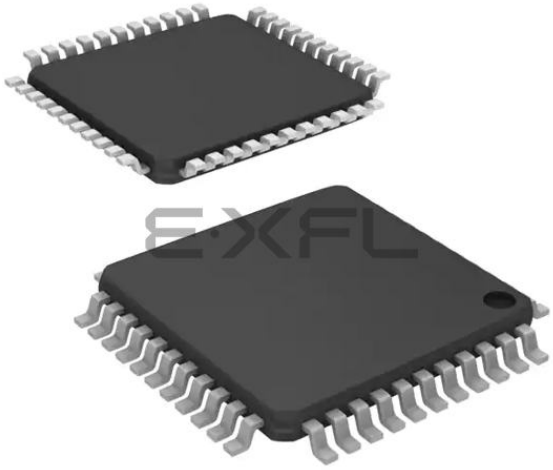


Welcome to E-XFL.COM

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 "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf45j11-i-pt

TABLE 1-4: PIC18F4XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	44-QFN	44-TQFP			
					PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/AN12/INT0/RP3	9	8			
RB0			I/O	DIG	Digital I/O.
AN12			I	Analog	Analog input 12.
INT0			I	ST	External interrupt 0.
RP3			I/O	DIG	Remappable peripheral pin 3.
RB1/AN10/PMBE/RTCC/RP4	10	9			
RB1			I/O	DIG	Digital I/O.
AN10			I	Analog	Analog input 10.
PMBE			O	DIG	Parallel Master Port byte enable.
RTCC			O	DIG	Real Time Clock Calendar output.
RP4			I/O	DIG	Remappable peripheral pin 4.
RB2/AN8/CTED1/PMA3/REFO/RP5	11	10			
RB2			I/O	DIG	Digital I/O.
AN8			I	Analog	Analog input 8.
CTED1			I	ST	CTMU edge 1 input.
PMA3			O	DIG	Parallel Master Port address.
REFO			O	DIG	Reference output clock.
RP5			I/O	DIG	Remappable peripheral pin 5.
RB3/AN9/CTED2/PMA2/RP6	12	11			
RB3			I/O	DIG	Digital I/O.
AN9			I	Analog	Analog input 9.
CTED2			I	ST	CTMU edge 2 input.
PMA2			O	DIG	Parallel Master Port address.
RP6			I/O	DIG	Remappable peripheral pin 6.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD}); DIG = Digital output

Note 1: RA7 and RA6 will be disabled if OSC1 and OSC2 are used for the clock function.

6.0 MEMORY ORGANIZATION

There are two types of memory in PIC18 Flash microcontrollers:

- Program Memory
- Data RAM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces.

Section 7.0 "Flash Program Memory" provides additional information on the operation of the Flash program memory.

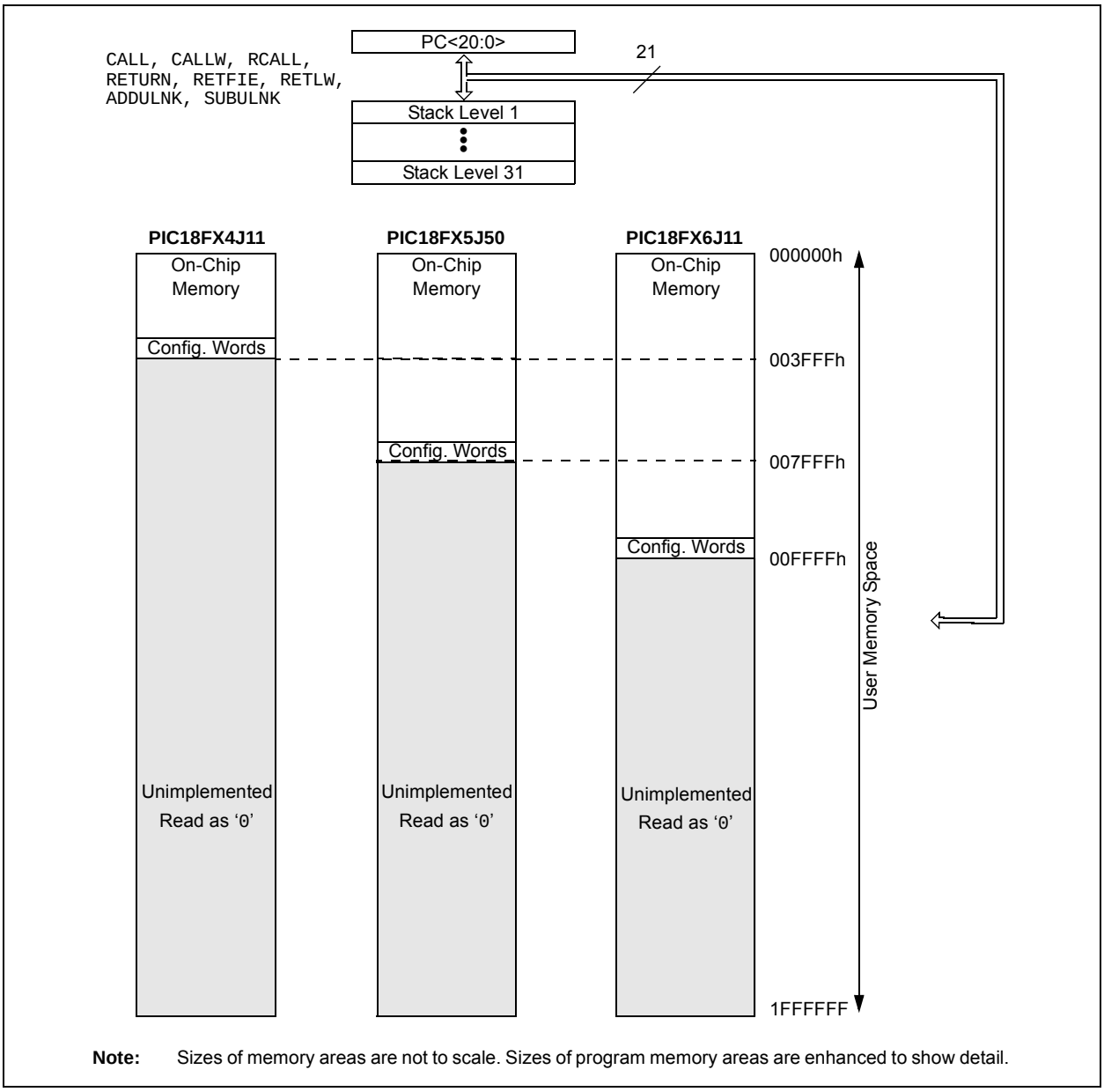
6.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address returns all '0's (a NOP instruction).

