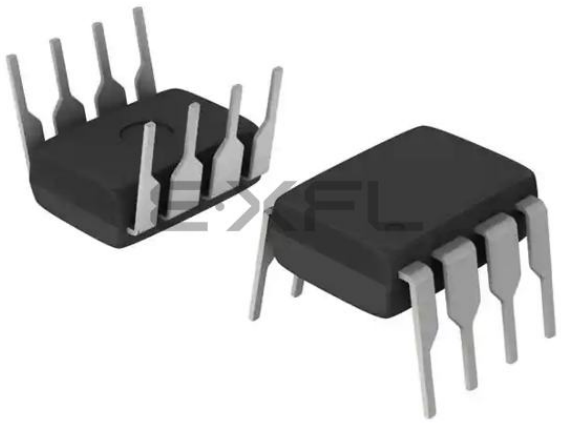


Welcome to E-XFL.COM

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	32MHz
Connectivity	LINbus, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	6
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 4x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12lf1572-e-p

Analog Peripherals:

- 10-Bit Analog-to-Digital Converter (ADC):
  - Up to four external channels
  - Conversion available during Sleep
- Comparator:
  - Low-Power/High-Speed modes
  - Fixed Voltage Reference at (non)inverting input(s)
  - Comparator outputs externally accessible
  - Synchronization with Timer1 clock source
  - Software hysteresis enable
- 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

Clocking Structure:

- Precision Internal Oscillator:
  - Factory calibrated ±1%, typical
  - Software-selectable clock speeds from 31 kHz to 32 MHz
- External Oscillator Block with:
  - Resonator modes up to 20 MHz
  - Two External Clock modes up to 32 MHz
- Fail-Safe Clock Monitor
- Digital Oscillator Input Available

PIC12(L)F1571/2 FAMILY TYPES

Device	Data Sheet Index	Program Memory Flash (K words)	Data SRAM (bytes)	High-Endurance Flash (bytes)	I/O Pins	8-Bit/16-Bit Timers	Comparators	16-Bit PWM	10-Bit ADC (ch)	5-Bit DAC	CWG	EUSART	Debug(1)	XLP
PIC12(L)F1571	**A**	1	128	128	6	2/4(2)	1	3	4	1	1	0	I	Y
PIC12(L)F1572	**A**	2	256	128	6	2/4(2)	1	3	4	1	1	1	I	Y

Note 1: I – Debugging integrated on chip.

2: Three additional 16-bit timers available when not using the 16-bit PWM outputs.

Data Sheet Index: (Unshaded devices are described in this document.)

A DS40001723 PIC12(L)F1571/2 Data Sheet, 8-Pin Flash, 8-Bit MCU with High-Precision 16-Bit PWM.

1.1.2.2 Long Bit Names

Long bit names are constructed by adding a peripheral abbreviation prefix to the short name. The prefix is unique to the peripheral, thereby making every long bit name unique. The long bit name for the COG1 enable bit is the COG1 prefix, G1, appended with the enable bit short name, EN, resulting in the unique bit name G1EN.

Long bit names are useful in both C and assembly programs. For example, in C, the COG1CON0 enable bit can be set with the `G1EN = 1` instruction. In assembly, this bit can be set with the `BSF COG1CON0,G1EN` instruction.

1.1.2.3 Bit Fields

Bit fields are two or more adjacent bits in the same register. Bit fields adhere only to the short bit naming convention. For example, the three Least Significant bits of the COG1CON0 register contain the mode control bits. The short name for this field is MD. There is no long bit name variant. Bit field access is only possible in C programs. The following example demonstrates a C program instruction for setting the COG1 to the Push-Pull mode:

```
COG1CON0bits.MD = 0x5;
```

Individual bits in a bit field can also be accessed with long and short bit names. Each bit is the field name appended with the number of the bit position within the field. For example, the Most Significant mode bit has the short bit name, MD2, and the long bit name is G1MD2. The following two examples demonstrate assembly program sequences for setting the COG1 to Push-Pull mode:

Example 1:

```
MOVLW  ~(1<<G1MD1)
ANDWF  COG1CON0,F
MOVLW  1<<G1MD2 | 1<<G1MD0
IORWF  COG1CON0,F
```

Example 2:

```
BSF    COG1CON0,G1MD2
BCF    COG1CON0,G1MD1
BSF    COG1CON0,G1MD0
```

1.1.3 REGISTER AND BIT NAMING EXCEPTIONS

1.1.3.1 Status, Interrupt and Mirror Bits

Status, interrupt enables, interrupt flags and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

1.1.3.2 Legacy Peripherals

There are some peripherals that do not strictly adhere to these naming conventions. Peripherals that have existed for many years and are present in almost every device are the exceptions. These exceptions were necessary to limit the adverse impact of the new conventions on legacy code. Peripherals that do adhere to the new convention will include a table in the registers section indicating the long name prefix for each peripheral instance. Peripherals that fall into the exception category will not have this table. These peripherals include, but are not limited to, the following:

