



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	40MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
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/pic18lf452-i-ml

Email: info@E-XFL.COM

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

TADLE 1-3. FIGIOLANZ FINOUT I/O DESCRIFTIONS	TABLE 1-3:	PIC18F4X2 PINOUT I/O DESCRIPTIONS
--	------------	-----------------------------------

Din Marra	Pi	in Numb	ber	Pin	Buffer	Decerimtica
	DIP	PLCC	TQFP	Туре	Туре	Description
MCLR/Vpp	1	2	18			Master Clear (input) or high voltage ICSP programming enable pin.
MCLR				I	ST	Master Clear (Reset) input. This pin is an active low RESET to the device.
Vpp				I	ST	High voltage ICSP programming enable pin.
NC				—	_	These pins should be left unconnected.
OSC1/CLKI OSC1	13	14	30	I	ST	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode, CMOS otherwise.
CLKI				I	CMOS	External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
OSC2/CLKO/RA6 OSC2	14	15	31	0	_	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode
CLKO				0	_	In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate
RA6				I/O	TTL	General Purpose I/O pin.
						PORTA is a bi-directional I/O port.
RA0/AN0 RA0 AN0	2	3	19	I/O I	TTL Analog	Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	3	4	20	I/O	TTL Analog	Digital I/O. Analog input 1
RA2/AN2/VREF-	4	5	21		, maneg	
RA2 AN2 VREE-				I/O I	TTL Analog Analog	Digital I/O. Analog input 2. A/D Beference Voltage (Low) input
BA3/AN3/VBEE+	5	6	22		7 maiog	The reference voltage (Low) input.
RA3 AN3 VREF+	0	0		I/O I I	TTL Analog Analog	Digital I/O. Analog input 3. A/D Reference Voltage (High) input.
RA4/T0CKI RA4 T0CKI	6	7	23	I/O I	ST/OD ST	Digital I/O. Open drain when configured as output. Timer0 external clock input.
RA5/AN4/SS/LVDIN RA5	7	8	24	I/O	TTL	Digital I/O.
AN4 SS					Analog ST Analog	Analog input 4. SPI Slave Select input. Low Voltage Detect Input
RA6					7 maiog	(See the OSC2/CLKO/RA6 pin.)
Legend: TTL = TTL c	compati	i ble inpu	t	1		CMOS = CMOS compatible input or output
ST = Schmit	tt Trigg	er input	with CM	IOS lev	els	I = Input

O = Output OD = Open Drain (no P diode to VDD)

P = Power

2.4 External Clock Input

The EC and ECIO Oscillator modes require an external clock source to be connected to the OSC1 pin. The feedback device between OSC1 and OSC2 is turned off in these modes to save current. There is no oscillator start-up time required after a Power-on Reset or after a recovery from SLEEP mode.

In the EC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION



The ECIO Oscillator mode functions like the EC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-5 shows the pin connections for the ECIO Oscillator mode.

FIGURE 2-5:

OPERATION (ECIO CONFIGURATION)

EXTERNAL CLOCK INPUT



2.5 HS/PLL

A Phase Locked Loop circuit is provided as a programmable option for users that want to multiply the frequency of the incoming crystal oscillator signal by 4. For an input clock frequency of 10 MHz, the internal clock frequency will be multiplied to 40 MHz. This is useful for customers who are concerned with EMI due to high frequency crystals.

The PLL can only be enabled when the oscillator configuration bits are programmed for HS mode. If they are programmed for any other mode, the PLL is not enabled and the system clock will come directly from OSC1.

The PLL is one of the modes of the FOSC<2:0> configuration bits. The Oscillator mode is specified during device programming.

A PLL lock timer is used to ensure that the PLL has locked before device execution starts. The PLL lock timer has a time-out that is called TPLL.



