



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	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
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 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4580-i-ml

Email: info@E-XFL.COM

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

NOTES:

4.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. There are two cases:

- PRI_IDLE mode where the primary clock source is not stopped; and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these instances, the primary clock source either does not require an oscillator start-up delay, since it is already running (PRI_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes). However, a fixed delay of interval, TCSD, following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

TABLE 4-2:EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE
(BY CLOCK SOURCES)

Clock Source Before Wake-up	Clock Source After Wake-up	Exit Delay	Clock Ready Status bit (OSCCON)	
	LP, XT, HS			
	HSPLL		OSTS	
(PRI IDLE mode)	EC, RC	Tcsd ⁽²⁾		
	INTRC ⁽¹⁾		—	
	INTOSC ⁽³⁾		IOFS	
	LP, XT, HS	Tost ⁽⁴⁾		
	HSPLL	Tost + t _{rc} (4)	OSTS	
T1OSC or INTRC ⁽¹⁾	EC, RC	Toop(2)		
	INTRC ⁽¹⁾	ICSD-7	_	
	INTOSC ⁽³⁾	TIOBST ⁽⁵⁾	IOFS	
	LP, XT, HS	Tost ⁽⁵⁾		
	HSPLL	Tost + t _{rc} (4)	OSTS	
INTOSC ⁽³⁾	EC, RC	Toop(2)		
	INTRC ⁽¹⁾	ICSD ⁽)	—	
	INTOSC ⁽³⁾	None	IOFS	
	LP, XT, HS	Tost ⁽⁴⁾		
	HSPLL	Tost + t _{rc} (4)	OSTS	
None (Sleen mode)	EC, RC	Toop(2)		
	INTRC ⁽¹⁾	ICSD'-'	_	
[INTOSC ⁽³⁾	TIOBST ⁽⁵⁾	IOFS	

Note 1: In this instance, refers specifically to the 31 kHz INTRC clock source.

2: TCSD (parameter 38) is a required delay when waking from Sleep and all Idle modes and runs concurrently with any other required delays (see Section 4.4 "Idle Modes").

3: Includes both the INTOSC 8 MHz source and postscaler derived frequencies.

4: TOST is the Oscillator Start-up Timer (parameter 32). t_{rc} is the PLL Lock-out Timer (parameter F12); it is also designated as TPLL.

5: Execution continues during TIOBST (parameter 39), the INTOSC stabilization period.

5.4 Brown-out Reset (BOR)

PIC18F2480/2580/4480/4580 devices implement a BOR circuit that provides the user with a number of configuration and power-saving options. The BOR is controlled by the BORV<1:0> and BOREN<1:0> Configuration bits. There are a total of four BOR configurations which are summarized in Table 5-1.

The BOR threshold is set by the BORV<1:0> bits. If BOR is enabled (any values of BOREN<1:0>, except '00'), any drop of VDD below VBOR (parameter D005) for greater than TBOR (parameter 35) will reset the device. A Reset may or may not occur if VDD falls below VBOR for less than TBOR. The chip will remain in Brown-out Reset until VDD rises above VBOR.

If the Power-up Timer is enabled, it will be invoked after VDD rises above VBOR; it then will keep the chip in Reset for an additional time delay, TPWRT (parameter 33). If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above VBOR, the Power-up Timer will execute the additional time delay.

BOR and the Power-on Timer (PWRT) are independently configured. Enabling a Brown-out Reset does not automatically enable the PWRT.

5.4.1 SOFTWARE ENABLED BOR

When BOREN<1:0> = 01, the BOR can be enabled or disabled by the user in software. This is done with the control bit, SBOREN (RCON<6>). Setting SBOREN enables the BOR to function as previously described. Clearing SBOREN disables the BOR entirely. The SBOREN bit operates only in this mode; otherwise it is read as '0'.