- EUSART
- MSSP

TABLE 3-5: PIC12(L)F1571/2 MEMORY MAP, BANK 8-23

BANK 8	BANK 9	BANK 10	BANK 11	BANK 12	BANK 13	BANK 14	BANK 15
400h Core Registers (Table 3-2) 40Bh	480h Core Registers (Table 3-2) 48Bh	500h Core Registers (Table 3-2) 50Bh	580h Core Registers (Table 3-2) 58Bh	600h Core Registers (Table 3-2) 60Bh	680h Core Registers (Table 3-2) 68Bh	700h Core Registers (Table 3-2) 70Bh	780h Core Registers (Table 3-2) 78Bh
40Ch —	48Ch —	50Ch —	58Ch —	60Ch —	68Ch —	70Ch —	78Ch —
40Dh —	48Dh —	50Dh —	58Dh —	60Dh —	68Dh —	70Dh —	78Dh —
40Eh —	48Eh —	50Eh —	58Eh —	60Eh —	68Eh —	70Eh —	78Eh —
40Fh —	48Fh —	50Fh —	58Fh —	60Fh —	68Fh —	70Fh —	78Fh —
410h —	490h —	510h —	590h —	610h —	690h —	710h —	790h —
411h —	491h —	511h —	591h —	611h —	691h CWG1DBR	711h —	791h —
412h —	492h —	512h —	592h —	612h —	692h CWG1DBF	712h —	792h —
413h —	493h —	513h —	593h —	613h —	693h CWG1CON0	713h —	793h —
414h —	494h —	514h —	594h —	614h —	694h CWG1CON1	714h —	794h —
415h —	495h —	515h —	595h —	615h —	695h CWG1CON2	715h —	795h —
416h —	496h —	516h —	596h —	616h —	696h —	716h —	796h —
417h —	497h —	517h —	597h —	617h —	697h —	717h —	797h —
418h —	498h —	518h —	598h —	618h —	698h —	718h —	798h —
419h —	499h —	519h —	599h —	619h —	699h —	719h —	799h —
41Ah —	49Ah —	51Ah —	59Ah —	61Ah —	69Ah —	71Ah —	79Ah —
41Bh —	49Bh —	51Bh —	59Bh —	61Bh —	69Bh —	71Bh —	79Bh —
41Ch —	49Ch —	51Ch —	59Ch —	61Ch —	69Ch —	71Ch —	79Ch —
41Dh —	49Dh —	51Dh —	59Dh —	61Dh —	69Dh —	71Dh —	79Dh —
41Eh —	49Eh —	51Eh —	59Eh —	61Eh —	69Eh —	71Eh —	79Eh —
41Fh —	49Fh —	51Fh —	59Fh —	61Fh —	69Fh —	71Fh —	79Fh —
420h Unimplemented Read as '0' 46Fh	4A0h Unimplemented Read as '0' 4EFh	520h Unimplemented Read as '0' 56Fh	5A0h Unimplemented Read as '0' 5EFh	620h Unimplemented Read as '0' 66Fh	6A0h Unimplemented Read as '0' 6EFh	720h Unimplemented Read as '0' 76Fh	7A0h Unimplemented Read as '0' 7EFh
470h Accesses 70h-7Fh 47Fh	4F0h Accesses 70h-7Fh 4FFh	570h Accesses 70h-7Fh 57Fh	5F0h Accesses 70h-7Fh 5FFh	670h Accesses 70h-7Fh 67Fh	6F0h Accesses 70h-7Fh 6FFh	770h Accesses 70h-7Fh 77Fh	7F0h Accesses 70h-7Fh 7FFh

BANK 16	BANK 17	BANK 18	BANK 19	BANK 20	BANK 21	BANK 22	BANK 23
800h Core Registers (Table 3-2) 80Bh	880h Core Registers (Table 3-2) 88Bh	900h Core Registers (Table 3-2) 90Bh	980h Core Registers (Table 3-2) 98Bh	A00h Core Registers (Table 3-2) A0Bh	A80h Core Registers (Table 3-2) A8Bh	B00h Core Registers (Table 3-2) B0Bh	B80h Core Registers (Table 3-2) B8Bh
80Ch Unimplemented Read as '0' 86Fh	88Ch Unimplemented Read as '0' 8EFh	90Ch Unimplemented Read as '0' 96Fh	98Ch Unimplemented Read as '0' 9EFh	A0Ch Unimplemented Read as '0' A6Fh	A8Ch Unimplemented Read as '0' AEFh	B0Ch Unimplemented Read as '0' B6Fh	B8Ch Unimplemented Read as '0' BEFh
870h Accesses 70h-7Fh 87Fh	8F0h Accesses 70h-7Fh 8FFh	970h Accesses 70h-7Fh 97Fh	9F0h Accesses 70h-7Fh 9FFh	A70h Accesses 70h-7Fh A7Fh	AF0h Accesses 70h-7Fh AFFh	B70h Accesses 70h-7Fh B7Fh	BF0h Accesses 70h-7Fh BFFh

Legend: ☐ = Unimplemented data memory locations, read as '0'.

4.7 Register Definitions: Device ID

REGISTER 4-3: DEVICEID: DEVICE ID REGISTER(1)

R	R	R	R	R	R
DEV<13:8>					
bit 13					bit 8

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7							bit 0

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

bit 13-0 **DEV<13:0>:** Device ID bits
Refer to Table 4-1 to determine what these bits will read on which device. A value of 3FFFh is invalid.

Note 1: This location cannot be written.

