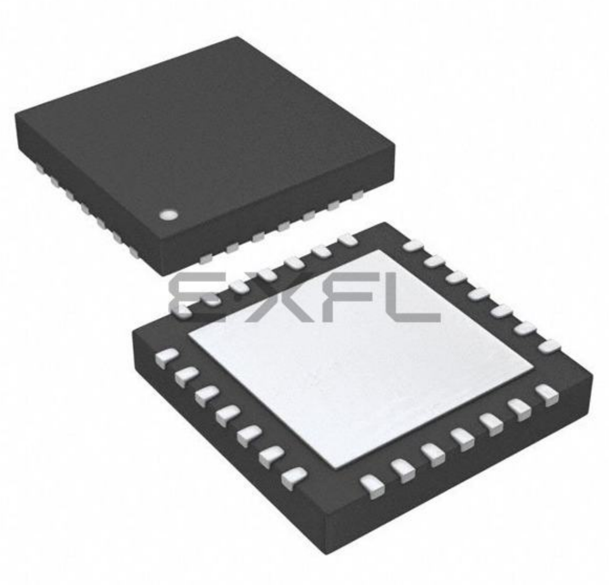


What is "Embedded - Microcontrollers"?

"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	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
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-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f24k42-i-mv

eXtreme Low-Power (XLP) Features

- Sleep mode: 60 nA @ 1.8V, typical
- Watchdog Timer: 720 nA @ 1.8V, typical
- Secondary Oscillator: 580 nA @ 32 kHz
- Operating Current:
  - 4 uA @ 32 kHz, 1.8V, typical
  - 45 uA/MHz @ 1.8V, typical

Digital Peripherals

- Three 8-Bit Timers (TMR2/4/6) with Hardware Limit Timer (HLT)
- Four 16-Bit Timers (TMR0/1/3/5)
- Four Configurable Logic Cell (CLC):
  - Integrated combinational and sequential logic
- Three Complementary Waveform Generators (CWGs):
  - Rising and falling edge dead-band control
  - Full-bridge, half-bridge, 1-channel drive
  - Multiple signal sources
  - Programmable dead band
  - Fault-shutdown input
- Four Capture/Compare/PWM (CCP) modules
- Four 10-bit Pulse Width Modulators (PWMs)
- Numerically Controlled Oscillator (NCO):
  - Generates true linear frequency control and increased frequency resolution
  - Input Clock: 0 Hz < f_{NCO} < 32 MHz
  - Resolution: f_{NCO}/220
- DSM: Data Signal Modulator
  - Multiplex two carrier clocks, with glitch prevention feature
  - Multiple sources for each carrier
- Programmable CRC with Memory Scan:
  - Reliable data/program memory monitoring for fail-safe operation (e.g., Class B)
  - Calculate CRC over any portion of program memory
- Two UART Modules:
  - Modules are Asynchronous, RS-232, RS-485 compatibility.
  - One of the UART modules supports LIN Master and Slave, DMX mode, DALI Gear and Device protocols
  - Automatic and user-timed BREAK period generation
  - DMA Compatible
  - Automatic checksums
  - Programmable 1, 1.5, and 2 stop bits
  - Wake-up on BREAK reception
  - Automatic and user-timed BREAK period generation
- One SPI module:
  - Configurable length bytes
  - Arbitrary length data packets
  - Receive-without-transmit option
  - Transmit-without-receive option
  - Transfer byte counter
  - Separate Transmit and Receive Buffers with 2-byte FIFO and DMA capabilities
- Two I^2C modules, SMBus, PMBus™ compatible:
  - Dedicated Address, Transmit and Receive buffers
  - Bus Collision Detection with arbitration
  - Bus time-out detection and handling
  - Multi-Master mode
  - Separate Transmit and Receive Buffers with 2-byte FIFO and DMA capabilities
  - I^2C, SMBus, 2.0 and SMBus 3.0, and 1.8V input level selections
- Device I/O Port Features:
  - 25 I/O pins (PIC18(L)F24/25/26/27K42)
  - 36 I/O pins (PIC18(L)F45/46/47K42)
  - 44 I/O pins (PIC18(L)F55/56/57K42)
  - One input only pin
  - Individually programmable I/O direction, open-drain, slew rate, weak pull-up control
  - High-current source/sink for direct LED drive
  - Interrupt-on-change
  - Three External Interrupt Pins
- Peripheral Pin Select (PPS):
  - Enables pin mapping of digital I/O
- Hardware Limit Timer (HLT):
  - Hardware monitoring and Fault detection
- Signal Measurement Timer (SMT):
  - 24-bit timer/counter with prescaler

4.4.3 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as either two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSb = `0`). To maintain alignment with instruction boundaries, the PC increments in steps of two and the LSb will always read ‘`0`’ (see **Section 4.2.4 “Program Counter”**).

Figure 4-3 shows an example of how instruction words are stored in the program memory.

