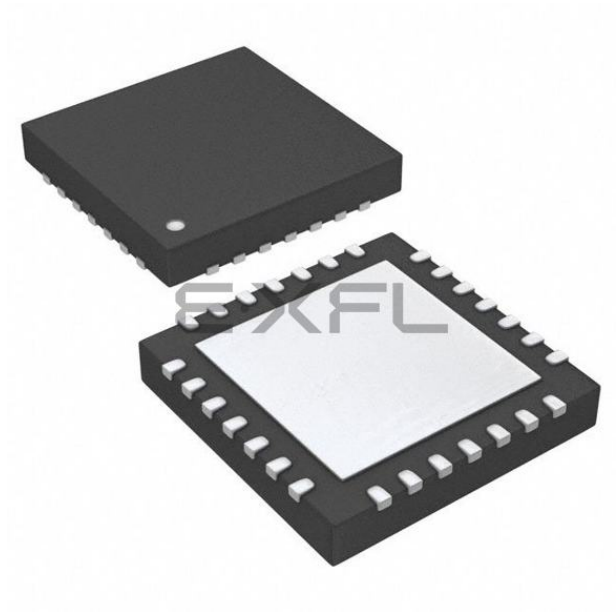


Welcome to E-XFL.COM

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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	32KB (11K 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 13x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f32ka302-e-ml

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
CTPLS	16	24	21	11	12	16	24	21	11	12	O	—	CTMU Pulse Output
HLVDIN	15	23	20	10	11	15	23	20	10	11	I	ST	High/Low-Voltage Detect Input
IC1	14	19	16	6	6	11	19	16	6	6	I	ST	Input Capture 1 Input
IC2	13	18	15	5	5	13	18	15	5	5	I	ST	Input Capture 2 Input
IC3	15	23	20	13	14	15	23	20	13	14	I	ST	Input Capture 3 Input
INT0	11	16	13	43	47	11	16	13	43	47	I	ST	Interrupt 0 Input
INT1	17	25	22	14	15	17	25	22	14	15	I	ST	Interrupt 1 Input
INT2	14	20	17	7	7	15	23	20	10	11	I	ST	Interrupt 2 Input
$\overline{\text{MCLR}}$	1	1	26	18	19	1	1	26	18	19	I	ST	Master Clear (Device Reset) Input (active-low)
OC1	14	20	17	7	7	11	16	13	43	47	O	—	Output Compare/PWM1 Output
OC2	4	22	19	4	4	4	22	19	4	4	O	—	Output Compare/PWM2 Output
OC3	5	21	18	12	13	5	21	18	12	13	O	—	Output Compare/PWM3 Output
OCFA	17	25	22	14	15	17	25	22	14	15	O	—	Output Compare Fault A
OFCB	16	24	21	32	35	16	24	21	32	35	O	—	Output Compare Fault B
OSCI	7	9	6	30	33	7	9	6	30	33	I	ANA	Main Oscillator Input
OSCO	8	10	7	31	34	8	10	7	31	34	O	ANA	Main Oscillator Output
PGEC1	5	5	2	22	24	5	5	2	22	24	I/O	ST	ICSP™ Clock 1
PCED1	4	4	1	21	23	4	4	1	21	23	I/O	ST	ICSP Data 1
PGEC2	2	22	19	19	10	2	22	19	19	10	I/O	ST	ICSP Clock 2
PGED2	3	21	18	8	9	3	21	18	8	9	I/O	ST	ICSP Data 2
PGEC3	10	15	12	42	46	10	15	12	42	46	I/O	ST	ICSP Clock 3
PGED3	9	14	11	41	45	9	14	11	41	45	I/O	ST	ICSP Data 3

REGISTER 3-2: CORCON: CPU CONTROL REGISTER

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

U-0	U-0	U-0	U-0	R/C-0, HSC	R/W-0	U-0	U-0
—	—	—	—	IPL3(1)	PSV	—	—
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable 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 15-4 **Unimplemented:** Read as '0'

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(1)
1 = CPU Interrupt Priority Level is greater than 7
0 = CPU Interrupt Priority Level is 7 or less

bit 2 **PSV:** Program Space Visibility in Data Space Enable bit
1 = Program space is visible in data space
0 = Program space is not visible in data space

bit 1-0 **Unimplemented:** Read as '0'

Note 1: User interrupts are disabled when IPL3 = 1.

3.3 Arithmetic Logic Unit (ALU)

The PIC24F ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. Depending on the operation, the ALU may affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array, or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

The PIC24F CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware division for 16-bit divisor.

3.3.1 MULTIPLIER

The ALU contains a high-speed, 17-bit x 17-bit multiplier. It supports unsigned, signed or mixed sign operation in several multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.3.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

1. 32-bit signed/16-bit signed divide
2. 32-bit unsigned/16-bit unsigned divide
3. 16-bit signed/16-bit signed divide
4. 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. Sixteen-bit signed and unsigned `DIV` instructions can specify any W register for both the 16-bit divisor (Wn), and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

3.3.3 MULTI-BIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multi-bit arithmetic and logic shifts. Multi-bit shifts are implemented using a shifter block, capable of performing up to a 15-bit arithmetic right shift, or up to a 15-bit left shift, in a single cycle. All multi-bit shift instructions only support Register Direct Addressing for both the operand source and result destination.

