



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

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

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1829-e-ml

Email: info@E-XFL.COM

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





Name	Function	Input Type	Output Type	Description		
RA0/AN0/CPS0/C1IN+/VREF-/	RA0	TTL	CMOS	General purpose I/O.		
	AN0	AN	_	A/D Channel 0 input.		
ICSPDAT/ICDDAT	CPS0	AN	—	Capacitive sensing input 0.		
	C1IN+	AN	—	Comparator C1 positive input.		
	VREF-	AN	—	A/D and DAC Negative Voltage Reference input.		
	DACOUT	_	AN	Digital-to-Analog Converter output.		
	TX	_	CMOS	USART asynchronous transmit.		
	CK	ST	CMOS	USART synchronous clock.		
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.		
	ICDDAT	ST	CMOS	In-Circuit Data I/O.		
RA1/AN1/CPS1/C12IN0-/VREF+/	RA1	TTL	CMOS	General purpose I/O.		
SRI/RX ⁽¹⁾ /DT ⁽¹⁾ /ICSPCLK/	AN1	AN	—	A/D Channel 1 input.		
ICDCLK	CPS1	AN	—	Capacitive sensing input 1.		
	C12IN0-	AN	—	Comparator C1 or C2 negative input.		
	VREF+	AN	—	A/D and DAC Positive Voltage Reference input.		
	SRI	ST	—	SR Latch input.		
	RX	ST	—	USART asynchronous input.		
	DT	ST	CMOS	USART synchronous data.		
	ICSPCLK	ST	—	Serial Programming Clock.		
	ICDCLK	ST	—	In-Circuit Debug Clock.		
RA2/AN2/CPS2/T0CKI/INT/	RA2	ST	CMOS	General purpose I/O.		
C1OUT/SRQ/CCP3/FLT0	AN2	AN	—	A/D Channel 2 input.		
	CPS2	AN	—	Capacitive sensing input 2.		
	TOCKI	ST	—	Timer0 clock input.		
	INT	ST	_	External interrupt.		
	C1OUT		CMOS	Comparator C1 output.		
	SRQ	—	CMOS	SR Latch non-inverting output.		
	CCP3	ST	CMOS	Capture/Compare/PWM3.		
	FLT0	ST	_	ECCP Auto-Shutdown Fault input.		
RA3/SS1 ⁽¹⁾ /T1G ⁽¹⁾ /VPP/MCLR	RA3	TTL	—	General purpose input.		
	SS1	ST	_	Slave Select input.		
	T1G	ST	—	Timer1 Gate input.		
	Vpp	HV	—	Programming voltage.		
	MCLR	ST		Master Clear with internal pull-up.		

TABLE 1-2: PIC16(L)F1825 PINOUT DESCRIPTION

Legend:AN = Analog input or outputCMOS = CMOS compatible input or outputOD= Open DrainTTL = TTL compatible inputST= Schmitt Trigger input with CMOS levels I^2C^{TM} = Schmitt Trigger input with I^2C HV = High VoltageXTAL = CrystalLevels

Note 1: Pin functions can be moved using the APFCON0 or APFCON1 register.

2: Default function location.

3.4 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figures 3-4 through 3-7). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are 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 if the STVREN bit is programmed to '0' (Configuration Word 2). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

Note 1: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

3.4.1 ACCESSING THE STACK

The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

Note:	Care should be taken when modifying the
	STKPTR while interrupts are enabled.

During normal program operation, CALL, CALLW and Interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. At any time STKPTR can be inspected to see how much stack is left. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will unload the PC and then decrement the STKPTR.

Reference Figure 3-4 through Figure 3-7 for examples of accessing the stack.

