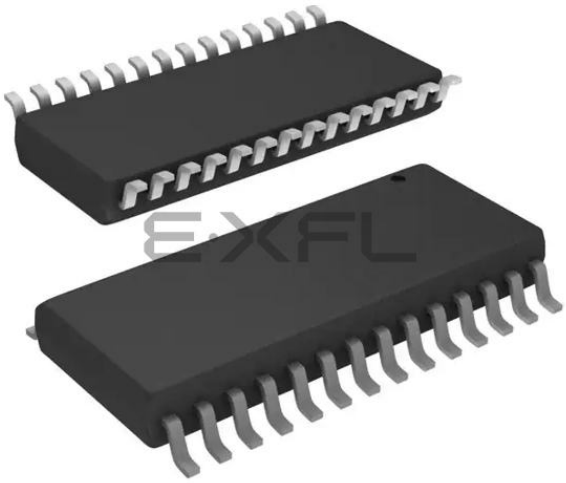


Welcome to E-XFL.COM

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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf27k42-e-so

6.1 Power-on Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or MCLR features can be used to extend the start-up period until all device operation conditions have been met.

6.2 Brown-out Reset (BOR)

The BOR circuit holds the device in Reset when VDD reaches a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Words. The four operating modes are:

- BOR is always on
- BOR is off when in Sleep
- BOR is controlled by software
- BOR is always off

Refer to Table 6-1 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV<1:0> bits in Configuration Words.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Table 44-13 for more information.

6.2.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Words are programmed to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

6.2.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Words are programmed to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

6.2.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Words are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.

6.2.4 BOR AND BULK ERASE

BOR is forced ON during PFM Bulk Erase operations to make sure that a safe erase voltage is maintained for a successful erase cycle.

During Bulk Erase, the BOR is enabled at 2.45V for F and LF devices, even if it is configured to some other value. If VDD falls, the erase cycle will be aborted, but the device will not be reset.

TABLE 11-3: SUMMARY OF REGISTERS ASSOCIATED WITH WINDOWED WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
WDTCON0	—	—	PS<4:0>					SEN	182
WDTCON1	—	CS<2:0>			—	WINDOW<2:0>			183
WDTPSL	PSCNT<7:0>								184
WDTPSH	PSCNT<15:8>								184
WDTTMR	WDTTMR<4:0>					STATE	PSCNT<17:16>		185

Legend: `x` = unknown, `u` = unchanged, – = unimplemented locations read as '`0`'. Shaded cells are not used by Windowed Watchdog Timer.

13.1.3 READING THE PROGRAM FLASH MEMORY

The `TBLRD` instruction retrieves data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

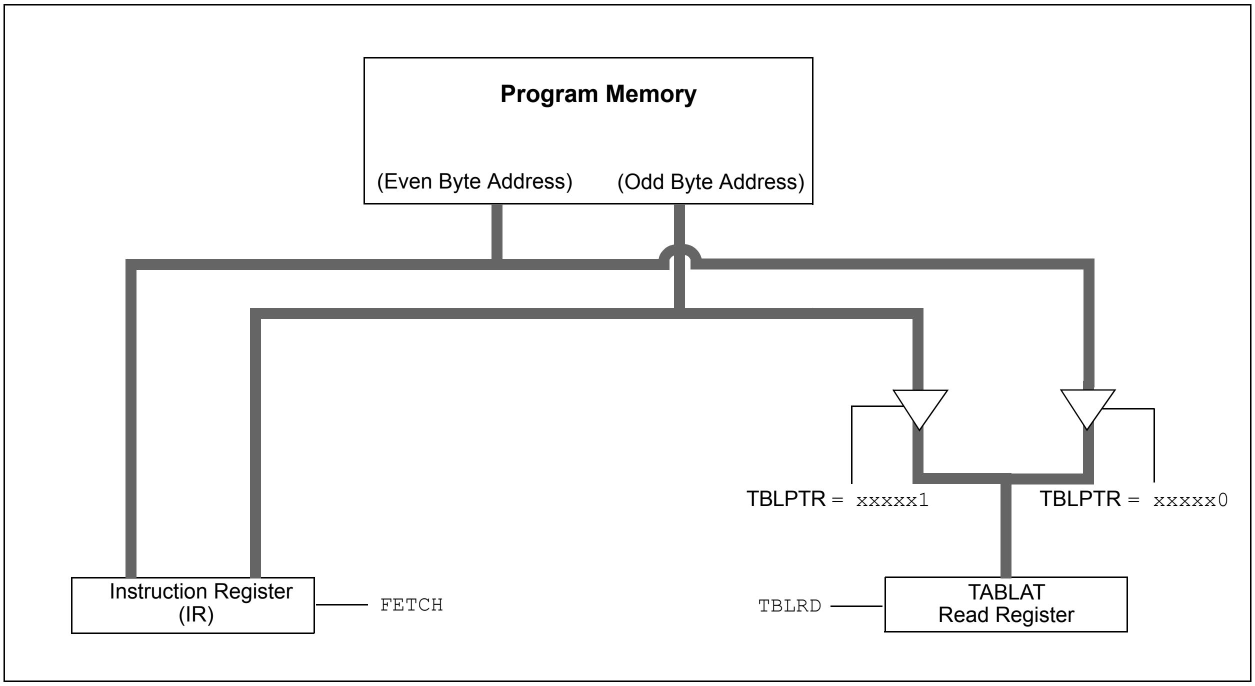
TBLPTR points to a byte address in program space. Executing `TBLRD` places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The CPU operation is suspended during the read, and it resumes immediately after. From the user point of view, TABLAT is valid in the next instruction cycle.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word.