A full summary of instructions that use the shift operation is provided in Table 3-2.

TABLE 3-2: INSTRUCTIONS THAT USE THE SINGLE AND MULTI-BIT SHIFT OPERATION

Instruction	Description
`ASR`	Arithmetic shift right source register by one or more bits.
`SL`	Shift left source register by one or more bits.
`LSR`	Logical shift right source register by one or more bits.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® MCU devices and improve data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address (EA) calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word, which contains the byte, using the LSB of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and the registers are organized as two parallel, byte-wide entities with shared (word) address decode, but separate write lines. Data byte writes only write to the corresponding side of the array or register, which matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed; if it occurred on a write, the instruction will be executed, but the write will not occur. In either case, a trap is then executed, allowing the system and/or user to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the LSB; the MSB is not modified.

A Sign-Extend (SE) instruction is provided to allow the users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, users can clear the MSB of any W register by executing a Zero-Extend (ZE) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions operate only on words.

4.2.3 NEAR DATA SPACE

The 8-Kbyte area between 0000h and 1FFFh is referred to as the Near Data Space (NDS). Locations in this space are directly addressable via a 13-bit absolute address field within all memory direct instructions. The remainder of the data space is addressable indirectly. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing (MDA) with a 16-bit address field. For PIC24FV32KA304 family devices, the entire implemented data memory lies in Near Data Space.

4.2.4 SFR SPACE

The first 2 Kbytes of the Near Data Space, from 0000h to 07FFh, are primarily occupied with Special Function Registers (SFRs). These are used by the PIC24F core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by the module. Much of the SFR space contains unused addresses; these are read as '0'. The SFR space, where the SFRs are actually implemented, is provided in Table 4-2. Each implemented area indicates a 32-byte region, where at least one address is implemented as an SFR. A complete listing of implemented SFRs, including their addresses, is provided in Table 4-3 through Table 4-25.

TABLE 4-2: IMPLEMENTED REGIONS OF SFR DATA SPACE

	SFR Space Address							
	xx00	**xx20**	**xx40**	**xx60**	**xx80**	**xxA0**	**xxC0**	**xxE0**
000h	Core			ICN	Interrupts			—
100h	Timers		Capture	—	Compare	—	—	—
200h	I^2C™	UART	SPI		—	—	I/O	
300h	A/D/CMTU				—	—	—	—
400h	—	—	—	—	—	—	—	—
500h	—	—	—	—	—	—	—	—
600h	—	RTC/Comp	CRC	—	—			
700h	—	—	System/DS/HLVD	NVM/PMD	—	—	—	—

Legend: — = No implemented SFRs in this block.

TABLE 4-3: CPU CORE REGISTERS MAP

File Name	Start 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
WREG0	0000	WREG0																0000
WREG1	0002	WREG1																0000
WREG2	0004	WREG2																0000
WREG3	0006	WREG3																0000
WREG4	0008	WREG4																0000
WREG5	000A	WREG5																0000
WREG6	000C	WREG6																0000
WREG7	000E	WREG7																0000
WREG8	0010	WREG8																0000
WREG9	0012	WREG9																0000
WREG10	0014	WREG10																0000
WREG11	0016	WREG11																0000
WREG12	0018	WREG12																0000
WREG13	001A	WREG13																0000
WREG14	001C	WREG14																0000
WREG15	001E	WREG15																0000
SPLIM	0020	SPLIM																xxxx
PCL	002E	PCL																0000
PCH	0030	—	—	—	—	—	—	—	—	—	PCH							0000
TBLPAG	0032	—	—	—	—	—	—	—	—	TBLPAG								0000
PSVPAG	0034	—	—	—	—	—	—	—	—	PSVPAG								0000
RCOUNT	0036	RCOUNT																xxxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—	0000
DISICNT	0052	—	—	DISICNT														xxxx

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

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;  // Global 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 = 0x4001;                                         // Initialize NVMCON

   //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

   //Perform TBLWT instructions to write necessary number of latches
   for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)
   {
       __builtin_tblwtl(offset, progData[i++]);             // Write to address low word
       __builtin_tblwth(offset, progData[i]);               // Write to upper byte
       offset = offset + 2;                                 // Increment address
   }
```

EXAMPLE 5-5: INITIATING A PROGRAMMING SEQUENCE – ASSEMBLY LANGUAGE CODE

```
        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                             ; 2 NOPs required after setting WR
        NOP                             ;
        BTSC    NVMCON, #15             ; Wait for the sequence to be completed
        BRA     $-2                     ;
```

EXAMPLE 5-6: INITIATING A PROGRAMMING SEQUENCE – 'C' LANGUAGE CODE