The `CALL` and `GOTO` instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 4-3 shows how the instruction `GOTO 0006h` is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 43.0 “Instruction Set Summary”** provides further details of the instruction set.

4.4.4 MULTI-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: `CALL`, `MOVFF`, `GOTO` and `LFSR` and two three-word instructions: `MOVFL` and `MOVSFL`. In all cases, the second and the third word of the instruction always has ‘`1111`’ as its four Most Significant bits; the other 12 bits are literal data, usually a data memory address.

The use of ‘`1111`’ in the 4 MSbs of an instruction specifies a special form of `NOP`. If the instruction is executed in proper sequence – immediately after the first word – the data in the second word is accessed and used by the instruction sequence. If the first word is skipped for some reason and the second word is executed by itself, a `NOP` is executed instead. This is necessary for cases when the two-word instruction is preceded by a conditional instruction that changes the PC. Example 4-4 shows how this works.

FIGURE 4-3: INSTRUCTIONS IN PROGRAM MEMORY

			LSB = 1	LSB = 0	Word Address ↓
Program Memory Byte Locations →					000000h
					000002h
					000004h
					000006h
Instruction 1:	`MOVLW`	`055h`	0Fh	55h	000008h
Instruction 2:	`GOTO`	`0006h`	EFh	03h	00000Ah
			F0h	00h	00000Ch
Instruction 3:	`MOVFF`	`123h, 456h`	C1h	23h	00000Eh
			F4h	56h	000010h
Instruction 4:	`MOVFFL`	`123h, 456h`	00h	60h	000012h
			F4h	8Ch	000014h
			F4h	56h	000016h
					000018h
					00001Ah

5.5 Device ID and Revision ID

The 16-bit device ID word is located at 3F FFFEh and the 16-bit revision ID is located at 3F FFFCh. These locations are read-only and cannot be erased or modified.

Development tools, such as device programmers and debuggers, may be used to read the Device ID, Revision ID and Configuration Words. Refer to **15.0 "Nonvolatile Memory (NVM) Control"** for more information on accessing these locations.

5.6 Register Definitions: Device ID and Revision ID

REGISTER 5-10: DEVICE ID: DEVICE ID REGISTER

R	R	R	R	R	R	R	R
DEV15	DEV14	DEV13	DEV12	DEV11	DEV10	DEV9	DEV8
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

Legend:			
R = Readable bit	'1' = Bit is set	0' = Bit is cleared	x = Bit is unknown

bit 15-0 **DEV<15:0>:** Device ID bits

Device	Device ID
PIC18F24K42	6CA0h
PIC18F25K42	6C80h
PIC18LF24K42	6DE0h
PIC18LF25K42	6DC0h

REGISTER 9-7: OSCEN: OSCILLATOR MANUAL 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	U-0	U-0
EXTOEN	HFOEN	MFOEN	LFOEN	SOSCEN	ADOEN	—	—

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 **EXTOEN:** External Oscillator Manual Request Enable bit
1 = EXTOSC is explicitly enabled, operating as specified by FEXTOSC
0 = EXTOSC could be enabled by requesting peripheral

bit 6 **HFOEN:** HFINTOSC Oscillator Manual Request Enable bit
1 = HFINTOSC is explicitly enabled, operating as specified by OSCFRQ (Register 9-5)
0 = HFINTOSC could be enabled by requesting peripheral

bit 5 **MFOEN:** MFINTOSC (500 kHz/31.25 kHz) Oscillator Manual Request Enable bit (Derived from HFINTOSC)
1 = MFINTOSC is explicitly enabled
0 = MFINTOSC could be enabled by requesting peripheral

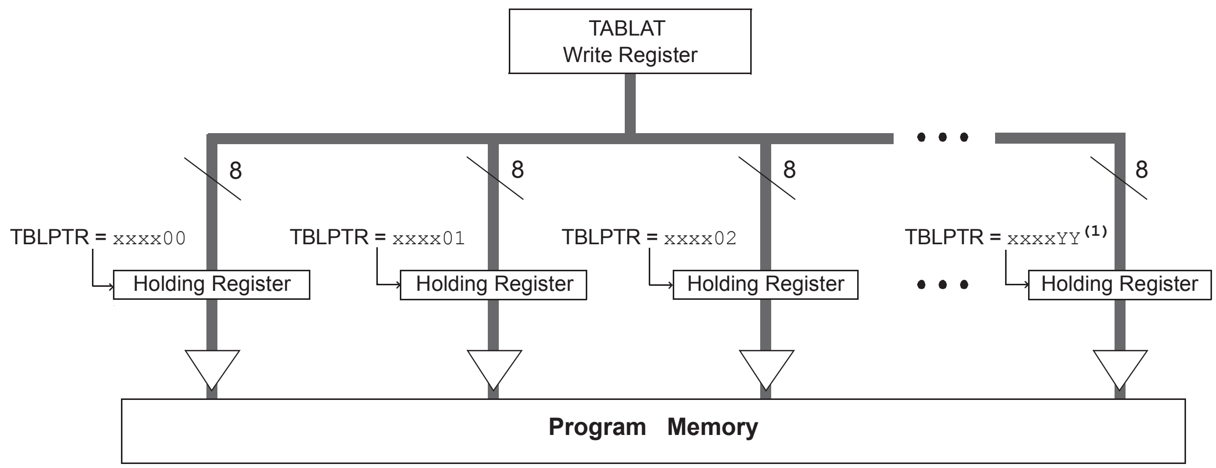
bit 4 **LFOEN:** LFINTOSC (31 kHz) Oscillator Manual Request Enable bit
1 = LFINTOSC is explicitly enabled
0 = LFINTOSC could be enabled by requesting peripheral

