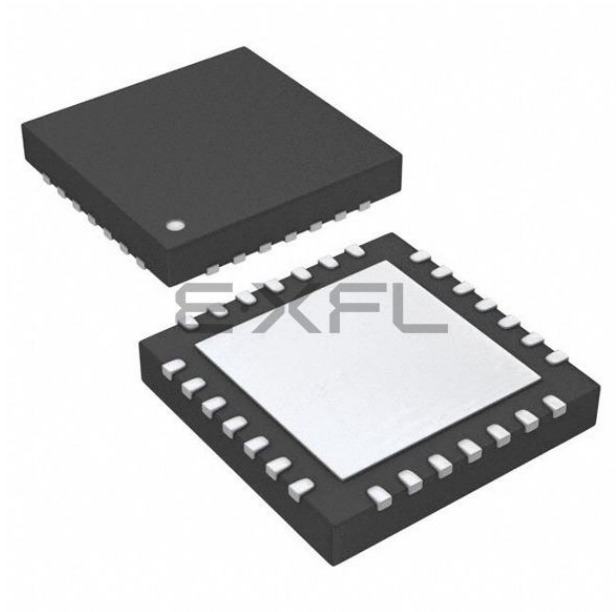


Welcome to E-XFL.COM

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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic16f1827-e-mv

TABLE 1: 18/20/28-PIN SUMMARY (PIC16(L)F1826/27)

I/O	18-Pin PDIP/SOIC	20-Pin SSOP	28-Pin QFN/UQFN	ANSEL	A/D	Reference	Cap Sense	Comparator	SR Latch	Timers	CCP	EUSART	MSSP	Interrupt	Modulator	Pull-up	Basic
RA0	17	19	23	Y	AN0	—	CPS0	C12IN0-	—	—	—	—	SDO2(2)	—	—	N	—
RA1	18	20	24	Y	AN1	—	CPS1	C12IN1-	—	—	—	—	$\overline{SS2}$(2)	—	—	N	—
RA2	1	1	26	Y	AN2	VREF- DACOUT	CPS2	C12IN2- C12IN+	—	—	—	—	—	—	—	N	—
RA3	2	2	27	Y	AN3	VREF+	CPS3	C12IN3- C1IN+ C1OUT	SRQ	—	CCP3(2)	—	—	—	—	N	—
RA4	3	3	28	Y	AN4	—	CPS4	C2OUT	SRNQ	T0CKI	CCP4(2)	—	—	—	—	N	—
RA5	4	4	1	N	—	—	—	—	—	—	—	—	$\overline{SS1}$(1)	—	—	Y(3)	$\overline{MCLR}$, VPP
RA6	15	17	20	N	—	—	—	—	—	—	P1D(1) P2B(1,2)	—	SDO1(1)	—	—	N	OSC2 CLKOUT CLKR
RA7	16	18	21	N	—	—	—	—	—	—	P1C(1) CCP2(1,2) P2A(1,2)	—	—	—	—	N	OSC1 CLKIN
RB0	6	7	7	N	—	—	—	—	SRI	T1G	CCP1(1) P1A(1) FLT0	—	—	INT IOC	—	Y	—
RB1	7	8	8	Y	AN11	—	CPS11	—	—	—	—	RX(1,4) DT(1,4)	SDA1 SDI1	IOC	—	Y	—
RB2	8	9	9	Y	AN10	—	CPS10	—	—	—	—	RX(1)DT(1) TX(1,4) CK(1,4)	SDA2(2) SDI2(2) SDO1(1,4)	IOC	MDMIN	Y	—
RB3	9	10	10	Y	AN9	—	CPS9	—	—	—	CCP1(1,4) P1A(1,4)	—	—	IOC	MDOUT	Y	—
RB4	10	11	12	Y	AN8	—	CPS8	—	—	—	—	—	SCL1 SCK1	IOC	MDCIN2	Y	—
RB5	11	12	13	Y	AN7	—	CPS7	—	—	—	P1B	TX(1) CK(1)	SCL2(2) $\overline{SCK2}$(2) $\overline{SS1}$(1,4)	IOC	—	Y	—
RB6	12	13	15	Y	AN5	—	CPS5	—	—	T1CKI T1OSI	P1C(1,4) CCP2(1,2,4) P2A(1,2,4)	—	—	IOC	—	Y	ICSPCLK/ ICDCLK
RB7	13	14	16	Y	AN6	—	CPS6	—	—	T1OSO	P1D(1,4) P2B(1,2,4)	—	—	IOC	MDCIN1	Y	ICSPDAT/ ICDDAT
VDD	14	15,16	17,19	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD
Vss	5	5,6	3,5	—	—	—	—	—	—	—	—	—	—	—	—	—	Vss

