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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1936-e-ss

PIC16(L)F1934/6/7

TABLE 3-11: **PIC16(L)F1934/6/7 MEMORY MAP, BANK 31**

Bank 31	
F8Ch	
	Unimplemented Read as '0'
FE3h	
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

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

3.2.6 SPECIAL FUNCTION REGISTERS SUMMARY

The Special Function Register Summary for the device family are as follows:

Device	Bank(s)	Page No.
PIC16(L)F1934/6/7	0	39
	1	40
	2	41
	3	42
	4	43
	5	44
	6	45
	7	46
	8	47
	9-14	48
	15	49
	16-30	51
	31	52

4.0 DEVICE CONFIGURATION

Device Configuration consists of Configuration Word 1 and Configuration Word 2, Code Protection and Device ID.

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 at 8007h and Configuration Word 2 at 8008h.

Note:	The $\overline{\text{DEBUG}}$ bit in Configuration Word 2 is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.

4.2 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection and data EEPROM protection are controlled independently. Internal access to the program memory and data EEPROM are unaffected by any code protection setting.

4.2.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in Configuration Word 1. When $\overline{CP}$ = 0, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. See **Section 4.3 "Write Protection"** for more information.

4.2.2 DATA EEPROM PROTECTION

The entire data EEPROM is protected from external reads and writes by the $\overline{CPD}$ bit. When $\overline{CPD}$ = 0, external reads and writes of data EEPROM are inhibited. The CPU can continue to read and write data EEPROM regardless of the protection bit settings.

4.3 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as bootloader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Word 2 define the size of the program memory block that is protected.

4.4 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 4.5 "Device ID and Revision ID"** for more information on accessing these memory locations. For more information on checksum calculation, see the "*PIC16F193X/LF193X/PIC16F194X/LF194X/PIC16LF 190X Memory Programming Specification*" (DS41397).

5.2.2.3 Internal Oscillator Frequency Adjustment

The 500 kHz internal oscillator is factory calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register (Register 5-3). Since the HFINTOSC and MFINTOSC clock sources are derived from the 500 kHz internal oscillator a change in the OSCTUNE register value will apply to both.

The default value of the OSCTUNE register is '0'. The value is a 5-bit two's complement number. A value of 0Fh will provide an adjustment to the maximum frequency. A value of 10h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

5.2.2.4 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is an uncalibrated 31 kHz internal clock source.

The output of the LFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). Select 31 kHz, via software, using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.7 "Internal Oscillator Clock Switch Timing"** for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<3:0> bits of the OSCCON register = 000) as the system clock source (SCS bits of the OSCCON register = 1x), or when any of the following are enabled:

• Configure the IRCF<3:0> bits of the OSCCON register for the desired LF frequency, and
• FOSC<2:0> = 100, or
• Set the System Clock Source (SCS) bits of the OSCCON register to '1x'

Peripherals that use the LFINTOSC are:

• Power-up Timer (PWRT)
• Watchdog Timer (WDT)
• Fail-Safe Clock Monitor (FSCM)

The Low-Frequency Internal Oscillator Ready bit (LFIOFR) of the OSCSTAT register indicates when the LFINTOSC is running and can be utilized.

5.2.2.5 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The output of the 16 MHz HFINTOSC and 31 kHz LFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

• 32 MHz (requires 4X PLL)
• 16 MHz
• 8 MHz
• 4 MHz
• 2 MHz
• 1 MHz
• 500 kHz (Default after Reset)
• 250 kHz
• 125 kHz
• 62.5 kHz
• 31.25 kHz
• 31 kHz (LFINTOSC)

> **Note:** Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

6.3 MCLR

The $\overline{\text{MCLR}}$ is an optional external input that can reset the device. The $\overline{\text{MCLR}}$ function is controlled by the MCLRE bit of Configuration Word 1 and the LVP bit of Configuration Word 2 (Table 6-2).

TABLE 6-2: MCLR CONFIGURATION

MCLRE	LVP	$\overline{\text{MCLR}}$
0	0	Disabled
1	0	Enabled
x	1	Enabled

6.3.1 $\overline{\text{MCLR}}$ ENABLED

When $\overline{\text{MCLR}}$ is enabled and the pin is held low, the device is held in Reset. The $\overline{\text{MCLR}}$ pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

> **Note:** A Reset does not drive the $\overline{\text{MCLR}}$ pin low.

6.3.2 $\overline{\text{MCLR}}$ DISABLED

When $\overline{\text{MCLR}}$ is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 12.6 "PORTE Registers"** for more information.

6.4 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a `CLRWDT` instruction within the time-out period. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register are changed to indicate the WDT Reset. See **Section 10.0 "Watchdog Timer"** for more information.

6.5 `RESET` Instruction

A `RESET` instruction will cause a device Reset. The RI bit in the PCON register will be set to '0'. See Table 6-4 for default conditions after a `RESET` instruction has occurred.

6.6 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Word 2. See **Section 3.4.2 "Overflow/Underflow Reset"** for more information.

6.7 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

6.8 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overline{\text{PWRTE}}$ bit of Configuration Word 1.

6.9 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for oscillator source).
3. $\overline{\text{MCLR}}$ must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 5.0 "Oscillator Module (With Fail-Safe Clock Monitor)"** for more information.