Figure 13-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 13-4: READS FROM PROGRAM FLASH MEMORY



EXAMPLE 13-1: READING A PROGRAM FLASH MEMORY WORD

```
            BCF       NVMCON1, REG0             ; point to Program Flash Memory
            BSF       NVMCON1, REG1             ; access Program Flash Memory
            MOVLW     CODE_ADDR_UPPER           ; Load TBLPTR with the base
            MOVWF     TBLPTRU                   ; address of the word
            MOVLW     CODE_ADDR_HIGH
            MOVWF     TBLPTRH
            MOVLW     CODE_ADDR_LOW
            MOVWF     TBLPTRL
READ_WORD
            TBLRD*+                             ; read into TABLAT and increment
            MOVF      TABLAT, W                 ; get data
            MOVWF     WORD_EVEN
            TBLRD*+                             ; read into TABLAT and increment
            MOVFW     TABLAT, W                 ; get data
            MOVF      WORD_ODD
```

14.2 CRC Functional Overview

The CRC module can be used to detect bit errors in the program memory using the built-in memory scanner or through user input RAM memory. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 14-1: CRC EXAMPLE

Rev. 10-000206A
1/8/2014

CRC-16-ANSI

$x^{16} + x^{15} + x^{2} + 1$ (17 bits)

Standard 16-bit representation = 0x8005

```
CRCXORH = 0b10000000
CRCXORL = 0b0000010- (1)
```

Data Sequence:
0x55, 0x66, 0x77, 0x88

```
DLEN = 0b0111
PLEN = 0b1111
```

Data entered into the CRC:
SHIFTM = 0:

```
01010101 01100110 01110111 10001000
```

SHIFTM = 1:

```
10101010 01100110 11101110 00010001
```

Check Value (ACCM = 1):

SHIFTM = 0: `0x32D6`

```
CRCACCH = 0b00110010
CRCACCL = 0b11010110
```

SHIFTM = 1: `0x6BA2`

```
CRCACCH = 0b01101011
CRCACCL = 0b10100010
```

Note 1: Bit 0 is unimplemented. The LSb of any CRC polynomial is always ‘1’ and will always be treated as a ‘1’ by the CRC for calculating the CRC check value. This bit will be read in software as a ‘0’.

15.0 DIRECT MEMORY ACCESS (DMA)

15.1 Introduction

The Direct Memory Access (DMA) module is designed to service data transfers between different memory regions directly without intervention from the CPU. By eliminating the need for CPU-intensive management of handling interrupts intended for data transfers, the CPU now can spend more time on other tasks.

PIC18(L)F26/27/45/46/47/55/56/57K42 family has two DMA modules which can be independently programmed to transfer data between different memory locations, move different data sizes, and use a wide range of hardware triggers to initiate transfers. The two DMA registers can even be programmed to work together, in order to carry out more complex data transfers without CPU overhead.

