



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

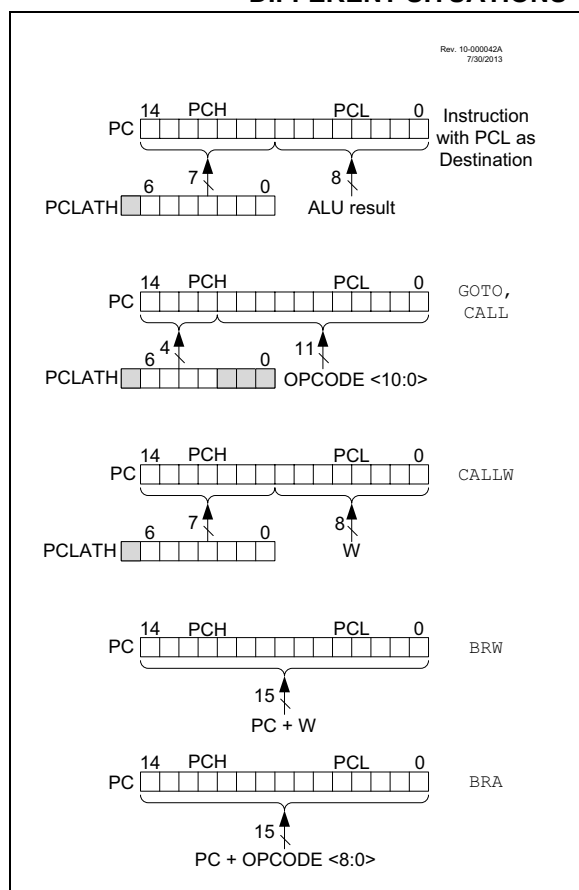
Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	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)	2.3V ~ 5.5V
Data Converters	A/D 4x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-TSSOP, 8-MSOP (0.118", 3.00mm Width)
Supplier Device Package	8-MSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic12f1572-i-ms

3.4 PCL and PCLATH

The Program Counter (PC) is 15 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<14:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 3-4 shows the five situations for the loading of the PC.

FIGURE 3-4: LOADING OF PC IN DIFFERENT SITUATIONS



3.4.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the Program Counter to be changed by writing the desired upper seven bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 15 bits of the Program Counter will change to the values contained in the PCLATH register and those being written to the PCL register.

3.4.2 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the Program Counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to Application Note AN556, "Implementing a Table Read" (DS00556).

3.4.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provides another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<6:3> is loaded with PCLATH<6:3>.

The CALLW instruction enables computed CALLs by combining PCLATH and W to form the destination address. A computed CALLW is accomplished by loading the W register with the desired address and executing CALLW. The PCL register is loaded with the value of W and PCH is loaded with PCLATH.

3.4.4 BRANCHING

The branching instructions add an offset to the PC. This allows relocatable code and code that crosses page boundaries. There are two forms of branching, BRW and BRA. The PC will have incremented to fetch the next instruction in both cases. When using either branching instruction, a PCL memory boundary may be crossed.

If using BRW, load the W register with the desired unsigned address and execute BRW. The entire PC will be loaded with the address, PC + 1 + W.

If using BRA, the entire PC will be loaded with PC + 1 +, the signed value of the operand of the BRA instruction.

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

PIC12(L)F1571/2

NOTES:

5.2 Clock Source Types

Clock sources can be classified as external or internal.

External clock sources rely on external circuitry for the clock source to function.

Internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators and a dedicated Phase-Locked Loop (HFPLL) that are used to generate three internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC), 500 kHz Medium Frequency Internal Oscillator (MFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS<1:0>) bits in the OSCCON register. See **Section 5.3 “Clock Switching”** for additional information.

5.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in the Configuration Words to select an external clock source that 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:
 - Timer1 oscillator during run time, or
 - An external clock source determined by the value of the FOSCx bits.

See **Section 5.3 “Clock Switching”** for more information.

5.2.1.1 EC Mode

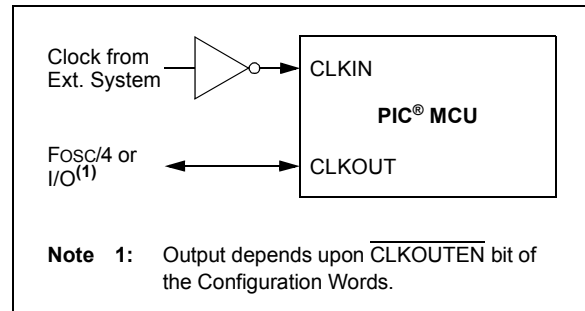
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input. CLKOUT is available for general purpose I/Os or CLKOUT. Figure 5-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through the FOSCx bits in the Configuration Words:

- ECH – High power, 4-20 MHz
- ECM – Medium power, 0.5-4 MHz
- ECL – Low power, 0-0.5 MHz

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC® MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

FIGURE 5-2: EXTERNAL CLOCK (EC) MODE OPERATION



PIC12(L)F1571/2

REGISTER 5-2: OSCSTAT: OSCILLATOR STATUS REGISTER

U-0	R-0/q	R-q/q	R-0/q	R-0/q	R-q/q	R-0/q	R-0/q
—	PLL R	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

q = Conditional 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 **Unimplemented:** Read as '0'

bit 6 **PLL R** 4x PLL Ready bit

1 = 4x PLL is ready

0 = 4x PLL is not ready

bit 5 **OSTS:** Oscillator Start-up Timer Status bit

1 = Running from the clock defined by the FOSC<1:0> bits of the Configuration Words

