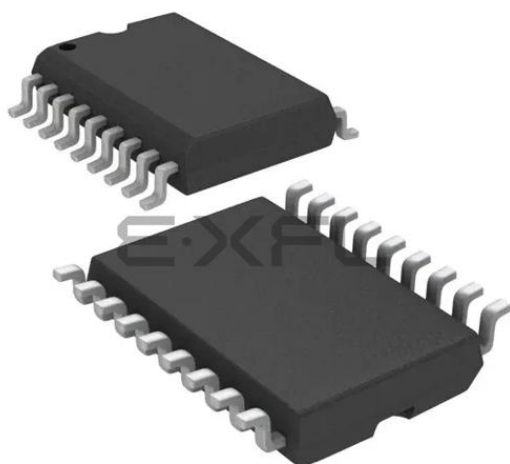




Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf1330-i-so

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniclient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICtail, PIC³² logo, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

3.8 Effects of Power-Managed Modes on the Various Clock Sources

When PRI_IDLE mode is selected, the designated primary oscillator continues to run without interruption. For all other power-managed modes, the oscillator using the OSC1 pin is disabled. The OSC1 pin (and OSC2 pin, if used by the oscillator) will stop oscillating.

In secondary clock modes (SEC_RUN and SEC_IDLE), the Timer1 oscillator is operating and providing the device clock. The Timer1 oscillator may also run in all power-managed modes if required to clock Timer1 or Timer3.

In internal oscillator modes (RC_RUN and RC_IDLE), the internal oscillator block provides the device clock source. The 31 kHz INTRC output can be used directly to provide the clock and may be enabled to support various special features, regardless of the power-managed mode (see **Section 20.2 “Watchdog Timer (WDT)”**, **Section 20.3 “Two-Speed Start-up”** and **Section 20.4 “Fail-Safe Clock Monitor”** for more information on WDT, Fail-Safe Clock Monitor and Two-Speed Start-up). The INTOSC output at 8 MHz may be used directly to clock the device or may be divided down by the postscaler. The INTOSC output is disabled if the clock is provided directly from the INTRC output.

If the Sleep mode is selected, all clock sources are stopped. Since all the transistor switching currents have been stopped, 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 INTRC is required to support WDT operation. The Timer1 oscillator may be operating to support a real-

time clock. Other features may be operating that do not require a device clock source (i.e., INTx pins and others). Peripherals that may add significant current consumption are listed in **Section 23.0 “Electrical Characteristics”**.

3.9 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 is stable under normal circumstances and the primary clock is operating and stable. For additional information on power-up delays, see **Section 5.5 “Device Reset Timers”**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up (parameter 33, Table 23-10). It is enabled by clearing (= 0) the PWRTEN Configuration bit.

The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable (LP, XT and HS modes). The OST does this by counting 1024 oscillator cycles before allowing the oscillator to clock the device.

When the HSPLL Oscillator mode is selected, the device is kept in Reset for an additional 2 ms, following the HS mode OST delay, so the PLL can lock to the incoming clock frequency.

There is a delay of interval T_{CSD} (parameter 38, Table 23-10), following POR, while the controller becomes ready to execute instructions. This delay runs concurrently with any other delays. This may be the only delay that occurs when any of the EC, RC or INTIO modes are used as the primary clock source.

TABLE 3-3: OSC1 AND OSC2 PIN STATES IN SLEEP MODE

Oscillator Mode	OSC1 Pin	OSC2 Pin
RC, INTIO1	Floating, external resistor should pull high	At logic low (clock/4 output)
RCIO	Floating, external resistor should pull high	Configured as PORTA, bit 6
INTIO2	Configured as PORTA, bit 7	Configured as PORTA, bit 6
ECIO	Floating, pulled by external clock	Configured as PORTA, bit 6
EC	Floating, pulled by external clock	At logic low (clock/4 output)
LP, XT and HS	Feedback inverter disabled at quiescent voltage level	Feedback inverter disabled at quiescent voltage level

Note: See Table 5-2 in **Section 5.0 “Reset”** for time-outs due to Sleep and MCLR Reset.

7.0 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.

7.1 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)

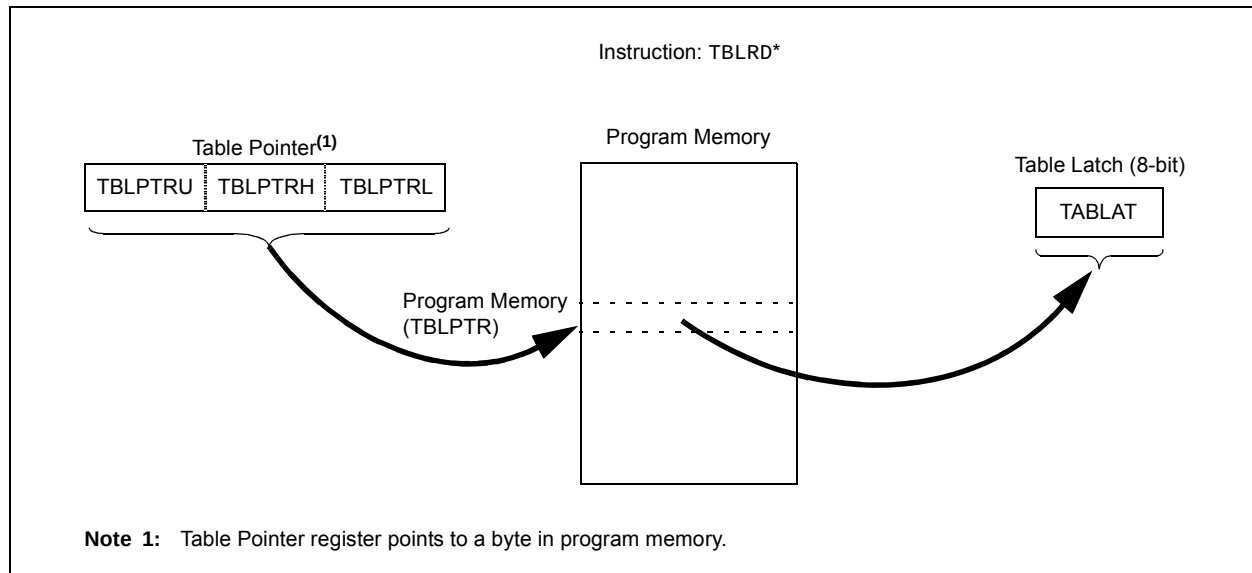
The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and place it into the data RAM space. Figure 7-1 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 7.5 “Writing to Flash Program Memory”**. Figure 7-2 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 7-1: TABLE READ OPERATION



13.0 TIMER1 MODULE

The Timer1 timer/counter module has the following features:

- 16-bit timer/counter (two 8-bit registers; TMR1H and TMR1L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- Status of system clock operation

Figure 13-1 is a simplified block diagram of the Timer1 module.