The Power-up Timer and oscillator start-up timer run independently of $\overline{\text{MCLR}}$ Reset. If $\overline{\text{MCLR}}$ is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. Upon bringing $\overline{\text{MCLR}}$ high, the device will begin execution immediately (see Figure 6-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

7.6.3   PIE2 REGISTER

The PIE2 register contains the interrupt enable bits, as shown in Register 7-3.

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

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

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE
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   **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enables the Oscillator Fail interrupt
0 = Disables the Oscillator Fail interrupt

bit 6   **C2IE:** Comparator C2 Interrupt Enable bit
1 = Enables the Comparator C2 interrupt
0 = Disables the Comparator C2 interrupt

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

bit 4   **EEIE:** EEPROM Write Completion Interrupt Enable bit
1 = Enables the EEPROM Write Completion interrupt
0 = Disables the EEPROM Write Completion interrupt

bit 3   **BCLIE:** MSSP Bus Collision Interrupt Enable bit
1 = Enables the MSSP Bus Collision Interrupt
0 = Disables the MSSP Bus Collision Interrupt

bit 2   **LCDIE:** LCD Module Interrupt Enable bit
1 = Enables the LCD module interrupt
0 = Disables the LCD module interrupt

bit 1   **Unimplemented:** Read as '0'

bit 0   **CCP2IE:** CCP2 Interrupt Enable bit
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

7.6.5 PIR1 REGISTER

The PIR1 register contains the interrupt flag bits, as shown in Register 7-5.

Note:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 7-5: PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
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   **TMR1GIF:** Timer1 Gate Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 6   **ADIF:** A/D Converter Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 5   **RCIF:** USART Receive Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 4   **TXIF:** USART Transmit Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 3   **SSPIF:** Synchronous Serial Port (MSSP) Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 2   **CCP1IF:** CCP1 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 1   **TMR2IF:** Timer2 to PR2 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 0   **TMR1IF:** Timer1 Overflow Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

TABLE 10-3: **SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		64
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	29
WDTCON	—	—	WDTPS<4:0>					SWDTEN	113

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

TABLE 10-4: **SUMMARY OF CONFIGURATION WORD WITH WATCHDOG TIMER**

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		$\overline{CPD}$	62
	7:0	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE<1:0>		FOSC<2:0>			

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Watchdog Timer.

15.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

> **Note 1:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
>
> **2:** The ADC operates during Sleep only when the FRC oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.
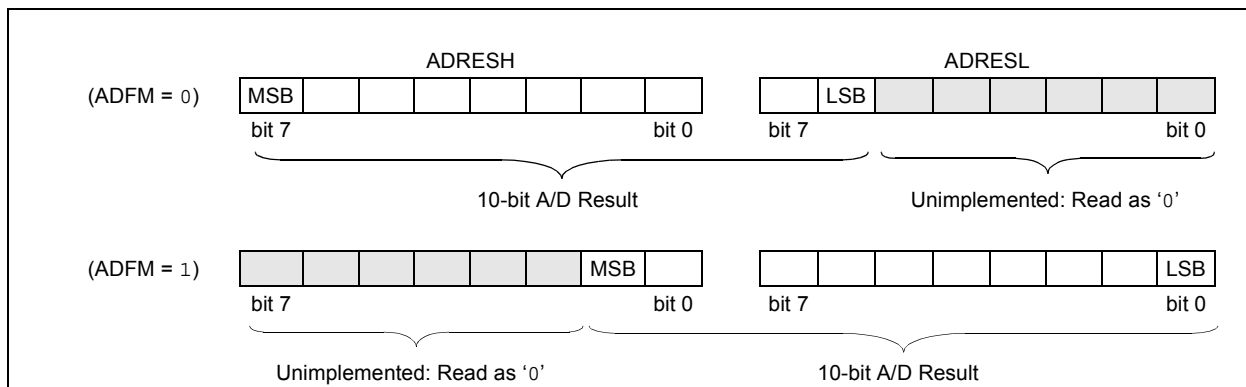
Please refer to **Section 15.1.5 "Interrupts"** for more information.

15.1.6 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure 15-3 shows the two output formats.

FIGURE 15-3: 10-BIT A/D CONVERSION RESULT FORMAT

# PIC16(L)F1934/6/7

23.4.2 FULL-BRIDGE MODE

In Full-Bridge mode, all four pins are used as outputs. An example of Full-Bridge application is shown in Figure 23-10.

In the Forward mode, pin CCPx/PxA is driven to its active state, pin PxD is modulated, while PxB and PxC will be driven to their inactive state as shown in Figure 23-11.

In the Reverse mode, PxC is driven to its active state, pin PxB is modulated, while PxA and PxD will be driven to their inactive state as shown Figure 23-11.

PxA, PxB, PxC and PxD outputs are multiplexed with the PORT data latches. The associated TRIS bits must be cleared to configure the PxA, PxB, PxC and PxD pins as outputs.