REGISTER 4-4: REVISIONID: REVISION ID REGISTER(1)

R	R	R	R	R	R
REV<13:8>					
bit 13					bit 8

R	R	R	R	R	R	R	R
REV<7:0>							
bit 7							bit 0

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

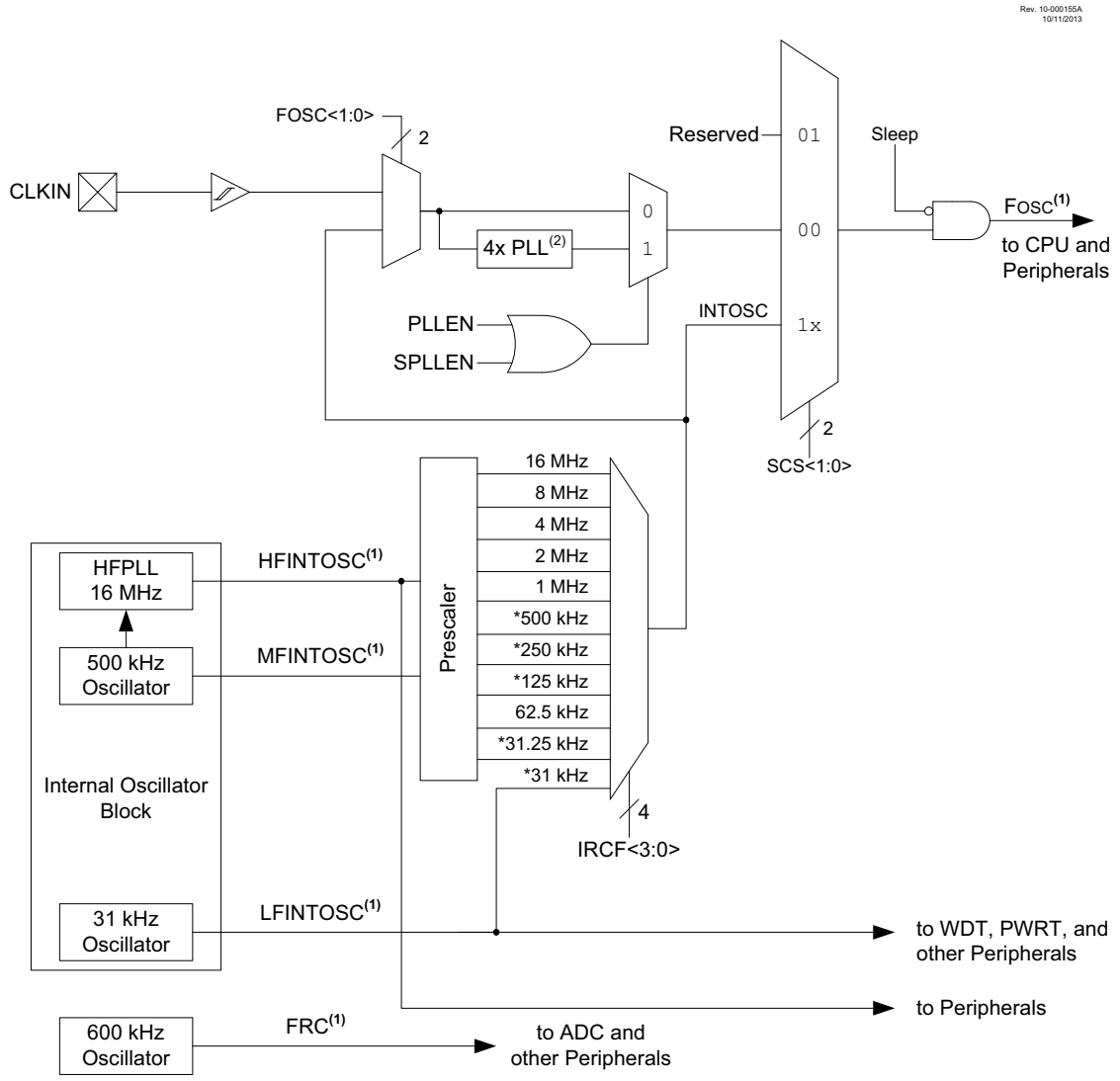
bit 13-0 **REV<13:0>:** Revision ID bits
These bits are used to identify the device revision.

Note 1: This location cannot be written.

TABLE 4-1: DEVICE ID VALUES

DEVICE	Device ID	Revision ID
PIC12F1571	3051h	2xxxh
PIC12LF1571	3053h	2xxxh
PIC12F1572	3050h	2xxxh
PIC12LF1572	3052h	2xxxh

FIGURE 5-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



* Available with more than one IRCF selection

Note 1: See **Section 5.2 "Clock Source Types"**.

2: ST Buffer is high-speed type when using T1CKI.

3: If FOSC<1:0> = 00, 4x PLL can only be used if IRCF<3:0> = 1110.

5.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in the Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to the internal oscillator during run time. See **Section 5.3 "Clock Switching"**for more information.

In **INTOSC** mode, CLKIN is available for general purpose I/O. CLKOUT is available for general purpose I/O or CLKOUT.

The function of the OSC2/CLKOUT pin is determined by the CLKOUTEN bit in the Configuration Words.