Register 13-1 details the Timer1 Control register. This register controls the operating mode of the Timer1 module and contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

The Timer1 oscillator can be used as a secondary clock source in power-managed modes. When the T1RUN bit is set, the Timer1 oscillator provides the system clock. If the Fail-Safe Clock Monitor is enabled and the Timer1 oscillator fails while providing the system clock, polling the T1RUN bit will indicate whether the clock is being provided by the Timer1 oscillator or another source.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

REGISTER 13-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
 1 = Enables register read/write of Timer1 in one 16-bit operation
 0 = Enables register read/write of Timer1 in two 8-bit operations
- bit 6 **T1RUN:** Timer1 System Clock Status bit
 1 = Device clock is derived from Timer1 oscillator
 0 = Device clock is derived from another source
- bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
 11 = 1:8 Prescale value
 10 = 1:4 Prescale value
 01 = 1:2 Prescale value
 00 = 1:1 Prescale value
- bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
 1 = Timer1 oscillator is enabled
 0 = Timer1 oscillator is shut off
 The oscillator inverter and feedback resistor are turned off to eliminate power drain.
- bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
 1 = Do not synchronize external clock input
 0 = Synchronize external clock input
When TMR1CS = 0:
 This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
- bit 1 **TMR1CS:** Timer1 Clock Source Select bit
 1 = External clock from T1OSO/T1CKI (on the rising edge)⁽¹⁾
 0 = Internal clock (Fosc/4)
- bit 0 **TMR1ON:** Timer1 On bit
 1 = Enables Timer1
 0 = Stops Timer1

Note 1: Placement of T1OSI and T1OSO/T1CKI depends on the value of the Configuration bit, T1OSCMX, of CONFIG3H.

PIC18F1230/1330

FIGURE 14-9: PWM PERIOD BUFFER UPDATES IN FREE-RUNNING MODE

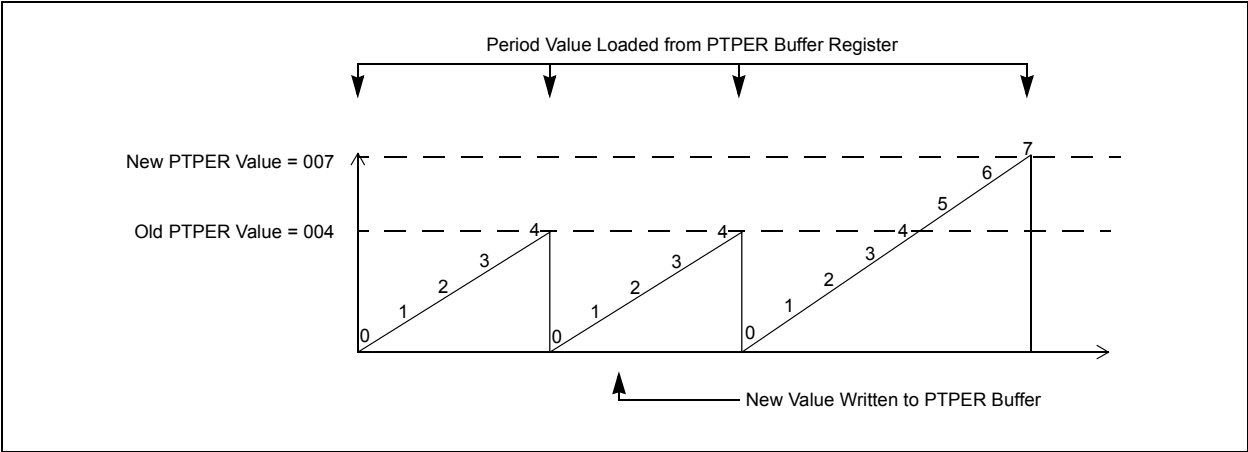
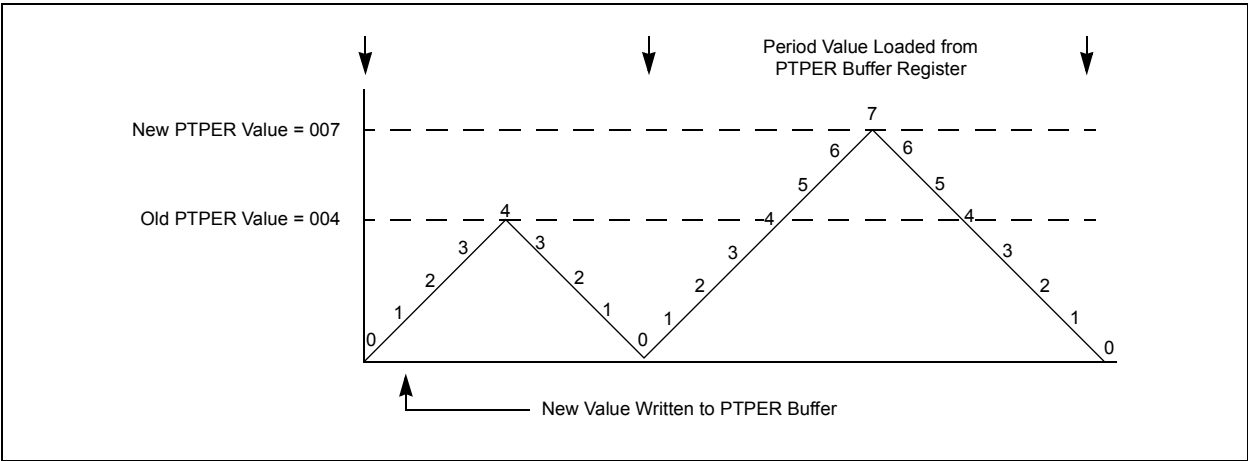


