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	17
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f08kl201t-i-ss

NOTES:

TABLE 4-16: SYSTEM 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
RCON	0740	TRAPR	IOPUWR	SBOREN	—	—	—	CM	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	(Note 1)
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN	(Note 2)
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	—	3100
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—	—	—	0000
HLVDCON	0756	HLVDEN	—	HLSIDL	—	—	—	—	—	VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0000

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

Note 1: RCON register Reset values are dependent on the type of Reset.

2: OSCCON register Reset values are dependent on configuration fuses and by type of Reset.

TABLE 4-17: NVM 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
NVMCON	0760	WR	WREN	WRERR	PGMONLY	—	—	—	—	—	ERASE	NVMOP5	NVMOP4	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000
NVMKEY	0766	—	—	—	—	—	—	—	—	NVM Key Register								0000

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

TABLE 4-18: ULTRA LOW-POWER WAKE-UP 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
ULPWCON	0768	ULPEN	—	ULPSIDL	—	—	—	—	ULPSINK	—	—	—	—	—	—	—	—	0000

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

TABLE 4-19: PMD 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
PMD1	0770	—	T4MD	T3MD	T2MD	T1MD	—	—	—	SSP1MD	U2MD	U1MD	—	—	—	—	ADC1MD	0000
PMD2	0772	—	—	—	—	—	—	—	—	—	—	—	—	—	CCP3MD	CCP2MD	CCP1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	—	—	—	—	—	—	—	—	SSP2MD	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	ULPWUMD	—	—	EEMD	REFOMD	—	HLVDMD	—	0000

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

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

```
// C example using MPLAB C30

  int __attribute__ ((space(auto_psv))) progAddr = &progAddr;  // Global variable located in Pgm Memory
  unsigned int offset;

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

  TBLPAG = __builtin_tblpage(&progAddr);                      // Initialize PM Page Boundary SFR
  offset = &progAddr & 0xFFFF;                                // 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
```

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

```
; Set up NVMCON for row programming operations
       MOV     #0x4004, W0                 ;
       MOV     W0, NVMCON                  ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000, W0                 ;
       MOV     W0, TBLPAG                  ; Initialize PM Page Boundary SFR
       MOV     #0x6000, W0                 ; An example program memory address
; Perform the TBLWT instructions to write the 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
; 1st_program_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_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     #HIGH_BYTE_31, W3           ;
       TBLWTL  W2, [W0]                    ; Write PM low word into program latch
       TBLWTH  W3, [W0]                    ; Write PM high byte into program latch
```

7.2 Device Reset Times

The Reset times for various types of device Reset are summarized in Table 7-3. Note that the System Reset Signal, $\overline{SYSRST}$, is released after the POR and PWRT delay times expire.

The time at which the device actually begins to execute code will also depend on the system oscillator delays, which include the Oscillator Start-up Timer (OST) and the PLL lock time. The OST and PLL lock times occur in parallel with the applicable $\overline{SYSRST}$ delay times.

The FSCM delay determines the time at which the FSCM begins to monitor the system clock source after the $\overline{SYSRST}$ signal is released.

TABLE 7-3: RESET DELAY TIMES FOR VARIOUS DEVICE RESETS

Reset Type	Clock Source	$\overline{SYSRST}$ Delay	System Clock Delay	Notes
POR(6)	EC	TPOR + TPWRT	—	1, 2
	FRC, FRCDIV	TPOR + TPWRT	TFRC	1, 2, 3
	LPRC	TPOR + TPWRT	TLPRC	1, 2, 3
	ECPLL	TPOR + TPWRT	TLOCK	1, 2, 4
	FRCPLL	TPOR + TPWRT	TFRC + TLOCK	1, 2, 3, 4
	XT, HS, SOSC	TPOR+ TPWRT	TOST	1, 2, 5
	XTPLL, HSPLL	TPOR + TPWRT	TOST + TLOCK	1, 2, 4, 5
BOR	EC	TPWRT	—	2
	FRC, FRCDIV	TPWRT	TFRC	2, 3
	LPRC	TPWRT	TLPRC	2, 3
	ECPLL	TPWRT	TLOCK	2, 4
	FRCPLL	TPWRT	TFRC + TLOCK	2, 3, 4
	XT, HS, SOSC	TPWRT	TOST	2, 5
	XTPLL, HSPLL	TPWRT	TFRC + TLOCK	2, 3, 4
All Others	Any Clock	—	—	None

Note 1: TPOR = Power-on Reset delay.

