



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	I ² C, 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	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1936-e-sp

Table of Contents

1.0	Device Overview	15
2.0	Enhanced Mid-Range CPU	23
3.0	Memory Organization	25
4.0	Device Configuration	61
5.0	Oscillator Module (With Fail-Safe Clock Monitor).....	67
6.0	Resets	85
7.0	Interrupts	93
8.0	Low Dropout (LDO) Voltage Regulator	107
9.0	Power-Down Mode (Sleep)	109
10.0	Watchdog Timer (WDT)	111
11.0	Data EEPROM and Flash Program Memory Control.....	115
12.0	I/O Ports	129
13.0	Interrupt-On-Change	151
14.0	Fixed Voltage Reference	155
15.0	Analog-to-Digital Converter (ADC) Module	157
16.0	Temperature Indicator Module	171
17.0	Digital-to-Analog Converter (DAC) Module	173
18.0	Comparator Module.....	177
19.0	SR Latch.....	187
20.0	Timer0 Module	191
21.0	Timer1 Module with Gate Control.....	197
22.0	Timer2/4/6 Modules.....	207
23.0	Capture/Compare/PWM Modules (ECCP1, ECCP2, ECCP3, CCP4, CCP5).....	211
24.0	Master Synchronous Serial Port (MSSP) Module	239
25.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART).....	291
26.0	Capacitive Sensing Module.....	319
27.0	Liquid Crystal Display (LCD) Driver Module.....	327
28.0	In-Circuit Serial Programming™ (ICSP™)	361
29.0	Instruction Set Summary	365
30.0	Electrical Specifications.....	379
31.0	DC and AC Characteristics Graphs and Charts	411
32.0	Development Support.....	439
33.0	Packaging Information.....	443
	Appendix A: Data Sheet Revision History.....	459
	Appendix B: Migrating From Other PIC® Devices.....	459
	Index	461
	The Microchip Web Site.....	469
	Customer Change Notification Service	469
	Customer Support.....	469
	Reader Response	470
	Product Identification System	471

TABLE 3-9: PIC16(L)F1936 MEMORY MAP, BANK 15

Bank 15	
791h	LCDCON
792h	LCDPS
793h	LCDREF
794h	LCDCST
795h	LCDRL
796h	—
797h	—
798h	LCDSE0
799h	LCDSE1
79Ah	—
79Bh	—
79Ch	—
79Dh	—
79Eh	—
79Fh	—
7A0h	LCDDATA0
7A1h	LCDDATA1
7A2h	—
7A3h	LCDDATA3
7A4h	LCDDATA4
7A5h	—
7A6h	LCDDATA6
7A7h	LCDDATA7
7A8h	—
7A9h	LCDDATA9
7AAh	LCDDATA10
7ABh	—
7ACh	—
7ADh	—
7AEh	—
7AFh	—
7B0h	—
7B1h	—
7B2h	—
7B3h	—
7B4h	—
7B5h	—
7B6h	—
7B7h	—
7B8h	Unimplemented Read as '0'
7EFh	

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

TABLE 3-10: PIC16(L)F1934/7 MEMORY MAP, BANK 15

Bank 15	
791h	LCDCON
792h	LCDPS
793h	LCDREF
794h	LCDCST
795h	LCDRL
796h	—
797h	—
798h	LCDSE0
799h	LCDSE1
79Ah	LCDSE2
79Bh	—
79Ch	—
79Dh	—
79Eh	—
79Fh	—
7A0h	LCDDATA0
7A1h	LCDDATA1
7A2h	LCDDATA2
7A3h	LCDDATA3
7A4h	LCDDATA4
7A5h	LCDDATA5
7A6h	LCDDATA6
7A7h	LCDDATA7
7A8h	LCDDATA8
7A9h	LCDDATA9
7AAh	LCDDATA10
7ABh	LCDDATA11
7ACh	—
7ADh	—
7AEh	—
7AFh	—
7B0h	—
7B1h	—
7B2h	—
7B3h	—
7B4h	—
7B5h	—
7B6h	—
7B7h	—
7B8h	Unimplemented Read as '0'
7EFh	

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

TABLE 3-12: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets	
Banks 16-30												
x00h/ x80h ⁽²⁾	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x00h/ x81h ⁽²⁾	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
x02h/ x82h ⁽²⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
x03h/ x83h ⁽²⁾	STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	---1 1000	---q quuu	
x04h/ x84h ⁽²⁾	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
x05h/ x85h ⁽²⁾	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
x06h/ x86h ⁽²⁾	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
x07h/ x87h ⁽²⁾	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
x08h/ x88h ⁽²⁾	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
x09h/ x89h ⁽²⁾	WREG	Working Register								0000 0000	uuuu uuuu	
x0Ah/ x8Ah ^{(1),(2)}	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
x0Bh/ x8Bh ⁽²⁾	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000	
x0Ch/ x8Ch — x1Fh/ x9Fh	—	Unimplemented								—	—	

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

- Note** 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.
- 2: These registers can be addressed from any bank.
- 3: These registers/bits are not implemented on PIC16(L)F1936 devices, read as '0'.
- 4: Unimplemented, read as '1'.

PIC16(L)F1934/6/7

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

PIC16(L)F1934/6/7

REGISTER 18-2: 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
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 18-3: 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

REGISTER 23-5: PWMxCON: ENHANCED PWM CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PxRSEN	PxDC<6:0>						
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 **PxRSEN:** PWM Restart Enable bit
1 = Upon auto-shutdown, the CCPxASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically
0 = Upon auto-shutdown, CCPxASE must be cleared in software to restart the PWM
- bit 6-0 **PxDC<6:0>:** PWM Delay Count bits
PxDCx = Number of Fosc/4 (4 * Tosc) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it transitions active

Note 1: Bit resets to '0' with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.