FIGURE 14-10: PWM PERIOD BUFFER UPDATES IN CONTINUOUS UP/DOWN COUNT MODES



PIC18F1230/1330

NOTES:

PIC18F1230/1330

EXAMPLE 15-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \text{FOSC} / (64 ([\text{SPBRGH}:\text{SPBRG}] + 1))$$

Solving for SPBRGH:SPBRG:

$$X = ((\text{FOSC} / \text{Desired Baud Rate}) / 64) - 1$$

$$= ((16000000 / 9600) / 64) - 1$$

$$= [25.042] = 25$$

$$\text{Calculated Baud Rate} = 16000000 / (64 (25 + 1))$$

$$= 9615$$

$$\text{Error} = (\text{Calculated Baud Rate} - \text{Desired Baud Rate}) / \text{Desired Baud Rate}$$

$$= (9615 - 9600) / 9600 = 0.16\%$$

TABLE 15-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	48
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	48
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	48
SPBRGH	EUSART Baud Rate Generator Register High Byte								48
SPBRG	EUSART Baud Rate Generator Register Low Byte								48

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

PIC18F1230/1330

15.2.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 15-6. The data is received on the RX pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

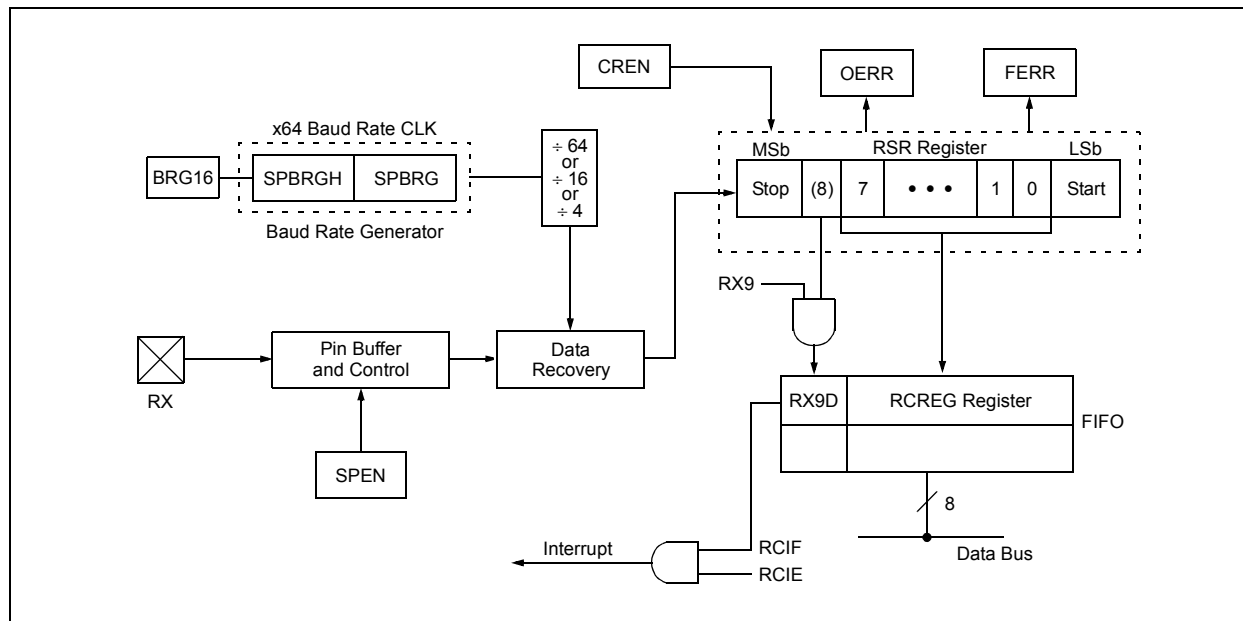
1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit RCIE.
4. If 9-bit reception is desired, set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCIE, was set.
7. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

15.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

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

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCIE and GIE bits are set.
8. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 15-6: EUSART RECEIVE BLOCK DIAGRAM



PIC18F1230/1330

15.3.2 EUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RX 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. If any error occurred, clear the error by clearing bit, CREN.
2. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.
3. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
4. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
5. Ensure bits, CREN and SREN, are clear.
6. If the signal from the CK pin is to be inverted, set the TXCKP bit.
7. If interrupts are desired, set enable bit, RCIE.
8. If 9-bit reception is desired, set bit, RX9.
9. If a single reception is required, set bit, SREN. For continuous reception, set bit, CREN.
10. Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if the enable bit, RCIE, was set.
11. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
12. Read the 8-bit received data by reading the RCREG register.