2: TPWRT = 64 ms nominal if the Power-up Timer is enabled; otherwise, it is zero.

3: TFRC and TLPRC = RC oscillator start-up times.

4: TLOCK = PLL lock time.

5: TOST = Oscillator Start-up Timer (OST). A 10-bit counter waits 1024 oscillator periods before releasing the oscillator clock to the system.

6: If Two-Speed Start-up is enabled, regardless of the primary oscillator selected, the device starts with FRC, and in such cases, FRC start-up time is valid.

Note: For detailed operating frequency and timing specifications, see **Section 26.0 "Electrical Characteristics"**.

REGISTER 8-2: 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	R/W-0	U-0	U-0
—	—	—	—	IPL3(2)	PSV(1)	—	—
bit 7							bit 0

Legend:	C = 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(2)
1 = CPU Interrupt Priority Level is greater than 7
0 = CPU Interrupt Priority Level is 7 or less

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

Note 1: See Register 3-2 for the description of this bit, which is not dedicated to interrupt control functions.
2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

Note: Bit 2 is described in **Section 3.0 "CPU"**.

9.0 OSCILLATOR CONFIGURATION

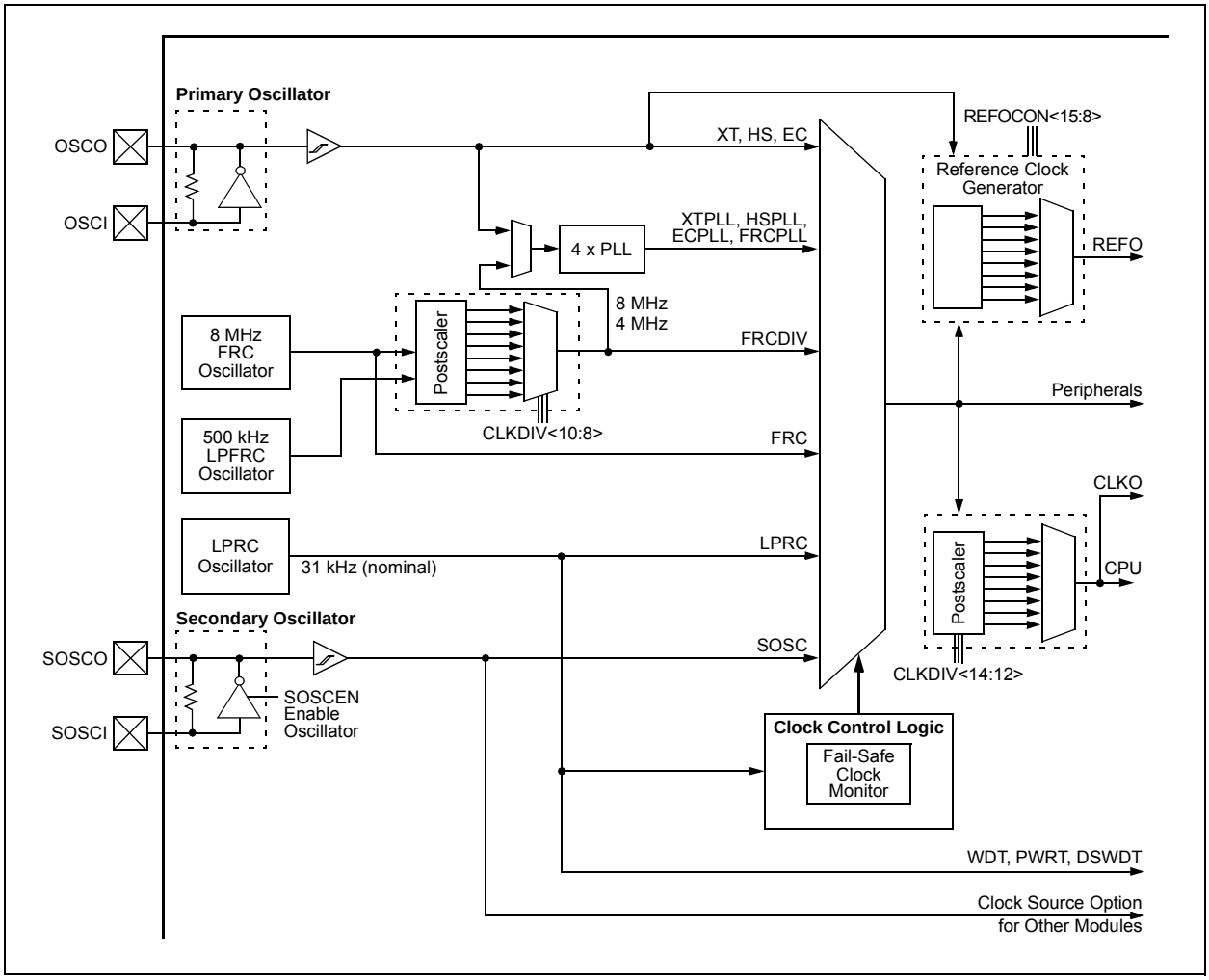
Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on Oscillator Configuration, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Oscillator with 500 kHz Low-Power FRC"** (DS39726).

