



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I <sup>2</sup> 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	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1936-e-so

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

#### Pin Diagram – 28-Pin SPDIP/SOIC/SSOP (PIC16(L)F1936)



#### 11.3.2 ERASING FLASH PROGRAM MEMORY

While executing code, program memory can only be erased by rows. To erase a row:

- 1. Load the EEADRH:EEADRL register pair with the address of new row to be erased.
- 2. Clear the CFGS bit of the EECON1 register.
- 3. Set the EEPGD, FREE, and WREN bits of the EECON1 register.
- 4. Write 55h, then AAh, to EECON2 (Flash programming unlock sequence).
- 5. Set control bit WR of the EECON1 register to begin the erase operation.
- 6. Poll the FREE bit in the EECON1 register to determine when the row erase has completed.

See Example 11-4.

After the "BSF EECON1, WR" instruction, the processor requires two cycles to set up the erase operation. The user must place two NOP instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the EECON1 write instruction.

## 11.3.3 WRITING TO FLASH PROGRAM MEMORY

Program memory is programmed using the following steps:

- 1. Load the starting address of the word(s) to be programmed.
- 2. Load the write latches with data.
- 3. Initiate a programming operation.
- 4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 11-2 (block writes to program memory with 8 write latches) for more details. The write latches are aligned to the address boundary defined by EEADRL as shown in Table 11-1. Write operations do not cross these boundaries. At the completion of a program memory write operation, the write latches are reset to contain 0x3FFF. The following steps should be completed to load the write latches and program a block of program memory. These steps are divided into two parts. First, all write latches are loaded with data except for the last program memory location. Then, the last write latch is loaded and the programming sequence is initiated. A special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. This unlock sequence should not be interrupted.

- 1. Set the EEPGD and WREN bits of the EECON1 register.
- 2. Clear the CFGS bit of the EECON1 register.
- Set the LWLO bit of the EECON1 register. When the LWLO bit of the EECON1 register is '1', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
- 4. Load the EEADRH:EEADRL register pair with the address of the location to be written.
- 5. Load the EEDATH:EEDATL register pair with the program memory data to be written.
- Write 55h, then AAh, to EECON2, then set the WR bit of the EECON1 register (Flash programming unlock sequence). The write latch is now loaded.
- 7. Increment the EEADRH:EEADRL register pair to point to the next location.
- 8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
- Clear the LWLO bit of the EECON1 register. When the LWLO bit of the EECON1 register is '0', the write sequence will initiate the write to Flash program memory.
- 10. Load the EEDATH:EEDATL register pair with the program memory data to be written.
- 11. Write 55h, then AAh, to EECON2, then set the WR bit of the EECON1 register (Flash programming unlock sequence). The entire latch block is now written to Flash program memory.

It is not necessary to load the entire write latch block with user program data. However, the entire write latch block will be written to program memory.

An example of the complete write sequence for eight words is shown in Example 11-5. The initial address is loaded into the EEADRH:EEADRL register pair; the eight words of data are loaded using indirect addressing.

Note: The code sequence provided in Example 11-5 must be repeated multiple times to fully program an erased program memory row.

REGISTER 11-6:	EECON2: EEPROM CONTROL 2 REGISTER
----------------	-----------------------------------

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
			EEPROM Co	ntrol Register 2			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplen	nented bit, read	as '0'	
S = Bit can onl	ly be set	x = Bit is unkr	nown	-n/n = Value a	t POR and BO	R/Value at all o	ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				

#### bit 7-0 Data EEPROM Unlock Pattern bits

To unlock writes, a 55h must be written first, followed by an AAh, before setting the WR bit of the EECON1 register. The value written to this register is used to unlock the writes. There are specific timing requirements on these writes. Refer to **Section 11.2.2** "Writing to the Data EEPROM Memory" for more information.

#### TABLE 11-3: SUMMARY OF REGISTERS ASSOCIATED WITH DATA EEPROM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
EECON1	EEPGD	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	127
EECON2	EEPROM Control Register 2 (not a physical register)								115*
EEADRL	EEADRL<7:0>							126	
EEADRH	—	EEADRH<6:0							126
EEDATL	EEDATL<7:0>							126	
EEDATH	—	—			EEDAT	H<5:0>			126
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	98
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	100
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	103

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the data EEPROM module.

\* Page provides register information.