24.5.3 SLAVE TRANSMISSION

When the $\overline{R/W}$ bit of the incoming address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register, and an \overline{ACK} pulse is sent by the slave on the ninth bit.

Following the \overline{ACK} , slave hardware clears the CKP bit and the SCL pin is held low (see **Section 24.5.6 “Clock Stretching”** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then the SCL pin should be released by setting the CKP bit of the SSPCON1 register. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The \overline{ACK} pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. This \overline{ACK} value is copied to the ACKSTAT bit of the SSPCON2 register. If ACKSTAT is set (not \overline{ACK}), then the data transfer is complete. In this case, when the not \overline{ACK} is latched by the slave, the slave goes Idle and waits for another occurrence of the Start bit. If the SDA line was low (\overline{ACK}), the next transmit data must be loaded into the SSPBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared by software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

24.5.3.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit of the SSPCON3 register is set, the BCLIF bit of the PIR register is set. Once a bus collision is detected, the slave goes Idle and waits to be addressed again. User software can use the BCLIF bit to handle a slave bus collision.

24.5.3.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 24-17 can be used as a reference to this list.

1. Master sends a Start condition on SDA and SCL.
2. S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with $\overline{R/W}$ bit set is received by the Slave setting SSPIF bit.
4. Slave hardware generates an \overline{ACK} and sets SSPIF.
5. SSPIF bit is cleared by user.
6. Software reads the received address from SSPBUF, clearing BF.
7. $\overline{R/W}$ is set so CKP was automatically cleared after the \overline{ACK} .
8. The slave software loads the transmit data into SSPBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPIF is set after the \overline{ACK} response from the master is loaded into the ACKSTAT register.
11. SSPIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master \overline{ACK} s the clock will be stretched.

2: ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not \overline{ACK} ; the clock is not held, but SSPIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

24.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSPCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

24.5.6.1 Normal Clock Stretching

Following an $\overline{\text{ACK}}$ if the $\text{R}\overline{\text{W}}$ bit of SSPSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPBUF with data to transfer to the master. If the SEN bit of SSPCON2 is set, the slave hardware will always stretch the clock after the ACK sequence. Once the slave is ready, CKP is set by software and communication resumes.

Note 1: The BF bit has no effect on whether the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPBUF was read before the 9th falling edge of SCL.

2: Previous versions of the module did not stretch the clock for a transmission if SSPBUF was loaded before the 9th falling edge of SCL. It is now always cleared for read requests.

24.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set, the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPADD.

Note: Previous versions of the module did not stretch the clock if the second address byte did not match.

24.5.6.3 Byte NACKing

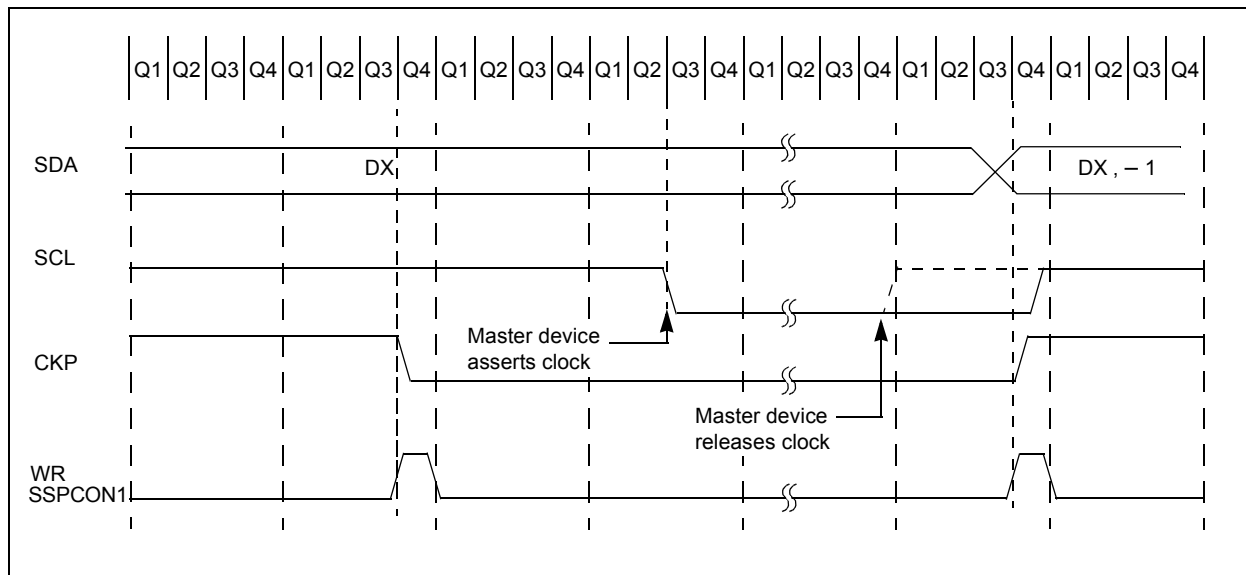
When the AHEN bit of SSPCON3 is set; CKP is cleared by hardware after the 8th falling edge of SCL for a received matching address byte. When the DHEN bit of SSPCON3 is set; CKP is cleared after the 8th falling edge of SCL for received data.

Stretching after the 8th falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

24.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I²C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I²C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 24-22).