The internal oscillator block has two independent oscillators and a dedicated Phase-Locked Loop, HFPLL, that can produce one of three internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The HFINTOSC source is generated from the 500 kHz MFINTOSC source and the dedicated Phase-Locked Loop, HFPLL. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 5-3).
2. The **MFINTOSC** (Medium Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 5-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

5.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of multiple frequencies derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.8 "Internal Oscillator Clock Switch Timing"** for more information.

The HFINTOSC is enabled by:

- Configuring the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- Setting FOSC<1:0> = 00, or
- Setting the System Clock Source x (SCSx) bits of the OSCCON register to '1x'.

A fast start-up oscillator allows internal circuits to power up and stabilize before switching to HFINTOSC.

The High-Frequency Internal Oscillator Ready bit (HFIOFR) of the OSCSTAT register indicates when the HFINTOSC is running.

The High-Frequency Internal Oscillator Status Locked bit (HFIOFL) of the OSCSTAT register indicates when the HFINTOSC is running within 2% of its final value.

The High-Frequency Internal Oscillator Stable bit (HFIOFS) of the OSCSTAT register indicates when the HFINTOSC is running within 0.5% of its final value.

5.2.2.2 MFINTOSC

The Medium Frequency Internal Oscillator (MFINTOSC) is a factory calibrated 500 kHz internal clock source. The frequency of the MFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the MFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of nine frequencies derived from the MFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.8 "Internal Oscillator Clock Switch Timing"** for more information.

The MFINTOSC is enabled by:

- Configuring the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- Setting FOSC<1:0> = 00, or
- Setting the System Clock Source x (SCSx) bits of the OSCCON register to '1x'

The Medium Frequency Internal Oscillator Ready bit (MFIOFR) of the OSCSTAT register indicates when the MFINTOSC is running.

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the interrupt enable bit of the interrupt event is contained in the PIE1, PIE2 and PIE3 registers)

The INTCON, PIR1, PIR2 and PIR3 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section 7.5 "Automatic Context Saving"."**)
- PC is loaded with the interrupt vector, 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

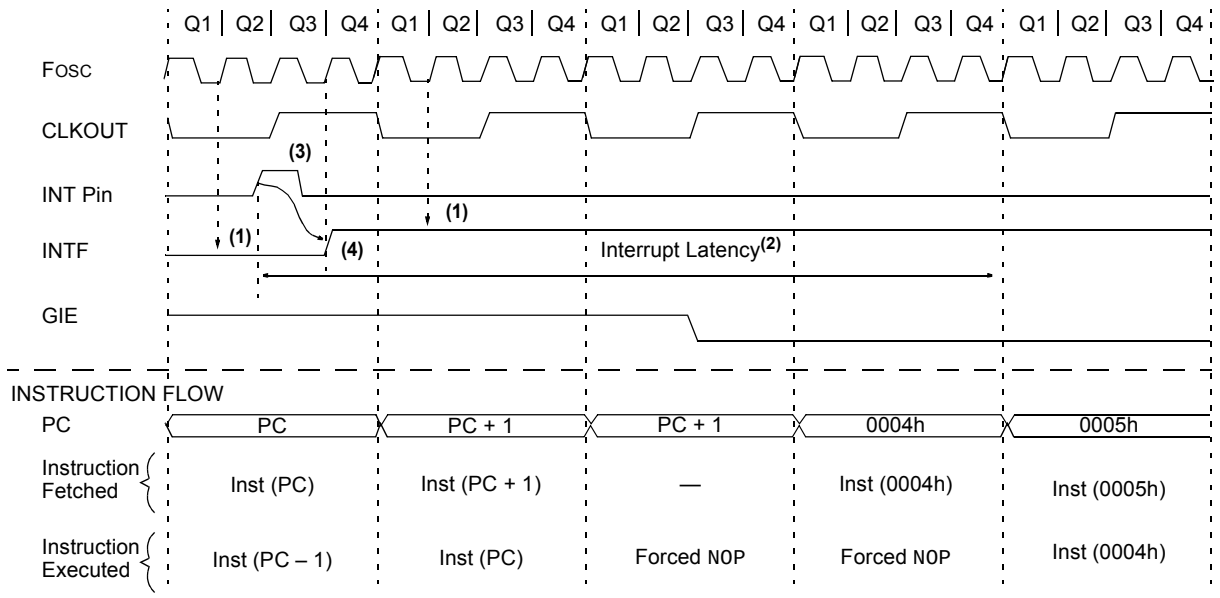
For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

FIGURE 7-3: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: For minimum width of INT pulse, refer to AC specifications in **Section 26.0 "Electrical Specifications"**.

4: INTF is enabled to be set any time during the Q4-Q1 cycles.

