Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf27k42t-i-ss

TABLE 1: 28-PIN ALLOCATION TABLE (PIC18(L)F2XK42) (CONTINUED)

I/O	28-Pin SPDIP/SOIC/SSOP	28-Pin (U)QFN	ADC	Voltage Reference	DAC	Comparators	Zero Cross Detect	I^2C	SPI	UART	DSM	Timers/SMT	CCP and PWM	CWG	CLC	NCO	Clock Reference (CLKR)	Interrupt-on-Change	Basic
RC0	11	8	ANC0	—	—	—	—	—	—	—	—	T1CKI(1) T3CKI(1) T3G(1) SMTWIN1(1)	—	—	—	—	—	IOCC0	SOSCO
RC1	12	9	ANC1	—	—	—	—	—	—	—	—	SMTSIG1(1)	CCP2(1)	—	—	—	—	IOCC1	SOSCI
RC2	13	10	ANC2	—	—	—	—	—	—	—	—	T5CKI(1)	CCP1(1)	—	—	—	—	IOCC2	—
RC3	14	11	ANC3	—	—	—	—	SCL1(3,4)	SCK1(1)	—	—	T2IN(1)	—	—	—	—	—	IOCC3	—
RC4	15	12	ANC4	—	—	—	—	SDA1(3,4)	SDI1(1)	—	—	—	—	—	—	—	—	IOCC4	—
RC5	16	13	ANC5	—	—	—	—	—	—	—	—	T4IN(1)	—	—	—	—	—	IOCC5	—
RC6	17	14	ANC6	—	—	—	—	—	—	CTS1(1)	—	—	—	—	—	—	—	IOCC6	—
RC7	18	15	ANC7	—	—	—	—	—	—	RX1(1)	—	—	—	—	—	—	—	IOCC7	—
RE3	1	26	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCE3	$\overline{MCLR}$ V_{PP}
V_{DD}	20	17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
V_{SS}	8, 19	5, 16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
OUT(2)	—	—	ADGRDA ADGRDB	—	—	C1OUT C2OUT	—	SDA1 SCL1 SDA2 SCL2	SS1 SCK1 SDO1	DTR1 RTS1 TX1 DTR2 RTS2 TX2	DSM	TMR0	CCP1 CCP2 CCP3 CCP4 PWM5OUT PWM6OUT PWM7OUT PWM8OUT	CWG1A CWG1B CWG1C CWG1D CWG2A CWG2B CWG2C CWG2D CWG3A CWG3B CWG3C CWG3D	CLC1OUT CLC2OUT CLC3OUT CLC4OUT	NCO	CLKR	—	—

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.

2: All output signals shown in this row are PPS remappable.

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

4: These pins can be configured for I^2C and SMB™ 3.0/2.0 logic levels; The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

2.0 GUIDELINES FOR GETTING STARTED WITH PIC18(L)F26/27/45/46/47/55/56/57K42 MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC18(L)F26/27/45/46/47/55/56/57K42 family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins (see **Section 2.2 "Power Supply Pins"**)
- $\overline{\text{MCLR}}$ pin (see **Section 2.3 "Master Clear (MCLR) Pin"**)

These pins must also be connected if they are being used in the end application:

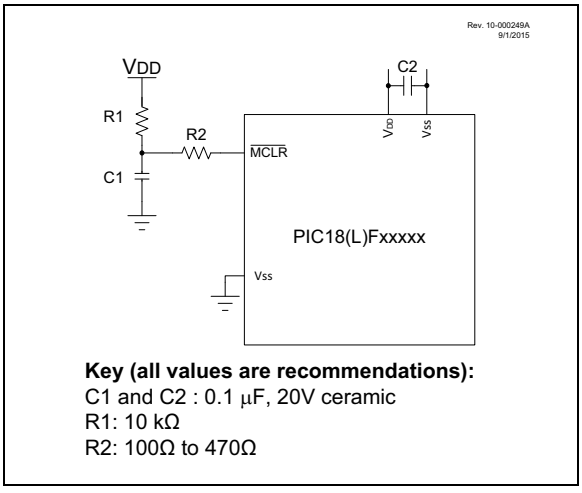
- ICSPCLK/ICSPDAT pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see **Section 2.4 "ICSP™ Pins"**)
- OSCI and OSCO pins when an external oscillator source is used (see **Section 2.5 "External Oscillator Pins"**)

Additionally, the following pins may be required:

- VREF+/VREF- pins are used when external voltage reference for analog modules is implemented

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):
C1 and C2 : 0.1 µF, 20V ceramic
R1: 10 kΩ
R2: 100Ω to 470Ω

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins (VDD and VSS) is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1 µF (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device, with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, make sure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 µF in parallel with 0.001 µF).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 µF to 47 µF.

2.5 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 7.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-3. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

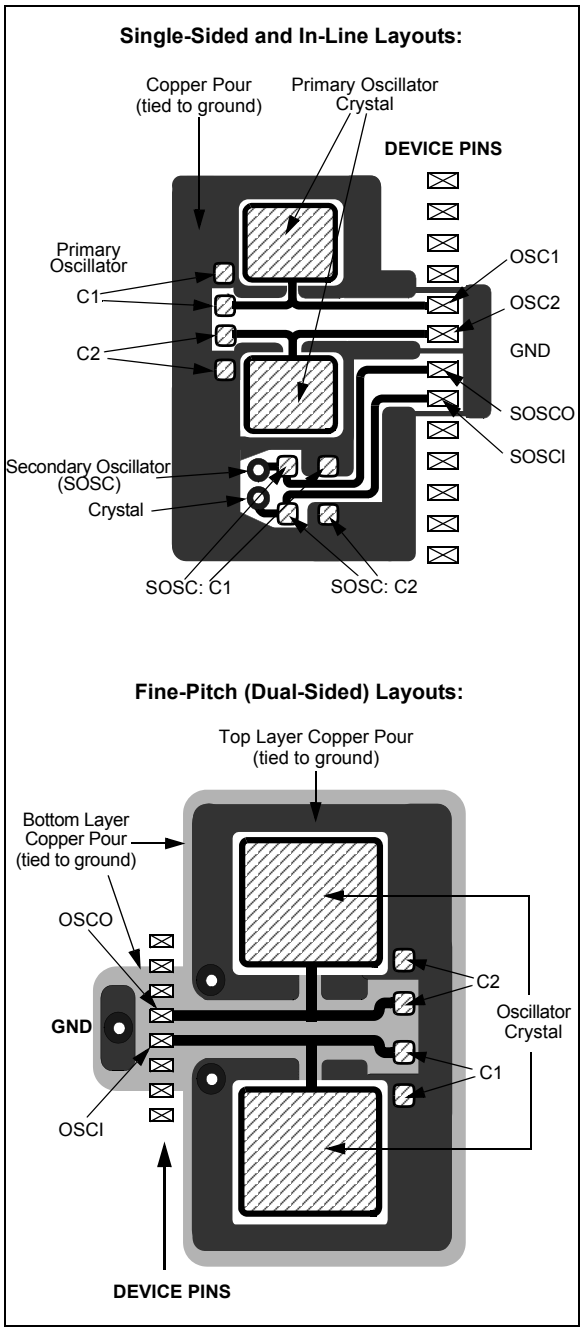
For additional information and design guidance on oscillator circuits, refer to these Microchip application notes, available at the corporate website (www.microchip.com):

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices*"
- AN849, *"Basic PICmicro® Oscillator Design"*
- AN943, *"Practical PICmicro® Oscillator Analysis and Design"*
- AN949, *"Making Your Oscillator Work"*

2.6 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to Vss on unused pins and drive the output to logic low.

FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



REGISTER 5-7: CONFIGURATION WORD 4L (30 0006h)

R/W-1	U-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{WRTAPP}}$ (1)	—	—	$\overline{\text{SAFEN}}$ (1)	$\overline{\text{BBEN}}$ (1)	BBSIZE<2:0> (2)		
bit 7							bit 0

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

bit 7 **$\overline{\text{WRTAPP}}$:** Application Block Write Protection bit(1)
1 = Application Block is NOT write-protected
0 = Application Block is write-protected

bit 6-5 **Unimplemented:** Read as '1'

bit 4 **$\overline{\text{SAFEN}}$:** Storage Area Flash Enable bit(1)
1 = SAF is disabled
0 = SAF is enabled

bit 3 **$\overline{\text{BBEN}}$:** Boot Block Enable bit(1)
1 = Boot Block disabled
0 = Boot Block enabled

bit 2-0 **BBSIZE<2:0>:** Boot Block Size Selection bits(2)
Refer to Table 5-1.

Note 1: Bits are implemented as sticky bits. Once protection is enabled through ICSP™ or a self-write, it can only be reset through a Bulk Erase.
2: BBSIZE<2:0> bits can only be changed when $\overline{\text{BBEN}}$ = 1. Once $\overline{\text{BBEN}}$ = 0, BBSIZE<2:0> can only be changed through a Bulk Erase.

TABLE 5-1: BOOT BLOCK SIZE BITS

$\overline{\text{BBEN}}$	BBSIZE<2:0>	Boot Block Size (words)	END_ADDRESS_BOOT	Device Size(1)		
				16k	32k	64k
1	xxx	0	—	X	X	X
0	111	512	00 03FFh	X	X	X
0	110	1024	00 07FFh	X	X	X
0	101	2048	00 0FFFh	X	X	X
0	100	4096	00 1FFFh	X	X	X
0	011	8192	00 3FFFh	X	X	X
0	010	16384	00 7FFFh	—	X	X
0	001	32768	00 FFFFh		Note 2	X
0	000	32768	00 FFFFh	—	—	—

Note 1: For each device, the quoted device size specification is listed in Table 4-1.
2: The maximum boot block size is half the user program memory size. All selections higher than the maximum size default to maximum boot block size of half PFM. For example, all settings of BBSIZE = 000 through BBSIZE = 010, default to a boot block size of 16 kW on a 32 kW device.

5.7.1 MICROCHIP UNIQUE IDENTIFIER (MUI)

The PIC18(L)F26/27/45/46/47/55/56/57K42 devices are individually encoded during final manufacturing with a Microchip Unique Identifier, or MUI. The MUI cannot be user-erased. This feature allows for manufacturing traceability of Microchip Technology devices in applications where this is a required. It may also be used by the application manufacturer for a number of functions that require unverified unique identification, such as:

- Tracking the device
- Unique serial number

The MUI consists of six program words. When read together, these fields form a unique identifier. The MUI is stored in nine read-only locations, located between 3F0000h to 3F000Fh in the DIA space. Table 5-3 lists the addresses of the identifier words.

Note:	For applications that require verified unique identification, contact your Microchip Technology sales office to create a Serialized Quick Turn Programming^{SM} option.

5.7.2 EXTERNAL UNIQUE IDENTIFIER (EUI)

The EUI data is stored at locations 3F0010h to 3F0023h in the Program Memory region. This region is an optional space for placing application specific information. The data is coded per customer requirements during manufacturing.

Note:	Data is stored in this address range on receiving a request from the customer. The customer may contact the local sales representative, or Field Applications Engineer, and provide them the unique identifier information that is supposed to be stored in this region.

5.7.3 ANALOG-TO-DIGITAL CONVERSION DATA OF THE TEMPERATURE SENSOR

The purpose of the Temperature Sensor module is to provide a temperature-dependent voltage that can be measured by an analog module, see **Section 35.0 “Temperature Indicator Module”**.

The DIA table contains the internal ADC measurement values of the Temperature sensor for Low and High range at fixed points of reference. The values are measured during test and are unique to each device. The measurement data is stored in the DIA memory region as hexadecimal numbers corresponding to the ADC conversion result. The calibration data can be used to plot the approximate sensor output voltage, V_{TSENSE} vs. Temperature curve without having to make calibration measurements in the application. For more information on the operation of the Temperature Sensor, refer to **Section 35.0 “Temperature Indicator Module”**.

- **TSLR2**: Address 3F0026h to 3F0027h store the measurements for the low-range setting of the Temperature Sensor at V_{DD} = 3V.
- **TSHR2**: Address 3F002Ch to 3F002Dh store the measurements for the High Range setting of the Temperature Sensor at V_{DD} = 3V.
- The stored measurements are made by the device ADC using the internal V_{REF} = 2.048V.

5.7.4 FIXED VOLTAGE REFERENCE DATA

The DIA stores measured FVR voltages for this device in mV for the different buffer settings of 1x, 2x or 4x at Program Memory locations 3F0030h to 3F003Bh. For more information on the FVR, refer to **Section 34.0 "Fixed Voltage Reference (FVR)"**.

- FVRA1X stores the value of ADC FVR1 Output voltage for 1x setting (in mV)
- FVRA2X stores the value of ADC FVR1 Output Voltage for 2x setting (in mV)
- FVRA4X stores the value of ADC FVR1 Output Voltage for 4x setting (in mV)
- FVRC1X stores the value of Comparator FVR2 output voltage for 2x setting (in mV)
- FVRC2X stores the value of Comparator FVR2 output voltage for 2x setting (in mV)
- FVRC4X stores the value of Comparator FVR2 output voltage for 4x setting (in mV)

5.8 Device Configuration Information

The Device Configuration Information (DCI) is a dedicated region in the Program memory space mapped from 3FFF00h to 3FFF09h. The data stored in these locations is read-only and cannot be erased.

Refer to Table 5-4: Device Configuration Information for PIC18(L)F26/27/45/55/46/47/56/57K42 for the complete DCI table address and description. The DCI holds information about the device which is useful for programming and Bootloader applications.

The erase size is the minimum erasable unit in the PFM, expressed as rows. The total device Flash memory capacity is (Row Size * Number of rows)

TABLE 5-4: DEVICE CONFIGURATION INFORMATION FOR PIC18(L)F26/27/45/55/46/47/56/57K42

ADDRESS	Name	DESCRIPTION	VALUE			UNITS
			PIC18(L)F45/55K42	PIC18(L)F26/46/56K42	PIC18(L)F27/47/57K42	
3F FF00h-3F FF01h	ERSIZ	Erase Row Size	64	64	64	Words
3F FF02h-3F FF03h	WLSIZ	Number of write latches per row	128	128	128	Bytes
3F FF04h-3F FF05h	URSIZ	Number of User Rows	256	512	1024	Rows
3F FF06h-3F FF07h	EESIZ	Data EEPROM memory size	256	1024	1024	Bytes
3F FF08h-3F FF09h	PCNT	Pin Count	40(1)/48	28/40(1)/48	28/40(1)/48	Pins

Note 1: Pin count of 40 is also used for 44-pin part.

8.5 Register Definitions: Reference Clock

Long bit name prefixes for the Reference Clock peripherals are shown below. Refer to **Section 1.3.2.2 "Long Bit Names"** for more information.

Peripheral	Bit Name Prefix
CLKR	CLKR

REGISTER 8-1: CLKRCON: REFERENCE CLOCK CONTROL REGISTER

R/W-0/0	U-0	U-0	R/W-1/1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	—	DC<1:0>		DIV<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **EN:** Reference Clock Module Enable bit
- 1 = Reference clock module enabled
- 0 = Reference clock module is disabled

bit 6-5 **Unimplemented:** Read as '0'

bit 4-3 **DC<1:0>:** Reference Clock Duty Cycle bits(1)
- 11 = Clock outputs duty cycle of 75%
- 10 = Clock outputs duty cycle of 50%
- 01 = Clock outputs duty cycle of 25%
- 00 = Clock outputs duty cycle of 0%

bit 2-0 **DIV<2:0>:** Reference Clock Divider bits
- 111 = Base clock value divided by 128
- 110 = Base clock value divided by 64
- 101 = Base clock value divided by 32
- 100 = Base clock value divided by 16
- 011 = Base clock value divided by 8
- 010 = Base clock value divided by 4
- 001 = Base clock value divided by 2
- 000 = Base clock value

Note 1: Bits are valid for reference clock divider values of two or larger, the base clock cannot be further divided.

EXAMPLE 9-4: SETTING UP VECTORED INTERRUPTS USING XC8

```
// NOTE 1: If IVTBASE is changed from its default value of 0x000008, then the
// "base(...)" argument must be provided in the ISR. Otherwise the vector
// table will be placed at 0x0008 by default regardless of the IVTBASE value.

// NOTE 2: When MVECEN=0 and IPEN=1, a separate argument as "high_priority"
// or "low_priority" can be used to distinguish between the two ISRs.
// If the argument is not provided, the ISR is considered high priority
// by default.

// NOTE 3: Multiple interrupts can be handled by the same ISR if they are
// specified in the "irq(...)" argument. Ex: irq(IRQ_TMR0, IRQ_CCP1)

void __interrupt(irq(IRQ_TMR0), base(0x4008)) TMR0_ISR(void)
{
        PIR3bits.TMR0IF = 0;                 // Clear the interrupt flag
        LATCbits.LC0 ^= 1;                   // ISR code goes here
}

void __interrupt(irq(default), base(0x4008)) DEFAULT_ISR(void)
{
        // Unhandled interrupts go here
}

void INTERRUPT_Initialize (void)
{
        INTCON0bits.GIEH = 1;                // Enable high priority interrupts
        INTCON0bits.GIEL = 1;                // Enable low priority interrupts
        INTCON0bits.IPEN = 1;                // Enable interrupt priority

        PIE3bits.TMR0IE = 1;                 // Enable TMR0 interrupt
        PIE4bits.TMR1IE = 1;                 // Enable TMR1 interrupt

        IPR3bits.TMR0IP = 0;                 // Make TMR0 interrupt low priority

        // Change IVTBASE if required
        IVTBASEU = 0x00;                     // Optional
        IVTBASEH = 0x40;                     // Default is 0x0008
        IVTBASEL = 0x08;
}
```