FIGURE 3-4: ACCESSING THE STACK EXAMPLE 1

TOSH:TOSL 0x0F	STKPTR = 0x1F Stack Reset Disabled (STVREN = 0)
` 0х0Е	
0x0D	
0x0C	
0x0B	
0x0A	Initial Staals Configuration
0x09	
0x08	After Reset, the stack is empty. The empty stack is initialized so the Stack
0x07	Pointer is pointing at 0x1F. If the Stack Overflow/Underflow Reset is enabled, the
0x06	TOSH/TOSL registers will return '0'. If the Stack Overflow/Underflow Reset is
0x05	disabled, the TOSH/TOSL registers will return the contents of stack address 0x0E
0x04	
0x03	
0x02	
0x01	
0x00	
TOSH:TOSL 0x1F 0x0000	STKPTR = 0x1F Stack Reset Enabled (STVREN = 1)

FIGURE 3-5: ACCESSING THE STACK EXAMPLE 2



5.2.1.4 4xPLL

The oscillator module contains a 4xPLL that can be used with both external and internal clock sources to provide a system clock source. The input frequency for the 4xPLL must fall within specifications. See the PLL Clock Timing Specifications in **Section 30.0 "Electrical Specifications**".

The 4xPLL may be enabled for use by one of two methods:

- 1. Program the PLLEN bit in Configuration Word 2 to a '1'.
- Write the SPLLEN bit in the OSCCON register to a '1'. If the PLLEN bit in Configuration Word 2 is programmed to a '1', then the value of SPLLEN is ignored.

5.2.1.5 TIMER1 Oscillator

The Timer1 Oscillator is a separate crystal oscillator that is associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the T1OSO and T1OSI device pins.

The Timer1 Oscillator can be used as an alternate system clock source and can be selected during run-time using clock switching. Refer to **Section 5.3 "Clock Switching"** for more information.

FIGURE 5-5: QUARTZ CRYSTAL

OPERATION (TIMER1 OSCILLATOR)



- Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.
 - **2:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.
 - **3:** For oscillator design assistance, reference the following Microchip Application Notes:
 - AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC[®] and PIC[®] Devices" (DS00826)
 - AN849, "Basic PIC[®] Oscillator Design" (DS00849)
 - AN943, "Practical PIC[®] Oscillator Analysis and Design" (DS00943)
 - AN949, "Making Your Oscillator Work" (DS00949)
 - TB097, "Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS" (DS91097)
 - AN1288, "Design Practices for Low-Power External Oscillators" (DS01288)

5.2.1.6 External RC Mode

The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required.

The RC circuit connects to OSC1. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. The function of the OSC2/CLKOUT pin is determined by the state of the CLKOUTEN bit in Configuration Word 1.

Figure 5-6 shows the external RC mode connections.

REGISTER 12-9: PORTB: PORTB REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4		—		_
bit 7							bit 0
Legend:							
R = Readable I	oit	W = Writable	bit	U = Unimpler	mented bit, read	as '0'	
u = Bit is unchanged		x = Bit is unknown		-n/n = Value at POR and BOR/Value at all other Resets			ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				

bit 7-4	RB<7:4> : PORTB General Purpose I/O Pin bits
	1 = Port pin is <u>></u> Vін
	0 = Port pin is <u><</u> VI ∟
bit 3-0	Unimplemented: Read as '0'

REGISTER 12-10: TRISB: PORTB TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0	
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	
bit 7 bit 0								

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **TRISB<7:4>:** PORTB Tri-State Control bits 1 = PORTB pin configured as an input (tri-stated) 0 = PORTB pin configured as an output

bit 3-0 Unimplemented: Read as '0'

REGISTER 12-11: LATB: PORTB DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	
LATB7	LATB6	LATB5	LATB4	—	—	—	—	
bit 7 bit 0								

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 LATB<7:4>: PORTB Output Latch Value bits⁽¹⁾

bit 3-0 Unimplemented: Read as '0'

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

16.2.6 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

- 1. Configure Port:
 - Disable pin output driver (Refer to the TRIS register)
 - Configure pin as analog (Refer to the ANSEL register)
- 2. Configure the ADC module:
 - Select ADC conversion clock
 - Configure voltage reference
 - Select ADC input channel
 - Turn on ADC module
- 3. Configure ADC interrupt (optional):
 - Clear ADC interrupt flag
 - Enable ADC interrupt
 - Enable peripheral interrupt
 - Enable global interrupt⁽¹⁾
- 4. Wait the required acquisition time⁽²⁾.
- 5. Start conversion by setting the GO/DONE bit.
- 6. Wait for ADC conversion to complete by one of the following:
 - Polling the GO/DONE bit
 - Waiting for the ADC interrupt (interrupts enabled)
- 7. Read ADC Result.
- 8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: Refer to Section 16.3 "A/D Acquisition Requirements".