REGISTER 7-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

U-0	U-0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
—	—	C1IE	—	—	—	—	—
bit 7							bit 0

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

bit 7-6 **Unimplemented:** Read as '0'

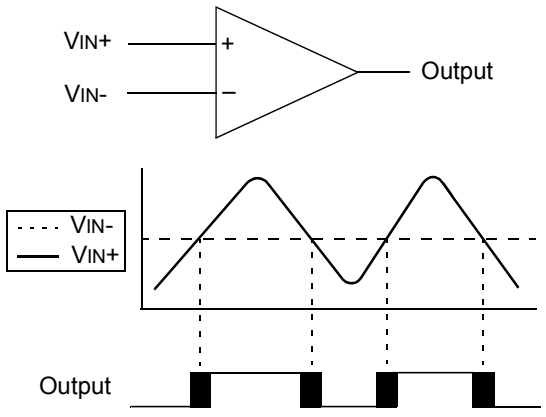
bit 5 **C1IE:** Comparator C1 Interrupt Enable bit
1 = Enables the Comparator C1 interrupt
0 = Disables the Comparator C1 interrupt

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

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

NOTES:

FIGURE 17-2: SINGLE COMPARATOR



Note: The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

17.2 Comparator Control

The comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 17-1) contains control and status bits for the following:

- Enable
- Output selection
- Output polarity
- Speed/power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 register (see Register 17-2) contains control bits for the following:

- Interrupt enable
- Interrupt edge polarity
- Positive input channel selection
- Negative input channel selection

17.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

17.2.2 COMPARATOR POSITIVE INPUT SELECTION

Configuring the CxPCH<1:0> bits of the CMxCON1 register directs an internal voltage reference or an analog pin to the non-inverting input of the comparator:

- CxIN+ analog pin
- DAC1_output
- FVR_buffer2
- VSS

See **Section 13.0 "Fixed Voltage Reference (FVR)"** for more information on the Fixed Voltage Reference module.

See **Section 16.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** for more information on the DAC input signal.

Any time the comparator is disabled (CxON = 0), all comparator inputs are disabled.

17.2.3 COMPARATOR NEGATIVE INPUT SELECTION

The CxNCH<2:0> bits of the CMxCON0 register direct one of the input sources to the comparator inverting input.

Note: To use CxIN+ and CxIN- pins as analog input, the appropriate bits must be set in the ANSELx register and the corresponding TRISx bits must also be set to disable the output drivers.

17.2.4 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- CxOE bit of the CMxCON0 register must be set
- Corresponding TRISx bit must be cleared
- CxON bit of the CMxCON0 register must be set

The synchronous comparator output signal (CxOUT_sync) is available to the following peripheral(s):

- Analog-to-Digital Converter (ADC)
- Timer1

The asynchronous comparator output signal (CxOUT_async) is available to the following peripheral(s):

- Complementary Waveform Generator (CWG)

Note 1: The CxOE bit of the CMxCON0 register overrides the port data latch. Setting the CxON bit of the CMxCON0 register has no impact on the port override.

2: The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

21.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the EUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit.

21.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read, but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by resetting the EUSART by clearing the SPEN bit of the RCSTA register.

21.1.2.6 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set, the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

21.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

REGISTER 21-3: BAUDCON: BAUD RATE CONTROL REGISTER

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

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

bit 7 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care.

bit 6 **RCIDL**: Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is Idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don't care.

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

bit 4 **SCKP**: Synchronous Clock Polarity Select bit
Asynchronous mode:
1 = Transmits inverted data to the TX/CK pin
0 = Transmits non-inverted data to the TX/CK pin
Synchronous mode:
1 = Data is clocked on rising edge of the clock
0 = Data is clocked on falling edge of the clock

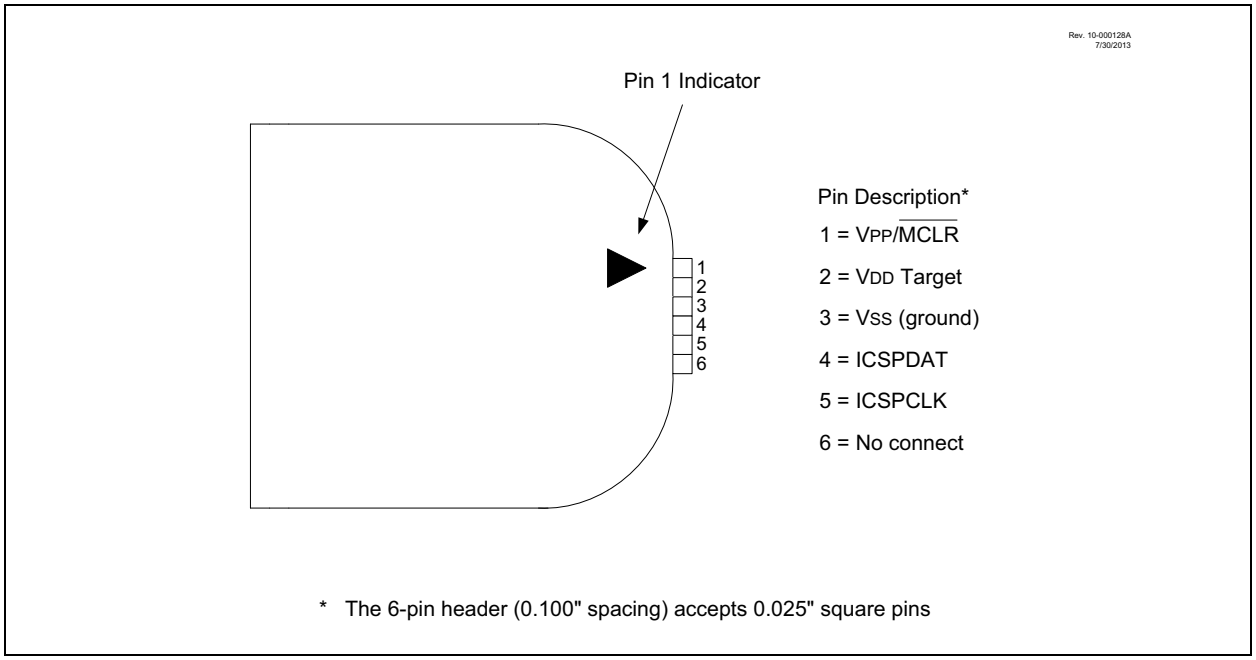
bit 3 **BRG16:** 16-Bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

bit 2 **Unimplemented:** Read as '0'