FIGURE 15-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

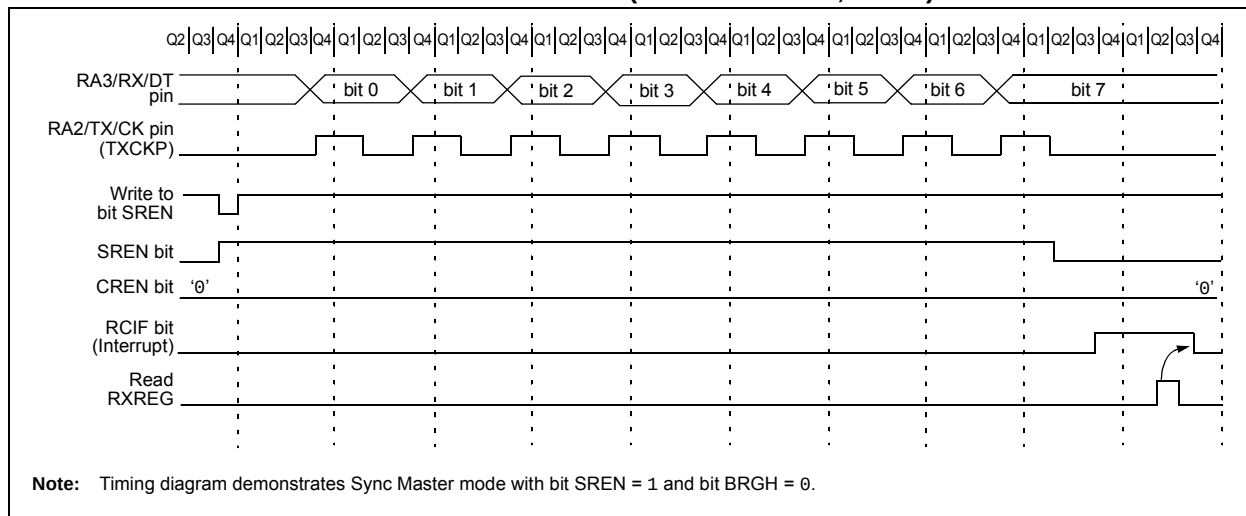


TABLE 15-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	47
PIR1	—	ADIF	RCIF	TXIF	CMP2IF	CMP1IF	CMP0IF	TMR1IF	49
PIE1	—	ADIE	RCIE	TXIE	CMP2IE	CMP1IE	CMP0IE	TMR1IE	49
IPR1	—	ADIP	RCIP	TXIP	CMP2IP	CMP1IP	CMP0IP	TMR1IP	49
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	48
RCREG	EUSART Receive Register								48
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	48
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	48
SPBRGH	EUSART Baud Rate Generator Register High Byte								48
SPBRG	EUSART Baud Rate Generator Register Low Byte								48

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception

PIC18F1230/1330

RETFIE Return from Interrupt

Syntax: RETFIE {s}

Operands: $s \in [0,1]$

Operation: (TOS) → PC,
 $1 \rightarrow \text{GIE/GIEH or PEIE/GIEL};$
 if $s = 1$,
 (WS) → W,
 (STATUS) → STATUS,
 (BSRS) → BSR,
 PCLATU, PCLATH are unchanged

Status Affected: GIE/GIEH, PEIE/GIEL

Encoding:

0000	0000	0001	000s
------	------	------	------

Description: Return from interrupt. Stack is popped and Top-of-Stack (TOS) is loaded into the PC. Interrupts are enabled by setting either the high or low-priority global interrupt enable bit. If 's' = 1, the contents of the shadow registers, WS, STATUS and BSRS, are loaded into their corresponding registers, W, STATUS and BSR. If 's' = 0, no update of these registers occurs.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	POP PC from stack Set GIEH or GIEL
No operation	No operation	No operation	No operation

Example: RETFIE 1

After Interrupt

PC	=	TOS
W	=	WS
BSR	=	BSRS
STATUS	=	STATUS
GIE/GIEH, PEIE/GIEL	=	1

RETLW Return Literal to W

Syntax: RETLW k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow W$,
 (TOS) → PC,
 PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	1100	kkkk	kkkk
------	------	------	------

Description: W is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	POP PC from stack, Write to W
No operation	No operation	No operation	No operation

Example:

```
CALL TABLE ; W contains table
              ; offset value
              ; W now has
              ; table value

:
TABLE
  ADDWF PCL ; W = offset
  RETLW k0 ; Begin table
  RETLW k1 ;
  :
  :
  RETLW kn ; End of table

Before Instruction
  W = 07h
After Instruction
  W = value of kn
```

SUBLW	Subtract W from Literal				
Syntax:	SUBLW k				
Operands:	$0 \leq k \leq 255$				
Operation:	$k - (W) \rightarrow W$				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table><tr><td>0000</td><td>1000</td><td>kkkk</td><td>kkkk</td></tr></table>	0000	1000	kkkk	kkkk
0000	1000	kkkk	kkkk		
Description	W is subtracted from the eight-bit literal 'k'. The result is placed in W.				
Words:	1				
Cycles:	1				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example 1: SUBLW 02h

Before Instruction	
W	= 01h
C	= ?
After Instruction	
W	= 01h
C	= 1 ; result is positive
Z	= 0
N	= 0

Example 2: SUBLW 02h

Before Instruction	
W	= 02h
C	= ?
After Instruction	
W	= 00h
C	= 1 ; result is zero
Z	= 1
N	= 0

Example 3: SUBLW 02h

Before Instruction	
W	= 03h
C	= ?
After Instruction	
W	= FFh ; (2's complement)
C	= 0 ; result is negative
Z	= 0
N	= 1

SUBWF	Subtract W from f				
Syntax:	SUBWF f {,d {,a}}				
Operands:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$				
Operation:	$(f) - (W) \rightarrow \text{dest}$				
Status Affected:	N, OV, C, DC, Z				
Encoding:	<table border="1"><tr><td>0101</td><td>11da</td><td>ffff</td><td>ffff</td></tr></table>	0101	11da	ffff	ffff
0101	11da	ffff	ffff		
Description:	Subtract W from register 'f' (2's				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: SUBWF REG, 1, 0

Before Instruction	
REG	= 3
W	= 2
C	= ?
After Instruction	
REG	= 1
W	= 2
C	= 1 ; result is positive
Z	= 0
N	= 0

Example 2: SUBWF REG, 0, 0

Before Instruction	
REG	= 2
W	= 2
C	= ?
After Instruction	
REG	= 2
W	= 0
C	= 1 ; result is zero
Z	= 1
N	= 0

Example 3: SUBWF REG, 1, 0

Before Instruction	
REG	= 1
W	= 2
C	= ?
After Instruction	
REG	= FFh ; (2's complement)
W	= 2
C	= 0 ; result is negative
Z	= 0
N	= 1

PIC18F1230/1330

TBLWT Table Write

Syntax: TBLWT (*, *+, *-, +*)

Operands: None

Operation: if TBLWT*,
(TABLAT) → Holding Register,
TBLPTR – No Change;
if TBLWT*+,
(TABLAT) → Holding Register,
(TBLPTR) + 1 → TBLPTR;
if TBLWT*-,
(TABLAT) → Holding Register,
(TBLPTR) – 1 → TBLPTR;
if TBLWT*+,
(TBLPTR) + 1 → TBLPTR,
(TABLAT) → Holding Register