```
// C example using MPLAB C30

asm("DISI #5");               // Block all interrupts for next 5 instructions

__builtin_write_NVM();        // Perform unlock sequence and set WR
```

REGISTER 14-1: ICxCON1: INPUT CAPTURE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable 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 15-14 **Unimplemented:** Read as '0'

bit 13 **ICSIDL:** Input Capture x Module Stop in Idle Control bit
1 = Input capture module halts in CPU Idle mode
0 = Input capture module continues to operate in CPU Idle mode

bit 12-10 **ICTSEL<2:0>:** Input Capture x Timer Select bits
111 = System clock (FOSC/2)
110 = Reserved
101 = Reserved
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer2
000 = Timer3

bit 9-7 **Unimplemented:** Read as '0'

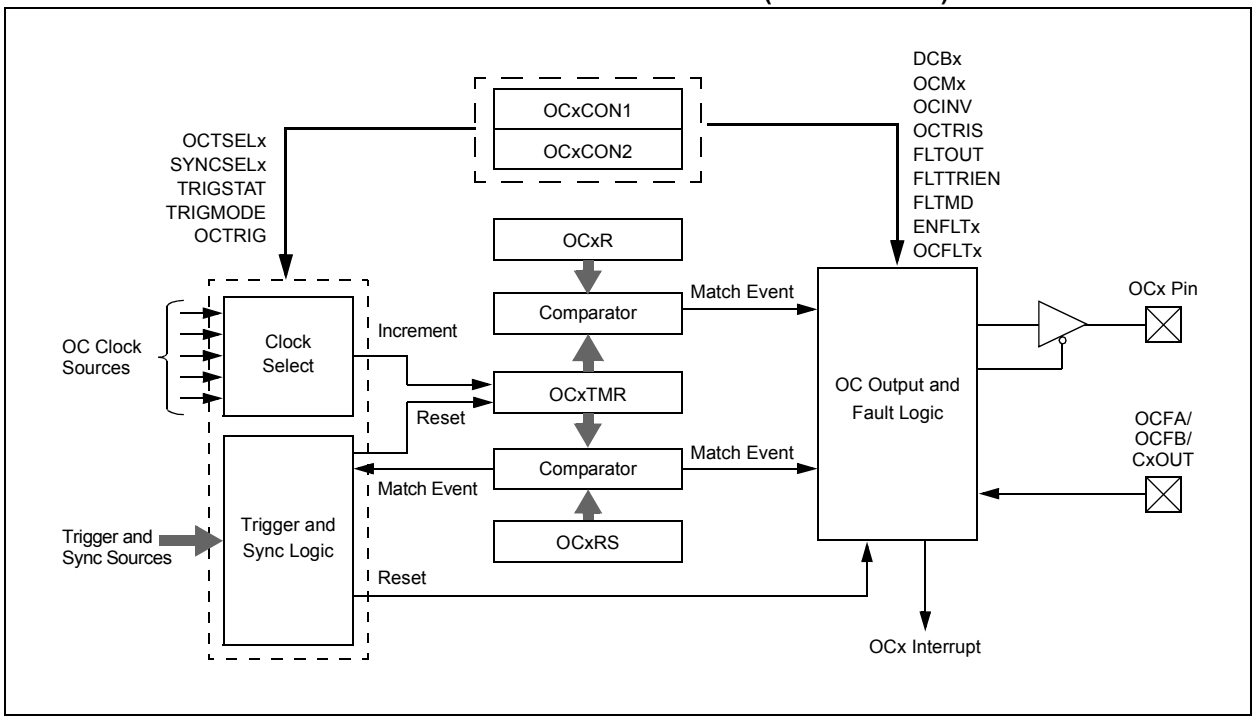
bit 6-5 **ICI<1:0>:** Select Number of Captures per Interrupt bits
11 = Interrupt on every fourth capture event
10 = Interrupt on every third capture event
01 = Interrupt on every second capture event
00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture x Overflow Status Flag bit (read-only)
1 = Input capture overflow occurred
0 = No input capture overflow occurred

bit 3 **ICBNE:** Input Capture x Buffer Empty Status bit (read-only)
1 = Input capture buffer is not empty, at least one more capture value can be read
0 = Input capture buffer is empty

bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits
111 = Interrupt mode: Input capture functions as an interrupt pin only when the device is in Sleep or Idle mode (rising edge detect only, all other control bits are not applicable)
110 = Unused (module disabled)
101 = Prescaler Capture mode: Capture on every 16th rising edge
100 = Prescaler Capture mode: Capture on every 4th rising edge
011 = Simple Capture mode: Capture on every rising edge
010 = Simple Capture mode: Capture on every falling edge
001 = Edge Detect Capture mode: Capture on every edge (rising and falling); ICI<1:0 bits do not control interrupt generation for this mode
000 = Input capture module is turned off

FIGURE 15-1: OUTPUT COMPARE x BLOCK DIAGRAM (16-BIT MODE)



REGISTER 15-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1 (CONTINUED)

bit 2-0 **OCM<2:0>:** Output Compare x Mode Select bits(1)

111 = Center-Aligned PWM mode on OCx
110 = Edge-Aligned PWM mode on OCx
101 = Double Compare Continuous Pulse mode: Initialize OCx pin low; toggle OCx state continuously on alternate matches of OCxR and OCxRS