#### REGISTER 12-6: PORTB: PORTB REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	
bit 7							bit 0	
Legend:								
R = Readable I	oit	W = Writable	bit	U = Unimplemented bit, read as '0'				
u = Bit is unchanged x = Bit is unknown		iown	-n/n = Value at POR and BOR/Value at all other Resets					
'1' = Bit is set		'0' = Bit is clea	ared					

bit 7-0 **RB<7:0>**: PORTB I/O Pin bit 1 = Port pin is > VIH 0 = Port pin is < VIL

#### REGISTER 12-7: TRISB: PORTB TRI-STATE REGISTER

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TRISB7  | TRISB6  | TRISB5  | TRISB4  | TRISB3  | TRISB2  | TRISB1  | TRISB0  |
| 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 7-0 **TRISB<7:0>:** PORTB Tri-State Control bit 1 = PORTB pin configured as an input (tri-stated)

0 = PORTB pin configured as an output

#### REGISTER 12-8: LATB: PORTB DATA LATCH REGISTER

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| LATB7   | LATB6   | LATB5   | LATB4   | LATB3   | LATB2   | LATB1   | LATB0   |
| 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 7-0 LATB<7:0>: PORTB Output Latch Value bits<sup>(1)</sup>

**Note 1:** Writes to PORTB are actually written to corresponding LATB register. Reads from PORTB register is return of actual I/O pin values.

#### REGISTER 12-11: PORTC: PORTC REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
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 clea	ared				

bit 7-0 RC<7:0>: PORTC General Purpose I/O Pin bits 1 = Port pin is > VIH 0 = Port pin is < VIL

#### REGISTER 12-12: TRISC: PORTC TRI-STATE REGISTER

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TRISC7  | TRISC6  | TRISC5  | TRISC4  | TRISC3  | TRISC2  | TRISC1  | TRISC0  |
| 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 7-0 **TRISC<7:0>:** PORTC Tri-State Control bits 1 = PORTC pin configured as an input (tri-stated)

0 = PORTC pin configured as an output

#### REGISTER 12-13: LATC: PORTC DATA LATCH REGISTER

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| LATC7   | LATC6   | LATC5   | LATC4   | LATC3   | LATC2   | LATC1   | LATC0   |
| 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		
1' = Bit is set	'0' = Bit is cleared		

bit 7-0 LATC<7:0>: PORTC Output Latch Value bits<sup>(1)</sup>

**Note 1:** Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.

NOTES:



#### FIGURE 18-2: COMPARATOR MODULE SIMPLIFIED BLOCK DIAGRAM

NOTES:

### 21.12 Timer1 Gate Control Register

The Timer1 Gate Control register (T1GCON), shown in Register 21-2, is used to control Timer1 gate.

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS	6<1:0>
bit 7		I	L			I	bit 0
Legend:							
R = Readable bit		W = Writable bit		U = Unimplemented bit, read as '0'			
u = Bit is unch	anged	x = Bit is unknown		-n/n = Value at POR and BOR/Value at all other Resets			
'1' = Bit is set		'0' = Bit is cleared		HC = Bit is cleared by hardware			
bit 7 <b>TMR1GE:</b> Timer1 Gate Enable bit If TMR1ON = 0: This bit is ignored If TMR1ON = 1: 1 = Timer1 counting is controlled by the Timer1 gate function							
bit 6	<b>T1GPOL:</b> Timer1 Gate Polarity bit 1 = Timer1 gate is active-high (Timer1 counts when gate is high) 0 = Timer1 gate is active-low (Timer1 counts when gate is low)						
bit 5	bit 5 <b>T1GTM:</b> Timer1 Gate Toggle Mode bit 1 = Timer1 Gate Toggle mode is enabled 0 = Timer1 Gate Toggle mode is disabled and toggle flip flop is cleared Timer1 gate flip-flop toggles on every rising edge						
bit 4	<b>T1GSPM:</b> Timer1 Gate Single-Pulse Mode bit						
	<ul> <li>1 = Timer1 gate Single-Pulse mode is enabled and is controlling Timer1 gate</li> <li>0 = Timer1 gate Single-Pulse mode is disabled</li> </ul>						
bit 3	<b>T1GGO/DONE:</b> Timer1 Gate Single-Pulse Acquisition Status bit						
	<ul> <li>1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge</li> <li>0 = Timer1 gate single-pulse acquisition has completed or has not been started</li> </ul>						
bit 2	T1GVAL: Timer1 Gate Current State bit						
	Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L. Unaffected by Timer1 Gate Enable (TMR1GE).						
bit 1-0	T1GSS<1:0>	: Timer1 Gate	Source Select	bits			
	<ul> <li>00 = Timer1 gate pin</li> <li>01 = Timer0 overflow output</li> <li>10 = Comparator 1 optionally synchronized output (SYNCC1OUT)</li> <li>11 = Comparator 2 optionally synchronized output (SYNCC2OUT)</li> </ul>						

#### REGISTER 21-2: T1GCON: TIMER1 GATE CONTROL REGISTER

#### 23.4.5 PROGRAMMABLE DEAD-BAND DELAY MODE

In Half-Bridge applications where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on, and the other turned off), both switches may be on for a short period of time until one switch completely turns off. During this brief interval, a very high current (*shoot-through current*) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. See Figure 23-16 for illustration. The lower seven bits of the associated PWMxCON register (Register 23-5) sets the delay period in terms of microcontroller instruction cycles (TcY or 4 Tosc).

#### FIGURE 23-16: EXAMPLE OF HALF-BRIDGE PWM OUTPUT



#### FIGURE 23-17: EXAMPLE OF HALF-BRIDGE APPLICATIONS



#### 24.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 24-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and Status bits appropriately set). The clock polarity is selected by appropriately programming the CKP bit of the SSPCON1 register and the CKE bit of the SSPSTAT register. This then, would give waveforms for SPI communication as shown in Figure 24-6, Figure 24-8 and Figure 24-9, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- Fosc/4 (or Tcy)
- Fosc/16 (or 4 \* Tcy)
- Fosc/64 (or 16 \* Tcy)
- Timer2 output/2
- Fosc/(4 \* (SSPADD + 1))

Figure 24-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 24-6: SPI MODE WAVEFORM (MASTER MODE)



### 24.3 I<sup>2</sup>C Mode Overview

The Inter-Integrated Circuit Bus  $(I^2C)$  is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A Slave device is controlled through addressing.

The I<sup>2</sup>C bus specifies two signal connections:

- · Serial Clock (SCL)
- Serial Data (SDA)

Figure 24-11 shows the block diagram of the MSSP module when operating in  $I^2C$  Mode.

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 24-11 shows a typical connection between two processors configured as master and slave devices.

The  $I^2C$  bus can operate with one or more master devices and one or more slave devices.

There are four potential modes of operation for a given device:

- Master Transmit mode (master is transmitting data to a slave)
- Master Receive mode
   (master is receiving data from a slave)
- Slave Transmit mode (slave is transmitting data to a master)
- Slave Receive mode (slave is receiving data from the master)

To begin communication, a master device starts out in Master Transmit mode. The master device sends out a Start bit followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an ACK. The master then continues in either Transmit mode or Receive mode and the slave continues in the complement, either in Receive mode or Transmit mode, respectively.

A Start bit is indicated by a high-to-low transition of the SDA line while the SCL line is held high. Address and data bytes are sent out, Most Significant bit (MSb) first. The Read/Write bit is sent out as a logical one when the master intends to read data from the slave, and is sent out as a logical zero when it intends to write data to the slave.

#### FIGURE 24-11: I<sup>2</sup>C MASTER/ SLAVE CONNECTION



The Acknowledge bit  $(\overline{ACK})$  is an active-low signal, which holds the SDA line low to indicate to the transmitter that the slave device has received the transmitted data and is ready to receive more.

The transition of a data bit is always performed while the SCL line is held low. Transitions that occur while the SCL line is held high are used to indicate Start and Stop bits.

If the master intends to write to the slave, then it repeatedly sends out a byte of data, with the slave responding after each byte with an  $\overline{ACK}$  bit. In this example, the master device is in Master Transmit mode and the slave is in Slave Receive mode.

If the master intends to read from the slave, then it repeatedly receives a byte of data from the slave, and responds after each byte with an ACK bit. In this example, the master device is in Master Receive mode and the slave is Slave Transmit mode.

On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends the Stop bit in place of the last ACK bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send another Start bit in place of the Stop bit or last ACK bit when it is in receive mode.

The I<sup>2</sup>C bus specifies three message protocols;

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.



- 25.1.2.8 Asynchronous Reception Set-up:
- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 25.3 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- 4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 5. If 9-bit reception is desired, set the RX9 bit.
- 6. Enable reception by setting the CREN bit.
- 7. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 8. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
- 9. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register.
- 10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

#### 25.1.2.9 9-bit Address Detection Mode Set-up

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 25.3 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 5. Enable 9-bit reception by setting the RX9 bit.
- 6. Enable address detection by setting the ADDEN bit.
- 7. Enable reception by setting the CREN bit.
- The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 9. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
- 10. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
- 11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
- 12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.



#### FIGURE 25-5: ASYNCHRONOUS RECEPTION

### 25.5 EUSART Operation During Sleep

The EUSART will remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore cannot generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

Synchronous Slave mode uses an externally generated clock to run the Transmit and Receive Shift registers.

#### 25.5.1 SYNCHRONOUS RECEIVE DURING SLEEP

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Reception (see Section 25.4.2.4 "Synchronous Slave Reception Set-up:").
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RCREG to unload any pending characters in the receive buffer.

Upon entering Sleep mode, the device will be ready to accept data and clocks on the RX/DT and TX/CK pins, respectively. When the data word has been completely clocked in by the external device, the RCIF interrupt flag bit of the PIR1 register will be set. Thereby, waking the processor from Sleep.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the Global Interrupt Enable (GIE) bit of the INTCON register is also set, then the Interrupt Service Routine at address 004h will be called.