Status Affected: None

Encoding:	0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*
-----------	------	------	------	---

Description: This instruction uses the 3 LSBs of TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 7.0 “Flash Program Memory”** for additional details on programming Flash memory.) The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSB of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word

TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLWT instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

TBLWT Table Write (Continued)

Example 1: TBLWT *+;

Before Instruction

TABLAT	=	55h
TBLPTR	=	00A356h
HOLDING REGISTER (00A356h)	=	FFh

After Instructions (table write completion)

TABLAT	=	55h
TBLPTR	=	00A357h
HOLDING REGISTER (00A356h)	=	55h

Example 2: TBLWT +*;

Before Instruction

TABLAT	=	34h
TBLPTR	=	01389Ah
HOLDING REGISTER (01389Ah)	=	FFh

After Instruction (table write completion)

TABLAT	=	34h
TBLPTR	=	01389Bh
HOLDING REGISTER (01389Ah)	=	FFh
HOLDING REGISTER (01389Bh)	=	34h

22.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, PIC18F1230/1330 devices also provide an optional extension to the core CPU functionality. The added features include eight additional instructions that augment indirect and indexed addressing operations and the implementation of Indexed Literal Offset Addressing mode for many of the standard PIC18 instructions.

The additional features of the extended instruction set are disabled by default. To enable them, users must set the XINST Configuration bit.

The instructions in the extended set (with the exception of CALLW, MOVSF and MOVSS) can all be classified as literal operations, which either manipulate the File Select Registers, or use them for indexed addressing. Two of the instructions, ADDFSR and SUBFSR, each have an additional special instantiation for using FSR2. These versions (ADDULNK and SUBULNK) allow for automatic return after execution.

The extended instructions are specifically implemented to optimize re-entrant program code (that is, code that is recursive or that uses a software stack) written in high-level languages, particularly C. Among other things, they allow users working in high-level languages to perform certain operations on data structures more efficiently. These include:

- Dynamic allocation and deallocation of software stack space when entering and leaving subroutines
- Function Pointer invocation
- Software Stack Pointer manipulation
- Manipulation of variables located in a software stack

A summary of the instructions in the extended instruction set is provided in Table 22-3. Detailed descriptions are provided in **Section 22.2.2 “Extended Instruction Set”**. The opcode field descriptions in Table 22-1 (page 216) apply to both the standard and extended PIC18 instruction sets.

Note: The instruction set extension and the Indexed Literal Offset Addressing mode were designed for optimizing applications written in C; the user may likely never use these instructions directly in the assembler. The syntax for these commands is provided as a reference for users who may be reviewing code that has been generated by a compiler.

22.2.1 EXTENDED INSTRUCTION SYNTAX

Most of the extended instructions use indexed arguments, using one of the File Select Registers and some offset to specify a source or destination register. When an argument for an instruction serves as part of indexed addressing, it is enclosed in square brackets (“[]”). This is done to indicate that the argument is used as an index or offset. The MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byte-oriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see **Section 22.2.3.1 “Extended Instruction Syntax with Standard PIC18 Commands”**.

Note: In the past, square brackets have been used to denote optional arguments in the PIC18 and earlier instruction sets. In this text and going forward, optional arguments are denoted by braces (“{ }”).

TABLE 22-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected
			MSb		LSb		
ADDFSR f, k	Add Literal to FSR	1	1110	1000	ffkk	kkkk	None
ADDULNK k	Add Literal to FSR2 and Return	2	1110	1000	11kk	kkkk	None
CALLW	Call Subroutine using WREG	2	0000	0000	0001	0100	None
MOVSF z _s , f _d	Move z _s (source) to 1st word f _d (destination) 2nd word	2	1110	1011	0zzz	zzzz	None
MOVSS z _s , z _d	Move z _s (source) to 1st word z _d (destination) 2nd word	2	1110	1011	1zzz	zzzz	None
PUSHL k	Store Literal at FSR2, Decrement FSR2	1	1110	1010	kkkk	kkkk	None
SUBFSR f, k	Subtract Literal from FSR	1	1110	1001	ffkk	kkkk	None
SUBULNK k	Subtract Literal from FSR2 and Return	2	1110	1001	11kk	kkkk	None

PIC18F1230/1330

23.2 DC Characteristics: Power-Down and Supply Current PIC18F1230/1330 (Industrial) PIC18LF1230/1330 (Industrial) (Continued)

PIC18LF1230/1330 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial					
PIC18F1230/1330 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended					
Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (I_{DD}) ⁽²⁾						
	PIC18LF1230/1330	165	347	μA	-40°C	$V_{DD} = 2.0\text{V}$	FOSC = 1 MHz (RC_IDLE mode, INTOSC source)
		175	347	μA	$+25^{\circ}\text{C}$		
		190	347	μA	$+85^{\circ}\text{C}$		
	PIC18LF1230/1330	250	497	μA	-40°C	$V_{DD} = 3.0\text{V}$	
		270	497	μA	$+25^{\circ}\text{C}$		
		290	497	μA	$+85^{\circ}\text{C}$		
	All devices	500	930	μA	-40°C	$V_{DD} = 5.0\text{V}$	
		520	930	μA	$+25^{\circ}\text{C}$		
		550	930	μA	$+85^{\circ}\text{C}$		
	Extended devices only	0.6	2.9	mA	$+125^{\circ}\text{C}$		
	PIC18LF1230/1330	340	497	μA	-40°C	$V_{DD} = 2.0\text{V}$	FOSC = 4 MHz (RC_IDLE mode, INTOSC source)
		350	497	μA	$+25^{\circ}\text{C}$		
		360	497	μA	$+85^{\circ}\text{C}$		
	PIC18LF1230/1330	520	830	μA	-40°C	$V_{DD} = 3.0\text{V}$	
		540	830	μA	$+25^{\circ}\text{C}$		
		580	830	μA	$+85^{\circ}\text{C}$		
	All devices	1.0	1.33	mA	-40°C	$V_{DD} = 5.0\text{V}$	
		1.1	1.33	mA	$+25^{\circ}\text{C}$		
		1.1	1.33	mA	$+85^{\circ}\text{C}$		
Extended devices only	1.1	5.0	mA	$+125^{\circ}\text{C}$			

- Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS} and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).
- 2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all I_{DD} measurements in active operation mode are:
 $OSC1$ = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD} or V_{SS} ;
 $MCLR = V_{DD}$; WDT enabled/disabled as specified.
- 3:** Low-power Timer1 oscillator selected.
- 4:** BOR and LVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

23.2 DC Characteristics: Power-Down and Supply Current PIC18F1230/1330 (Industrial) PIC18LF1230/1330 (Industrial) (Continued)

PIC18LF1230/1330 (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F1230/1330 (Industrial, Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (IDD) ⁽²⁾						
	PIC18LF1230/1330	250	497	μA	-40°C	VDD = 2.0V	FOSC = 1 MHz (PRI_RUN , EC oscillator)
		260	497	μA	+25°C		
		250	497	μA	+85°C		
	PIC18LF1230/1330	550	750	μA	-40°C	VDD = 3.0V	
		480	750	μA	+25°C		
		460	750	μA	+85°C		
	All devices	1.2	3	mA	-40°C	VDD = 5.0V	
		1.1	3	mA	+25°C		
		1.0	3	mA	+85°C		
	Extended devices only	1.0	3.0	mA	+125°C		
	PIC18LF1230/1330	0.72	1.93	mA	-40°C	VDD = 2.0V	FOSC = 4 MHz (PRI_RUN , EC oscillator)
		0.74	1.93	mA	+25°C		
		0.74	1.93	mA	+85°C		
	PIC18LF1230/1330	1.3	2.93	mA	-40°C	VDD = 3.0V	
		1.3	2.93	mA	+25°C		
		1.3	2.93	mA	+85°C		
	All devices	2.7	5.93	mA	-40°C	VDD = 5.0V	
		2.6	5.93	mA	+25°C		
		2.5	5.93	mA	+85°C		
	Extended devices only	2.6	7.0	mA	+125°C		
	Extended devices only	8.4	27.7	mA	+125°C	VDD = 4.2V	FOSC = 25 MHz (PRI_RUN , EC oscillator)
		11	27.7	mA	+125°C	VDD = 5.0V	
	All devices	15	26	mA	-40°C	VDD = 4.2V	FOSC = 40 MHz (PRI_RUN , EC oscillator)
		16	25	mA	+25°C		
		16	24	mA	+85°C		
	All devices	21	39.3	mA	-40°C	VDD = 5.0V	
		21	39.3	mA	+25°C		
21		39.3	mA	+85°C			

- Note 1:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS} and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).
- 2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all I_{DD} measurements in active operation mode are:
 $OSC1$ = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD} or V_{SS} ;
 $MCLR$ = V_{DD} ; WDT enabled/disabled as specified.
- 3:** Low-power Timer1 oscillator selected.
- 4:** BOR and LVD enable internal band gap reference. With both modules enabled, current consumption will be less than the sum of both specifications.

