

Welcome to [E-XFL.COM](https://www.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	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18346-i-p

PIC16(L)F18326/18346

- Up to 18 I/O Pins:
 - Individually programmable pull-ups
 - Slew rate control
 - Interrupt-on-change with edge-select
 - Input level selection control (ST or TTL)
 - Digital open-drain enable
- Peripheral Pin Select (PPS):
 - I/O pin remapping of digital peripherals
- Timer modules:
 - Timer0:
 - 8/16-bit timer/counter
 - Synchronous or asynchronous operation
 - Programmable prescaler/postscaler
 - Time base for capture/compare function
 - Timer1/3/5 with gate control:
 - 16-bit timer/counter
 - Programmable internal or external clock sources
 - Multiple gate sources
 - Multiple gate modes
 - Time base for capture/compare function
 - Timer2/4/6:
 - 8-bit timers
 - Programmable prescaler/postscaler
 - Time base for PWM function

Analog Peripherals

- 10-bit Analog-to-Digital Converter (ADC):
 - 17 external channels
 - Conversion available during Sleep
- Comparator:
 - Two comparators
 - Fixed Voltage Reference at non-inverting input(s)
 - Comparator outputs externally accessible
- 5-bit Digital-to-Analog Converter (DAC):
 - 5-bit resolution, rail-to-rail
 - Positive Reference Selection
 - Unbuffered I/O pin output
 - Internal connections to ADCs and comparators
- Voltage Reference:
 - Fixed Voltage Reference with 1.024V, 2.048V and 4.096V output levels Flexible Oscillator Structure
- High-Precision Internal Oscillator:
 - Software-selectable frequency range up to 32 MHz
 - $\pm 1\%$ at nominal 4 MHz calibration point
- 4x PLL with External Sources
- Low-Power Internal 31 kHz Oscillator (LFINTOSC)
- External Low-Power 32 kHz Crystal Oscillator (SOSC)
- External Oscillator Block with:
 - Three Crystal/Resonator modes up to 20 MHz
 - Three External Clock modes up to 20 MHz
 - Fail-Safe Clock Monitor
 - Detects clock source failure
 - Oscillator Start-up Timer (OST)
 - Ensures stability of crystal oscillator sources

Pin Allocation Tables

TABLE 2: 14/16-PIN ALLOCATION TABLE (PIC16(L)F18326)

I/O ⁽²⁾	14-Pin PDIP/SOIC/TSSOP	16-Pin UQFN	ADC	Reference	Comparator	NCO	DAC	DSM	Timers	CCP	PWM	CWG	MSSP	EUSART	CLC	CLKR	Interrupt	Pull-up	Basic
RA0	13	12	ANA0	—	C1IN0+	—	DAC1OUT	—	—	—	—	—	SS2 ⁽¹⁾	—	—	—	IOC	Y	ICDDAT/ ICSPDAT
RA1	12	11	ANA1	VREF+	C1IN0- C2IN0-	—	DAC1REF+	—	—	—	—	—	—	—	—	—	IOC	Y	ICDCLK/ ICSPCLK
RA2	11	10	ANA2	VREF-	—	—	DAC1REF-	—	T0CKI ⁽¹⁾	CCP3 ⁽¹⁾	—	CWG1IN ⁽¹⁾ CWG2IN ⁽¹⁾	—	—	—	—	INT ⁽¹⁾ IOC	Y	—
RA3	4	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOC	Y	MCLR V _{PP}
RA4	3	2	ANA4	—	—	—	—	—	T1G ⁽¹⁾ SOSCO	—	—	—	—	—	—	—	IOC	Y	CLKOUT OSC2
RA5	2	1	ANA5	—	—	—	—	—	T1CKI ⁽¹⁾ SOSCIN SOSCI	—	—	—	—	—	CLCIN3 ⁽¹⁾	—	IOC	Y	CLKIN OSC1
RC0	10	9	ANC0	—	C2IN0+	—	—	—	T5CKI ⁽¹⁾	—	—	—	SCK1 ⁽¹⁾ SCL1 ^(1,3,4)	—	—	—	IOC	Y	—
RC1	9	8	ANC1	—	C1IN1- C2IN1-	—	—	—	—	CCP4 ⁽¹⁾	—	—	SD1 ⁽¹⁾ SDA1 ^(1,3,4)	—	CLCIN2 ⁽¹⁾	—	IOC	Y	—
RC2	8	7	ANC2	—	C1IN2- C2IN2-	—	—	MDCIN1 ⁽¹⁾	—	—	—	—	—	—	—	—	IOC	Y	—
RC3	7	6	ANC3	—	C1IN3- C2IN3-	—	—	MDMIN ⁽¹⁾	T5G ⁽¹⁾	CCP2 ⁽¹⁾	—	—	SS1 ⁽¹⁾	—	CLCIN0 ⁽¹⁾	—	IOC	Y	—
RC4	6	5	ANC4	—	—	—	—	—	T3G ⁽¹⁾	—	—	—	SCK2 ⁽¹⁾ SCL2 ^(1,3,4)	—	CLCIN1 ⁽¹⁾	—	IOC	Y	—
RC5	5	4	ANC5	—	—	—	—	MDCIN2 ⁽¹⁾	T3CKI ⁽¹⁾	CCP1 ⁽¹⁾	—	—	SDI2 ⁽¹⁾ SDA2 ^(1,3,4)	RX ⁽¹⁾	—	—	IOC	Y	—
VDD	1	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD

- Note**
- 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.
 - 2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.
 - 3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.
 - 4: These pins are configured for I²C logic levels; clock and data signals may be assigned to any of these pins. Assignments to the other pins (e.g., RA5) will operate, but logic levels will be standard TTL/ST as selected by the INLVL register.

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18326 PIC16(L)F18346	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 14												
CPU CORE REGISTERS; see Table 4-2 for specifics												
70Ch	—	—	Unimplemented								—	—
70Dh	—	—	Unimplemented								—	—
70Eh	—	—	Unimplemented								—	—
70Fh	—	—	Unimplemented								—	—
710h	—	—	Unimplemented								—	—
711h	CWG2CLKCON		—	—	—	—	—	—	—	CS	---- --0	---- --0
712h	CWG2DAT		—	—	—	—	DAT<3:0>				---- 0000	---- 0000
713h	CWG2DBR		—	—	DBR<5:0>						--00 0000	--00 0000
714h	CWG2DBF		—	—	DBF<5:0>						--00 0000	--00 0000
715h	CWG2CON0		EN	LD	—	—	—	MODE<2:0>			00-- -000	00-- -000
716h	CWG2CON1		—	—	IN	—	POLD	POLC	POLB	POLA	--x- 0000	--x- 0000
717h	CWG2AS0		SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	0001 01--	0001 01--
718h	CWG2AS1		—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	---0 0000	---0 0000
719h	CWG2STR		OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	0000 0000	0000 0000
71Ah to 71Fh	—	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: Only on PIC16F18326/18346.

Note 2: Register accessible from both User and ICD Debugger.

7.0 OSCILLATOR MODULE

7.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. [Figure 7-1](#) illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators, quartz-crystal resonators and ceramic resonators. In addition, the system clock source can be supplied from one of two internal oscillators and PLL circuits, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, ECH, ECM, ECL) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

The RSTOSC bits of Configuration Word 1 determine the type of oscillator that will be used when the device is reset, including when it is first powered-up.

The internal clock modes, LFINTOSC, HFINTOSC (set at 1 MHz), or HFINTOSC (set at 32 MHz) can be set through the RSTOSC bits.

If an external clock source is selected, the FEXTOSC bits of Configuration Word 1 must be used in conjunction with the RSTOSC bits to select the External Clock mode.

The external oscillator module can be configured in one of the following clock modes by setting the FEXTOSC<2:0> bits of Configuration Word 1:

1. ECL – External Clock Low-Power mode (<= 100 kHz)
2. ECM – External Clock Medium-Power mode (<= 8 MHz)
3. ECH – External Clock High-Power mode (above 8 MHz)
4. LP – 32 kHz Low-Power Crystal mode.
5. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode (between 100 kHz and 4 MHz)
6. HS – High Gain Crystal or Ceramic Resonator mode (above 4 MHz)

The ECH, ECM, and ECL Clock modes rely on an external logic level signal as the device clock source. The LP, XT, and HS Clock modes require an external crystal or resonator to be connected to the device. Each mode is optimized for a different frequency range. The INTOSC internal oscillator block produces low and high-frequency clock sources, designated LFINTOSC and HFINTOSC. (see Internal Oscillator Block, [Figure 7-1](#)).