Placing the BOR under software control gives the user the additional flexibility of tailoring the application to its environment without having to reprogram the device to change BOR configuration. It also allows the user to tailor device power consumption in software by eliminating the incremental current that the BOR consumes. While the BOR current is typically very small, it may have some impact in low-power applications.

Note:	Even when BOR is under software control,
	the Brown-out Reset voltage level is still
	set by the BORV<1:0> Configuration bits.
	It cannot be changed in software.

5.4.2 DETECTING BOR

When Brown-out Reset is enabled, the BOR bit always resets to '0' on any Brown-out Reset or Power-on Reset event. This makes it difficult to determine if a Brown-out Reset event has occurred just by reading the state of BOR alone. A more reliable method is to simultaneously check the state of both POR and BOR. This assumes that the POR bit is reset to '1' in software immediately after any Power-on Reset event. IF BOR is '0' while POR is '1', it can be reliably assumed that a Brown-out Reset event has occurred.

5.4.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, the BOR remains under hardware control and operates as previously described. Whenever the device enters Sleep mode, however, the BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code, when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

BOR Con	figuration	Status of	
BOREN1	BOREN0	SBOREN (RCON<6>)	BOR Operation
0	0	Unavailable	BOR disabled; must be enabled by reprogramming the Configuration bits.
0	1	Available	BOR enabled in software; operation controlled by SBOREN.
1	0	Unavailable	BOR enabled in hardware in Run and Idle modes, disabled during Sleep mode.
1	1	Unavailable	BOR enabled in hardware; must be disabled by reprogramming the Configuration bits.

TABLE 5-1:BOR CONFIGURATIONS

	1						- ,
Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt		
BSR	2480	2580	4480	4580	0000	0000	uuuu
INDF2	2480	2580	4480	4580	N/A	N/A	N/A
POSTINC2	2480	2580	4480	4580	N/A	N/A	N/A
POSTDEC2	2480	2580	4480	4580	N/A	N/A	N/A
PREINC2	2480	2580	4480	4580	N/A	N/A	N/A
PLUSW2	2480	2580	4480	4580	N/A	N/A	N/A
FSR2H	2480	2580	4480	4580	0000	0000	uuuu
FSR2L	2480	2580	4480	4580	XXXX XXXX	սսսս սսսս	սսսս սսսս
STATUS	2480	2580	4480	4580	x xxxx	u uuuu	u uuuu
TMR0H	2480	2580	4480	4580	0000 0000	0000 0000	սսսս սսսս
TMR0L	2480	2580	4480	4580	XXXX XXXX	นนนน นนนน	սսսս սսսս
T0CON	2480	2580	4480	4580	1111 1111	1111 1111	นนนน นนนน
OSCCON	2480	2580	4480	4580	0100 q000	0100 00q0	uuuu uuqu
HLVDCON	2480	2580	4480	4580	0-00 0101	0-00 0101	0-uu uuuu
WDTCON	2480	2580	4480	4580	0	0	u
RCON ⁽⁴⁾	2480	2580	4480	4580	0q-1 11q0	0q-q qquu	uq-u qquu
TMR1H	2480	2580	4480	4580	XXXX XXXX	սսսս սսսս	սսսս սսսս
TMR1L	2480	2580	4480	4580	XXXX XXXX	นนนน นนนน	սսսս սսսս
T1CON	2480	2580	4480	4580	0000 0000	u0uu uuuu	սսսս սսսս
TMR2	2480	2580	4480	4580	0000 0000	0000 0000	սսսս սսսս
PR2	2480	2580	4480	4580	1111 1111	1111 1111	1111 1111
T2CON	2480	2580	4480	4580	-000 0000	-000 0000	-uuu uuuu
SSPBUF	2480	2580	4480	4580	XXXX XXXX	սսսս սսսս	սսսս սսսս
SSPADD	2480	2580	4480	4580	0000 0000	0000 0000	սսսս սսսս
SSPSTAT	2480	2580	4480	4580	0000 0000	0000 0000	սսսս սսսս
SSPCON1	2480	2580	4480	4580	0000 0000	0000 0000	นนนน นนนน
SSPCON2	2480	2580	4480	4580	0000 0000	0000 0000	นนนน นนนน
ADRESH	2480	2580	4480	4580	XXXX XXXX	นนนน นนนน	սսսս սսսս
ADRESL	2480	2580	4480	4580	XXXX XXXX	นนนน นนนน	นนนน นนนน
ADCON0	2480	2580	4480	4580	00 0000	00 0000	uu uuuu
ADCON1	2480	2580	4480	4580	00 0qqq	00 0qqq	uu uuuu