PIC18F1230/1330

FIGURE 23-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

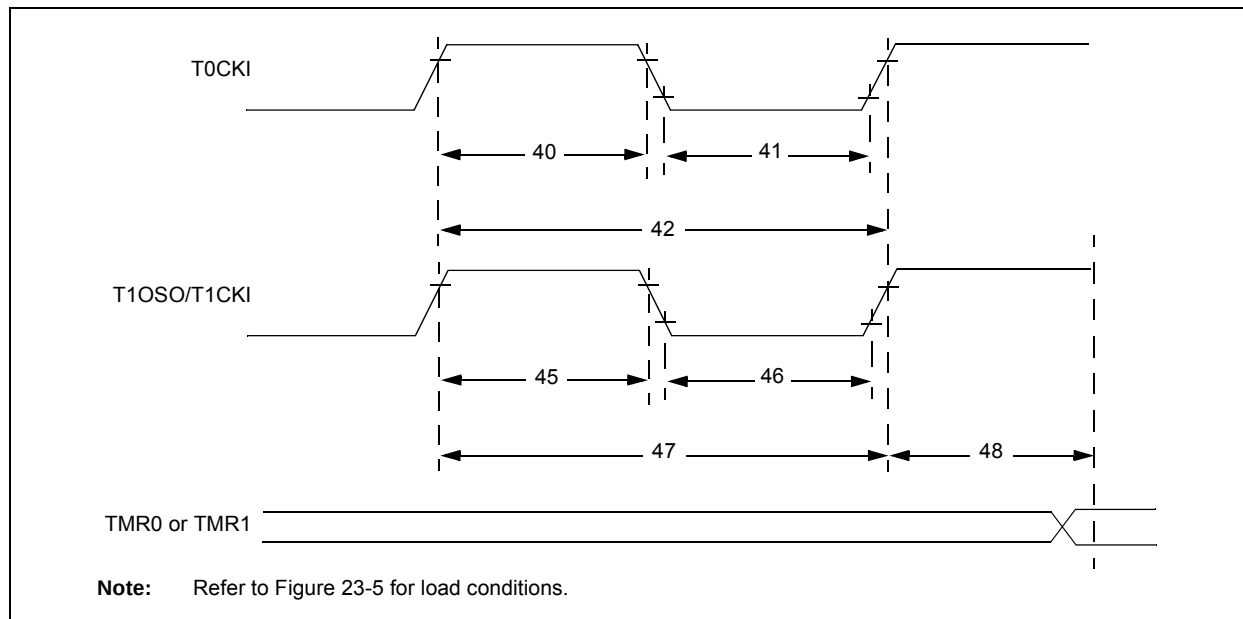


TABLE 23-11: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions	
40	Tt0H	T0CKI High Pulse Width	No prescaler	0.5 Tcy + 20	—	ns		
			With prescaler	10	—	ns		
41	Tt0L	T0CKI Low Pulse Width	No prescaler	0.5 Tcy + 20	—	ns		
			With prescaler	10	—	ns		
42	Tt0P	T0CKI Period	No prescaler	Tcy + 10	—	ns	N = prescale value (1, 2, 4,..., 256)	
			With prescaler	Greater of: 20 ns or (Tcy + 40)/N	—	ns		
45	Tt1H	T1CKI High Time	Synchronous, no prescaler		0.5 Tcy + 20	—	ns	VDD = 2.0V
			Synchronous, with prescaler	PIC18FXXXX	10	—	ns	
				PIC18LFXXXX	25	—	ns	
			Asynchronous	PIC18FXXXX	30	—	ns	
				PIC18LFXXXX	50	—	ns	
46	Tt1L	T1CKI Low Time	Synchronous, no prescaler		0.5 Tcy + 5	—	ns	VDD = 2.0V
			Synchronous, with prescaler	PIC18FXXXX	10	—	ns	
				PIC18LFXXXX	25	—	ns	
			Asynchronous	PIC18FXXXX	30	—	ns	
				PIC18LFXXXX	50	—	ns	
47	Tt1P	T1CKI Input Period	Synchronous		Greater of: 20 ns or (Tcy + 40)/N	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous		60	—	ns	
	Ft1	T1CKI Oscillator Input Frequency Range		DC	50	kHz		
48	Tcke2tmr1	Delay from External T1CKI Clock Edge to Timer Increment		2 Tosc	7 Tosc	—		

APPENDIX A: REVISION HISTORY

Revision A (November 2005)

Original data sheet for PIC18F1230/1330 devices.

Revision B (February 2006)

Data bank information was updated and a note was added for calculating the PCPWM duty cycle.

Revision C (March 2007)

Updated **Section 23.0 “Electrical Characteristics”** and **Section 24.0 “Packaging Information”**.

Revision D (November 2009)

Updated LIN 1.2 to LIN/J2602 throughout document along with minor corrections throughout document. Added the PIC18LF1230 and PIC18LF1330 devices. Refer to Table A-1 for additional revision history.

TABLE A-1: SECTION REVISION HISTORY

Section Name	Update Description
Section 1.0 “Device Overview”	Updated Table 1-2
Section 6.0 “Memory Organization”	Updated Table 6-2
Section 7.0 “Flash Program Memory”	Updated Section 7.2.4 “Table Pointer Boundaries” , Figure 7-3
Section 8.0 “Data EEPROM Memory”	Updated Section 8.2 “EECON1 and EECON2 Registers” , Section 8.8 “Using the Data EEPROM”
Section 10.0 “I/O Ports”	Updated Section 10.2 “PORTB, TRISB and LATB Registers”
Section 14.0 “Power Control PWM Module”	Updated Register 14-6, Section 14.11.2 “Output Polarity Control”
Section 15.0 “Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)”	Updated Register 15-3, Section 15.1 “Baud Rate Generator (BRG)” , Table 15-2, Section 15.1.3 “Auto-Baud Rate Detect” , Section 15.2 “EUSART Asynchronous Mode” , Table 15-5, Table 15-6, Section 15.3 “EUSART Synchronous Master Mode” , Figure 15-11, Table 15-7, Figure 15-13, Table 15-8, Table 15-9, Table 15-10
Section 16.0 “10-Bit Analog-to-Digital Converter (A/D) Module”	Updated Register 16-2
Section 17.0 “Comparator Module”	Updated Figure 17-2
Section 18.0 “Comparator Voltage Reference Module”	Updated Section 18.1 “Configuring the Comparator Voltage Reference” , Register 18-1, Figure 18-1
Section 20.0 “Special Features of the CPU”	Updated Register 20-6, Register 20-13, Register 20-14
Section 22.0 “Instruction Set Summary”	Updated Table 22-2
Section 23.0 “Electrical Characteristics”	Updated Table 23-1, Figure 23-3, Table 23-2, Table 23-3, Table 23-4, Table 23-5, Table 23-6, Table 23-8, Table 23-14, Table 23-15

PIC18F1230/1330

How to Clear RAM (Bank 0) Using Indirect Addressing	65	Data Memory	57
Implementing a Real-Time Clock Using a Timer1 Interrupt Service	115	Access Bank	59
Initializing PORTA	87	and the Extended Instruction Set	67
Initializing PORTB	90	Bank Select Register (BSR)	57
Reading a Flash Program Memory Word	75	General Purpose Registers	59
Saving STATUS, WREG and BSR Registers in RAM	105	Map for PIC18F1230/1330	58
Writing to Flash Program Memory	78–79	Special Function Registers	60
Code Protection	191, 207	DAW	234
Associated Registers	207	DC Characteristics	279
Configuration Register Protection	210	Power-Down and Supply Current	269
Data EEPROM	210	Supply Voltage	268
Program Memory	208	DCFSNZ	235
COMF	232	DECf	234
Comparator	179	DECFSZ	235
Analog Input Connection Considerations	181	Development Support	211
Associated Registers	182	Device Differences	304
Configuration	180	Device Overview	9
Effects of a Reset	181	Details on Individual Family Members	10
Interrupts	180	Features (table)	11
Operation	180	New Core Features	9
Operation During Sleep	181	Other Special Features	10
Outputs	180	Device Reset Timers	43
Reference	180	Oscillator Start-up Timer (OST)	43
Response Time	180	PLL Lock Time-out	43
Comparator Specifications	282	Power-up Timer (PWRT)	43
Comparator Voltage Reference	183	Time-out Sequence	43
Accuracy and Error	185	Direct Addressing	66
Associated Registers	185	E	
Configuring	183	Effect on Standard PIC MCU Instructions	262
Effects of a Reset	185	Effects of Power-Managed Modes on Various Clock Sources	29
Operation During Sleep	185	Electrical Characteristics	265
Computed GOTO	54	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART). See EUSART.	
Configuration Bits	191	Equations	
Context Saving During Interrupts	105	A/D Acquisition Time	174
Conversion Considerations	305	A/D Minimum Charging Time	174
CPFSEQ	232	Calculating the Minimum Required Acquisition Time	174
CPFSGT	233	PWM Frequency	129
CPFSLT	233	PWM Period for Continuous Up/Down Count Mode	129
Crystal Oscillator/Ceramic Resonator	21	PWM Period for Free-Running Mode	129
Customer Change Notification Service	314	PWM Resolution	129
Customer Notification Service	314	Errata	7
Customer Support	314	EUSART	
D		Asynchronous Mode	157
Data Addressing Modes	65	12-Bit Break Character Sequence	163
Comparing Options with the Extended Instruction Set		Associated Registers, Receive	161
Enabled	68	Associated Registers, Transmit	159
Direct	65	Auto-Wake-up on Sync Break Character	161
Indexed Literal Offset	67	Receiver	160
Instructions Affected	67	Receiving a Break Character	163
Indirect	65	Setting Up 9-Bit Mode with Address Detect	160
Inherent and Literal	65	Transmitter	157
Data EEPROM Memory	81	Baud Rate Generator	
Associated Registers	84	Operation in Power-Managed Modes	151
EEADR Register	81	Baud Rate Generator (BRG)	151
EECON1 and EECON2 Registers	81	Associated Registers	152
Operation During Code-Protect	84	Auto-Baud Rate Detect	155
Protection Against Spurious Write	83	Baud Rate Error, Calculating	152
Reading	83	Baud Rates, Asynchronous Modes	153
Using	84	High Baud Rate Select (BRGH Bit)	151
Write Verify	83	Sampling	151
Writing	83	Synchronous Master Mode	164
		Associated Registers, Receive	166