The oscillator system for the PIC24F16KL402 family of devices has the following features:

- A total of five external and internal oscillator options as clock sources, providing 11 different clock modes.
- On-chip, 4x Phase Locked Loop (PLL) to boost internal operating frequency on select internal and external oscillator sources.
- Software-controllable switching between various clock sources.
- Software-controllable postscaler for selective clocking of CPU for system power savings.
- System frequency range declaration bits for EC mode. When using an external clock source, the current consumption is reduced by setting the declaration bits to the expected frequency range.
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown.

A simplified diagram of the oscillator system is shown in Figure 9-1.

FIGURE 9-1: PIC24F16KL402 FAMILY CLOCK DIAGRAM



The following code sequence for a clock switch is recommended:

1. Disable interrupts during the OSCCON register unlock and write sequence.
2. Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8>, in two back-to-back instructions.
3. Write the new oscillator source to the NOSCx bits in the instruction immediately following the unlock sequence.
4. Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0>, in two back-to-back instructions.
5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
6. Continue to execute code that is not clock-sensitive (optional).
7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
8. Check to see if OSWEN is '0'. If it is, the switch was successful. If OSWEN is still set, then check the LOCK bit to determine the cause of failure.

The core sequence for unlocking the OSCCON register and initiating a clock switch is shown in Example 9-1.