FIGURE 23-10: EXAMPLE OF FULL-BRIDGE APPLICATION
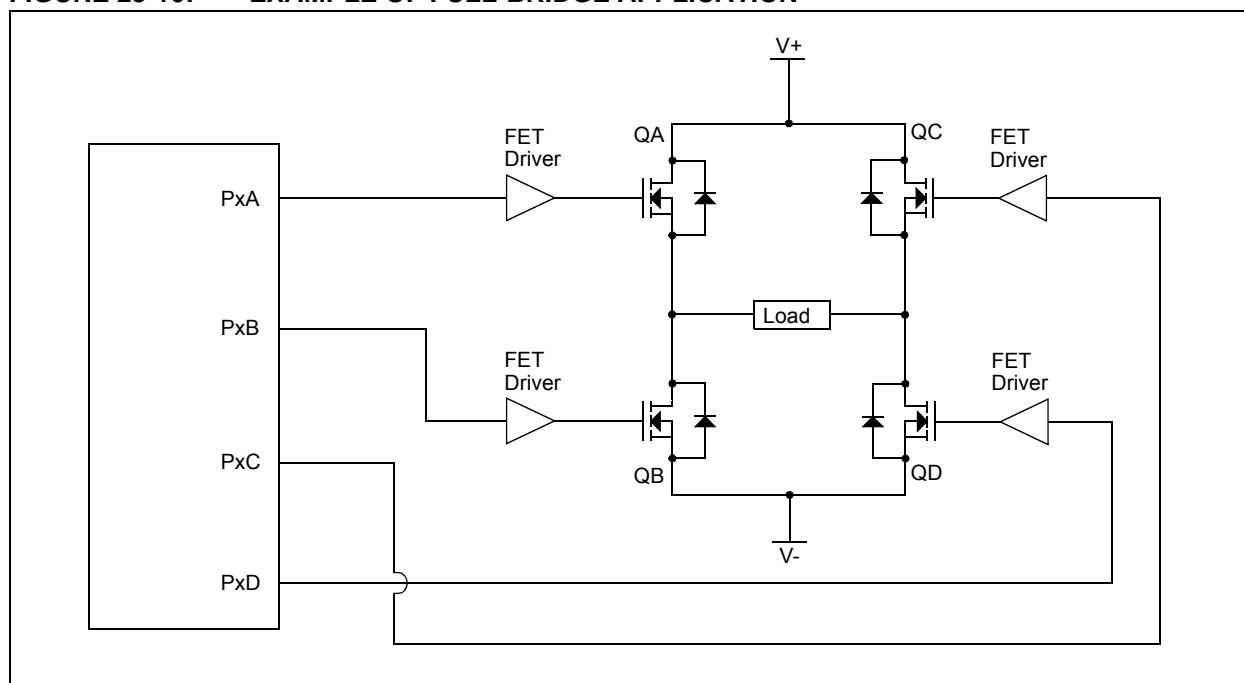
24.5.8 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I²C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is a reserved address in the I²C protocol, defined as address 0x00. When the GCEN bit of the SSPCON2 register is set, the slave module will automatically $\overline{ACK}$ the reception of this address regardless of the value stored in SSPADD. After the slave clocks in an address of all zeros with the R/$\overline{W}$ bit clear, an interrupt is generated and slave software can read SSPBUF and respond. Figure 24-23 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the 8th falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 24-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE
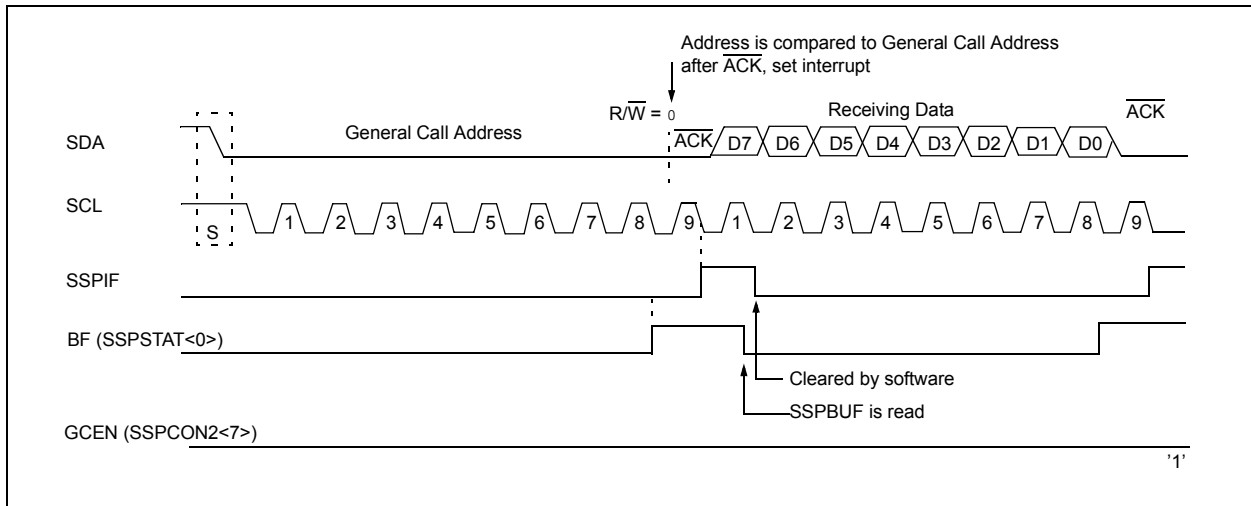
24.5.9 SSP MASK REGISTER

An SSP Mask (SSPMSK) register (Register 24-5) is available in I²C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

• 7-bit Address mode: address compare of A<7:1>.
• 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

TABLE 24-3: SUMMARY OF REGISTERS ASSOCIATED WITH I²C™ OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	98
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	99
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	—	—	CCP2IE[(1)]	100
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	102
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	—	—	CCP2IF[(1)]	103
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	142
SSPADD	ADD<7:0>								290
SSPBUF	MSSP Receive Buffer/Transmit Register								243*
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				287
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	288
SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	289
SSPMSK	MSK<7:0>								290
SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	286

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the MSSP module in I²C™ mode.
   * Page provides register information.
Note 1: PIC16F1934 only.

