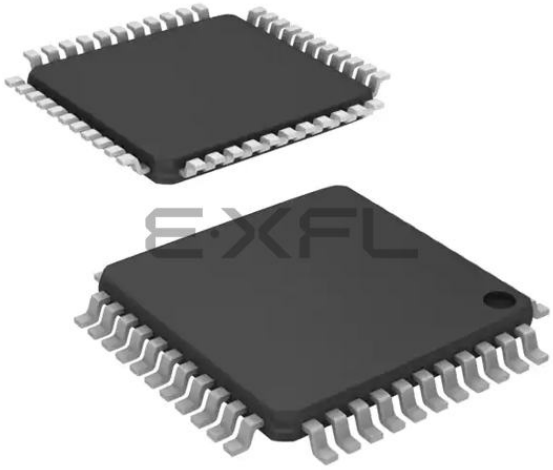


Welcome to E-XFL.COM

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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.15V ~ 3.6V
Data Converters	A/D 13x10b/12b
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/pic18f47j13t-i-pt

3.4 Reference Clock Output

In addition to the peripheral clock/4 output in certain oscillator modes, the device clock in the PIC18F47J13 Family can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 3-4). Setting the ROON bit (REFOCON<7>) makes the clock signal available on the REFO (RB2) pin. The RODIV<3:0> bits enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<5:4>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator is on OSC1 and OSC2, or the current system clock source is used for the reference clock output. The ROSSLP bit determines if the reference source is available on RB2 when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for an EC or HS mode; otherwise, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

REGISTER 3-4: REFOCON: REFERENCE OSCILLATOR CONTROL REGISTER (BANKED F3Dh)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROON	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
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 **ROON:** Reference Oscillator Output Enable bit
1 = Reference oscillator is enabled on REFO pin
0 = Reference oscillator is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **ROSSLP:** Reference Oscillator Output Stop in Sleep bit
1 = Reference oscillator continues to run in Sleep
0 = Reference oscillator is disabled in Sleep

bit 4 **ROSEL:** Reference Oscillator Source Select bit
1 = Primary oscillator crystal/resonator is used as the base clock(1)
0 = System clock (FOSC) is used as the base clock; the base clock reflects any clock switching of the device

bit 3-0 **RODIV<3:0>:** Reference Oscillator Divisor Select bits
1111 = Base clock value divided by 32,768
1110 = Base clock value divided by 16,384
1101 = Base clock value divided by 8,192
1100 = Base clock value divided by 4,096
1011 = Base clock value divided by 2,048
1010 = Base clock value divided by 1,024
1001 = Base clock value divided by 512
1000 = Base clock value divided by 256
0111 = Base clock value divided by 128
0110 = Base clock value divided by 64
0101 = Base clock value divided by 32
0100 = Base clock value divided by 16
0011 = Base clock value divided by 8
0010 = Base clock value divided by 4
0001 = Base clock value divided by 2
0000 = Base clock value

Note 1: The crystal oscillator must be enabled using the FOSC<2:0> bits; the crystal maintains the operation in Sleep mode.

TABLE 4-1: LOW-POWER MODES

Mode	DSCONH<7>	OSCCON<7,1:0>		Module Clocking		Available Clock and Oscillator Source
	DSEN(1)	IDLEN(1)	SCS<1:0>	CPU	Peripherals	
Sleep	0	0	N/A	Off	Off	Timer1 oscillator and/or RTCC may optionally be enabled
Deep Sleep(3)	1	0	N/A	Powered off(2)	Powered off	RTCC can run uninterrupted using the Timer1 or internal low-power RC oscillator
PRI_RUN	0	N/A	00	Clocked	Clocked	The normal, full-power execution mode; primary clock source (defined by FOSC<2:0>)
SEC_RUN	0	N/A	01	Clocked	Clocked	Secondary – Timer1 oscillator
RC_RUN	0	N/A	11	Clocked	Clocked	Postscaled internal clock
PRI_IDLE	0	1	00	Off	Clocked	Primary clock source (defined by FOSC<2:0>)
SEC_IDLE	0	1	01	Off	Clocked	Secondary – Timer1 oscillator
RC_IDLE	0	1	11	Off	Clocked	Postscaled internal clock

Note 1: IDLEN and DSEN reflect their values when the SLEEP instruction is executed.

2: Deep Sleep turns off the internal core voltage regulator to power down core logic. See **Section 4.6 "Deep Sleep Mode"** for more information.

3: Deep Sleep mode is only available on "F" devices, not "LF" devices.

4.1.3 CLOCK TRANSITIONS AND STATUS INDICATORS

The length of the transition between clock sources is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Two bits indicate the current clock source and its status: OSTS (OSCCON<3>) and SOSCRUN (OSCCON2<6>). In general, only one of these bits will be set in a given power-managed mode. When the OSTS bit is set, the primary clock would be providing the device clock. When the SOSCRUN bit is set, the Timer1 oscillator would be providing the clock. If neither of these bits is set, INTRC would be clocking the device.

Note:	Executing a SLEEP instruction does not necessarily place the device into Sleep mode. It acts as the trigger to place the controller into either the Sleep or Deep Sleep mode, or one of the Idle modes, depending on the setting of the IDLEN bit.

4.1.4 MULTIPLE SLEEP COMMANDS

The power-managed mode that is invoked with the SLEEP instruction is determined by the setting of the IDLEN and DSEN bits at the time the instruction is executed. If another SLEEP instruction is executed, the device will enter the power-managed mode specified by IDLEN and DSEN at that time. If IDLEN or DSEN have changed, the device will enter the new power-managed mode specified by the new setting.

4.2.3 RC_RUN MODE

In RC_RUN mode, the CPU and peripherals are clocked from the internal oscillator; the primary clock is shut down. This mode provides the best power conservation of all the Run modes while still executing code. It works well for user applications, which are not highly timing-sensitive or do not require high-speed clocks at all times.

This mode is entered by setting the SCS<1:0> bits (OSCCON<1:0>) to '11'. When the clock source is switched to the internal oscillator block (see Figure 4-3), the primary oscillator is shut down and the OSTS bit is cleared.