EXAMPLE 16-1: A/D CONVERSION

;This code block configures the ADC
;for polling, Vdd and Vss references, Frc
;clock and ANO input.
;

;Conversion start & polling for completion ; are included.

;						
BANKSEL	ADCON1	;				
MOVLW	B'11110000'	;Right justify, Frc				
		;clock				
MOVWF	ADCON1	;Vdd and Vss Vref				
BANKSEL TRISA		;				
BSF TRISA,0		;Set RA0 to input				
BANKSEL ANSEL		;				
BSF ANSEL,0		;Set RA0 to analog				
BANKSEL ADCON0		;				
MOVLW B'0000001'		;Select channel ANO				
MOVWF ADCON0		;Turn ADC On				
CALL	SampleTime	;Acquisiton delay				
BSF	ADCON0, ADGO	;Start conversion				
BTFSC	ADCON0, ADGO	;Is conversion done?				
GOTO	\$-1	;No, test again				
BANKSEL	ADRESH	;				
MOVF	ADRESH,W	;Read upper 2 bits				
MOVWF RESULTHI		;store in GPR space				
BANKSEL ADRESL		;				
MOVF	ADRESL,W	;Read lower 8 bits				
MOVWF	RESULTLO	;Store in GPR space				

R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u		
MDCLODIS	MDCLPOL	MDCLSYNC	_		MDCL	_<3:0>			
bit 7							bit 0		
Legend:									
R = Readable	bit	W = Writable b	it	U = Unimplen	nented bit, read	l as '0'			
u = Bit is unchanged		x = Bit is unkno	own	-n/n = Value a	at POR and BO	R/Value at all c	other Resets		
'1' = Bit is set		'0' = Bit is clea	red						
 bit 7 MDCLODIS: Modulator Low Carrier Output Disable bit 1 = Output signal driving the peripheral output pin (selected by MDCL<3:0> of the MDCARL register is disabled 0 = Output signal driving the peripheral output pin (selected by MDCL<3:0> of the MDCARL register is enabled 									
bit 6	 MDCLPOL: Modulator Low Carrier Polarity Select bit 1 = Selected low carrier signal is inverted 0 = Selected low carrier signal is not inverted 								
bit 5	<pre>MDCLSYNC 1 = Modulate time carr 0 = Modulate</pre>	: Modulator Low or waits for a fallin rier or Output is not s	Carrier Syn ng edge on t synchronized	chronization En he low time carr d to the low time	able bit ier signal before e carrier signal ^{(*}	e allowing a sw 1)	itch to the high		
bit 4	Unimplemer	ted: Read as '0							
bit 3-0	MDCL<3:0> 1111 = Res 0111 = CCF 0110 = CCF 0101 = CCF 0101 = CCF 0101 = Ref 0011 = Ref 0001 = MDC 0001 = MDC	Modulator Data erved. No chann P4 output (PWM P3 output (PWM P2 output (PWM P1 output (PWM P1 output (PWM CIN2 port pin CIN1 port pin	High Carrier nel connecte Output mod Output mod Output mod dule signal	Selection bits (ed. ed. le only) le only) le only) le only)	1)				

REGISTER 23-4: MDCARL: MODULATION LOW CARRIER CONTROL REGISTER

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

TABLE 23-1:	SUMMARY OF REGISTERS ASSOCIATED WITH DATA SIGNAL MODULATOR MODE
-	

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
MDCARH	MDCHODIS	MDCHPOL	MDCHSYNC	—		199			
MDCARL	MDCLODIS	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				200
MDCON	MDEN	MDOE	MDSLR	MDOPOL	MDOUT	—	—	MDBIT	197
MDSRC	MDMSODIS		—	—	MDMS<3:0>				198

Legend: — Unimplemented, read as '0'. Shaded cells are not used in the Data Signal Modulator mode.







25.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

- a) A low level is sampled on SDAx when SCLx goes from low level to high level. (CASE 1)
- b) SCLx goes low before SDAx is asserted low, indicating that another master is attempting to transmit a data '1'. (CASE 2)

When the user releases SDAx and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCLx pin is then deasserted and when sampled high, the SDAx pin is sampled. If SDAx is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0' (Figure 25-36). If SDAx is sampled high, the BRG is reloaded and begins counting. If SDAx goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDAx at exactly the same time.