TABLE 5-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 5-3 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: This register reads all '0's until ECAN[™] technology is set up in Mode 1 or Mode 2.

6.4 Data Addressing Modes

Note: The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See Section 6.6 "Data Memory and the Extended Instruction Set" for more information.

While the program memory can be addressed in only one way – through the program counter – information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in greater detail in **Section 6.6.1 "Indexed Addressing with Literal Offset**".

6.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all; they either perform an operation that globally affects the device or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples include SLEEP, RESET and DAW.

Other instructions work in a similar way but require an additional explicit argument in the opcode. This is known as Literal Addressing mode because they require some literal value as an argument. Examples include ADDLW and MOVLW which, respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

6.4.2 DIRECT ADDRESSING

Direct Addressing specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit literal address as their Least Significant Byte. This address specifies either a register address in one of the banks of data RAM (Section 6.3.3 "General **Purpose Register File**") or a location in the Access Bank (Section 6.3.2 "Access Bank") as the data source for the instruction.

The Access RAM bit 'a' determines how the address is interpreted. When 'a' is '1', the contents of the BSR (Section 6.3.1 "Bank Select Register (BSR)") are used with the address to determine the complete 12-bit address of the register. When 'a' is '0', the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit, 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction; their destination is either the target register being operated on or the W register.

6.4.3 INDIRECT ADDRESSING

Indirect Addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as Special File Registers, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures, such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code, using loops, such as the example of clearing an entire RAM bank in Example 6-5.

EXAMPLE 6-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

	LFSR	FSR0, 100h	;	
NEXT	CLRF	POSTINCO	;	Clear INDF
			;	register then
			;	inc pointer
	BTFSS	FSROH,1	;	All done with
			;	Bank1?
	BRA	NEXT	;	NO, clear next
CONTINU	Έ		;	YES, continue

NOTES:

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0					
OSCFIF	CMIF ⁽¹⁾	_	EEIF	BCLIF	HLVDIF	TMR3IF	ECCP1IF ⁽¹⁾					
bit 7	ł						bit 0					
Legend:												
R = Readable	e bit	W = Writable	bit	U = Unimple	mented bit, rea	d as '0'						
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cle	eared	x = Bit is unk	known					
DIT /		cillator Fall Intel	rupt Flag bit	an abangad ta		the cleared in	ooffwara)					
	0 = System 0	clock operating	CIOCK INPUL I	ias changeu to	in 1030 (inus	t be cleared in	soltware)					
bit 6	CMIF: Comp	arator Interrupt	Flag bit ⁽¹⁾									
	1 = Compara	ator input has c	hanged (mus	t be cleared in	software)							
	0 = Compara	ator input has n	ot changed									
bit 5	Unimplemer	nted: Read as '	0'									
bit 4	EEIF: Data E	EEIF: Data EEPROM/Flash Write Operation Interrupt Flag bit										
	1 = The write	1 = The write operation is complete (must be cleared in software) 0 = The write operation is not complete, or has not been started										
bit 3	BCLIF: Bus (Collision Interru	ot Complete,		i starteu							
	1 = A bus collision occurred (must be cleared in software)											
	0 = No bus collision occurred											
bit 2	HLVDIF: Higl	HLVDIF: High/Low-Voltage Detect Interrupt Flag bit										
	1 = A low-vo	1 = A low-voltage condition occurred (must be cleared in software)										
L:1 4		ice voltage is al	bove the Higr	i/Low-voitage i	Jetect trip poin	t						
DICI	1 - TMP3 re	TMR3IF: TMR3 Overflow Interrupt Flag bit										
	0 = TMR3 re	1 = TMR3 register did not overflow										
bit 0	ECCP1IF: CO	ECCP1IF: CCPx Interrupt Flag bit ⁽¹⁾										
	Capture mod	Capture mode:										
	1 = A TMR1 0 = No TMR	register captur 1 register captu	e occurred (n ire occurred	nust be cleared	l in software)							
	Compare mo	<u>de:</u>	wa waatab aaa	unred (never be	alaarad in aaff							
	$\perp = A \square V R \square$ 0 = No TMR	1 register compa	are match occ	curred (must be	cieared in soft	ware)						
	PWM mode:	egictor comp										
	Unused in thi	<u>Privit mode.</u> Unused in this mode.										

Note 1: These bits are available in PIC18F4X80 and reserved in PIC18F2X80 devices.

11.6 Parallel Slave Port

Note:	The Parallel Slave Port is only available on
	PIC18F4X80 devices.

In addition to its function as a general I/O port, PORTD can also operate as an 8-bit wide Parallel Slave Port (PSP) or microprocessor port. PSP operation is controlled by the 4 upper bits of the TRISE register (Register 11-1). Setting control bit, PSPMODE (TRISE<4>), enables PSP operation, as long as the Enhanced CCP module is not operating in dual output or quad output PWM mode. In Slave mode, the port is asynchronously readable and writable by the external world.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting the control bit PSPMODE enables the PORTE I/O pins to become control inputs for the microprocessor port. When set, port pin RE0 is the RD input, RE1 is the WR input and RE2 is the CS (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set). The A/D port Configuration bits, PFCG<3:0> (ADCON1<3:0>), must also be set to '1010'.

A write to the PSP occurs when both the \overline{CS} and \overline{WR} lines are first detected low and ends when either are detected high. The PSPIF and IBF flag bits are both set when the write ends.

A read from the PSP occurs when both the \overline{CS} and \overline{RD} lines are first detected low. The data in PORTD is read out and the OBF bit is set. If the user writes new data to PORTD to set OBF, the data is immediately read out; however, the OBF bit is not set.

When either the \overline{CS} or \overline{RD} lines are detected high, the PORTD pins return to the input state and the PSPIF bit is set. User applications should wait for PSPIF to be set before servicing the PSP; when this happens, the IBF and OBF bits can be polled and the appropriate action taken. The timing for the control signals in Write and Read modes is shown in Figure 11-3 and Figure 11-4, respectively.

FIGURE 11-2: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)





FIGURE 11-4: PARALLEL SLAVE PORT READ WAVEFORMS



TABLE 11-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:	
PORTD ⁽¹⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	58	
LATD ⁽¹⁾	LATD Outp	LATD Output Latch Register								
TRISD ⁽¹⁾	PORTD Da	ta Direction R	egister						58	
PORTE ⁽¹⁾	—	_	_	—	RE3	RE2	RE1	RE0	58	
LATE ⁽¹⁾	—	—	—	—	—	LATE Outp	58			
TRISE ⁽¹⁾	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	58	
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55	
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	58	
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	58	
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	58	
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	56	
CMCON ⁽¹⁾	C2OUT	C10UT	C2INV	C1INV	CIS	CM2	CM1	CM0	57	

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

Note 1: These registers are available on PIC18F4X80 devices only.

17.4.5 FULL-BRIDGE MODE

In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin P1A is continuously active and pin P1D is modulated. In the Reverse mode, pin P1C is continuously active and pin P1B is modulated. These are illustrated in Figure 17-6. P1A, P1B, P1C and P1D outputs are multiplexed with the PORTD<4>, PORTD<5>, PORTD<6> and PORTD<7> data latches. The TRISD<4>, TRISD<5>, TRISD<6> and TRISD<7> bits must be cleared to make the P1A, P1B, P1C and P1D pins outputs.





18.4.3.2 Reception

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

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

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

If SEN is enabled (SSPCON2<0> = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPCON<4>). See **Section 18.4.4** "Clock **Stretching**" for more details.

18.4.3.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low regardless of SEN (see Section 18.4.4 "Clock Stretching" for more details). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then, the RC3/ SCK/SCL pin should be enabled by setting bit, CKP (SSPCON1<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 18-9).

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

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.



19.2.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 19-6. 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.
- 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.

19.2.3 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 19-7: ASYNCHRONOUS RECEPTION



TABLE 19-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS 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	55
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	58
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREG	EUSART F	Receive Reg	ister						57
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	57
SPBRGH	EUSART Baud Rate Generator Register, High Byte								
SPBRG	EUSART E	Baud Rate G	enerator Re	gister, Low	Byte				57

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

Note 1: Reserved in PIC18F2X80 devices; always maintain these bits clear.

REGISTER 24-4:	COMSTAT: COMMUNICATION STATUS REGISTER
----------------	--

Mode 0	R/C-0	R/C-0	R-0	R-0	R-0	R-0	R-0	R-0				
woue u	RXB00VFL	RXB10VFL	ТХВО	TXBP	RXBP	TXWARN	RXWARN	EWARN				
	D/0.0	D/0.0		D 0			D 0	D 0				
Mode 1	R/C-0			R-0	K-U							
	—	RABIOVEL	IXBU	IXBP	RYRL	IXWARN	RAWARN	EWARN				
	R/C-0	R/C-0	R-0	R-0	R-0	R-0	R-0	R-0				
Mode 2	FIFOEMPTY	RXBnOVFL	ТХВО	TXBP	RXBP	TXWARN	RXWARN	EWARN				
	bit 7							bit 0				
Legend			C = Clearab	le bit								
R = Rea	dable bit		W = Writable	e bit	U = Unimpl	emented bit, r	read as '0'					
-n = Valu	ue at POR		'1' = Bit is se	et	'0' = Bit is c	leared	x = Bit is unk	nown				
								-				
bit 7	Mode 0: RXB0OVFL:	Receive Buffe	r 0 Overflow	bit								
	1 = Receive 0 = Receive	Buffer 0 overflo Buffer 0 has no	owed ot overflowed	1								
	<u>Mode 1:</u> Unimplemer	nted: Read as	'O'									
	Mode 2:											
	FIFOEMPTY	FIFOEMPTY: FIFO Not Empty bit										
	1 = Receive 0 = Receive	FIFO is not em FIFO is empty	pty									
bit 6	Mode 0: RXB10VFL:	Mode 0: RXB10VFL: Receive Buffer 1 Overflow bit										
	1 = Receive 0 = Receive	1 = Receive Buffer 1 overflowed0 = Receive Buffer 1 has not overflowed										
	Mode 1, 2:											
	RXBnOVFL: Receive Buffer n Overflow bit											
	1 = Receive Buffer n has overflowed 0 = Receive Buffer n has not overflowed											
bit 5	TXBO: Trans	smitter Bus-Off	bit									
	1 = Transmit 0 = Transmit	error counter a	> 255 ≤ 255									
bit 4	TXBP: Trans	mitter Bus Pas	sive bit									
	1 = Transmit 0 = Transmit	error counter a	> 127 ≤ 127									
bit 3	RXBP: Rece	iver Bus Passi	ve bit									
	1 = Receive 0 = Receive	error counter > error counter ≤	127 127									
bit 2	TXWARN: Tr	TXWARN: Transmitter Warning bit										
	1 = Transmit 0 = Transmit	error counter a	> 95 ≤ 95									
bit 1	RXWARN: R	eceiver Warnir	ng bit									
	1 = 127 ≥ Re 0 = Receive	eceive error cou error counter ≤	unter > 95 95									
bit 0	EWARN: Err This bit is a f	or Warning bit lag of the RXW	/ARN and T>	WARN bits.								
	1 = The RXV 0 = Neither th	VARN or the T	KWARN bits r the TXWAF	are set RN bits are se	et							