100 = Double Compare Single-Shot mode: Initialize OCx pin low; toggle OCx state on matches of OCxR and OCxRS for one cycle
011 = Single Compare Continuous Pulse mode: Compare events continuously toggle the OCx pin
010 = Single Compare Single-Shot mode: Initialize OCx pin high; compare event forces the OCx pin low
001 = Single Compare Single-Shot mode: Initialize OCx pin low, compare event forces the OCx pin high
000 = Output compare channel is disabled

Note 1: The comparator module used for Fault input varies with the OCx module. OC1 and OC2 use Comparator 1; OC3 and OC4 use Comparator 2; OC5 uses Comparator 3.

19.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

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 Real-Time Clock and Calendar, refer to the *"PIC24F Family Reference Manual"*, **Section 29. "Real-Time Clock and Calendar (RTCC)"** (DS39696).

The RTCC provides the user with a Real-Time Clock and Calendar (RTCC) function that can be calibrated.

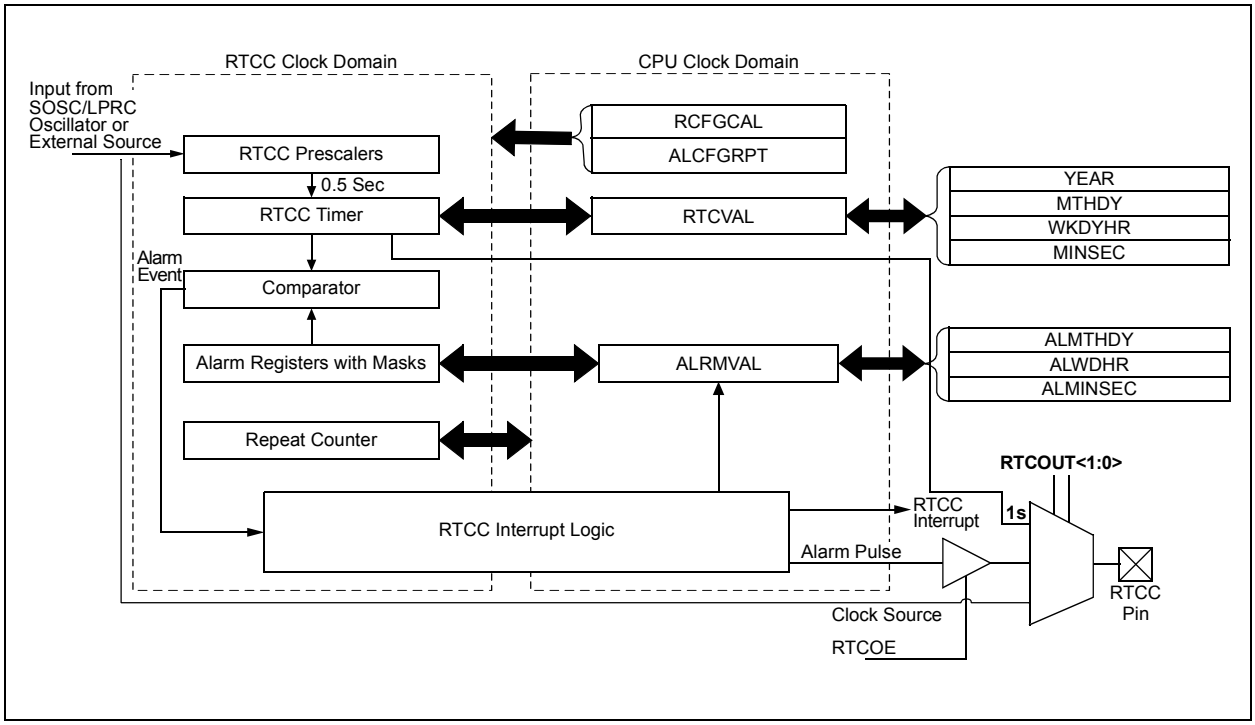
Key features of the RTCC module are:

- Operates in Deep Sleep mode
- Selectable clock source
- Provides hours, minutes and seconds using 24-hour format
- Visibility of one half second period
- Provides calendar – weekday, date, month and year
- Alarm-configurable for half a second, one second, 10 seconds, one minute, 10 minutes, one hour, one day, one week, one month or one year
- Alarm repeat with decrementing counter
- Alarm with indefinite repeat chime
- Year 2000 to 2099 leap year correction
- BCD format for smaller software overhead
- Optimized for long-term battery operation
- User calibration of the 32.768 kHz clock crystal/32K INTRC frequency with periodic auto-adjust
- Optimized for long-term battery operation
- Fractional second synchronization
- Calibration to within ±2.64 seconds error per month
- Calibrates up to 260 ppm of crystal error
- Ability to periodically wake-up external devices without CPU intervention (external power control)
- Power control output for external circuit control
- Calibration takes effect every 15 seconds
- Runs from any one of the following:
  - External Real-Time Clock of 32.768 kHz
  - Internal 31.25 kHz LPRC Clock
  - 50 Hz or 60 Hz External Input

19.1 RTCC Source Clock