The PIC18F46J11 family offers a range of on-chip Flash program memory sizes, from 16 Kbytes (up to 8,192 single-word instructions) to 64 Kbytes (32,768 single-word instructions).

Figure 6-1 provides the program memory maps for individual family devices.

FIGURE 6-1: MEMORY MAPS FOR PIC18F46J11 FAMILY DEVICES



Note: Sizes of memory areas are not to scale. Sizes of program memory areas are enhanced to show detail.

6.1.4.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration register 1L. When STVREN is set, a full or underflow condition sets the appropriate STKFUL or STKUNF bit and then causes a device Reset. When STVREN is cleared, a full or underflow condition sets the appropriate STKFUL or STKUNF bit, but does not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a POR.

6.1.5 FAST REGISTER STACK (FRS)

A Fast Register Stack (FRS) is provided for the STATUS, WREG and BSR registers to provide a "fast return" option for interrupts. This stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources push values into the Stack registers. The values in the registers are then loaded back into the working registers if the RETFIE, FAST instruction is used to return from the interrupt.

If both low-priority and high-priority interrupts are enabled, the Stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the Stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the FRS for returns from interrupt. If no interrupts are used, the FRS can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a CALL label, FAST instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A RETURN, FAST instruction is then executed to restore these registers from the FRS.

Example 6-1 provides a source code example that uses the FRS during a subroutine call and return.

EXAMPLE 6-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST    ;STATUS, WREG, BSR
                   ;SAVED IN FAST REGISTER
                   ;STACK
       •
       •

SUB1   •
       •
   RETURN FAST     ;RESTORE VALUES SAVED
                   ;IN FAST REGISTER STACK
```

6.1.6 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures or look-up tables in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed GOTO
- Table Reads

6.1.6.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the PC. An example is shown in Example 6-2.

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW nn instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next executed instruction will be one of the RETLW nn instructions that returns the value 'nn' to the calling function.

The offset value (in WREG) specifies the number of bytes that the PC should advance and should be multiples of 2 (LSb = 0).

In this method, only one byte may be stored in each instruction location; room on the return address stack is required.

EXAMPLE 6-2: COMPUTED GOTO USING AN OFFSET VALUE

```
        MOVF    OFFSET, W
        CALL    TABLE
ORG     nn00h
TABLE   ADDWF   PCL
        RETLW   nnh
        RETLW   nnh
        RETLW   nnh
        .
        .
        .
```

6.1.6.2 Table Reads

A better method of storing data in program memory allows two bytes to be stored in each instruction location.

Look-up table data may be stored two bytes per program word while programming. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from the program memory. Data is transferred from program memory one byte at a time.

Table read operation is discussed further in **Section 7.1 "Table Reads and Table Writes"**.

7.5.2 FLASH PROGRAM MEMORY WRITE SEQUENCE (WORD PRORAMMING).

The PIC18F46J11 family of devices has a feature that allows programming a single word (two bytes). This feature is enabled when the WPROG bit is set. If the memory location is already erased, the following sequence is required to enable this feature:

1. Load the Table Pointer register with the address of the data to be written. (It must be an even address.)
2. Write the 2 bytes into the holding registers by performing table writes. (Do not post-increment on the second table write.)
3. Set the WREN bit (EECON1<2>) to enable writes and the WPROG bit (EECON1<5>) to select Word Write mode.
4. Disable interrupts.
5. Write 55h to EECON2.
6. Write 0AAh to EECON2.
7. Set the WR bit; this will begin the write cycle.
8. The CPU will stall for the duration of the write for TIW (see parameter D133A).
9. Re-enable interrupts.

EXAMPLE 7-4: SINGLE-WORD WRITE TO FLASH PROGRAM MEMORY