On transitions from RC_RUN mode to PRI_RUN mode, the device continues to be clocked from the INTOSC block while the primary clock is started. When the primary clock becomes ready, a clock switch to the primary clock occurs (see Figure 4-4). When the clock switch is complete, the OSTS bit is set and the primary clock is providing the device clock. The IDLEN and SCS bits are not affected by the switch. The INTRC clock source will continue to run if either the WDT or the FSCM is enabled.

FIGURE 4-3: TRANSITION TIMING TO RC_RUN MODE

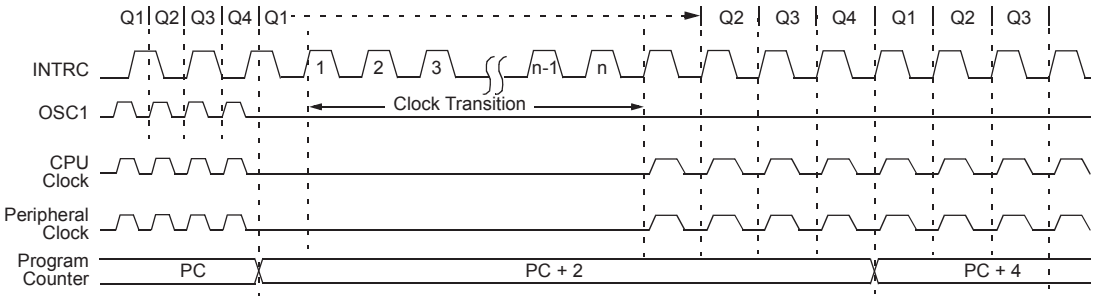
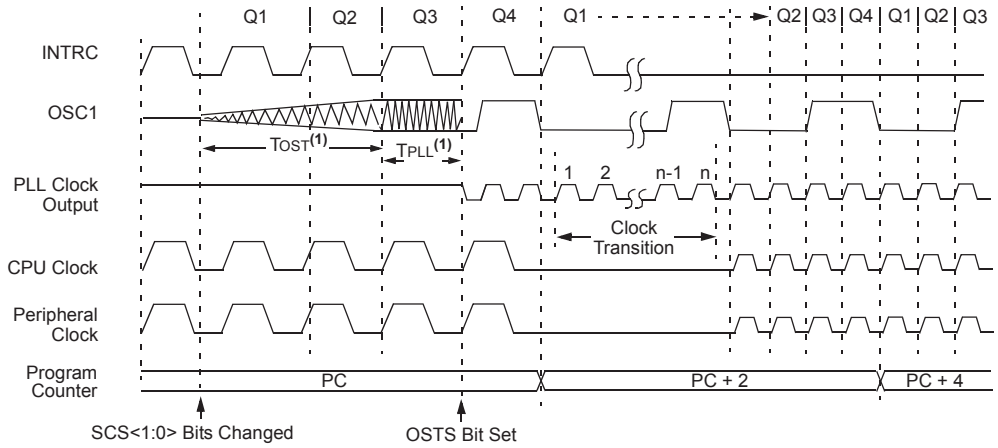


FIGURE 4-4: TRANSITION TIMING FROM RC_RUN MODE TO PRI_RUN MODE



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

FIGURE 4-7: TRANSITION TIMING FOR ENTRY TO IDLE MODE

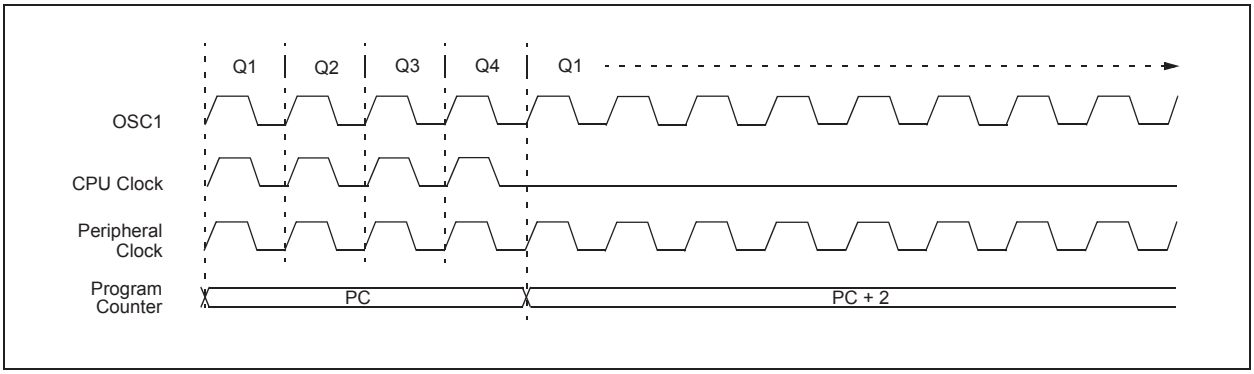
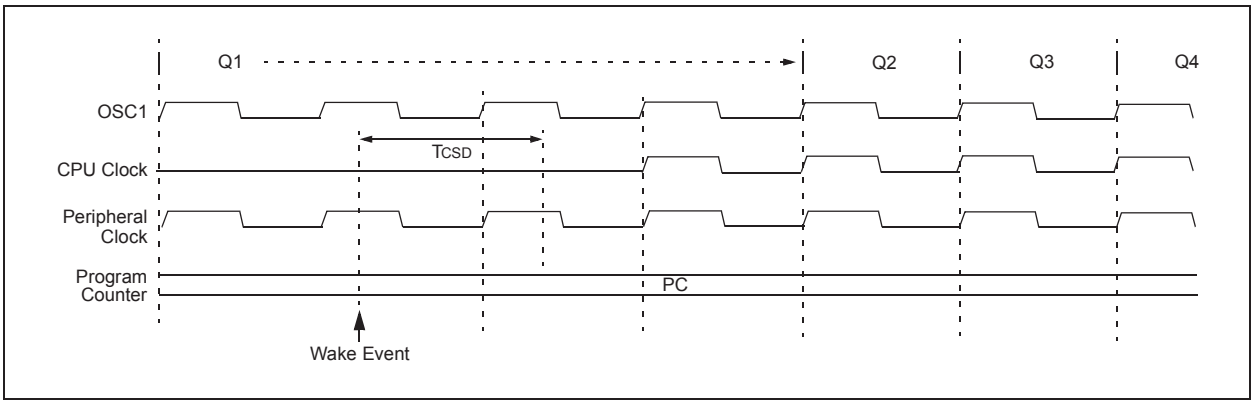


FIGURE 4-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE



4.6.2 I/O PINS DURING DEEP SLEEP

During Deep Sleep, the general purpose I/O pins will retain their previous states.