The user can select between the SOSC crystal oscillator, LPRC internal oscillator or an external 50 Hz/60 Hz power line input as the clock reference for the RTCC module. This gives the user an option to trade off system cost, accuracy and power consumption, based on the overall system needs.

FIGURE 19-1: RTCC BLOCK DIAGRAM



REGISTER 21-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	—	HLSIDL	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
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 **HLVDEN:** High/Low-Voltage Detect Power Enable bit
1 = HLVD is enabled
0 = HLVD is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **HLSIDL:** HLVD Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-8 **Unimplemented:** Read as '0'

bit 7 **VDIR:** Voltage Change Direction Select bit
1 = Event occurs when voltage equals or exceeds trip point (HLVDL<3:0>)
0 = Event occurs when voltage equals or falls below trip point (HLVDL<3:0>)

bit 6 **BGVST:** Band Gap Voltage Stable Flag bit
1 = Indicates that the band gap voltage is stable
0 = Indicates that the band gap voltage is unstable

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit
1 = Indicates that the internal reference voltage is stable and the high-voltage detect logic generates the interrupt flag at the specified voltage range
0 = Indicates that the internal reference voltage is unstable and the high-voltage detect logic will not generate the interrupt flag at the specified voltage range, and the HLVD interrupt should not be enabled

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

bit 3-0 **HLVDL<3:0>:** High/Low-Voltage Detection Limit bits
1111 = External analog input is used (input comes from the HLVDIN pin)
1110 = Trip Point 1(1)
1101 = Trip Point 2(1)
1100 = Trip Point 3(1)
.
.
.
0000 = Trip Point 15(1)

Note 1: For the actual trip point, see **Section 29.0 "Electrical Characteristics"**.

REGISTER 22-10: AD1CTMUENH: A/D CTMU ENABLE REGISTER (HIGH WORD)(1)

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

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	CTMEN17	CTMEN16
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 15-2 **Unimplemented:** Read as '0'.

bit 1-0 **CTMEN<17:16>:** CTMU Enabled During Conversion bits

1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: Unimplemented channels are read as '0'.

REGISTER 22-11: AD1CTMUENL: A/D CTMU ENABLE REGISTER (LOW WORD)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN15	CTMEN14	CTMEN13	CTMEN12	CTMUEN11	CTMEN10	CTMEN9	CTMEN8
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	R/W-0