EXAMPLE 9-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV        #OSCCONH, w1
MOV        #0x78, w2
MOV        #0x9A, w3
MOV.b      w2, [w1]
MOV.b      w3, [w1]
;Set new oscillator selection
MOV.b      WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV        #OSCCONL, w1
MOV        #0x46, w2
MOV        #0x57, w3
MOV.b      w2, [w1]
MOV.b      w3, [w1]
;Start oscillator switch operation
BSET       OSCCON,#0
```

9.5 Reference Clock Output

In addition to the CLKO output (FOSC/2) available in certain oscillator modes, the device clock in the PIC24F16KL402 family devices can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 9-4). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for one of the primary modes (EC, HS or XT). Therefore, if the ROSEL bit is also not set, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

REGISTER 10-1: ULPWCON: ULPWU CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ULPEN	—	ULPSIDL	—	—	—	—	ULPSINK
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 = 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 **ULPEN:** ULPWU Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ULPSIDL:** ULPWU Stop in Idle Select bit
1 = Discontinues module operation when the device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-9 **Unimplemented:** Read as ‘0’

bit 8 **ULPSINK:** ULPWU Current Sink Enable bit
1 = Current sink is enabled
0 = Current sink is disabled

bit 7-0 **Unimplemented:** Read as ‘0’

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	T1ECS1(1)	T1ECS0(1)
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
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 **TON:** Timer1 On bit
1 = Starts 16-bit Timer1
0 = Stops 16-bit Timer1

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

bit 13 **TSIDL:** Timer1 Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-10 **Unimplemented:** Read as '0'

bit 9-8 **T1ECS <1:0>:** Timer1 Extended Clock Select bits(1)
11 = Reserved; do not use
10 = Timer1 uses the LPRC as the clock source
01 = Timer1 uses the external clock from T1CK
00 = Timer1 uses the Secondary Oscillator (SOSC) as the clock source

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

bit 6 **TGATE:** Timer1 Gated Time Accumulation Enable bit
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **Unimplemented:** Read as '0'

bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit
When TCS = 1:
1 = Synchronizes external clock input
0 = Does not synchronize external clock input
When TCS = 0:
This bit is ignored.

bit 1 **TCS:** Timer1 Clock Source Select bit
1 = Timer1 clock source is selected by T1ECS<1:0>