REGISTER 5-1: EECON1 REGISTER (ADDRESS FA6h)

	R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0				
	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD				
	bit 7	-						bit 0				
bit 7	EEPGD: F 1 = Access	LASH Progr s FLASH Pro	ram or Data ogram mem		√emory Select I	bit						
	0 = Access	s Data EEPF	ROM memo	ry								
bit 6	CFGS: FL/	ASH Progra	.m/Data EE	or Configura	ation Select bit							
	1 = Access 0 = Access	ঃ Configurati s FLASH Prc	ion registers ogram or Da	₃ ata EEPRO№	√ memory							
bit 5	Unimplem	ented: Rea	d as '0'									
bit 4	FREE: FLASH Row Erase Enable bit											
	 1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation) 0 = Perform write only WPERP: ELASH Program/Data EE Error Elag bit 											
bit 3	WRERR: F	FLASH Prog	jram/Data E	E Error Flaç	y bit							
	 1 = A write operation is prematurely terminated (any RESET during self-timed programming in normal operation) = The write operation completed 											
	Note: Wh tra	hen a WREF acing of the e	R occurs, t error conditi	the EEPGD ion.	and CFGS bits	are not cle	ared. This	allows				
bit 2	WREN: FL	_ASH Progra	am/Data EE	Write Enab	le bit							
	1 = Allows 0 = Inhibits	write cycles write to the	; ; EEPROM									
bit 1	WR: Write	Control bit										
	 1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle. (The operation is self timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.) 0 = Write cycle to the EEPROM is complete 											
bit 0	RD: Read	Control bit										
	1 = Initiates (Read f in softv 0 = Does n	s an EEPRC takes one cy vare. RD bit not initiate ar)M read /cle. RD is c cannot be s n EEPROM	cleared in ha set when EE read	Irdware. The RE PGD = 1.)) bit can or	ıly be set (n	ot cleared)				
	·											
	Legend:											

W = Writable bit

'1' = Bit is set

R = Readable bit

- n = Value at POR

x = Bit is unknown

U = Unimplemented bit, read as '0'

'0' = Bit is cleared

11.1 Timer1 Operation

Timer1 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

The Operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

When TMR1CS = 0, Timer1 increments every instruction cycle. When TMR1CS = 1, Timer1 increments on every rising edge of the external clock input or the Timer1 oscillator, if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored, and the pins are read as '0'.

Timer1 also has an internal "RESET input". This RESET can be generated by the CCP module (Section 14.0).



FIGURE 11-2: TIMER1 BLOCK DIAGRAM: 16-BIT READ/WRITE MODE



FIGURE 11-1: TIMER1 BLOCK DIAGRAM

15.4.6.1 I²C Master Mode Operation

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a Repeated START condition. Since the Repeated START condition is also the beginning of the next serial transfer, the I^2C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. START and STOP conditions indicate the beginning and end of transmission.

The baud rate generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I^2C operation. See Section 15.4.7 ("Baud Rate Generator"), for more detail. A typical transmit sequence would go as follows:

- 1. The user generates a START condition by setting the START enable bit, SEN (SSPCON2<0>).
- SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
- 3. The user loads the SSPBUF with the slave address to transmit.
- 4. Address is shifted out the SDA pin until all 8 bits are transmitted.
- 5. The MSSP Module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
- 6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- 7. The user loads the SSPBUF with eight bits of data.
- 8. Data is shifted out the SDA pin until all 8 bits are transmitted.
- 9. The MSSP Module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
- 10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- 11. The user generates a STOP condition by setting the STOP enable bit PEN (SSPCON2<2>).
- 12. Interrupt is generated once the STOP condition is complete.

16.1 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 16-1 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 16-1. From this, the error in baud rate can be determined. Example 16-1 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.