#### 25.5.2 SYNCHRONOUS TRANSMIT DURING SLEEP

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

- RCSTA and TXSTA Control registers must be configured for Synchronous Slave Transmission (see Section 25.4.2.2 "Synchronous Slave Transmission Set-up:").
- The TXIF interrupt flag must be cleared by writing the output data to the TXREG, thereby filling the TSR and transmit buffer.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.
- Interrupt enable bits TXIE of the PIE1 register and PEIE of the INTCON register must set.

Upon entering Sleep mode, the device will be ready to accept clocks on TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TXREG will transfer to the TSR and the TXIF flag will be set. Thereby, waking the processor from Sleep. At this point, the TXREG is available to accept another character for transmission, which will clear the TXIF flag.

Upon waking from Sleep, the instruction following the SLEEP instruction will be executed. If the Global Interrupt Enable (GIE) bit is also set then the Interrupt Service Routine at address 0004h will be called.



### 28.0 IN-CIRCUIT SERIAL PROGRAMMING<sup>™</sup> (ICSP<sup>™</sup>)

ICSP<sup>™</sup> 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<sup>™</sup> 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<sup>TM</sup> 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 VIHH.

Some programmers produce VPP greater than VIHH (9.0V), an external circuit is required to limit the VPP voltage. See Figure 28-1 for example circuit.



FIGURE 28-1: VPP LIMITER EXAMPLE CIRCUIT

© 2008-2011 Microchip Technology Inc.







FIGURE 31-34: PIC16LF1937 EC OSCILLATOR, HIGH-POWER MODE, Fosc = 32 MHz



PWM Setup	
CCP1CON Register	
CCPR1H Register	
CCPR1L Register	44. 45
CCPTMRS0 Register	
CCPTMRS1 Register	
CCPxAS Register	
CCPxCON (ECCPx) Register	
Clock Accuracy with Asynchronous Operation	302
Clock Sources	
External Modes	71
FC	71
HS.	
IP	71
OST	72
RC	73
ХТ	71
Internal Modes	
HEINTOSC	
Internal Oscillator Clock Switch Timing	76
I FINTOSC	75
MEINTOSC	70
Clock Switching	78
CMOLIT Register	186
CMxCON0 Register	185
CMxCON1 Register	186
Code Examples	
A/D Conversion	164
Changing Between Capture Prescalers	214
Initializing PORTA	131
	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	e/
PWM (ECCP)	
CONFIG1 Register	
CONFIG2 Register	64
Core Registers	
CPSCON0 Register	
CPSCON1 Register	
Customer Change Notification Service	
Customer Notification Service	
Customer Support	
••	-

## 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

### Е

ECCP/CCP. See Enhanced Capture/Compare/PWM	
EEADR Registers	. 117
EEADRH Registers	. 117
EEADRL Register	. 128
EEADRL Registers	. 117
EECON1 Register	129
EECON2 Register	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 (PIC16E/LE103//36/37)	381
Enhanced Canture/Compare/DW/M (ECCP)	213
Enhanced Capture/Compare/T WM (LCCT )	210
	222
	231
Auto-shuldown	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	0.0
Receive	301
Transmit	297
Auto-Wake-up on Break	311
Baud Pate Generator (BPG)	305
Clock Accuracy	302
Receiver	202
Setting up 9-bit Mode with Address Datact	300
Transmitter	205
Baud Rate Concrator (PPC)	200
Auto Baud Bate Dotoot	310
Raud Rate Error Calculating	305
Daud Nate LITUI, Calculating	

### **PRODUCT IDENTIFICATION SYSTEM**

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO. Device	[X] <sup>(1)</sup> - X /XX XXX T Tape and Reel Temperature Package Pattern Option Range	<ul> <li>Examples:</li> <li>a) PIC16LF1937 - I/P = Industrial temp., Plastic DIP package, low-voltage VDD limits.</li> <li>b) PIC16F1934 - I/PT = Industrial temp., TQFP package, standard VDD limits.</li> </ul>
Device:	PIC16F1934, PIC16LF1934, PIC16F1934, PIC16LF1934 PIC16F1936, PIC16LF1936, PIC16F1936, PIC16LF1936 PIC16F1937, PIC16LF1937, PIC16F1937, PIC16LF1937	
Tape and Reel Option:	Blank = Standard packaging (tube or tray) T = Tape and Reel <sup>(1)</sup>	
Temperature Range:	$I = -40^{\circ}C \text{ to } +85^{\circ}C$ $E = -40^{\circ}C \text{ to } +125^{\circ}C$	
Package:	ML = Micro Lead Frame (QFN) MV = Micro Lead Frame (UQFN) 4x4 P = Plastic DIP PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic DIP SS = SSOP	Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
Pattern:	3-Digit Pattern Code for QTP (blank otherwise)	