



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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	32
Program Memory Size	24KB (12K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1408 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4539t-i-ml

Email: info@E-XFL.COM

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

PIC18FXX39

TABLE 1-2:	PIC18F2X39 PINOUT I/O DESCRIPTIONS	(CONTINUED)
------------	------------------------------------	-------------

Din Nome	Pin Number		Pin Buffer		Description
Pin Name	DIP	SOIC	Туре	Туре	Description
					PORTC is a bi-directional I/O port.
RC0/T13CKI	11	11			
RC0			I/O	ST	Digital I/O.
T13CKI			I	ST	Timer1/Timer3 external clock input.
RC3/SCK/SCL	14	14			
RC3			I/O	ST	Digital I/O.
SCK			I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL			I/O	ST	Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA	15	15			
RC4			I/O	ST	Digital I/O.
SDI			I	ST	SPI Data in.
SDA			I/O	ST	I ² C Data I/O.
RC5/SDO	16	16			
RC5			I/O	ST	Digital I/O.
SDO			0	—	SPI Data out.
RC6/TX/CK	17	17			
RC6			I/O	ST	Digital I/O.
ТХ			0	—	USART Asynchronous Transmit.
СК			I/O	ST	USART Synchronous Clock (see related RX/DT).
RC7/RX/DT	18	18			
RC7			I/O	ST	Digital I/O.
RX			I	ST	USART Asynchronous Receive.
DT			I/O	ST	USART Synchronous Data (see related TX/CK).
PWM1	13	13	0	—	PWM Channel 1 (motor control) output.
PWM2	12	12	0	—	PWM Channel 2 (motor control) output.
Vss	8, 19	8, 19	Р	—	Ground reference for logic and I/O pins.
VDD	20	20	Р		Positive supply for logic and I/O pins.
Legend: TTL = TTL	compat	tible inp	ut		CMOS = CMOS compatible input or output

ST = Schmitt Trigger input with CMOS levels O = Output

I = Input = Power Ρ

OD = Open Drain (no P diode to VDD)

© 2002-2013 Microchip Technology Inc.





2.5 Effects of SLEEP Mode on the On-Chip Oscillator

When the device executes a SLEEP instruction, the oscillator is turned off and the device is held at the beginning of an instruction cycle (Q1 state). With the oscillator off, the OSC1 and OSC2 signals will stop oscillating. Since all the transistor switching currents have been removed, SLEEP mode achieves the lowest current consumption of the device (only leakage currents). Enabling any on-chip feature that will operate during SLEEP will increase the current consumed during SLEEP. The user can wake from SLEEP through external RESET, Watchdog Timer Reset, or through an interrupt.

2.6 Power-up Delays

Power-up delays are controlled by two timers, so that no external RESET circuitry is required for most applications. The delays ensure that the device is kept in RESET, until the device power supply and clock are stable. For additional information on RESET operation, see Section 3.0. The first timer is the Power-up Timer (PWRT), which optionally provides a fixed delay of 72 ms (nominal) on power-up only (POR and BOR). The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable.

With the PLL enabled (HS/PLL Oscillator mode), the time-out sequence following a Power-on Reset is different from other Oscillator modes. The time-out sequence is as follows:

- 1. The PWRT time-out is invoked after a POR time delay has expired.
- 2. The Oscillator Start-up Timer (OST) is invoked. However, this is still not a sufficient amount of time to allow the PLL to lock at high frequencies.
- 3. The PWRT timer is used to provide an additional fixed 2 ms (nominal) time-out to allow the PLL ample time to lock to the incoming clock frequency.

TABLE 2-2: OSC1 AND OSC2 PIN STATES IN SLEEP MODE

OSC Mode	OSC1 Pin	OSC2 Pin
ECIO	Floating	Configured as PORTA, bit 6
EC	Floating	At logic low
HS	Feedback inverter disabled, at quiescent voltage level	Feedback inverter disabled, at quiescent voltage level

Note: See Table 3-1 in the "**Reset**" section, for time-outs due to SLEEP and MCLR Reset.

^{© 2002-2013} Microchip Technology Inc.

3.1 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected. To take advantage of the POR circuitry, just tie the MCLR pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004). For a slow rise time, see Figure 3-2.

When the device starts normal operation (i.e., exits the RESET condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in RESET until the operating conditions are met.