0 = Running from an internal oscillator (FOSC<1:0> = 00)

bit 4 **HFIOFR:** High-Frequency Internal Oscillator Ready bit

1 = HFINTOSC is ready

0 = HFINTOSC is not ready

bit 3 **HFIOFL:** High-Frequency Internal Oscillator Locked bit

1 = HFINTOSC is at least 2% accurate

0 = HFINTOSC is not 2% accurate

bit 2 **MFIOFR:** Medium Frequency Internal Oscillator Ready bit

1 = MFINTOSC is ready

0 = MFINTOSC is not ready

bit 1 **LFIOFR:** Low-Frequency Internal Oscillator Ready bit

1 = LFINTOSC is ready

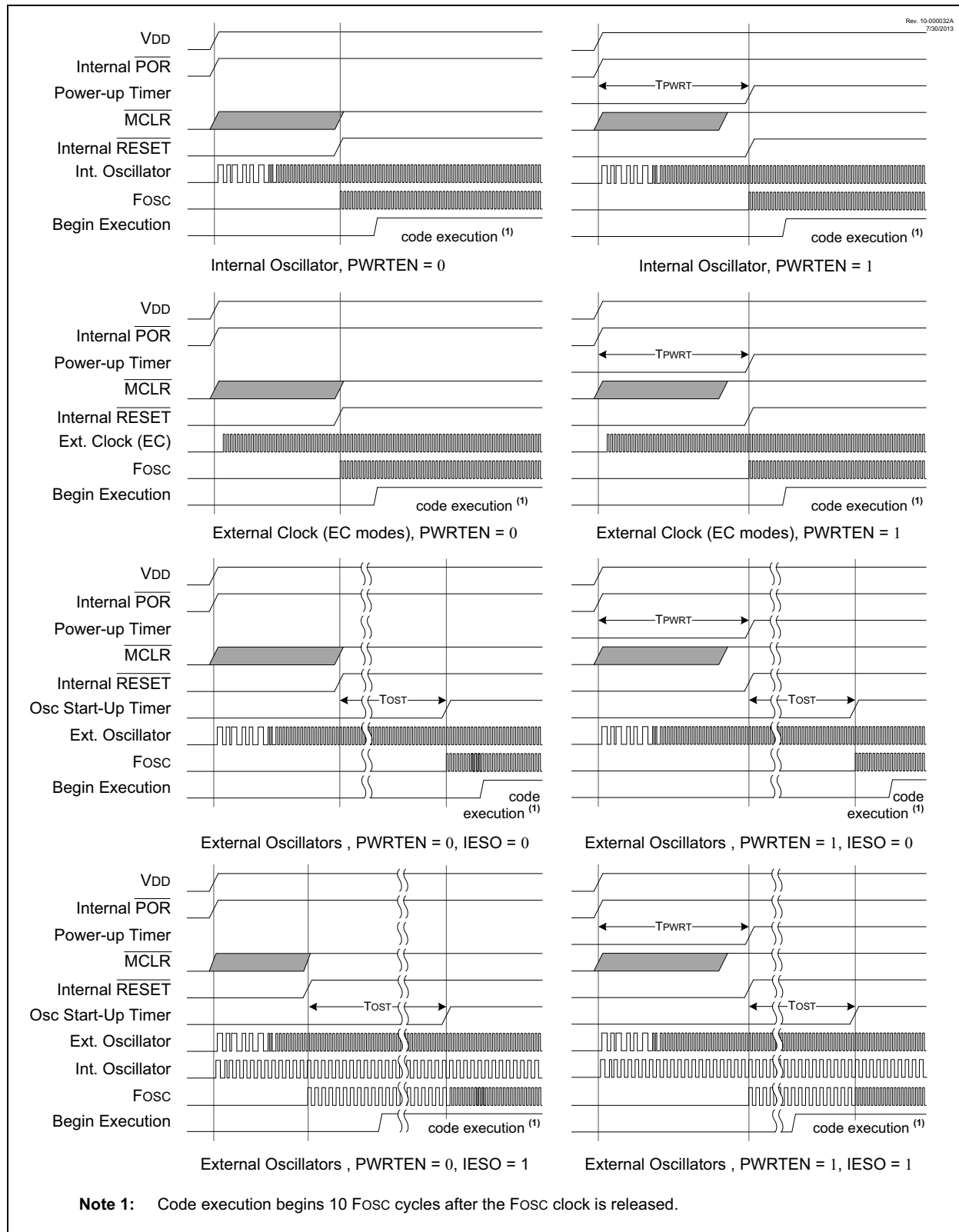
0 = LFINTOSC is not ready

bit 0 **HFIOFS:** High-Frequency Internal Oscillator Stable bit

1 = HFINTOSC is at least 0.5% accurate

0 = HFINTOSC is not 0.5% accurate

FIGURE 6-3: RESET START-UP SEQUENCE



11.0 I/O PORTS

Each port has three standard registers for its operation. These registers are:

- TRISx registers (Data Direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (Output Latch)
- INLVLx (Input Level Control)
- ODCONx registers (Open-Drain Control)
- SLRCONx registers (Slew Rate Control)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (Analog Select)
- WPUx (Weak Pull-up)

In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

TABLE 11-1: PORT AVAILABILITY PER DEVICE

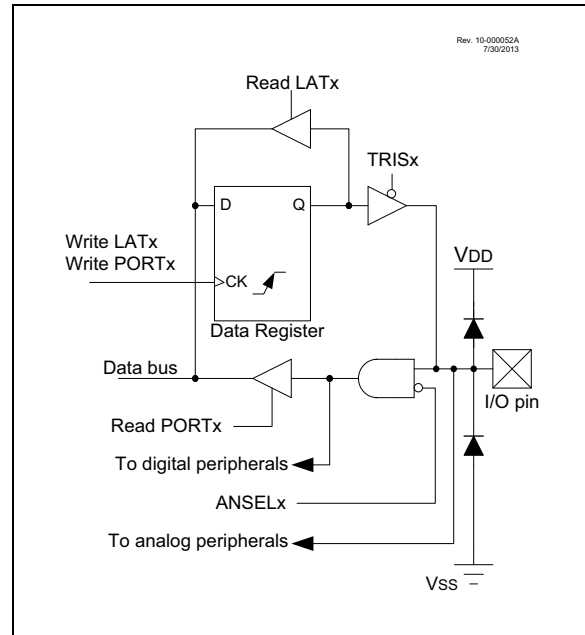
Device	PORTA
PIC12(L)F1571	•
PIC12(L)F1572	•

The Data Latch (LATx registers) is useful for Read-Modify-Write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads the values held in the I/O port latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSELx bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



17.2.5 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 17-2 shows the output state versus input conditions, including polarity control.

TABLE 17-2: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS

Input Condition	CxPOL	CxOUT
$CxVN > CxVP$	0	0
$CxVN < CxVP$	0	1
$CxVN > CxVP$	1	1
$CxVN < CxVP$	1	0

17.2.6 COMPARATOR SPEED/POWER SELECTION

The trade-off between speed or power can be optimized during program execution with the CxSP control bit. The default state for this bit is '1', which selects the Normal Speed mode. Device power consumption can be optimized at the cost of slower comparator propagation delay by clearing the CxSP bit to '0'.