CTMEN7	CTMEN6	CTMEN5	CTMEN4	CTMEN3	CTMEN2	CTMEN1	CTMEN0
bit 7							bit 0

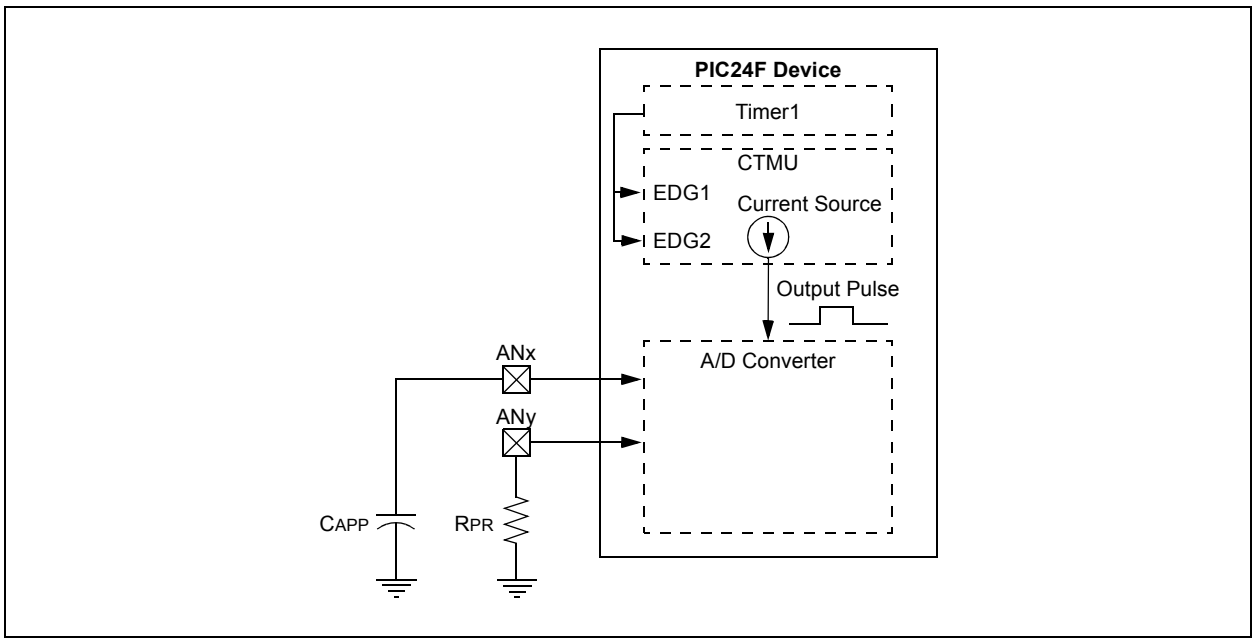
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **CTMEN<15:0>:** CTMU Enabled During Conversion bits

1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: Unimplemented channels are read as '0'.

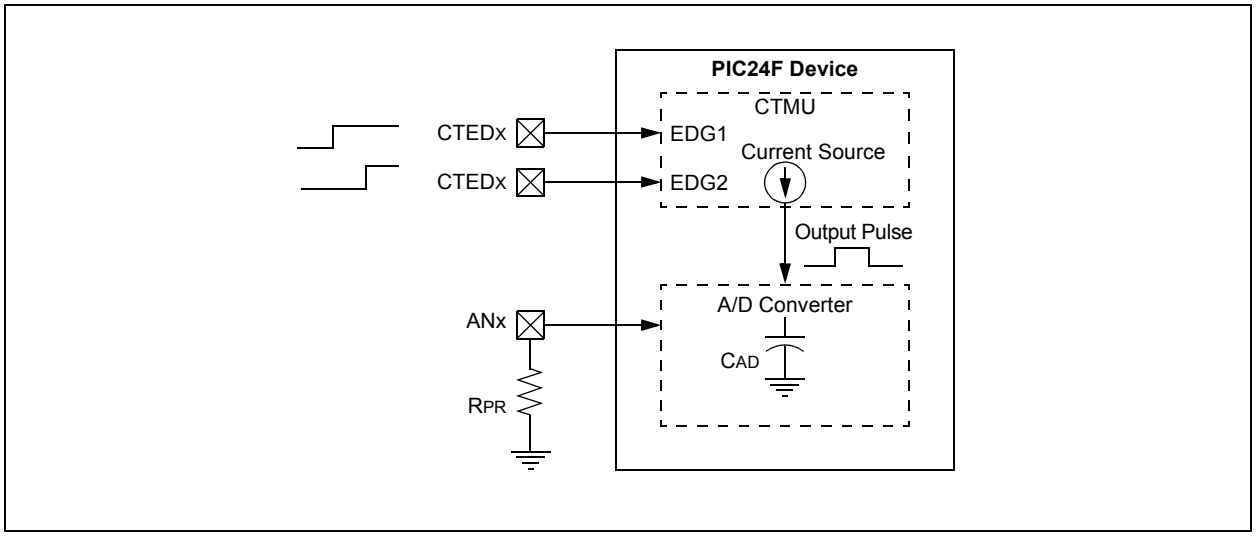
FIGURE 25-1: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR CAPACITANCE MEASUREMENT



25.2 Measuring Time

Time measurements on the pulse width can be similarly performed using the A/D module's Internal Capacitor (C_{AD}) and a precision resistor for current calibration. Figure 25-2 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 25-2: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT



26.4 Deep Sleep Watchdog Timer (DSWDT)

In PIC24FV32KA304 family devices, in addition to the WDT module, a DSWDT module is present which runs while the device is in Deep Sleep, if enabled. It is driven by either the SOSC or LPRC oscillator. The clock source is selected by the Configuration bit, DSWDTOSC (FDS<4>).

The DSWDT can be configured to generate a time-out, at 2.1 ms to 25.7 days, by selecting the respective postscaler. The postscaler can be selected by the Configuration bits, DSWDTPS<3:0> (FDS<3:0>). When the DSWDT is enabled, the clock source is also enabled.

DSWDT is one of the sources that can wake-up the device from Deep Sleep mode.

26.5 Program Verification and Code Protection

For all devices in the PIC24FV32KA304 family, code protection for the boot segment is controlled by the Configuration bit, BSS0, and the general segment by the Configuration bit, GSS0. 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.

26.6 In-Circuit Serial Programming

PIC24FV32KA304 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.

26.7 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 $\overline{\text{MCLR}}$, V_{DD}, V_{SS}, 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.

28.0 INSTRUCTION SET SUMMARY

Note: This chapter is a brief summary of the PIC24F instruction set architecture and is not intended to be a comprehensive reference source.

The PIC24F instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from previous PIC MCU instruction sets. Most instructions are a single program memory word. Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction. The instruction set is highly orthogonal and is grouped into four basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