FIGURE 24-23: CLOCK SYNCHRONIZATION TIMING



PIC16(L)F1934/6/7

REGISTER 25-3: BAUDCON: BAUD RATE CONTROL REGISTER

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

Legend:

R = Readable bit

W = Writable bit

U = 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 **ABDOVF:** Auto-Baud Detect Overflow bit

Asynchronous mode:

1 = Auto-baud timer overflowed

0 = Auto-baud timer did not overflow

Synchronous mode:

Don't care

bit 6 **RCIDL:** Receive Idle Flag bit

Asynchronous mode:

1 = Receiver is Idle

0 = Start bit has been received and the receiver is receiving

Synchronous mode:

Don't care

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

bit 4 **SCKP:** Synchronous Clock Polarity Select bit

Asynchronous mode:

1 = Transmit inverted data to the TX/CK pin

0 = Transmit non-inverted data to the TX/CK pin

Synchronous mode:

1 = Data is clocked on rising edge of the clock

0 = Data is clocked on falling edge of the clock

bit 3 **BRG16:** 16-bit Baud Rate Generator bit

1 = 16-bit Baud Rate Generator is used

0 = 8-bit Baud Rate Generator is used

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

bit 1 **WUE:** Wake-up Enable bit

Asynchronous mode:

1 = Receiver is waiting for a falling edge. No character will be received, byte RCIF will be set. WUE will automatically clear after RCIF is set.

0 = Receiver is operating normally

Synchronous mode:

Don't care

bit 0 **ABDEN:** Auto-Baud Detect Enable bit

Asynchronous mode:

1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)

0 = Auto-Baud Detect mode is disabled

Synchronous mode:

Don't care

PIC16(L)F1934/6/7

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

If CPSON = 0:

These bits are ignored. No channel is selected.

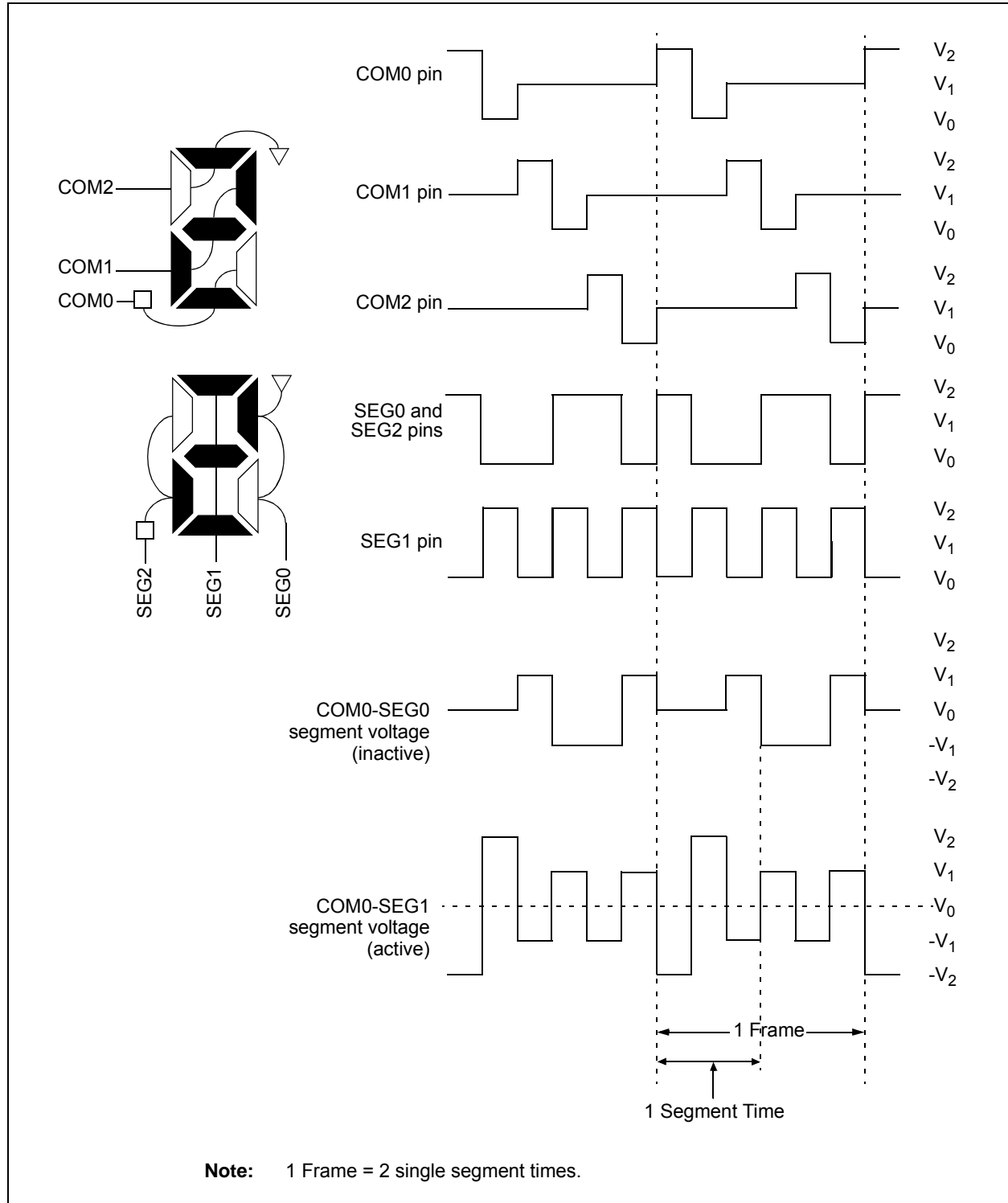
If CPSON = 1:

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.