17.3 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 17-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward-biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 17-3: ANALOG INPUT MODEL

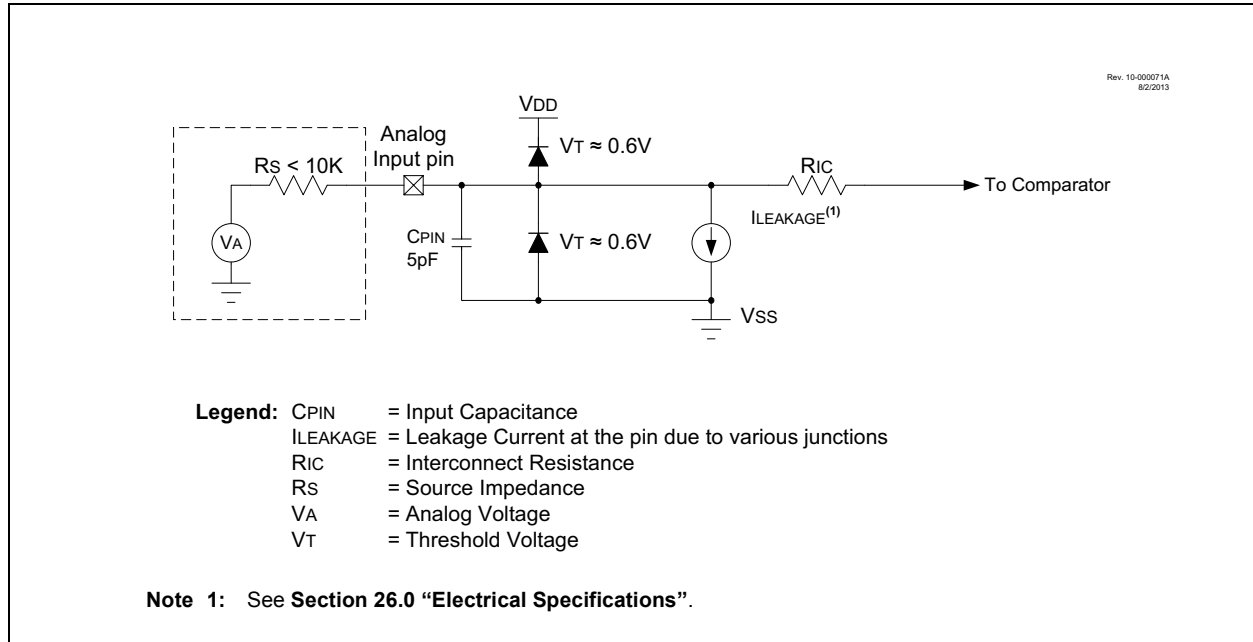


FIGURE 19-3: TIMER1 GATE ENABLE MODE

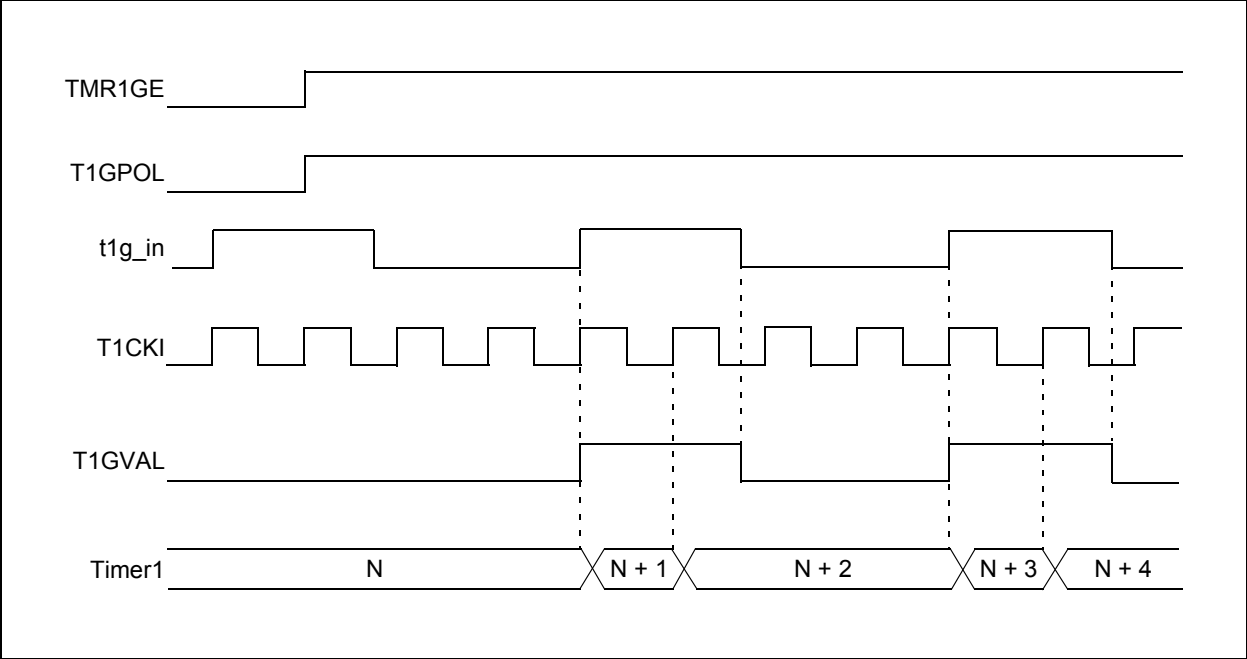


FIGURE 19-4: TIMER1 GATE TOGGLE MODE

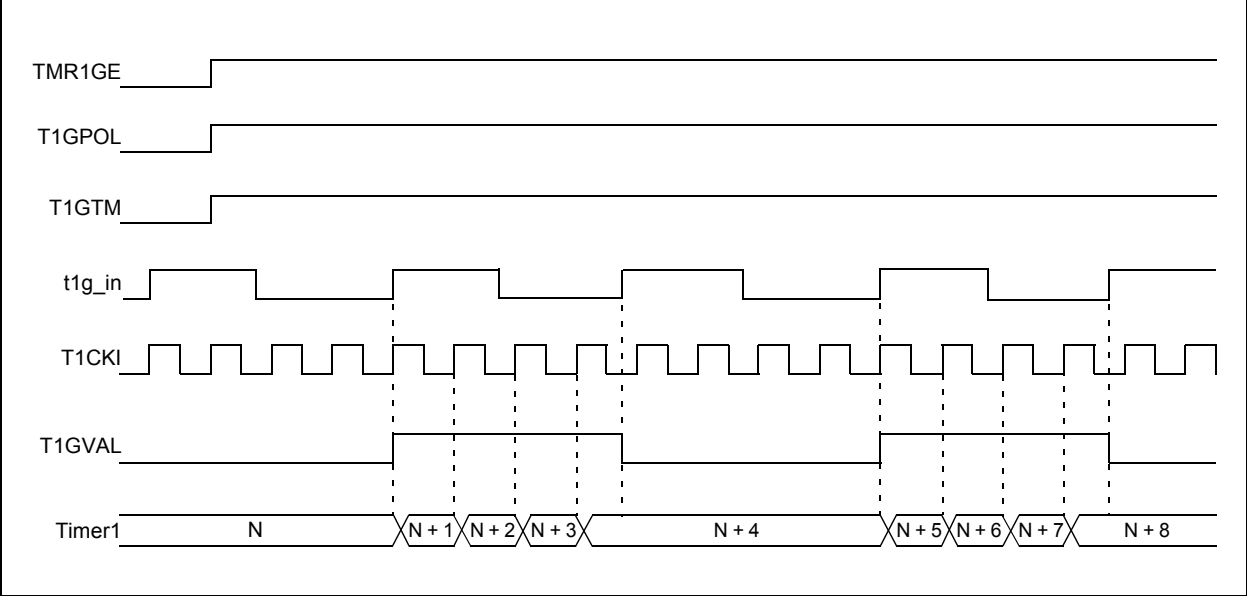
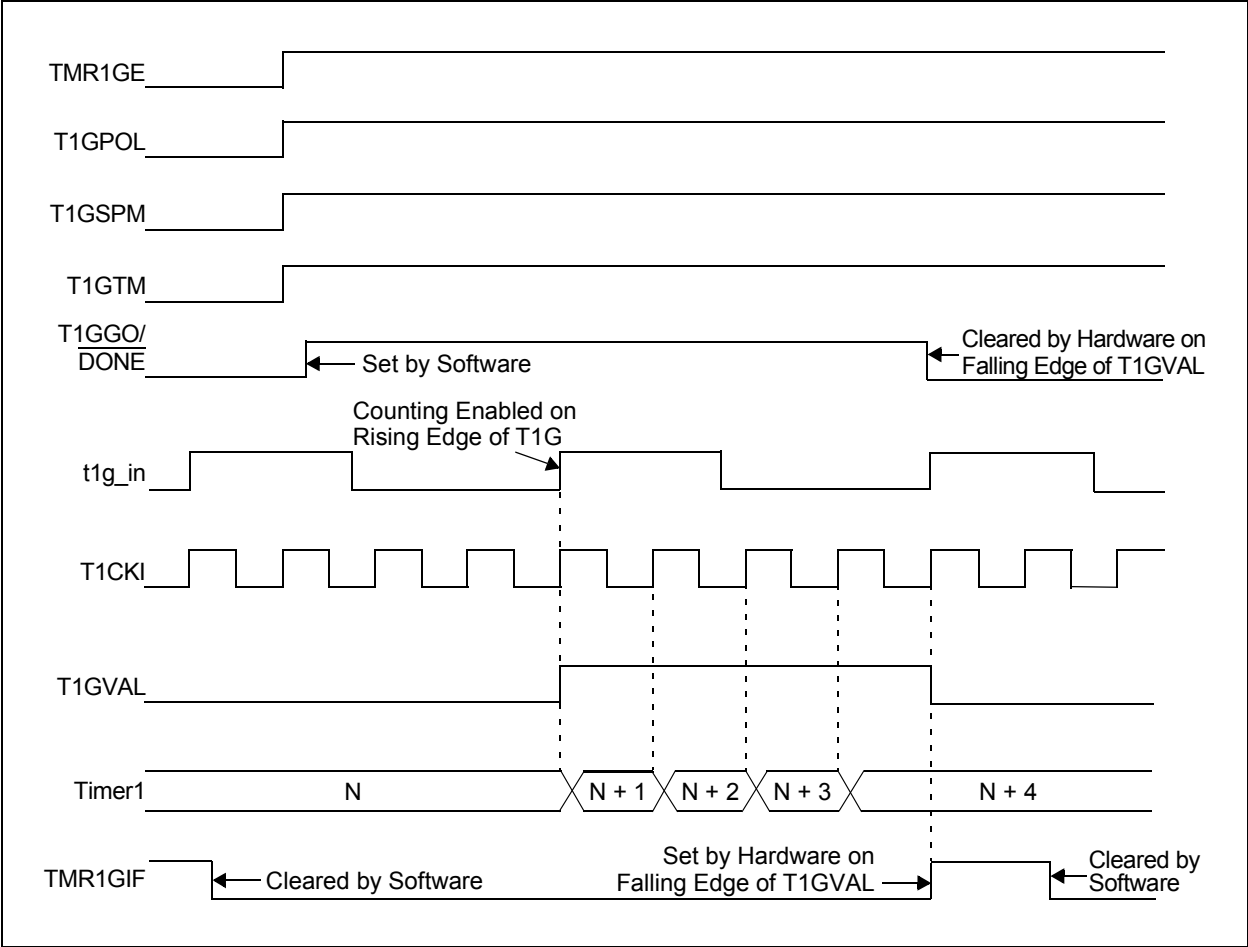


FIGURE 19-6: TIMER1 GATE SINGLE-PULSE AND TOGGLE COMBINED MODE



21.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard Non-Return-to-Zero (NRZ) format. NRZ is implemented with two levels: a V_{OH} mark state which represents a '1' data bit, and a V_{OL} space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port Idles in the mark state. Each character transmission consists of one Start bit, followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 21-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

21.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 21-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

21.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

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

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSELx bit.

Note: The TXIF transmitter interrupt flag is set when the TXEN enable bit is set.

21.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one T_{cy} immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

21.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDCON register. The default state of this bit is '0' which selects high true transmit Idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true Idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section 21.5.1.2 "Clock Polarity"**.

21.1.1.4 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of the TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	186
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	74
PIE1	TMR1GIE	ADIE	RCIE ⁽¹⁾	TXIE ⁽¹⁾	—	—	TMR2IE	TMR1IE	75
PIR1	TMR1GIF	ADIF	RCIF ⁽¹⁾	TXIF ⁽¹⁾	—	—	TMR2IF	TMR1IF	78
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	185*
SPBRGL	BRG<7:0>								187*
SPBRGH	BRG<15:8>								187*
TXREG	EUSART Transmit Data Register								177
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	184

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for asynchronous transmission.