Table 28-1 lists the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 28-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand, which is typically a register ‘Wb’ without any address modifier
- The second source operand, which is typically a register ‘Ws’ with or without an address modifier
- The destination of the result, which is typically a register ‘Wd’ with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value, ‘f’
- The destination, which could either be the file register, ‘f’, or the W0 register, which is denoted as ‘WREG’

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of ‘Ws’ or ‘f’)
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register ‘Wb’)

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of ‘k’)
- The W register or file register where the literal value is to be loaded (specified by ‘Wb’ or ‘f’)

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand, which is a register ‘Wb’ without any address modifier
- The second source operand, which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register ‘Wd’ with or without an address modifier

The control instructions may use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

All instructions are a single word, except for certain double-word instructions, which were made double-word instructions so that all of the required information is available in these 48 bits. In the second word, the 8 MSbs are ‘0’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the Program Counter (PC) is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all table reads and writes, and RETURN/RETFIE instructions, which are single-word instructions but take two or three cycles.

Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

TABLE 29-9: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX 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**
	VIL	**Input Low Voltage(4)**					
DI10		I/O Pins	VSS	—	0.2 VDD	V	
DI15		$\overline{MCLR}$	VSS	—	0.2 VDD	V	
DI16		OSCI (XT mode)	VSS	—	0.2 VDD	V	
DI17		OSCI (HS mode)	VSS	—	0.2 VDD	V	
DI18		I/O Pins with I2C™ Buffer	VSS	—	0.3 VDD	V	SMBus is disabled
DI19		I/O Pins with SMBus Buffer	VSS	—	0.8	V	SMBus is enabled
	VIH	**Input High Voltage(4)**					
DI20		I/O Pins:					
		with Analog Functions	0.8 VDD	—	VDD	V	
		Digital Only	0.8 VDD	—	VDD	V	
DI25		$\overline{MCLR}$	0.8 VDD	—	VDD	V	
DI26		OSCI (XT mode)	0.7 VDD	—	VDD	V	