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = Receiver is waiting for a falling edge; no character will be received, RCIF bit will be set, WUE will automatically clear after RCIF is set
0 = Receiver is operating normally
Synchronous mode:
Don't care

bit 0 **ABDEN**: A.uto-Baud Detect Enable bit
Asynchronous mode:
1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud Detect mode is disabled
Synchronous mode:
Don't care.

FIGURE 24-2: PICkit™ PROGRAMMER STYLE CONNECTOR INTERFACE



For additional interface recommendations, refer to your specific device programmer manual prior to PCB design.

It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices, such as resistors, diodes or even jumpers. See Figure 24-3 for more information.

FIGURE 24-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING

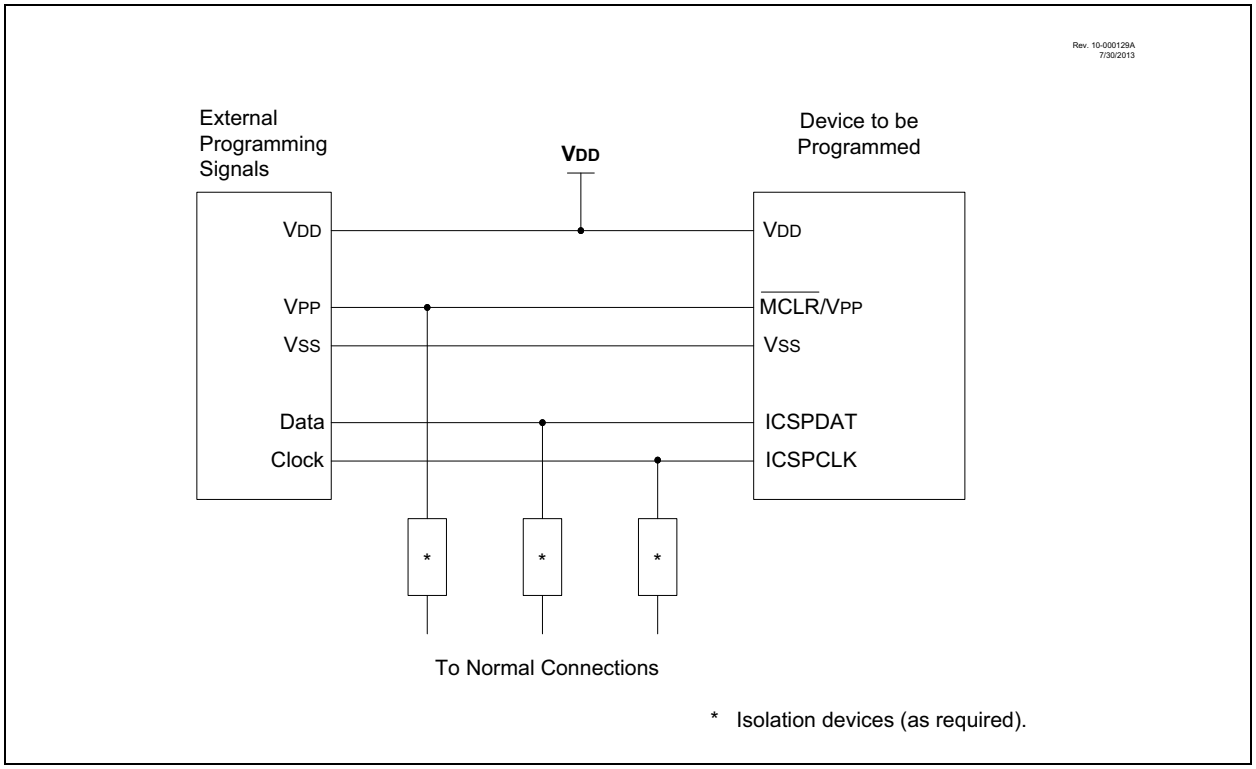


TABLE 25-3: ENHANCED MID-RANGE INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb	LSb	Status Affected	Notes
CONTROL OPERATIONS							
BRA	k	Relative Branch	2	11 001k kkkk kkkk			
BRW	–	Relative Branch with W	2	00 0000 0000 1011			
CALL	k	Call Subroutine	2	10 0kkk kkkk kkkk			
CALLW	–	Call Subroutine with W	2	00 0000 0000 1010			
GOTO	k	Go to address	2	10 1kkk kkkk kkkk			
RETFIE	k	Return from interrupt	2	00 0000 0000 1001			
RETLW	k	Return with literal in W	2	11 0100 kkkk kkkk			
RETURN	–	Return from Subroutine	2	00 0000 0000 1000			
INHERENT OPERATIONS							
CLRWDT	–	Clear Watchdog Timer	1	00 0000 0110 0100		$\overline{TO}$, $\overline{PD}$	
NOP	–	No Operation	1	00 0000 0000 0000			
OPTION	–	Load OPTION_REG register with W	1	00 0000 0110 0010			
RESET	–	Software device Reset	1	00 0000 0000 0001			
SLEEP	–	Go into Standby mode	1	00 0000 0110 0011		$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register with W	1	00 0000 0110 0fff			
C COMPILER OPTIMIZED							
ADDFSR	n, k	Add Literal k to FSRn	1	11 0001 0nkk kkkk			
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00 0000 0001 0nmm	kkkk	Z	**2, 3**
	k[n]	Move INDFn to W, Indexed Indirect.	1	11 1111 0nkk 1nmm		Z	**2**