Note 1: Pin functions can be moved using the APFCON0 or APFCON1 register.
2: Functions are only available on the PIC16(L)F1827.
3: Weak pull-up always enabled when $\overline{MCLR}$ is enabled, otherwise the pull-up is under user control.
4: Default function location.

Table of Contents

3.0 MEMORY ORGANIZATION

There are three types of memory in PIC16(L)F1826/27: Data Memory, Program Memory and Data EEPROM Memory(1).

- Program Memory
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM
  - Common RAM
  - Device Memory Maps
  - Special Function Registers Summary
- Data EEPROM memory(1)

Note 1: The Data EEPROM Memory and the method to access Flash memory through the EECON registers is described in **Section 11.0 "Data EEPROM and Flash Program Memory Control"**.

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing a 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16(L)F1826/27 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 3-1 and 3-2).

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address
PIC16(L)F1826	2,048	07FFh
PIC16(L)F1827	4,096	0FFFh

5.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS bits:

- Default system oscillator determined by FOSC bits in Configuration Word 1
- Timer1 32 kHz crystal oscillator
- Internal Oscillator Block (INTOSC)

5.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS) bits of the OSCCON register selects the system clock source that is used for the CPU and peripherals.

- When the SCS bits of the OSCCON register = 00, the system clock source is determined by value of the FOSC<2:0> bits in the Configuration Word 1.
- When the SCS bits of the OSCCON register = 01, the system clock source is the Timer1 oscillator.
- When the SCS bits of the OSCCON register = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCS bits of the OSCCON register are always cleared.

Note:	Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bits of the OSCCON register. The user can monitor the OSTS bit of the OSCSTAT register to determine the current system clock source.

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 5-1.

5.3.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCSTAT register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Word 1, or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes. The OST does not reflect the status of the Timer1 Oscillator.

5.3.3 TIMER1 OSCILLATOR

The Timer1 Oscillator is a separate crystal oscillator associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the T1OSO and T1OSI device pins.

The Timer1 oscillator is enabled using the T1OSCEN control bit in the T1CON register. See **Section 21.0 "Timer1 Module with Gate Control"** for more information about the Timer1 peripheral.

5.3.4 TIMER1 OSCILLATOR READY (T1OSCR) BIT

The user must ensure that the Timer1 Oscillator is ready to be used before it is selected as a system clock source. The Timer1 Oscillator Ready (T1OSCR) bit of the OSCSTAT register indicates whether the Timer1 oscillator is ready to be used. After the T1OSCR bit is set, the SCS bits can be configured to select the Timer1 oscillator.