bit 3 **SOSCEN:** Secondary Oscillator Manual Request Enable bit
1 = Secondary Oscillator is explicitly enabled, operating as specified by SOSCPWR
0 = Secondary Oscillator could be enabled by requesting peripheral

bit 2 **ADOEN:** ADC Oscillator Manual Request Enable bit
1 = ADC oscillator is explicitly enabled
0 = ADC oscillator could be enabled by requesting peripheral

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

FIGURE 15-8: TABLE WRITES TO PROGRAM FLASH MEMORY



Note 1: Refer to Table 15-2 for number of holding registers (e.g., YY = 3F for 64 holding registers).

15.1.6.1 Program Flash Memory Write Sequence

The sequence of events for programming an internal program memory location should be:

1. Read appropriate number of bytes into RAM. Refer to Table 15-2 for Write latch size.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the block erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the n-byte block into the holding registers with auto-increment. Refer to Table 10-2 for Write latch size.
7. Set REG<1:0> bits to point to program memory.
8. Clear FREE bit and set WREN bit in NVMCON1 register.
9. Disable interrupts.
10. Execute the unlock sequence (see **Section 15.1.4 "NVM Unlock Sequence"**).
11. WR bit is set in NVMCON1 register.
12. The CPU will stall for the duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Verify the memory (table read).

This procedure will require about 6 ms to update each write block of memory. An example of the required code is given in Example 15-4.

Note:	Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the bytes in the holding registers.

TABLE 17-6: EXAMPLE DMA USE CASE TABLE

Source Module	Source Register(s)	Destination Module	Destination Register(s)	DCHxSIRQ	Comment
Signal Measurement Timer (SMT)	SMTxCPW[U:H:L]	GPR	GPR[x,y,z]	SMTxPWAIF	Store Captured Pulse-width values
	SMTxCPR[U:H:L]			SMTxPRAIF	Store Captured Period values
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x,y]	TMR0	TMR0[H:L]	TMR0IF	Use as a Timer0 reload for custom 16-bit value
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x]	TMR0	PR0	ANY	Update TMR0 frequency based on a specific trigger
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x,y]	TMR1	TMR1[H:L]	TMR1IF	Use as a Timer1 reload for custom 16-bit value
TMR1	TMR1[H:L]	GPR	GPR[x,y]	TMR1GIF	Use TMR1 Gate interrupt flag to read data out of TMR1 register
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x]	TMR2	PR2	TMR2IF	
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x,y,z]	TMR2 CCP or PWM	PR2 CCPR[H:L] or PWMDC[H:L]	ANY	Frequency generator with 50% duty cycle look up table
CCP	CCPR[H:L]	GPR	GPR[x,y]	CCPxIF	Move data from CCP 16b Capture
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x,y]	CCP	CCPR[H:L]	ANY	Load Compare value or PWM values into the CCP
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY [x,y,z,u,v,w]	CCPx CCPy CCPz	CCPxR[H:L] CCPyR[H:L] CCPzR[H:L]	ANY	Update multiple PWM values at the same time e.g. 3-phase motor control
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x,y,z]	NCO	NCOxINC[U:H:L]	ANY	Frequency Generator look up table
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x]	DAC	DACxCON0	ANY	Update DAC values
GPR/SFR/ Program Flash/ Data EEPROM	MEMORY[x]	OSCTUNE	OSCTUNE	ANY	Automated Frequency dithering

REGISTER 18-7: SLRCONx: SLEW RATE 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
SLRx7	SLRx6	SLRx5	SLRx4	SLRx3	SLRx2	SLRx1	SLRx0
bit 7							bit 0