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00 0000 0001 kkkk			**2, 3**
	k[n]	Move W to INDFn, Indexed Indirect.	1	11 1111 1nkk			**2**

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

3: See the table in the `MOVIW` and `MOVWI` instruction descriptions.

TABLE 26-2: SUPPLY CURRENT (I_{DD})(1,2)

PIC12LF1571/2		Standard Operating Conditions (unless otherwise stated)					
PIC12F1571/2							
Param. No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						V_{DD}	Note
D013		—	35	44	μA	1.8	F_{OSC} = 1 MHz, External Clock (ECM), Medium Power mode
		—	60	69	μA	3.0	
D013		—	68	93	μA	2.3	F_{OSC} = 1 MHz, External Clock (ECM), Medium Power mode
		—	91	120	μA	3.0	
		—	131	160	μA	5.0	
D014		—	116	132	μA	1.8	F_{OSC} = 4 MHz, External Clock (ECM), Medium Power mode
		—	203	233	μA	3.0	
D014		—	174	221	μA	2.3	F_{OSC} = 4 MHz, External Clock (ECM), Medium Power mode
		—	234	286	μA	3.0	
		—	299	374	μA	5.0	
D015		—	5.5	11	μA	1.8	F_{OSC} = 31 kHz, LFINTOSC, -40°C ≤ T_A ≤ +85°C
		—	7.3	12	μA	3.0	
D015		—	13	21	μA	2.3	F_{OSC} = 31 kHz, LFINTOSC, -40°C ≤ T_A ≤ +85°C
		—	15	24	μA	3.0	
		—	17	25	μA	5.0	
D016		—	111	151	μA	1.8	F_{OSC} = 500 kHz, MFINTOSC
		—	133	176	μA	3.0	
D016		—	144	209	μA	2.3	F_{OSC} = 500 kHz, MFINTOSC
		—	162	237	μA	3.0	
		—	216	288	μA	5.0	
D017*		—	0.5	0.6	mA	1.8	F_{OSC} = 8 MHz, HFINTOSC
		—	0.7	0.9	mA	3.0	
D017*		—	0.6	0.8	mA	2.3	F_{OSC} = 8 MHz, HFINTOSC
		—	0.8	0.9	mA	3.0	
		—	0.9	1.0	mA	5.0	
D018		—	0.7	0.8	mA	1.8	F_{OSC} = 16 MHz, HFINTOSC
		—	1.1	1.2	mA	3.0	
D018		—	0.9	1.1	mA	2.3	F_{OSC} = 16 MHz, HFINTOSC
		—	1.1	1.3	mA	3.0	
		—	1.3	1.5	mA	5.0	

* These parameters are characterized but not tested.

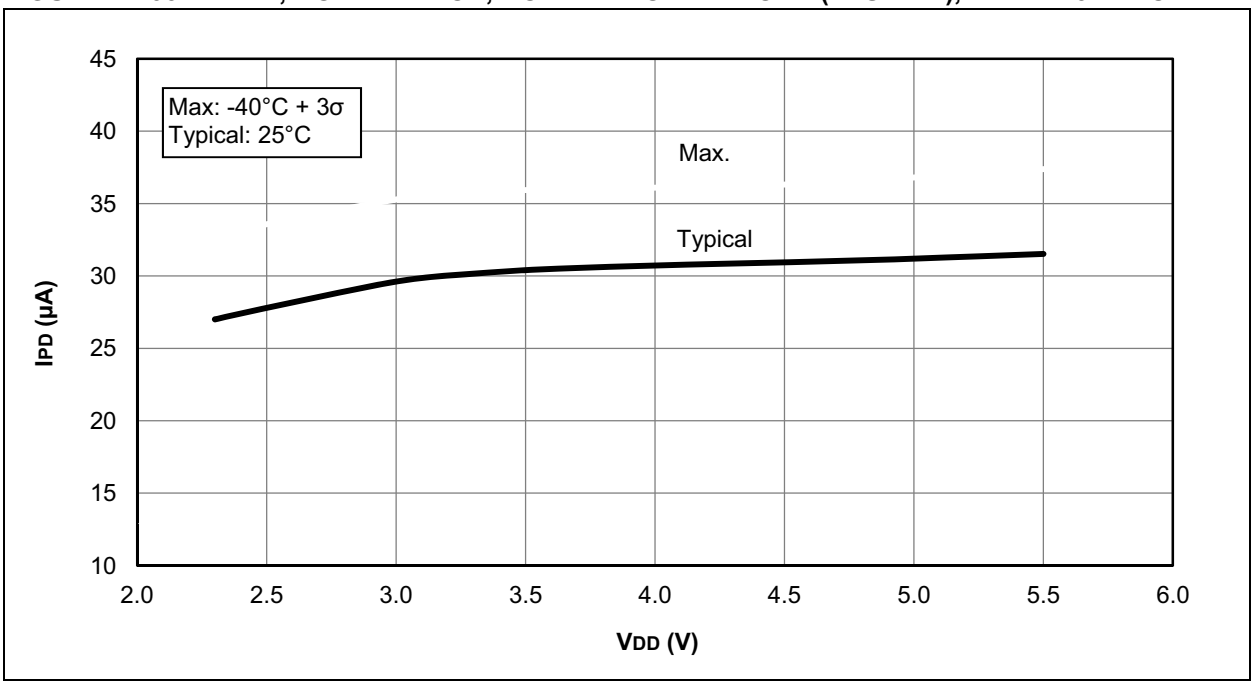
† Data in "Typ" column is at 3.0V, +25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all I_{DD} measurements in active operation mode are: CLKIN = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{SS}; $\overline{MCLR}$ = V_{DD}; 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: PLL required for 32 MHz operation.