0 = Internal clock (FOSC/2)

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

Note 1: The T1ECSx bits are valid only when TCS = 1.

REGISTER 16-3: ECCP1AS: ECCP1 AUTO-SHUTDOWN CONTROL REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
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 15-8 **Unimplemented:** Read as '0'

bit 7 **ECCPASE:** ECCP1 Auto-Shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in a shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCPAS<2:0>:** ECCP1 Auto-Shutdown Source Select bits
111 = VIL on $\overline{FLT0}$ pin, or either C1OUT or C2OUT is high
110 = VIL on $\overline{FLT0}$ pin or C2OUT comparator output is high
101 = VIL on $\overline{FLT0}$ pin or C1OUT comparator output is high
100 = VIL on $\overline{FLT0}$ pin
011 = Either C1OUT or C2OUT is high
010 = C2OUT comparator output is high
001 = C1OUT comparator output is high
000 = Auto-shutdown is disabled

bit 3-2 **PSSAC<1:0>:** P1A and P1C Pins Shutdown State Control bits
1x = P1A and P1C pins tri-state
01 = Drive pins, P1A and P1C, to '1'
00 = Drive pins, P1A and P1C, to '0'

bit 1-0 **PSSBD<1:0>:** P1B and P1D Pins Shutdown State Control bits
1x = P1B and P1D pins tri-state
01 = Drive pins, P1B and P1D, to '1'
00 = Drive pins, P1B and P1D, to '0'

Note 1: This register is implemented only on PIC24FXXKL40X/30X devices.

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.
2: Writing to the ECCPASE bit is disabled while an auto-shutdown condition persists.
3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

REGISTER 16-5: PSTR1CON: ECCP1 PULSE STEERING CONTROL REGISTER(1)

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

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA
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 15-8 **Unimplemented:** Read as '0'

bit 7-6 **CMPL<1:0>:** Complementary Mode Output Assignment Steering bits
- 00 = Complementary output assignment is disabled; the STR<D:A> bits are used to determine Steering mode
- 01 = P1A and P1B are selected as the complementary output pair
- 10 = P1A and P1C are selected as the complementary output pair
- 11 = P1A and P1D are selected as the complementary output pair

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

bit 4 **STRSYNC:** Steering Sync bit
- 1 = Output steering update occurs on the next PWM period
- 0 = Output steering update occurs at the beginning of the instruction cycle boundary

bit 3 **STRD:** Steering Enable D bit
- 1 = P1D pin has the PWM waveform with polarity control from CCP1M<1:0>
- 0 = P1D pin is assigned to port pin

bit 2 **STRC:** Steering Enable C bit
- 1 = P1C pin has the PWM waveform with polarity control from CCP1M<1:0>
- 0 = P1C pin is assigned to port pin

bit 1 **STRB:** Steering Enable B bit
- 1 = P1B pin has the PWM waveform with polarity control from CCP1M<1:0>
- 0 = P1B pin is assigned to port pin

bit 0 **STRA:** Steering Enable A bit
- 1 = P1A pin has the PWM waveform with polarity control from CCP1M<1:0>
- 0 = P1A pin is assigned to port pin

Note 1: This register is only implemented on PIC24FXXKL40X/30X devices. In addition, PWM Steering mode is available only when CCP1M<3:2> = 11 and PM<1:0> = 00.

REGISTER 17-2: SSPxSTAT: MSSPx STATUS REGISTER (I^2C™ MODE) (CONTINUED)

bit 0 **BF:** Buffer Full Status bit

In Transmit mode:
1 = Transmit is in progress, SSPxBUF is full
0 = Transmit is complete, SSPxBUF is empty

In Receive mode:
1 = SSPxBUF is full (does not include the $\overline{ACK}$ and Stop bits)
0 = SSPxBUF is empty (does not include the $\overline{ACK}$ and Stop bits)

Note 1: This bit is cleared on RESET and when SSPEN is cleared.

2: This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{ACK}$ bit.

3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Active mode.

REGISTER 17-3: SSPxCON1: MSSPx CONTROL REGISTER 1 (SPI MODE)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
WCOL	SSPOV(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
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 15-8 **Unimplemented:** Read as '0'

bit 7 **WCOL:** Write Collision Detect bit
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** MSSPx Receive Overflow Indicator bit(1)
SPI Slave mode:
1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. The user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
0 = No overflow

bit 5 **SSPEN:** MSSPx Enable bit(2)
1 = Enables serial port and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

bit 3-0 **SSPM<3:0>:** MSSPx Mode Select bits(3)
1010 = SPI Master mode, Clock = FOSC/(2 * ([SSPxADD] + 1))(4)
0101 = SPI Slave mode, Clock = SCKx pin; $\overline{SSx}$ pin control is disabled, $\overline{SSx}$ can be used as an I/O pin
0100 = SPI Slave mode, Clock = SCKx pin; $\overline{SSx}$ pin control is enabled
0011 = SPI Master mode, Clock = TMR2 output/2
0010 = SPI Master mode, Clock = FOSC/32
0001 = SPI Master mode, Clock = FOSC/8
0000 = SPI Master mode, Clock = FOSC/2

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
2: When enabled, these pins must be properly configured as input or output.
3: Bit combinations not specifically listed here are either reserved or implemented in I^2C mode only.
4: SSPxADD value of 0 is not supported when the Baud Rate Generator is used in SPI mode.