16.1.1 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 16-1: 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 16-1: 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 16-2: 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 Ra	te Genera	0000 0000	0000 0000						

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

TABLE 16-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD	Fosc =	40 MHz	SPBRG	33	MHz	SPBRG	25	MHz	SPBRG	20	MHz	SPBRG
RATE (Kbps)	KBAUD	% ERROR	value (decimal)									
0.3	NA	-	-									
1.2	NA	-	-									
2.4	NA	-	-	2.40	-0.07	214	2.40	-0.15	162	2.40	+0.16	129
9.6	9.62	+0.16	64	9.55	-0.54	53	9.53	-0.76	40	9.47	-1.36	32
19.2	18.94	-1.36	32	19.10	-0.54	26	19.53	+1.73	19	19.53	+1.73	15
76.8	78.13	+1.73	7	73.66	-4.09	6	78.13	+1.73	4	78.13	+1.73	3
96	89.29	-6.99	6	103.13	+7.42	4	97.66	+1.73	3	104.17	+8.51	2
300	312.50	+4.17	1	257.81	-14.06	1	NA	-	-	312.50	+4.17	0
500	625	+25.00	0	NA	-	-	NA	-	-	NA	-	-
HIGH	625	-	0	515.63	-	0	390.63	-	0	312.50	-	0
LOW	2.44	-	255	2.01	-	255	1.53	-	255	1.22	-	255
BAUD	Fosc =	16 MHz	SPBRG	10	MHz	SPBRG	7.159	09 MHz	SPBRG	5.068	8 MHz	SPBRG
RATE (Kbps)	KBAUD	% ERROR	value (decimal)									
0.3	NA	-	-									
1.2	1.20	+0.16	207	1.20	+0.16	129	1.20	+0.23	92	1.20	0	65
2.4	2.40	+0.16	103	2.40	+0.16	64	2.38	-0.83	46	2.40	0	32
9.6	9.62	+0.16	25	9.77	+1.73	15	9.32	-2.90	11	9.90	+3.13	7
19.2	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5	19.80	+3.13	3
76.8	83.33	+8.51	2	78.13	+1.73	1	111.86	+45.65	0	79.20	+3.13	0
96	83.33	-13.19	2	78.13	-18.62	1	NA	-	-	NA	-	-
300	250	-16.67	0	156.25	-47.92	0	NA	-	-	NA	-	-
500	NA	-	-									
HIGH	250	-	0	156.25	-	0	111.86	-	0	79.20	-	0
LOW	0.98	-	255	0.61	-	255	0.44	-	255	0.31	-	255
BAUD	Fosc	= 4 MHz	SPBRG	3.5795	645 MHz	SPBRG	1	MHz	SPBRG	32.76	8 kHz	SPBRG
RATE (Kbps)	KBAUD	% ERROR	value (decimal)									
0.3	0.30	-0.16	207	0.30	+0.23	185	0.30	+0.16	51	0.26	-14.67	1
1.2	1.20	+1.67	51	1.19	-0.83	46	1.20	+0.16	12	NA	-	-
2.4	2.40	+1.67	25	2.43	+1.32	22	2.23	-6.99	6	NA	-	-
9.6	8.93	-6.99	6	9.32	-2.90	5	7.81	-18.62	1	NA	-	-
19.2	20.83	+8.51	2	18.64	-2.90	2	15.63	-18.62	0	NA	-	-
76.8	62.50	-18.62	0	55.93	-27.17	0	NA	-	-	NA	-	-
96	NA	-	-									
300	NA	-	-									
500	NA	-	-									
HIGH	62.50	-	0	55.93	-	0	15.63	-	0	0.51	-	0
LOW	0.24	-	255	0.22	-	255	0.06	-	255	0.002	-	255

16.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>), or enable bit CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

- 1. Initialize the SPBRG register for the appropriate baud rate (Section 16.1).
- 2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
- 3. Ensure bits CREN and SREN are clear.

- 4. If interrupts are desired, set enable bit RCIE.
- 5. If 9-bit reception is desired, set bit RX9.
- 6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
- Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if the enable bit RCIE was set.
- 8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 9. Read the 8-bit received data by reading the RCREG register.
- 10. If any error occurred, clear the error by clearing bit CREN.
- 11. 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 on POR, BOR	Value on All Other RESETS
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INTOIE	RBIE	TMR0IF	INTOIF	RBIF	0000 000x	0000 000u
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
RCREG	USART R	eceive R	0000 0000	0000 0000						
TXSTA	CSRC	TX9	TXEN	SYNC	_	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate	e Genera		0000 0000	0000 0000					

TABLE 16-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for Synchronous Master Reception. **Note 1:** The PSPIF, PSPIE and PSPIP bits are reserved on the PIC18F2X2 devices; always maintain these bits clear.