FIGURE 3-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



ing into MCLR from external capacitor C, in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

3.2 Power-up Timer (PWRT)

The Power-up Timer provides a fixed nominal time-out (parameter 33) only on power-up from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip-to-chip due to VDD, temperature and process variation. See DC parameter D033 for details.

3.3 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (parameter 32). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

3.4 PLL Lock Time-out

With the PLL enabled, the time-out sequence following a Power-on Reset is different from other Oscillator modes. A portion of the Power-up Timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (TPLL) is typically 2 ms and follows the Oscillator Start-up Time-out (OST).

3.5 Brown-out Reset (BOR)

A configuration bit, BOREN, can disable (if clear/ programmed), or enable (if set) the Brown-out Reset circuitry. If VDD falls below parameter D005 for greater than parameter 35, the brown-out situation will reset the chip. A RESET may not occur if VDD falls below parameter D005 for less than parameter 35. The chip will remain in Brown-out Reset until VDD rises above BVDD. If the Power-up Timer is enabled, it will be invoked after VDD rises above BVDD; it then will keep the chip in RESET for an additional time delay (parameter 33). If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute the additional time delay.

3.6 Time-out Sequence

On power-up, the time-out sequence is as follows: First, PWRT time-out is invoked after the POR time delay has expired. Then, OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 3-3, Figure 3-4, Figure 3-5, Figure 3-6 and Figure 3-7 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if MCLR is kept low long enough, the time-outs will expire. Bringing MCLR high will begin execution immediately (Figure 3-5). This is useful for testing purposes or to synchronize more than one PIC18FXXX device operating in parallel.

Table 3-2 shows the RESET conditions for some Special Function Registers, while Table 3-3 shows the RESET conditions for all the registers.

5.0 FLASH PROGRAM MEMORY

The FLASH Program Memory is readable, writable, and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

5.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

The program memory space is 16-bits wide, while the data RAM space is 8-bits wide. Table Reads and Table Writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table Read operations retrieve data from program memory and places it into the data RAM space. Figure 5-1 shows the operation of a Table Read with program memory and data RAM.

Table Write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in Section 5.5, "Writing to FLASH Program Memory". Figure 5-2 shows the operation of a Table Write with program memory and data RAM.

Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a Table Write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 5-1: TABLE READ OPERATION

REGISTER 8-3: INTCON3 REGISTER

- n = Value at POR

	R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0		
	INT2IP ⁽¹⁾	INT1IP ⁽¹⁾	_	INT2IE	INT1IE	—	INT2IF	INT1IF		
	bit 7							bit 0		
bit 7	INT2IP ⁽¹⁾ :	INT2 Externa	al Interrupt	Priority bit						
	1 = High priority									
	0 = Low priority									
bit 6	INT1IP ⁽¹⁾ :	INT1 Externa	al Interrupt	Priority bit						
	1 = High p	riority								
	0 = Low pr	iority								
bit 5	Unimplem	nented: Read	l as '0'							
bit 4	INT2IE: IN	IT2 External I	Interrupt En	able bit						
	1 = Enable	es the INT2 e	xternal inte	rrupt						
	0 = Disable	es the INT2 e	external inte	errupt						
bit 3	INT1IE: IN	IT1 External I	Interrupt En	able bit						
	1 = Enable	es the INT1 e	xternal inte	rrupt						
	0 = Disable	es the INT1 e	external inte	errupt						
bit 2	Unimplem	nented: Read	l as '0'							
bit 1	INT2IF: IN	T2 External I	nterrupt Fla	ag bit						
	1 = The IN	T2 external i	nterrupt occ	curred (must	be cleared	in software)				
	0 = The IN	T2 external i	nterrupt did	not occur						
bit 0	INT1IF: IN	T1 External I	nterrupt Fla	ag bit						
	1 = The IN	T1 external i	nterrupt occ	curred (must	be cleared	in software)				
	0 = The IN	T1 external i	nterrupt did	not occur						
	Note 1:	Maintain thi	s bit cleare	d (= 0).						
	Legend:									
	R = Reada	able bit	W = V	Vritable bit	U = Unir	nplemented	bit, read as	'0'		

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. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