25.3.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII "U") which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCON register starts the auto-baud calibration sequence (Figure 25-6). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRG begins counting up using the BRG counter clock as shown in Table 25-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPBRGH, SPBRGL register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRGL register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 25-6. During ABD, both the SPBRGH and SPBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH and SPBRGL registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

> **Note 1:** If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte <u>following</u> the Break character (see **Section 25.3.3 "Auto-Wake-up on Break"**).
>
> **2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.
>
> **3:** During the auto-baud process, the auto-baud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRGL register pair.

TABLE 25-6: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	$F_{OSC}/64$	$F_{OSC}/512$
0	1	$F_{OSC}/16$	$F_{OSC}/128$
1	0	$F_{OSC}/16$	$F_{OSC}/128$
1	1	$F_{OSC}/4$	$F_{OSC}/32$

Note: During the ABD sequence, SPBRGL and SPBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

FIGURE 25-6: AUTOMATIC BAUD RATE CALIBRATION

Note 1: The ABD sequence requires the **EUSART** module to be configured in Asynchronous mode.
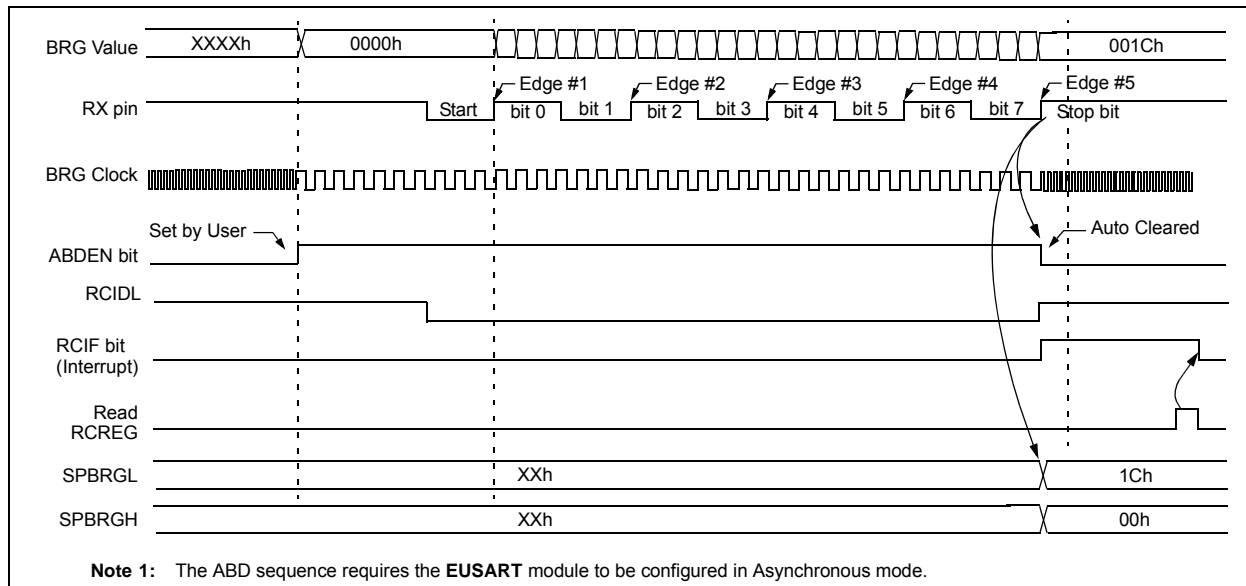
REGISTER 26-2: **CPSCON1: CAPACITIVE SENSING CONTROL REGISTER 1**

U-0	U-0	U-0	U-0	R/W-0/0**(1)**	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CPSCH<3:0>			
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 7-4 **Unimplemented:** Read as '0'

bit 3-0 **CPSCH<3:0>:** Capacitive Sensing Channel Select bits
<u>If CPSON = 0</u>:
    These bits are ignored. No channel is selected.
<u>If CPSON = 1</u>:
    0000 = channel 0, (CPS0)
    0001 = channel 1, (CPS1)
    0010 = channel 2, (CPS2)
    0011 = channel 3, (CPS3)
    0100 = channel 4, (CPS4)
    0101 = channel 5, (CPS5)
    0110 = channel 6, (CPS6)
    0111 = channel 7, (CPS7)
    1000 = channel 8, (CPS8**(1)**)
    1001 = channel 9, (CPS9**(1)**)
    1010 = channel 10, (CPS10**(1)**)
    1011 = channel 11, (CPS11**(1)**)
    1100 = channel 12, (CPS12**(1)**)
    1101 = channel 13, (CPS13**(1)**)
    1110 = channel 14, (CPS14**(1)**)
    1111 = channel 15, (CPS15**(1)**)

Note 1: These channels are not implemented on the PIC16(L)F1936.
   **2:** This bit is not implemented on PIC16(L)F1936, read as '0'

27.4.3 AUTOMATIC POWER MODE SWITCHING

As an LCD segment is electrically only a capacitor, current is drawn only during the interval where the voltage is switching. To minimize total device current, the LCD internal reference ladder can be operated in a different power mode for the transition portion of the duration. This is controlled by the LCDRL register (Register 27-7).