FIGURE 16-8: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



16.4.2 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.
- 5. 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 on POR, BOR		Value on OR, BOR RESETS	
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INTOIE	RBIE	TMR0IF	INTOIF	RBIF	0000	000x	0000	000u
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000	0000	0000	0000
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000	0000	0000	0000
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	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	Generat		0000	0000	0000	0000					

TABLE 16-11: 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 PIC18F2X2 devices; always maintain these bits clear.

REGISTER 19-12: DEVICE ID REGISTER 1 FOR PIC18FXX2 (DEVID1: BYTE ADDRESS 3FFFFEh)

	R	R	R	R	R	R	R	R					
	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0					
	bit 7							bit 0					
bit 7-5	DEV2:DEV 000 = PIC 001 = PIC 100 = PIC 101 = PIC	DEV2:DEV0: Device ID bits 000 = PIC18F252 001 = PIC18F452 100 = PIC18F242 101 = PIC18F442 BEV4:BEV0: Bevision ID bits											
bit 4-0	REV4:REV These bits	REV4:REV0: Revision ID bits These bits are used to indicate the device revision.											
	Legend:												
	R = Reada	ole bit	P =Progra	ammable bit	U = Unir	nplemented	bit, read as	'0'					
	- n = Value	programme	ed state										
REGISTER 19-13:	DEVICEID	REGISTE	R2FORP	IC18FXX2	(DEVID2: E	BYTE ADD	RESS 3FFI	FFh)					

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

bit 7-0 **DEV10:DEV3:** Device ID bits These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number.

Legend:		
R = Readable bit	P =Programmable bit	U = Unimplemented bit, read as '0'
- n = Value when device	e is unprogrammed	u = Unchanged from programmed state

TABLE 20-1: OPCODE FIELD DESCRIPTIONS

Field	Description
a	RAM access bit
	a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
hhh	Bit address within an 8-bit file register (0 to 7)
BSR	Bank Select Begister Lised to select the current RAM bank
d	
u	d = 0: store result in WREG,
	d = 1: store result in file register f.
dest	Destination either the WREG register or the specified register file location
f	8-bit Register file address (0x00 to 0xFF)
fs	12-bit Register file address (0x000 to 0xFFF). This is the source address.
fd	12-bit Register file address (0x000 to 0xFFF). This is the destination address.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value)
label	Label name
mm	The mode of the TBLPTR register for the Table Read and Table Write instructions. Only used with Table Read and Table Write instructions:
*	No Change to register (such as TBLPTR with Table reads and writes)
*+	Post-Increment register (such as TBLPTR with Table reads and writes)
* _	Post-Decrement register (such as TBLPTR with Table reads and writes)
+*	Pre-Increment register (such as TBLPTR with Table reads and writes)
n	The relative address (2's complement number) for relative branch instructions, or the direct address for
	Call/Branch and Return instructions
PRODH	Product of Multiply high byte
PRODL	Product of Multiply low byte
s	Fast Call/Return mode select bit.
	s = 0: do not update into/trom shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
u	Unused or Unchanged
WREG	Working register (accumulator)
x	Don't care (0 or 1)
	The assembler will generate code with $x = 0$. It is the recommended form of use for compatibility with all Microchip software tools.
TBLPTR	21-bit Table Pointer (points to a Program Memory location)
TABLAT	8-bit Table Latch
TOS	Top-of-Stack
PC	Program Counter
PCL	Program Counter Low Byte
PCH	Program Counter High Byte
PCLATH	Program Counter High Byte Latch
PCLATU	Program Counter Upper Byte Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer
TO	Time-out bit
PD	Power-down bit
C, DC, Z, OV, N	ALU status bits Carry, Digit Carry, Zero, Overflow, Negative
[]	Optional
()	Contents
\rightarrow	Assigned to
< >	Register bit field
E	In the set of
italics	User defined term (font is courier)

PIC18FXX2

BCF	Bit Clear	f					
Syntax:	[<i>label</i>] E	[label] BCF f,b[,a]					
Operands:	0 ≤ f ≤ 255 0 ≤ b ≤ 7 a ∈ [0,1]	$0 \le f \le 255$ $0 \le b \le 7$ $a \in [0,1]$					
Operation:	$0 \rightarrow f < b >$						
Status Affected:	None						
Encoding:	1001	bbba	ffff	ffff			
Description:	Bit 'b' in register 'f' is cleared. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).						
Words:	1						
Cycles:	1						
Q Cycle Activity:							
Q1	Q2	Q	}	Q4			
Decode	Read register 'f'	Proce Data	ss a re	Write gister 'f'			
Example:	BCF 1	FLAG_RE	G, 7,	D			
Before Instruction FLAG_REG = 0xC7 After Instruction FLAG_REG = 0x47							

BN		Branch if	Branch if Negative				
Synt	ax:	[<i>label</i>] B	N n				
Ope	rands:	-128 ≤ n ≤	127				
Ope	ration:	if negative (PC) + 2 +	bit is '1 2n \rightarrow F	, PC			
Statu	us Affected:	None					
Enco	oding:	1110	0110	nnnn	nnnn		
Desi		The 2's co added to t have incre instruction PC+2+2n. a two-cycl	The Negative bit is 1, then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC+2+2n. This instruction is then a two cycle instruction				
Wor	ds:	1	1				
Cycl	es:	1(2)	1(2)				
Q C If Ju	ycle Activity	:					
	Q1	Q2	Q3		Q4		
	Decode	Read literal 'n'	Proce Data	ss W ເ	/rite to PC		
	No operation	No operation	No operat	ion d	No operation		
lf N	o Jump:						
	Q1	Q2	Q3		Q4		
	Decode	Read literal 'n'	Proce Data	ss i (No operation		
<u>Exai</u>	<u>mple</u> :	HERE	BN	Jump			
	Before Instr	uction					
	PC	= ad	dress (H	ERE)			

PC	=	address	(HERE)
After Instruction			
If Negative PC If Negative PC	= = =	1; address 0; address	(Jump) (HERE+2)

PIC18FXX2

COMF	Complement f						
Syntax:	[label] (COMF	f [,d	[,a]			
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$					
Operation:	$(\overline{f}) \rightarrow de$	est					
Status Affected:	N, Z						
Encoding:	0001	11da	fff	f	ffff		
Description:	The contents of register 'f' are com plemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f' (default). I 'a' is 0, the Access Bank will be selected, overriding the BSR value If 'a' = 1, then the bank will be selected as per the BSR value (default)						
Words:	1						
Cycles:	1						
Q Cycle Activity:							
Q1	Q2	Q	3		Q4		
Decode	Read register 'f'	Proce Data	ess a	W des	rite to tination		
Example:	COMF	REG,	0, 0				
Before Instru REG After Instruct REG W	ction = 0x13 ion = 0x13 = 0xEC						

CPFSEQ	Compare	Compare f with W, skip if f = W				
Syntax:	[label] C	[label] CPFSEQ f[,a]				
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]	5				
Operation:	(f) – (W), skip if (f) = (unsigned	= (W) comparison)				
Status Affected:	None					
Encoding:	0110	001a fff	f ffff			
Description:	Compares memory lo of W by pe subtraction If 'f' = W, t tion is disc cuted inste cycle inste Access Ba riding the the bank v BSR value	the contents bocation 'f' to tl erforming an n. hen the fetch carded and a ead, making ' uction. If 'a' is ank will be se BSR value. If vill be selecte e (default).	s of data he contents unsigned ned instruc- NOP is exe- this a two- s 0, the elected, over- ta' = 1, then ed as per the			
Words [.]	1					
Cycles: Q Cycle Activit	1(2) Note: 3 o by y:	cycles if skip a 2-word ins	and followed truction.			
Q1	Q2	Q3	Q4			
Decode	Read register 'f'	Process Data	No operation			
lf skip:		l				
Q1	Q2	Q3	Q4			
No	No	No	No			
If ckip and follo	operation	d instruction	operation			
Q1	02 Neu by 2-wor	Q3	Q4			
No	No	No	No			
operation	operation	operation	operation			
No operation	No operation	No operation	No operation			
Example:	HERE NEQUAL EQUAL	CPFSEQ REG :	e, 0			
PC Add W	lress = HE = ?	RE				
REG	= ?					
After Instru	ction					
lf REG	= W; = Ad	dress (FOID)	r.)			
If REG	, = Au ≠ W;		u,			
PC	2 = Ad	dress (NEQUA	AL)			