```
                MOVLW   CODE_ADDR_UPPER       ; Load TBLPTR with the base address
                MOVWF   TBLPTRU
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW         ; The table pointer must be loaded with an even
                                                address
                MOVWF   TBLPTRL

                MOVLW   DATA0                 ; LSB of word to be written
                MOVWF   TABLAT
                TBLWT*+
                MOVLW   DATA1                 ; MSB of word to be written
                MOVWF   TABLAT
                TBLWT*                        ; The last table write must not increment the table
                                                pointer! The table pointer needs to point to the
                                                MSB before starting the write operation.


PROGRAM_MEMORY
                BSF     EECON1, WPROG         ; enable single word write
                BSF     EECON1, WREN          ; enable write to memory
                BCF     INTCON, GIE           ; disable interrupts
                MOVLW   0x55
    Required    MOVWF   EECON2                ; write 55h
    Sequence    MOVLW   0xAA
                MOVWF   EECON2                ; write AAh
                BSF     EECON1, WR            ; start program (CPU stall)
                BSF     INTCON, GIE           ; re-enable interrupts
                BCF     EECON1, WPROG         ; disable single word write
                BCF     EECON1, WREN          ; disable write to memory
```

9.1 INTCON Registers

The INTCON registers are readable and writable registers, which contain various enable, priority and flag bits.

Note:	Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER (ACCESS FF2h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
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 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1 and GIEH = 1:
1 = Enables all low-priority peripheral interrupts
0 = Disables all low-priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB<7:4> pins changed state (must be cleared in software)
0 = None of the RB<7:4> pins have changed state

Note 1: A mismatch condition will continue to set this bit. Reading PORTB and waiting 1 TCY will end the mismatch condition and allow the bit to be cleared.

10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to five ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

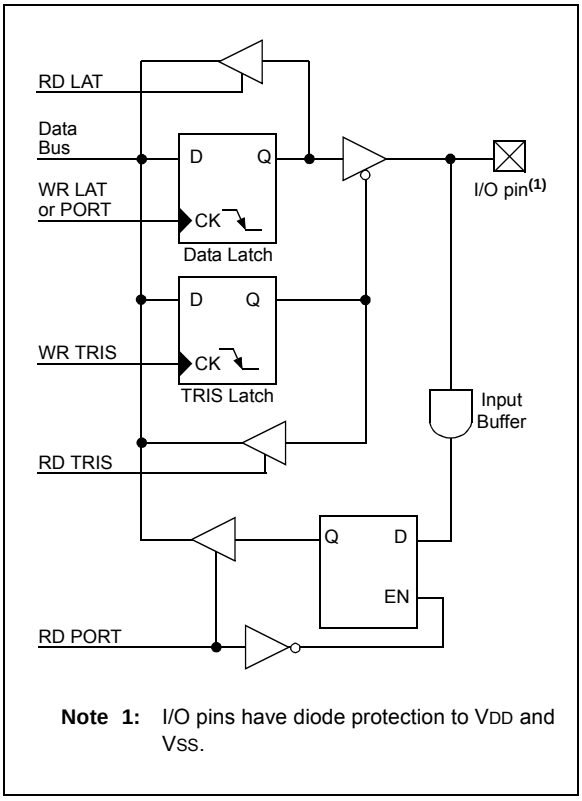
Each port has three registers for its operation. These registers are:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Data Latch)

The Data Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving.

Figure 10-1 displays a simplified model of a generic I/O port, without the interfaces to other peripherals.

FIGURE 10-1: GENERIC I/O PORT OPERATION



Note 1: I/O pins have diode protection to VDD and VSS.

10.1 I/O Port Pin Capabilities

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than VDD input levels.

10.1.1 PIN OUTPUT DRIVE

The output pin drive strengths vary for groups of pins intended to meet the needs for a variety of applications. PORTB and PORTC are designed to drive higher loads, such as LEDs. All other ports are designed for small loads, typically indication only. Table 10-1 summarizes the output capabilities. Refer to **Section 29.0 "Electrical Characteristics"** for more details.

TABLE 10-1: OUTPUT DRIVE LEVELS

Port	Drive	Description
PORTA (except RA6)	Minimum	Intended for indication.
PORTD		
PORTE		
PORTB	High	Suitable for direct LED drive levels.
PORTC		
PORTA<6>		