* Page provides register information.

Note 1: PIC12(L)F1572 only.

21.4.2 AUTO-BAUD OVERFLOW

During the course of Automatic Baud Detection, the ABDOVF bit of the BAUDCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPBRGH:SPBRGL register pair. The overflow condition will set the RCIF flag. The counter continues to count until the fifth rising edge is detected on the RX pin. The RCIDL bit will remain false ('0') until the fifth rising edge, at which time, the RDICL bit will set. If the RCREG is read after the overflow occurs, but before the fifth rising edge, the fifth rising edge will set the RCIF again.

Terminating the auto-baud process early to clear an overflow condition will prevent proper detection of the Sync character fifth rising edge. If any falling edges of the Sync character have not yet occurred when the ABDEN bit is cleared, then those will be falsely detected as Start bits. The following steps are recommended to clear the overflow condition:

1. Read RCREG to clear RCIF.
2. If RCIDL is zero, then wait for RCIF and repeat Step 1.
3. Clear the ABDOVF bit.

21.4.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The auto-wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The auto-wake-up feature is enabled by setting the WUE bit of the BAUDCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 21-7), and asynchronously if the device is in Sleep mode (Figure 21-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

21.4.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled, the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be ten or more bit times; 13-bit times are recommended for LIN bus or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

PIC12(L)F1571/2

REGISTER 22-7: PWMxPHH: PWMx PHASE COUNT HIGH 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
PH<15:8>							
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 7-0 **PH<15:8>**: PWMx Phase High bits
Upper eight bits of PWM phase count.

REGISTER 22-8: PWMxPHL: PWMx PHASE COUNT LOW 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
PH<7:0>							
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 7-0 **PH<7:0>**: PWMx Phase Low bits
Lower eight bits of PWM phase count.

Note: There are no long and short bit name variants for the following three mirror registers

REGISTER 22-17: PWMEN: PWMEN BIT ACCESS REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	PWM3EN_A	PWM2EN_A	PWM1EN_A
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **PWMxEN_A:** PWM3/PWM2/PWM1 Enable bits
Mirror copy of EN bit (PWMxCON<7>).

REGISTER 22-18: PWMLD: LD BIT ACCESS REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **PWMxLDA_A:** PWM3/PWM2/PWM1 LD bits
Mirror copy of LD bit (PWMxLDCON<7>).

REGISTER 22-19: PWMOUT: PWMOUT BIT ACCESS REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **PWMxOUT_A:** PWM3/PWM2/PWM1 Output bits
Mirror copy of OUT bit (PWMxCON<5>).

