



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	20MHz
Connectivity	I²C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f818t-i-ss

Email: info@E-XFL.COM

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

```
FIGURE 2-3:
```

PIC16F818 REGISTER FILE MAP

A	File ddress		File Address		File Address	۵	File ddre
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180
TMR0	01h	OPTION REG	81h	TMR0	101h	OPTION REG	181
PCL	02h	PCI	82h	PCL	102h	PCL	182
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183
FSR	04h	FSR	84h	FSR	104h	FSR	184
PORTA	05h	TRISA	85h		105h		185
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186
	07h	ITTIOD	87h		107h		187
	08h		88h		108h		188
	09h		89h		109h		189
PCLATH	0Ah	PCI ATH	8Ah	PCLATH	10Ah	PCI ATH	184
INTCON	0Bh		8Bh	INTCON	10Bh		18F
PIR1	0Ch	PIF1	8Ch	FEDATA	10Ch	EFCON1	180
PIR2	0Dh	PIE2	8Dh	FEADR	10Dh	EECON2	180
TMR1I	0Eh	PCON	8Eh		10Eh	Reserved ⁽¹⁾	185
	0Eh	OSCCON	0EH		10Eh	Reserved(1)	100
	10h		00h		110h	Reserveu	100
	1011 11h	OSCIONE	9011 01h				190
	12h	DD2	9111 02h				
	1211 13h		9211 02h				
SSPCON	1/h		9311 04b				
	1 4 11 15h	55P5TAT	9411 05b				
	16h		9511 06b				
	17h		9011 07h				
CONTOON	18h		9711 09h				
	1011 10h		9011 00h				
	1Δh		9911 04b				
	1Rh		9AN 0Ph				
	1011 101		9011 00h				
	1011 1Dh		901 006				
	1011 1Eb		9DN				
ADCONO	1Eh		905h		11Fh		19F
ADCONU	20h	ADCON1	9711		120h		140
	2011	Purpose Register	A0h		12011		.7 (
General		32 Bytes	BFh				
Purpose			C0h	Accesses		Accesses	
Register		Accesses		20h-7Fh		20h-7Fh	
96 Bytes		40h-7Fh					
	7Fh		FFh		17Fh		1FF
Bank 0		Bank 1		Bank 2		Bank 3	
Unimple * Not a ph lote 1: These re	mented d nysical reg egisters a	ata memory locati jister. re reserved; maint	ons, read ain these	as '0'. registers clear.			

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The upper bits (PC<12:8>) are not readable but are indirectly writable through the PCLATH register. On any Reset, the upper bits of the PC will be cleared. Figure 2-5 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> \rightarrow PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> \rightarrow PCH).

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the application note *AN556, "Implementing a Table Read"* (DS00556).

2.3.2 STACK

The PIC16F818/819 family has an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no status bits to indicate stack overflow or stack underflow conditions.

2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

2.4 Indirect Addressing: INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

EXAMPLE 2-1: INDIRECT ADDRESSING

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 06)
- A read of the INDF register now will return the value of 0Ah

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no operation (although status bits may be affected).

A simple program to clear RAM locations, 20h-2Fh, using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

	MOVLW	0x20	;initialize pointer				
NEXT	MOVWF	FSR	;to RAM				
	CLRF	INDF	;clear INDF register				
	INCF	FSR	;inc pointer				
	BTFSS	FSR, 4	;all done?				
	GOTO	NEXT	;NO, clear next				
CONTINUE	2						
	:		;YES, continue				

An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (Status<7>) as shown in Figure 2-6.

3.3 Reading Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit (EECON1<7>) and then set control bit, RD (EECON1<0>). The data is available in the very next cycle in the EEDATA register; therefore, it can be read in the next instruction (see Example 3-1). EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

The steps to reading the EEPROM data memory are:

- 1. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
- 2. Clear the EEPGD bit to point to EEPROM data memory.
- 3. Set the RD bit to start the read operation.
- 4. Read the data from the EEDATA register.

		DA	• *	
BANKSEL	EEADR		;	Select Bank of EEADR
MOVF	ADDR, W		;	
MOVWF	EEADR		;	Data Memory Address
			;	to read
BANKSEL	EECON1		;	Select Bank of EECON1
BCF	EECON1,	EEPGD	;	Point to Data memory
BSF	EECON1,	RD	;	EE Read
BANKSEL	EEDATA		;	Select Bank of EEDATA
MOVF	EEDATA,	W	;	W = EEDATA

EXAMPLE 3-1: DATA EEPROM READ

3.4 Writing to Data EEPROM Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then, the user must follow a specific write sequence to initiate the write for each byte.