If SCLx goes from high-to-low before the BRG times out and SDAx has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (Figure 25-37).

If, at the end of the BRG time-out, both SCLx and SDAx are still high, the SDAx pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCLx pin, the SCLx pin is driven low and the Repeated Start condition is complete.

FIGURE 25-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)



FIGURE 25-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



26.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

- · Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- · Programmable 8-bit or 9-bit character length
- · Address detection in 9-bit mode
- · Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- · Half-duplex synchronous slave
- Programmable clock polarity in synchronous modes
- · Sleep operation

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- · Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- · 13-bit Break character transmit

Block diagrams of the EUSART transmitter and receiver are shown in Figure 26-1 and Figure 26-2.

FIGURE 26-1: EUSART TRANSMIT BLOCK DIAGRAM



26.3 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCON register selects 16-bit mode.

The SPBRGH, SPBRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCON register. In Synchronous mode, the BRGH bit is ignored.

Table 26-3 contains the formulas for determining the baud rate. Example 26-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 26-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

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

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

EXAMPLE 26-1: CALCULATING BAUD RATE ERROR

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

Desired Baud Rate = $\frac{FOSC}{64([SPBRGH:SPBRGL] + 1)}$

Solving for SPBRGH:SPBRGL:

C

$X = \frac{Fosc}{\frac{Desired Baud Rate}{64} - 1}$
$=\frac{\frac{16000000}{9600}}{64}-1$
$= [25.042] = 25$ alculated Baud Rate $= \frac{16000000}{64(25+1)}$
= 9615
Error = $\frac{Calc. Baud Rate - Desired Baud Rate}{Desired Baud Rate}$
$= \frac{(9615 - 9600)}{9600} = 0.16\%$

26.4 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

26.4.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for Synchronous Master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART.

26.4.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

26.4.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCON register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

26.4.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXREG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

Note: The TSR register is not mapped in data memory, so it is not available to the user.

- 26.4.1.4 Synchronous Master Transmission Setup:
- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section 26.3 "EUSART Baud Rate Generator (BRG)").
- 2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
- 3. Disable Receive mode by clearing bits SREN and CREN.
- 4. Enable Transmit mode by setting the TXEN bit.
- 5. If 9-bit transmission is desired, set the TX9 bit.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
- 8. Start transmission by loading data to the TXREG register.

RETFIE	Return from Interrupt						
Syntax:	[label] RETFIE						
Operands:	None						
Operation:	$\begin{array}{l} TOS \to PC, \\ 1 \to GIE \end{array}$						
Status Affected:	None						
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.						
Words:	1						
Cycles:	2						
Example:	RETFIE						
	After Interrupt PC = TOS GIE = 1						

RETURN	Return from Subroutine					
Syntax:	[label] RETURN					
Operands:	None					
Operation:	$TOS \rightarrow PC$					
Status Affected:	None					
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.					

RETLW	Return with literal in W		Pototo Loft f through Corry			
Syntax:	[<i>label</i>] RETLW k		Rotate Left I through Carry			
Operands:	$0 \le k \le 255$	Syntax:	[<i>label</i>] RLF f,d			
Operation:	$k \rightarrow (W);$ TOS \rightarrow PC	Operands:	$0 \le f \le 127$ $d \in [0,1]$			
Status Affected:	None	Operation:	See description below			
Description:	The W register is leaded with the 8 bit	Status Affected:	С			
Description.	literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.	Description:	The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is			
Words:	1		stored back in register T.			
Cycles:	2					
Example:	CALL TABLE;W contains table	Words:	1			
	<pre>;offset value ,W now has table value</pre>	Cycles:	1			
TABLE	•	Example:	RLF REG1,0			
	• ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • RETLW kn ; End of table		Before Instruction REG1 = 1110 0110 C = 0 -<			
	Before Instruction W = 0x07 After Instruction W = value of k8					