7.0 RESETS

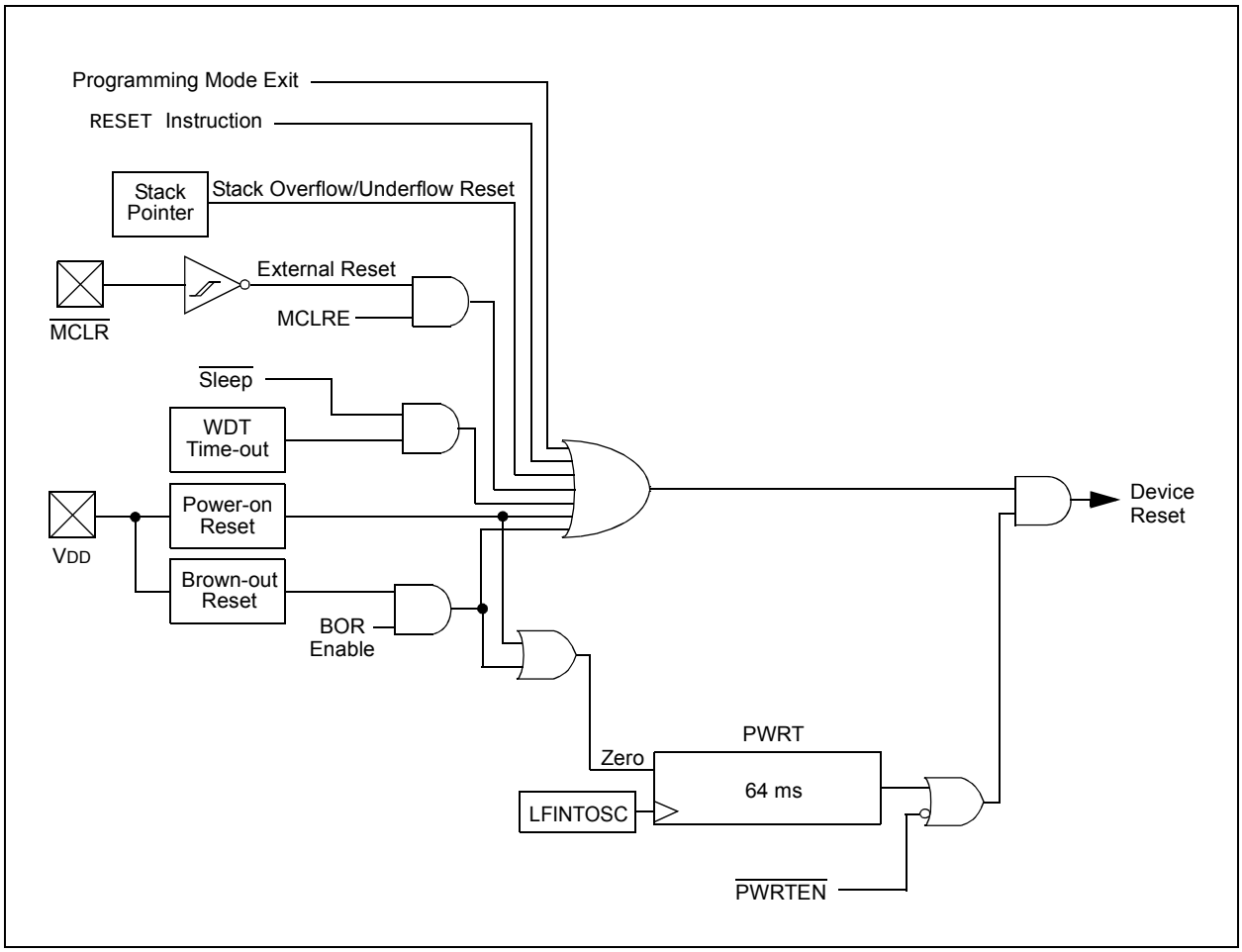
There are multiple ways to reset this device:

- Power-on Reset (POR)
- Brown-out Reset (BOR)
- $\overline{\text{MCLR}}$ Reset
- WDT Reset
- RESET instruction
- Stack Overflow
- Stack Underflow
- Programming mode exit

To allow VDD to stabilize, an optional power-up timer can be enabled to extend the Reset time after a BOR or POR event.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 7-1.

FIGURE 7-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



EXAMPLE 11-3: FLASH PROGRAM MEMORY READ

```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI : PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  EEADRL           ; Select Bank for EEPROM registers
    MOVLW    PROG_ADDR_LO     ;
    MOVWF    EEADRL           ; Store LSB of address
    MOVLW    PROG_ADDR_HI     ;
    MOVWL    EEADRH           ; Store MSB of address

    BCF      EECON1,CFGS      ; Do not select Configuration Space
    BSF      EECON1,EEPGD     ; Select Program Memory
    BCF      INTCON,GIE       ; Disable interrupts
    BSF      EECON1,RD        ; Initiate read
    NOP                       ; Ignored (Figure 11-1)
    NOP                       ; Ignored (Figure 11-1)
    BSF      INTCON,GIE       ; Restore interrupts

    MOVF     EEDATL,W         ; Get LSB of word
    MOVWF    PROG_DATA_LO     ; Store in user location
    MOVF     EEDATH,W         ; Get MSB of word
    MOVWF    PROG_DATA_HI     ; Store in user location
```