10.1.2 INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages up to 5.5V; a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should be avoided. Table 10-2 summarizes the input capabilities. Refer to **Section 29.0 "Electrical Characteristics"** for more details.

TABLE 10-2: INPUT VOLTAGE LEVELS

Port or Pin	Tolerated Input	Description
PORTA<7:0>	VDD	Only VDD input levels tolerated.
PORTB<3:0>		
PORTC<2:0>		
PORTE<2:0>		
PORTB<7:4>	5.5V	Tolerates input levels above VDD, useful for most standard logic.
PORTC<7:3>		
PORTD<7:0>		

TABLE 10-5: PORTB I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RB4/PMA1/ KBI0/RP7	RB4	0	O	DIG	LATB<4> data output; not affected by analog input.
		1	I	TTL	PORTB<4> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input enabled.(1)
	PMA1(3)	0	O	DIG	Parallel Master Port address.
		1	I	ST/TTL	Parallel Slave Port address input.
	KBI0	1	I	TTL	Interrupt-on-change pin.
	RP7	1	I	ST	Remappable peripheral pin 7 input.
		0	O	DIG	Remappable peripheral pin 7 output.
RB5/PMA0/ KBI1/RP8	RB5	0	O	DIG	LATB<5> data output.
		1	I	TTL	PORTB<5> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	PMA0(3)	0	O	DIG	Parallel Master Port address.
		1	I	ST/TTL	Parallel Slave Port address input.
	KBI1	1	I	TTL	Interrupt-on-change pin.
	RP8	1	I	ST	Remappable peripheral pin 8 input.
		0	O	DIG	Remappable peripheral pin 8 output.
RB6/KBI2/ PGC/RP9	RB6	0	O	DIG	LATB<6> data output.
		1	I	TTL	PORTB<6> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI2	1	I	TTL	Interrupt-on-change pin.
	PGC	x	I	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation.(2)
	RP9	1	I	ST	Remappable peripheral pin 9 input.
		0	O	DIG	Remappable peripheral pin 9 output.
RB7/KBI3/ PGD/RP10	RB7	0	O	DIG	LATB<7> data output.
		1	I	TTL	PORTB<7> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI3	1	I	TTL	Interrupt-on-change pin.
	PGD	x	O	DIG	Serial execution data output for ICSP and ICD operation.(2)
		x	I	ST	Serial execution data input for ICSP and ICD operation.(2)
	RP10	1	I	ST	Remappable peripheral pin 10 input.
		0	O	ST	Remappable peripheral pin 10 output.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: Pins are configured as analog inputs by default on POR. Using these pins for digital inputs requires setting the appropriate bits in ANCON1 first.

2: All other pin functions are disabled when ICSP™ or ICD are enabled.

3: This bit is not available on 28-pin devices.

TABLE 10-9: PORTD I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RD6/PMD6/ RP23	RD6	1	I	ST	PORTD<6> data input.
		0	O	DIG	LATD<6> data output.
	PMD6	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	RP23	1	I	ST	Remappable peripheral pin 23 input.
		0	O	DIG	Remappable peripheral pin 23 output.
RD7/PMD7/ RP24	RD7	1	I	ST	PORTD<7> data input.
		0	O	DIG	LATD<7> data output.
	PMD7	1	I	ST/TTL	Parallel Master Port data in.
		0	O	DIG	Parallel Master Port data out.
	RP24	1	I	ST	Remappable peripheral pin 24 input.
		0	O	DIG	Remappable peripheral pin 24 output.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; I^2C/SMB = I^2C/SMBus input buffer; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 10-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

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	93
LATD(1)	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	92
TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	92

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

Note 1: These registers are not available in 28-pin devices.

REGISTER 10-24: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3 (BANKED EC9h)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable if IOLOCK = 0		
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 7-5 **Unimplemented:** Read as '0'