FIGURE 27-35: I_{PD}, COMPARATOR, NORMAL POWER MODE (CxSP = 1), PIC12F1571/2 ONLY



28.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

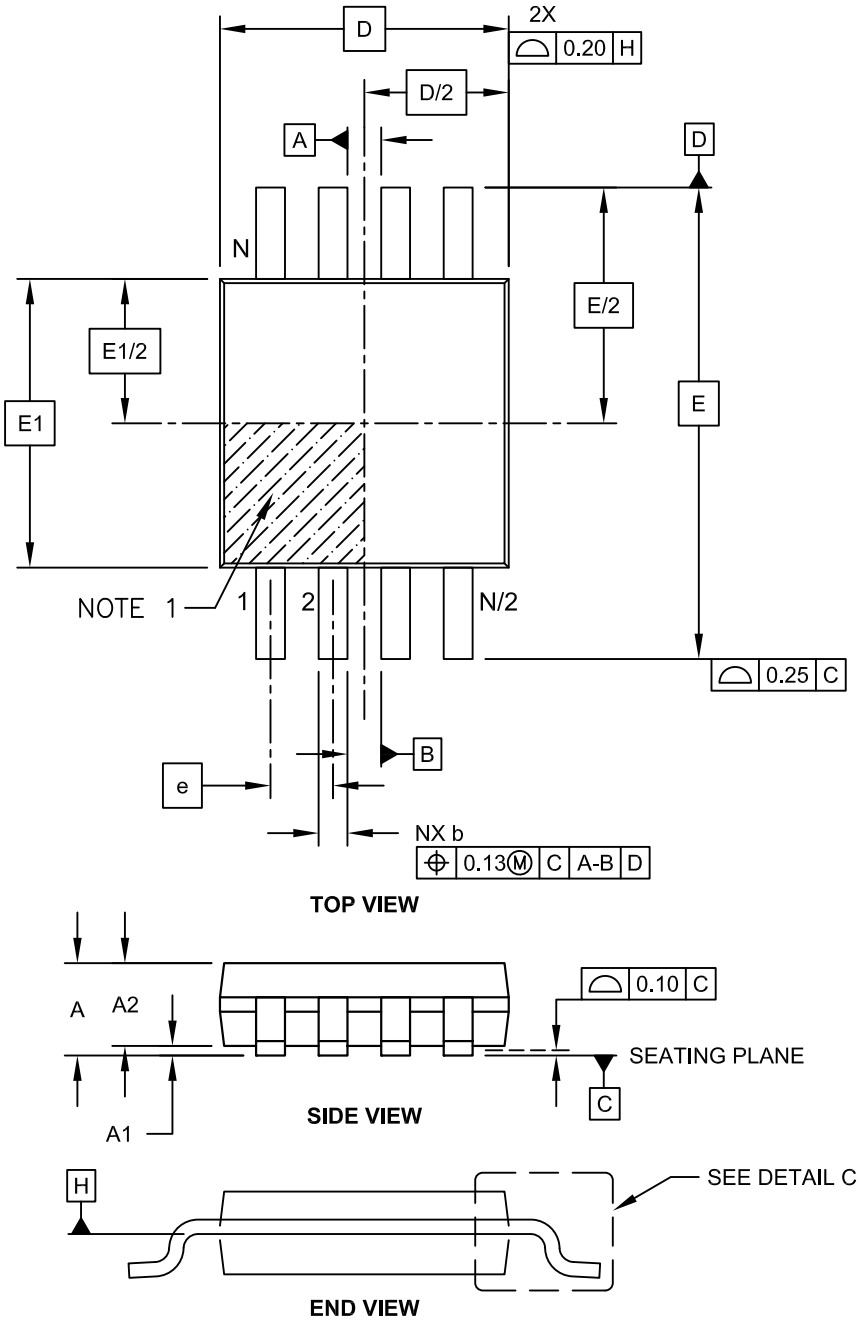
28.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

8-Lead Plastic Micro Small Outline Package (MS) [MSOP]

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



Microchip Technology Drawing C04-111C Sheet 1 of 2

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

07/14/15