REGISTER 24-21: RXERRCNT: RECEIVE ERROR COUNT REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
bit 7							bit 0
l agand.							

Legena.			
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 7-0 **REC<7:0>:** Receive Error Counter bits

This register contains the receive error value as defined by the CAN specifications. When RXERRCNT > 127, the module will go into an error-passive state. RXERRCNT does not have the ability to put the module in "bus-off" state.

EXAMPLE 24-5: READING A CAN MESSAGE

; Need to read a pending message from RXB0 buffer. ; To receive any message, filter, mask and RXM1:RXM0 bits in RXB0CON registers must be ; programmed correctly. ; Make sure that there is a message pending in RXBO. BTFSS RXBOCON, RXFUL ; Does RXB0 contain a message? BRA NoMessage ; No. Handle this situation... ; We have verified that a message is pending in RXBO buffer. ; If this buffer can receive both Standard or Extended Identifier messages, ; identify type of message received. ; Is this Extended Identifier? BTFSS RXBOSIDL, EXID BRA StandardMessage ; No. This is Standard Identifier message. ; Yes. This is Extended Identifier message. ; Read all 29-bits of Extended Identifier message. . . . ; Now read all data bytes MOVFF RXB0DO, MY DATA BYTE1 . . . ; Once entire message is read, mark the RXBO that it is read and no longer FULL. BCF RXB0CON, RXFUL ; This will allow CAN Module to load new messages ; into this buffer. . . .

24.3.4 LISTEN ONLY MODE

Listen Only mode provides a means for the PIC18F2480/2580/4480/4580 devices to receive all messages, including messages with errors. This mode can be used for bus monitor applications or for detecting the baud rate in 'hot plugging' situations. For auto-baud detection, it is necessary that there are at least two other nodes which are communicating with each other. The baud rate can be detected empirically by testing different values until valid messages are received. The Listen Only mode is a silent mode, meaning no messages will be transmitted while in this state, including error flags or Acknowledge signals. The filters and masks can be used to allow only particular messages to be loaded into the receive registers or the filter masks can be set to all zeros to allow a message with any identifier to pass. The error counters are reset and deactivated in this state. The Listen Only mode is activated by setting the mode request bits in the CANCON register.

24.3.5 LOOPBACK MODE

This mode will allow internal transmission of messages from the transmit buffers to the receive buffers without actually transmitting messages on the CAN bus. This mode can be used in system development and testing. In this mode, the ACK bit is ignored and the device will allow incoming messages from itself, just as if they were coming from another node. The Loopback mode is a silent mode, meaning no messages will be transmitted while in this state, including error flags or Acknowledge signals. The TXCAN pin will revert to port I/O while the device is in this mode. The filters and masks can be used to allow only particular messages to be loaded into the receive registers. The masks can be set to all zeros to provide a mode that accepts all messages. The Loopback mode is activated by setting the mode request bits in the CANCON register.

24.3.6 ERROR RECOGNITION MODE

The module can be set to ignore all errors and receive any message. In functional Mode 0, the Error Recognition mode is activated by setting the RXM<1:0> bits in the RXBnCON registers to '11'. In this mode, the data which is in the message assembly buffer until the error time, is copied in the receive buffer and can be read via the CPU interface.

24.4 CAN Module Functional Modes

In addition to CAN modes of operation, the ECAN module offers a total of 3 functional modes. Each of these modes are identified as Mode 0, Mode 1 and Mode 2.