EXAMPLE 11-5: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:
; 1. The 16 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)
;
        BCF         INTCON,GIE     ; Disable ints so required sequences will execute properly
        BANKSEL     EEADRH         ; Bank 3
        MOVF        ADDRH,W        ; Load initial address
        MOVWF       EEADRH         ;
        MOVF        ADDRL,W        ;
        MOVWF       EEADRL         ;
        MOVLW       LOW DATA_ADDR  ; Load initial data address
        MOVWF       FSR0L          ;
        MOVLW       HIGH DATA_ADDR ; Load initial data address
        MOVWF       FSR0H          ;
        BSF         EECON1,EEPGD   ; Point to program memory
        BCF         EECON1,CFGS    ; Not configuration space
        BSF         EECON1,WREN    ; Enable writes
        BSF         EECON1,LWLO    ; Only Load Write Latches
LOOP
        MOVIW       FSR0++         ; Load first data byte into lower
        MOVWF       EEDATL         ;
        MOVIW       FSR0++         ; Load second data byte into upper
        MOVWF       EEDATH         ;

        MOVF        EEADRL,W       ; Check if lower bits of address are '000'
        XORLW       0x07           ; Check if we're on the last of 8 addresses
        ANDLW       0x07           ;
        BTFSC       STATUS,Z       ; Exit if last of eight words,
        GOTO        START_WRITE    ;

  Required Sequence
        MOVLW       55h            ; Start of required write sequence:
        MOVWF       EECON2         ; Write 55h
        MOVLW       0AAh           ;
        MOVWF       EECON2         ; Write AAh
        BSF         EECON1,WR      ; Set WR bit to begin write
        NOP                        ; Any instructions here are ignored as processor
                                   ; halts to begin write sequence
        NOP                        ; Processor will stop here and wait for write to complete.

                                   ; After write processor continues with 3rd instruction.

        INCF        EEADRL,F       ; Still loading latches Increment address
        GOTO        LOOP           ; Write next latches

START_WRITE
        BCF         EECON1,LWLO    ; No more loading latches - Actually start Flash program
                                   ; memory write

  Required Sequence
        MOVLW       55h            ; Start of required write sequence:
        MOVWF       EECON2         ; Write 55h
        MOVLW       0AAh           ;
        MOVWF       EECON2         ; Write AAh
        BSF         EECON1,WR      ; Set WR bit to begin write
        NOP                        ; Any instructions here are ignored as processor
                                   ; halts to begin write sequence
        NOP                        ; Processor will stop here and wait for write complete.

                                   ; after write processor continues with 3rd instruction
        BCF         EECON1,WREN    ; Disable writes
        BSF         INTCON,GIE     ; Enable interrupts
```

11.6 Write Verify

Depending on the application, good programming practice may dictate that the value written to the data EEPROM or program memory should be verified (see Example 11-6) to the desired value to be written. Example 11-6 shows how to verify a write to EEPROM.

EXAMPLE 11-6: EEPROM WRITE VERIFY