PIC16LF1825/9		$\begin{tabular}{lllllllllllllllllllllllllllllllllll$							
PIC16F1825/9			$\begin{array}{llllllllllllllllllllllllllllllllllll$						
Param. No.	Sym.	Characteristic	Min. Typ† Max. Units				Conditions		
D001	Vdd	Supply Voltage (VDDMIN, VDDMAX)							
		PIC16LF1825/9	1.8 2.5		3.6 3.6	V V	Fosc ≤ 16 MHz: Fosc ≤ 32 MHz (Note 2)		
D001		PIC16F1825/9	1.8 2.5	-	5.5 5.5	V V	Fosc ≤ 16 MHz: Fosc ≤ 32 MHz (Note 2)		
D002* VDR RAM Data Retention Voltage ⁽¹⁾									
	PIC16LF1825/9	1.5	_	-	V	Device in Sleep mode			
D002*		PIC16F1825/9	1.7	—	—	V	Device in Sleep mode		
	VPOR*	Power-on Reset Release Voltage	_	1.6	_	V			
D002B*	VPORR*	Power-on Reset Rearm Voltage							
		PIC16LF1825/9	—	0.8	—	V	Device in Sleep mode		
		PIC16F1825/9	_	1.5	_	V	Device in Sleep mode		
D003	VADFVR	Fixed Voltage Reference Voltage for ADC	-8	_	6	%	1.024V, VDD ≥ 2.5V 2.048V, VDD ≥ 2.5V 4.096V, VDD ≥ 4.75V		
D003A	VCDAFVR	Fixed Voltage Reference Voltage for Comparator and DAC	-11	_	7	%	$\begin{array}{l} 1.024V, \ VDD \geq 2.5V \\ 2.048V, \ VDD \geq 2.5V \\ 4.096V, \ VDD \geq 4.75V \end{array}$		
D003C*	TCVFVR	Temperature Coefficient, Fixed Voltage Reference	_	-130	_	ppm/°C			
D003D*	$\Delta VFVR/$ ΔVIN	Line Regulation, Fixed Voltage Reference	—	0.270	—	%/V			
D004*	SVDD	VDD Rise Rate to ensure internal Power-on Reset signal	0.05	—	_	V/ms	See Section 7.1 "Power-on Reset (POR)" for details.		

30.1 DC Characteristics: PIC16(L)F1825/9-I/E (Industrial, Extended)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

2: PLL required for 32 MHz operation.



3.5 Max 3.0 Typical: 25°C 2.5 Typical 2.0 Ibb (mA) 1.5 1.0 0.5 0.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 VDD (V)

FIGURE 31-20: IDD, HS OSCILLATOR, 32 MHz (8 MHz + 4xPLL), PIC16F1825/9 ONLY

FIGURE 31-35: IPD, CAPACITIVE SENSING (CPS) MODULE, HIGH-CURRENT RANGE, CPSRM = 0, PIC16LF1825/9 ONLY



FIGURE 31-36: IPD, CAPACITIVE SENSING (CPS) MODULE, HIGH-CURRENT RANGE, CPSRM = 0, PIC16F1825/9 ONLY





FIGURE 31-55: COMPARATOR HYSTERESIS, LOW-POWER MODE (CxSP = 0, CxHYS = 1)



32.0 DEVELOPMENT SUPPORT

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

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

32.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows[®], Linux and Mac $OS^{®}$ X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- · Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- · Call graph window
- Project-Based Workspaces:
- Multiple projects
- Multiple tools
- · Multiple configurations
- · Simultaneous debugging sessions

File History and Bug Tracking:

- · Local file history feature
- Built-in support for Bugzilla issue tracker



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

	Units			MILLIMETERS			
Dimension	Dimension Limits			MAX			
Number of Pins	Ν	20					
Pitch	е		0.65 BSC				
Overall Height	А	-	-	2.00			
Molded Package Thickness	A2	1.65	1.75	1.85			
Standoff	A1	0.05	-	-			
Overall Width	E	7.40	7.80	8.20			
Molded Package Width	E1	5.00	5.30	5.60			
Overall Length	D	6.90	7.20	7.50			
Foot Length	L	0.55	0.75	0.95			
Footprint	L1	1.25 REF					
Lead Thickness	С	0.09	-	0.25			
Foot Angle	φ	0°	4°	8°			
Lead Width	b	0.22	-	0.38			

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- 3. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B