPRH. It is intended to be used only as a periodic interrupt source for scheduling CPU events. It does not generate an Output Sync/Trigger signal like the primary time base. In Dual Timer mode, the Secondary Timer Period register, CCPxPRH, generates the MCCP Compare Event (CCPxIF) used by many other modules on the device.

The 32-Bit Timer mode uses the CCPxTMRL and CCPxTMRH registers, together, as a single 32-bit timer. When CCPxTMRL overflows, CCPxTMRH increments by one. This mode provides a simple timer function when it is important to track long time periods. Note that

FIGURE 13-3: DUAL	16-BIT TIMER	MODE
-------------------	---------------------	------

the T32 bit (CCPxCON1L<5>) should be set before the CCPxTMRL or CCPxPRH registers are written to initialize the 32-bit timer.

13.2.1 SYNC AND TRIGGER OPERATION

In both 16-bit and 32-bit modes, the timer can also function in either Synchronization ("Sync") or Trigger operation. Both use the SYNC<4:0> bits (CCPxCON1H<4:0>) to determine the input signal source. The difference is how that signal affects the timer.

In Sync operation, the timer Reset or clear occurs when the input selected by SYNC<4:0> is asserted. The timer immediately begins to count again from zero unless it is held for some other reason. Sync operation is used whenever the TRIGEN bit (CCPxCON1H<7>) is cleared. SYNC<4:0> can have any value except '11111'.

In Trigger operation, the timer is held in Reset until the input selected by SYNC<4:0> is asserted; when it occurs, the timer starts counting. Trigger operation is used whenever the TRIGEN bit is set. In Trigger mode, the timer will continue running after a Trigger event as long as the CCPTRIG bit (CCPxSTATL< 7>) is set. To clear CCPTRIG, the TRCLR bit (CCPxSTATL<5>) must be set to clear the Trigger event, reset the timer and hold it at zero until another Trigger event occurs. On PIC24FV16KM204 family devices, Trigger operation can only be used when the system clock is the time base source (CLKSEL<2:0> = 000).



NOTES:

15.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note:	This data sheet summarizes the features of									
	this group of PIC24F devices. It is not									
	intended to be a comprehensive reference									
	source. For more information on the Univer-									
	sal Asynchronous Receiver Transmitter,									
	refer to the "PIC24F Family Reference									
	<i>Manual"</i> , " UART" (DS39708).									

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in this PIC24F device family. The UART is a full-duplex, asynchronous system that can communicate with peripheral devices, such as personal computers, LIN/J2602, RS-232 and RS-485 interfaces. This module also supports a hardware flow control option with the UxCTS and UxRTS pins, and also includes an IrDA[®] encoder and decoder.

The primary features of the UART module are:

- Full-Duplex, 8-Bit or 9-Bit Data Transmission through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- · One or Two Stop bits
- Hardware Flow Control Option with UxCTS and UxRTS Pins
- Fully Integrated Baud Rate Generator (IBRG) with 16-Bit Prescaler

- Baud Rates Ranging from 1 Mbps to 15 bps at 16 MIPS
- 4-Deep, First-In-First-Out (FIFO) Transmit Data Buffer
- · 4-Deep FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-Bit mode with Address Detect (9th bit = 1)
- · Transmit and Receive Interrupts
- · Loopback mode for Diagnostic Support
- · Support for Sync and Break Characters
- · Supports Automatic Baud Rate Detection
- IrDA[®] Encoder and Decoder Logic
- 16x Baud Clock Output for IrDA Support

A simplified block diagram of the UARTx module is shown in Figure 15-1. The UARTx module consists of these important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- · Asynchronous Receiver
- Note: Throughout this section, references to register and bit names that may be associated with a specific USART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "UxSTA" might refer to the USART Status register for either USART1 or USART2.

FIGURE 15-1: UARTX MODULE SIMPLIFIED BLOCK DIAGRAM



16.2.5 RTCVAL REGISTER MAPPINGS

REGISTER 16-4: YEAR: YEAR VALUE REGISTER⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15			•	•			bit 8

| R/W-x |
|--------|--------|--------|--------|--------|--------|--------|--------|
| YRTEN3 | YRTEN2 | YRTEN1 | YRTEN0 | YRONE3 | YRONE2 | YRONE1 | YRONE0 |
| bit 7 | | | | | | | bit 0 |

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

bit 15-8 Unimplemented: Read as '0'

- bit 7-4 **YRTEN<3:0>:** Binary Coded Decimal Value of Year's Tens Digit bits Contains a value from 0 to 9.
- bit 3-0 **YRONE<3:0>:** Binary Coded Decimal Value of Year's Ones Digit bits Contains a value from 0 to 9.

Note 1: A write to the YEAR register is only allowed when RTCWREN = 1.

REGISTER 16-5: MTHDY: MONTH AND DAY VALUE REGISTER⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

bit 15-13 Unimplemented: Read as '0'

- bit 12 MTHTEN0: Binary Coded Decimal Value of Month's Tens Digit bit Contains a value of '0' or '1'.
- bit 11-8 MTHONE<3:0>: Binary Coded Decimal Value of Month's Ones Digit bits Contains a value from 0 to 9.
- bit 7-6 Unimplemented: Read as '0'
- bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit bits Contains a value from 0 to 3.
- bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day's Ones Digit bits Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

19.0 12-BIT A/D CONVERTER WITH THRESHOLD DETECT

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the 12-Bit A/D Converter with Threshold Detect, refer to the "PIC24F Family Reference Manual", "12-Bit A/D Converter with Threshold Detect" (DS39739).

The PIC24F 12-bit A/D Converter has the following key features:

- Successive Approximation Register (SAR)
 Conversion
- Conversion Speeds of up to 100 ksps
- Up to 32 Analog Input Channels (internal and external)
- Multiple Internal Reference Input Channels
- External Voltage Reference Input Pins
- Unipolar Differential Sample-and-Hold (S/H)
 Amplifier
- Automated Threshold Scan and Compare
 Operation to Pre-Evaluate Conversion Results
- Selectable Conversion Trigger Source
- Fixed-Length (one word per channel), Configurable Conversion Result Buffer
- Four Options for Results Alignment
- Configurable Interrupt Generation
- Operation During CPU Sleep and Idle modes

The 12-bit A/D Converter module is an enhanced version of the 10-bit module offered in some PIC24 devices. Both modules are Successive Approximation Register (SAR) converters at their cores, surrounded by a range of hardware features for flexible configuration. This version of the module extends functionality by providing 12-bit resolution, a wider range of automatic sampling options and tighter integration with other analog modules, such as the CTMU, and a configurable results buffer. There is a legacy 10-bit mode on this A/D to allow the option to run with lower resolution in order to obtain higher throughput. This module also includes a unique Threshold Detect feature that allows the module itself to make simple decisions based on the conversion results.

A simplified block diagram for the module is illustrated in Figure 19-1.

19.1 A/D Control Registers

The 12-bit A/D Converter module uses up to 43 registers for its operation. All registers are mapped in the data memory space.

19.1.1 CONTROL REGISTERS

Depending on the specific device, the module has up to eleven control and status registers:

- AD1CON1: A/D Control Register 1
- AD1CON2: A/D Control Register 2
- AD1CON3: A/D Control Register 3
- AD1CON5: A/D Control Register 5
- AD1CHS: A/D Sample Select Register
- AD1CHITH and AD1CHITL: A/D Scan Compare Hit Registers
- AD1CSSH and AD1CSSL: A/D Input Scan Select Registers
- AD1CTMENH and AD1CTMENL: CTMU Enable Registers

The AD1CON1, AD1CON2 and AD1CON3 registers (Register 19-1, Register 19-2 and Register 19-3) control the overall operation of the A/D module. This includes enabling the module, configuring the conversion clock and voltage reference sources, selecting the sampling and conversion Triggers, and manually controlling the sample/convert sequences. The AD1CON5 register (Register 19-4) specifically controls features of the Threshold Detect operation, including its function in power-saving modes.

The AD1CHS register (Register 19-5) selects the input channels to be connected to the S/H amplifier. It also allows the choice of input multiplexers and the selection of a reference source for differential sampling.

The AD1CHITH and AD1CHITL registers (Register 19-6 and Register 19-7) are semaphore registers used with Threshold Detect operations. The status of individual bits, or bit pairs in some cases, indicates if a match condition has occurred. AD1CHITL is always implemented, whereas AD1CHITH may not be implemented in devices with 16 or fewer channels.

The AD1CSSH/L registers (Register 19-8 and Register 19-9) select the channels to be included for sequential scanning.

The AD1CTMENH/L registers (Register 19-10 and Register 19-11) select the channel(s) to be used by the CTMU during conversions. Selecting a particular channel allows the A/D Converter to control the CTMU (particularly, its current source) and read its data through that channel. AD1CTMENL is always implemented, whereas AD1CTMENH may not be implemented in devices with 16 or fewer channels.

19.1.2 A/D RESULT BUFFERS

The module incorporates a multi-word, dual port buffer, called ADC1BUFx. Each of the locations is mapped into the data memory space and is separately addressable. The buffer locations are referred to as ADC1BUF0 through ADC1BUFx (x = up to 17).

The A/D result buffers are both readable and writable. When the module is active (AD1CON<15> = 1), the buffers are read-only and store the results of A/D conversions. When the module is inactive (AD1CON<15> = 0), the buffers are both readable and writable. In this state, writing to a buffer location programs a conversion threshold for Threshold Detect operations.

Buffer contents are not cleared when the module is deactivated with the ADON bit (AD1CON1<15>). Conversion results and any programmed threshold values are maintained when ADON is set or cleared.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	2 CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
bit 15							bit 8
D/M/ 0	D/M/ O	D/M/ O	D/M/ 0	DAM 0			
	2 CHONA1			CH0SA3			
bit 7	2 CHUNAI	CHUNAU	0110074	CHUGAS	CHUSAZ	CHUSAT	bit 0
Sit 1							5110
Legend:							
R = Reada	able bit	W = Writable	bit	U = Unimplen	nented bit, read	1 as '0'	
-n = Value	at POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkn	own
bit 15-13	CHONB<2:0> 111 = AN6 ⁽¹⁾ 110 = AN5 ⁽²⁾ 101 = AN4 100 = AN3 011 = AN2 010 = AN1 001 = AN0 000 = AVss	: Sample B Ch	annel 0 Negat	ive Input Select	bits		
	11111 = Unii 11111 = AVE 11101 = AVE 11101 = AVS 11100 = Upp 11011 = Low 11010 = Inte 11000-11007 10001 = No G 10111 = No G 10111 = No G 10111 = Cha 10100 = Cha 10011 = Cha 10001 = Cha 00101 = Cha	mplemented, do pp(3) (3) (3) (3) (3) (3) (4) (4) (4) (4) (5) (5) (5) (5) (5) (5) (5) (5) (5) (5	ail (0.785 * VD rail (0.215 * VD Reference (VE nted, do not us onnected, all ir onnected, all in onnected, all in e correspondir input is AN21 input is AN20 input is AN19 input is AN19 input is AN19 input is AN17 ⁽¹⁾ input is AN1 input is AN1 input is AN1	D) D) 3G)(3) Se aputs are floatin puts are floatin puts are floatin g CTMEN22 (<i>A</i> (2) (2)	g (used for CTI g (used for CTI g (used for CTN AD1CTMENH<	MU) MU) /U temperature 6>) bit)	sensor input);
Note 1: 2:	This is implement This is implement	ted on 44-pin de ted on 28-pin a	evices only. nd 44-pin devid	ces only.			

3: The band gap value used for this input is 2x or 4x the internal VBG, which is selected when PVCFG<1:0> = 1x.

REGISTER 19-5: AD1CHS: A/D SAMPLE SELECT REGISTER

DS30003030B-page 218

19.4 Buffer Data Formats

The A/D conversions are fully differential 12-bit values when MODE12 = 1 (AD1CON1<10>) and 10-bit values when MODE12 = 0. When absolute fractional or absolute integer formats are used, the results are 12 or 10 bits wide, respectively. When signed decimal formatting is used, the conversion also includes a Sign bit, making 12-bit conversions 13 bits wide and 10-bit conversions 11 bits wide. The signed decimal format yields 12-bit and 10-bit values, respectively. The Sign bit (bit 12 or bit 10) is sign-extended to fill the buffer. The FORM<1:0> bits (AD1CON1<9:8>) select the format. Figure 19-4 and Figure 19-5 show the data output formats that can be selected. Table 19-1 through Table 19-4 show the numerical equivalents for the various conversion result codes.

FIGURE 19-4: A/D OUTPUT DATA FORMATS (12-BIT)

RAM Contents:					d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Read to Bus:																
Integer	0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
					r										r	1
Signed Integer	s0	s0	s0	s0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Fractional (1.15)	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0
Signed Fractional (1.15)	s0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0

TABLE 19-1:NUMERICAL EQUIVALENTS OF VARIOUS RESULT CODES:
12-BIT INTEGER FORMATS

VIN/VREF	12-Bit Differential Output Code (13-bit result)	16-Bit Integer Format/16-Bit Signed Integer Format/Equivalent Decimal ValueEquivalent Decimal Value								
+4095/4096	0 1111 1111 1111	0000 1111 1111 1111	+4095	0000 1111 1111 1111	+4095					
+4094/4096	0 1111 1111 1110	0000 1111 1111 1110	+4094	0000 1111 1111 1110	+4094					
	•••									
+1/4096	0 1000 0000 0001	0000 0000 0000 0001	+1	0000 0000 0000 0001	+1					
0/4096	0 0000 0000 0000	0000 0000 0000 0000	0	0000 0000 0000 0000	0					
-1/4096	1 0111 1111 1111	0000 0000 0000 0000	0	1111 1111 1111 1111	-1					
• • •										
-4095/4096	1 0000 0000 0001	0000 0000 0000 0000	0	1111 0000 0000 0001	-4095					
-4096/4096	1 0000 0000 0000	0000 0000 0000 0000	0	1111 0000 0000 0000	-4096					

FIGURE 24-1: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR CAPACITANCE MEASUREMENT



24.2 Measuring Time

Time measurements on the pulse width can be similarly performed using the A/D module's Internal Capacitor (CAD) and a precision resistor for current calibration. Figure 24-2 displays the external connections used for time measurements, and how the CTMU and A/D modules are related in this application. This example also shows both edge events coming from the external CTEDx pins, but other configurations using internal edge sources are possible.

FIGURE 24-2: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT



R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1		
FWDTEN1	WINDIS	FWDTEN0	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0		
bit 7							bit 0		
Legend:									
R = Readab	le bit	P = Programn	nable bit	U = Unimplen	nented bit, read	l as '0'			
-n = Value a	t POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	iown		
bit 7,5	FWDTEN<1:0 11 = WDT is en	>: Watchdog Ti nabled in hardy antrolled with the second secon	mer Enable bi vare	ts					
	10 = WDT is co 01 = WDT is er 00 = WDT is di	nabled only whi isabled in hard	ile the device i ware; SWDTE	s active, WDT is N bit is disabled	s disabled in SI I	eep; SWDTEN	bit is disabled		
bit 6	WINDIS: Winde	owed Watchdo	g Timer Disab	le bit					
	 1 = Standard WDT is selected; windowed WDT is disabled 0 = Windowed WDT is enabled; note that executing a CLRWDT instruction while the WDT is disabled in hardware and software (FWDTEN<1:0> = 00 and SWDTEN (RCON<5>) = 0) will not cause a device Reset 								
bit 4	FWPSA: WDT	Prescaler bit							
	1 = WDT preso 0 = WDT preso	caler ratio of 1: caler ratio of 1:	128 32						
bit 3-0	WDTPS<3:0>:	Watchdog Tim	er Postscale S	Select bits					
	1111 = 1:32,76	68							
	1110 = 1:16,38	34							
	1101 = 1.8, 192 1100 = 1:4.096	5							
	1011 = 1:2,048	3							
	1010 = 1:1,024	4							
	1001 = 1.512 1000 = 1.256								
	0111 = 1:128								
	0110 = 1:64								
	0101 = 1:32								
	0100 = 1.10 0011 = 1.8								
	0010 = 1 :4								
	0001 = 1:2								
	0000 = 1:1								

REGISTER 25-5: FWDT: WATCHDOG TIMER CONFIGURATION REGISTER

25.4 Program Verification and Code Protection

For all devices in the PIC24FXXXXX family, code protection for the Boot Segment is controlled by the Configuration bit, BSS0, and the General Segment by the Configuration bit, GCP. These bits inhibit external reads and writes to the program memory space This has no direct effect in normal execution mode.

Write protection is controlled by bit, BWRP, for the Boot Segment and bit, GWRP, for the General Segment in the Configuration Word. When these bits are programmed to '0', internal write and erase operations to program memory are blocked.

25.5 In-Circuit Serial Programming

PIC24FXXXXX family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock (PGECx) and data (PGEDx), and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

25.6 In-Circuit Debugger

When MPLAB[®] ICD 3, MPLAB REAL ICE[™] or PICkit[™] 3 is selected as a debugger, the in-circuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGECx and PGEDx pins.

To use the in-circuit debugger function of the device, the design must implement ICSP connections to MCLR, VDD, VSS, PGECx, PGEDx and the pin pair. In addition, when the feature is enabled, some of the resources are not available for general use. These resources include the first 80 bytes of data RAM and two I/O pins.

27.1 DC Characteristics





FIGURE 27-2: PIC24F16KM204 FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	Тнідн	Clock High Time	100 kHz mode	4.0	-	μS	Must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μS	Must operate at a minimum of 10 MHz
			MSSPx module	1.5 TCY	_	_	
101	TLOW	Clock Low Time	100 kHz mode	4.7	-	μS	Must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	-	μS	Must operate at a minimum of 10 MHz
			MSSPx module	1.5 TCY	—	_	
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
90	Tsu:sta	Start Condition Setup Time	100 kHz mode	4.7	_	μS	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μS	
91	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	μS	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μS	
106	Thd:dat	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μS	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250		ns	(Note 2)
			400 kHz mode	100	—	ns	
92	Tsu:sto	Stop Condition Setup Time	100 kHz mode	4.7		μS	
			400 kHz mode	0.6	—	μS	
109	ΤΑΑ	Output Valid from Clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	—	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	μS	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μS	
D102	Св	Bus Capacitive Loading		—	400	pF	

TABLE 27-34: I²C[™] BUS DATA REQUIREMENTS (SLAVE MODE)

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCLx to avoid unintended generation of Start or Stop conditions.

2: A Fast mode I²C[™] bus device can be used in a Standard mode I²C bus system, but the requirement, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCLx line is released.