7.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) bits of the OSCCON1 register. The following clock sources can be selected:

- External Oscillator (EXTOSC)
- High-Frequency Internal Oscillator (HFINTOSC)
- Low-Frequency Internal Oscillator (LFINTOSC)
- Secondary Oscillator (SOSC)
- EXTOSC with 4x PLL
- HFINTOSC with 2x PLL

7.3.1 NEW OSCILLATOR SOURCE (NOSC) AND NEW DIVIDER SELECTION REQUEST (NDIV) BITS

The New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) bits of the OSCCON1 register select the system clock source and frequencies that are used for the CPU and peripherals.

When the new values of NOSC<2:0> and NDIV<3:0> are written to OSCCON1, the current oscillator selection will continue to operate as the system clock while waiting for the new source to indicate that it is stable and ready. In some cases, the newly requested source may already be in use, and is ready immediately. In the case of a divider-only change, the new and old sources are the same and are ready immediately. The device may enter Sleep while waiting for the switch as described in [Section 7.3.3 “Clock Switch and Sleep”](#).

When the new oscillator is ready, the New Oscillator is Ready (NOSCR) bit of OSCCON3 and the Clock Switch Interrupt Flag (CSWIF) bit of PIR3 become set (CSWIF = 1). If Clock Switch Interrupts are enabled (CSWIE = 1), an interrupt will be generated at that time. The Oscillator Ready (ORDY) bit of OSCCON3 can also be polled to determine when the oscillator is ready in lieu of an interrupt.

If the Clock Switch Hold (CSWHOLD) bit of OSCCON3 is clear, the oscillator switch will occur when the New Oscillator Ready bit (NOSCR) is set and the interrupt (if enabled) will be serviced at the new oscillator setting. If CSWHOLD is set, the oscillator switch is suspended, while execution continues using the current (old) clock source. When the NOSCR bit is set, software should:

- Set CSWHOLD = 0 so the switch can complete, or
- Copy COSC into NOSC<2:0> to abandon the switch.

If DOZE is in effect, the switch occurs on the next clock cycle, whether or not the CPU is operating during that cycle.

Changing the clock post-divider without changing the clock source (i.e., changing Fosc from 1 MHz to 2 MHz) is handled in the same manner as a clock source change, as described previously. The clock source will already be active, so the switch is relatively quick. CSWHOLD must be clear (CSWHOLD = 0) for the switch to complete.

The current COSC and CDIV are indicated in the OSCCON2 register up to the moment when the switch actually occurs, at which time OSCCON2 is updated and ORDY is set. NOSCR is cleared by hardware to indicate that the switch is complete.

7.3.2 PLL INPUT SWITCH

Switching between the PLL and any non-PLL source is managed as described above. The input to the PLL is established when NOSC<2:0> selects the PLL, and maintained by the COSC setting.

When NOSC<2:0> and COSC select the PLL with different input sources, the system continues to run using the COSC setting, and the new source is enabled per NOSC<2:0>. When the new oscillator is ready (and CSWHOLD = 0), system operation is suspended while the PLL input is switched and the PLL acquires lock.

7.3.3 CLOCK SWITCH AND SLEEP

If OSCCON1 is written with a new value and the device is put to Sleep before the switch completes, the switch will not take place and the device will enter Sleep mode.

When the device wakes from Sleep and the CSWHOLD bit is clear, the device will wake with the ‘new’ clock active, and the Clock Switch Interrupt Flag bit (CSWIF) will be set.

When the device wakes from Sleep and the CSWHOLD bit is set, the device will wake with the ‘old’ clock active and the new clock will be requested again.

8.3 Interrupts During Sleep

All interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the `SLEEP` instruction. The instruction directly after the `SLEEP` instruction will always be executed before branching to the ISR. Refer to **Section 9.0 “Power-Saving Operation Modes”** for more details.

8.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the PIE0 register. The INTEDG bit of the INTCON register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the PIR0 register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

8.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for $\overline{\text{TO}}$ and $\overline{\text{PD}}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

PIC16(L)F18326/18346

11.2 Data EEPROM

Data EEPROM consists of 256 bytes of user data memory. The EEPROM provides storage locations for 8-bit user defined data.

EEPROM can be read and/or written through:

- FSR/INDF indirect access ([Section 11.3 “FSR and INDF Access”](#))
- NVMREG access ([Section 11.4 “NVMREG Access”](#))
- In-Circuit Serial Programming (ICSP)

Unlike Program Flash Memory, which must be written to by row, EEPROM can be written to byte-by-byte.

11.3 FSR and INDF Access

The FSR and INDF registers allow indirect access to the Program Flash Memory or EEPROM.