PIC18FXX2

SLEEP	Enter SL	EEP mode		SUBFWB	B Subtract f from W with		ith borrow		
Syntax:	[label]	SLEEP		Syntax:	[label]	[label] SUBFWB			
Operands:	None		ds: None O		Operands:	$0 \le f \le 25$	$0 \le f \le 255$		
Operation:	$00h \rightarrow W$	/DT,			d ∈ [0,1]	d ∈ [0,1]			
	$0 \rightarrow WD^{-1}$	T postscaler,		Operation:	a ∈ [0,1]	$\left(\frac{\overline{C}}{C}\right) \rightarrow doct$	ŀ		
	$1 \rightarrow 10,$ $0 \rightarrow PD$			Operation.	(I) – (V) 	$-(C) \rightarrow uesi$	L		
Status Affected:	TO, PD					, DC, Z			
Encodina:	0000	0000 000	0 0011	Encoding:	0101 Subtract	Ulda II			
Description:	The powe	er-down statu	ıs bit (PD) is	Description:	(borrow)	from W (2's c	omplement		
	cleared.	The time-out	status bit		method).	If 'd' is 0, the	result is		
	(TO) is se	et. Watchdog	Timer and		stored in	W. If 'd' is 1, t	the result is		
	The proc	essor is put i	nto SLEEP		0, the Ac	cess Bank will	be selected,		
	mode wit	h the oscillat	or stopped.		overriding	g the BSR val	ue. If 'a' is 1,		
Words:	1				then the l	bank will be s SB value (de	elected as		
Cycles:	1			Words:	1		iauny.		
Q Cycle Activity:				Cycles:	1				
Q1	Q2	Q3	Q4	O Cycle Activ	itv.				
Decode	No operation	Process Data	Go to sleep	Q1	Q2	Q3	Q4		
				Decode	Read	Process	Write to		
Example:	SLEEP				register 'f'	Data	destination		
Befo <u>re I</u> nstru	iction			Example 1:	SUBFWB	REG, 1, 0)		
<u>TO</u> = PD =	? ?			Before Ins	struction				
After Instruct	tion			REG W	= 3 = 2				
<u>TO</u> = PD =	1† 0			C	= 1				
+ If WDT caused	e wake-up tł	hie hit ie cloa	red	After Instr	uction				
	s wake-up, ii		ieu.	W	= 2				
				C Z	= 0 = 0				
				Ν	= 1 ; re	esult is negativ	е		
				Example 2:	SUBFWB	REG, 0, 0)		
				Before Ins	struction				
				W	= 2 = 5				
				C	= 1				
				Alter Instr REG	= 2				
				W	= 3				
				Z	= 1 = 0				
				N	= 0 ; re	esult is positive	•		
				Example 3:	SUBFWB	REG, 1, 0)		
				Before Ins REG	struction = 1				

W 2 = С = 0 After Instruction REG = 0 W = 2 C Z N = = 1 1 ; result is zero ò =

21.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for precompiled code to be used with the MPLINK object linker. When a routine from a library is called from another source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

21.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multiproject software development tool.

21.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro microcontrollers.

The MPLAB ICE in-circuit emulator system has been designed as a real-time emulation system, with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft[®] Windows environment were chosen to best make these features available to you, the end user.