REGISTER 19-5: AD1CSSL: A/D INPUT SCAN SELECT 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
CSSL<15:8>(1)							
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL<7:6>		—	CSSL<4:0>(1)				
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 15-6 **CSSL<15:6>:** A/D Input Pin Scan Selection bits(1)
1 = Corresponding analog channel selected for input scan
0 = Analog channel omitted from input scan

bit 5 **Unimplemented:** Read as ‘0’

bit 4-0 **CSSL<4:0>:** A/D Input Pin Scan Selection bits(1)
1 = Corresponding analog channel selected for input scan
0 = Analog channel omitted from input scan

Note 1: CSSL<12:11,4:2> bits are unimplemented on 14-pin devices.

REGISTER 19-6: ANCFG: ANALOG INPUT CONFIGURATION 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	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	VBGEN
bit 7							bit 0

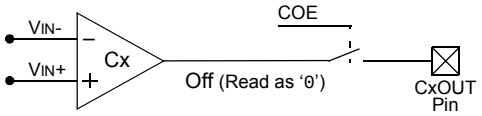
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-1 **Unimplemented:** Read as ‘0’

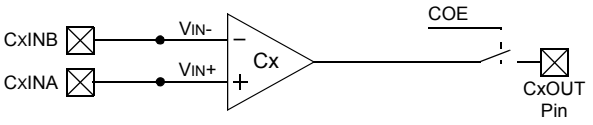
bit 0 **VBGEN:** Internal Band Gap Reference Enable bit
1 = Internal band gap voltage is available as a channel input to the A/D Converter
0 = Band gap is not available to the A/D Converter

FIGURE 20-2: INDIVIDUAL COMPARATOR CONFIGURATIONS

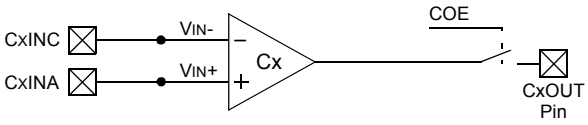
Comparator Off
CON = 0, CREF = x, CCH<1:0> = xx



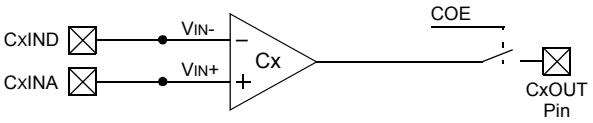
Comparator CxINB > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 00



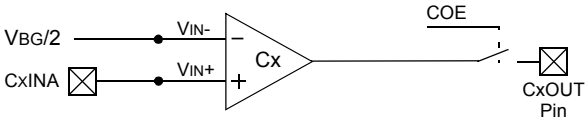
Comparator CxINC > CxINA Compare(1)
CON = 1, CREF = 0, CCH<1:0> = 01



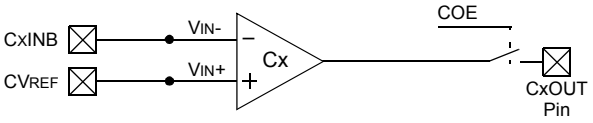
Comparator CxIND > CxINA Compare(1)
CON = 1, CREF = 0, CCH<1:0> = 10



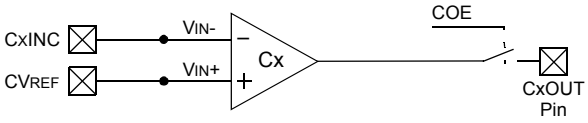
Comparator VBG > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 11



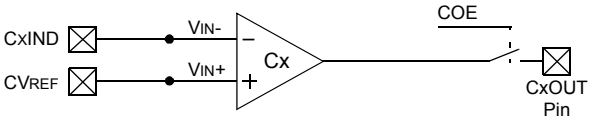
Comparator CxINB > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 00



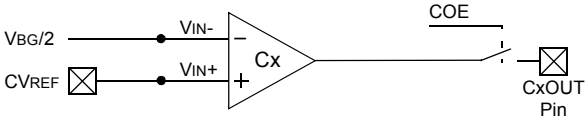
Comparator CxINC > CVREF Compare(1)
CON = 1, CREF = 1, CCH<1:0> = 01



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



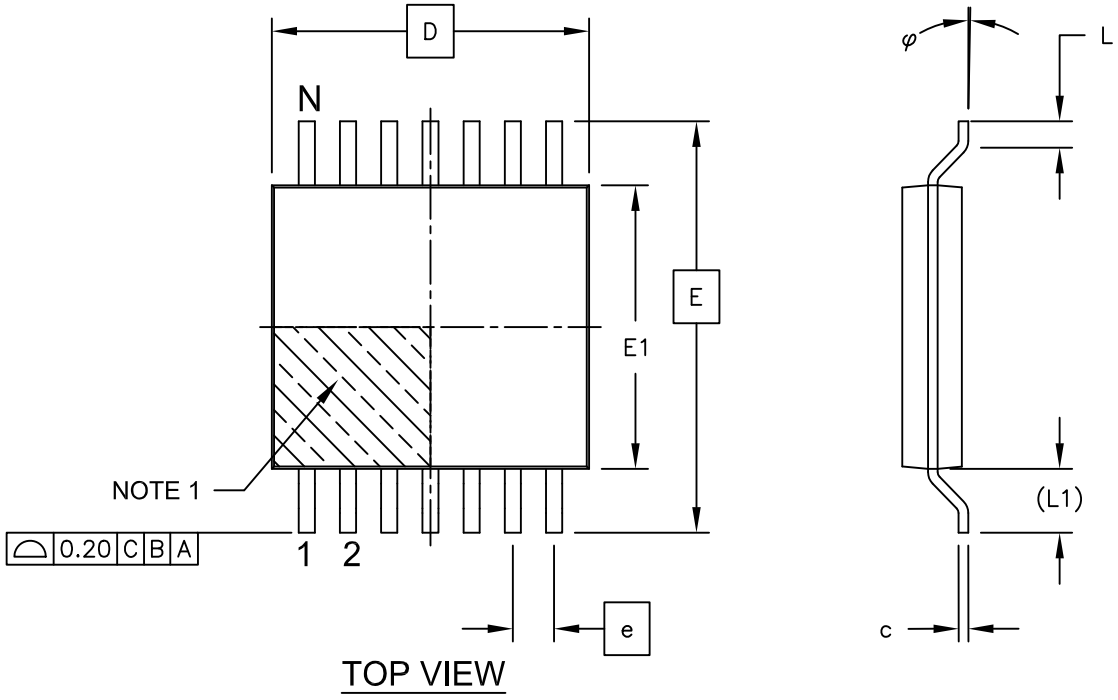
Comparator VBG > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 11

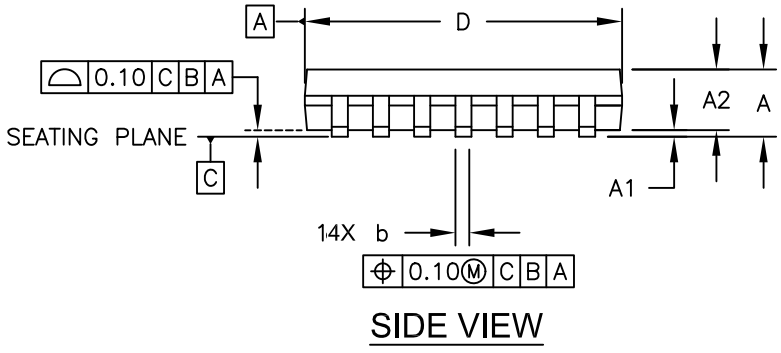


Note 1: This configuration is unavailable on 14-pin (PIC24FXXKL100/200) devices.

14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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

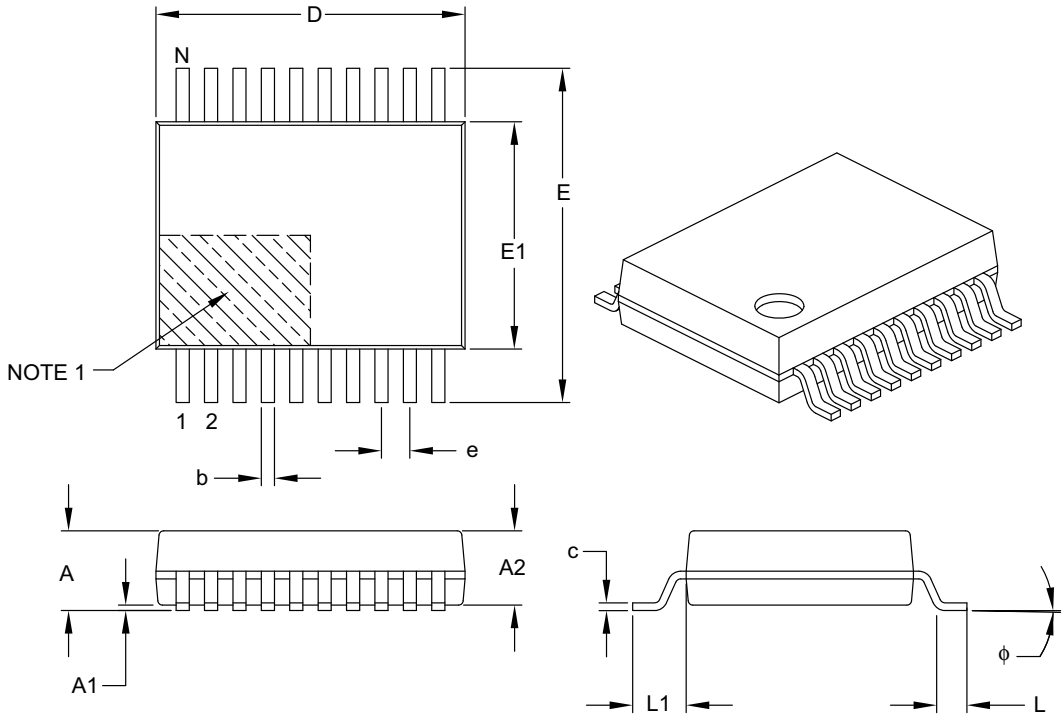




Microchip Technology Drawing C04-087C Sheet 1 of 2