Note 2: This bit is not implemented on PIC16(L)F1936, read as '0'

FIGURE 27-13: TYPE-A WAVEFORMS IN 1/3 MUX, 1/2 BIAS DRIVE



27.12 Configuring the LCD Module

The following is the sequence of steps to configure the LCD module.

1. Select the frame clock prescale using bits LP<3:0> of the LCDPS register.
2. Configure the appropriate pins to function as segment drivers using the LCDSEn registers.
3. Configure the LCD module for the following using the LCDCON register:
 - Multiplex and Bias mode, bits LMUX<1:0>
 - Timing source, bits CS<1:0>
 - Sleep mode, bit SLPEN
4. Write initial values to pixel data registers, LCDDATA0 through LCDDATA11.
5. Clear LCD Interrupt Flag, LCDIF bit of the PIR2 register and if desired, enable the interrupt by setting bit LCDIE of the PIE2 register.
6. Configure bias voltages by setting the LCDRL, LCDREF and the associated ANSELx registers as needed.
7. Enable the LCD module by setting bit LCDEN of the LCDCON register.

27.13 Disabling the LCD Module

To disable the LCD module, write all '0's to the LCDCON register.

27.14 LCD Current Consumption

When using the LCD module the current consumption consists of the following three factors:

- Oscillator Selection
- LCD Bias Source
- Capacitance of the LCD segments

The current consumption of just the LCD module can be considered negligible compared to these other factors.

27.14.1 OSCILLATOR SELECTION

The current consumed by the clock source selected must be considered when using the LCD module. See the applicable Electrical Specifications Chapter for oscillator current consumption information.

27.14.2 LCD BIAS SOURCE

The LCD bias source, internal or external, can contribute significantly to the current consumption. Use the highest possible resistor values while maintaining contrast to minimize current.

27.14.3 CAPACITANCE OF THE LCD SEGMENTS

The LCD segments which can be modeled as capacitors which must be both charged and discharged every frame. The size of the LCD segment and its technology determines the segment's capacitance.

28.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP™ programming allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP™ programming:

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- VSS

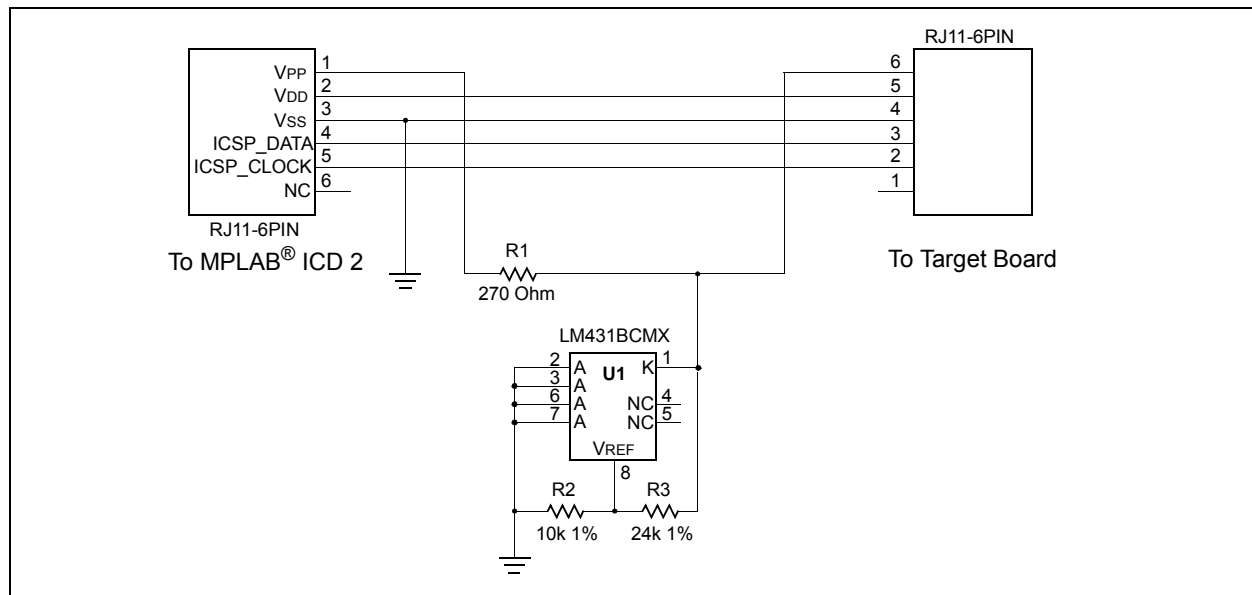
In Program/Verify mode the Program Memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. For more information on ICSP™ refer to the “PIC16193X/PIC16LF193X Memory Programming Specification” (DS41360).

28.1 High-Voltage Programming Entry Mode

The device is placed into High-Voltage Programming Entry mode by holding the ICSPCLK and ICSPDAT pins low then raising the voltage on MCLR/VPP to V_{IH} .

Some programmers produce V_{PP} greater than V_{IH} (9.0V), an external circuit is required to limit the V_{PP} voltage. See Figure 28-1 for example circuit.

FIGURE 28-1: VPP LIMITER EXAMPLE CIRCUIT



Note: The MPLAB® ICD 2 produces a V_{PP} voltage greater than the maximum V_{PP} specification of the PIC16(L)F1934/6/7.