The LCDRL register allows switching between two power modes, designated 'A' and 'B'. 'A' Power mode is active for a programmable time, beginning at the time when the LCD segments transition. 'B' Power mode is the remaining time before the segments or commons change again. The LRLAT<2:0> bits select how long, if any, that the 'A' Power mode is active. Refer to Figure 27-4.

To implement this, the 5-bit prescaler used to divide the 32 kHz clock down to the LCD controller's 1 kHz base rate is used to select the power mode.

FIGURE 27-4: **LCD INTERNAL REFERENCE LADDER POWER MODE SWITCHING DIAGRAM – TYPE A**
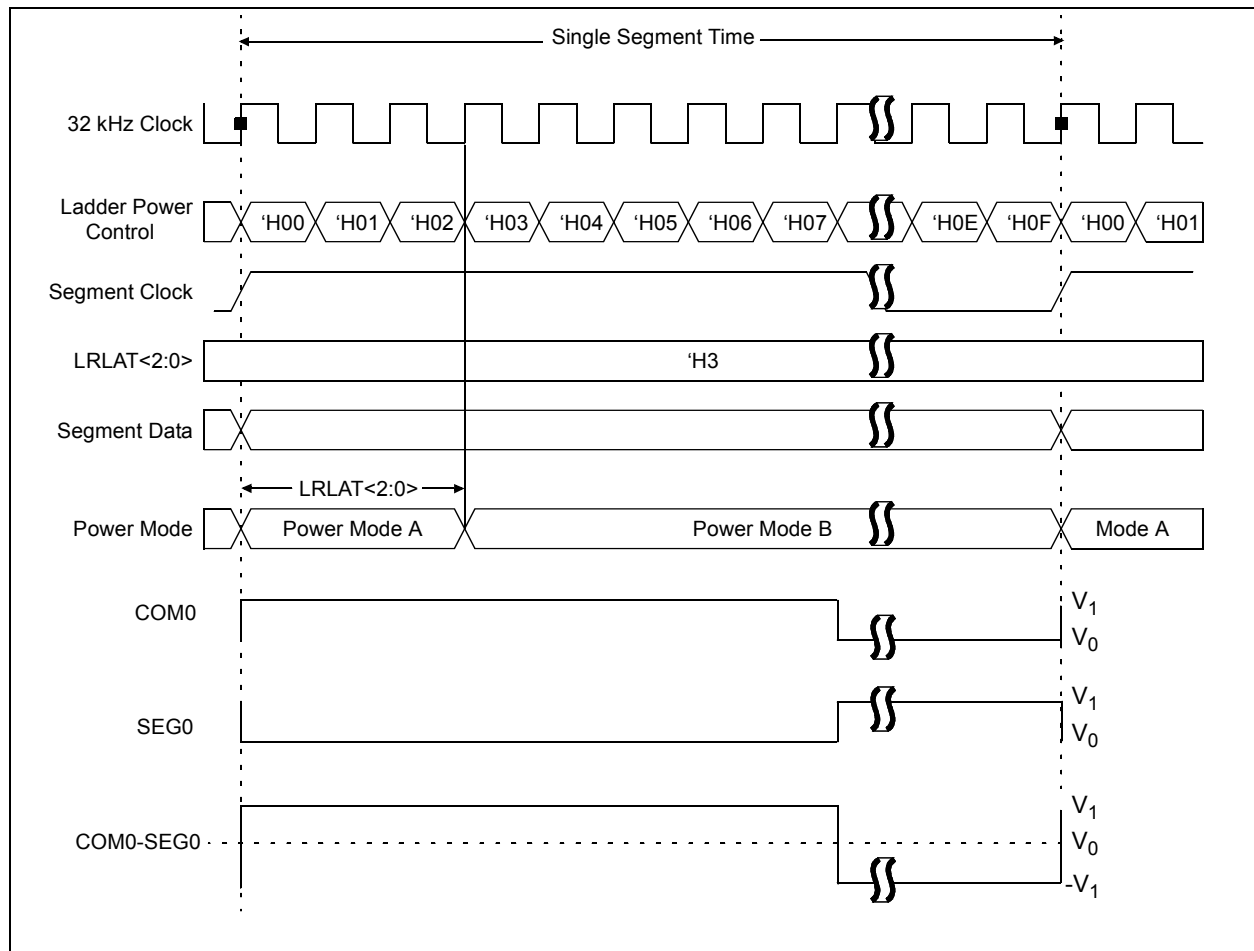
PIC16(L)F1934/6/7