bit 4-0 **RP3R<4:0>:** Peripheral Output Function is Assigned to RP3 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-25: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4 (BANKED ECAh)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable if IOLOCK = 0		
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 7-5 **Unimplemented:** Read as '0'

bit 4-0 **RP4R<4:0>:** Peripheral Output Function is Assigned to RP4 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-26: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5 (BANKED ECBh)

U-0	U-0	U-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0	R/$\overline{W}$-0
—	—	—	RP5R4	RP5R3	RP5R2	RP5R1	RP5R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable, Writable if IOLOCK = 0		
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 7-5 **Unimplemented:** Read as '0'

bit 4-0 **RP5R<4:0>:** Peripheral Output Function is Assigned to RP5 Output Pin bits
(see Table 10-14 for peripheral function numbers)

14.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP modules

The module is controlled through the T2CON register (Register 14-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 14-1.

14.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A 4-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options. These are selected by the prescaler control bits, T2CKPS<1:0> (T2CON<1:0>). The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 14.2 "Timer2 Interrupt"**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset (POR), $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset (WDTR) or Brown-out Reset (BOR))

TMR2 is not cleared when T2CON is written.

REGISTER 14-1: T2CON: TIMER2 CONTROL REGISTER (ACCESS FCAh)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

REGISTER 17-4: ALRMCFG: ALARM CONFIGURATION REGISTER (ACCESS F91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
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 **ALRMEN:** Alarm Enable bit

1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 0000 0000 and CHIME = 0)
0 = Alarm is disabled

bit 6 **CHIME:** Chime Enable bit

1 = Chime is enabled; ALRMRPT<7:0> bits are allowed to roll over from 00h to FFh
0 = Chime is disabled; ALRMRPT<7:0> bits stop once they reach 00h

bit 5-2 **AMASK<3:0>:** Alarm Mask Configuration bits

0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29th, once every four years)
101x = Reserved – do not use
11xx = Reserved – do not use

bit 1-0 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits

Points to the corresponding Alarm Value registers when reading the ALRMVALH and ALRMVALL registers. The ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.

ALRMVALH<15:8>:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = Unimplemented

ALRMVALL<7:0>:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = Unimplemented

REGISTER 19-3: DMACON1: DMA CONTROL REGISTER 1 (ACCESS F88h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSCON1	SSCON0	TXINC	RXINC	DUPLEX1	DUPLEX0	DLYINTEN	DMAEN
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 7-6 **SSCON<1:0>:** $\overline{\text{SSDMA}}$ Output Control bits (Master modes only)
11 = $\overline{\text{SSDMA}}$ is asserted for the duration of 4 bytes; DLYINTEN is always reset low
01 = $\overline{\text{SSDMA}}$ is asserted for the duration of 2 bytes; DLYINTEN is always reset low
10 = $\overline{\text{SSDMA}}$ is asserted for the duration of 1 byte; DLYINTEN is always reset low
00 = $\overline{\text{SSDMA}}$ is not controlled by the DMA module; DLYINTEN bit is software programmable

bit 5 **TXINC:** Transmit Address Increment Enable bit
Allows the transmit address to increment as the transfer progresses.
1 = The transmit address is to be incremented from the initial value of TXADDR<11:0>
0 = The transmit address is always set to the initial value of TXADDR<11:0>