'0' = Bit is cleared

'1' = Bit is set

x = Bit is unknown

PIC18FXX39

REGISTER 9-1: TRISE REGISTER

- n = Value at POR

	R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1						
	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0						
	bit 7	·		·		·		bit 0						
bit 7	IBF: Input	Buffer Full S	Status bit	-l oiting to be	read by the									
	1 = A word 0 = No word	rd has been i	received and	d waiting to be) read by the	9 0 9 0								
bit 6	OBF: Outp	out Buffer Fu	ull Status bit	t										
	1 = The output buffer still holds a previously written word													
	0 = The ou	Itput buffer I	nas been re	ad										
bit 5	IBOV : Inpu	ut Buffer Ove	erflow Dete	ct bit (in Micro	processor n	node) boon read								
	⊥ = A write (must	be cleared in	n software)		JU 1185 1101	Deen reau								
	0 = No ove	erflow occur	red											
bit 4	PSPMODE	E: Parallel S	lave Port M	lode Select bit										
	1 = Paralle	I Slave Port	t mode											
hit 3		arr urpose r rented: Res	/Unioue has '0'											
hit 2	TRISE2 R	PE2 Direction	n Control hi	t										
	1 = Input			L										
	0 = Output	t												
bit 1	TRISE1: R	E1 Direction	n Control bi	t										
	1 = Input	۰.												
hit ()		PEO Direction	n Control bi	t										
	1 = Input			L										
	0 = Output	t												
	<u> </u>							1						
	Legend:													
	R = Reada	able bit	W = V	Nritable bit	U = Unim	plemented l	bit, read as '	0'						

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

14.4 Developing Applications Using the Motor Control Kernel

The Motor Control kernel allows users to develop their applications without having knowledge of motor control. The key parameters of the motor control kernel can be set and read through the Application Program Interface (API) methods discussed in the previous section.

The overall application can be thought of as a protocol stack, as shown in Figure 14-3. In this case, the API methods reside between the user's application and the ProMPT kernel, and are used to exchange parameter values. The motor control kernel sets the PWM duty cycles based on the inputs from the application software.

A typical motor control routine is shown in Example 14-1. In this case, the motor will run at 20 Hz for 10 seconds, accelerate to 60 Hz at the rate of 10 Hz/s, remain at 60 Hz for 20 seconds, and finally stop.

FIGURE 14-3:

LAYERS OF THE MOTOR CONTROL ARCHITECTURE STACK



EXAMPLE 14-1: MOTOR CONTROL ROUTINE USING THE ProMPT APIS

```
Void main()
   unsigned char i;
   unsigned char j;
   ProMPT_Init(0);
                                         // Initialize the ProMPT block
   i = ProMPT_SetFrequency(10);
                                         // Set motor frequency to 10Hz
   for (i=0;i<161;i++)</pre>
                                         // Set counter for 10 sec @ 1/16 sec per tick
       {
       j = ProMPT_Tick(void);
                                         // Tick of 1/16 sec
       ProMPT_ClearTick(void);
                                         // Clearing the Tick flag
       }
                                         // Set acceleration rate to 10 Hz/sec
   ProMPT_SetAccelRate(10);
   i = ProMPT_SetFrequency(60);
                                         // Set motor frequency to 60 Hz
   for (i=0;i<161;i++)</pre>
                                         // Set counter for 20 Sec @ 1/16 sec per tick
                                         // (2 loops of 10 Sec each)
       {
                                        // Tick of 1/16 Sec
       j = ProMPT_Tick(void);
                                         // Clearing the Tick flag
       ProMPT_ClearTick(void);
       j = ProMPT_Tick(void);
                                         // Tick of 1/16 Sec
       ProMPT_ClearTick(void);
                                         // Clearing the Tick flag
   i = ProMPT SetFrequency(0);
                                         // Set motor frequency to 0 Hz (stop)
   while(1);
                                          // End of the task
```

16.4.3.2 Reception

When the R/\overline{W} bit of the address byte is clear and an address match occurs, the R/\overline{W} bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and the SDA line is held low (ACK).

When the address byte overflow condition exists, then the no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set, or bit SSPOV (SSPCON1<6>) is set.

An MSSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPCON1<0> = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit CKP (SSPCON<4>). See Section 16.4.4 ("Clock Stretching"), for more detail.