FIGURE 27-3: I_{DD}, EC OSCILLATOR, LOW-POWER MODE, F_{osc} = 500 kHz, PIC12LF1571/2 ONLY

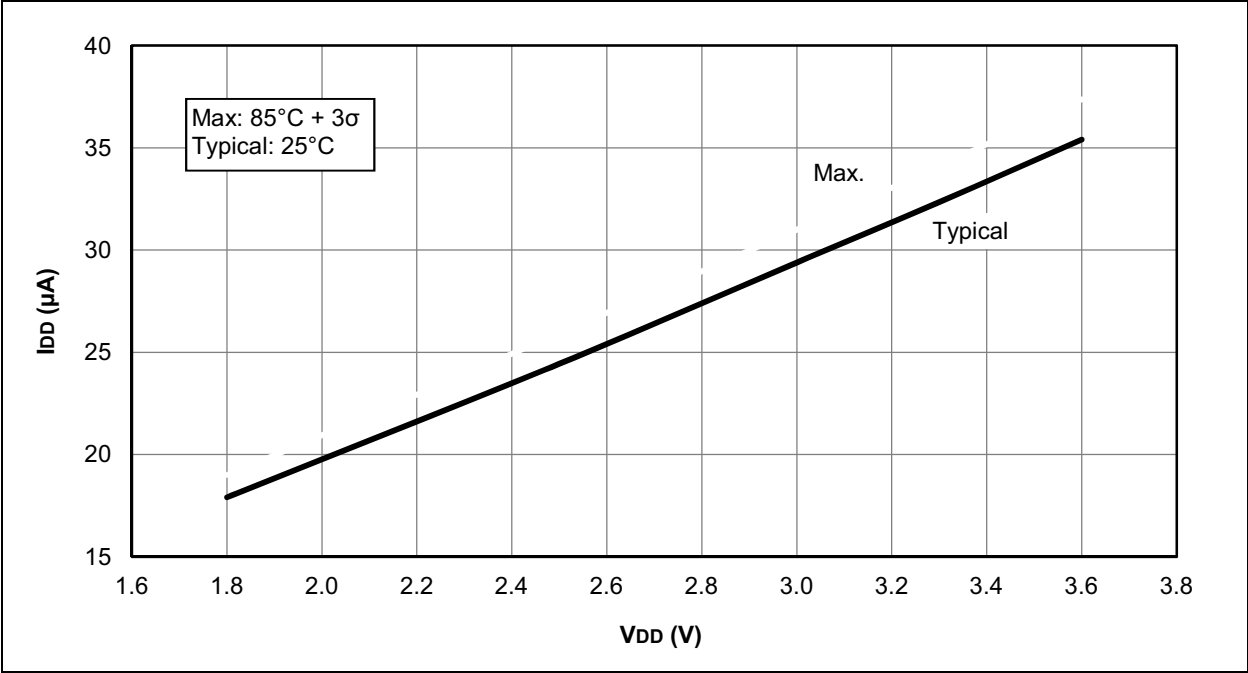


FIGURE 27-4: I_{DD}, EC OSCILLATOR, LOW-POWER MODE, F_{osc} = 500 kHz, PIC12F1571/2 ONLY

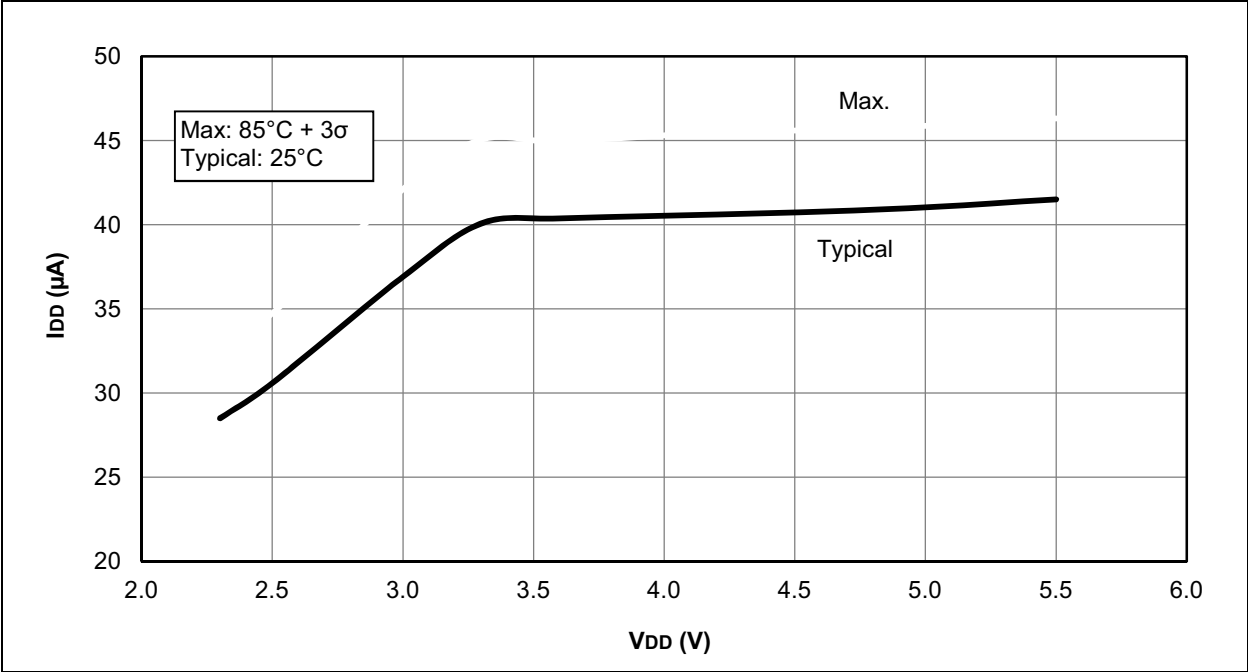


FIGURE 27-7: I_{DD} TYPICAL, EC OSCILLATOR, MEDIUM POWER MODE, PIC12F1571/2 ONLY