20-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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

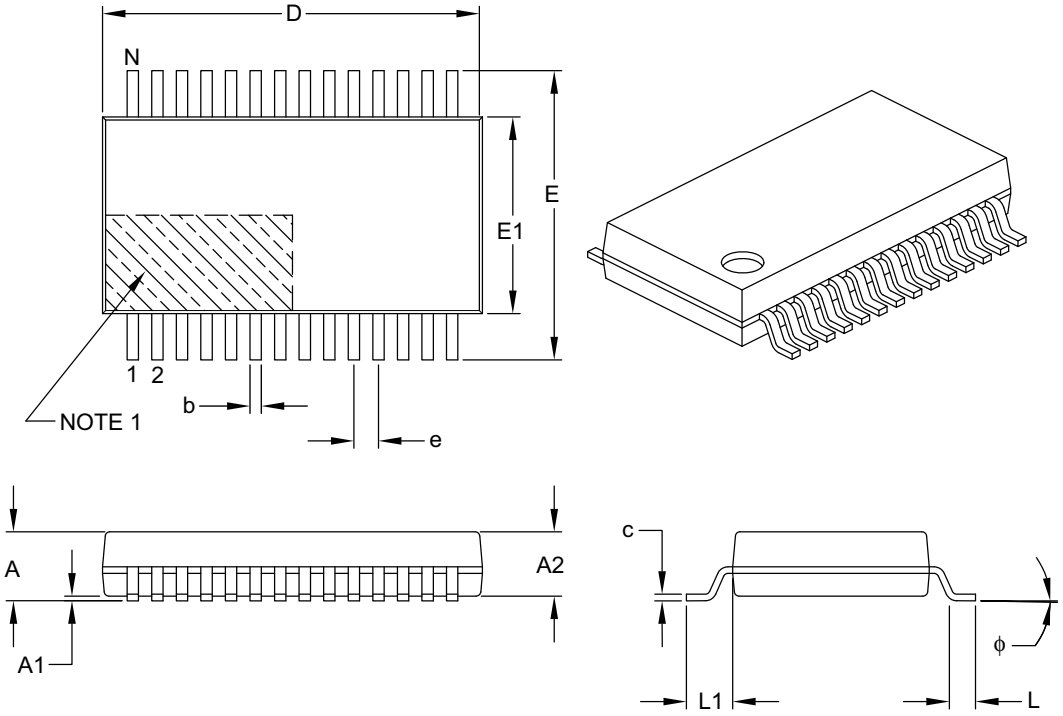
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-072B

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

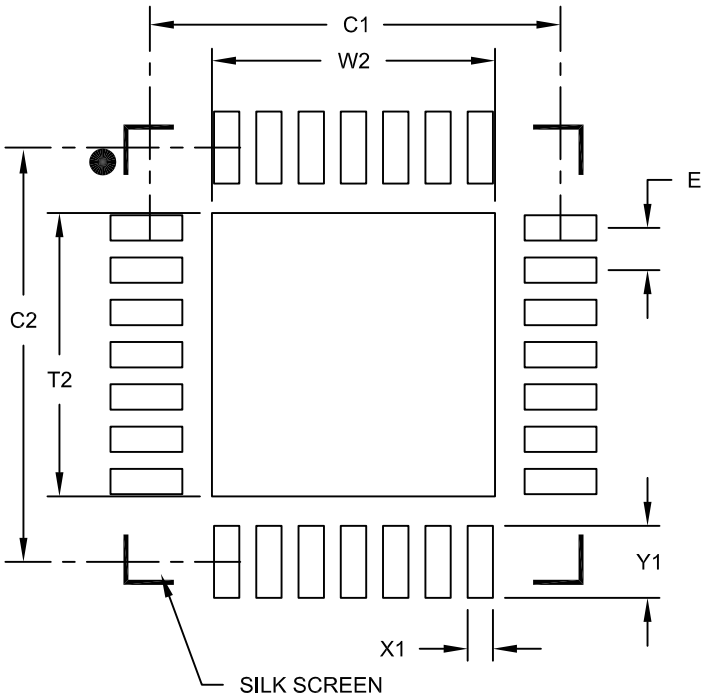
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5 mm Body [QFN] Land Pattern With 0.55 mm Contact Length

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		0.50 BSC	
Optional Center Pad Width	W2			3.35
Optional Center Pad Length	T2			3.35
Contact Pad Spacing	C1		4.90	
Contact Pad Spacing	C2		4.90	
Contact Pad Width (X28)	X1			0.30
Contact Pad Length (X28)	Y1			0.85

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2140A

APPENDIX A: REVISION HISTORY

Revision A (September 2011)

Original data sheet for the PIC24F16KL402 family of devices.

Revision B (November 2011)

Updates DC Specifications in Tables 26-6 through 26-9 (all Typical and Maximum values).

Updates AC Specifications in Tables 26-7 through 26-30 (SPI Timing Requirements) with the addition of the FSCK specification.

Other minor typographic corrections throughout.

Revision C (October 2013)

Adds +125°C Extended Temperature information.

Updates several packaging drawings in **Section 27.0 "Packaging Information"**. Other minor typographic corrections throughout.

APPENDIX B: MIGRATING FROM PIC18/PIC24 TO PIC24F16KL402

The PIC24F16KL402 family combines traditional PIC18 peripherals with a faster PIC24 core to provide a low-cost, high-performance microcontroller with low-power consumption.

Code written for PIC18 devices can be migrated to the PIC24F16KL402 by using a C compiler that generates PIC24 machine level instructions. Assembly language code will need to be rewritten using PIC24 instructions. The PIC24 instruction set shares similarities to the PIC18 instruction set, which should ease porting of assembly code. Application code will require changes to support certain PIC24 peripherals.

Code written for PIC24 devices can be migrated to the PIC24F16KL402 without many code changes. Certain peripherals, however, will require application changes to support modules that were traditionally available only on PIC18 devices.

Refer to Table B-1 for a list of peripheral modules on the PIC24F16KL402 and where they originated from.

TABLE B-1: TABLE B-1: PIC24F16KL402 PERIPHERAL MODULE ORIGINATING ARCHITECTURE

Peripheral Module	PIC18	PIC24
ECCP/CCP	X	—
MSSP (I^2C™/SPI)	X	—
Timer2/4 (8-bit)	X	—
Timer3 (16-bit)	X	—
Timer1 (16-bit)	—	X
10-Bit A/D Converter	—	X
Comparator	—	X
Comparator Voltage Reference	—	X
UART	—	X
HLVD	—	X