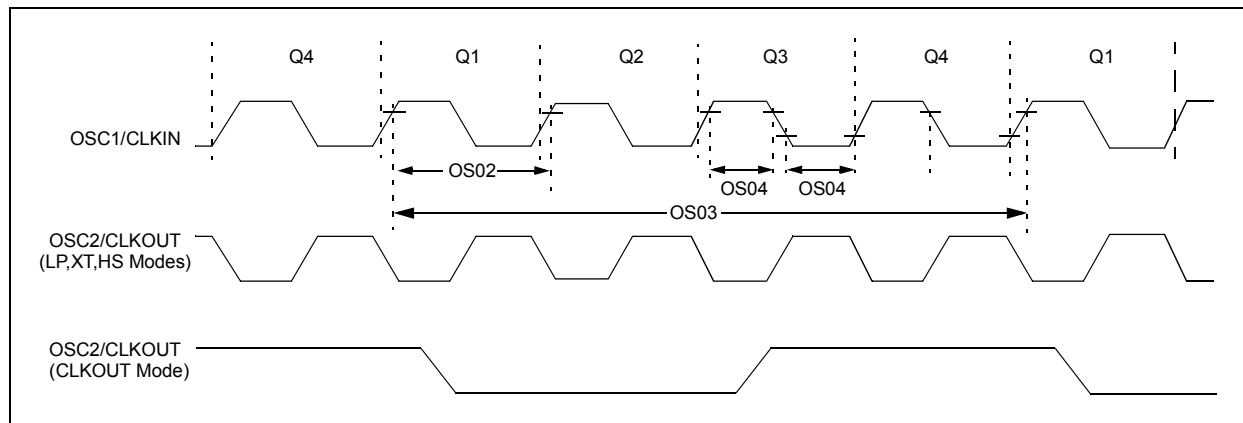## 30.8   AC Characteristics: PIC16(L)F1934/6/7-I/E

FIGURE 30-6:   CLOCK TIMING

TABLE 30-1:   CLOCK OSCILLATOR TIMING REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Operating temperature -40°C ≤ TA ≤ +125°C							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS01	FOSC	External CLKIN Frequency[1]	DC	—	0.5	MHz	EC Oscillator mode (low)
			DC	—	4	MHz	EC Oscillator mode (medium)
			DC	—	20	MHz	EC Oscillator mode (high)
		Oscillator Frequency[1]	—	32.768	—	kHz	LP Oscillator mode
			0.1	—	4	MHz	XT Oscillator mode
			1	—	4	MHz	HS Oscillator mode
			1	—	20	MHz	HS Oscillator mode, VDD > 2.7V
			DC	—	4	MHz	RC Oscillator mode, VDD ≤ 2.0V
OS02	TOSC	External CLKIN Period[1]	27	—	∞	μs	LP Oscillator mode
			250	—	∞	ns	XT Oscillator mode
			50	—	∞	ns	HS Oscillator mode
			50	—	∞	ns	EC Oscillator mode
		Oscillator Period[1]	—	30.5	—	μs	LP Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			50	—	1,000	ns	HS Oscillator mode
			250	—	—	ns	RC Oscillator mode
OS03	TCY	Instruction Cycle Time[1]	200	TCY	DC	ns	TCY = 4/FOSC
OS04*	TosH, TosL	External CLKIN High, External CLKIN Low	2	—	—	μs	LP oscillator
			100	—	—	ns	XT oscillator
			20	—	—	ns	HS oscillator
OS05*	TosR, TosF	External CLKIN Rise, External CLKIN Fall	0	—	∞	ns	LP oscillator
			0	—	∞	ns	XT oscillator
			0	—	∞	ns	HS oscillator

    *   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:   Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

TABLE 30-10: COMPARATOR SPECIFICATIONS

Operating Conditions: 1.8V < V_{DD} < 5.5V, -40°C < T_A < +125°C (unless otherwise stated).

Param No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
CM01	V_{IOFF}	Input Offset Voltage	—	±7.5	±60	mV	High-Power mode
CM02	V_{ICM}	Input Common Mode Voltage	0	—	V_{DD}	V	
CM03	CMRR	Common Mode Rejection Ratio	—	50	—	dB	
CM04A		Response Time Rising Edge	—	400	800	ns	High-Power mode
CM04B	T_{RESP}	Response Time Falling Edge	—	200	400	ns	High-Power mode
CM04C		Response Time Rising Edge	—	1200	—	ns	Low-Power mode
CM04D		Response Time Falling Edge	—	550	—	ns	Low-Power mode
CM05	T_{MC2OV}	Comparator Mode Change to Output Valid*	—	—	10	µs	
CM06	C_{HYSTER}	Comparator Hysteresis	—	45	—	mV	**Hysteresis on**

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at $V_{DD}/2$, while the other input transitions from V_{SS} to V_{DD}.

2: Comparator Hysteresis is available when the CxHYS bit of the CMxCON0 register is enabled.

TABLE 30-11: DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS

Operating Conditions: 2.5V < V_{DD} < 5.5V, -40°C < T_A < +125°C (unless otherwise stated).

Param No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
DAC01*	C_{LSB}	Step Size	—	$V_{DD}/32$	—	V	
DAC02*	C_{ACC}	Absolute Accuracy	—	—	± 1/2	LSb	
DAC03*	C_R	Unit Resistor Value (R)	—	5000	—	Ω	
DAC04*	C_{ST}	Settling Time[1]	—	—	10	µs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while DACR<4:0> transitions from '0000' to '1111'.