Key features of the DMA module include:

- Support access to the following memory regions:
  - GPR and SFR space (R/W)
  - Program Flash Memory (R only)
  - Data EEPROM Memory (R only)
- Programmable priority between the DMA and CPU Operations. Refer to **Section 3.1 “System Arbitration”** for details.
- Programmable Source and Destination address modes
  - Fixed address
  - Post-increment address
  - Post-decrement address
- Programmable Source and Destination sizes
- Source and destination pointer register, dynamically updated and reloadable
- Source and destination count register, dynamically updated and reloadable
- Programmable auto-stop based on Source or Destination counter
- Software triggered transfers
- Multiple user selectable sources for hardware triggered transfers
- Multiple user selectable sources for aborting DMA transfers

15.2 DMA Registers

The operation of the DMA module has the following registers:

- Control registers (DMAxCON0, DMAxCON1)
- Data buffer register (DMAxBUF)
- Source Start Address Register (DMAxSSAU:H:L)
- Source Pointer Register (DMAxSPTRU:H:L)
- Source Message Size Register (DMAxSSZH:L)
- Source Count Register (DMAxSCNTH:L)
- Destination Start Address Register (DMAxDSAH:L)
- Destination Pointer Register (DMAxDPTRH:L)
- Destination Message Size Register (DMAxDSZH:L)
- Destination Count Register (DMAxDCNTH:L)
- Start Interrupt Request Source Register (DMAxSIRQ)
- Abort Interrupt Request Source Register (DMAxAIRQ)

These registers are detailed in **Section 15.13 “Register definitions: DMA”**.

EXAMPLE 16-2: INITIALIZING PORTE

```
CLRF    PORTE       ;Initialize PORTE by
                    ;clearing output
                    ;data latches
CLRF    LATE        ;Alternate method
                    ;to clear output
                    ;data latches
CLRF    ANSELE      ;Configure analog pins
                    ;for digital only
MOVLW   05h         ;Value used to
                    ;initialize data
                    ;direction
MOVWF   TRISE       ;Set RE<0> as input
                    ;RE<1> as output
                    ;RE<2> as input
```

16.3.2 PORTE ON 28-PIN DEVICES

For PIC18(L)F26/27K42 devices, PORTE is only available when Master Clear functionality is disabled (MCLRE = 0). In this case, PORTE is a single bit, input-only port comprised of RE3 only. The pin operates as previously described. RE3 in PORTE register is a read-only bit and will read '1' when MCLRE = 1 (i.e., Master Clear enabled).

16.3.3 RE3 WEAK PULL-UP

The port RE3 pin has an individually controlled weak internal pull-up. When set, the WPUE3 bit enables the RE3 pin pull-up. When the RE3 port pin is configured as $\overline{\text{MCLR}}$, (CONFIG2L, MCLRE = 1 and CONFIG4H, LVP = 0), or configured for Low-Voltage Programming, (MCLRE = x and LVP = 1), the pull-up is always enabled and the WPUE3 bit has no effect.

16.3.4 INTERRUPT-ON-CHANGE

The interrupt-on-change feature is available only on the RE3 pin of PORTE for all devices. If MCLRE = 1 or LVP = 1, RE3 port functionality is disabled and interrupt-on-change on RE3 is not available. For further details refer to **Section 18.0 "Interrupt-on-Change"**.

REGISTER 16-3: LATx: LATx REGISTER(1)

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
LATx7	LATx6	LATx5	LATx4	LATx3	LATx2	LATx1	LATx0
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 **LATx<7:0>:** Rx7:Rx0 Output Latch Value bits

Note 1: Writes to LATx are equivalent with writes to the corresponding PORTx register. Reads from LATx register return register values, not I/O pin values.

TABLE 16-4: LAT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
LATD(1)	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0
LATE(1)	—	—	—	—	—	LATE2	LATE1	LATE0
LATF(2)	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0

Note 1: Unimplemented in PIC18(L)F26/27K42.
2: Unimplemented in PIC18(L)F26/45/46/47K42.

REGISTER 16-9: RxyI2C: I^2C PAD Rxy CONTROL REGISTER

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	SLEW	PU<1:0>		—	—	TH<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	HS = Hardware set

bit 7 **Unimplemented**: Read as ‘0’

bit 6 **SLEW**: I^2C specific slew rate limiting is enabled
- 1 = I^2C specific slew rate limiting is enabled. Standard pad slew limiting is disabled. The SLRxy bit is ignored.
- 0 = Standard GPIO Slew Rate; enabled/disabled via SLRxy bit.

bit 5-4 **PU<1:0>**: I^2C Pull-up Selection bits
- 11 = Reserved
- 10 = 10x current of standard weak pull-up
- 01 = 2x current of standard weak pull-up
- 00 = Standard GPIO weak pull-up, enabled via WPUxy bit

bit 3-2 **Unimplemented**: Read as ‘0’

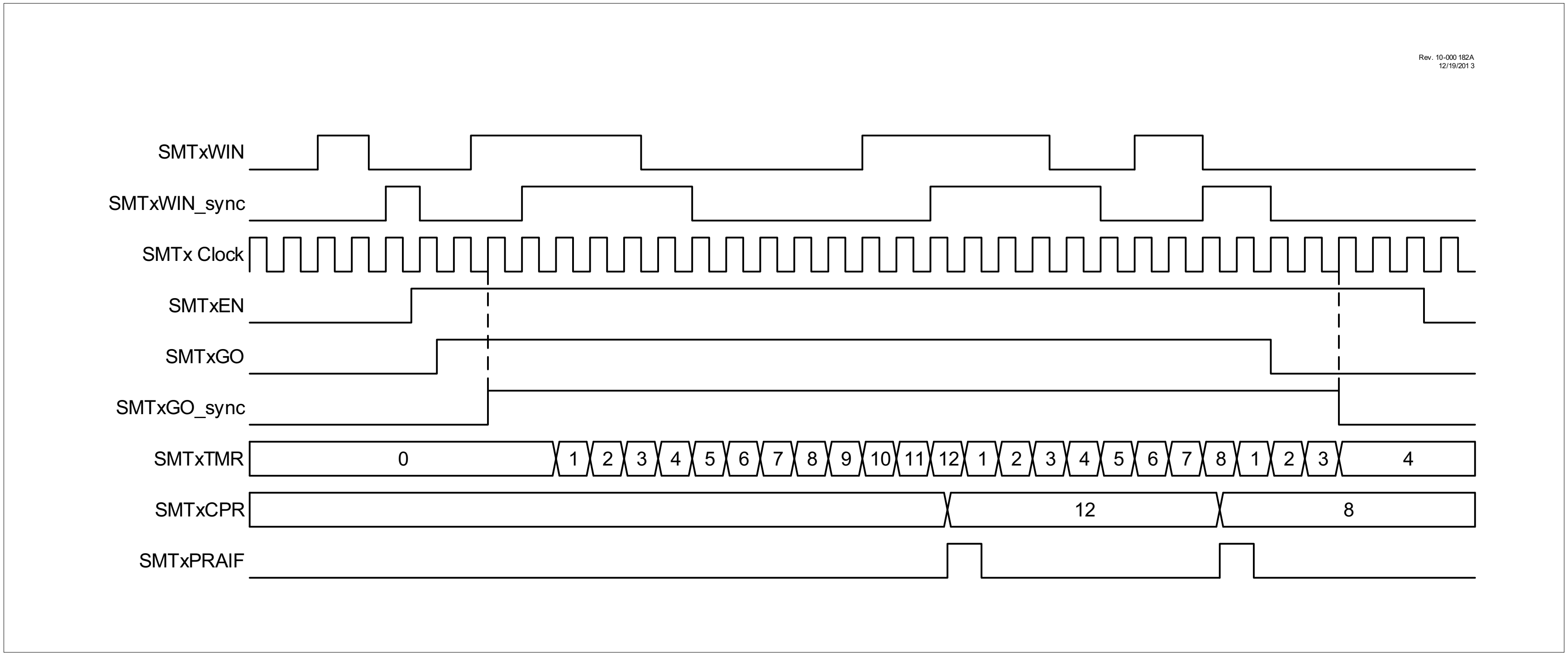
bit 1-0 **TH<1:0>**: I^2C Input Threshold Selection bits
- 11 = SMBus 3.0 (1.35 V) input threshold
- 10 = SMBus 2.0 (2.1 V) input threshold
- 01 = I^2C specific input thresholds
- 00 = Standard GPIO Input pull-up, enabled via INLVLxy registers

TABLE 16-10: I2C PAD CONTROL REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RB1I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>	
RB2I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>	
RC3I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>	
RC4I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>	
RD0I2C(1)	—	SLEW	PU<1:0>		—	—	TH<1:0>	
RD1I2C(1)	—	SLEW	PU<1:0>		—	—	TH<1:0>	

Note 1: Unimplemented in PIC18(L)F26/27K42.

FIGURE 25-10: WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



Preliminary

REGISTER 26-3: CWGxCLK: CWGx CLOCK INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	CS
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 7-1 **Unimplemented:** Read as '0'

bit 0 **CS:** CWG Clock Source Selection bits

CS	CWG1	CWG2	CWG3
1	HFINTOSC (1)	HFINTOSC (1)	HFINTOSC (1)
0	FOSC	FOSC	FOSC

Note 1: HFINTOSC remains operating during Sleep.

28.0 NUMERICALLY CONTROLLED OSCILLATOR (NCO) MODULE

The Numerically Controlled Oscillator (NCO) module is a timer that uses overflow from the addition of an increment value to divide the input frequency. The advantage of the addition method over simple counter driven timer is that the output frequency resolution does not vary with the divider value. The NCO is most useful for applications that require frequency accuracy and fine resolution at a fixed duty cycle.

Features of the NCO include:

- 20-bit Increment Function
- Fixed Duty Cycle mode (FDC) mode
- Pulse Frequency (PF) mode
- Output Pulse-Width Control
- Multiple Clock Input Sources
- Output Polarity Control
- Interrupt Capability

Figure 28-1 is a simplified block diagram of the NCO module.

28.8 NCO Control Registers

REGISTER 28-1: NCO1CON: NCO CONTROL REGISTER

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
EN	—	OUT	POL	—	—	—	PFM
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:** NCO1 Enable bit 1 = NCO1 module is enabled 0 = NCO1 module is disabled
bit 6	**Unimplemented**: Read as ‘0’
bit 5	**OUT:** NCO1 Output bit Displays the current output value of the NCO1 module.
bit 4	**POL:** NCO1 Polarity 1 = NCO1 output signal is inverted 0 = NCO1 output signal is not inverted
bit 3-1	**Unimplemented**: Read as ‘0’
bit 0	**PFM:** NCO1 Pulse Frequency Mode bit 1 = NCO1 operates in Pulse Frequency mode 0 = NCO1 operates in Fixed Duty Cycle mode, divide by 2

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 32-8: SPIxCON1: SPI CONFIGURATION REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
SMP	CKE	CKP	FST	—	SSP	SDIP	SDOP
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’

bit 7 **SMP**: SPI Input Sample Phase Control bit
Slave mode:
1 = Reserved
0 = SDI input is sampled in the middle of data output time
Master mode:
1 = SDI input is sampled at the end of data output time
0 = SDI input is sampled in the middle of data output time

bit 6 **CKE**: Clock Edge Select bit
1 = Output data changes on transition from active to idle clock state
0 = Output data changes on transition from idle to active clock state

bit 5 **CKP**: Clock Polarity Select bit
1 = Idle state for SCK is high level
0 = Idle state for SCK is low level

bit 4 **FST**: Fast Start Enable bit
Slave mode:
This bit is ignored
Master mode:
1 = Delay to first SCK may be less than ½ baud period
0 = Delay to first SCK will be at least ½ baud period

bit 3 **Unimplemented**: Read as ‘0’

bit 2 **SSP**: SS Input/Output Polarity Control bit
1 = SS is active-low
0 = SS is active-high

bit 1 **SDIP**: SDI Input Polarity Control bit
1 = SDI input is active-low
0 = SDI input is active-high

bit 0 **SDOP**: SDI Output Polarity Control bit
1 = SDO output is active-low
0 = SDO output is active-high

36.7 Register Definitions: ADC Control

REGISTER 36-1: ADCON0: ADC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W/HC-0
ON	CONT	—	CS	—	FM	—	GO
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 **ON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled

bit 6 **CONT:** ADC Continuous Operation Enable bit
1 = GO is retriggered upon completion of each conversion trigger until ADTIF is set (if ADSOI is set) or until GO is cleared (regardless of the value of ADSOI)
0 = ADC is cleared upon completion of each conversion trigger

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **CS:** ADC Clock Selection bit
1 = Clock supplied from FRC dedicated oscillator
0 = Clock supplied by FOSC, divided according to ADCLK register

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **FM:** ADC results Format/alignment Selection
1 = ADRES and PREV data are right-justified
0 = ADRES and PREV data are left-justified, zero-filled

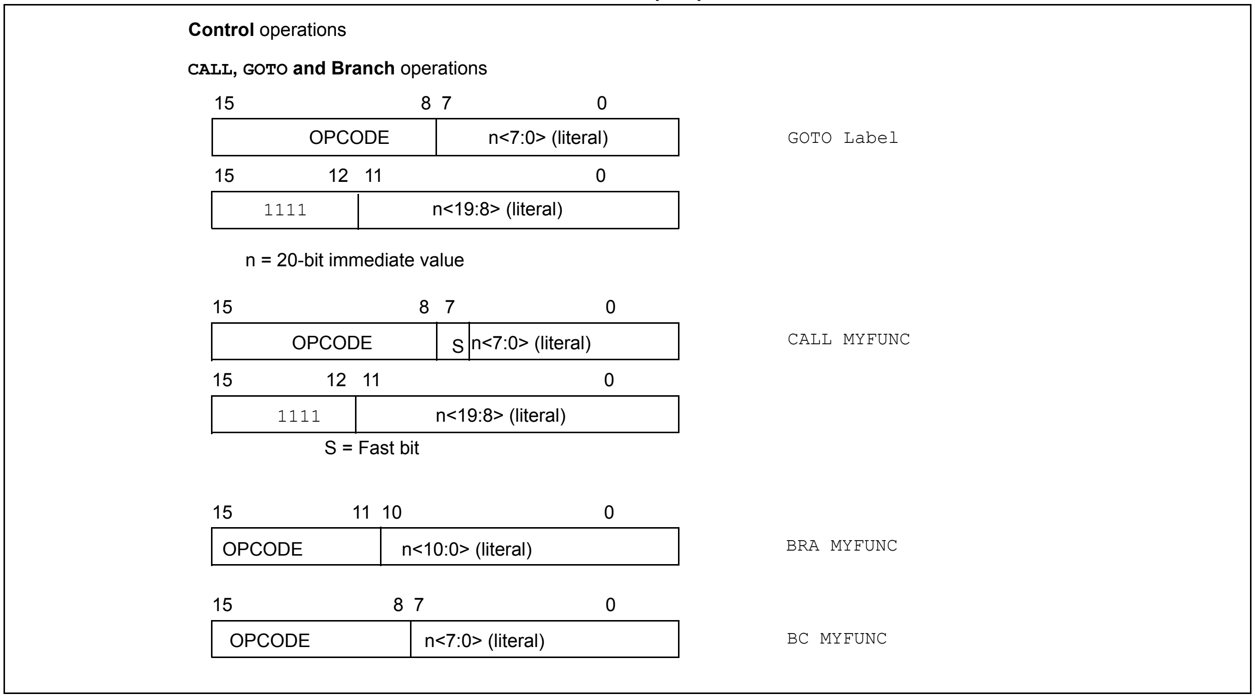
bit 1 **Unimplemented:** Read as ‘0’

bit 0 **GO:** ADC Conversion Status bit(1)
1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle. The bit is cleared by hardware as determined by the CONT bit
0 = ADC conversion completed/not in progress

Note 1: This bit requires ON bit to be set.

2: If cleared by software while a conversion is in progress, the results of the conversion up to this point will be transfered to ADRES and the state machine will be reset, but the ADIF interrupt flag bit will not be set; filter and threshold operations will not be performed.

FIGURE 41-2: General Format for Instructions (2/2)



POP — Pop Top of Return Stack

Syntax: POP

Operands: None

Operation: (TOS) $\rightarrow$ bit bucket

Status Affected: None

Encoding:

0000	0000	0000	0110

Description: The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	POP TOS value	No operation

Example:

```
POP
GOTO    NEW
```

Before Instruction
TOS = 0031A2h
Stack (1 level down) = 014332h

After Instruction
TOS = 014332h
PC = NEW

PUSH — Push Top of Return Stack

Syntax: PUSH

Operands: None

Operation: (PC + 2) $\rightarrow$ TOS

Status Affected: None

Encoding:

0000	0000	0000	0101

Description: The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	PUSH PC + 2 onto return stack	No operation	No operation

Example:

```
PUSH
```

Before Instruction
TOS = 345Ah
PC = 0124h

After Instruction
PC = 0126h
TOS = 0126h
Stack (1 level down) = 345Ah

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3A98h	—	Reserved, maintain as '0'								
3A97h-3A95h	—	Unimplemented								
3A94h	INLVLF(3)	INLVLF7	INLVLF6	INLVLF5	INLVLF4	INLVLF3	INLVLF2	INLVLF1	INLVLF0	270
3A93h	SLRCONF(3)	SLRCONF7	SLRCONF6	SLRCONF5	SLRCONF4	SLRCONF3	SLRCONF2	SLRCONF1	SLRCONF0	269
3A92h	ODCONF(3)	ODCONF7	ODCONF6	ODCONF5	ODCONF4	ODCONF3	ODCONF2	ODCONF1	ODCONF0	268
3A91h	WPUF(3)	WPUF7	WPUF6	WPUF5	WPUF4	WPUF3	WPUF2	WPUF1	WPUF0	267
3A90h	ANSELF(3)	ANSELF7	ANSELF6	ANSELF5	ANSELF4	ANSELF3	ANSELF2	ANSELF1	ANSELF0	266
3A8Fh-3A8Ah	—	Unimplemented								
3A89h	—	Reserved, maintain as '0'								
3A88h	—	Reserved, maintain as '0'								
3A87h	IOCEF	—	—	—	—	IOCEF3	—	—	—	287
3A86h	IOCEN	—	—	—	—	IOCEN3	—	—	—	287
3A85h	IOCEP	—	—	—	—	IOCEP3	—	—	—	287
3A84h	INLVLE	—	—	—	—	INLVLE3	INLVLE2(2)	INLVLE1(2)	INLVLE0(2)	270
3A83h	SLRCONE(2)	—	—	—	—	—	SRLE2(2)	SRLE1(2)	SRLE0(2)	269
3A82h	ODCONE(2)	—	—	—	—	—	ODCE2(2)	ODCE1(2)	ODCE0(2)	268
3A81h	WPUE	—	—	—	—	WPUE3	WPUE2(2)	WPUE1(2)	WPUE0(2)	267
3A80h	ANSELE(2)	ANSELE7	ANSELE6	ANSELE5	ANSELE4	ANSELE3	ANSELE2	ANSELE1	ANSELE0	266
3A7Fh-3A7CH	—	Unimplemented								
3A7Bh	RD1I2C(2)	—	IOCEN3	PU		—	—	TH		263
3A7Ah	RD0I2C(2)	—	IOCEN3	PU		—	—	TH		263
3A79h	—	Reserved, maintain as '0'								
3A78h	—	Reserved, maintain as '0'								
3A77h-3A75h	—	Unimplemented								
3A74h	INLVLD(2)	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1	INLVLD0	270
3A73h	SLRCOND(2)	SRLD7	SRLD6	SRLD5	SRLD4	SRLD3	SRLD2	SRLD1	SRLD0	269
3A72h	ODCOND(2)	ODCD7	ODCD6	ODCD5	ODCD4	ODCD3	ODCD2	ODCD1	ODCD0	268
3A71h	WPUD(2)	WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0	267
3A70h	ANSELD(2)	ANSELD7	ANSELD6	ANSELD5	ANSELD4	ANSELD3	ANSELD2	ANSELD1	ANSELD0	266
3A6Fh-3A6Ch	—	Unimplemented								
3A6Bh	RC4I2C	—	SLEW	PU		—	—	TH		263
3A6Ah	RC3I2C	—	SLEW	PU		—	—	TH		263
3A69h	—	Reserved, maintain as '0'								
3A68h	—	Reserved, maintain as '0'								
3A67h	IOCCF	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	287
3A66h	IOCCN	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	287
3A65h	IOCCP	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	287
3A64h	INLVLC	INLVLC7	INLVLC6	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	270
3A63h	SLRCONC	SLRC7	SLRC6	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	269
3A62h	ODCONC	ODCC7	ODCC6	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	268
3A61h	WPUC	WPUC7	WPUC6	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	267
3A60h	ANSELC	ANSELC7	ANSELC6	ANSELC5	ANSELC4	ANSELC3	ANSELC2	ANSELC1	ANSELC0	266
3A5Fh - 3A5Ch	—	Unimplemented								

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

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3989h	IPR9	—	—	—	—	CLC3IP	CWG3IP	CCP3IP	TMR6IP	165
3988h	IPR8	TMR5GIP	TMR5IP	—	—	—	—	—	—	164
3987h	IPR7	—	—	INT2IP	CLC2IP	CWG2IP	-	CCP2IP	TMR4IP	164
3986h	IPR6	TMR3GIP	TMR3IP	U2IP	U2EIP	U2TXIP	U2RXIP	I2C2EIP	I2C2IP	163
3985h	IPR5	I2C2TXIP	I2C2RXIP	DMA2AIP	DMA2ORIP	DMA2DCN-TIP	DMA2SCN-TIP	C2IP	INT1IP	162
3984h	IPR4	CLC1IP	CWG1IP	NCO1IP	—	CCP1IP	TMR2IP	TMR1GIP	TMR1IP	161
3983h	IPR3	TMR0IP	U1IP	U1EIP	U1TXIP	U1RXIP	I2C1EIP	I2C1IP	I2C1TXIP	160
3982h	IPR2	I2C1RXIP	SPI1IP	SPI1TXIP	SPI1RXIP	DMA1AIP	DMA1ORIP	DMA1DCN-TIP	DMA1SCNTIP	159
3981h	IPR1	SMT1PWAIP	SMT1PRAIP	SMT1IP	C1IP	ADTIP	ADIP	ZCDIP	INT0IP	158
3980h	IPR0	IOCIP	CRCIP	SCANIP	NVMIP	CSWIP	OSFIP	HLVDIP	SWIP	157
397Fh - 397Eh	—	Unimplemented								
397Dh	SCANTRIG	—	—	—	—	TSEL				226
397Ch	SCANCON0	EN	TRIGEN	SGO	—	—	MREG	BURSTMD	BUSY	222
397Bh	SCANHADRU	—	—	HADR						224
397Ah	SCANHADRH	HADR								225
3979h	SCANHADRL	HADR								225
3978h	SCANLADRU	—	—	LADR						223
3977h	SCANLADRH	LADR								223
3976h	SCANLADRL	LADR								224
3975h - 396Ah	—	Unimplemented								
3969h	CRCCON1	DLEN				PLEN				218
3968h	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	218
3967h	CRCXORH	X15	X14	X13	X12	X11	X10	X9	X8	221
3966h	CRCXORL	X7	X6	X5	X4	X3	X2	X1	—	221
3965h	CRCSHIFTH	SHFT15	SHFT14	SHFT13	SHFT12	SHFT11	SHFT10	SHFT9	SHFT8	220
3964h	CRCSHIFTL	SHFT7	SHFT6	SHFT5	SHFT4	SHFT3	SHFT2	SHFT1	SHFT0	220
3963h	CRCACCH	ACC15	ACC14	ACC13	ACC12	ACC11	ACC10	ACC9	ACC8	219
3962h	CRCACCL	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0	220
3961h	CRCDATH	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8	219
3960h	CRCDATL	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	219
395Fh	WDTTMR	WDTTMR					STATE	PSCNT		185
395Eh	WDTPSH	PSCNT								184
395Dh	WDTPSL	PSCNT								184
395Ch	WDTCON1	—	CS			—	WINDOW			183
395Bh	WDTCON0	—	—	PS					SEN	182
395Ah - 38A0h	—	Unimplemented								
389Fh	IVTADU	AD								167
389Eh	IVTADH	AD								167
389Dh	IVTADL	AD								167
389Ch - 3891h	—	Unimplemented								
3890h	PRODH_SHAD	PRODH								125
388Fh	PRODL_SHAD	PRODL								125
388Eh	FSR2H_SHAD	—	—	FSR2H						125