bit 4 **RXINC:** Receive Address Increment Enable bit
Allows the receive address to increment as the transfer progresses.
1 = The received address is to be incremented from the initial value of RXADDR<11:0>
0 = The received address is always set to the initial value of RXADDR<11:0>

bit 3-2 **DUPLEX<1:0>:** Transmit/Receive Operating Mode Select bits
10 = SPI DMA operates in Full-Duplex mode, data is simultaneously transmitted and received
01 = DMA operates in Half-Duplex mode, data is transmitted only
00 = DMA operates in Half-Duplex mode, data is received only

bit 1 **DLYINTEN:** Delay Interrupt Enable bit
Enables the interrupt to be invoked after the number of SCK cycles specified in DLYCYC<2:0> has elapsed from the latest completed transfer.
1 = The interrupt is enabled, SSCON<1:0> must be set to '00'
0 = The interrupt is disabled

bit 0 **DMAEN:** DMA Operation Start/Stop bit
This bit is set by the users' software to start the DMA operation. It is reset back to zero by the DMA engine when the DMA operation is completed or aborted.
1 = DMA is in session
0 = DMA is not in session

20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs and so on.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

All members of the PIC18F46J11 family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-wake-up on character reception
  - Auto-baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half-duplex) with selectable clock polarity
- Synchronous – Slave (half-duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/PMA5/TX1/CK1/RP17 and RC7/PMA4/RX1/DT1/RP18) and remapped (RPn1/TX2/CK2 and RPn2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
  - SPEN bit (RCSTA1<7>) must be set (= 1)
  - TRISC<7> bit must be set (= 1)
  - TRISC<6> bit must be cleared (= 0) for Asynchronous and Synchronous Master modes
  - TRISC<6> bit must be set (= 1) for Synchronous Slave mode
- For EUSART2:
  - SPEN bit (RCSTA2<7>) must be set (= 1)
  - TRIS bit for RPn2/RX2/DT2 = 1
  - TRIS bit for RPn1/TX2/CK2 = 0 for Asynchronous and Synchronous Master modes
  - TRISC<6> bit must be set (= 1) for Synchronous Slave mode

Note:	The EUSART control will automatically reconfigure the pin from input to output as needed.

The TXx/CKx I/O pins have an optional open-drain output capability. By default, when this pin is used by the EUSART as an output, it will function as a standard push-pull CMOS output. The TXx/CKx I/O pins' open-drain, output feature can be enabled by setting the corresponding UxOD bit in the ODCON2 register. For more details, see **Section 19.3.3 "Open-Drain Output Option"**.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are covered in detail in Register 20-1, Register 20-2 and Register 20-3, respectively.

Note:	Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of 'x' in place of the specific module number. Thus, "RCSTAx" might refer to the Receive Status register for either EUSART1 or EUSART2.

The WUE bit is automatically cleared once a low-to-high transition is observed on the RXx line following the wake-up event. At this point, the EUSART module is in Idle mode and returns to normal operation. This signals to the user that the Sync Break event is over.

20.2.4.1 Special Considerations Using Auto-Wake-up

Since auto-wake-up functions by sensing rising edge transitions on RXx/DTx, information with any state changes before the Stop bit may signal a false End-Of-Character (EOC) and cause data or framing errors. To work properly, therefore, the initial character in the transmission must be all '0's. This can be 00h (8 bits) for standard RS-232 devices or 000h (12 bits) for LIN/J2602 bus.