24.4.1 MODE 0 – LEGACY MODE

Mode 0 is designed to be fully compatible with CAN modules used in PIC18CXX8 and PIC18FXX8 devices. This is the default mode of operation on all Reset conditions. As a result, module code written for the PIC18XX8 CAN module may be used on the ECAN module without any code changes.

The following is the list of resources available in Mode 0:

- Three transmit buffers: TXB0, TXB1 and TXB2
- Two receive buffers: RXB0 and RXB1
- Two acceptance masks, one for each receive buffer: RXM0, RXM1
- Six acceptance filters, 2 for RXB0 and 4 for RXB1: RXF0, RXF1, RXF2, RXF3, RXF4, RXF5

24.4.2 MODE 1 – ENHANCED LEGACY MODE

Mode 1 is similar to Mode 0, with the exception that more resources are available in Mode 1. There are 16 acceptance filters and two acceptance mask registers. Acceptance Filter 15 can be used as either an acceptance filter or an acceptance mask register. In addition to three transmit and two receive buffers, there are six more message buffers. One or more of these additional buffers can be programmed as transmit or receive buffers. These additional buffers can also be programmed to automatically handle RTR messages.

Fourteen of sixteen acceptance filter registers can be dynamically associated to any receive buffer and acceptance mask register. One can use this capability to associate more than one filter to any one buffer.

When a receive buffer is programmed to use standard identifier messages, part of the full acceptance filter register can be used as a data byte filter. The length of the data byte filter is programmable from 0 to 18 bits. This functionality simplifies implementation of high-level protocols, such as the DeviceNet[™] protocol.

The following is the list of resources available in Mode 1:

- Three transmit buffers: TXB0, TXB1 and TXB2
- · Two receive buffers: RXB0 and RXB1
- Six buffers programmable as TX or RX: B0-B5
- · Automatic RTR handling on B0-B5
- Sixteen dynamically assigned acceptance filters: RXF0-RXF15
- Two dedicated acceptance mask registers; RXF15 programmable as third mask: RXM0-RXM1, RXF15
- Programmable data filter on standard identifier messages: SDFLC

24.7.3 ENHANCED FIFO MODE

When configured for Mode 2, two of the dedicated receive buffers in combination with one or more programmable transmit/receive buffers, are used to create a maximum of an 8 buffer deep FIFO buffer. In this mode, there is no direct correlation between filters and receive buffer registers. Any filter that has been enabled can generate an acceptance. When a message has been accepted, it is stored in the next available receive buffer register and an internal Write Pointer is incremented. The FIFO can be a maximum of 8 buffers deep. The entire FIFO must consist of contiguous receive buffers. The FIFO head begins at RXB0 buffer and its tail spans toward B5. The maximum length of the FIFO is limited by the presence or absence of the first transmit buffer starting from B0. If a buffer is configured as a transmit buffer, the FIFO length is reduced accordingly. For instance, if B3 is configured as a transmit buffer, the actual FIFO will consist of RXB0, RXB1, B0, B1 and B2, a total of 5 buffers. If B0 is configured as a transmit buffer, the FIFO length will be 2. If none of the programmable buffers are configured as a transmit buffer, the FIFO will be 8 buffers deep. A system that requires more transmit buffers should try to locate transmit buffers at the very end of B0-B5 buffers to maximize available FIFO length.

When a message is received in FIFO mode, the interrupt flag code bits (EICODE<4:0>) in the CANSTAT register will have a value of '10000', indicating the FIFO has received a message. FIFO Pointer bits, FP<3:0> in the CANCON register, point to the buffer that contains data not yet read. The FIFO Pointer bits, in this sense, serve as the FIFO Read Pointer. The user should use FP bits and read corresponding buffer data. When receive data is no longer needed, the RXFUL bit in the current buffer must be cleared, causing FP<3:0> to be updated by the module.