FIGURE 30-14: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

Note: Refer to Figure 30-5 for load conditions.
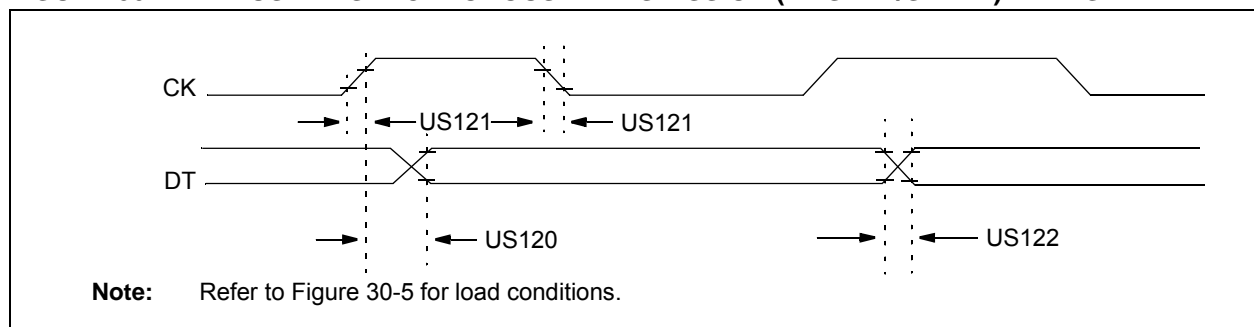
PIC16(L)F1934/6/7

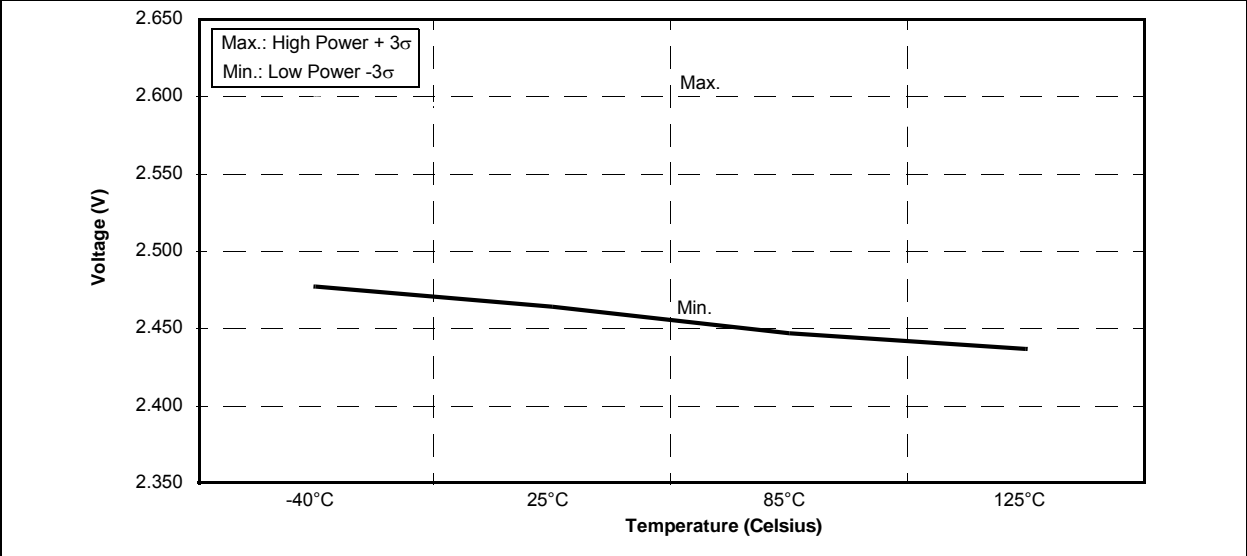**FIGURE 31-3:** **PIC16F1934/6/7 RESET VOLTAGE, BOR = 2.5V**
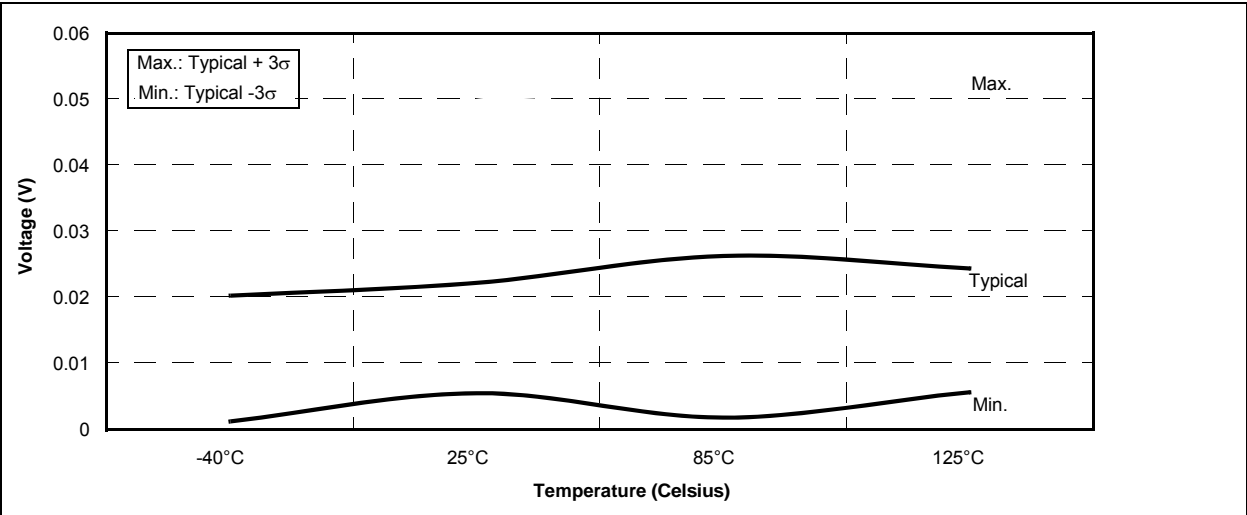