Oscillator start-up time must also be considered, especially in applications using oscillators with longer start-up intervals (i.e., HS or HSPLL mode). The Sync Break (or Wake-up Signal) character must be of sufficient length and be followed by a sufficient interval to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

20.2.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RCxIF events may cause some confusion when it comes to determining the validity of received data. As noted, setting the WUE bit places the EUSART in an Idle mode. The wake-up event causes a receive interrupt by setting the RCxIF bit. The WUE bit is cleared after this when a rising edge is seen on RXx/DTx. The interrupt condition is then cleared by reading the RCREGx register. Ordinarily, the data in RCREGx will be dummy data and should be discarded.

The fact that the WUE bit has been cleared (or is still set) and the RCxIF flag is set should not be used as an indicator of the integrity of the data in RCREGx. Users should consider implementing a parallel method in firmware to verify received data integrity.

To assure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 20-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING NORMAL OPERATION

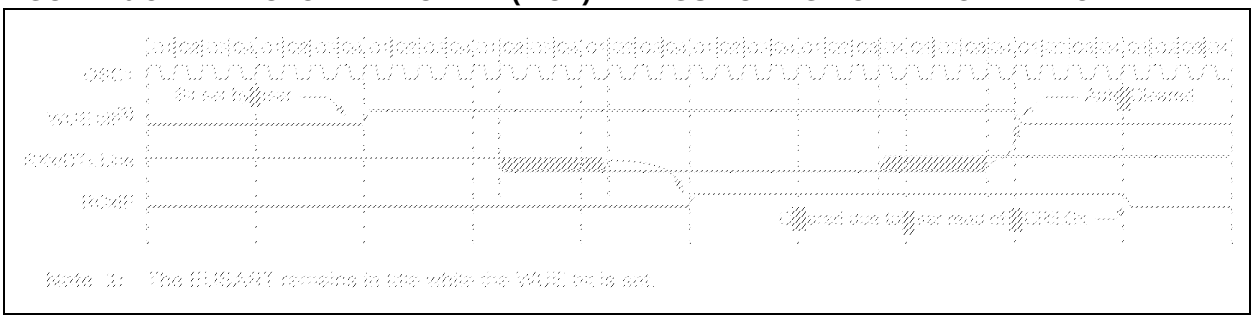


FIGURE 20-9: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP

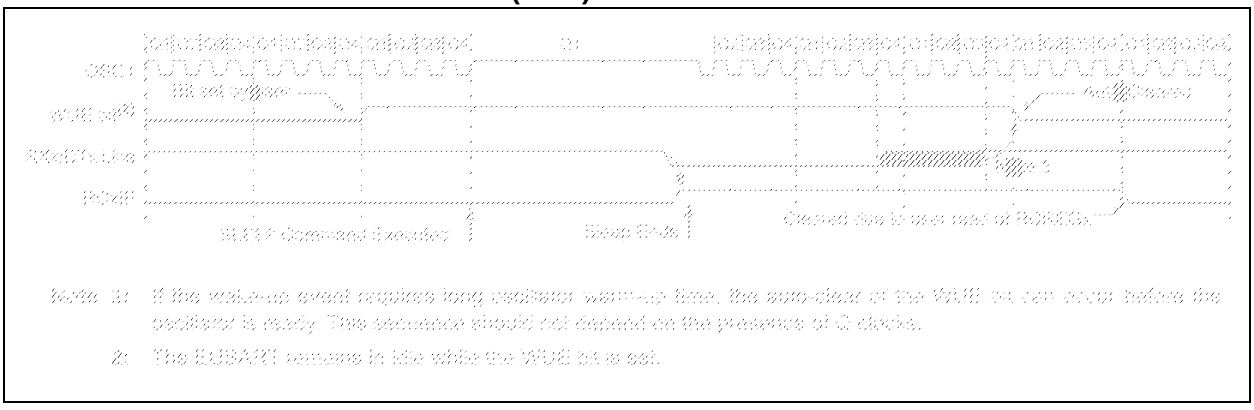


TABLE 21-2: SUMMARY OF A/D REGISTERS

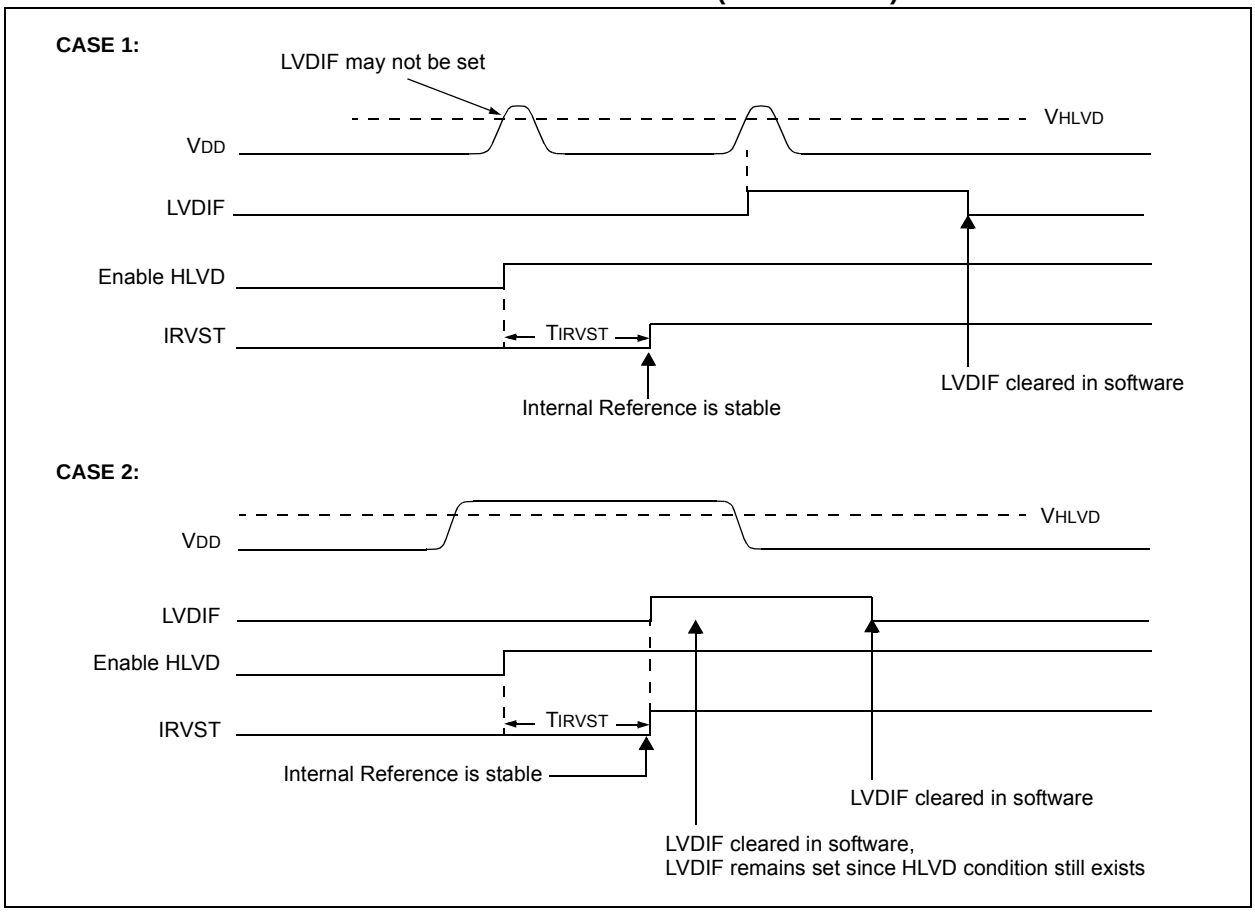
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF(1)	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE(1)	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP(1)	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	72
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	72
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	72
ADRESH	A/D Result Register High Byte								70
ADRESL	A/D Result Register Low Byte								70
ADCON0	VCFG1	VCFG0	CHS3	CHS3	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	70
ANCON0	PCFG7(1)	PCFG6(1)	PCFG5(1)	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	74
ADCON1	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	70
ANCON1	VBGEN	r(2)	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	74
CCPxCON	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	71
PORTA	RA7	RA6	RA5	—	RA3	RA2	RA1	RA0	72
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	72

Legend: — = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: These bits are only available on 44-pin devices.

2: Reserved. Always maintain as '0' for minimum power consumption.

FIGURE 24-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)



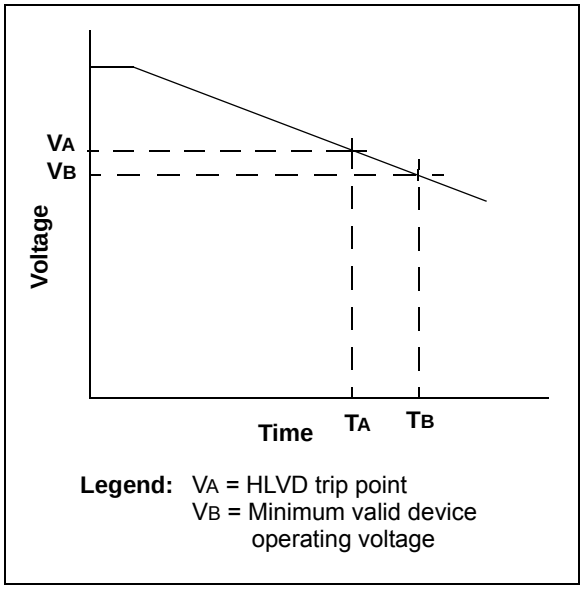
24.5 Applications

In many applications, it is desirable to have the ability to detect a drop below, or rise above, a particular threshold. For general battery applications, Figure 24-4 provides a possible voltage curve.

Over time, the device voltage decreases. When the device voltage reaches voltage, VA, the HLVD logic generates an interrupt at time, TA. The interrupt could cause the execution of an ISR, which would allow the application to perform "housekeeping tasks" and perform a controlled shutdown before the device voltage exits the valid operating range at TB.

The HLVD, thus, would give the application a time window, represented by the difference between TA and TB, to safely exit.

FIGURE 24-4: TYPICAL HIGH/ LOW-VOLTAGE DETECT APPLICATION



26.7 In-Circuit Serial Programming (ICSP)

PIC18F46J11 family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

26.8 In-Circuit Debugger

When the $\overline{\text{DEBUG}}$ Configuration bit is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® IDE. When the microcontroller has this feature enabled, some resources are not available for general use.