```
BANKSEL EEDATL      ;
MOVF    EEDATL, W   ;EEDATL not changed
                    ;from previous write
BSF     EECON1, RD  ;YES, Read the
                    ;value written
XORWF   EEDATL, W   ;
BTFSS   STATUS, Z   ;Is data the same
GOTO    WRITE_ERR   ;No, handle error
:                   ;Yes, continue
```

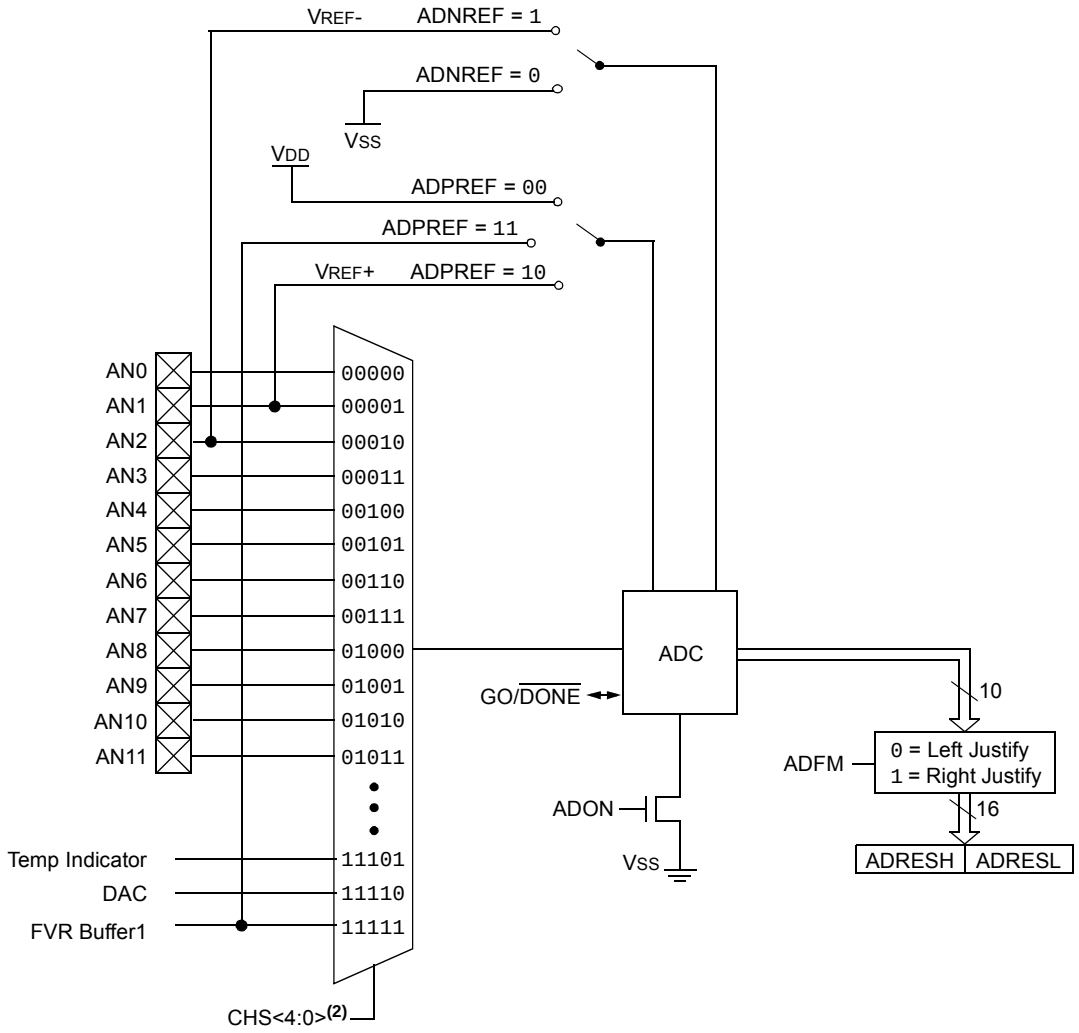
16.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 16-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

FIGURE 16-1: ADC BLOCK DIAGRAM



Note 1: When ADON = 0, all multiplexer inputs are disconnected.

2: See ADCON0 register (Register 16-1) for detailed analog channel selection per device.

16.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (Refer to the TRIS register)
   - Configure pin as analog (Refer to the ANSEL register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: Refer to **Section 16.4 "A/D Acquisition Requirements"**.

EXAMPLE 16-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd and Vss references, Frc
;clock and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1      ;
MOVLW     B'11110000' ;Right justify, Frc
                      ;clock
MOVWF     ADCON1      ;Vdd and Vss Vref
BANKSEL   TRISA       ;
BSF       TRISA,0     ;Set RA0 to input
BANKSEL   ANSEL       ;
BSF       ANSEL,0     ;Set RA0 to analog
BANKSEL   ADCON0      ;
MOVLW     B'00000001' ;Select channel AN0
MOVWF     ADCON0      ;Turn ADC On
CALL      SampleTime  ;Acquisiton delay
BSF       ADCON0,ADGO ;Start conversion
BTFSC     ADCON0,ADGO ;Is conversion done?
GOTO      $-1         ;No, test again
BANKSEL   ADRESH      ;
MOVF      ADRESH,W    ;Read upper 2 bits
MOVWF     RESULTHI    ;store in GPR space
BANKSEL   ADRESL      ;
MOVF      ADRESL,W    ;Read lower 8 bits
MOVWF     RESULTLO    ;Store in GPR space
```

NOTES:

REGISTER 19-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>		—	—	CxNCH<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 **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive going edge of the CxOUT bit

bit 6 **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative going edge of the CxOUT bit

bit 5-4 **CxPCH<1:0>:** Comparator Positive Input Channel Select bits
00 = CxVP connects to CxIN+ pin
01 = CxVP connects to DAC Voltage Reference
10 = CxVP connects to FVR Voltage Reference
For C1:
11 = CxVP connects to C12IN+ pin
For C2:
11 = CxVP connects to VSS

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **CxNCH<1:0>:** Comparator Negative Input Channel Select bits
00 = CxVN connects to C12IN0- pin
01 = CxVN connects to C12IN1- pin
10 = CxVN connects to C12IN2- pin
11 = CxVN connects to C12IN3- pin

REGISTER 19-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
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 7-2 **Unimplemented:** Read as '0'

bit 1 **MC2OUT:** Mirror Copy of C2OUT bit

bit 0 **MC1OUT:** Mirror Copy of C1OUT bit

21.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

21.4 Timer1 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

Note:	The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to using Timer1. A suitable delay similar to the OST delay can be implemented in software by clearing the TMR1IF bit then presetting the TMR1H:TMR1L register pair to FC00h. The TMR1IF flag will be set when 1024 clock cycles have elapsed, thereby indicating that the oscillator is running and reasonably stable.

21.5 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 21.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

Note:	When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

21.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

21.6 Timer1 Gate

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

Timer1 gate can also be driven by multiple selectable sources.

21.6.1 TIMER1 GATE ENABLE

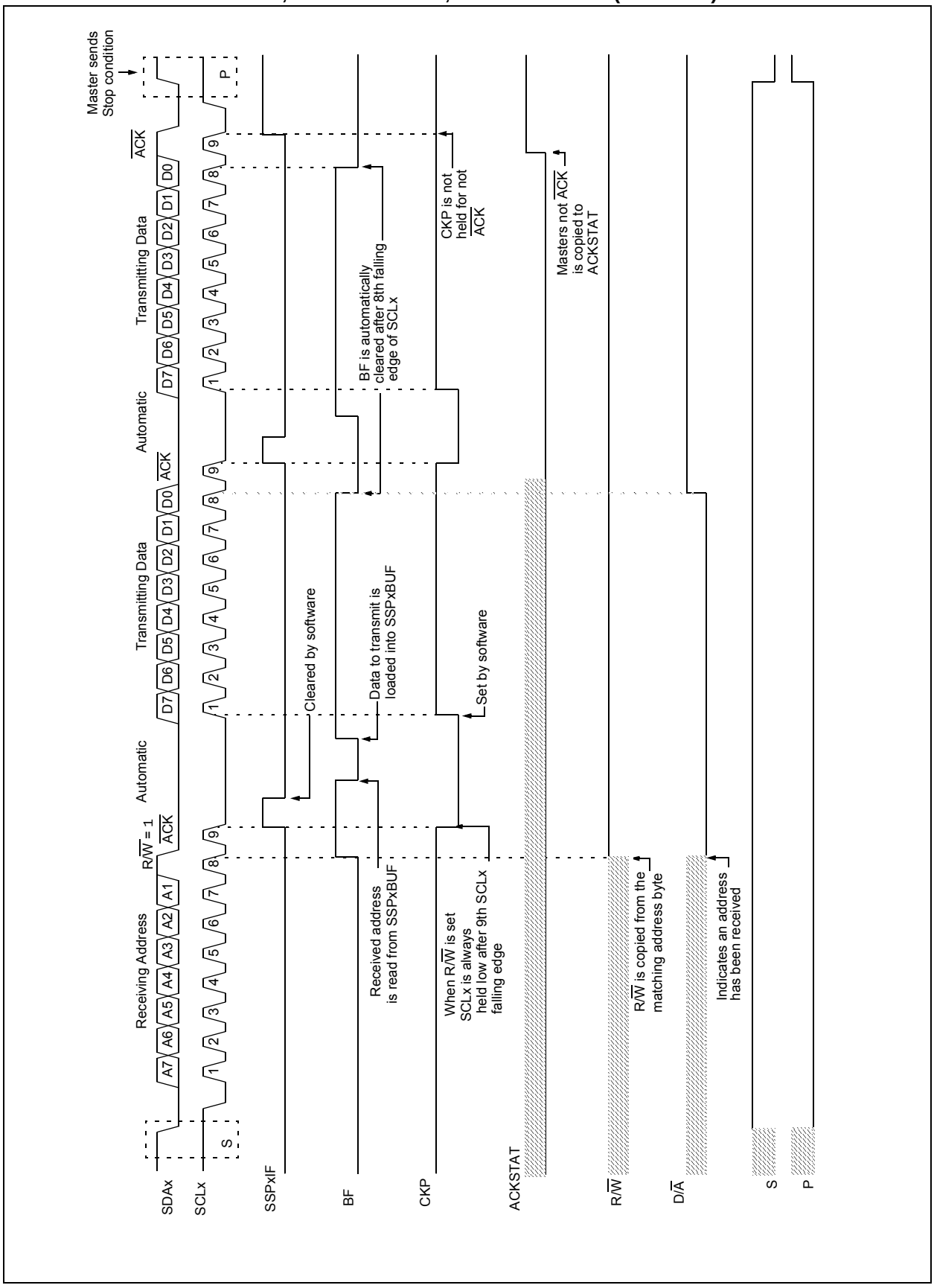
The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 21-3 for timing details.