DI27		OSCI (HS mode)	0.7 VDD	—	VDD	V	
DI28		I/O Pins with I2C Buffer:					
		with Analog Functions	0.7 VDD	—	VDD	V	
		Digital Only	0.7 VDD	—	VDD	V	
DI29		I/O Pins with SMBus	2.1	—	VDD	V	2.5V ≤ VPIN ≤ VDD
DI30	ICNPU	**CNx Pull-up Current**	50	250	500	µA	VDD = 3.3V, VPIN = VSS
	IIL	**Input Leakage Current(2,3)**					
DI50		I/O Ports	—	0.05	0.1	µA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI55		$\overline{MCLR}$	—	—	0.1	µA	VSS ≤ VPIN ≤ VDD
DI56		OSCI	—	—	5	µA	VSS ≤ VPIN ≤ VDD, XT and HS modes

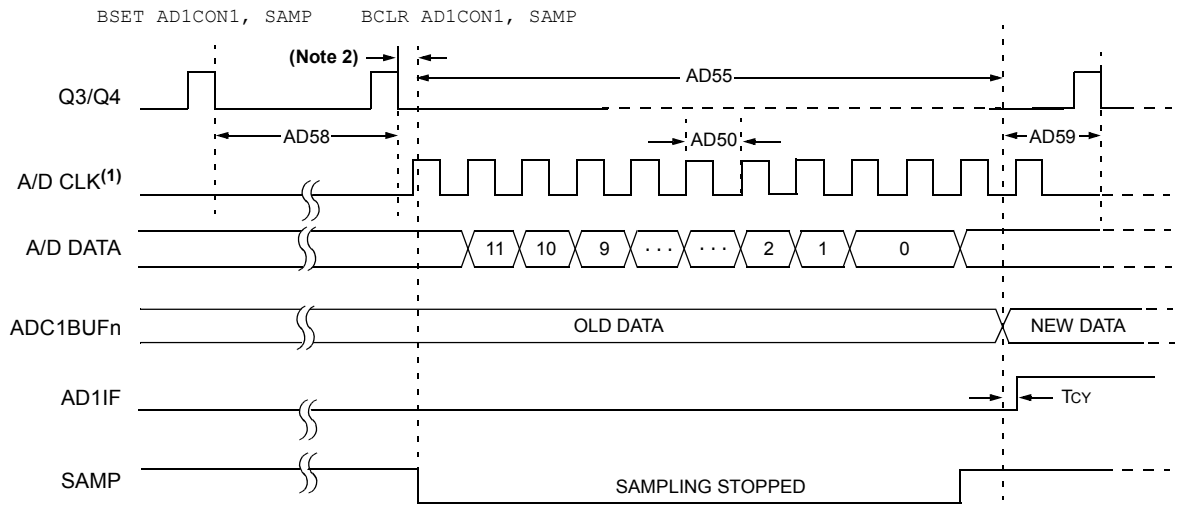
Note 1: Data in "Typ" column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: Refer to Table 1-3 for I/O pin buffer types.

FIGURE 29-22: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

2: This is a minimal RC delay (typically 100 ns) which also disconnects the holding capacitor from the analog input.

TABLE 29-41: A/D CONVERSION TIMING REQUIREMENTS(1)

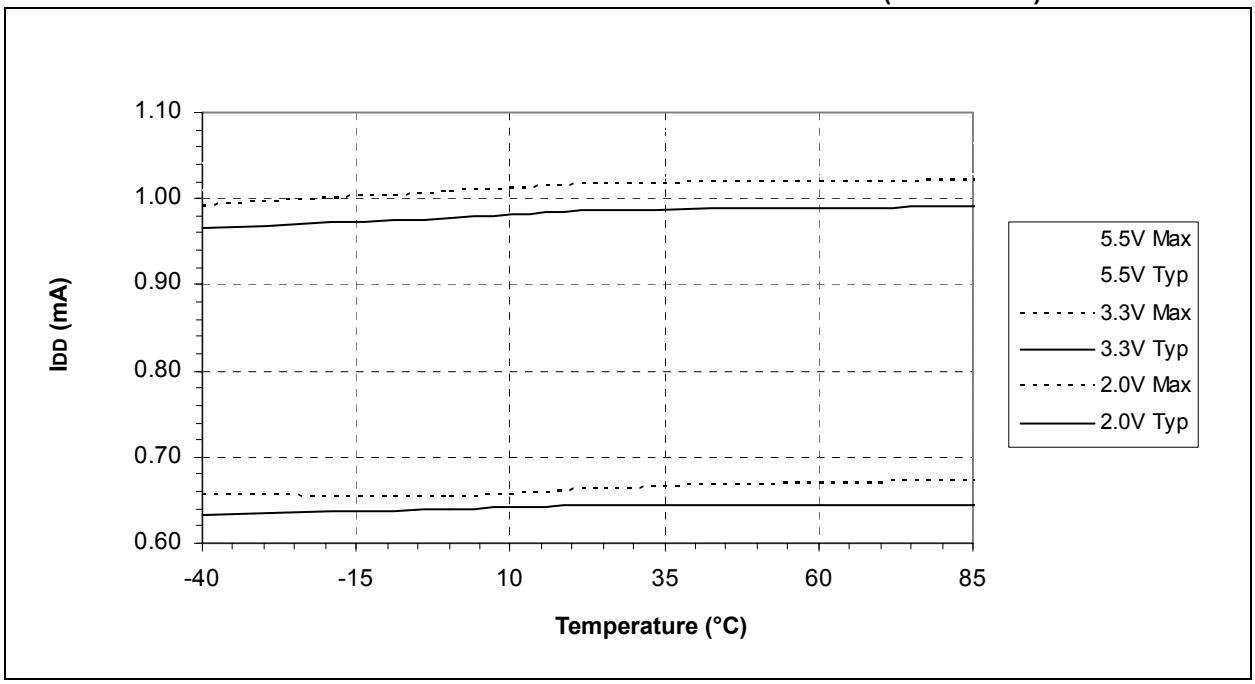
AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Clock Parameters							
AD50	TAD	A/D Clock Period	600	—	—	ns	TCY = 75 ns, AD1CON3 in default state
AD51	TRC	A/D Internal RC Oscillator Period	—	1.67	—	µs	
Conversion Rate							
AD55	TCONV	Conversion Time	— —	12 14	— —	TAD TAD	10-bit results 12-bit results
AD56	FCNV	Throughput Rate	—	—	100	ksps	
AD57	TSAMP	Sample Time	—	1	—	TAD	
AD58	TACQ	Acquisition Time	750	—	—	ns	**(Note 2)**
AD59	TSWC	Switching Time from Convert to Sample	—	—	**(Note 3)**		
AD60	TDIS	Discharge Time	12	—	—	TAD	
Clock Parameters							
AD61	TPSS	Sample Start Delay from Setting Sample bit (SAMP)	2	—	3	TAD	

Note 1: Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

2: The time for the holding capacitor to acquire the “New” input voltage when the voltage changes full scale after the conversion (VDD to VSS or VSS to VDD).

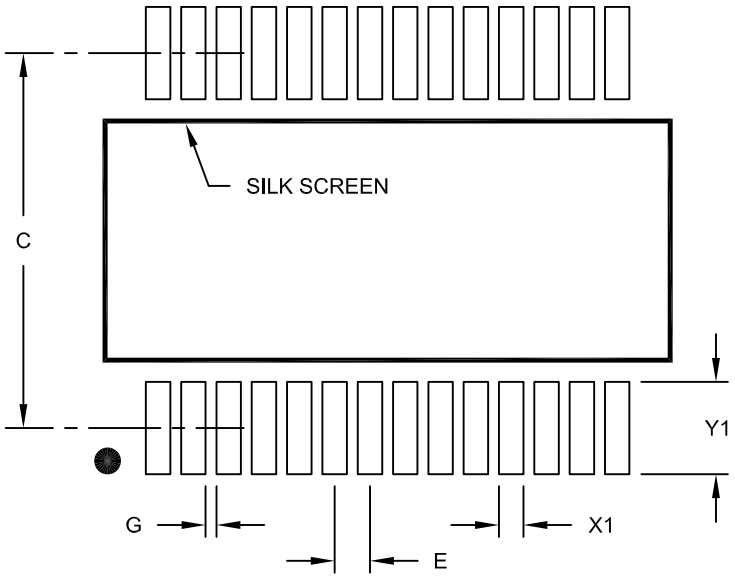
3: On the following cycle of the device clock.

FIGURE 30-9: TYPICAL AND MAXIMUM I_{IDLE} vs. TEMPERATURE (FRC MODE)



28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

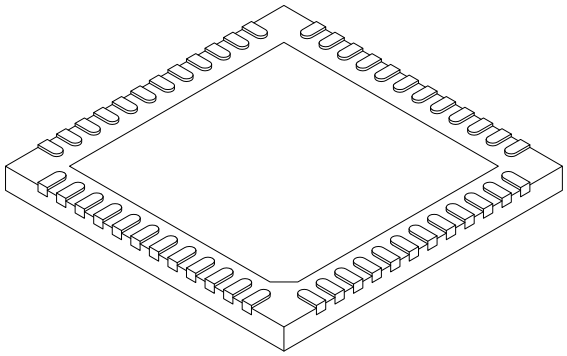
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 6x6x0.5 mm Body [UQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		48	
Pitch	e		0.40 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	4.45	4.60	4.75
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	4.45	4.60	4.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-153A Sheet 2 of 2