16.4.3.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low, regardless of SEN (see "Clock Stretching", Section 16.4.4, 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, pin RC3/ SCK/SCL should be enabled by setting bit CKP (SSPCON1<4>). 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 (Figure 16-9).

The ACK pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not ACK), then the data transfer is complete. In this case, when the ACK is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the START bit. If the SDA line was low (ACK), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3/SCK/SCL must be enabled by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in 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.

16.4.5 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that, the first byte after the START condition usually determines which device will be the slave addressed by the master. 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 one of eight addresses reserved for specific purposes by the I^2C protocol. It consists of all '0's with R/W = 0.

The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7> set). Following a START bit detect, 8-bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eighth bit), and on the falling edge of the ninth bit (ACK bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device specific or a general call address.

In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match, and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-bit Address mode, then the second half of the address is not necessary, the UA bit will not be set, and the slave will begin receiving data after the Acknowledge (Figure 16-15).





16.4.9 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated START condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I^2C logic module is in the IDLE state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the baud rate generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one baud rate generator count (TBRG). When the baud rate generator times out, if SDA is sampled high, the SCL pin will be de-asserted (brought high). When SCL is sampled high, the baud rate generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the baud rate generator will not be reloaded, leaving the SDA pin held low. As soon as a START condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the baud rate generator has timed out.

- Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.
 - 2: A bus collision during the Repeated START condition occurs if:
 - SDA is sampled low when SCL goes from low to high.
 - SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data "1".

Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

16.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated START sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note: Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated START condition is complete.

FIGURE 16-20: REPEAT START CONDITION WAVEFORM



17.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 17-1 shows the formula for computation of the baud rate for different USART modes, which only apply in Master mode (internal clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRG register can be calculated using the formula in Table 17-1. From this, the error in baud rate can be determined. Example 17-1 shows the calculation of the baud rate error for the following conditions:

- Fosc = 16 MHz
- Desired Baud Rate = 9600
- BRGH = 0
- SYNC = 0

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the FOSC/(16(X + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

17.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

EXAMPLE 17-1: CALCULATING BAUD RATE ERROR

Desired Baud Rate	= Fosc / $(64 (X + 1))$
Solving for X:	
X X X	= $((Fosc / Desired Baud Rate) / 64) - 1$ = $((16000000 / 9600) / 64) - 1$ = $[25.042] = 25$
Calculated Baud Rate	= 16000000 / (64 (25 + 1)) = 9615
Error	 <u>(Calculated Baud Rate – Desired Baud Rate)</u> Desired Baud Rate (9615 – 9600) / 9600 0.16%

TABLE 17-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate = Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	N/A

Legend: X = value in SPBRG (0 to 255)

TABLE 17-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

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
TXSTA	CSRC	TX9	TXEN	SYNC	_	BRGH	TRMT	TX9D	0000 -010	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.



FIGURE 17-7: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



17.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of the SLEEP mode and bit SREN, which is a "don't care" in Slave mode.

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR register will transfer the data to the RCREG register, and if enable bit RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

- Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- 2. If interrupts are desired, set enable bit RCIE.
- 3. If 9-bit reception is desired, set bit RX9.
- 4. To enable reception, set enable bit CREN.
- Flag bit RCIF will be set when reception is complete. An interrupt will be generated if enable bit RCIE was set.
- Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 7. Read the 8-bit received data by reading the RCREG register.
- 8. If any error occurred, clear the error by clearing bit CREN.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value POR,	e on BOR	Valu All C RES	e on other ETS
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INTOIE	RBIE	TMR0IF	INTOIF	RBIF	0000	000x	0000	000u
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF		TMR2IF	TMR1IF	0000	0000	0000	0000
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	_	TMR2IE	TMR1IE	0000	0000	0000	0000
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP		TMR2IP	TMR1IP	0000	0000	0000	0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000	-00x	0000	-00x
RCREG	USART Receive Register									0000	0000	0000
TXSTA	CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D	0000	-010	0000	-010
SPBRG	Baud Rate	0000	0000	0000	0000							

TABLE 17-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Legend: x = unknown, - = unimplemented, read as '0'.

Shaded cells are not used for Synchronous Slave Reception.

Note 1: The PSPIF, PSPIE and PSPIP bits are reserved on the PIC18F2X39 devices; always maintain these bits clear.