TABLE 21-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

FIGURE 25-18: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)



25.6 I²C MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPxM bits in the SSPxCON1 register and by setting the SSPxEN bit. In Master mode, the SDAx and SCKx pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDAx and SCLx lines.

The following events will cause the SSPx Interrupt Flag bit, SSPxIF, to be set (SSPx interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1: The MSSPx module, when configured in I²C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur

2: When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

25.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDAx, while SCLx outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDAx, while SCLx outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCLx. See **Section 25.7 "Baud Rate Generator"** for more detail.

TABLE 26-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

26.3.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTA register. The Break character transmission is then initiated by a write to the TXREG. The value of data written to TXREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTA register indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 26-9 for the timing of the Break character sequence.

26.3.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXREG with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREG becomes empty, as indicated by the TXIF, the next data byte can be written to TXREG.

26.3.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTA register and the Received data as indicated by RCREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

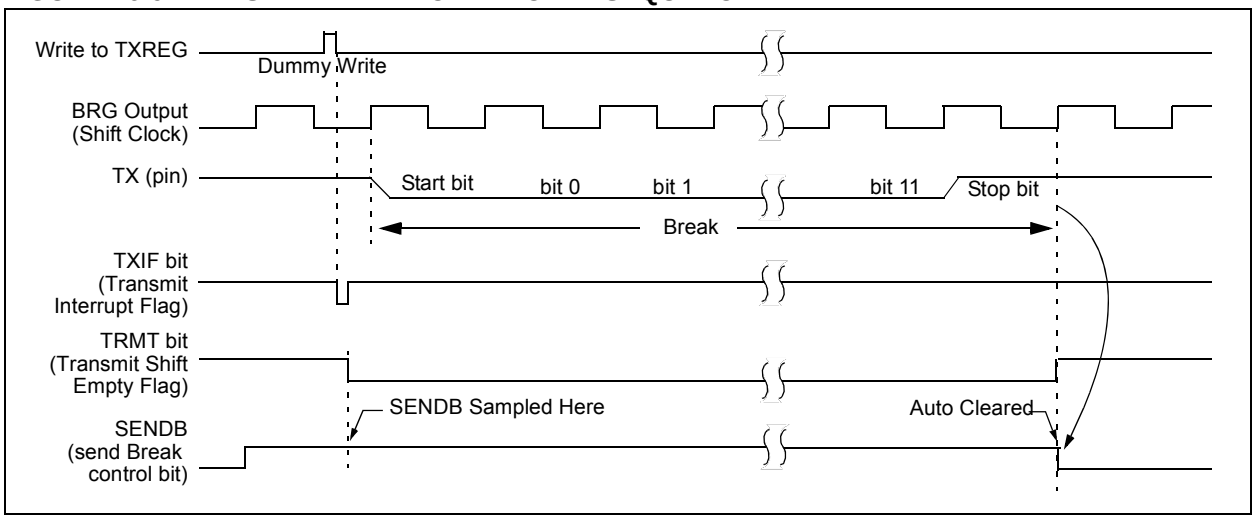
A Break character has been received when;

- RCIF bit is set
- FERR bit is set
- RCREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 26.3.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCON register before placing the EUSART in Sleep mode.