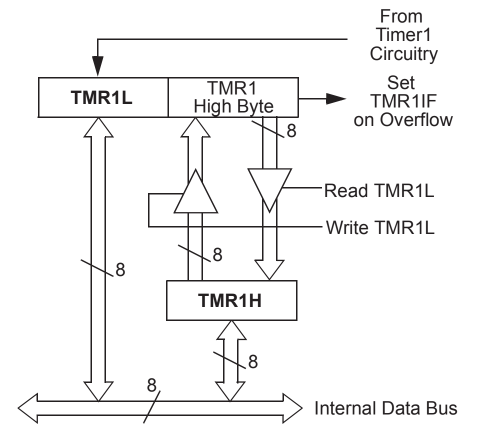
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown
-n/n = Value at POR and BOR/Value at all other Resets		

bit 7-0 **SLRx<7:0>**: Slew Rate Control on Pins Rx<7:0>, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

TABLE 18-7: SLEW RATE CONTROL REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SLRCONA	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0
SLRCONC	SLRC7	SLRC6	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0

FIGURE 23-2: TIMER1/3/5 16-BIT READ/WRITE MODE BLOCK DIAGRAM



Block Diagram of Timer1 Example of TIMER1/3/5

23.6 Timer1/3/5 Gate

Timer1/3/5 can be configured to count freely or the count can be enabled and disabled using Timer1/3/5 gate circuitry. This is also referred to as Timer1/3/5 gate enable.

Timer1/3/5 gate can also be driven by multiple selectable sources.

23.6.1 TIMER1/3/5 GATE ENABLE

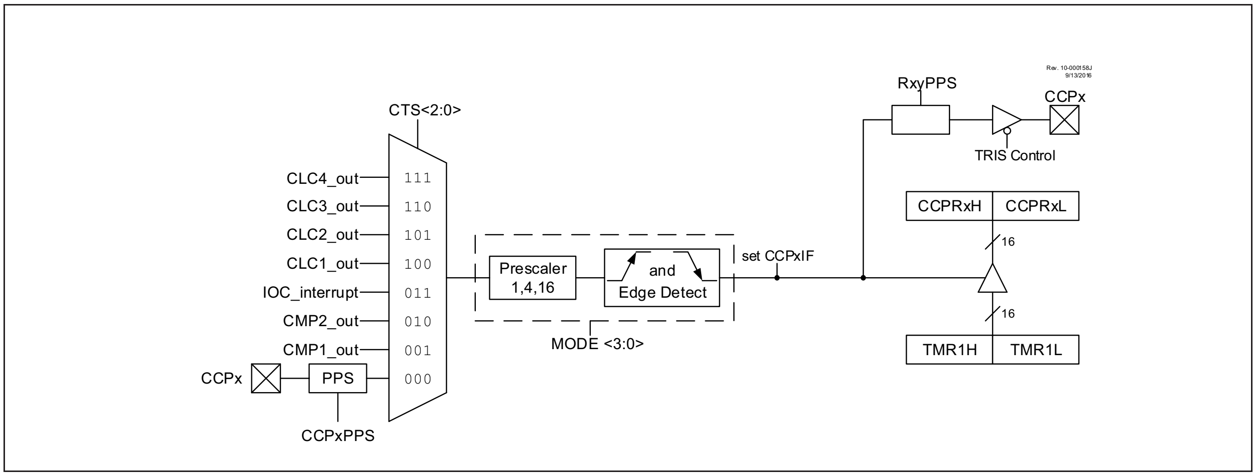
The Timer1/3/5 Gate Enable mode is enabled by setting the TMRxGE bit of the TxGCON register. The polarity of the Timer1/3/5 Gate Enable mode is configured using the TxGPOL bit of the TxGCON register.

When Timer1/3/5 Gate Enable mode is enabled, Timer1/3/5 will increment on the rising edge of the Timer1/3/5 clock source. When Timer1/3/5 Gate signal is inactive, the timer will not increment and hold the current count. See Figure 23-4 for timing details.

TABLE 23-2: TIMER1/3/5 GATE ENABLE SELECTIONS

TMRxCLK	TxGPOL	TxG	Timer1/3/5 Operation
↑	1	1	Counts
↑	1	0	Holds Count
↑	0	1	Holds Count
↑	0	0	Counts

FIGURE 25-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



27.6.2 GATED TIMER MODE

Gated Timer mode uses the SMTSIGx input to control whether or not the SMT1TMR will increment. Upon a falling edge of the external signal, the SMT1CPW register will update to the current value of the SMT1TMR. Example waveforms for both repeated and single acquisitions are provided in Figure 27-4 and Figure 27-5.