REGISTER 9-5: PIR2: PERIPHERAL INTERRUPT REGISTER 2(1)

R-0/0	R-0/0	R-0/0	R-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
I2C1RXIF(2)	SPI1IF(3)	SPI1TXIF(4)	SPI1RXIF(4)	DMA1AIF	DMA1ORIF	DMA1DCNTIF	DMA1SCNTIF
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HS = Hardware set

bit 7 **I2C1RXIF:** I^2C1 Receive Interrupt Flag bit(2)
- 1 = Interrupt has occurred
- 0 = Interrupt event has not occurred

bit 6 **SPI1IF:** SPI1 Interrupt Flag bit(3)
- 1 = Interrupt has occurred
- 0 = Interrupt event has not occurred

bit 5 **SPI1TXIF:** SPI1 Transmit Interrupt Flag bit(4)
- 1 = Interrupt has occurred
- 0 = Interrupt event has not occurred

bit 4 **SPI1RXIF:** SPI1 Receive Interrupt Flag bit(4)
- 1 = Interrupt has occurred
- 0 = Interrupt event has not occurred

bit 3 **DMA1AIF:** DMA1 Abort Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 2 **DMA1ORIF:** DMA1 Overrun Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 1 **DMA1DCNTIF:** DMA1 Destination Count Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

bit 0 **DMA1SCNTIF:** DMA1 Source Count Interrupt Flag bit
- 1 = Interrupt has occurred (must be cleared by software)
- 0 = Interrupt event has not occurred

Note 1: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

2: I2CxTXIF and I2CxRXIF are read-only bits. To clear the interrupt condition, the CLRBF bit in I2CxSTAT1 register must be set.

3: SPIxIF is a read-only bit. To clear the interrupt condition, all bits in the SPIxINTF register must be cleared.

4: SPIxTXIF and SPIxRXIF are read-only bits and cannot be set/cleared by the software.

11.6 Operation During Sleep

When the device enters Sleep, the WWDT is cleared. If the WWDT is enabled during Sleep, the WWDT resumes counting. When the device exits Sleep, the WWDT is cleared again.

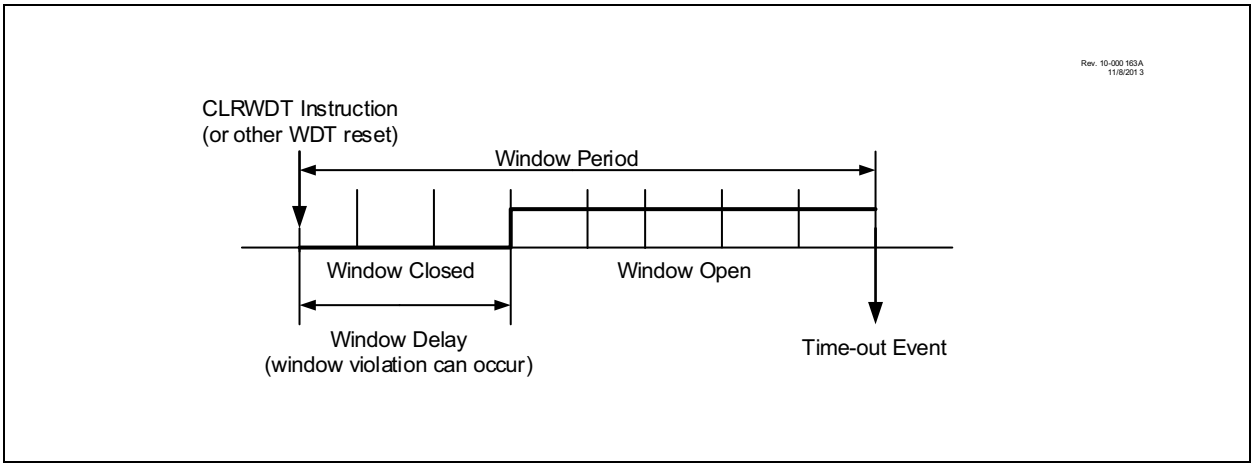
The WWDT remains clear until the Oscillator Start-up Timer (OST) completes, if enabled. See **Section 7.2.1.3 "Oscillator Start-up Timer (OST)"** for more information on the OST.

When a WWDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the event. The $\overline{RWDT}$ bit in the PCON0 register can also be used. See **Section 4.0 "Memory Organization"** for more information.

TABLE 11-2: WWDT CLEARING CONDITIONS

Conditions	WWDT
WDTE<1:0> = 00	Cleared
WDTE<1:0> = 01 and SEN = 0	
WDTE<1:0> = 10 and enter Sleep	
CLRWDT Command	
Oscillator Fail Detected	
Exit Sleep + System Clock = SOSC, EXTRC, INTOSC, EXTCLK	
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST
Change INTOSC divider (IRCF bits)	Unaffected

FIGURE 11-2: WINDOW PERIOD AND DELAY



14.8 Scanner Module Overview

The Scanner allows segments of the Program Flash Memory or Data EEPROM, to be read out (scanned) to the CRC Peripheral. The Scanner module interacts with the CRC module and supplies it data one word at a time. Data is fetched from the address range defined by SCANLADR registers up to the SCANHADR registers.

The Scanner begins operation when the SGO bit is set (SCANCON0 Register) and ends when either SGO is cleared by the user or when SCANLADR increments past SCANHADR. The SGO bit is also cleared by clearing the EN bit (CRCCON0 register).

14.9 Configuring the Scanner

The scanner module may be used in conjunction with the CRC module to perform a CRC calculation over a range of program memory or Data EEPROM addresses. In order to set up the scanner to work with the CRC, perform the following steps:

1. Set up the CRC module (See **Section 14.7 "Configuring the CRC"**) and enable the Scanner module by setting the EN bit in the SCANCON0 register.
2. Choose which memory region the Scanner module should operate on and set the MREG bit of the SCANCON0 register appropriately.
3. If trigger is used for scanner operation, set the TRIGEN bit of the SCANCON0 register and select the trigger source using SCANTRIG register. Select the trigger source using SCANTRIG register and then set the TRIGEN bit of the SCANCON0 register. See Table 14-1 for Scanner Operation.
4. If Burst mode of operation is desired, set the BURSTMD bit (SCANCON0 register). See Table 14-1 for Scanner Operation.
5. Set the SCANLADRL/H/U and SCANHADRL/H/U registers with the beginning and ending locations in memory that are to be scanned.
6. Select the priority level for the Scanner module (See **Section 3.1 "System Arbitration"**) and lock the priorities (See **Section 3.1.1 "Priority Lock"**).
7. Both CRCEN and CRCGO bits must be enabled to use the scanner. Setting the SGO bit will start the scanner operation.

14.10 Scanner Interrupt

The scanner will trigger an interrupt when the SCANLADR increments past SCANHADR. The SCANIF bit can only be cleared in software.

14.11 Scanning Modes

The interaction of the scanner with the system operation is controlled by the priority selection in the System Arbiter (see **Section 3.2 "Memory Access Scheme"**). Additionally, BURSTMD and TRIGEN also determine the operation of the Scanner.

14.11.1 TRIGEN = 0, BURSTMD = 0

In this case, the memory access request is granted to the scanner if no other higher priority source is requesting access.

All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.

14.11.2 TRIGEN = 1, BURSTMD = 0

In this case, the memory access request is generated when the CRC module is ready to accept.

The memory access request is granted to the scanner if no other higher priority source is requesting access. All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.

The memory access request is granted to the scanner if no other higher priority source is requesting access. All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.

14.11.3 TRIGEN = x, BURSTMD = 1

In this case, the memory access is always requested by the scanner.

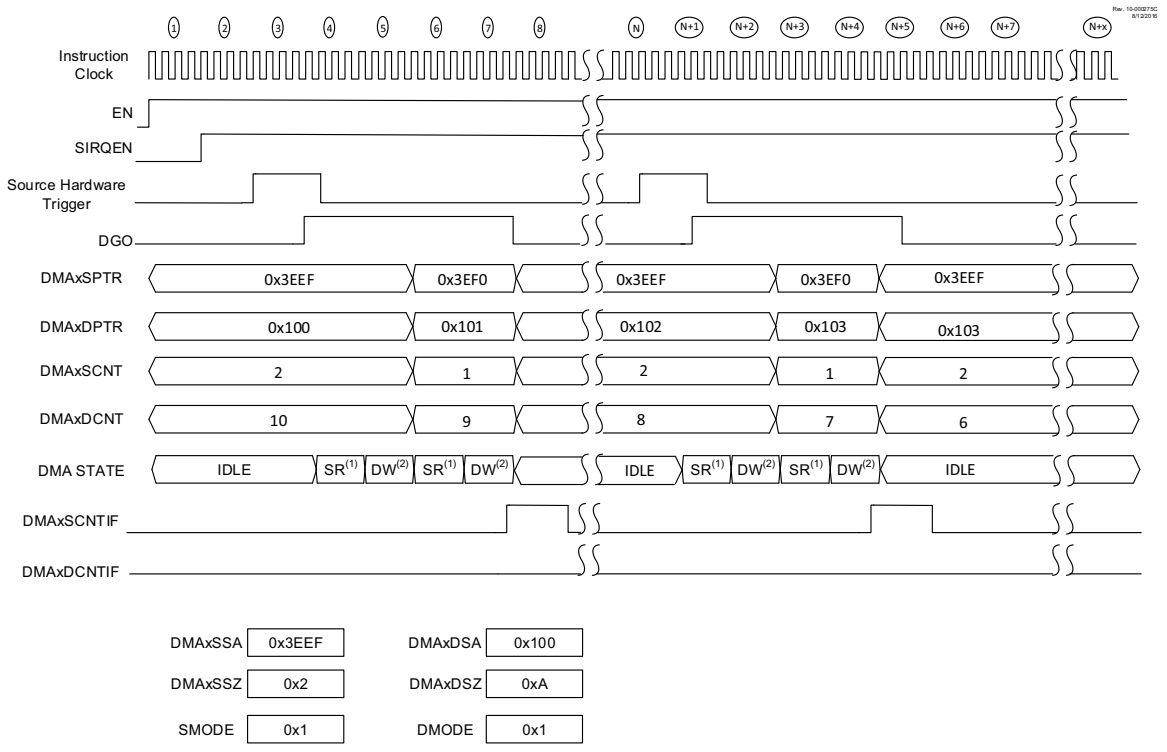
The memory access request is granted to the scanner if no other higher priority source is requesting access. The memory access cycles will not be granted to lower priority sources than the scanner until it completes operation i.e. SGO = 0 (SCANCON0 register)

Note:	If TRIGEN = 1 and BURSTMD = 1, the user should ensure that the trigger source is active for the Scanner operation to complete.

15.9.4 TRANSFER FROM SFR TO GPR

The following visual reference describes the sequence of events when copying ADC results to a GPR location. The ADC Interrupt Flag can be chosen as the Source Hardware trigger, the Source address can be set to point to the ADC Result registers at 3EEF, the Destination address can be set to point to any GPR location of our choice (Example 0x100).

FIGURE 15-8: SFR SPACE TO GPR SPACE TRANSFER



Note 1: SR - Source Read

2: DW - Destination Write

REGISTER 16-8: INLVLx: INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLx7	INLVLx6	INLVLx5	INLVLx4	INLVLx3	INLVLx2	INLVLx1	INLVLx0
bit 7							bit 0

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

bit 7-0 **INLVLx<7:0>**: Input Level Select on Pins Rx<7:0>, respectively
1 = ST input used for port reads and interrupt-on-change
0 = TTL input used for port reads and interrupt-on-change

TABLE 16-9: INPUT LEVEL PORT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INLVLA	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2(1)	INLVLB1(1)	INLVLB0
INLVLC	INLVLC7	INLVLC6	INLVLC5	INLVLC4(1)	INLVLC3(1)	INLVLC2	INLVLC1	INLVLC0
INLVLD(2)	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1(1)	INLVLD0(1)
INLVLE	—	—	—	—	INLVLE3	INLVLE2(2)	INLVLE1(2)	INLVLE0(2)
INLVLF(3)	INLVLF7	INLVLF6	INLVLF5	INLVLF4	INLVLF3	INLVLF2	INLVLF1	INLVLF0

Note 1: Any peripheral using the I^2C pins read the I^2C ST inputs when enabled via RxyI2C.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented in PIC18(L)F26/27/45/46/47K42.

20.1 Timer0 Operation

Timer0 can operate as either an 8-bit timer/counter or a 16-bit timer/counter. The mode is selected with the MD16 bit of the T0CON register.

20.1.1 16-BIT MODE

The register pair TMR0H:TMR0L increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, CKPS<3:0> in the T0CON1 register).

20.1.1.1 Timer0 Reads and Writes in 16-Bit Mode

In 16-bit mode, in order to avoid rollover between reading high and low registers, the TMR0H register is a buffered copy of the actual high byte of Timer0, which is neither directly readable, nor writable (see Figure 20-1). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte was valid, due to a rollover between successive reads of the high and low byte.

Similarly, a write to the high byte of Timer0 must also take place through the TMR0H Buffer register. The high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

20.1.2 8-BIT MODE

In 8-bit mode, the value of TMR0L is compared to that of the Period buffer, a copy of TMR0H, on each clock cycle. When the two values match, the following events happen:

- TMR0_out goes high for one prescaled clock period
- TMR0L is reset
- The contents of TMR0H are copied to the period buffer

In 8-bit mode, the TMR0L and TMR0H registers are both directly readable and writable. The TMR0L register is cleared on any device Reset, while the TMR0H register initializes at FFh.

Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR0L register
- A write to either the T0CON0 or T0CON1 registers
- Any device Reset – Power-on Reset (POR), MCLR Reset, Watchdog Timer Reset (WDTR) or
- Brown-out Reset (BOR)

20.1.3 COUNTER MODE

In Counter mode, the prescaler is normally disabled by setting the CKPS bits of the T0CON1 register to '0000'. Each rising edge of the clock input (or the output of the prescaler if the prescaler is used) increments the counter by '1'.

20.1.4 TIMER MODE

In Timer mode, the Timer0 module will increment every instruction cycle as long as there is a valid clock signal and the CKPS bits of the T0CON1 register (Register 20-2) are set to '0000'. When a prescaler is added, the timer will increment at the rate based on the prescaler value.

20.1.5 ASYNCHRONOUS MODE

When the ASYNC bit of the T0CON1 register is set (ASYNC = '1'), the counter increments with each rising edge of the input source (or output of the prescaler, if used). Asynchronous mode allows the counter to continue operation during Sleep mode provided that the clock also continues to operate during Sleep.

20.1.6 SYNCHRONOUS MODE

When the ASYNC bit of the T0CON1 register is clear (ASYNC = '0'), the counter clock is synchronized to the system clock (FOSC/4). When operating in Synchronous mode, the counter clock frequency cannot exceed FOSC/4.

20.2 Clock Source Selection

The CS<2:0> bits of the T0CON1 register are used to select the clock source for Timer0. Register 20-2 displays the clock source selections.

20.2.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, Timer0 operates as a timer and will increment on multiples of the clock source, as determined by the Timer0 prescaler.

20.2.2 EXTERNAL CLOCK SOURCE

When an external clock source is selected, Timer0 can operate as either a timer or a counter. Timer0 will increment on multiples of the rising edge of the external clock source, as determined by the Timer0 prescaler.

31.13 Checksum (UART1 only)

This section does not apply to the LIN mode, which handles checksums automatically.

The transmit and receive checksum adders are enabled when the C0EN bit in the UxCON2 register is set. When enabled, the adders accumulate every byte that is transmitted or received. The accumulated sum includes the carry of the addition. Software is responsible for clearing the checksum registers before a transaction and performing the check at the end of the transaction.

The following is an example of how the checksum registers could be used in the asynchronous modes.

31.13.1 TRANSMIT CHECKSUM METHOD

1. Clear the UxTXCHK register.
2. Set the C0EN bit.
3. Send all bytes of the transaction output.
4. Invert UxTXCHK and send the result as the last byte of the transaction.

31.13.2 RECEIVE CHECKSUM METHOD

1. Clear the UxRXCHK register.
2. Set the C0EN bit.
3. Receive all bytes in the transaction including the checksum byte.
4. Set MSb of UxRXCHK if 7-bit mode is selected.
5. Add 1 to UxRXCHK.
6. If the result is '0', the checksum passes, otherwise it fails.

The CERIF checksum interrupt flag is not active in any mode other than LIN.

31.14 Collision Detection

External forces that interfere with the transmit line are detected in all modes of operation with collision detection. Collision detection is always active when RXEN and TXEN are both set.

When the receive input is connected to the transmit output through either the same I/O pin or external circuitry, a character will be received for every character transmitted. The collision detection circuit provides a warning when the word received does not match the word transmitted.

The TXCIF flag in the UxERRIR register is used to signal collisions. This signal is only useful when the TX output is looped back to the RX input and everything that is transmitted is expected to be received. If more than one transmitter is active at the same time, it can be assumed that the TX word will not match the RX word. The TXCIF detects this mismatch and flags an interrupt. The TXCIF bit will also be set in DALI mode transmissions when the received bit is missing the expected mid-bit transition.

Collision detection is always active, regardless of whether or not the RX input is connected to the TX output. It is up to the user to disable the TXCIE bit when collision interrupts are not required.

The software overhead of unloading the receive buffer of transmitted data is avoided by setting the RUNOVF bit in UxCON2 and ignoring the receive interrupt and letting the receive buffer overflow. When the transmission is complete, prepare for receiving data by flushing the receive buffer (see Section 31.11.2, FIFO Reset) and clearing the RXFOIF overflow flag in the UxERRIR register.

31.15 RX/TX Activity Timeout

The UART works in conjunction with the HLT timers to monitor activity on the RX and TX lines. Use this feature to determine when there has been no activity on the receive or transmit lines for a user specified period of time.

To use this feature, set the HLT to the desired timeout period by a combination of the HLT clock source, timer prescale value, and timer period registers. Configure the HLT to reset on the UART TX or RX line and start the HLT at the same time the UART is started. UART activity will keep resetting the HLT to prevent a full HLT period from elapsing. When there has been no activity on the selected TX or RX line for longer than the HLT period then an HLT interrupt will occur signaling the timeout event.

For example, the following register settings will configure HLT2 for a 5 ms timeout of no activity on U1RX:

- T2PR = 0x9C (156 prescale periods)
- T2CLKCON = 0x05 (500 kHz internal oscillator)
- T2HLT = 0x04 (free running, reset on rising edge)
- T2RST = 0x15 (reset on U1RX)
- T2CON = 0xC0 (Timer2 on with 1:16 prescale)

REGISTER 31-5: UxERRIE: UART ERROR INTERRUPT ENABLE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TXMTIE	PERIE	ABDOVE	CERIE	FERIE	RXBKIE	RXFOIE	TXCIE
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **TXMTIE:** Transmit Shift Register Empty Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

bit 6 **PERIE**: Parity Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

bit 5 **ABDOVE:** Auto-baud Detect Overflow Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

bit 4 **CERIE**: Checksum Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

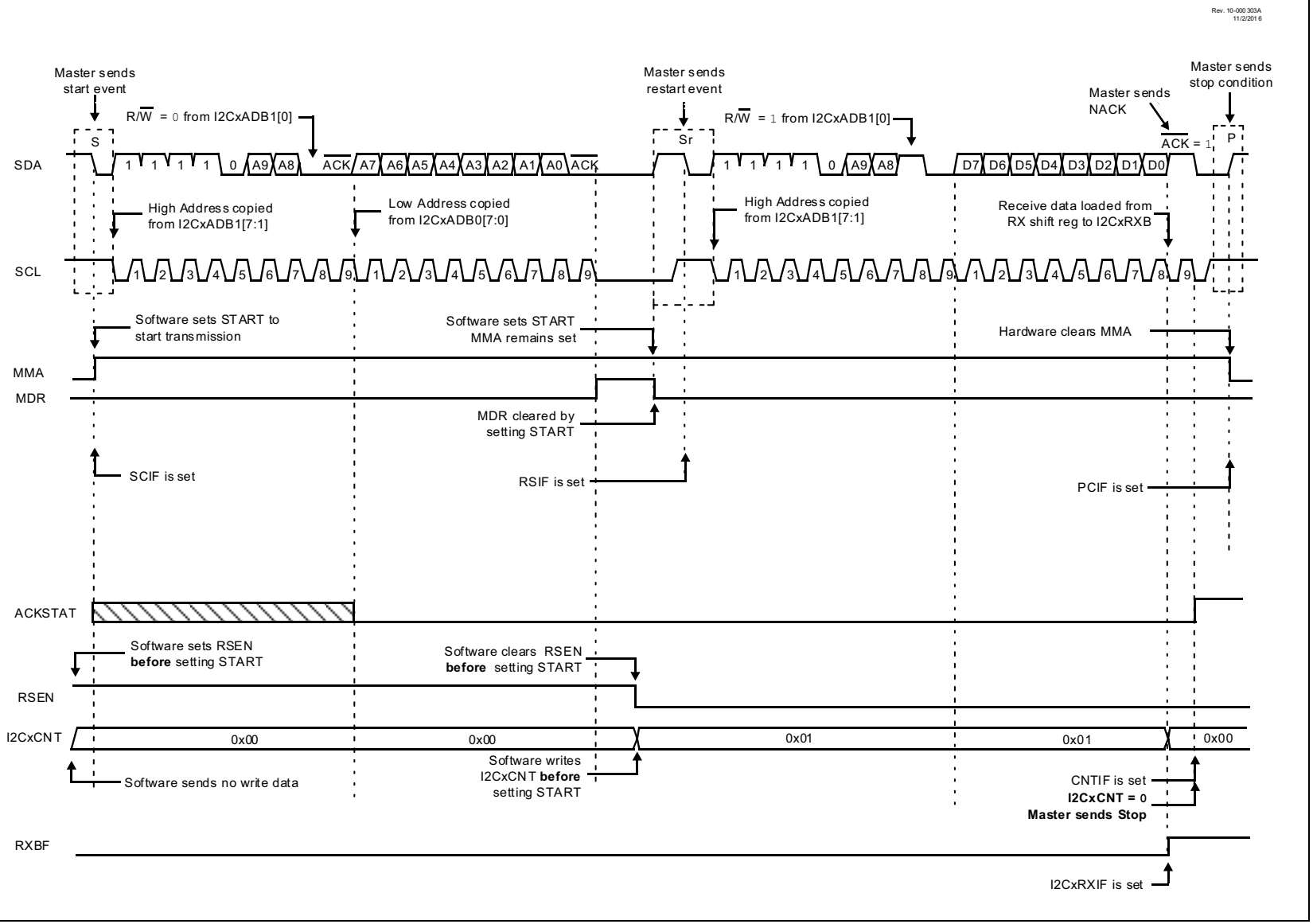
bit 3 **FERIE:** Framing Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

bit 2 **RXBKIE:** Break Reception Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

bit 1 **RXFOIE:** Receive FIFO Overflow Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

bit 0 **TXCIE:** Transmit Collision Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt not enabled

FIGURE 33-22: I²C MASTER, 10-BIT ADDRESS, RECEPTION (USING RSTEN BIT)



36.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting
- Conversion Trigger Selection
- ADC Acquisition Time
- ADC Precharge Time
- Additional Sample and Hold Capacitor
- Single/Double Sample Conversion
- Guard Ring Outputs

36.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 16.0 "I/O Ports"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

36.1.2 CHANNEL SELECTION

There are several channel selections available:

- Eight PORTA pins (RA<7:0>)
- Eight PORTB pins (RB<7:0>)
- Eight PORTC pins (RC<7:0>)
- Eight PORTD pins (RD<7:0>, PIC18(L)F45/46/47/55/56/57K42 only)
- Three PORTE pins (RE<2:0>, PIC18(L)F45/46/47/55/56/57K42 only)
- Eight PORTF pins (RD<7:0>, PIC18(L)F55/56/57K42 only)
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- VSS (ground)

The ADPCH register determines which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion.

Refer to **Section 36.2 "ADC Operation"** for more information.

36.1.3 ADC VOLTAGE REFERENCE

The PREF<1:0> bits of the ADREF register provide control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR outputs

The NREF bit of the ADREF register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 34.0 "Fixed Voltage Reference (FVR)"** for more details on the Fixed Voltage Reference.

36.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCLK register and the CS bits of the ADCON0 register. If FOSC is selected as the ADC clock, there is a prescaler available to divide the clock so that it meets the ADC clock period specification. The ADC clock source options are the following:

- FOSC/(2*n)(where n is from 1 to 128)
- FRC (dedicated RC oscillator)

The time to complete one bit conversion is defined as TAD. Refer Figure 36-2 for the complete timing details of the ADC conversion.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 44-16 for more information. Table 36-1 gives examples of appropriate ADC clock selections.

Note 1:	Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.
2:	The internal control logic of the ADC runs off of the clock selected by the CS bit of ADCON0. What this can mean is when the CS bit of ADCON0 is set to '1' (ADC runs on FRC), there may be unexpected delays in operation when setting ADC control bits.

REGISTER 36-4: ADCON3: ADC CONTROL REGISTER 3

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W/HC-0	R/W-0/0	R/W-0/0	R/W-0/0
—	CALC<2:0>			SOI	TMD<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

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

bit 6-4 **CALC<2:0>:** ADC Error Calculation Mode Select bits

CALC	DSEN = 0 Single-Sample Mode	DSEN = 1 CVD Double-Sample Mode(1)	Application
111	Reserved	Reserved	Reserved
110	Reserved	Reserved	Reserved
101	FLTR-STPT	FLTR-STPT	Average/filtered value vs. setpoint
100	PREV-FLTR	PREV-FLTR	First derivative of filtered value(3) (negative)
011	Reserved	Reserved	Reserved
010	RES-FLTR	(RES-PREV)-FLTR	Actual result vs. averaged/ filtered value
001	RES-STPT	(RES-PREV)-STPT	Actual result vs.setpoint
000	RES-PREV	RES-PREV	First derivative of single measurement(2)
			Actual CVD result in CVD mode(2)

bit 3 **SOI:** ADC Stop-on-Interrupt bit

If CONT = 1:
1 = GO is cleared when the threshold conditions are met, otherwise the conversion is retriggered
0 = GO is not cleared by hardware, must be cleared by software to stop retriggers

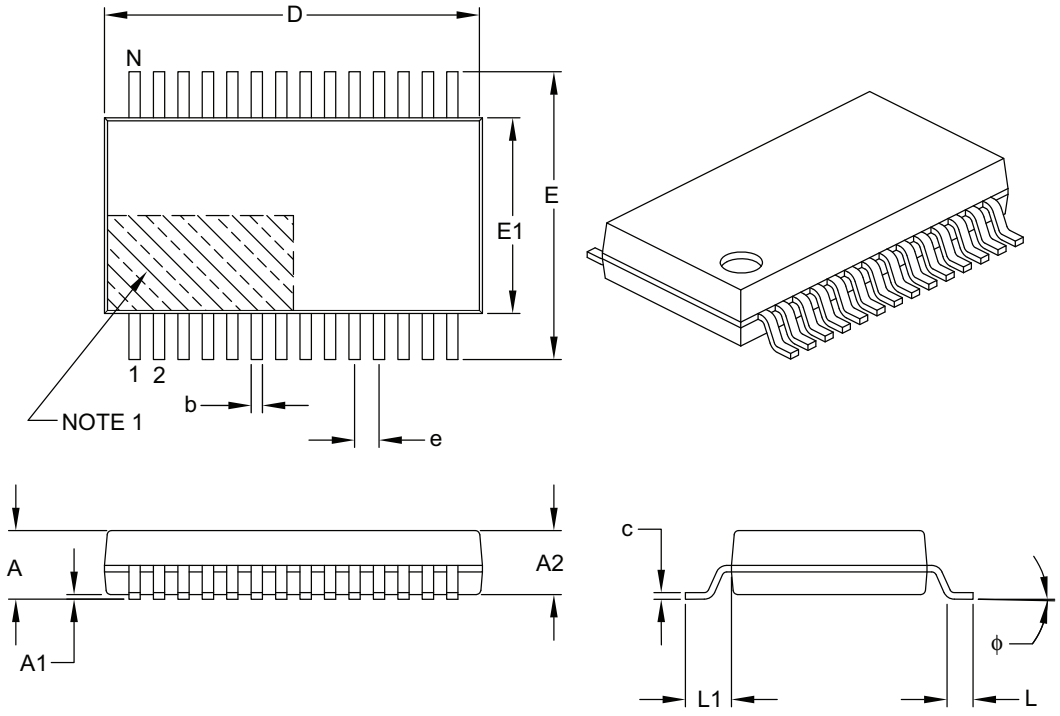
bit 2-0 **TMD<2:0>:** Threshold Interrupt Mode Select bits

111 = Interrupt regardless of threshold test results
110 = Interrupt if ERR>UTH
101 = Interrupt if ERR≤UTH
100 = Interrupt if ERR<LTH or ERR>UTH
011 = Interrupt if ERR>LTH and ERR<UTH
010 = Interrupt if ERR≥LTH
001 = Interrupt if ERR<LTH
000 = Never interrupt

Note 1: When PSIS = 0, the value of (RES-PREV) is the value of (S2-S1) from Table 36-2.
2: When PSIS = 0
3: When PSIS = 1.

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