Pins that are configured as inputs (TRISx bit set) prior to entry into Deep Sleep will remain high-impedance during Deep Sleep.

Pins that are configured as outputs (TRISx bit clear) prior to entry into Deep Sleep will remain as output pins during Deep Sleep. While in this mode, they will drive the output level determined by their corresponding LAT bit at the time of entry into Deep Sleep.

When the device wakes back up, the I/O pin behavior depends on the type of wake up source.

If the device wakes back up by an RTCC alarm, INT0 interrupt, DSWDT or ULPWU event, all I/O pins will continue to maintain their previous states, even after the device has finished the POR sequence and is executing application code again. Pins configured as inputs during Deep Sleep will remain high-impedance, and pins configured as outputs will continue to drive their previous value.

After waking up, the TRIS and LAT registers will be reset, but the I/O pins will still maintain their previous states. If firmware modifies the TRIS and LAT values for the I/O pins, they will not immediately go to the newly configured states. Once the firmware clears the RELEASE bit (DSCONL<0>), the I/O pins will be "released". This causes the I/O pins to take the states configured by their respective TRIS and LAT bit values.

If the Deep Sleep BOR (DSBOR) circuit is enabled, and V_{DD} drops below the DSBOR and V_{DD} rail POR thresholds, the I/O pins will be immediately released similar to clearing the RELEASE bit. All previous state information will be lost, including the general purpose DSGPR0 and DSGPR1 contents. See **Section 4.6.5 "Deep Sleep Brown-out Reset (DSBOR)"** for additional details regarding this scenario.

If a $\overline{MCLR}$ Reset event occurs during Deep Sleep, the I/O pins will also be released automatically, but in this case, the DSGPR0 and DSGPR1 contents will remain valid.

In all other Deep Sleep wake-up cases, application firmware needs to clear the RELEASE bit in order to reconfigure the I/O pins.

4.6.3 DEEP SLEEP WAKE-UP SOURCES

The device can be awakened from Deep Sleep mode by a $\overline{MCLR}$, POR, RTCC, INT0 I/O pin interrupt, DSWDT or ULPWU event. After waking, the device performs a POR. When the device is released from Reset, code execution will begin at the device's Reset vector.

The software can determine if the wake-up was caused from an exit from Deep Sleep mode by reading the DS bit (WDTCON<3>). If this bit is set, the POR was caused by a Deep Sleep exit. The DS bit must be manually cleared by the software.

The software can determine the wake event source by reading the DSWAKEH and DSWAKEL registers. When the application firmware is done using the DSWAKEH and DSWAKEL status registers, individual bits do not need to be manually cleared before entering Deep Sleep again. When entering Deep Sleep mode, these registers are automatically cleared.

4.6.3.1 Wake-up Event Considerations

Deep Sleep wake-up events are only monitored while the processor is fully in Deep Sleep mode. If a wake-up event occurs before Deep Sleep mode is entered, the event status will not be reflected in the DSWAKE registers. If the wake-up source asserts prior to entering Deep Sleep, the CPU will either go to the interrupt vector (if the wake source has an interrupt bit and the interrupt is fully enabled) or will abort the Deep Sleep entry sequence by executing past the `SLEEP` instruction, if the interrupt was not enabled. In this case, a wake-up event handler should be placed after the `SLEEP` instruction to process the event and re-attempt entry into Deep Sleep if desired.

When the device is in Deep Sleep with more than one wake-up source simultaneously enabled, only the first wake-up source to assert will be detected and logged in the DSWAKEH/DSWAKEL Status registers.

4.6.4 DEEP SLEEP WATCHDOG TIMER (DSWDT)

Deep Sleep has its own dedicated WDT (DSWDT) with a postscaler for time-outs of 2.1 ms to 25.7 days, configurable through the bits, DSWDTPS<3:0>.

The DSWDT can be clocked from either the INTRC or the T1OSC/T1CKI input. If the T1OSC/T1CKI source will be used with a crystal, the T1OSCEN bit in the T1CON register needs to be set prior to entering Deep Sleep. The reference clock source is configured through the DSWDTOSC bit.

DSWDT is enabled through the DSWDTEN bit. Entering Deep Sleep mode automatically clears the DSWDT. See **Section 27.0 "Special Features of the CPU"** for more information.

6.2.3 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as 2 bytes or 4 bytes in program memory. The Least Significant Byte (LSB) of an instruction word is always stored in a program memory location with an even address (LSB = 0). To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSB will always read '0' (see **Section 6.1.3 "Program Counter"**).

Figure 6-5 provides an example of how instruction words are stored in the program memory.

The CALL and GOTO instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 6-5 displays how the instruction, GOTO 0006h, is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 28.0 "Instruction Set Summary"** provides further details of the instruction set.

FIGURE 6-5: INSTRUCTIONS IN PROGRAM MEMORY

Program Memory Byte Locations

Instruction 1: MOVLW 055h
Instruction 2: GOTO 0006h
Instruction 3: MOVFF 123h, 456h

LSB = 1	LSB = 0	Word Address
		000000h
		000002h
		000004h
		000006h
0Fh	55h	000008h
EFh	03h	00000Ah
F0h	00h	00000Ch
C1h	23h	00000Eh
F4h	56h	000010h
		000012h
		000014h

6.2.4 TWO-WORD INSTRUCTIONS

The standard PIC18 instruction set has four, two-word instructions: CALL, MOVFF, GOTO and LSFR. In all cases, the second word of the instructions always has '1111' as its four Most Significant bits (MSbs); the other 12 bits are literal data, usually a data memory address.

The use of '1111' in the 4 MSbs of an instruction specifies a special form of NOP. If the instruction is executed in proper sequence immediately after the first word, the data in the second word is accessed and used by the instruction sequence. If the first word is skipped for some reason, and the second word is executed by itself, a NOP is executed instead. This is necessary for cases when the two-word instruction is preceded by a conditional instruction that changes the PC. Example 6-4 illustrates how this works.

Note: See **Section 6.5 "Program Memory and the Extended Instruction Set"** for information on two-word instructions in the extended instruction set.

EXAMPLE 6-4: TWO-WORD INSTRUCTIONS

CASE 1:

Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; No, skip this word
1111 0100 0101 0110			; Execute this word as a NOP
0010 0100 0000 0000	ADDWF	REG3	; continue code

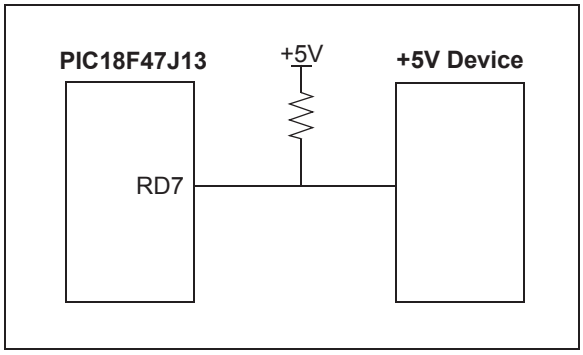
CASE 2:

Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; Yes, execute this word
1111 0100 0101 0110			; 2nd word of instruction
0010 0100 0000 0000	ADDWF	REG3	; continue code

10.1.3 INTERFACING TO A 5V SYSTEM

Though the V_{DDMAX} of the PIC18F47J13 Family is 3.6V, these devices are still capable of interfacing with 5V systems, even if the V_{IH} of the target system is above 3.6V. This is accomplished by adding a pull-up resistor to the port pin (Figure 10-2), clearing the LAT bit for that pin and manipulating the corresponding TRISx bit (Figure 10-1) to either allow the line to be pulled high or to drive the pin low. Only port pins that are tolerant of voltages up to 5.5V can be used for this type of interface (refer to **Section 10.1.2 "Input Pins and Voltage Considerations"**).

FIGURE 10-2: +5V SYSTEM HARDWARE INTERFACE



EXAMPLE 10-1: COMMUNICATING WITH THE +5V SYSTEM

```
BCF   LATD, 7   ; set up LAT register so
                ; changing TRIS bit will
                ; drive line low
BCF   TRISD, 7  ; send a 0 to the 5V system
BSF   TRISD, 7  ; send a 1 to the 5V system
```

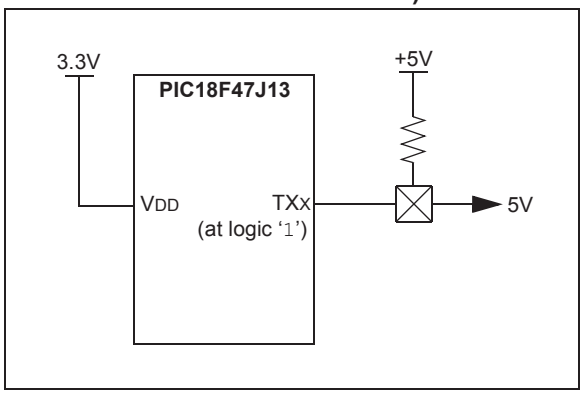
10.1.4 OPEN-DRAIN OUTPUTS

The output pins for several peripherals are also equipped with a configurable, open-drain output option. This allows the peripherals to communicate with external digital logic operating at a higher voltage level, without the use of level translators.

The open-drain option is implemented on port pins specifically associated with the data and clock outputs of the EUSARTs, the MSSP modules (in SPI mode) and the ECCP modules. It is selectively enabled by setting the open-drain control bit for the corresponding module in the ODCON registers (Register 10-1, Register 10-2 and Register 10-3). Their configuration is discussed in more detail with the individual port where these peripherals are multiplexed. Output functions that are routed through the PPS module may also use the open-drain option. The open-drain functionality will follow the I/O pin assignment in the PPS module.

When the open-drain option is required, the output pin must also be tied through an external pull-up resistor provided by the user to a higher voltage level, up to 5.5V (Figure 10-3). When a digital logic high signal is output, it is pulled up to the higher voltage level.

FIGURE 10-3: USING THE OPEN-DRAIN OUTPUT (USART SHOWN AS EXAMPLE)



10.1.5 TTL INPUT BUFFER OPTION

Many of the digital I/O ports use Schmitt Trigger (ST) input buffers. While this form of buffering works well with many types of input, some applications may require TTL level signals to interface with external logic devices. This is particularly true for the Parallel Master Port (PMP), which is likely to be interfaced to TTL level logic or memory devices.

The inputs for the PMP can be optionally configured for TTL buffers with the PMPTTL bit in the PADCFG1 register (Register 10-4). Setting this bit configures all data and control input pins for the PMP to use TTL buffers. By default, these PMP inputs use the port's ST buffers.

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

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RB7/CCP7/ KBI3/PGD/ RP10	RB7	0	O	DIG	LATB<7> data output.
		1	I	TTL	PORTB<7> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared.
	CCP7	1	I	ST	Capture input.
		0	O	DIG	Compare/PWM output.
	KBI3	1	O	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	DIG	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 (TRISx 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 the ANCON1 register.

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

3: Only on 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13).

4: Only on 28-pin devices (PIC18F26J13, PIC18F27J13, PIC18LF26J13 and PIC18LF27J13).

TABLE 10-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
ANCON1	VBGEN	—	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
REFOCON	ROON	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
CM3CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
PADCFG1	—	—	—	—	—	RTSECSEL1	RTSECSEL0	PMPTTL
RTCCFG	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0

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

TABLE 10-7: PORTC I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RC5/SDO1/ RP16	RC5	0	O	DIG	PORTC<5> data output.
	SDO1	x	O	DIG	SPI data output (MSSP1 module).
	RP16	1	I	ST	Remappable Peripheral Pin 16 input.
		0	O	DIG	Remappable Peripheral Pin 16 output.
RC6/CCP9/ PMA5/TX1/ CK1/RP17	RC6	1	I	ST	PORTC<6> data input.
		0	O	DIG	LATC<6> data output.
	CCP9	1	I	ST	Capture input.
		0	O	DIG	Compare/PWM output.
	PMA5(1)	1	I	ST/TTL	Parallel Master Port io_addr_in<5>.
		0	O	DIG	Parallel Master Port address.
	TX1	0	O	DIG	Asynchronous serial transmit data output (EUSART module); takes priority over port data. User must configure as an output.
	CK1	1	I	ST	Synchronous serial clock input (EUSART module).
		0	O	DIG	Synchronous serial clock output (EUSART module); takes priority over port data.
	RP17	1	I	ST	Remappable Peripheral Pin 17 input.
		0	O	DIG	Remappable Peripheral Pin 17 output.
RC7/CCP10/ PMA4/RX1/ DT1/RP18	RC7	1	I	ST	PORTC<7> data input.
		0	O	DIG	LATC<7> data output.
	CCP10	1	I	ST	Capture input.
		0	O	DIG	Compare/PWM output.
	PMA4(1)	x	I/O	ST/TTL/ DIG	Parallel Master Port address.
	RX1	1	I	ST	Asynchronous serial receive data input (EUSART module).
	DT1	1	1	ST	Synchronous serial data input (EUSART module). User must configure as an input.
		0	O	DIG	Synchronous serial data output (EUSART module); takes priority over port data.
	RP18	1	I	ST	Remappable Peripheral Pin 18 input.
		0	O	DIG	Remappable Peripheral Pin 18 output.

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

Note 1: This bit is only available on 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13).

TABLE 10-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
ANCON1	VBGEN	—	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
RTCCFG	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0

When a shutdown event occurs, two things happen:

- The ECCPxASE bit is set to '1'. The ECCPxASE will remain set until cleared in firmware or an auto-restart occurs. (See **Section 19.4.5 "Auto-Restart Mode"**.)
- The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs (PxA/PxC and PxB/PxD). The state of each pin pair is determined by the PSSxAC and PSSxBD bits (ECCPxAS<3:0>).

Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

REGISTER 19-4: ECCPxAS: ECCP1/2/3 AUTO-SHUTDOWN CONTROL REGISTER (1, ACCESS FBEh; 2, FB8h; 3, BANKED F19h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0
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 **ECCPxASE:** ECCP Auto-Shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in a shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCPxAS<2:0>:** ECCP Auto-Shutdown Source Select bits
000 = Auto-shutdown is disabled
001 = Comparator, C1OUT, output is high
010 = Comparator, C2OUT, output is high
011 = Either comparator, C1OUT or C2OUT, is high
100 = VIL on FLT0 pin
101 = VIL on FLT0 pin or comparator, C1OUT, output is high
110 = VIL on FLT0 pin or comparator, C2OUT, output is high
111 = VIL on FLT0 pin or comparator, C1OUT, or comparator, C2OUT, is high

bit 3-2 **PSSxAC<1:0>:** PxA and PxC Pins Shutdown State Control bits
00 = Drive pins, PxA and PxC, to '0'
01 = Drive pins, PxA and PxC, to '1'
1x = PxA and PxC pins tri-state

bit 1-0 **PSSxBD<1:0>:** PxB and PxD Pins Shutdown State Control bits
00 = Drive pins, PxB and PxD, to '0'
01 = Drive pins, PxB and PxD, to '1'
1x = PxB and PxD pins tri-state

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.
2: Writing to the ECCPxASE bit is disabled while an auto-shutdown condition persists.
3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

REGISTER 20-2: SSPxCON1: MSSPx CONTROL REGISTER 1 (SPI MODE) (1, ACCESS FC6h; 2, F72h)

R/W-0	R/C-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
bit 7							bit 0

Legend:	C = Clearable bit		
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 **WCOL:** Write Collision Detect bit
- 1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
- 0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit(1)

SPI Slave mode:
- 1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. The user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
- 0 = No overflow

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(2)
- 1 = Enables serial port and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
- 1 = Idle state for clock is a high level
- 0 = Idle state for clock is a low level

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(3)
- 0101 = SPI Slave mode, clock = SCKx pin; $\overline{SSx}$ pin control disabled, $\overline{SSx}$ can be used as I/O pin
- 0100 = SPI Slave mode, clock = SCKx pin; $\overline{SSx}$ pin control enabled
- 0011 = SPI Master mode, clock = TMR2 output/2
- 0010 = SPI Master mode, clock = FOSC/64
- 0001 = SPI Master mode, clock = FOSC/16
- 1010 = SPI Master mode, clock = FOSC/8
- 0000 = SPI Master mode, clock = FOSC/4

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.

2: When enabled, this pin must be properly configured as input or output.

3: Bit combinations not specifically listed here are either reserved or implemented in I^2C mode only.

20.5.17.2 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 a low level to a high level.

b) SCLx goes low before SDAx is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user deasserts SDAx and the pin is allowed to float high, the BRG is loaded with SSPxADD<6:0> and counts down to 0. 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'; see Figure 20-31). 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 (see Figure 20-32).

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 20-31: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

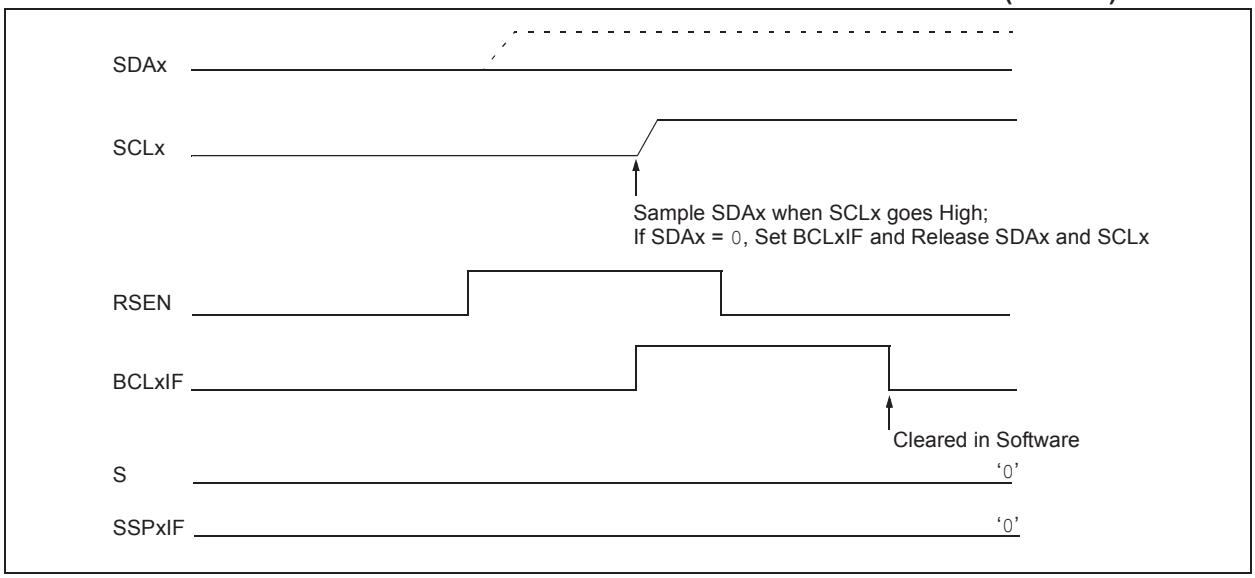
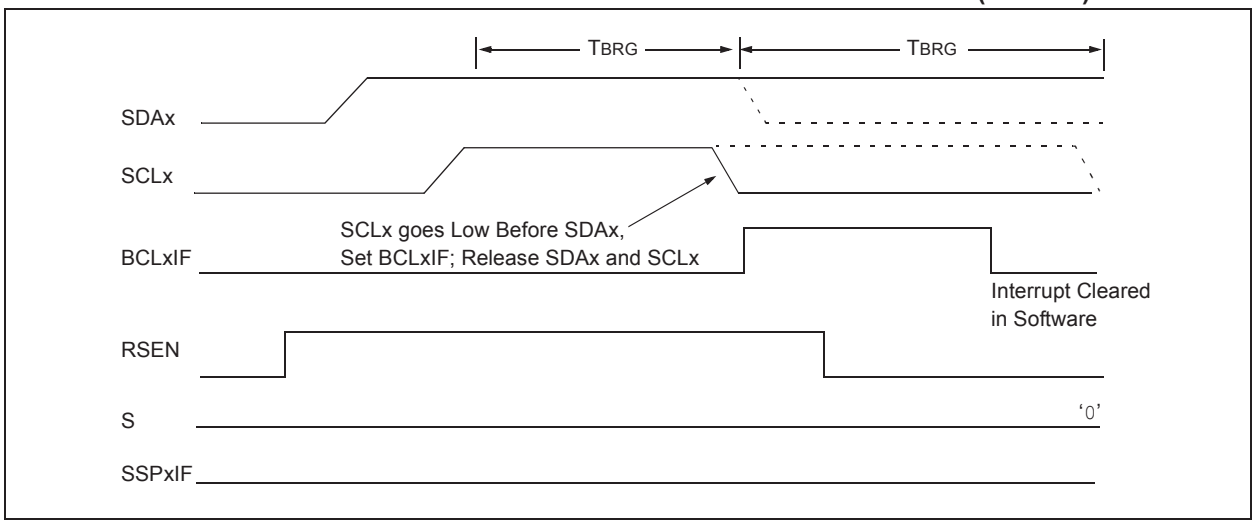


FIGURE 20-32: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



REGISTER 21-1: TXSTAx: TRANSMIT STATUS AND CONTROL REGISTER (1, ACCESS FADh; 2, FA8h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit

Asynchronous mode:
Don't care.

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-Bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)

1 = Transmit is enabled and the TXx/CKx pin is configured as an output
0 = Transmit is disabled

bit 4 **SYNC:** EUSART Mode Select bit

1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit

Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed

Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit

1 = TSR is empty
0 = TSR is full

bit 0 **TX9D:** 9th bit of Transmit Data

Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

21.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN/J2602 bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTAx<3> and TXSTAx<5>) are set while the Transmit Shift Register is loaded with data.

Note that the value of data written to TXREGx will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN/J2602 specification).

Note that the data value written to the TXREGx for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 21-10 for the timing of the Break character sequence.

21.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN/J2602 bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREGx to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREGx becomes empty, as indicated by the TXxIF, the next data byte can be written to TXREGx.

21.2.6 RECEIVING A BREAK CHARACTER

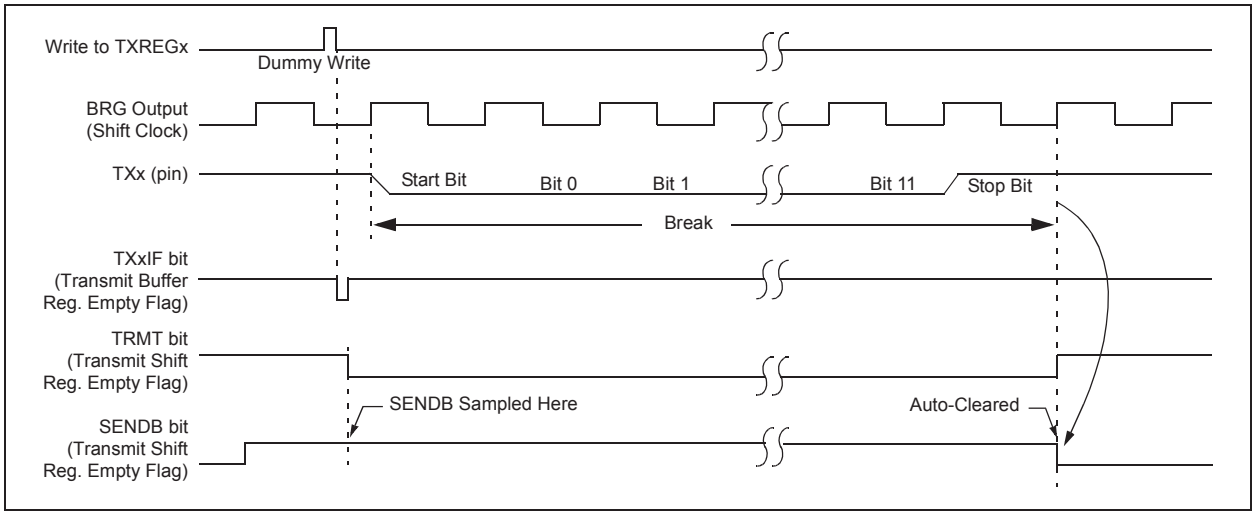
The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 of the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 21.2.4 "Auto-Wake-up on Sync Break Character"**. By enabling this feature, the EUSART will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TXxIF interrupt is observed.

FIGURE 21-10: SEND BREAK CHARACTER SEQUENCE



EXAMPLE 26-3: CAPACITANCE CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 25                            //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5                     //time in uS
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023                        //for unsigned conversion 10 sig bits
#define ADREF 3.3                           //Vdd connected to A/D Vr+
#define RCAL .027                           //R value is 4200000 (4.2M)
                                            //scaled so that result is in
                                            //1/100th of uA

int main(void)
{
    int i;
    int j = 0;                               //index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

                                             //assume CTMU and A/D have been setup correctly
                                             //see Example 25-1 for CTMU & A/D setup
    setup();

        CTMUCONHbits.CTMUEN = 1;             // Enable the CTMU
        CTMUCONLbits.EDG1STAT = 0;           // Set Edge status bits to zero
        CTMUCONLbits.EDG2STAT = 0;
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1;           //drain charge on the circuit
        DELAY;                              //wait 125us
        CTMUCONHbits.IDISSEN = 0;           //end drain of circuit

        CTMUCONLbits.EDG1STAT = 1;          //Begin charging the circuit
                                            //using CTMU current source
        DELAY;                              //wait for 125us
        CTMUCONLbits.EDG1STAT = 0;          //Stop charging circuit

        PIR1bits.ADIF = 0;                  //make sure A/D Int not set
        ADCON0bits.GO=1;                    //and begin A/D conv.
        while(!PIR1bits.ADIF);              //Wait for A/D convert complete


        Vread = ADRES;                      //Get the value from the A/D
        PIR1bits.ADIF = 0;                  //Clear A/D Interrupt Flag
        VTot += Vread;                      //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);            //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;                   //CTMUISrc is in 1/100ths of uA
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
}
```

26.9 Registers

There are three control registers for the CTMU:

- CTMUCONH
- CTMUCONL
- CTMUICON

The CTMUCONH and CTMUCONL registers (Register 26-1 and Register 26-2) contain control bits for configuring the CTMU module edge source selection, edge source polarity selection, edge sequencing, A/D trigger, analog circuit capacitor discharge and enables. The CTMUICON register (Register 26-3) has bits for selecting the current source range and current source trim.

REGISTER 26-1: CTMUCONH: CTMU CONTROL REGISTER HIGH (ACCESS FB3h)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 4 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 3 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 2 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 1 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 0 **CTTRIG:** CTMU Special Event Trigger bit
1 = CTMU Special Event Trigger is enabled
0 = CTMU Special Event Trigger is disabled

XORWF	Exclusive OR W with f
Syntax:	XORWF f {,d {,a}}
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(W) .XOR. (f) → dest
Status Affected:	N, Z

Encoding:

0001	10da	ffff	ffff

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 28.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `XORWF REG, 1, 0`

Before Instruction
REG = AFh
W = B5h
After Instruction
REG = 1Ah
W = B5h

FIGURE 30-1: PIC18F47J13 FAMILY VDD FREQUENCY GRAPH (INDUSTRIAL)

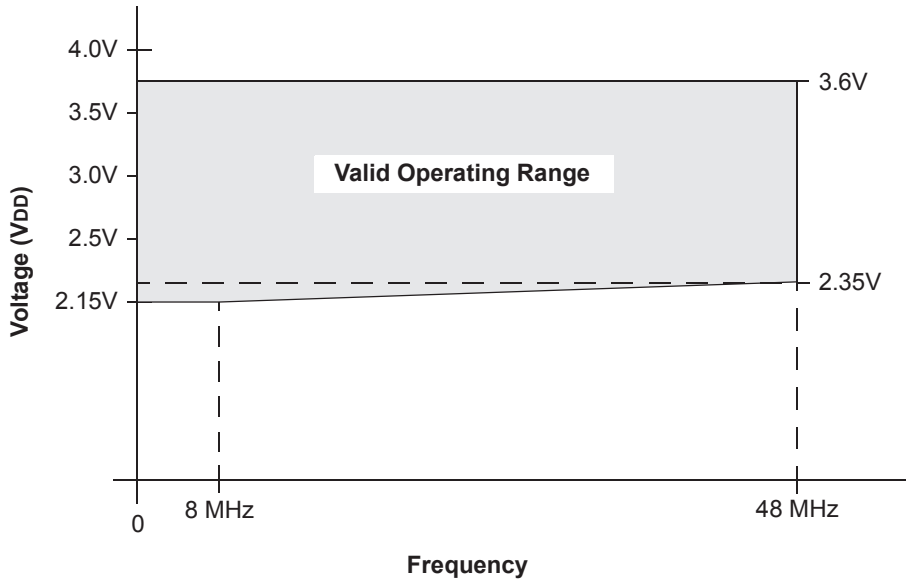
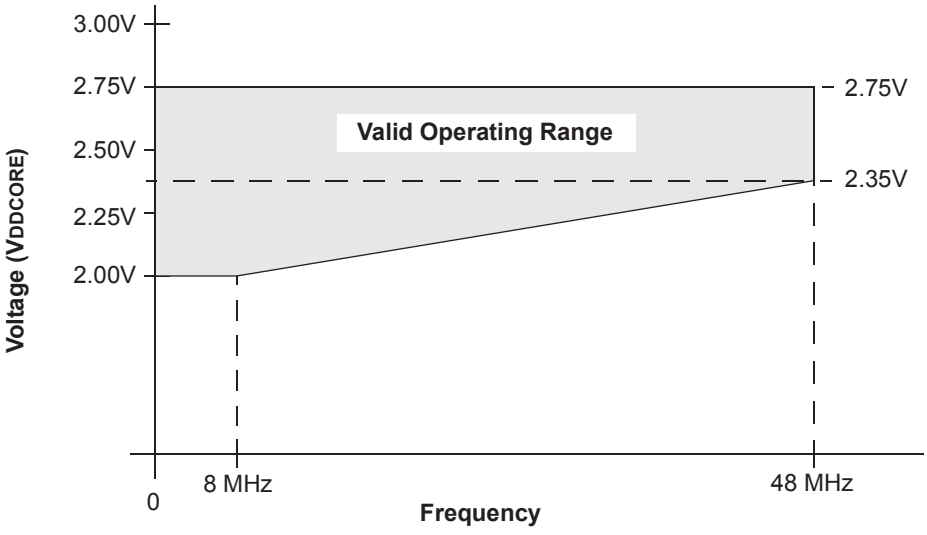


FIGURE 30-2: PIC18LF47J13 FAMILY VDDCORE FREQUENCY GRAPH (INDUSTRIAL)(1)



Note 1: VDD and VDDCORE must be maintained so that VDDCORE ≤ VDD.

30.4 DC Characteristics: PIC18F47J13 Family (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial			
Param No.	**Symbol**	**Characteristic**	**Typ**	**Max**	**Units**	**Conditions**
	IIL	**Input Leakage Current(1,2)**				
D060		I/O Ports without 5.5V Tolerance	±5	±200	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
		I/O Ports with 5.5V Tolerance	±5	±200	nA	VSS ≤ VPIN ≤ 5.5V, Pin at high-impedance
D061		$\overline{\text{MCLR}}$	±5	±200	nA	Vss ≤ VPIN ≤ VDD
D063		OSC1	±5	±200	nA	Vss ≤ VPIN ≤ VDD

Note 1: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

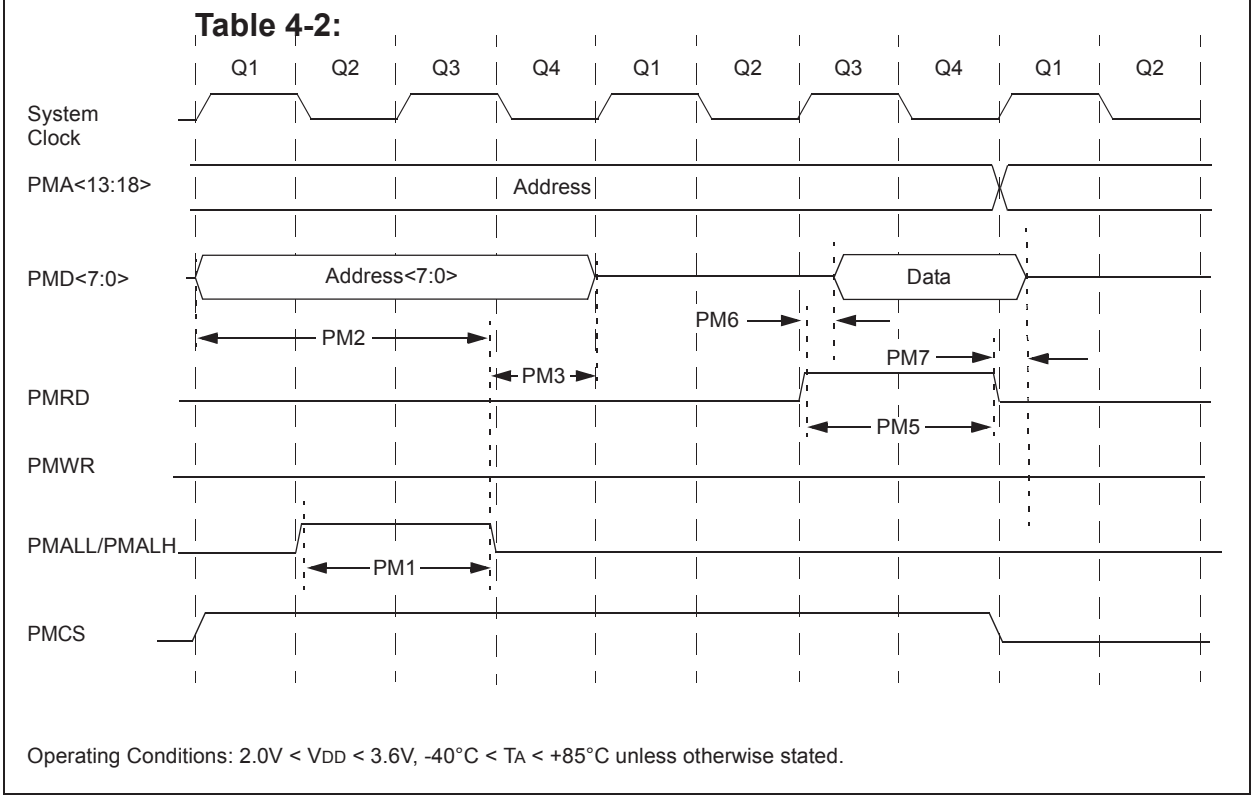
2: Negative current is defined as current sourced by the pin.

TABLE 30-1: MEMORY PROGRAMMING REQUIREMENTS

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**
		Program Flash Memory					
D130	EP	Cell Endurance	10K	—	—	E/W	-40°C to +85°C
D131	VPR	VDDcore for Read	VMIN	—	2.75	V	VMIN = Minimum operating voltage
D132B	VPEW	VDDCORE for Self-Timed Erase or Write	2.25	—	2.75	V	
D133A	TIW	Self-Timed Write Cycle Time	—	2.8	—	ms	64 bytes
D133B	TIE	Self-Timed Block Erase Cycle Time	—	33.0	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	3	—	mA	

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

FIGURE 30-10: PARALLEL MASTER PORT READ TIMING DIAGRAM



Operating Conditions: 2.0V < V_{DD} < 3.6V, -40°C < T_A < +85°C unless otherwise stated.

TABLE 30-18: PARALLEL MASTER PORT READ TIMING REQUIREMENTS

Param. No	Symbol	Characteristics	Min	Typ	Max	Units
PM1		PMALL/PMALH Pulse Width	—	0.5 T_{CY}	—	ns
PM2		Address Out Valid to PMALL/PMALH Invalid (address setup time)	—	0.75 T_{CY}	—	ns
PM3		PMALL/PMALH Invalid to Address Out Invalid (address hold time)	—	0.25 T_{CY}	—	ns
PM5		PMRD Pulse Width	—	0.5 T_{CY}	—	ns
PM6		PMRD or PMENB Active to Data In Valid (data setup time)	—	—	—	ns
PM7		PMRD or PMENB Inactive to Data In Invalid (data hold time)	—	—	—	ns