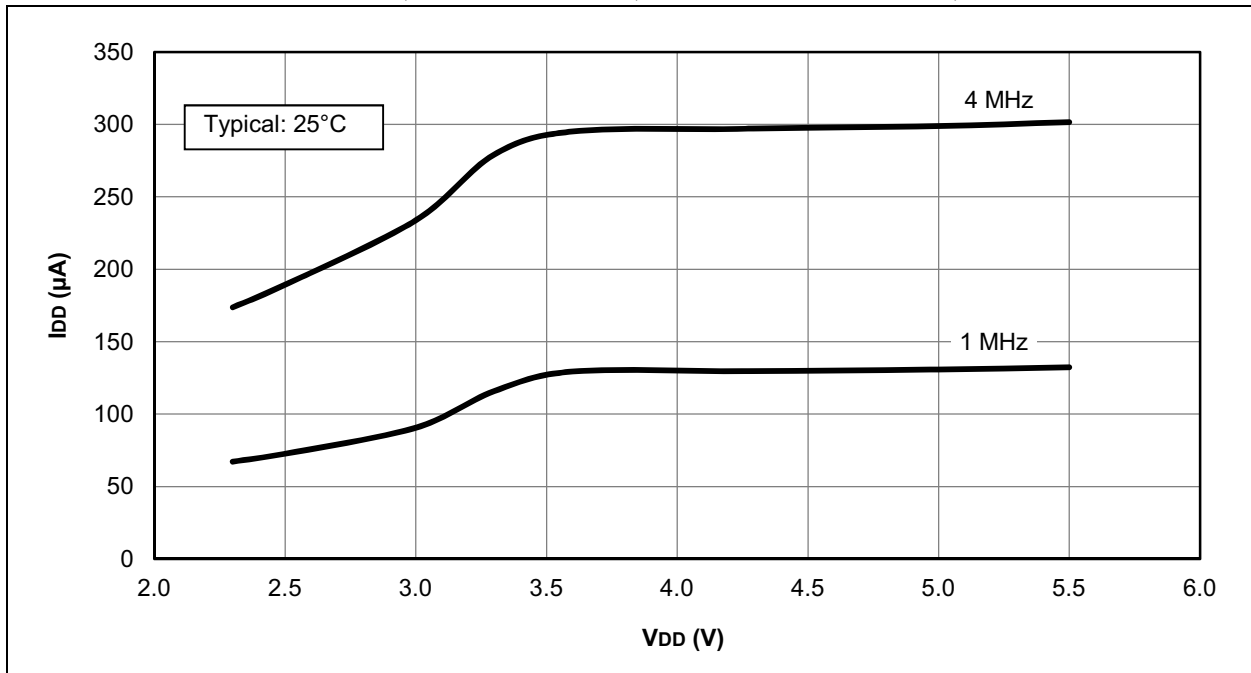


FIGURE 27-8: I_{DD} MAXIMUM, EC OSCILLATOR, MEDIUM POWER MODE, PIC12F1571/2 ONLY

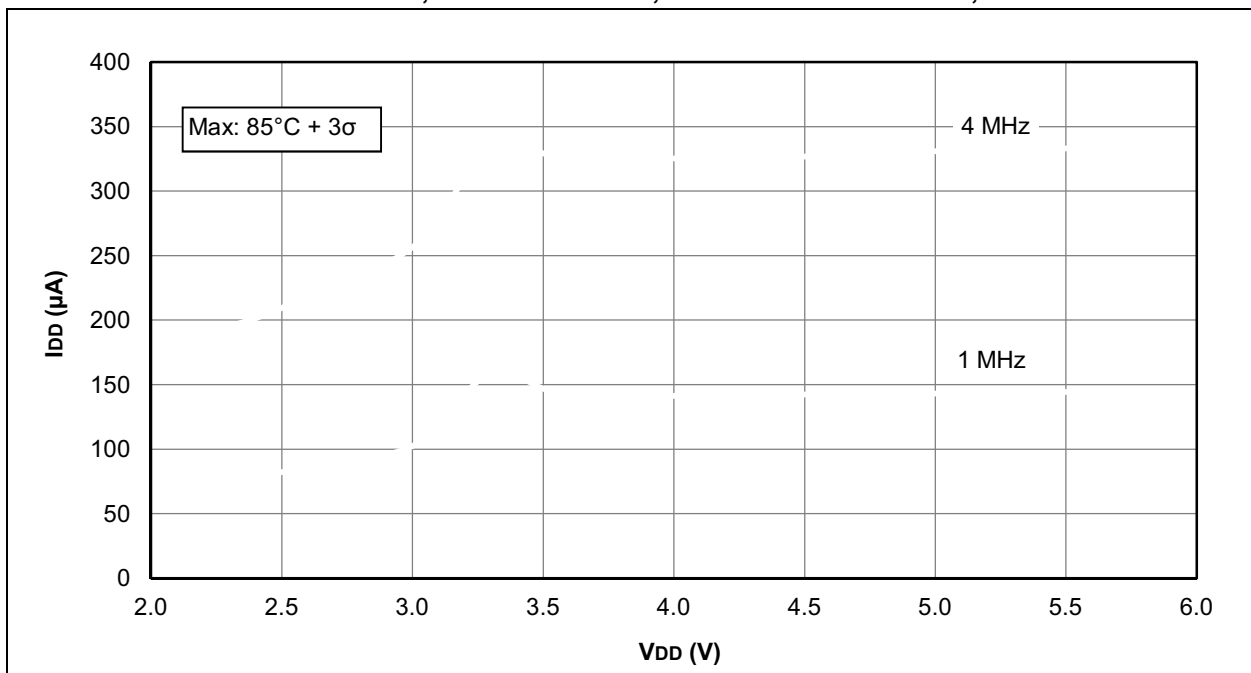


FIGURE 27-11: I_{DD} TYPICAL, EC OSCILLATOR, HIGH-POWER MODE, PIC12F1571/2 ONLY

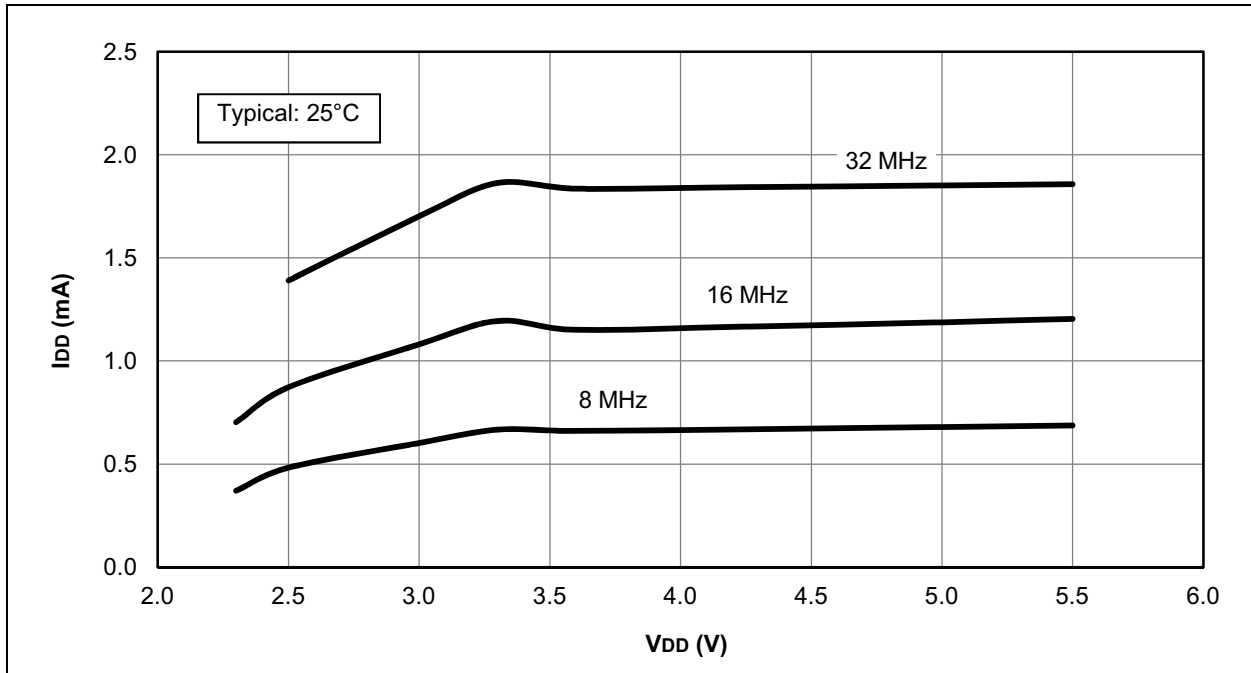
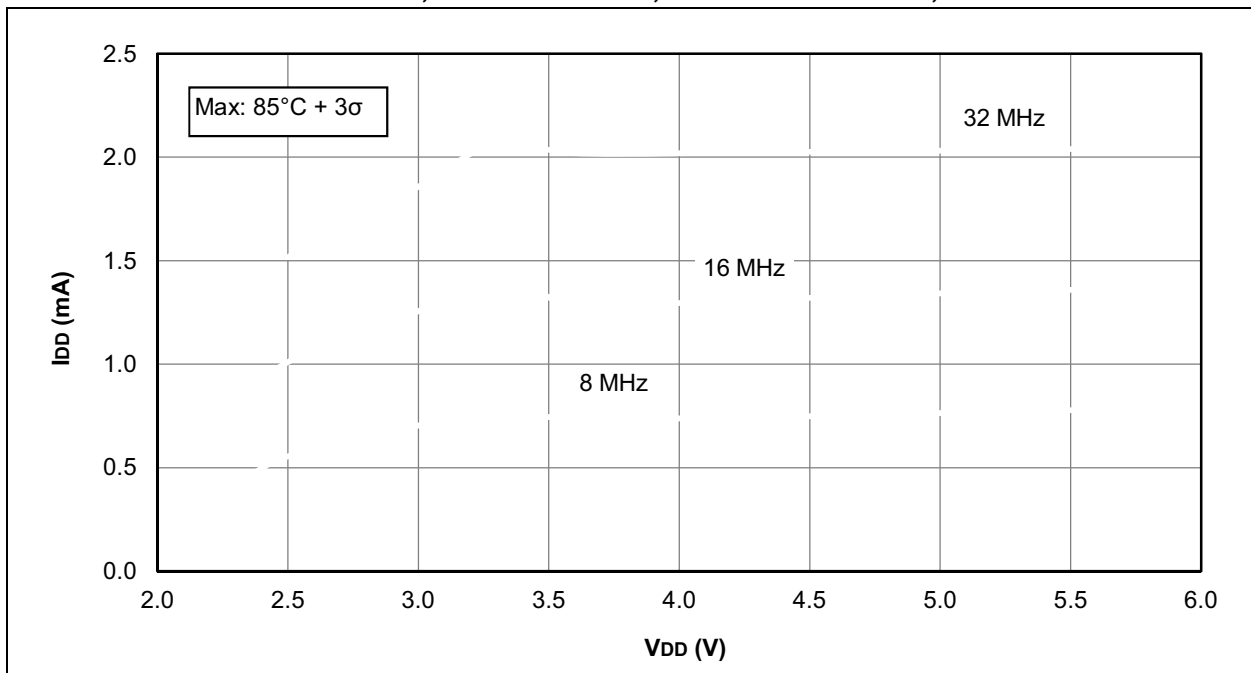


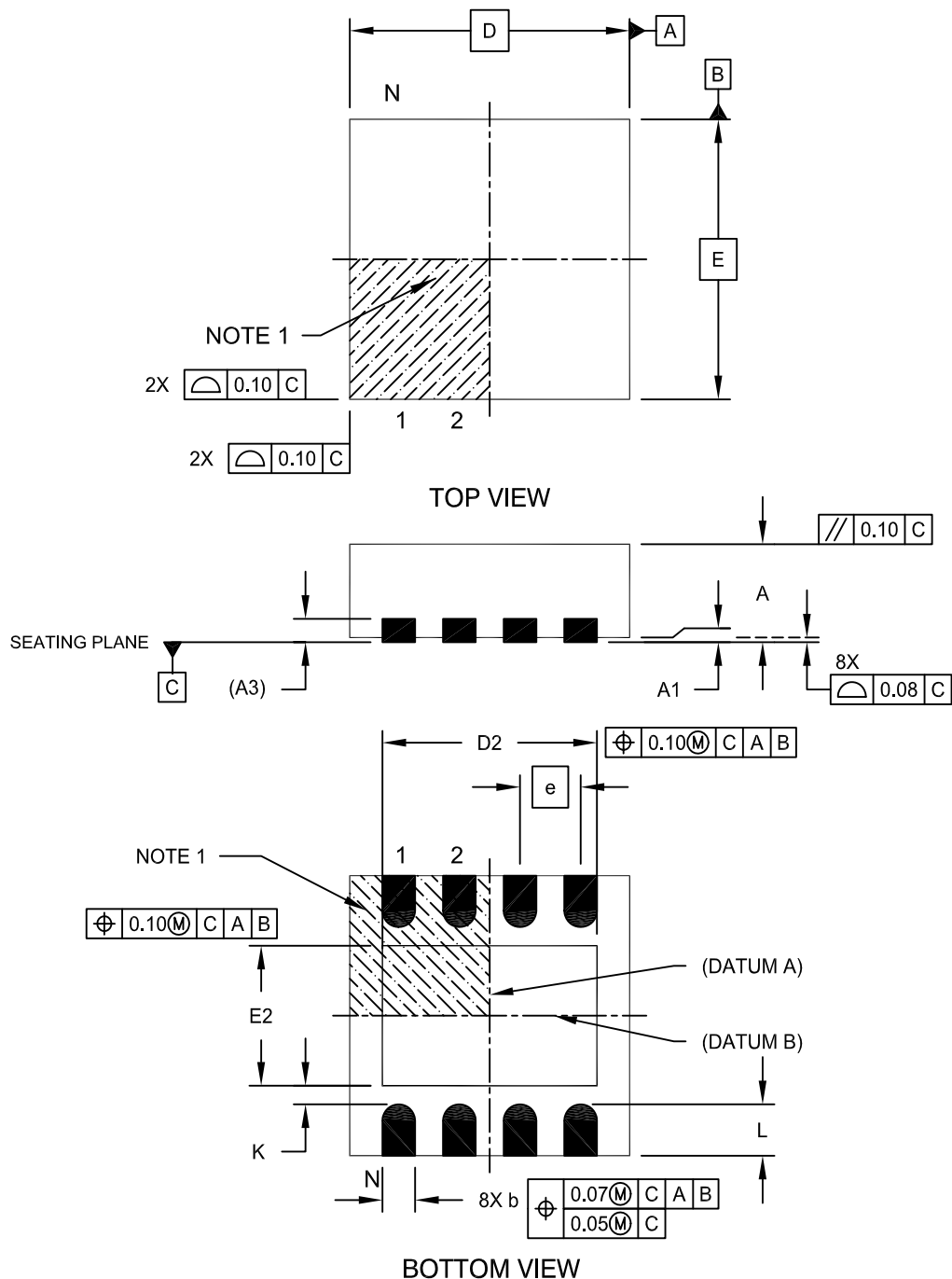
FIGURE 27-12: I_{DD} MAXIMUM, EC OSCILLATOR, HIGH-POWER MODE, PIC12F1571/2 ONLY



PIC12(L)F1571/2

8-Lead Plastic Dual Flat, No Lead Package (MF) - 3x3x0.9mm Body [DFN]

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

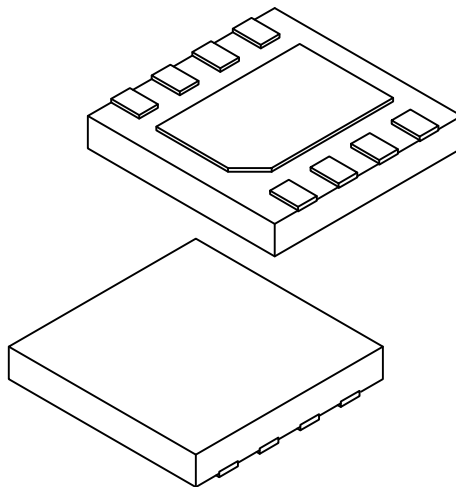


Microchip Technology Drawing No. C04-062C Sheet 1 of 2

PIC12(L)F1571/2

8-Lead Ultra Thin Plastic Dual Flat, No Lead Package (RF) - 3x3x0.50 mm Body [UDFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N	8		
Pitch	e	0.65 BSC		
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.065 REF		
Overall Width	E	3.00 BSC		
Exposed Pad Width	E2	1.40	1.50	1.60
Overall Length	D	3.00 BSC		
Exposed Pad Length	D2	2.20	2.30	2.40
Terminal Width	b	0.25	0.30	0.35
Terminal Length	L	0.35	0.45	0.55
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-254A Sheet 2 of 2