REGISTER 29-10: CLCxGLS3: GATE 3 LOGIC SELECT 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
G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N
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 **G4D4T:** Gate 3 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 3
0 = CLCIN3 (true) is not gated into CLCx Gate 3

bit 6 **G4D4N:** Gate 3 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 3
0 = CLCIN3 (inverted) is not gated into CLCx Gate 3

bit 5 **G4D3T:** Gate 3 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 3
0 = CLCIN2 (true) is not gated into CLCx Gate 3

bit 4 **G4D3N:** Gate 3 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 3
0 = CLCIN2 (inverted) is not gated into CLCx Gate 3

bit 3 **G4D2T:** Gate 3 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 3
0 = CLCIN1 (true) is not gated into CLCx Gate 3

bit 2 **G4D2N:** Gate 3 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 3
0 = CLCIN1 (inverted) is not gated into CLCx Gate 3

bit 1 **G4D1T:** Gate 4 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 3
0 = CLCIN0 (true) is not gated into CLCx Gate 3

bit 0 **G4D1N:** Gate 3 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 3
0 = CLCIN0 (inverted) is not gated into CLCx Gate 3

REGISTER 33-5: 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

34.8.3.1 Shift Register Empty Interrupt

The Shift Register Empty interrupt flag and enable are the SRMTIF and SRMTIE bits respectively. This interrupt is only available in master mode and triggers when a data transfer completes and conditions are not present to start a new transfer, as dictated by the TXR and RXR bits (see Table 34-1 for conditions for starting a new Master mode data transfer with different TXR/ RXR settings). This interrupt will be triggered at the end of the last full bit period, after SCK has been low for one ½-baud period. See Figure 30-13 for more details of the timing of this interrupt as well as other interrupts. This bit will not clear itself when the conditions for starting a new transfer occur, and must be cleared in software.

34.8.3.2 Transfer Counter is Zero Interrupt

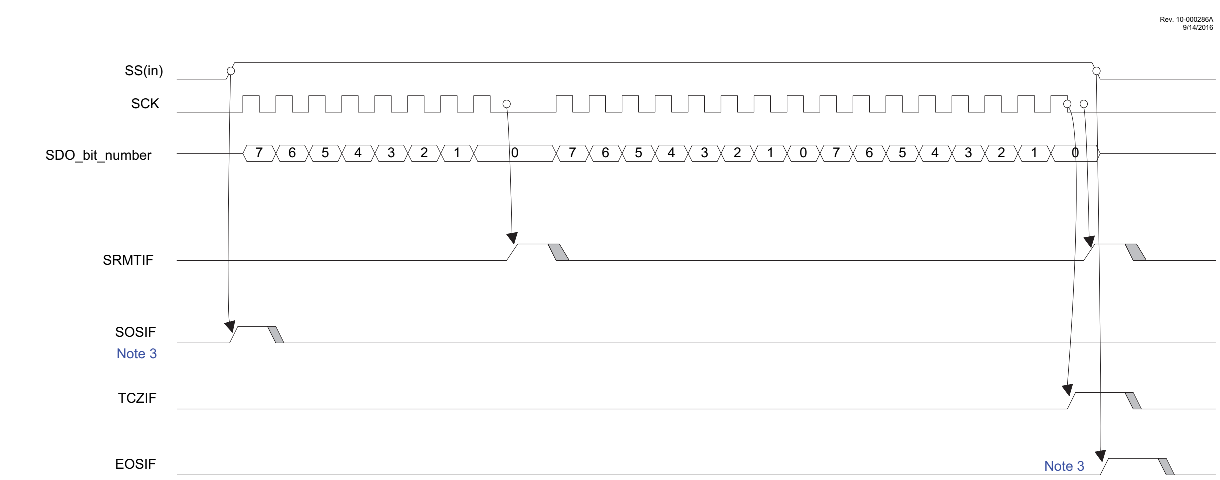
The Transfer Counter is zero interrupt flag and enable are the TCZIF and TCZIE bits, respectively. This interrupt will trigger when the transfer counter (defined by BMODE, SPIxTCTH/L and SPIxTWIDTH) decrements from one to zero. See Figure 30-13 for more details on the timing of this interrupt as well as other interrupts. This bit must be cleared in software.

Note:	The TCZIF flag only indicates that the transfer counter has decremented from one to zero, and may not indicate that the entire data transfer process is complete. Either poll the BUSY bit of SPIxCON2 and wait for it to be cleared or use the Shift Register Empty Interrupt (SRMTIF) to determine if a data transfer is fully complete.

34.8.3.3 Start of Slave Select and End of Slave Select Interrupts

The start of slave select interrupt flag and enable are the SOSIF and SOSIE bits, respectively, and the end of slave select interrupt flag and enable are similarly designated by the EOSIF and EOSIE bits. These interrupts trigger at the leading and trailing edges of the slave select input. Note that the interrupts are active in both master and slave mode, and will trigger on transitions of the slave select input regardless of which mode the SPI is in. In Master mode, PPS should be used to route the slave select input to the same pin as the slave select output, allowing these interrupts to trigger on changes to the slave select output. Also note that in slave mode, changing the SSET bit can trigger these interrupts, as it changes the effective input value of slave select. Both SOSIF and EOSIF must be cleared in software

FIGURE 34-14: TRANSFER AND SLAVE SELECT INTERRUPT TIMINGS



Note 1: SRMTIF available only in Master mode
2: Clearing of interrupt flags is shown for illustration; actual interrupt flags must be cleared in software
3: SOSIF and EOSIF are set according to SS(in), even in Master mode.

REGISTER 34-9: SPIxCON2: SPI CONFIGURATION REGISTER 2

R-0/0	R-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
BUSY	SSFLT	—	—	—	SSET	TXR(1)	RXR(1)
bit 7							bit 0

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

bit 7 **BUSY**: SPI Module Busy Status bit
1 = Data exchange is busy
0 = Data exchange is not taking place

bit 6 **SSFLT**: SS(in) Fault Status bit
If SSET = 0:
1 = SS(in) ended the transaction unexpectedly, and the data byte being received was lost
0 = SS(in) ended normally
If SSET = 1:
This bit is unchanged.

bit 5-3 **Unimplemented**: Read as '0'

bit 2 **SSET**: Slave Select Enable bit
Master mode:
1 = SS(out) is driven to the active state continuously
0 = SS(out) is driven to the active state while the transmit counter is not zero
Slave mode:
1 = SS(in) is ignored and data is clocked on all SCK(in) (as though SS = TRUE at all times)
0 = SS(in) enables/disables data input and tri-states SDO if the TRIS bit associated with the SDO pin is set (see Table 30-2 for details)

bit 1 **TXR**: Transmit Data-Required Control bit(1)
1 = TxFIFO data is required for a transfer
0 = TxFIFO data is not required for a transfer

bit 0 **RXR**: Receive FIFO Space-Required Control bit(1)
1 = Data transfers are suspended if the RxFIFO is full
0 = Received data is not stored in the FIFO

Note 1: See Table 34-1 as well as **Section 34.5 "Master mode"** and **Section 34.6 "Slave Mode"** for more details pertaining to TXR and RXR function.

2: This register should not be written to while a transfer is in progress (BUSY bit of SPIxCON2 is set).

35.3.2 BYTE FORMAT

All communication in I^2C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent by the receiver. After the 8th falling edge of the SCL line, the device transmitting data on the SDA line releases control of that pin to an input, and reads in an acknowledge value on the next clock pulse. The clock signal is provided by the master. Data is valid to change while the SCL line is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define Start and Stop conditions on the bus which are explained further in the chapter.

35.3.3 SDA AND SCL PINS

The user must configure these pins as open drain inputs. This is done by setting the appropriate TRIS and ODCON bits. The user may also select the input threshold, slew-rate and internal pull-up settings using the RxyI2C control registers (Register 35-18).

35.3.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT<1:0> bits of the I2CxCON2 register. Hold time is the time SDA is held valid after the falling edge of SCL. These bits are used to set the hold time of 30/100/300 ns. A longer hold time setting may help on buses with large capacitance.

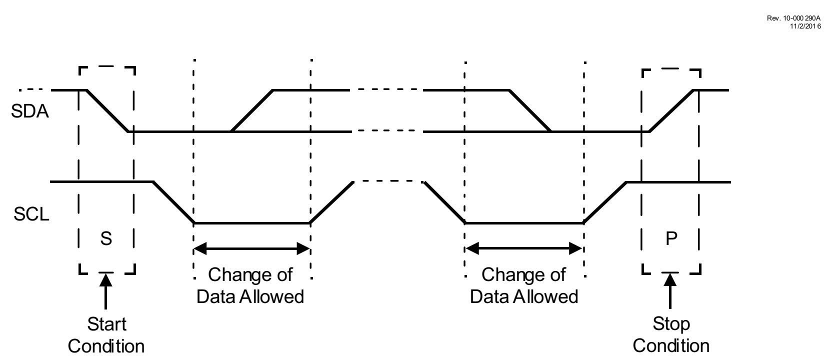
35.3.5 START CONDITION

The I^2C specification defines a Start condition as a transition of SDA line from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 35-3 shows waveforms for Start conditions. Master hardware waits for the BFRE bit of I2CxSTAT0 to be set, before asserting a Start condition on the SCL and SDA lines. If two masters assert a start at the same time, a collision will occur during the addressing phase.

35.3.6 STOP CONDITION

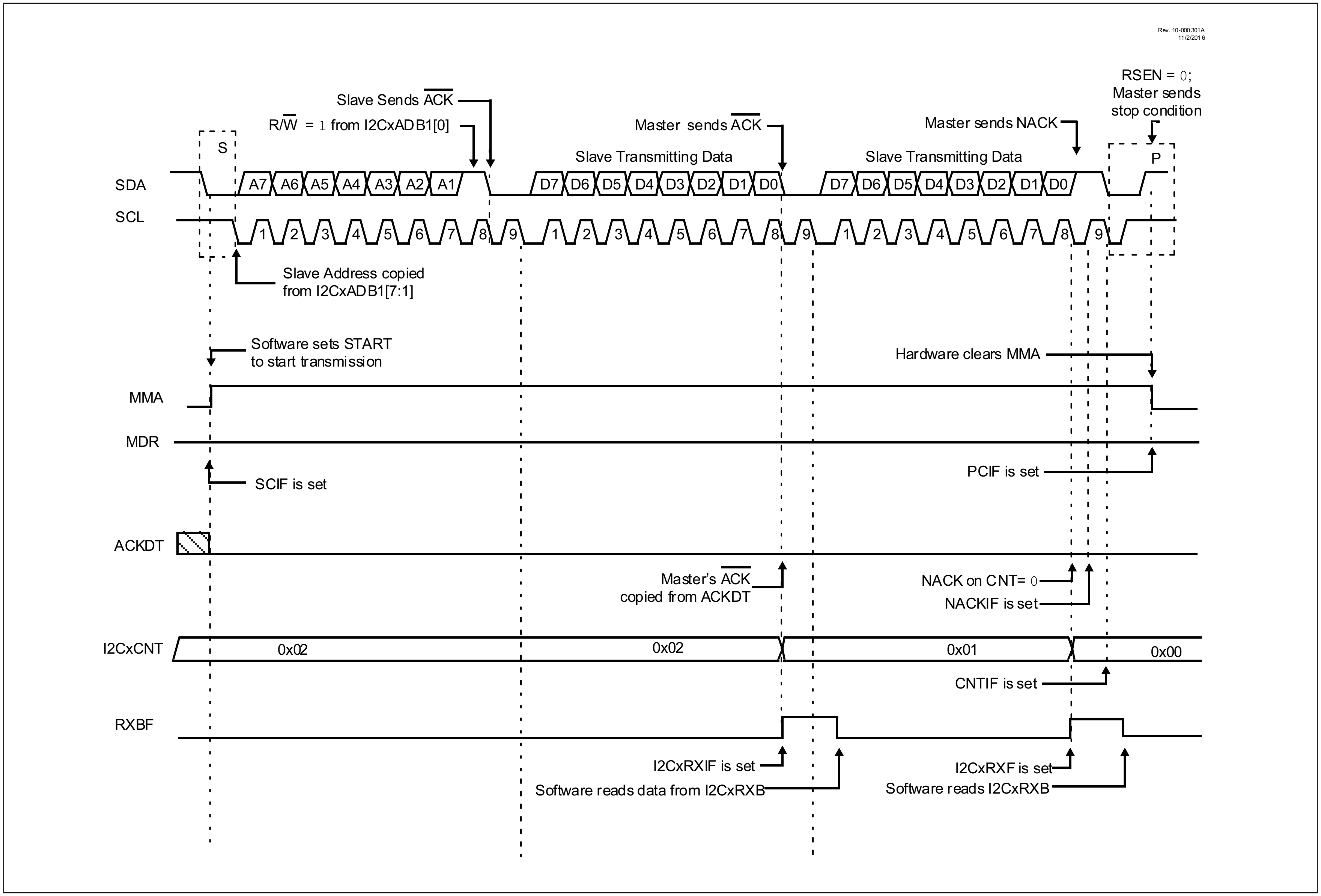
A Stop condition is a transition of the SDA line from low to high while the SCL line is high. Figure 35-3 shows waveforms for Stop conditions.

FIGURE 35-3: START AND STOP CONDITIONS



Note:	At least one SCL low time must appear before a Stop is valid. Therefore if the SDA line goes low then high again while the SCL line is high, only the Start condition is detected.

FIGURE 35-20: I²C MASTER, 7-BIT ADDRESS, RECEPTION



Preliminary

REGISTER 38-6: ADCLK: ADC CLOCK SELECTION 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
—	—	CS<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	

bit 7-6 **Unimplemented**: Read as ‘0’

bit 5-0 **CS<5:0>**: ADC Conversion Clock Select bits
111111 = FOSC/128
111110 = FOSC/126
111101 = FOSC/124
•
•
•
000000 = FOSC/2

REGISTER 38-7: ADREF: ADC REFERENCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	NREF	—	—	PREF<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 7-5 **Unimplemented**: Read as ‘0’

bit 4 **NREF**: ADC Negative Voltage Reference Selection bit
1 = VREF- is connected to external VREF-
0 = VREF- is connected to VSS

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

bit 1-0 **PREF**: ADC Positive Voltage Reference Selection bits
11 = VREF+ is connected to internal Fixed Voltage Reference (FVR) module
10 = VREF+ is connected to external VREF+
01 = Reserved
00 = VREF+ is connected to VDD

TABLE 46-2: SUPPLY CURRENT (IDD)[1,2,4]