The write will not initiate if the write sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment (see Example 3-2).

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times except when updating EEPROM. The WREN bit is not cleared by hardware

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set. At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

The steps to write to EEPROM data memory are:

- 1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
- 2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
- 3. Write the 8-bit data value to be programmed in the EEDATA register.
- 4. Clear the EEPGD bit to point to EEPROM data memory.
- 5. Set the WREN bit to enable program operations.
- 6. Disable interrupts (if enabled).
- 7. Execute the special five instruction sequence:
 - Write 55h to EECON2 in two steps (first to W, then to EECON2)
 - Write AAh to EECON2 in two steps (first to W, then to EECON2)
 - · Set the WR bit
- 8. Enable interrupts (if using interrupts).
- 9. Clear the WREN bit to disable program operations.
- 10. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set (EEIF must be cleared by firmware). If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to be clear, to indicate the end of the program cycle.

EXAMPLE 3-2: DATA EEPROM WRITE

		BANKSEL	EECON1		;	Select Bank of
		BTFSC GOTO BANKSEL	EECON1, \$-1 EEADR	WR	; ; ; ;	Wait for write to complete Select Bank of
		MOVF	ADDR, W		; ;	EEADR
		MOAME	EEADR		; ;	Data Memory Address to write
		MOVF MOVWF	VALUE, V EEDATA	1	; ; ;	Data Memory Value
		BANKSEL	EECON1		;;;	Select Bank of EECON1
		BCF	EECON1,	EEPGD	;	Point to DATA
		BSF	EECON1,	WREN	; ;	Enable writes
		BCF MOVLW	INTCON, 55h	GIE	;;	Disable INTs.
ed	nce	MOVWF	EECON2		;	Write 55h
equir	enbe	MOVLW MOVWF	AAh EECON2		; ;	Write AAh
æ	ű	BSF	EECON1,	WR	;	Set WR bit to
L	•	BSF	INTCON,	GIE	; ;	Enable INTs.
		BCF	EECON1,	WREN	;	Disable writes

3.5 Reading Flash Program Memory

To read a program memory location, the user must write two bytes of the address to the EEADR and EEADRH registers, set the EEPGD control bit (EECON1<7>) and then set control bit, RD (EECON1<0>). Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF EECON1, RD" instruction to be ignored. The data is available in the very next cycle in the EEDATA and EEDATH registers; therefore, it can be read as two bytes in the following instructions. EEDATA and EEDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 3-3:	FLASH PROGRAM READ

BANKSEL EEADRH		;	Select Bank of EEADRH
MOVF ADDRH,	W	;	
MOVWF EEADRH		;	MS Byte of Program
		;	Address to read
MOVF ADDRL,	W	;	
MOVWF EEADR		;	LS Byte of Program
		;	Address to read
BANKSEL EECON1		;	Select Bank of EECON1
BSF EECON1	, EEPGD	;	Point to PROGRAM
		;	memory
BSF EECON1	, RD	;	EE Read
		;	
NOP		;	Any instructions
		;	here are ignored as
NOP		;	program memory is
		;	read in second cycle
		;	after BSF EECON1,RD
BANKSEL EEDATA		;	Select Bank of EEDATA
MOVF EEDATA	, W	;	DATAL = EEDATA
MOVWF DATAL		;	
MOVF EEDATH	, W	;	DATAH = EEDATH
MOVWF DATAH		;	

3.6 Erasing Flash Program Memory

The minimum erase block is 32 words. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be bulk erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 32 words of program memory is erased. The Most Significant 11 bits of the EEADRH:EEADR point to the block being erased. EEADR< 4:0> are ignored.

The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation.

For protection, the write initiate sequence for EECON2 must be used.

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, only during the cycle in which the erase takes place. 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.

3.6.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

- 1. Load EEADRH:EEADR with address of row being erased.
- Set EEPGD bit to point to program memory; set WREN bit to enable writes and set FREE bit to enable the erase.
- 3. Disable interrupts.
- 4. Write 55h to EECON2.
- 5. Write AAh to EECON2.
- 6. Set the WR bit. This will begin the row erase cycle.
- 7. The CPU will stall for duration of the erase.



CERAMIC RESONATOR OPERATION (HS OR XT

OSC CONFIGURATION)



- **2:** A series resistor (Rs) may be required.
- 3: RF varies with the resonator chosen (typically between 2 M Ω to 10 M Ω).

TABLE 4-2: CERAMIC RESONATORS (FOR DESIGN GUIDANCE ONLY)

Typical Capacitor Values Used:										
Mode	Freq	OSC1	OSC2							
ХТ	455 kHz	56 pF	56 pF							
	2.0 MHz	47 pF	47 pF							
	4.0 MHz	33 pF	33 pF							
HS	8.0 MHz	27 pF	27 pF							
	16.0 MHz	22 pF	22 pF							

Capacitor values are for design guidance only.

These capacitors were tested with the resonators listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following this table for additional information.

Note: When using resonators with frequencies above 3.5 MHz, the use of HS mode rather than XT mode is recommended. HS mode may be used at any VDD for which the controller is rated. If HS is selected, it is possible that the gain of the oscillator will overdrive the resonator. Therefore, a series resistor should be placed between the OSC2 pin and the resonator. As a good starting point, the recommended value of Rs is 330Ω .

4.3 External Clock Input

The ECIO Oscillator mode requires an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

In the ECIO Oscillator mode, the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 4-3 shows the pin connections for the ECIO Oscillator mode.



EXTERNAL CLOCK INPUT OPERATION (ECIO CONFIGURATION)



4.5.3 OSCILLATOR CONTROL REGISTER

The OSCCON register (Register 4-2) controls several aspects of the system clock's operation.

The Internal Oscillator Select bits, IRCF2:IRCF0, select the frequency output of the internal oscillator block that is used to drive the system clock. The choices are the INTRC source (31.25 kHz), the INTOSC source (8 MHz) or one of the six frequencies derived from the INTOSC postscaler (125 kHz to 4 MHz). Changing the configuration of these bits has an immediate change on the multiplexor's frequency output.

4.5.4 MODIFYING THE IRCF BITS

The IRCF bits can be modified at any time regardless of which clock source is currently being used as the system clock. The internal oscillator allows users to change the frequency during run time. This is achieved by modifying the IRCF bits in the OSCCON register. The sequence of events that occur after the IRCF bits are modified is dependent upon the initial value of the IRCF bits before they are modified. If the INTRC (31.25 kHz, IRCF<2:0> = 000) is running and the IRCF bits are modified to any other value than '000', a 4 ms (approx.) clock switch delay is turned on. Code execution continues at a higher than expected frequency while the new frequency stabilizes. Time sensitive code should wait for the IOFS bit in the OSCCON register to become set before continuing. This bit can be monitored to ensure that the frequency is stable before using the system clock in time critical applications.

If the IRCF bits are modified while the internal oscillator is running at any other frequency than INTRC (31.25 kHz, IRCF<2:0> \neq 000), there is no need for a 4 ms (approx.) clock switch delay. The new INTOSC frequency will be stable immediately after the **eight** falling edges. The IOFS bit will remain set after clock switching occurs.

Note: Caution must be taken when modifying the IRCF bits using BCF or BSF instructions. It is possible to modify the IRCF bits to a frequency that may be out of the VDD specification range; for example, VDD = 2.0V and IRCF = 111 (8 MHz).

4.5.5 CLOCK TRANSITION SEQUENCE WHEN THE IRCF BITS ARE MODIFIED

Following are three different sequences for switching the internal RC oscillator frequency.

- Clock before switch: 31.25 kHz (IRCF<2:0> = 000)
 - 1. IRCF bits are modified to an INTOSC/INTOSC postscaler frequency.
 - 2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
 - 3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
 - The IOFS bit is clear to indicate that the clock is unstable and a 4 ms (approx.) delay is started. Time dependent code should wait for IOFS to become set.
 - 5. Switchover is complete.
- Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ 000)
 - 1. IRCF bits are modified to INTRC (IRCF<2:0> = 000).
 - 2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
 - 3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
 - 4. Oscillator switchover is complete.
- Clock before switch: One of INTOSC/INTOSC postscaler (IRCF<2:0> ≠ 000)
 - 1. IRCF bits are modified to a different INTOSC/ INTOSC postscaler frequency.
 - 2. The clock switching circuitry waits for a falling edge of the current clock, at which point CLKO is held low.
 - 3. The clock switching circuitry then waits for eight falling edges of requested clock, after which it switches CLKO to this new clock source.
 - 4. The IOFS bit is set.
 - 5. Oscillator switchover is complete.









7.2 Timer1 Operation in Timer Mode

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is FOSC/4. The synchronize control bit, T1SYNC (T1CON<2>), has no effect since the internal clock is always in sync.

7.3 Timer1 Counter Operation

Timer1 may operate in Asynchronous or Synchronous mode depending on the setting of the TMR1CS bit.

When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in Counter mode, the module must first have a falling edge before the counter begins to increment.

7.4 Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RB7/T1OSI/PGD when bit T1OSCEN is set, or on pin RB6/T1OSO/T1CKI/PGC when bit T1OSCEN is cleared.

If $\overline{\text{T1SYNC}}$ is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

In this configuration, during Sleep mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut-off. The prescaler, however, will continue to increment.



FIGURE 7-1: TIMER1 INCREMENTING EDGE





8.0 TIMER2 MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time base for the PWM mode of the CCP1 module. The TMR2 register is readable and writable and is cleared on any device Reset.

The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>).

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon Reset.

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit, TMR2IF (PIR1<1>)).

Timer2 can be shut-off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

Register 8-1 shows the Timer2 Control register.

Additional information on timer modules is available in the "*PIC*[®] *Mid-Range MCU Family Reference Manual*" (DS33023).

8.1 Timer2 Prescaler and Postscaler

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, MCLR, WDT Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

8.2 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module which optionally uses it to generate a shift clock.

FIGURE 8-1: TIMER2 BLOCK DIAGRAM



9.2 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the CCP1 pin is:

- Driven high
- Driven low
- · Remains unchanged

The action on the pin is based on the value of control bits, CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

FIGURE 9-2: COMPARE MODE OPERATION BLOCK DIAGRAM



9.2.1 CCP PIN CONFIGURATION

The user must configure the CCP1 pin as an output by clearing the TRISB<x> bit.

- Note 1: Clearing the CCP1CON register will force the CCP1 compare output latch to the default low level. This is not the data latch.
 - 2: The TRISB bit (2 or 3) is dependent upon the setting of configuration bit 12 (CCPMX).

9.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

9.2.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen, the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

9.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated that may be used to initiate an action.

The special event trigger output of CCP1 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled). This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

Note: The special event trigger from the CCP1 module will not set interrupt flag bit, TMR1IF (PIR1<0>).

TABLE 9-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value POR,	e on BOR	Valu all o Res	e on ther sets
0Bh,8Bh 10BH,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000	000x	0000	000u
0Ch	PIR1		ADIF	_	_	SSPIF	CCP1IF	TMR2IF	TMR1IF	- 0	0000	- 0	0000
8Ch	PIE1		ADIE	_		SSPIE	CCP1IE	TMR2IE	TMR1IE	- 0	0000	- 0	0000
86h	TRISB	PORTE	PORTB Data Direction Register									1111	1111
0Eh	TMR1L	Holding	g Register	r for the Lea	ast Significa	ant Byte of t	he 16-bit T	MR1 Regi	ster	xxxx	xxxx	uuuu	uuuu
0Fh	TMR1H	Holding	g Register	r for the Mo	st Significa	nt Byte of th	ne 16-bit T	MR1 Regis	ster	xxxx	xxxx	uuuu	uuuu
10h	T1CON		_	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00	0000	uu	uuuu
15h	CCPR1L	Capture	Capture/Compare/PWM Register 1 (LSB)									uuuu	uuuu
16h	CCPR1H	Capture	Capture/Compare/PWM Register 1 (MSB)									uuuu	uuuu
17h	CCP1CON		_	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	00	0000	00	0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by Capture and Timer1.

10.3 SSP I²C Mode Operation

The SSP module in I²C mode fully implements all slave functions, except general call support and provides interrupts on Start and Stop bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer. These are the RB4/SCK/SCL pin, which is the clock (SCL) and the RB1/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISB<4,1> bits.

To ensure proper communication of the I²C Slave mode, the TRIS bits (TRISx [SDA, SCL]) corresponding to the I²C pins must be set to '1'. If any TRIS bits (TRISx<7:0>) of the port containing the I²C pins (PORTx [SDA, SCL]) are changed in software during I²C communication using a Read-Modify-Write instruction (BSF, BCF), then the I²C mode may stop functioning properly and I²C communication may suspend. Do not change any of the TRISx bits (TRIS bits of the port containing the I²C pins) using the instruction BSF or BCF during I²C communication. If it is absolutely necessary to change the TRISx bits during communication, the following method can be used:

EXAMPLE 10-1:

MOVF IORLW	TRISC, W 0x18	; Example for an 18-pin part such as the PIC16F818/819 ; Ensures <4:3> bits are `11′
ANDLW	B'11111001'	; Sets <2:1> as output, but will not alter other bits
MOVWF	TRISC	; USET CAN USE CHEIT OWN TOGIC HELE, SUCH as TORLW, AORLW AND AND W

The SSP module functions are enabled by setting SSP Enable bit, SSPEN (SSPCON<5>).



FIGURE 10-5: SSP BLOCK DIAGRAM (I²C™ MODE)

The SSP module has five registers for I²C operation:

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I^2C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I^2C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address) with Start and Stop bit interrupts enabled to support Firmware Master mode
- I²C Slave mode (10-bit address) with Start and Stop bit interrupts enabled to support Firmware Master mode
- I²C Firmware Controlled Master mode with Start and Stop bit interrupts enabled, slave is Idle

Selection of any I^2C mode, with the SSPEN bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISB bits. Pull-up resistors must be provided externally to the SCL and SDA pins for proper operation of the I^2C module.

Additional information on SSP I²C operation may be found in the *"PIC[®] Mid-Range MCU Family Reference Manual"* (DS33023).

The ADRESH:ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the A/D Result register pair, 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 11-1.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see **Section 11.1** "**A/D Acquisition Requirements**". After this sample time has elapsed, the A/D conversion can be started.

These steps should be followed for doing an A/D conversion:

- 1. Configure the A/D module:
 - Configure analog pins/voltage reference and digital I/O (ADCON1)
 - Select A/D input channel (ADCON0)
 - Select A/D conversion clock (ADCON0)
 - Turn on A/D module (ADCON0)
- 2. Configure A/D interrupt (if desired):
 - Clear ADIF bit
 - Set ADIE bit
 - Set GIE bit
- 3. Wait the required acquisition time.
- 4. Start conversion:
 - Set GO/DONE bit (ADCON0)
- 5. Wait for A/D conversion to complete by either:
 - Polling for the GO/DONE bit to be cleared (with interrupts disabled); OR
 - Waiting for the A/D interrupt
- 6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
- 7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before the next acquisition starts.

FIGURE 11-1:





© 2001-2013 Microchip Technology Inc.

11.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 11-2. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD), see Figure 11-2. The maximum recommended impedance for analog sources is 2.5 k\Omega. As the impedance is decreased, the acquisition time may be decreased.

After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 11-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the "*PIC*[®] *Mid-Range MCU Family Reference Manual*" (DS33023).

EQUATION 11-1: ACQUISITION TIME

TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient = TAMP + TC + TCOFF $= 2 \mu s + TC + [(Temperature - 25^{\circ}C)(0.05 \mu s/^{\circ}C)]$ TC = CHOLD (RIC + Rss + Rs) In(1/2047) $= -120 pF (1 k\Omega + 7 k\Omega + 10 k\Omega) In(0.0004885)$ $= 16.47 \mu s$ $TACQ = 2 \mu s + 16.47 \mu s + [(50^{\circ}C - 25^{\circ}C)(0.05 \mu s/^{\circ}C)]$ $= 19.72 \mu s$

Note 1: The reference voltage (VREF) has no effect on the equation since it cancels itself out.

- **2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.
- **3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.
- **4:** After a conversion has completed, a 2.0 TAD delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

FIGURE 11-2: ANALOG INPUT MODEL



14.0 DEVELOPMENT SUPPORT

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

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

14.1 MPLAB Integrated Development Environment Software

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

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

The MPLAB IDE allows you to:

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

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

15.3 DC Characteristics: Internal RC Accuracy PIC16F818/819, PIC16F818/819 TSL (Industrial, Extended) PIC16LF818/819, PIC16LF818/819 TSL (Industrial)

PIC16LF818/819 ⁽³⁾ Standard Operating Conditions (unless otherwise stated) PIC16LF818/819 TSL ⁽³⁾ Operating temperature -40°C ≤ TA ≤ +85°C for industrial (Industrial) -40°C ≤ TA ≤ +85°C for industrial											
PIC16F8 PIC16F8 (Indu	818/819 ⁽³⁾ 318/819 TSL ⁽³⁾ ustrial, Extended)	Standard Op Operating ter	Standard Operating Conditions (unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for extended								
Param No.	Device	Min	Тур	Max	Units	Conditions					
	INTOSC Accuracy @ F	req = 8 MHz,	4 MHz, 2 M	Hz, 1 MHz,	500 kHz, 25	50 kHz, 125 kHz ⁽¹⁾					
	PIC16LF818/819	-5	±1	5	%	+25°C					
		-25	_	25	%	-10°C to +85°C	VDD = 2.7-3.3V				
		-30	_	30	%	-40°C to +85°C					
	PIC16F818/819 ⁽⁴⁾	-5	±1	5	%	+25°C					
		-25	_	25	%	-10°C to +85°C	Vpp – 4 5-5 5V				
		-30	_	30	%	-40°C to +85°C	100 - 4.5 5.51				
		-35	_	35	%	-40°C to +125°C					
	PIC16LF818/819 TSL	-2	±1	2	%	+25°C					
		-5	_	5	%	-10°C to +85°C	VDD = 2.7-3.3V				
		-10	_	10	%	-40°C to +85°C					
	PIC16F818/819 TSL ⁽⁵⁾	-2	±1	2	%	+25°C					
		-5	_	5	%	-10°C to +85°C	Vpp – 4 5-5 5V				
		-10	_	10	%	-40°C to +85°C	100 - 4.5 5.51				
		-15	_	15	%	-40°C to +125°C					
	INTRC Accuracy @ Fre	eq = 31 kHz ⁽²⁾)								
	PIC16LF818/819	26.562		35.938	kHz	-40°C to +85°C	VDD = 2.7-3.3V				
	PIC16F818/819 ⁽⁴⁾	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 4.5-5.5V				
	PIC16LF818/819 TSL	26.562	_	35.938	kHz	-40°C to +85°C	VDD = 2.7-3.3V				
	PIC16F818/819 TSL ⁽⁵⁾	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 4.5-5.5V				

Legend: Shading of rows is to assist in readability of the table.

Note 1: Frequency calibrated at 25°C. OSCTUNE register can be used to compensate for temperature drift.

2: INTRC frequency after calibration.

3: The only specification difference between a non-TSL device and a TSL device is the internal RC oscillator specifications listed above. All other specifications are maintained.

4: Example part number for the specifications listed above: PIC16F818-I/SS (PIC16F818 device, Industrial temperature, SSOP package).

5: Example part number for the specifications listed above: PIC16F818-I/SSTSL (PIC16F818 device, Industrial temperature, SSOP package).

15.4 DC Characteristics: PIC16F818/819 (Industrial, Extended) PIC16LF818/819 (Industrial)

DC CHA	ARACTI	ERISTICS	$\begin{array}{llllllllllllllllllllllllllllllllllll$						
Param No.	Sym	Characteristic	Min	Тур†	Мах	Units	Conditions		
	VIL	Input Low Voltage							
		I/O ports:							
D030		with TTL buffer	Vss	—	0.15 Vdd	V	For entire VDD range		
D030A			Vss	—	0.8V	V	$4.5V \leq V\text{DD} \leq 5.5V$		
D031		with Schmitt Trigger buffer	Vss	—	0.2 Vdd	V			
D032		MCLR, OSC1 (in RC mode)	Vss	—	0.2 Vdd	V	(Note 1)		
D033		OSC1 (in XT and LP mode)	Vss	—	0.3V	V			
		OSC1 (in HS mode)	Vss	—	0.3 Vdd	V			
		Ports RB1 and RB4:							
D034		with Schmitt Trigger buffer	Vss	—	0.3 Vdd	V	For entire VDD range		
	Vih	Input High Voltage							
		I/O ports:							
D040		with TTL buffer	2.0	—	Vdd	V	$4.5V \le VDD \le 5.5V$		
D040A			0.25 VDD + 0.8V	—	Vdd	V	For entire VDD range		
D041		with Schmitt Trigger buffer	0.8 Vdd	—	Vdd	V	For entire VDD range		
D042		MCLR	0.8 Vdd	—	Vdd	V			
D042A		OSC1 (in XT and LP mode)	1.6V	—	Vdd	V			
		OSC1 (in HS mode)	0.7 Vdd	—	Vdd	V			
D043		OSC1 (in RC mode)	0.9 Vdd	—	Vdd	V	(Note 1)		
		Ports RB1 and RB4:							
D044		with Schmitt Trigger buffer	0.7 Vdd	—	Vdd	V	For entire VDD range		
D070	IPURB	PORTB Weak Pull-up Current	50	250	400	μΑ	VDD = 5V, VPIN = VSS		
	lı∟	Input Leakage Current (Notes	2, 3)		I	1	I		
D060		I/O ports	—	—	±1	μA	Vss \leq VPIN \leq VDD, pin at high-impedance		
D061		MCLR	—	—	±5	μΑ	$Vss \leq V \text{PIN} \leq V \text{DD}$		
D063		OSC1	_	_	±5	μA	Vss \leq VPIN \leq VDD, XT, HS and LP oscillator configuration		

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F818/819 be driven with external clock in RC mode.

- 2: 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.
- 3: Negative current is defined as current sourced by the pin.











FIGURE 16-21: MINIMUM AND MAXIMUM VIN vs. VDD (TTL INPUT, -40°C TO +125°C)





20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	С		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

PIC16F818/819

Writing to Flash Program Memory	31
Code Protection	. 89, 100
Computed GOTO	23
Configuration Bits	
Crystal Oscillator and Ceramic Resonators	
Customer Change Notification Service	173
Customer Notification Service	173
Customer Support	173

D

Data EEPROM Memory	25
Associated Registers	
EEADR Register	25
EEADRH Register	25
EECON1 Register	25
EECON2 Register	25
EEDATA Register	25
EEDATH Register	25
Operation During Code-Protect	
Protection Against Spurious Writes	
Reading	27
Write Interrupt Enable Flag (EEIF Bit)	
Writing	27
Data Memory	
Special Function Registers	13
DC and AC Characteristics	
Graphs and Tables	
DC Characteristics	
Internal RC Accuracy	125
PIC16F818/819, PIC16LF818/819	126
Power-Down and Supply Current	118
Supply Voltage	117
Development Support	
Device Differences	165
Device Overview	5
Direct Addressing	24

Ε

EEADR Register 24
EEADRH Register
EECON1 Register2
EECON2 Register
EEDATA Register
EEDATH Register
Electrical Characteristics
Endurance
Errata
External Clock Input
External Interrupt Input (RB0/INT). See Interrupt Sources.

F

Flash Program Memory2	25
Associated Registers	32
EEADR Register2	25
EEADRH Register2	25
EECON1 Register2	25
EECON2 Register2	25
EEDATA Register2	25
EEDATH Register2	25
Erasing2	28
Reading2	28
Writing	30
FSR Register13, 14, 15, 2	23
G	
General Purpose Register File1	10

L

I/O Ports ² C	39
Associated Registers	79
Master Mode Operation	79
Mode	76
Mode Selection	76
Multi Master Made Operation	70
Multi-Master Mode Operation	. 79
Slave Mode	. / /
Addressing	. 77
Reception	. 77
SCL, SDA Pins	.77
Transmission	77
ID Locations 89	100
In-Circuit Debugger	100
In Circuit Serial Programming	00
	109
	101
INDF Register14, 15	, 23
Indirect Addressing23	, 24
Instruction Format	103
Instruction Set	103
Descriptions	105
Read-Modify-Write Operations	103
Summary Table	100
	104
ADDLW	105
ADDWF	105
ANDLW	105
ANDWF	105
BCF	105
BSF	105
BTESC	106
DTECC	100
DIF 33	100
CALL	106
CLRF	106
CLRW	106
CLRWDT	106
COMF	107
DECF	107
DECESZ	107
COTO	107
	107
	107
INCFSZ	107
IORLW	108
IORWF	108
MOVF	108
MOVLW	108
MOVWF	108
NOP	108
DETEIE	100
	109
REILW	109
RETURN	109
RLF	109
RRF	109
SLEEP	109
SUBLW	110
SUBWE	110
SW/ADE	110
	110
	110
	110
INT Interrupt (RB0/INT). See Interrupt Sources.	
INTCON Register	15
GIE Bit	18
INTE Bit	18
INTF Bit	18
RBIF Bit	18
	10