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and Vss), or the voltage level on the RA3/AN3/ VREF+ pin and RA2/AN2/VREF- pin.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

A device RESET forces all registers to their RESET state. This forces the A/D module to be turned off and any conversion is aborted.

Each port pin associated with the A/D converter can be configured as an analog input (RA3 can also be a voltage reference) or as a digital I/O.

The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH/ ADRESL registers, the GO/DONE bit (ADCON0<2>) is cleared, and A/D interrupt flag bit, ADIF is set. The block diagram of the A/D module is shown in Figure 18-1.



FIGURE 18-1: A/D BLOCK DIAGRAM

© 2002-2013 Microchip Technology Inc.

PIC18FXX39

	•••••		•••••••									
	R/P-1	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1				
	DEBUG	_	_	_	_	LVP	_	STVREN				
	bit 7							bit 0				
bit 7	DEBUG: Background Debugger Enable bit 1 = Background Debugger disabled. RB6 and RB7 configured as general purpose I/O pins. 0 = Background Debugger enabled. RB6 and RB7 are dedicated to In-Circuit Debug.											
bit 6-3	Unimplem	Unimplemented: Read as '0'										
bit 2	LVP: Low Voltage ICSP Enable bit 1 = Low Voltage ICSP enabled 0 = Low Voltage ICSP disabled											
bit 1	Unimplem	ented: Rea	d as '0'									
bit 0	STVREN: Stack Full/Underflow Reset Enable bit 1 = Stack Full/Underflow will cause RESET 0 = Stack Full/Underflow will not cause RESET											
	Legend:											
	R = Reada	ble bit	C = Clear	able bit	U = Unin	nplemented	bit, read as	'0'				

REGISTER 20-4:	CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)
----------------	---

- n = Value when device is unprogrammed

u = Unchanged from programmed state

20.3 Power-down Mode (SLEEP)

Power-down mode is entered by executing a $\ensuremath{\mathtt{SLEEP}}$ instruction.

If enabled, the Watchdog Timer will be cleared, but keeps running, the \overline{PD} bit (RCON<3>) is cleared, the \overline{TO} (RCON<4>) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, low or hi-impedance).

For lowest current consumption in this mode, place all I/O pins at either VDD or VSS, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and disable external clocks. Pull all I/O pins that are hi-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The TOCKI input should also be at VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The MCLR pin must be at a logic high level (VIHMC).

20.3.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

- 1. External RESET input on MCLR pin.
- 2. Watchdog Timer Wake-up (if WDT was enabled).
- 3. Interrupt from INT pin, RB port change or a Peripheral Interrupt.

The following peripheral interrupts can wake the device from SLEEP:

- 1. PSP read or write.
- 2. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
- 3. TMR3 interrupt. Timer3 must be operating as an asynchronous counter.
- 4. CCP Capture mode interrupt.
- 5. Special event trigger (Timer1 in Asynchronous mode using an external clock).
- 6. MSSP (START/STOP) bit detect interrupt.
- MSSP transmit or receive in Slave mode (SPI/I²C).
- 8. USART RX or TX (Synchronous Slave mode).
- 9. A/D conversion (when A/D clock source is RC).
- 10. EEPROM write operation complete.
- 11. LVD interrupt.

Other peripherals cannot generate interrupts, since during SLEEP, no on-chip clocks are present.

External MCLR Reset will cause a device RESET. All other events are considered a continuation of program execution and will cause a "wake-up". The TO and PD bits in the RCON register can be used to determine the cause of the device RESET. The PD bit, which is set on power-up, is cleared when SLEEP is invoked. The TO bit is cleared, if a WDT time-out occurred (and caused wake-up).

When the SLEEP instruction is being executed, the next instruction (PC + 2) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

20.3.2 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If an interrupt condition (interrupt flag bit and interrupt enable bits are set) occurs before the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT postscaler will not be cleared, the TO bit will not be set and PD bits will not be cleared.
- If the interrupt condition occurs during or after the execution of a SLEEP instruction, the device will immediately wake-up from SLEEP. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the TO bit will be set and the PD bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the PD bit. If the PD bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

22.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PIC MCUs and can be used to develop for this and other PIC microcontrollers. The MPLAB ICD utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial ProgrammingTM protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

22.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable VDD and VPP supplies, which allow it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode, the PRO MATE II device programmer can read, verify, or program PIC devices. It can also set code protection in this mode.