FIGURE 26-9: SEND BREAK CHARACTER SEQUENCE



30.3 DC Characteristics: PIC16(L)F1826/27-I/E (Power-Down)

PIC16LF1826/27			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F1826/27			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	**Device Characteristics**	**Min.**	**Typ†**	**Max. +85°C**	**Max. +125°C**	**Units**	**Conditions**	
							VDD	**Note**
	Power-down Base Current (IPD)(2)							
D022		—	0.02	1.0	4.0	µA	1.8	WDT, BOR, FVR, and T1OSC disabled, all Peripherals Inactive
		—	0.03	1.1	7.0	µA	3.0	
D022		—	15	35	50	µA	1.8	WDT, BOR, FVR, and T1OSC disabled, all Peripherals Inactive
		—	18	40	60	µA	3.0	
		—	19	45	70	µA	5.0	
D023		—	0.5	1.1	5.0	µA	1.8	LPWDT Current **(Note 1)**
		—	0.8	2.0	8.0	µA	3.0	
D023		—	16	35	50	µA	1.8	LPWDT Current **(Note 1)**
		—	19	40	60	µA	3.0	
		—	20	45	70	µA	5.0	
D023A		—	8.5	23	32	µA	1.8	FVR current **(Note 1)**
		—	8.5	26	40	µA	3.0	
D023A		—	32	62	66	µA	1.8	FVR current **(Note 1)**
		—	39	70	80	µA	3.0	
		—	70	110	120	µA	5.0	
D024		—	8.1	14	20	µA	3.0	BOR Current **(Note 1)**
D024		—	34	57	70	µA	3.0	BOR Current **(Note 1)**
		—	67	100	115	µA	5.0	
D025		—	0.6	1.5	5.0	µA	1.8	T1OSC Current **(Note 1)**
		—	0.8	2.5	8.0	µA	3.0	
D025		—	16	35	50	µA	1.8	T1OSC Current **(Note 1)**
		—	21	40	60	µA	3.0	
		—	25	45	70	µA	5.0	
D026		—	0.1	1.1	5.0	µA	1.8	A/D Current **(Note 1, Note 3)**, no conversion in progress
		—	0.1	2.0	8.0	µA	3.0	
D026		—	16	35	50	µA	1.8	A/D Current **(Note 1, Note 3)**, no conversion in progress
		—	21	40	60	µA	3.0	
		—	25	45	70	µA	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 peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins set to inputs state and tied to VDD.

3: A/D oscillator source is FRC.

30.6 Thermal Considerations

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	**Sym.**	**Characteristic**	**Typ.**	**Units**	**Conditions**
TH01	θJA	Thermal Resistance Junction to Ambient	65.5	°C/W	18-pin PDIP package
			76	°C/W	18-pin SOIC package
			89.3	°C/W	20-pin SSOP package
			TBD	°C/W	28-pin UQFN 4x4mm package
			31.1	°C/W	28-pin QFN 6x6mm package
TH02	θJC	Thermal Resistance Junction to Case	29.5	°C/W	18-pin PDIP package
			23.5	°C/W	18-pin SOIC package
			31.1	°C/W	20-pin SSOP package
			TBD	°C/W	28-pin UQFN 4x4mm package
			5	°C/W	28-pin QFN 6x6mm package
TH03	TJMAX	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	—	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	Internal Power Dissipation	—	W	PINTERNAL = IDD x VDD(1)
TH06	PI/O	I/O Power Dissipation	—	W	PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))
TH07	PDER	Derated Power	—	W	PDER = PDMAX (TJ - TA)/θJA(2)

Legend: TBD = To Be Determined

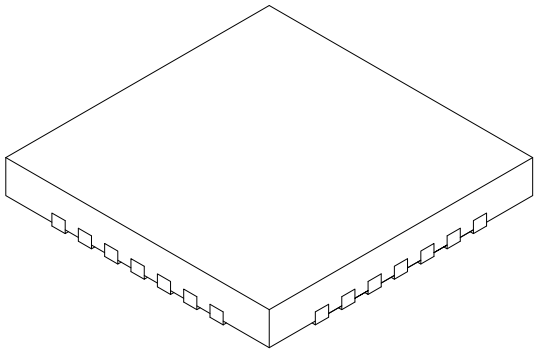
Note 1: IDD is current to run the chip alone without driving any load on the output pins.

2: TA = Ambient Temperature

3: TJ = Junction Temperature

28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.40 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.55	2.65	2.75
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.55	2.65	2.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-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 C04-152A Sheet 2 of 2