Table 26-4 lists the resources required by the background debugger.

TABLE 26-4: DEBUGGER RESOURCES

I/O pins:	RB6, RB7
Stack:	TOSx registers reserved

TABLE 27-2: PIC18F46J11 FAMILY INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word MSb			LSb	Status Affected	Notes
BIT-ORIENTED OPERATIONS									
BCF	f, b, a	Bit Clear f	1	1001	bbba	ffff	ffff	None	1, 2
BSF	f, b, a	Bit Set f	1	1000	bbba	ffff	ffff	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	3, 4
BTG	f, b, a	Bit Toggle f	1	0111	bbba	ffff	ffff	None	1, 2
CONTROL OPERATIONS									
BC	n	Branch if Carry	1 (2)	1110	0010	nnnn	nnnn	None	
BN	n	Branch if Negative	1 (2)	1110	0110	nnnn	nnnn	None	
BNC	n	Branch if Not Carry	1 (2)	1110	0011	nnnn	nnnn	None	
BNN	n	Branch if Not Negative	1 (2)	1110	0111	nnnn	nnnn	None	
BNOV	n	Branch if Not Overflow	1 (2)	1110	0101	nnnn	nnnn	None	
BNZ	n	Branch if Not Zero	1 (2)	1110	0001	nnnn	nnnn	None	
BOV	n	Branch if Overflow	1 (2)	1110	0100	nnnn	nnnn	None	
BRA	n	Branch Unconditionally	2	1101	0nnn	nnnn	nnnn	None	
BZ	n	Branch if Zero	1 (2)	1110	0000	nnnn	nnnn	None	
CALL	n, s	Call Subroutine 1st word	2	1110	110s	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0000	0100	$\overline{TO}$, $\overline{PD}$	
DAW	—	Decimal Adjust WREG	1	0000	0000	0000	0111	C	
GOTO	n	Go to Address 1st word	2	1110	1111	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
NOP	—	No Operation	1	0000	0000	0000	0000	None	
NOP	—	No Operation	1	1111	xxxx	xxxx	xxxx	None	4
POP	—	Pop Top of Return Stack (TOS)	1	0000	0000	0000	0110	None	
PUSH	—	Push Top of Return Stack (TOS)	1	0000	0000	0000	0101	None	
RCALL	n	Relative Call	2	1101	1nnn	nnnn	nnnn	None	
RESET		Software Device Reset	1	0000	0000	1111	1111	All	
RETFIE	s	Return from Interrupt Enable	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	Return with Literal in WREG	2	0000	1100	kkkk	kkkk	None	
RETURN	s	Return from Subroutine	2	0000	0000	0001	001s	None	
SLEEP	—	Go into Standby mode	1	0000	0000	0000	0011	$\overline{TO}$, $\overline{PD}$	

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

4: Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

TABLE 29-5: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	VRGOUT	Regulator Output Voltage	2.35	2.5	2.7	V	Regulator enabled, VDD = 3.0V
	CEFC	External Filter Capacitor Value(1)	5.4	10	18	µF	ESR < 3Ω recommended ESR < 5Ω required

Note 1: CEFC applies for PIC18F devices in the family. For PIC18LF devices in the family, there is no specific minimum or maximum capacitance for VDDCORE, although proper supply rail bypassing should still be used.

TABLE 29-6: ULPWU SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D100	IULP	**Ultra Low-Power Wake-up Current**	—	60	—	nA	Net of I/O leakage and current sink at 1.6V on pin, VDD = 3.3V See Application Note *AN879,* "*Using the Microchip Ultra Low-Power Wake-up Module*" (DS00879)

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

TABLE 29-27: MSSPx I²C™ BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
101	TLOW	Clock Low Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	Only relevant for Repeated Start condition
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	ms	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	**(Note 1)**
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ms	
			400 kHz mode	2(TOSC)(BRG + 1)	—	ms	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	ms	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	ms	
D102	CB	Bus Capacitive Loading		—	400	pF	

Note 1: A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but parameter #107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, parameter #102 + parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCLx line is released.