PIC16(L)F1934/6/7

MOVIW Move INDFn to W

Syntax: [*label*] MOVIW ++FSRn
[*label*] MOVIW --FSRn
[*label*] MOVIW FSRn++
[*label*] MOVIW FSRn--
[*label*] MOVIW k[FSRn]

Operands: $n \in [0,1]$
 $mm \in [00,01, 10, 11]$
 $-32 \leq k \leq 31$

Operation: INDFn \rightarrow W
Effective address is determined by

- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:

- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap around.

MOVLB Move literal to BSR

Syntax: [*label*] MOVLB k

Operands: $0 \leq k \leq 15$

Operation: $k \rightarrow$ BSR

Status Affected: None

Description: The five-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVL P Move literal to PCLATH

Syntax: [*label*] MOVL P k

Operands: $0 \leq k \leq 127$

Operation: $k \rightarrow$ PCLATH

Status Affected: None

Description: The seven-bit literal 'k' is loaded into the PCLATH register.

MOVLW Move literal to W

Syntax: [*label*] MOVLW k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow$ (W)

Status Affected: None

Description: The eight-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

Words: 1

Cycles: 1

Example: MOVLW 0x5A
After Instruction
W = 0x5A

MOVWF Move W to f

Syntax: [*label*] MOVWF f

Operands: $0 \leq f \leq 127$

Operation: (W) \rightarrow (f)

Status Affected: None

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example: MOVWF OPTION_REG
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

PIC16(L)F1934/6/7

FIGURE 30-1: PIC16F1934/36/37 VOLTAGE FREQUENCY GRAPH, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$

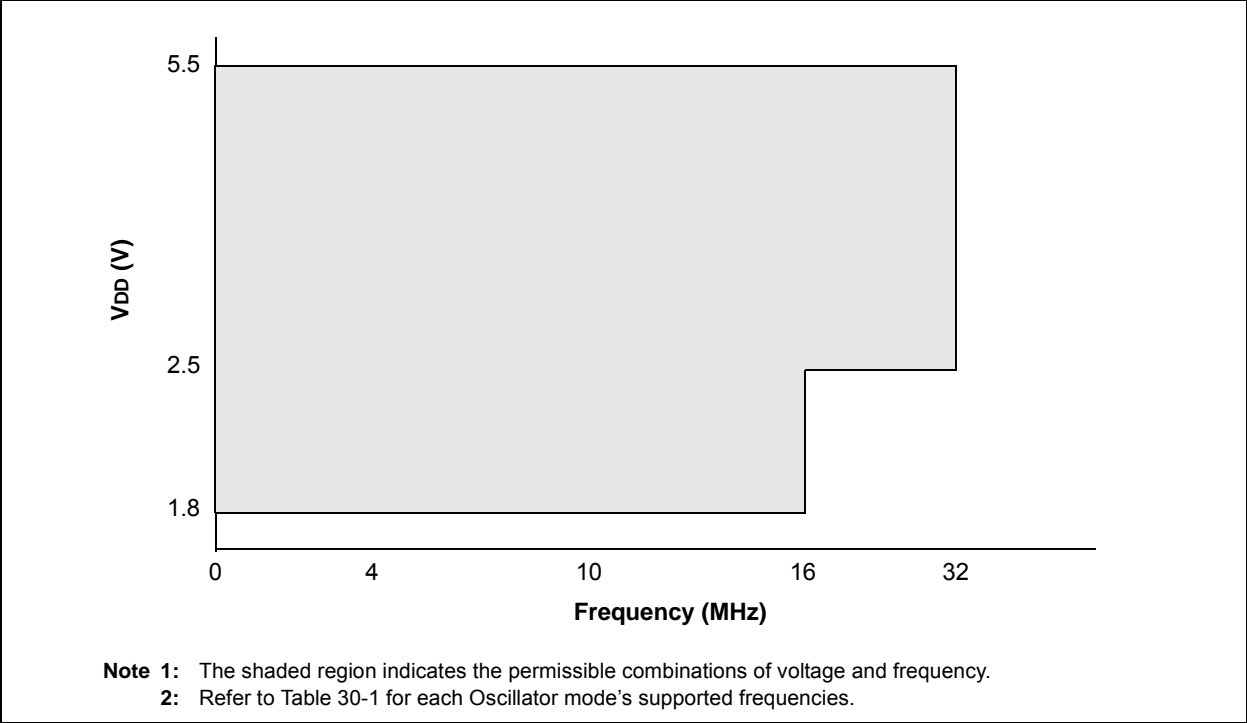
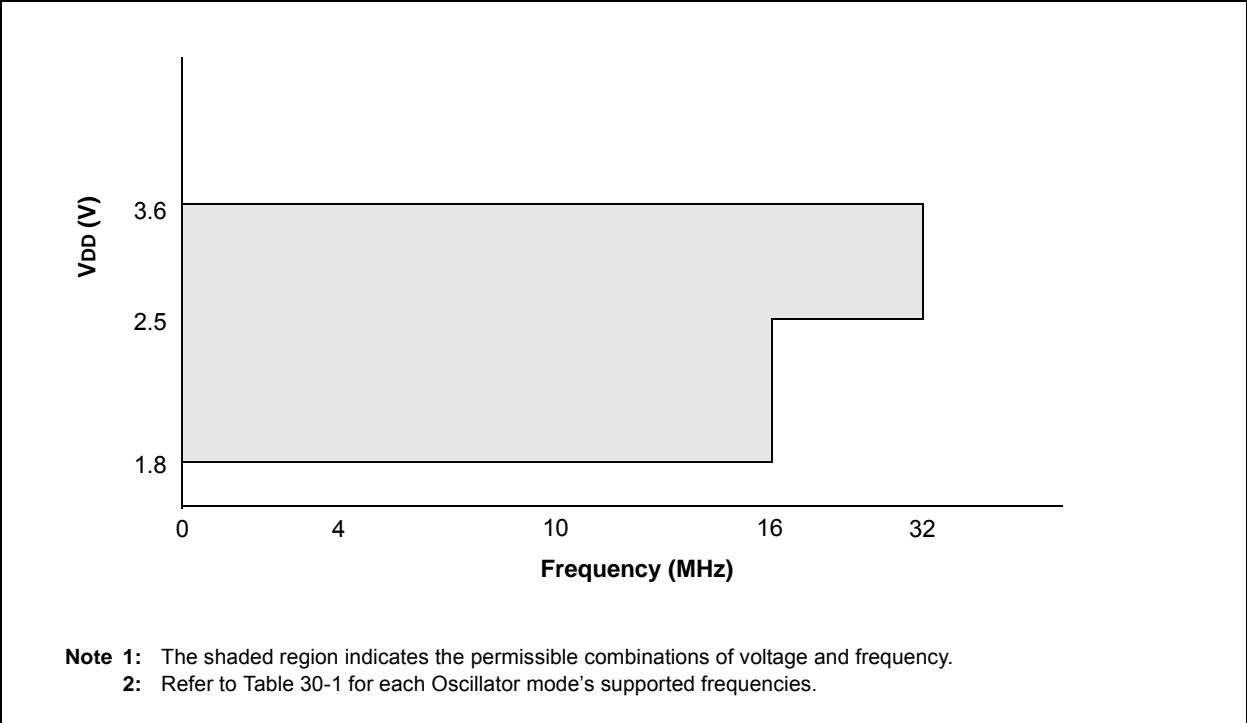


FIGURE 30-2: PIC16LF1934/36/37 VOLTAGE FREQUENCY GRAPH, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$



PIC16(L)F1934/6/7

30.4 DC Characteristics: PIC16(L)F1934/6/7-I/E (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
D102 D102A		V_{CAP} Capacitor Charging					
		Charging current	—	200	—	μA	
		Source/sink capability when charging complete	—	0.0	—	mA	

* 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:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.
- 2:** Negative current is defined as current sourced by the pin.
- 3:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 4:** Including OSC2 in CLKOUT mode.

PIC16(L)F1934/6/7

FIGURE 31-29: PIC16LF1937 EXTRC MODE, Fosc = 4 MHz

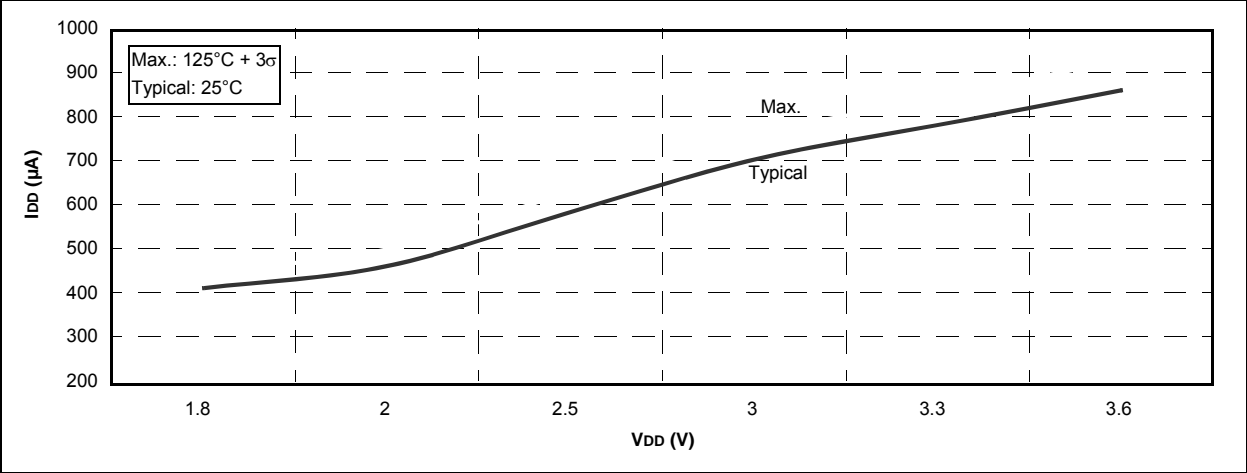


FIGURE 31-30: PIC16LF1937 XT OSCILLATOR, Fosc = 1 MHz

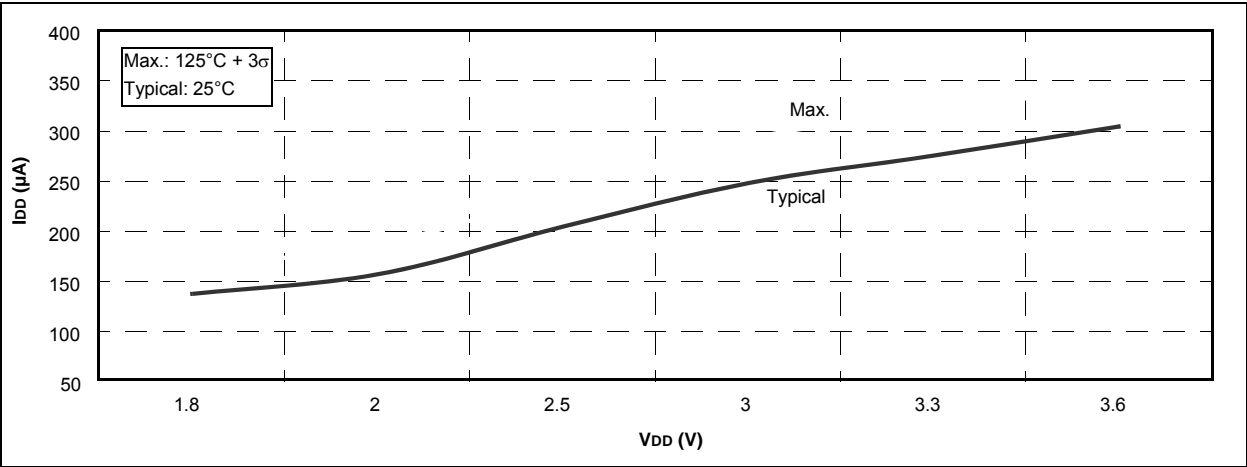
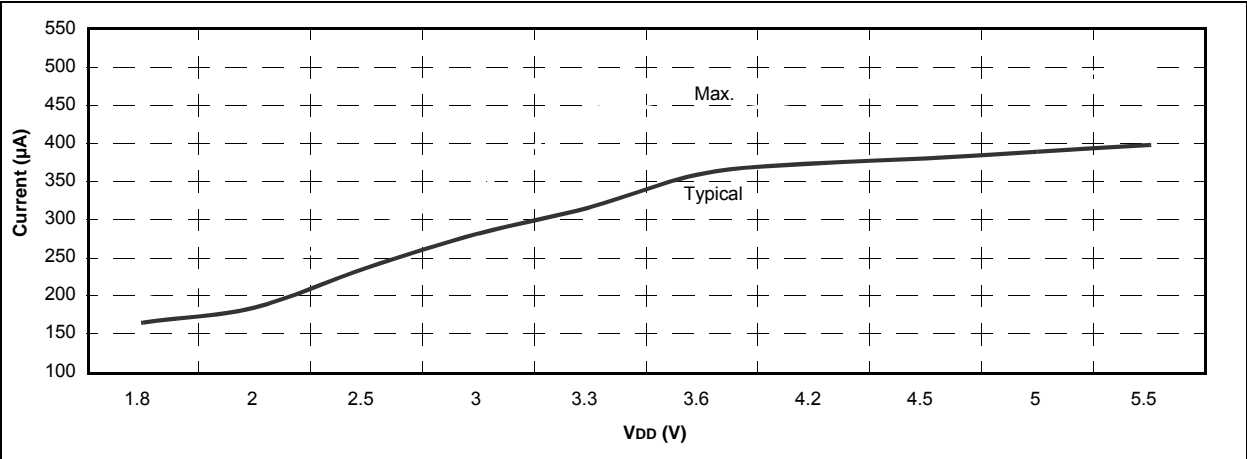


FIGURE 31-31: PIC16F1937 XT OSCILLATOR, Fosc = 1 MHz



32.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

32.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

32.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

32.10 PICkit 3 In-Circuit Debugger/Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

PIC16(L)F1934/6/7

PWM Setup	219
CCP1CON Register	44, 45
CCPR1H Register	44, 45
CCPR1L Register	44, 45
CCPTMRS0 Register	237
CCPTMRS1 Register	237
CCPxAS Register	238
CCPxCON (ECCPx) Register	236
Clock Accuracy with Asynchronous Operation	302
Clock Sources	
External Modes	71
EC	71
HS	71
LP	71
OST	72
RC	73
XT	71
Internal Modes	74
HFINTOSC	74
Internal Oscillator Clock Switch Timing	76
LFINTOSC	75
MFINTOSC	74
Clock Switching	78
CMOUT Register	186
CMxCON0 Register	185
CMxCON1 Register	186
Code Examples	
A/D Conversion	164
Changing Between Capture Prescalers	214
Initializing PORTA	131
Initializing PORTE	150
Write Verify	127
Writing to Flash Program Memory	125
Comparator	
Associated Registers	187, 188
Operation	179
Comparator Module	179
Cx Output State Versus Input Conditions	181
Comparator Specifications	405
Comparators	
C2OUT as T1 Gate	199
Compare Module. See Enhanced Capture/Compare/ PWM (ECCP)	
CONFIG1 Register	62
CONFIG2 Register	64
Core Registers	39
CPSCON0 Register	325
CPSCON1 Register	326
Customer Change Notification Service	471
Customer Notification Service	471
Customer Support	471

D

DACCON0 (Digital-to-Analog Converter Control 0) Register	178
DACCON1 (Digital-to-Analog Converter Control 1) Register	178
Memory	117
Associated Registers	130
Code Protection	118
Reading	118
Writing	118
Data Memory	28, 31
DC and AC Characteristics	413
DC Characteristics	
Extended and Industrial (PIC16F/LF1934/36/37)	391

Industrial and Extended (PIC16F/LF1934/36/37)	384
Development Support	441
Device Configuration	61
Code Protection	65
Configuration Word	61
User ID	65, 66
Device Overview	15, 113
Digital-to-Analog Converter (DAC)	175
Associated Registers	178
Effects of a Reset	176
Specifications	405

E

ECCP/CCP. See Enhanced Capture/Compare/PWM	
EEADR Registers	117
EEADRH Registers	117
EEADRL Register	128
EEADRL Registers	117
EECON1 Register	117, 129
EECON2 Register	117, 130
EEDATH Register	128
EEDATL Register	128
EEPROM Data Memory	
Avoiding Spurious Write	118
Write Verify	127
Effects of Reset	
PWM mode	221
Electrical Specifications (PIC16F/LF1934/36/37)	381
Enhanced Capture/Compare/PWM (ECCP)	213
Enhanced PWM Mode	222
Auto-Restart	231
Auto-shutdown	230
Direction Change in Full-Bridge Output Mode	228
Full-Bridge Application	226
Full-Bridge Mode	226
Half-Bridge Application	225
Half-Bridge Application Examples	232
Half-Bridge Mode	225
Output Relationships (Active-High and Active-Low)	223
Output Relationships Diagram	224
Programmable Dead Band Delay	232
Shoot-through Current	232
Start-up Considerations	234
Specifications	402
Enhanced Mid-range CPU	23
Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)	293
Errata	14
EUSART	293
Associated Registers	
Baud Rate Generator	306
Asynchronous Mode	295
12-bit Break Transmit and Receive	313
Associated Registers	
Receive	301
Transmit	297
Auto-Wake-up on Break	311
Baud Rate Generator (BRG)	305
Clock Accuracy	302
Receiver	298
Setting up 9-bit Mode with Address Detect	300
Transmitter	295
Baud Rate Generator (BRG)	
Auto Baud Rate Detect	310
Baud Rate Error, Calculating	305

PIC16(L)F1934/6/7

WDTCN (Watchdog Timer Control).....	115	TMR1L Register.....	197
WPUB (Weak Pull-up PORTB)	141	Timer2	
RESET	377	Associated registers	212
Reset.....	87	Timer2/4/6.....	209
Reset Instruction	90	Associated registers	212
Resets	87	Timers	
Associated Registers	94	Timer1	
Revision History	461	T1CON	205
S		T1GCON.....	206
Shoot-through Current	232	Timer2/4/6	
Software Simulator (MPLAB SIM).....	443	TXCON	211
SPBRG Register	41, 42	Timing Diagrams	
SPBRGH	305	A/D Conversion.....	404
SPBRGL.....	305	A/D Conversion (Sleep Mode).....	404
Special Event Trigger.....	163	Acknowledge Sequence	280
Special Function Registers (SFRs).....	39	Asynchronous Reception.....	300
SPI Mode (MSSP)		Asynchronous Transmission.....	296
Associated Registers	251	Asynchronous Transmission (Back to Back)	296
SPI Clock	247	Auto Wake-up Bit (WUE) During Normal Operation ..	312
SR Latch	189	Auto Wake-up Bit (WUE) During Sleep	312
Associated registers w/ SR Latch	192	Automatic Baud Rate Calibration.....	310
SRCON0 Register.....	191	Baud Rate Generator with Clock Arbitration	273
SRCON1 Register.....	192	BRG Reset Due to SDA Arbitration During Start	
SSPAD Register.....	43, 292	Condition	283
SSPBUF Register	43	Brown-out Reset (BOR).....	400
SSPCON Register.....	43	Brown-out Reset Situations	89
SSPCON1 Register.....	289	Bus Collision During a Repeated Start Condition	
SSPCON2 Register.....	290	(Case 1).....	284
SSPCON3 Register.....	291	Bus Collision During a Repeated Start Condition	
SSPMSK Register.....	292	(Case 2).....	284
SSPOV	278	Bus Collision During a Start Condition (SCL = 0)	283
SSPOV Status Flag.....	278	Bus Collision During a Stop Condition (Case 1)	285
SSPSTAT Register	43, 288	Bus Collision During a Stop Condition (Case 2)	285
R/W Bit	257	Bus Collision During Start Condition (SDA only)	282
Stack	54	Bus Collision for Transmit and Acknowledge	281
Accessing.....	54	CLKOUT and I/O	398
Reset.....	56	Clock Synchronization	270
Stack Overflow/Underflow	90	Clock Timing	396
STATUS Register.....	29	Comparator Output.....	179
SUBWFB.....	379	Enhanced Capture/Compare/PWM (ECCP).....	402
T		Fail-Safe Clock Monitor (FSCM).....	82
T1CON Register.....	39, 205	First Start Bit Timing	274
T1GCON Register.....	206	Full-Bridge PWM Output.....	227
T2CON Register.....	39, 47	Half-Bridge PWM Output	225, 232
Temperature Indicator Module	173	I ² C Bus Data.....	410
Thermal Considerations (PIC16F/LF1934/36/37)	394	I ² C Bus Start/Stop Bits	409
Timer0	193	I ² C Master Mode (7-Bit Reception).....	277, 279
Associated Registers	195	I ² C Stop Condition Receive or Transmit Mode.....	280
Operation	193	INT Pin Interrupt	98
Specifications	402	Internal Oscillator Switch Timing	77
Timer1	197	LCD Interrupt Timing in Quarter-Duty Cycle Drive ...	358
Associated registers.....	207	LCD Sleep Entry/Exit when SLPEN = 1 or CS = 00 ..	360
Asynchronous Counter Mode	199	PWM Auto-shutdown	231
Reading and Writing	199	Firmware Restart	230
Clock Source Selection	198	PWM Direction Change	228
Interrupt.....	201	PWM Direction Change at Near 100% Duty Cycle... ..	229
Operation	198	PWM Output (Active-High)	223
Operation During Sleep	201	PWM Output (Active-Low)	224
Oscillator	199	Repeat Start Condition	275
Prescaler.....	199	Reset Start-up Sequence	91
Specifications.....	402	Reset, WDT, OST and Power-up Timer	399
Timer1 Gate		Send Break Character Sequence	313
Selecting Source.....	199	SPI Master Mode (CKE = 1, SMP = 1)	407
TMR1H Register	197	SPI Mode (Master Mode).....	247
		SPI Slave Mode (CKE = 0).....	408