21.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.





TABLE 22-15:	I ² C BUS START/STOP	BITS REQUIREMENTS	(SLAVE MODE)
--------------	---------------------------------	--------------------------	--------------

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	START condition	100 kHz mode	4700	—	ns	Only relevant for Repeated
		Setup time	400 kHz mode	600			START condition
91	THD:STA	START condition	100 kHz mode	4000	_	ns	After this period, the first
		Hold time	400 kHz mode	600			clock pulse is generated
92	Tsu:sto	STOP condition	100 kHz mode	4700		ns	
		Setup time	400 kHz mode	600	_		
93	THD:STO	STOP condition	100 kHz mode	4000	_	ns	
		Hold time	400 kHz mode	600			









FIGURE 23-28: A/D NON-LINEARITY vs. VREFH (VDD = VREFH, -40°C TO +125°C)



DS39564C-page 302

40-Lead Plastic Dual In-line (P) – 600 mil Body (PDIP)

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



wolded Package Thickness	AZ	.140	.150	.160	3.50	3.01	4.00
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	Е	.595	.600	.625	15.11	15.24	15.88
Molded Package Width	E1	.530	.545	.560	13.46	13.84	14.22
Overall Length	D	2.045	2.058	2.065	51.94	52.26	52.45
Tip to Seating Plane	L	.120	.130	.135	3.05	3.30	3.43
Lead Thickness	С	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.030	.050	.070	0.76	1.27	1.78
Lower Lead Width	В	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.620	.650	.680	15.75	16.51	17.27
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter § Significant Characteristic

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: MO-011

Drawing No. C04-016

44-Lead Plastic Leaded Chip Carrier (L) – Square (PLCC)

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



	Units INCHES* MILLIMETE			IILLIMETERS	ERS		
Dimensio	n Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		44			44	
Pitch	р		.050			1.27	
Pins per Side	n1		11			11	
Overall Height	Α	.165	.173	.180	4.19	4.39	4.57
Molded Package Thickness	A2	.145	.153	.160	3.68	3.87	4.06
Standoff §	A1	.020	.028	.035	0.51	0.71	0.89
Side 1 Chamfer Height	A3	.024	.029	.034	0.61	0.74	0.86
Corner Chamfer 1	CH1	.040	.045	.050	1.02	1.14	1.27
Corner Chamfer (others)	CH2	.000	.005	.010	0.00	0.13	0.25
Overall Width	Е	.685	.690	.695	17.40	17.53	17.65
Overall Length	D	.685	.690	.695	17.40	17.53	17.65
Molded Package Width	E1	.650	.653	.656	16.51	16.59	16.66
Molded Package Length	D1	.650	.653	.656	16.51	16.59	16.66
Footprint Width	E2	.590	.620	.630	14.99	15.75	16.00
Footprint Length	D2	.590	.620	.630	14.99	15.75	16.00
Lead Thickness	С	.008	.011	.013	0.20	0.27	0.33
Upper Lead Width	B1	.026	.029	.032	0.66	0.74	0.81
Lower Lead Width	В	.013	.020	.021	0.33	0.51	0.53
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter

§ Significant Characteristic

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: MO-047

Drawing No. C04-048

APPENDIX A: REVISION HISTORY

Revision A (June 2001)

Original data sheet for the PIC18FXX2 family.

Revision B (August 2002)

This revision includes the DC and AC Characteristics Graphs and Tables. The Electrical Specifications in Section 22.0 have been updated and there have been minor corrections to the data sheet text.

Revision C (October 2006)

Packaging diagrams updated.

APPENDIX B: DEVICE DIFFERENCES

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

Feature	PIC18F242	PIC18F252	PIC18F442	PIC18F452
Program Memory (Kbytes)	16	32	16	32
Data Memory (Bytes)	768	1536	768	1536
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 PLCC 44-pin TQFP	40-pin DIP 44-pin PLCC 44-pin TQFP

TABLE B-1:DEVICE DIFFERENCES