To determine whether FIFO is empty or not, the user may use the FP<3:0> bits to access the RXFUL bit in the current buffer. If RXFUL is cleared, the FIFO is considered to be empty. If it is set, the FIFO may contain one or more messages. In Mode 2, the module also provides a bit called FIFO High Water Mark (FIFOWM) in the ECANCON register. This bit can be used to cause an interrupt whenever the FIFO contains only one or four empty buffers. The FIFO high water mark interrupt can serve as an early warning to a full FIFO condition.

24.7.4 TIME-STAMPING

The CAN module can be programmed to generate a time-stamp for every message that is received. When enabled, the module generates a capture signal for CCP1, which in turn captures the value of either Timer1 or Timer3. This value can be used as the message time-stamp.

To use the time-stamp capability, the CANCAP bit (CIOCON<4>) must be set. This replaces the capture input for CCP1 with the signal generated from the CAN module. In addition, CCP1CON<3:0> must be set to '0011' to enable the CCP Special Event Trigger for CAN events.

24.8 Message Acceptance Filters and Masks

The message acceptance filters and masks are used to determine if a message in the Message Assembly Buffer should be loaded into any of the receive buffers. Once a valid message has been received into the MAB, the identifier fields of the message are compared to the filter values. If there is a match, that message will be loaded into the appropriate receive buffer. The filter masks are used to determine which bits in the identifier are examined with the filters. A truth table is shown below in Table 24-2 that indicates how each bit in the identifier is compared to the masks and filters to determine if a message should be loaded into a receive buffer. The mask essentially determines which bits to apply the acceptance filters to. If any mask bit is set to a zero, then that bit will automatically be accepted regardless of the filter bit.

TABLE 24-2:	FILTER/MASK TRUTH TABLE
-------------	-------------------------

Mask bit n	Filter bit n	Message Identifier bit n001	Accept or Reject bit n
0	Х	х	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

Legend: x = don't care

In Mode 0, acceptance filters, RXF0 and RXF1, and filter mask, RXM0, are associated with RXB0. Filters, RXF2, RXF3, RXF4 and RXF5, and mask, RXM1, are associated with RXB1.



Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH	$\overline{SS} \downarrow$ to SCK \downarrow or SCK \uparrow Input		3 Тсү	-	ns	
	, TssL2scL						
71	TscH	SCK Input High Time	Continuous	1.25 Tcy + 30		ns	
71A			Single Byte	40		ns	(Note 1)
72	TscL	SCK Input Low Time	Continuous	1.25 Tcy + 30		ns	
72A			Single Byte	40	—	ns	(Note 1)
73	TDIV2scH, TDIV2scL	Setup Time of SDI Data Input to SCK E	20		ns		
73A	Тв2в	Last Clock Edge of Byte1 to the First Clock Edge of Byte 2		1.5 Tcy + 40	—	ns	(Note 2)
74	TscH2DIL, TscL2DIL	Hold Time of SDI Data Input to SCK Edge		40		ns	
75	75 TDOR SDO Data Output Rise Time		PIC18FXXXX	_	25	ns	
			PIC18LFXXXX		45	ns	VDD = 2.0V
76	TdoF	SDO Data Output Fall Time		—	25	ns	
77	TssH2doZ	SS ↑ to SDO Output High-Impedance		10	50	ns	
80	TscH2doV	CH2DOV SDO Data Output Valid after SCK			50	ns	
, Tso	, TscL2doV	Edge	PIC18LFXXXX		100	ns	VDD = 2.0V
83	TscH2ssH , TscL2ssH	SS ↑ after SCK Edge		1.5 Tcy + 40		ns	

TABLE 28-16: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

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

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

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length





		MILLIM	AILLIMETERS DM MAX BSC 4.25 4.25 4.25 70 70 0.37 0.37	
Dimension	MIN	NOM	MAX	
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2105A