11.3.1 FSR READ

With the intended address loaded into an FSR register, a `MOVIW` instruction or read of INDF will read data from the Program Flash Memory or EEPROM. The CPU operation is suspended during the read, and resumes immediately after. Read operations return a single word of memory. When the MSB of the FSR (ex: `FSRxH`) is set to 0x70, the lower 8-bit address value (in `FSRxL`) determines the EEPROM location that may be read from (through the INDF register). In other words, the EEPROM address range 0x00-0xFF is mapped into the FSR address space between 0x7000-0x70FF. Writing to the EEPROM cannot be accomplished via the FSR/INDF interface.

11.3.2 FSR WRITE

Writing/erasing the NVM through the FSR registers (ex. `MOVWI` instruction) is not supported in the PIC16(L)F18326/18346 devices.

11.4 NVMREG Access

The NVMREG interface allows read/write access to all the locations accessible by FSRs, and also read/write access to the User ID locations and EEPROM, and read-only access to the device identification, revision, and Configuration data.

Reading, writing, or erasing of NVM via the NVMREG interface is prevented when the device is code-protected.

11.4.1 NVMREG READ OPERATION

To read a NVM location using the NVMREG interface, the user must:

1. Clear the NVMREGS bit of the NVMCON1 register if the user intends to access Program Flash Memory locations, or set NVMREGS if the user intends to access User ID, Configuration, or EEPROM locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair ([Table 11-2](#)).
3. Set the RD bit of the NVMCON1 register to initiate the read.

Once the read control bit is set, the CPU operation is suspended during the read, and resumes immediately after. The data is available in the very next cycle, in the NVMDATH:NVMDATL register pair; therefore, it can be read as two bytes in the following instructions.

NVMDATH:NVMDATL register pair will hold this value until another read or until it is written to by the user.

Upon completion, the RD bit is cleared by hardware.

PIC16(L)F18326/18346

11.4.2 NVM UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the NVM from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- Program Flash Memory Row Erase
- Load of Program Flash Memory write latches
- Write of Program Flash Memory write latches to Program Flash Memory memory
- Write of Program Flash Memory write latches to User IDs
- Write to EEPROM

The unlock sequence consists of the following steps and must be completed in order:

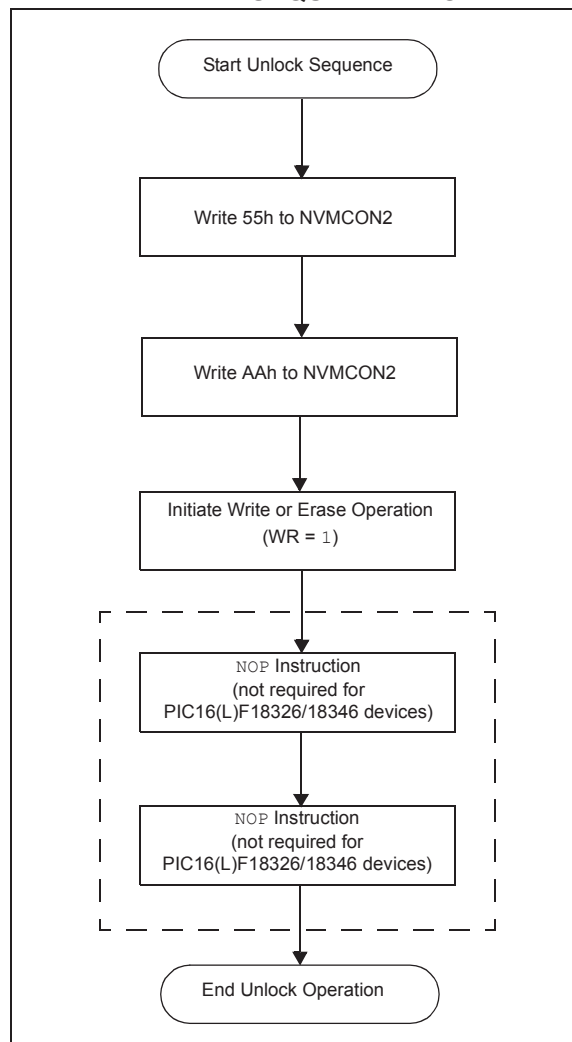
- Write 55h to NVMCON2
- Write AAh to NVMCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

Note: The two NOP instructions after setting the WR bit, which were required in previous devices, are not required for PIC16(L)F18326/18346 devices. See Figure 11-2.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