FIGURE 31-4: **PIC16F1934/6/7 HYSTERESIS, BOR = 2.5V**

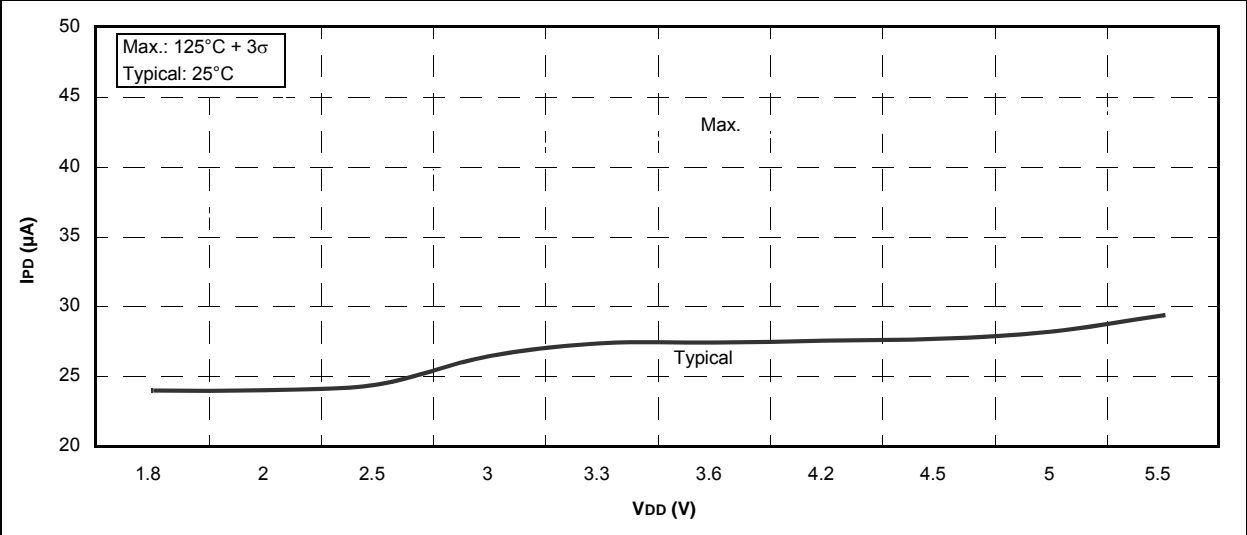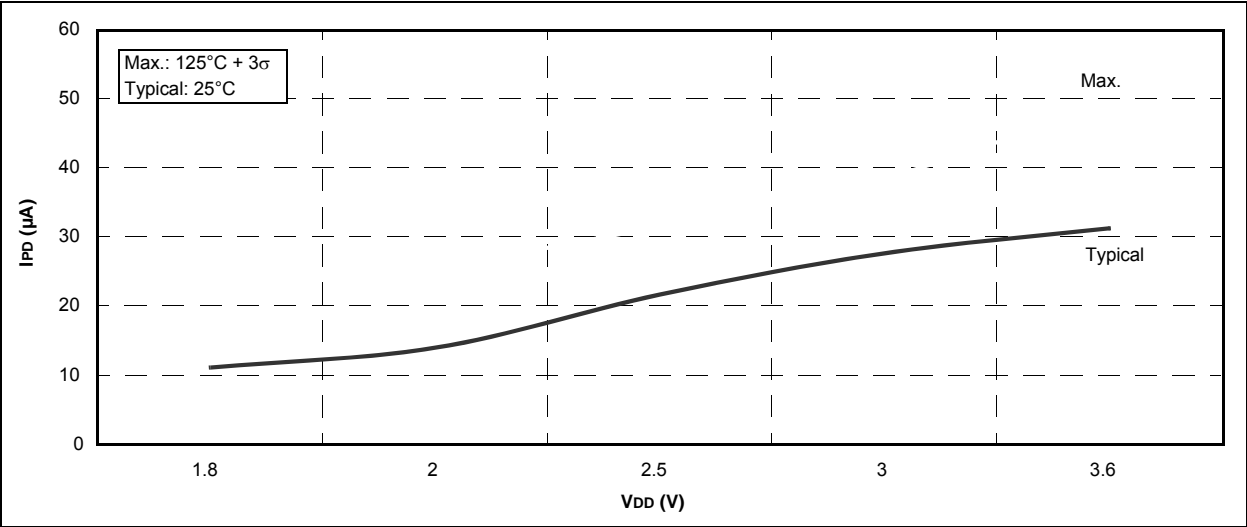**FIGURE 31-51:** **PIC16F1937 COMPARATOR 1, LOW POWER**

FIGURE 31-52: **PIC16LF1937 CAP SENSE, HIGH POWER**

32.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

32.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.