PIC18LF24/25K42			Standard Operating Conditions (unless otherwise stated)					
PIC18F24/25K42								
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max.	Units	Conditions	
							VDD	Note
D100	IDD_{XT4}	XT = 4 MHz	—	560	1000	µA	3.0V	
D100	IDD_{XT4}	XT = 4 MHz	—	730	1100	µA	3.0V	
D100A	IDD_{XT4}	XT = 4 MHz	—	370	—	µA	3.0V	PMD's all 1's
D100A	IDD_{XT4}	XT = 4 MHz	—	590	—	µA	3.0V	PMD's all 1's
D101	IDD_{HFO16}	HFINTOSC = 16 MHz	—	2.2	3.7	mA	3.0V	
D101	IDD_{HFO16}	HFINTOSC = 16 MHz	—	2.3	3.8	mA	3.0V	
D101A	IDD_{HFO16}	HFINTOSC = 16 MHz	—	1.4	—	mA	3.0V	PMD's all 1's
D101A	IDD_{HFO16}	HFINTOSC = 16 MHz	—	1.4	—	mA	3.0V	PMD's all 1's
D102	IDD_{HFOPLL}	HFINTOSC = 64 MHz	—	8.0	11.1	mA	3.0V	
D102	IDD_{HFOPLL}	HFINTOSC = 64 MHz	—	8.0	11.2	mA	3.0V	
D102A	IDD_{HFOPLL}	HFINTOSC = 64 MHz	—	4.8	—	mA	3.0V	PMD's all 1's
D102A	IDD_{HFOPLL}	HFINTOSC = 64 MHz	—	4.8	—	mA	3.0V	PMD's all 1's
D103	$IDD_{HSPLL64}$	HS+PLL = 64 MHz	—	8.0	11.1	mA	3.0V	
D103	$IDD_{HSPLL64}$	HS+PLL = 64 MHz	—	8.0	11.2	mA	3.0V	
D103A	$IDD_{HSPLL64}$	HS+PLL = 64 MHz	—	4.8	—	mA	3.0V	PMD's all 1's
D103A	$IDD_{HSPLL64}$	HS+PLL = 64 MHz	—	4.8	—	mA	3.0V	PMD's all 1's
D104	IDD_{IDLE}	IDLE mode, HFINTOSC = 16 MHz	—	1.8	—	mA	3.0V	
D104	IDD_{IDLE}	IDLE mode, HFINTOSC = 16 MHz	—	1.8	—	mA	3.0V	
D105	IDD_{DOZE}[3]	DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16	—	1.8	—	mA	3.0V	
D105	IDD_{DOZE}[3]	DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16	—	1.9	—	mA	3.0V	

† 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: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins are outputs driven low; $\overline{MCLR}$ = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

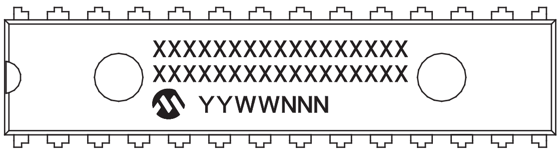
3: $IDD_{DOZE} = [IDD_{IDLE}*(N-1)/N] + IDD_{HFO}16/N$ where N = DOZE Ratio (Register 12-2).

4: PMD bits are all in the default state, no modules are disabled.

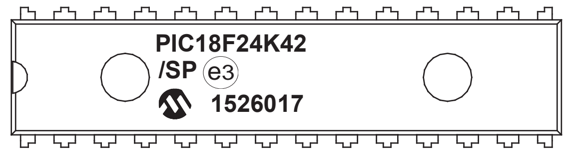
48.0 PACKAGING INFORMATION

Package Marking Information

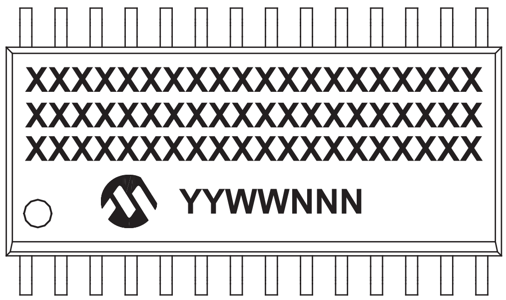
28-Lead SPDIP (.300”)



Example

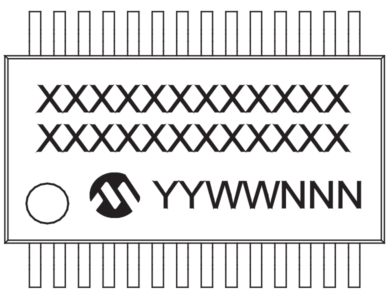


28-Lead SOIC (7.50 mm)

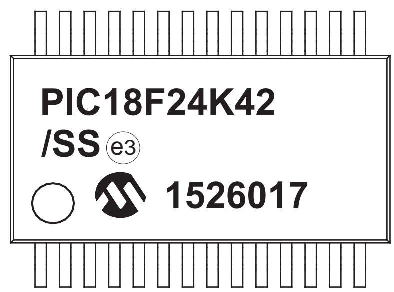


Example

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.

APPENDIX A: REVISION HISTORY

Revision A (11/2016)

Initial release of this document.

Revision B (2/2017)

Updated all Summary of Associated Registers; added Register Summary chapter; Removed Appendix B; Other minor corrections.