FIGURE 11-2: NVM UNLOCK SEQUENCE FLOWCHART



EXAMPLE 11-2: NVM UNLOCK SEQUENCE

BANKSEL	NVMCON1	
BSF	NVMCON1, WREN	; Enable write/erase
MOVLW	55h	; Load 55h
BCF	INTCON, GIE	; Recommended so sequence is not interrupted
MOVWF	NVMCON2	; Step 1: Load 55h into NVMCON2
MOVLW	AAh	; Step 2: Load W with AAh
MOVWF	NVMCON2	; Step 3: Load AAh into NVMCON2
BSF	NVMCON1, WR	; Step 4: Set WR bit to begin write/erase
BSF	INTCON, GIE	; Re-enable interrupts

Note 1: Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown.

2: Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

PIC16(L)F18326/18346

REGISTER 12-19: LATC: PORTC DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
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 7-6 **LATC<7:6>**: PORTC Output Latch Value bits⁽¹⁾

bit 5-0 **LATC<5:0>**: PORTC Output Latch Value bits

Note 1: PIC16(L)F18346 only; otherwise read as '0'.

20.9 Operation During Sleep

The CWGx module will operate during Sleep, provided that the input sources remain active.

If the HFINTOSC is selected as the module clock source, dead-band generation will remain active. This will have a direct effect on the Sleep mode current.

20.10 Configuring the CWG

1. Ensure that the TRIS control bits corresponding to CWG outputs are set so that all are configured as inputs, ensuring that the outputs are inactive during setup. External hardware may ensure that pin levels are held to safe levels.
2. Clear the EN bit, if not already cleared.
3. Configure the MODE<2:0> bits of the CWGxCON0 register to set the output operating mode.
4. Configure the POLy bits of the CWGxCON1 register to set the output polarities.
5. Configure the DAT<3:0> bits of the CWGxDAT register to select the data input source.
6. If a Steering mode is selected, configure the STRy bits to select the desired output on the CWG outputs.
7. Configure the LSB<1:0> and LSAC<1:0> bits of the CWGxAS0 register to select the auto-shutdown output override states (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
8. If auto-restart is desired, set the REN bit of CWGxAS0.
9. If auto-shutdown is desired, configure the ASxE bits of the CWGxAS1 register to select the shutdown source.
10. Set the desired rising and falling dead-band times with the CWGxDBR and CWGxDBF registers.
11. Select the clock source in the CWGxCLKCON register.
12. Set the EN bit to enable the module.
13. Clear the TRIS bits that correspond to the CWG outputs to set them as outputs.
14. If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit in software to start the CWG.

PIC16(L)F18326/18346

REGISTER 20-7: CWGxAS1: CWG AUTO-SHUTDOWN CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E
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	q = Value depends on condition

- bit 7-5 **Unimplemented:** Read as '0'
- bit 4 **AS4E:** CWG Auto-Shutdown Source 4 (CLC4) Enable bit
 1 = Auto-shutdown for CLC4 is enabled
 0 = Auto-shutdown for CLC4 is disabled
- bit 3 **AS3E:** CWG Auto-Shutdown Source 3 (CLC2) Enable bit
 1 = Auto-shutdown from CLC2 is enabled
 0 = Auto-shutdown from CLC2 is disabled
- bit 2 **AS2E:** CWG Auto-Shutdown Source 2 (C2) Enable bit
 1 = Auto-shutdown from Comparator 2 is enabled
 0 = Auto-shutdown from Comparator 2 is disabled
- bit 1 **AS1E:** CWG Auto-Shutdown Source 1 (C1) Enable bit
 1 = Auto-shutdown from Comparator 1 is enabled
 0 = Auto-shutdown from Comparator 1 is disabled
- bit 0 **AS0E:** CWG Auto-Shutdown Source 0 (CWGxPPS) Enable bit
 1 = Auto-shutdown from CWGxPPS is enabled
 0 = Auto-shutdown from CWGxPPS is disabled

REGISTER 20-8: CWGxDBR: CWGx RISING DEAD-BAND COUNT REGISTER

U-0	U-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
—	—	DBR<5:0>					
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	q = Value depends on condition

- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-0 **DBR<5:0>:** CWG Rising Edge Triggered Dead-Band Count bits
 11 1111 = 63-64 CWG clock periods
 11 1110 = 62-63 CWG clock periods
 .
 .
 .
 00 0010 = 2-3 CWG clock periods
 00 0001 = 1-2 CWG clock periods
 00 0000 = 0 CWG clock periods. Dead-band generation is bypassed.

REGISTER 22-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
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 7-0 **ADRES<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

REGISTER 22-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>	—	—	—	—	—	—	—
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 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

PIC16(L)F18326/18346

REGISTER 23-4: NCO1ACCH: NCO1 ACCUMULATOR REGISTER – HIGH BYTE

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
NCO1ACC<15:8>							
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 7-0 **NCO1ACC<15:8>**: NCO1 Accumulator, high byte

REGISTER 23-5: NCO1ACCU: NCO1 ACCUMULATOR REGISTER – UPPER BYTE⁽¹⁾

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCO1ACC<19:16>			
bit 7				bit			

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 7-4 **Unimplemented**: Read as '0'

bit 3-0 **NCO1ACC<19:16>**: NCO1 Accumulator, upper byte

Note 1: The accumulator spans registers NCO1ACCU:NCO1ACCH:NCO1ACCL. The 24 bits are reserved but not all are used. This register updates in real-time, asynchronously to the CPU; there is no provision to ensure atomic access to this 24-bit space using an 8-bit bus. Writing to this register while the module is operating will produce undefined results.

REGISTER 23-6: NCO1INCL^(1,2): NCO1 INCREMENT REGISTER – LOW BYTE

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-1/1
NCO1INC<7:0>							
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 7-0 **NCO1INC<7:0>**: NCO1 Increment, low byte

Note 1: The logical increment spans NCO1INCUN:NCO1INCH:NCO1INCL.

Note 2: NCO1INC is double-buffered as INCBUF; INCBUF is updated on the next falling edge of NCOCLK after writing to NCO1INCL; NCO1INCUN and NCO1INCH should be written prior to writing NCO1INCL.

26.8 Register Definitions: Timer0 Register

REGISTER 26-1: TMR0L: TIMER0 COUNT 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
TMR0L<7:0>							
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 7-0 **TMR0L<7:0>**:TMR0 Counter bits 7..0

REGISTER 26-2: TMR0H: TIMER0 PERIOD 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
TMR0H<7:0> or TMR0<15:8>							
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 7-0 When T016BIT = 0
TMR0H<7:0>:TMR0 Period Register Bits 7..0
When T016BIT = 1
TMR0<15:8>: TMR0 Counter bits 15..8

30.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

30.3 I²C Mode Overview

The Inter-Integrated Circuit (I²C) bus is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing.

The I²C bus specifies two signal connections:

- Serial Clock (SCL)
- Serial Data (SDA)

Figure 30-11 shows the block diagram of the MSSPx module when operating in I²C mode.

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 30-2 and Figure 30-3 show a typical connection between two processors configured as master and slave devices.

The I²C bus can operate with one or more master devices and one or more slave devices.

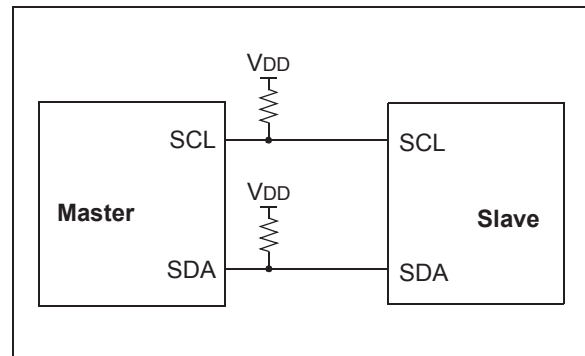
There are four potential modes of operation for a given device:

- Master Transmit mode
(master is transmitting data to a slave)
- Master Receive mode
(master is receiving data from a slave)
- Slave Transmit mode
(slave is transmitting data to a master)
- Slave Receive mode
(slave is receiving data from the master)

To begin communication, the master device sends out a Start condition followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an ACK. The master then continues in either transmit or receive data from the slave device.

FIGURE 30-11: I²C MASTER/SLAVE CONNECTION



On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends a NACK in place of the last ACK bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send a Restart condition in place of the Stop bit or last ACK bit when it is in Receive mode.

The I²C bus specifies three message protocols;

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.

30.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

30.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 30-19 displays a standard waveform of a 7-bit address slave transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with $\overline{R/\overline{W}}$ bit set. After the eighth falling edge of the SCL line the CKP bit is cleared and SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and $\overline{R/\overline{W}}$ and D/A of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to ACK or not ACK and sets the ACKDT bit of the SSPxCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the \overline{ACK} value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the ACK if the R/W bit is set.
11. Slave software clears SSPxIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

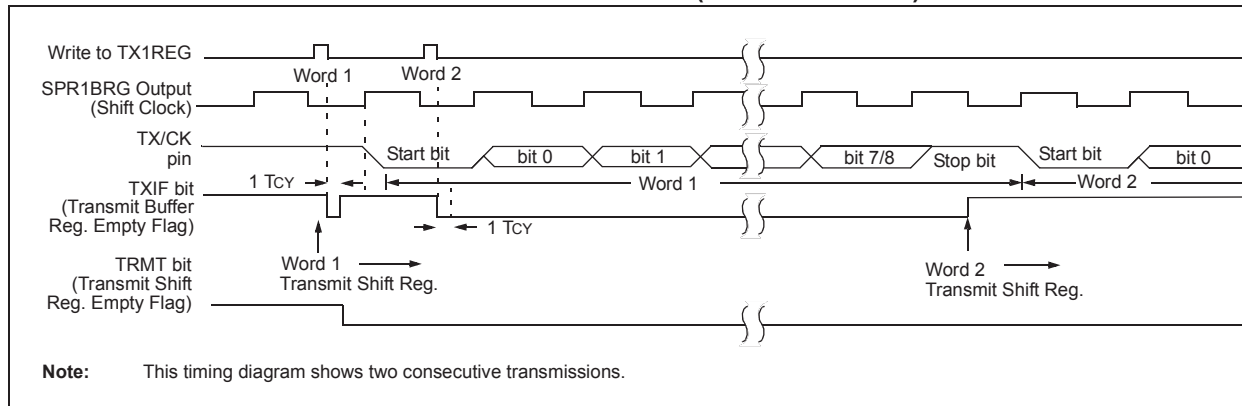
Note: SSPxBUF cannot be loaded until after the \overline{ACK} .

13. Slave sets the CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an \overline{ACK} value on the ninth SCL pulse.
15. Slave hardware copies the \overline{ACK} value into the ACKSTAT bit of the SSPxCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not \overline{ACK} the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not \overline{ACK} on the last byte to ensure that the slave releases the SCL line to receive a Stop.

PIC16(L)F18326/18346

FIGURE 31-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



31.1.2 EUSART1 ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 31-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART1 receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RC1REG register.

31.1.2.1 Enabling the Receiver

The EUSART1 receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART1 control bits are assumed to be in their default state.

Setting the CREN bit of the RC1STA register enables the receiver circuitry of the EUSART1. Clearing the SYNC bit of the TX1STA register configures the EUSART1 for asynchronous operation. Setting the SPEN bit of the RC1STA register enables the EUSART1. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

Note: If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

31.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See Section 31.1.2.4 "Receive Framing Error" for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART1 receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RC1REG register.

Note: If the receive FIFO is overrun, no additional characters will be received until the Overrun condition is cleared. See Section 31.1.2.5 "Receive Overrun Error" for more information on overrun errors.

TABLE 35-12: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS^(1,2)

Standard Operating Conditions (unless otherwise stated) VDD = 3.0V, TA = 25°C							
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
AD01	NR	Resolution	—	—	10	bit	
AD02	EIL	Integral Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD03	EDL	Differential Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD04	EOFF	Offset Error	—	0.5	2	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD05	EGN	Gain Error	—	±0.2	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD06	VADREF	ADC Reference Voltage (ADREF+) ⁽³⁾	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	VSS	—	ADREF+	V	
AD06	VADREF	ADC Reference Voltage (ADREF+ - ADREF-) ⁽³⁾	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	ADREF-	—	ADREF+	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	10	—	kΩ	
AD09	RVREF	ADC Voltage Reference Ladder Impedance	—	—	—	kΩ	

* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

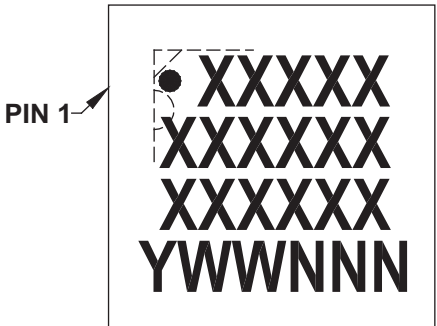
Note 1: Total Absolute Error is the sum of the offset, gain and integral non-linearity (INL) errors.