22.10 PICSTART Plus Entry Level Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

The PICSTART Plus development programmer supports all PIC devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

22.11 PICDEM 1 Low Cost PIC MCU Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A). PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE incircuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

22.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I^2C^{TM} bus and separate headers for connection to an LCD module and a keypad.



FIGURE 23-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)

TABLE 23-14: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	SS↓ to SCK↓ or SCK↑ input		Тсү	_	ns	
71	TscH	SCK input high time	Continuous	1.25 Tcy + 30		ns	
71A		(Slave mode)	Single Byte	40	_	ns	(Note 1)
72	TscL	SCK input low time (Slave mode)	Continuous	1.25 Tcy + 30		ns	
72A			Single Byte	40	_	ns	(Note 1)
73A	Тв2в	Last clock edge of Byte 1 to the first cloc	1.5 Tcy + 40	_	ns	(Note 2)	
74	TscH2diL, TscL2diL	Hold time of SDI data input to SCK ed	100	—	ns		
75	TdoR	SDO data output rise time	PIC18FXXXX	—	25	ns	
			PIC18LFXXXX	_	60	ns	VDD = 2V
76	TdoF	SDO data output fall time	PIC18FXXXX	_	25	ns	
			PIC18LFXXXX	_	60	ns	VDD = 2V
77	TssH2doZ	SS↑ to SDO output hi-impedance		10	50	ns	
78	TscR	SCK output rise time (Master mode)	PIC18FXXXX	_	25	ns	
			PIC18LFXXXX	_	60	ns	VDD = 2V
79	TscF	SCK output fall time (Master mode)	PIC18FXXXX	_	25	ns	
			PIC18LFXXXX	_	60	ns	VDD = 2V
80	TscH2doV, TscL2doV	SDO data output valid after SCK edge	PIC18FXXXX	_	50	ns	
			PIC18LFXXXX	_	150	ns	VDD = 2V
82	TssL2doV	SDO data output valid after $\overline{\text{SS}}\downarrow$ edge	PIC18FXXXX	_	50	ns	
			PIC18LFXXXX	_	150	ns	VDD = 2V
83	TscH2ssH, TscL2ssH	SS ↑ after SCK edge		1.5 Tcy + 40	—	ns	

Note 1: Requires the use of Parameter # 73A.

2: Only if Parameter # 71A and # 72A are used.

44-Lead Plastic Quad Flat No Lead Package (ML) 8x8 mm Body (QFN)

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



	Units	INCHES			MILLIMETERS*		
Dimensi	on Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		44			44	
Pitch	р	.026 BSC 0.65 BSC					
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0	0.02	0.05
Base Thickness	A3		.010 REF			0.25 REF	
Overall Width	E	.315 BSC 8.00 BSC		8.00 BSC			
Exposed Pad Width	E2	.262	.268	.274	6.65	6.80	6.95
Overall Length		.315 BSC			8.00 BSC		
Exposed Pad Length	D2	.262	.268	.274	6.65	6.80	6.95
Lead Width	В	.012	.013	.013	0.30	0.33	0.35
Lead Length	L	.014	.016	.018	0.35	0.40	0.45

*Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side. JEDEC equivalent: M0-220

Drawing No. C04-103

APPENDIX A: REVISION HISTORY

Revision A (November 2002)

Original data sheet for the PIC18FXX39 family.

Revision B (January 2013)

Added a note to each package outline drawing.

TABLE B-1: DEVICE DIFFERENCES

APPENDIX B: DEVICE DIFFERENCES

The differences between the devices listed in this data sheet are shown in Table B-1.

Feature	PIC18F2439	PIC18F2539	PIC18F4439	PIC18F4539
Program Memory (Kbytes)	12	24	12	24
Data Memory (Bytes)	640	1408	640	1408
A/D Channels	5	5	8	8
Parallel Slave Port (PSP)	No	No	Yes	Yes
Package Types	28-pin DIP 28-pin SOIC	28-pin DIP 28-pin SOIC	40-pin DIP 44-pin TQFP 44-pin QFN	40-pin DIP 44-pin TQFP 44-pin QFN