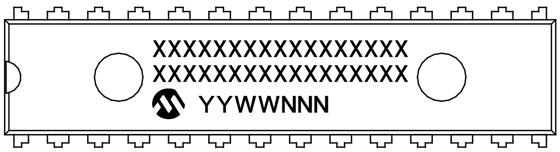
Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.

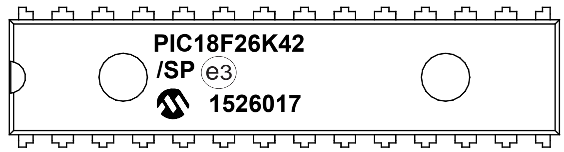
46.0 PACKAGING INFORMATION

Package Marking Information

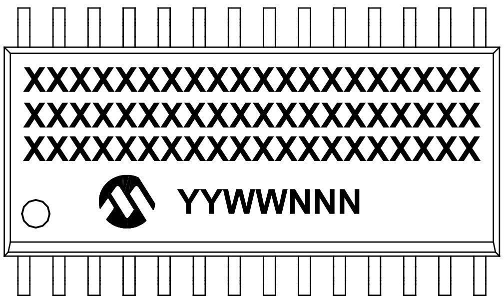
28-Lead SPDIP (.300”)



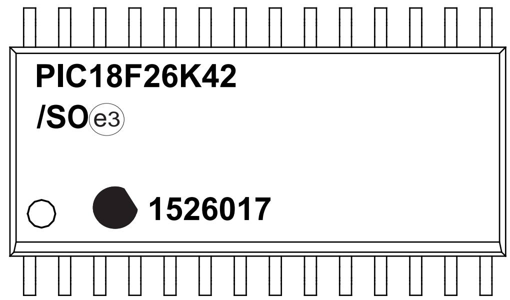
Example



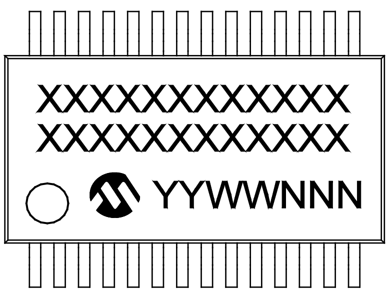
28-Lead SOIC (7.50 mm)



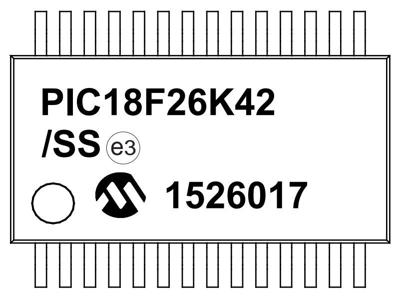
Example



28-Lead SSOP (5.30 mm)



Example



Legend:	XX...X	Customer-specific information or Microchip part number
	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.