2: The ADC conversion result never decreases with an increase in the input and has no missing codes.

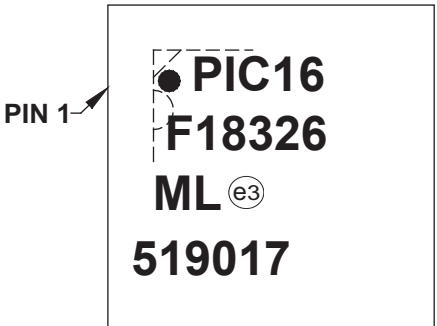
PIC16(L)F18326/18346

Package Marking Information (Continued)

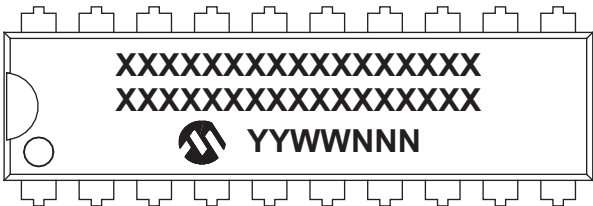
16-Lead UQFN (4x4x0.5mm)



Example



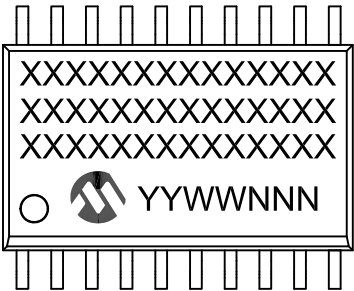
20-Lead PDIP (300 mil)



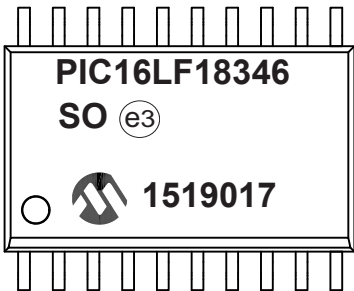
Example



20-Lead SOIC (7.50 mm)



Example



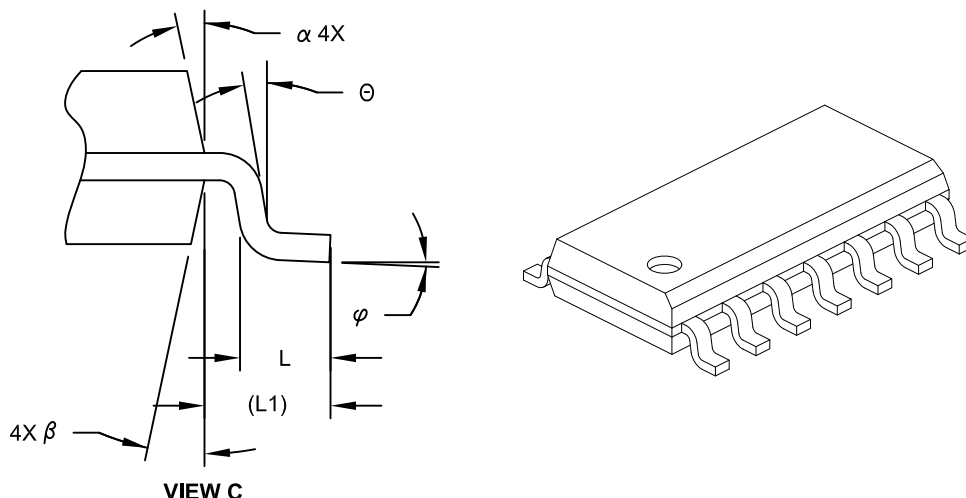
Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

PIC16(L)F18326/18346

14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

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		14		
Pitch	e		1.27 BSC		
Overall Height	A		-	-	1.75
Molded Package Thickness	A2		1.25	-	-
Standoff §	A1		0.10	-	0.25
Overall Width	E		6.00 BSC		
Molded Package Width	E1		3.90 BSC		
Overall Length	D		8.65 BSC		
Chamfer (Optional)	h		0.25	-	0.50
Foot Length	L		0.40	-	1.27
Footprint	L1		1.04 REF		
Lead Angle	θ		0°	-	-
Foot Angle	φ		0°	-	8°
Lead Thickness	c		0.10	-	0.25
Lead Width	b		0.31	-	0.51
Mold Draft Angle Top	α		5°	-	15°
Mold Draft Angle Bottom	β		5°	-	15°

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- 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.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2