PIC18F1230/1330

Postscaler	125
Prescaler	125
Single-Shot Mode	125

R

RAM. See Data Memory.

RBIF Bit	90
RC Oscillator	23
RCIO Oscillator Mode	23
RC_IDLE Mode	37
RC_RUN Mode	33
RCALL	245
RCON Register	
Bit Status During Initialization	46
Reader Response	315
Register File Summary	61–63
Registers	
ADCON0 (A/D Control 0)	169
ADCON1 (A/D Control 1)	170
ADCON2 (A/D Control 2)	171
BAUDCON (Baud Rate Control)	150
CMCON (Comparator Control)	179
CONFIG1H (Configuration 1 High)	192
CONFIG2H (Configuration 2 High)	194
CONFIG2L (Configuration 2 Low)	193
CONFIG3H (Configuration 3 High)	196
CONFIG3L (Configuration 3 Low)	195
CONFIG4L (Configuration 4 Low)	197
CONFIG5H (Configuration 5 High)	198
CONFIG5L (Configuration 5 Low)	198
CONFIG6H (Configuration 6 High)	199
CONFIG6L (Configuration 6 Low)	199
CONFIG7H (Configuration 7 High)	200
CONFIG7L (Configuration 7 Low)	200
CVRCON (Comparator Voltage Reference Control)	184
DEVID1 (Device ID 1)	201
DEVID2 (Device ID 2)	201
DTCON (Dead-Time Control)	136
EECON1 (EEPROM Control 1)	73, 82
FLTCONFIG (Fault Configuration)	143
INTCON (Interrupt Control)	95
INTCON2 (Interrupt Control 2)	96
INTCON3 (Interrupt Control 3)	97
IPR1 (Peripheral Interrupt Priority 1)	102
IPR2 (Peripheral Interrupt Priority 2)	103
IPR3 (Peripheral Interrupt Priority 3)	103
LVDCON (Low-Voltage Detect Control)	187
OSCCON (Oscillator Control)	28
OSCTUNE (Oscillator Tuning)	25
OVDCOND (Output Override Control)	140
OVDCONS (Output State)	140
PIE1 (Peripheral Interrupt Enable 1)	100
PIE2 (Peripheral Interrupt Enable 2)	101
PIE3 (Peripheral Interrupt Enable 3)	101
PIR1 (Peripheral Interrupt Request (Flag) 1)	98
PIR2 (Peripheral Interrupt Request (Flag) 2)	99
PIR3 (Peripheral Interrupt Request (Flag) 3)	99
PTCON0 (PWM Timer Control 0)	122
PTCON1 (PWM Timer Control 1)	122
PWMCON0 (PWM Control 0)	123
PWMCON1 (PWM Control 1)	124
RCON (Reset Control)	40, 104
RCSTA (Receive Status and Control)	149
STATUS	64
STKPTR (Stack Pointer)	53
T0CON (Timer0 Control)	107

T1CON (Timer1 Control)	111
TXSTA (Transmit Status and Control)	148
WDTCON (Watchdog Timer Control)	203
RESET	245
Reset State of Registers	46
Resets	39, 191
Brown-out Reset (BOR)	191
Oscillator Start-up Timer (OST)	191
Power-on Reset (POR)	191
Power-up Timer (PWRT)	191
RETFIE	246
RETLW	246
RETURN	247
Return Address Stack	52
Associated Registers	52
Return Stack Pointer (STKPTR)	53
Revision History	303
RLCF	247
RLNCF	248
RRCF	248
RRNCF	249

S

SEC_IDLE Mode	36
SEC_RUN Mode	32
SETF	249
Single-Supply ICSP Programming	210
Single-Supply ICSP Programming	
SLEEP	250
Sleep	
OSC1 and OSC2 Pin States	29
Software Simulator (MPLAB SIM)	212
Special Features of the CPU	191
Special Function Registers	
Map	60
Stack Full/Underflow Resets	54
SUBFSR	261
SUBFWB	250
SUBLW	251
SUBULNK	261
SUBWF	251
SUBWFB	252
SWAPF	252

T

Table Reads/Table Writes	54
TBLRD	253
TBLWT	254
Time-out in Various Situations (table)	43
Timer0	107
16-Bit Mode Timer Reads and Writes	109
Associated Registers	109
Clock Source Edge Select (T0SE Bit)	109
Clock Source Select (T0CS Bit)	109
Interrupt	109
Operation	109
Prescaler	109
Switching the Assignment	109
Prescaler Assignment (PSA Bit)	109
Prescaler Select (T0PS2:T0PS0 Bits)	109
Prescaler. See Prescaler, Timer0.	

PIC18F1230/1330

NOTES: