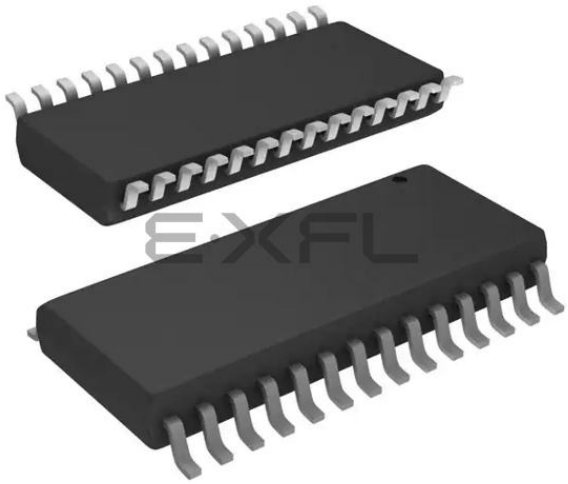


Welcome to E-XFL.COM

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, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
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 10x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f27j53t-i-so

TABLE 1-3: PIC18F2XJ53 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	28-SPDIP/ SSOP/ SOIC	28-QFN			
					PORTB (continued)
RB4/CCP4/KBI0/SCK1/SCL1/ RP7	25(2)	22(2)			
RB4			I/O	TTL/DIG	Digital I/O.
CCP4			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI0			I	TTL	Interrupt-on-change pin.
SCK1			I/O	ST/DIG	Synchronous serial clock input/output.
SCL1			I/O	I^2C	I^2C clock input/output.
RP7			I/O	ST/DIG	Remappable Peripheral Pin 7 input/output.
RB5/CCP5/KBI1/SDI1/SDA1/ RP8	26(2)	23(2)			
RB5			I/O	TTL/DIG	Digital I/O.
CCP5			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI1			I	TTL	Interrupt-on-change pin.
SDI1			I	ST	SPI data input.
SDA1			I/O	I^2C	I^2C data input/output.
RP8			I/O	ST/DIG	Remappable Peripheral Pin 8 input/output.
RB6/CCP6/KBI2/PGC/RP9	27(2)	24(2)			
RB6			I/O	TTL/DIG	Digital I/O.
CCP6			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI2			I	TTL	Interrupt-on-change pin.
PGC			I	ST	ICSP™ clock input.
RP9			I/O	ST/DIG	Remappable Peripheral Pin 9 input/output.
RB7/CCP7/KBI3/PGD/RP10	28(2)	25(2)			
RB7			I/O	TTL/DIG	Digital I/O.
CCP7			I/O	ST/DIG	Capture/Compare/PWM input/output.
KBI3			I	TTL	Interrupt-on-change pin.
PGD			I/O	ST/DIG	In-Circuit Debugger and ICSP programming data pin.
RP10			I/O	ST/DIG	Remappable Peripheral Pin 10 input/output.

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; I^2C = Open-Drain, I^2C specific

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

2: 5.5V tolerant.

REGISTER 4-6: DSWAKEL: DEEP SLEEP WAKE LOW BYTE REGISTER (BANKED F4Ah)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1
DSFLT	—	DSULP	DSWDT	DSRTC	DSMCLR	—	DSPOR
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 **DSFLT:** Deep Sleep Fault Detected bit
1 = A Deep Sleep Fault was detected during Deep Sleep
0 = A Deep Sleep Fault was not detected during Deep Sleep

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

bit 5 **DSULP:** Ultra Low-Power Wake-up Status bit
1 = An ultra low-power wake-up event occurred during Deep Sleep
0 = An ultra low-power wake-up event did not occur during Deep Sleep

bit 4 **DSWDT:** Deep Sleep Watchdog Timer Time-out bit
1 = The Deep Sleep Watchdog Timer timed out during Deep Sleep
0 = The Deep Sleep Watchdog Timer did not time out during Deep Sleep

bit 3 **DSRTC:** Real-Time Clock and Calendar Alarm bit
1 = The Real-Time Clock/Calendar triggered an alarm during Deep Sleep
0 = The Real-Time Clock /Calendar did not trigger an alarm during Deep Sleep

bit 2 **DSMCLR:** $\overline{\text{MCLR}}$ Event bit
1 = The $\overline{\text{MCLR}}$ pin was asserted during Deep Sleep
0 = The $\overline{\text{MCLR}}$ pin was not asserted during Deep Sleep

bit 1 **Unimplemented:** Read as '0'

bit 0 **DSPOR:** Power-on Reset Event bit
1 = The V_{DD} supply POR circuit was active and a POR event was detected(1)
0 = The V_{DD} supply POR circuit was not active, or was active, but did not detect a POR event

Note 1: Unlike the other bits in this register, this bit can be set outside of Deep Sleep.

FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

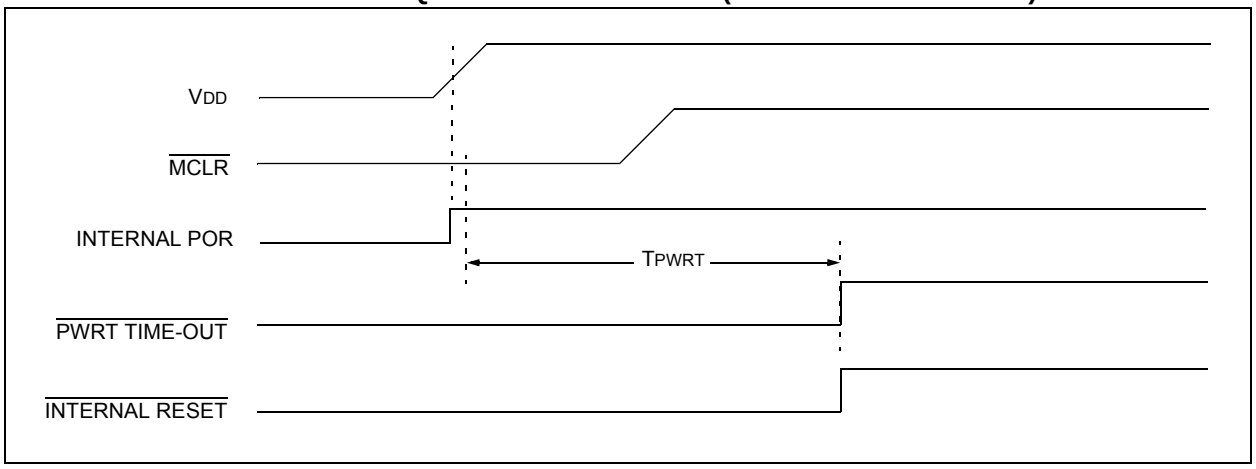


FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

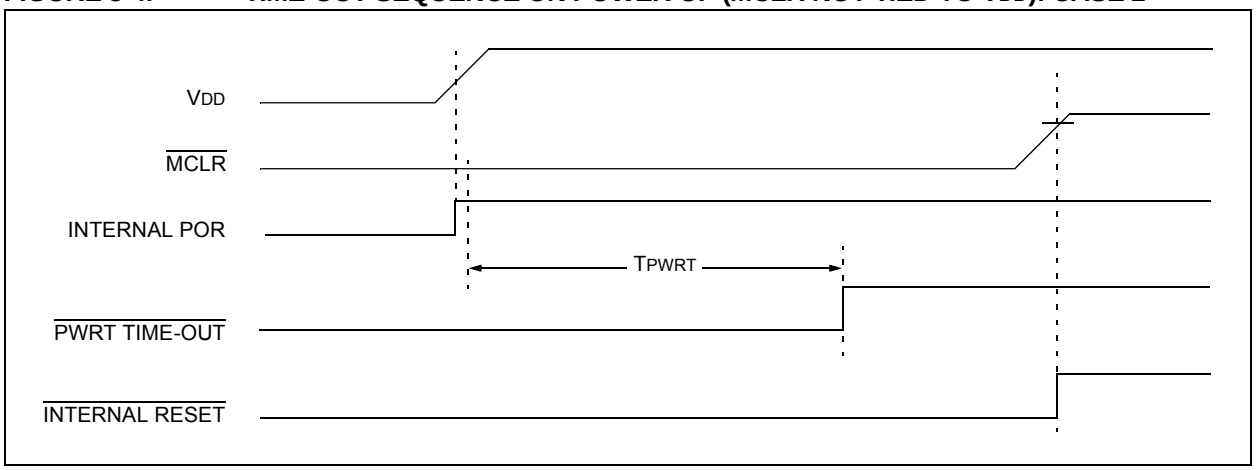
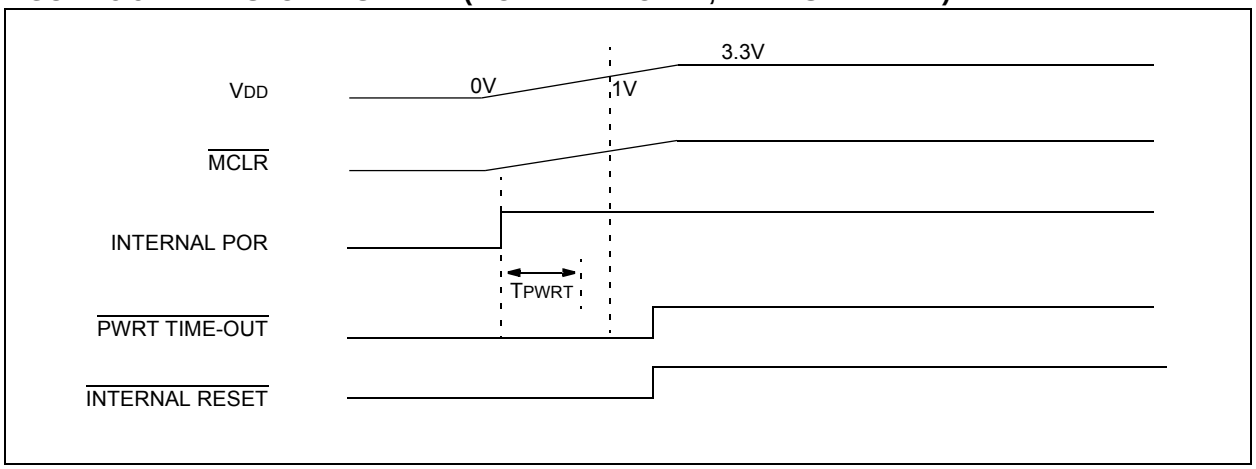


FIGURE 5-5: SLOW RISE TIME (MCLR TIED TO VDD, VDD RISE > TPWRT)



REGISTER 10-21: RPINR22: PERIPHERAL PIN SELECT INPUT REGISTER 22 (BANKED EFDh)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit 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 **SCK2R<4:0>:** Assign SPI2 Clock Input (SCK2) to the Corresponding RPn Pin bits

REGISTER 10-22: RPINR23: PERIPHERAL PIN SELECT INPUT REGISTER 23 (BANKED EFEh)

U-0	U-0	U-0	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1	R/$\overline{W}$-1
—	—	—	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit 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 **SS2R<4:0>:** Assign SPI2 Slave Select Input (SS2) to the Corresponding RPn Pin bits

REGISTER 10-23: RPINR24: PERIPHERAL PIN SELECT INPUT REGISTER 24 (BANKED EFFh)

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
—	—	—	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

Legend:	R/$\overline{W}$ = Readable bit, Writable bit 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 **OCFAR<4:0>:** Assign PWM Fault Input (FLT0) to the Corresponding RPn Pin bits

11.1.2 DATA REGISTERS

The PMP module uses eight registers for transferring data into and out of the microcontroller. They are arranged as four pairs to allow the option of 16-bit data operations:

- PMDIN1H and PMDIN1L
- PMDIN2H and PMDIN2L
- PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L
- PMDOUT2H and PMDOUT2L

The PMDIN1 registers are used for incoming data in Slave modes, and both input and output data in Master modes. The PMDIN2 registers are used for buffering input data in select Slave modes.

The PMADDR/PMDOUT1 registers are actually a single register pair; the name and function are dictated by the module's operating mode. In Master modes, the registers function as the PMADDRH and PMADDRL registers, and contain the address of any incoming or outgoing data. In Slave modes, the registers function as PMDOUT1H and PMDOUT1L, and are used for outgoing data.

PMADDRH differs from PMADDRL in that it can also have limited PMP control functions. When the module is operating in select Master mode configurations, the upper two bits of the register can be used to determine the operation of chip select signals. If these are not used, PMADDR simply functions to hold the upper 8 bits of the address. Register 11-9 provides the function of the individual bits in PMADDRH.

The PMDOUT2H and PMDOUT2L registers are only used in Buffered Slave modes and serve as a buffer for outgoing data.

11.1.3 PAD CONFIGURATION CONTROL REGISTER

In addition to the module level configuration options, the PMP module can also be configured at the I/O pin for electrical operation. This option allows users to select either the normal Schmitt Trigger input buffer on digital I/O pins shared with the PMP, or use TTL level compatible buffers instead. Buffer configuration is controlled by the PMPTTL bit in the PADCFG1 register.

13.2 Timer1 Operation

The Timer1 module is an 8-bit or 16-bit incrementing counter, which is accessed through the TMR1H:TMR1L register pair.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively.

When Timer1 is enabled, the RC1/CCP8/T1OSI/$\overline{UOE}$/RP12 and RC0/T1OSO/T1CKI/RP11 pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

13.3 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Register 13-1 displays the clock source selections.

13.3.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

13.3.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input, T1CKI, or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR Reset
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0)

when T1CKI is high, then Timer1 is enabled (TMR1ON = 1) when T1CKI is low.

TABLE 13-1: TIMER1 CLOCK SOURCE SELECTION

TMR1CS1	TMR1CS0	T1OSCEN	Clock Source
0	1	x	Clock Source (FOSC)
0	0	x	Instruction Clock (FOSC/4)
1	0	0	External Clock on T1CKI Pin
1	0	1	Oscillator Circuit on T1OSI/T1OSO Pin

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

R/W-0	R/W-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:			
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.

20.3.4 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPxCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCON1 registers and then set the SSPEN bit. This configures the SDIx, SDOx, SCKx and $\overline{SS}$x pins as serial port pins. For the pins to behave as the serial port function, the appropriate TRISx bits, PCFGx bits and Peripheral Pin Select registers (if using MSSP2) should be correctly initialized prior to setting the SSPEN bit.

A typical SPI serial port initialization process follows:

- Initialize the ODCON3 register (optional open-drain output control)
- Initialize the remappable pin functions (if using MSSP2, see **Section 10.7 "Peripheral Pin Select (PPS)"**)
- Initialize the SCKx/LAT value to the desired Idle SCKx level (if master device)
- Initialize the SCKx/PCFGx bit (if in Slave mode and multiplexed with the ANx function)
- Initialize the SCKx/TRIS bit as output (Master mode) or input (Slave mode)
- Initialize the SDIx/PCFGx bit (if SDIx is multiplexed with the ANx function)
- Initialize the SDIx/TRIS bit
- Initialize the $\overline{SS}$x/PCFG bit (if in Slave mode and multiplexed the with ANx function)
- Initialize the $\overline{SS}$x/TRIS bit (Slave modes)
- Initialize the SDOx/TRIS bit
- Initialize the SSPxSTAT register
- Initialize the SSPxCON1 register
- Set the SSPEN bit to enable the module

Any MSSP1 serial port function that is not desired may be overridden by programming the corresponding Data Direction (TRIS) register to the opposite value. If individual MSSP2 serial port functions will not be used, they may be left unmapped.

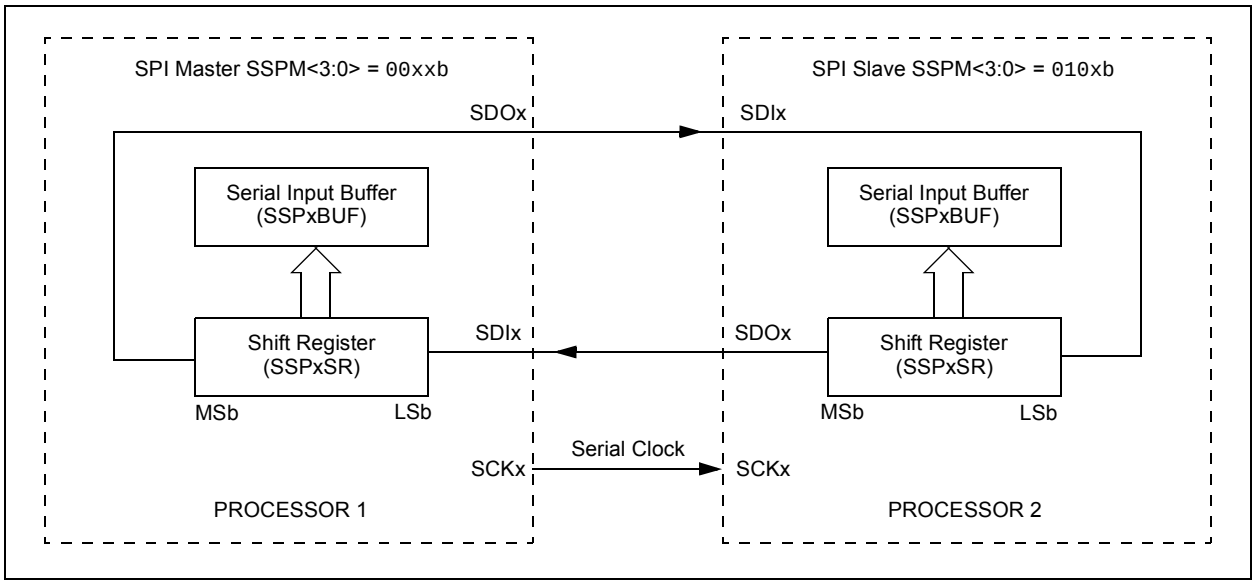
Note:	When MSSP2 is used in SPI Master mode, the SCK2 function must be configured as both an output and an input in the PPS module. SCK2 must be initialized as an output pin (by writing 0x0A to one of the RPORx registers). Additionally, SCK2IN must also be mapped to the same pin by initializing the RPINR22 register. Failure to initialize SCK2/SCK2IN as both output and input will prevent the module from receiving data on the SDI2 pin, as the module uses the SCK2IN signal to latch the received data.

20.3.5 TYPICAL CONNECTION

Figure 20-2 illustrates a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends valid data – Slave sends dummy data
- Master sends valid data – Slave sends valid data
- Master sends dummy data – Slave sends valid data

FIGURE 20-2: SPI MASTER/SLAVE CONNECTION



EXAMPLE 20-2: 512-BYTE SPI MASTER MODE `Init` AND TRANSFER

```
                                ;For this example, let's use RP5(RB2) for SCK2,
                                ;RP4(RB1) for SDO2, and RP3(RB0) for SDI2

                                ;Let’s use SPI master mode, CKE = 0, CKP = 0,
                                ;without using slave select signalling.

InitSPIPins:
   movlb     0x0F               ;Select bank 15, for access to ODCON3 register
   bcf       ODCON3, SPI2OD     ;Let’s not use open drain outputs in this example

   bcf       LATB, RB2          ;Initialize our (to be) SCK2 pin low (idle).
   bcf       LATB, RB1          ;Initialize our (to be) SDO2 pin to an idle state
   bcf       TRISB, RB1         ;Make SDO2 output, and drive low
   bcf       TRISB, RB2         ;Make SCK2 output, and drive low (idle state)
   bsf       TRISB, RB0         ;SDI2 is an input, make sure it is tri-stated

                                ;Now we should unlock the PPS registers, so we can
                                ;assign the MSSP2 functions to our desired I/O pins.

   movlb     0x0E               ;Select bank 14 for access to PPS registers
   bcf       INTCON, GIE        ;I/O Pin unlock sequence will not work if CPU
                                ;services an interrupt during the sequence
   movlw     0x55               ;Unlock sequence consists of writing 0x55
   movwf     EECON2             ;and 0xAA to the EECON2 register.
   movlw     0xAA
   movwf     EECON2
   bcf       PPSCON, IOLOCK     ;We may now write to RPINRx and RPORx registers
   bsf       INTCON, GIE        ;May now turn back on interrupts if desired

   movlw     0x03               ;RP3 will be SDI2
   movwf     RPINR21            ;Assign the SDI2 function to pin RP3

   movlw     0x0A               ;Let’s assign SCK2 output to pin RP4
   movwf     RPOR4              ;RPOR4 maps output signals to RP4 pin
   movlw     0x04               ;SCK2 also needs to be configured as an input on the
                                same pin
   movwf     RPINR22            ;SCK2 input function taken from RP4 pin
   movlw     0x09               ;0x09 is SDO2 output
   movwf     RPOR5              ;Assign SDO2 output signal to the RP5 (RB2) pin
   movlb     0x0F               ;Done with PPS registers, bank 15 has other SFRs

InitMSSP2:
   clrf      SSP2STAT           ;CKE = 0, SMP = 0 (sampled at middle of bit)
   movlw     b'00000000'        ;CKP = 0, SPI Master mode, Fosc/4
   movwf     SSP2CON1           ;MSSP2 initialized
   bsf       SSP2CON1, SSPEN    ;Enable the MSSP2 module

InitSPIDMA:
   movlw     b'00111010'        ;Full duplex, RX/TXINC enabled, no SSCON
   movwf     DMACON1            ;DLYINTEN is set, so DLYCYC3:DLYCYC0 = 1111
   movlw     b'11110000'        ;Minimum delay between bytes, interrupt
   movwf     DMACON2            ;only once when the transaction is complete
```

20.5.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPxCON2<4>). When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The BRG then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the BRG counts for TBRG; the SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the BRG is turned off and the MSSP module then goes into an inactive state (Figure 20-25).

20.5.12.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write does not occur).

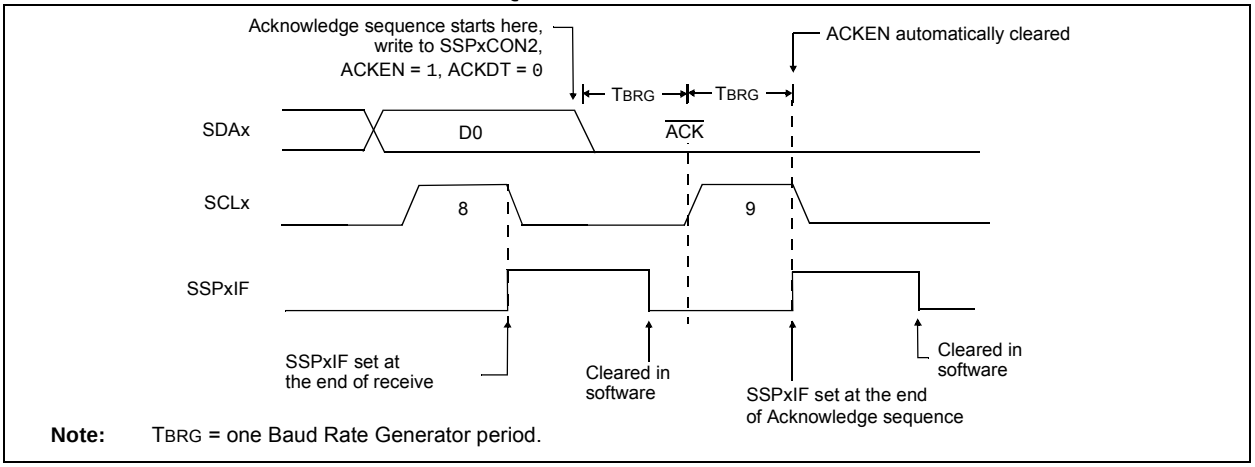
20.5.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPxCON2<2>). At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the BRG is reloaded and counts down to 0. When the BRG times out, the SCLx pin will be brought high and one Baud Rate Generator rollover count (TBRG) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the Stop bit (SSPxSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 20-26).

20.5.13.1 WCOL Status Flag

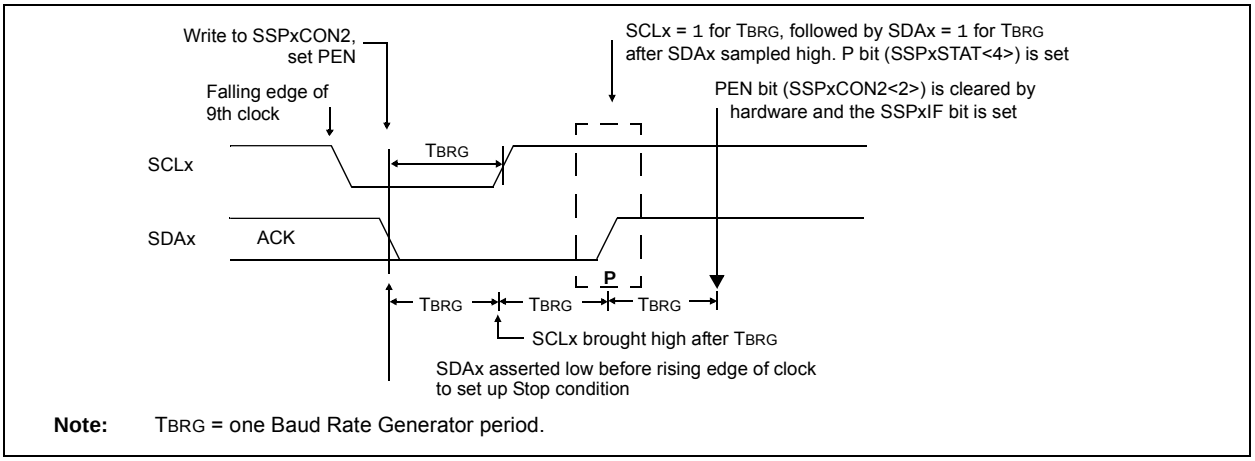
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 20-25: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 20-26: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

TABLE 21-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9615.	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

REGISTER 22-1: ADCON0: A/D CONTROL REGISTER 0 (ACCESS FC2h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VCFG1	VCFG0	CHS3(2)	CHS2(2)	CHS1(2)	CHS0(2)	GO/$\overline{\text{DONE}}$	ADON
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 **VCFG1:** Voltage Reference Configuration bit (V_{REF-} source)
1 = V_{REF-} (AN2)
0 = AV_{SS}(4)

bit 6 **VCFG0:** Voltage Reference Configuration bit (V_{REF+} source)
1 = V_{REF+} (AN3)
0 = AV_{DD}(4)

bit 5-2 **CHS<3:0>:** Analog Channel Select bits(2)
0000 = Channel 00 (AN0)
0001 = Channel 01 (AN1)
0010 = Channel 02 (AN2)
0011 = Channel 03 (AN3)
0100 = Channel 04 (AN4)
0101 = Channel 05 (AN5)(1)
0110 = Channel 06 (AN6)(1)
0111 = Channel 07 (AN7)(1)
1000 = Channel 08 (AN8)
1001 = Channel 09 (AN9)
1010 = Channel 10 (AN10)
1011 = Channel 11 (AN11)
1100 = Channel 12 (AN12)
1101 = (Reserved)
1110 = V_{DDCORE}
1111 = V_{BG} Absolute Reference (~1.2V)(3)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
When ADON = 1:
1 = A/D conversion is in progress
0 = A/D is Idle

bit 0 **ADON:** A/D On bit
1 = A/D Converter module is enabled
0 = A/D Converter module is disabled

Note 1: These channels are not implemented on 28-pin devices.
2: Performing a conversion on unimplemented channels will return random values.
3: For best accuracy, the band gap reference circuit should be enabled (ANCON1<7> = 1) at least 10 ms before performing a conversion on this channel.
4: On 44-pin, QFN devices, AV_{DD} and AV_{SS} reference sources are intended to be externally connected to V_{DD} and V_{SS} levels. Other package types tie AV_{DD} and AV_{SS} to V_{DD} and V_{SS} internally.

FIGURE 22-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

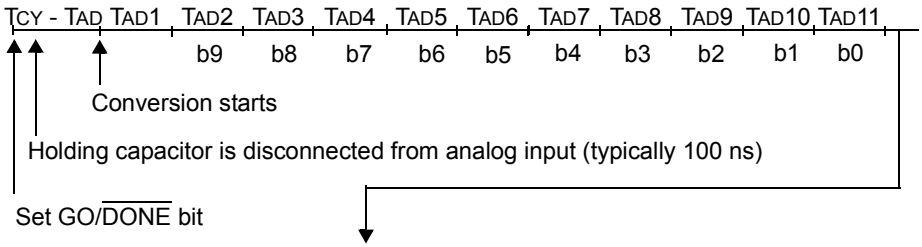
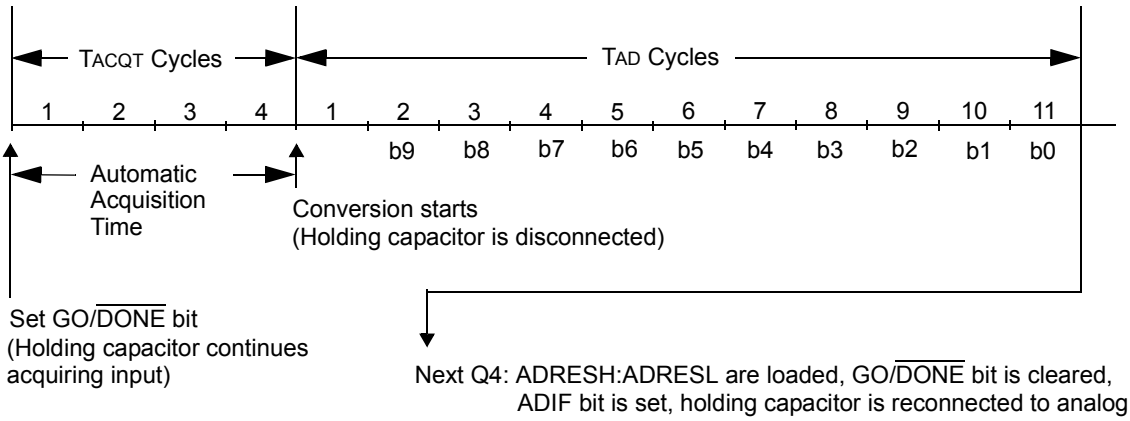


FIGURE 22-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



22.7 A/D Converter Calibration

The A/D Converter in the PIC18F47J53 family of devices includes a self-calibration feature, which compensates for any offset generated within the module. The calibration process is automated and is initiated by setting the ADCAL bit (ADCON1<6>). The next time the GO/DONE bit is set, the module will perform an offset calibration and store the result internally. Thus, subsequent offsets will be compensated.

Example 22-1 provides an example of a calibration routine.

The calibration process assumes that the device is in a relatively steady-state operating condition. If A/D calibration is used, it should be performed after each device Reset or if there are other major changes in operating conditions.

23.5 USB Interrupts

The USB module can generate multiple interrupt conditions. To accommodate all of these interrupt sources, the module is provided with its own interrupt logic structure, similar to that of the microcontroller. USB interrupts are enabled with one set of control registers and trapped with a separate set of flag registers. All sources are funneled into a single USB interrupt request, USBIF (PIR2<4>), in the microcontroller's interrupt logic.

Figure 23-7 provides the interrupt logic for the USB module. There are two layers of interrupt registers in the USB module. The top level consists of overall USB status interrupts; these are enabled and flagged in the UIE and UIR registers, respectively. The second level consists of USB error conditions, which are enabled and flagged in the UEIR and UEIE registers. An interrupt condition in any of these triggers a USB Error Interrupt Flag (UERRIF) in the top level.

Interrupts may be used to trap routine events in a USB transaction. Figure 23-8 provides some common events within a USB frame and its corresponding interrupts.

FIGURE 23-7: USB INTERRUPT LOGIC FUNNEL

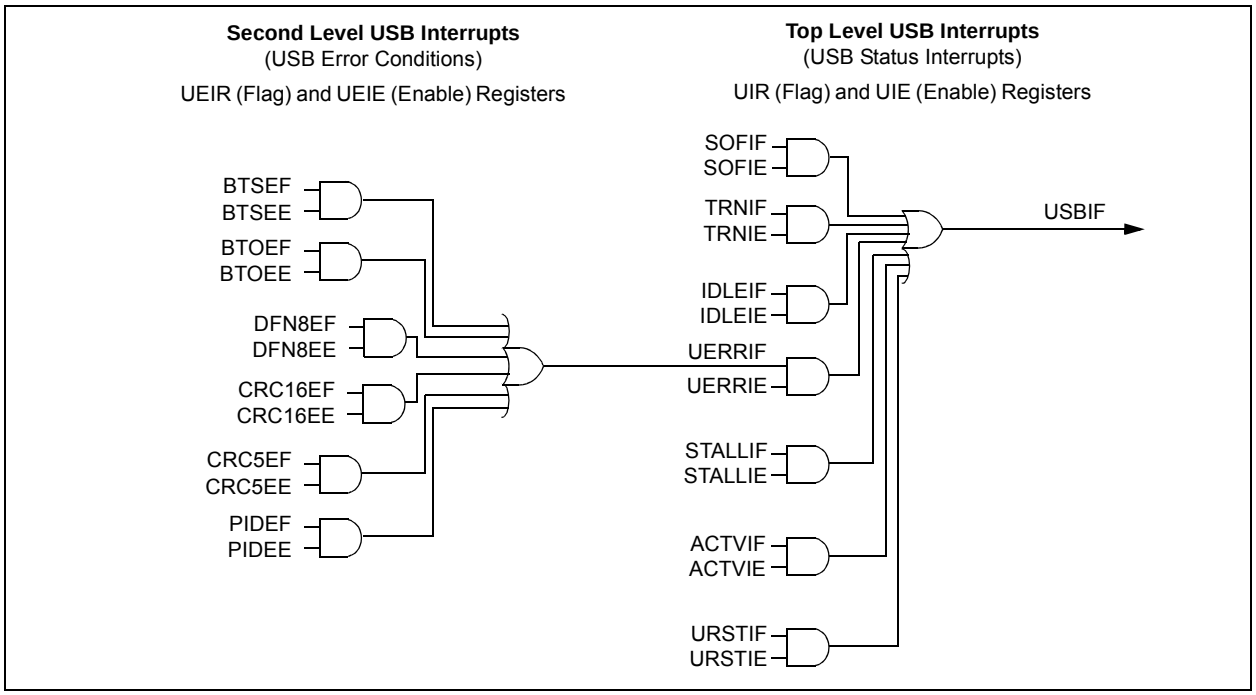
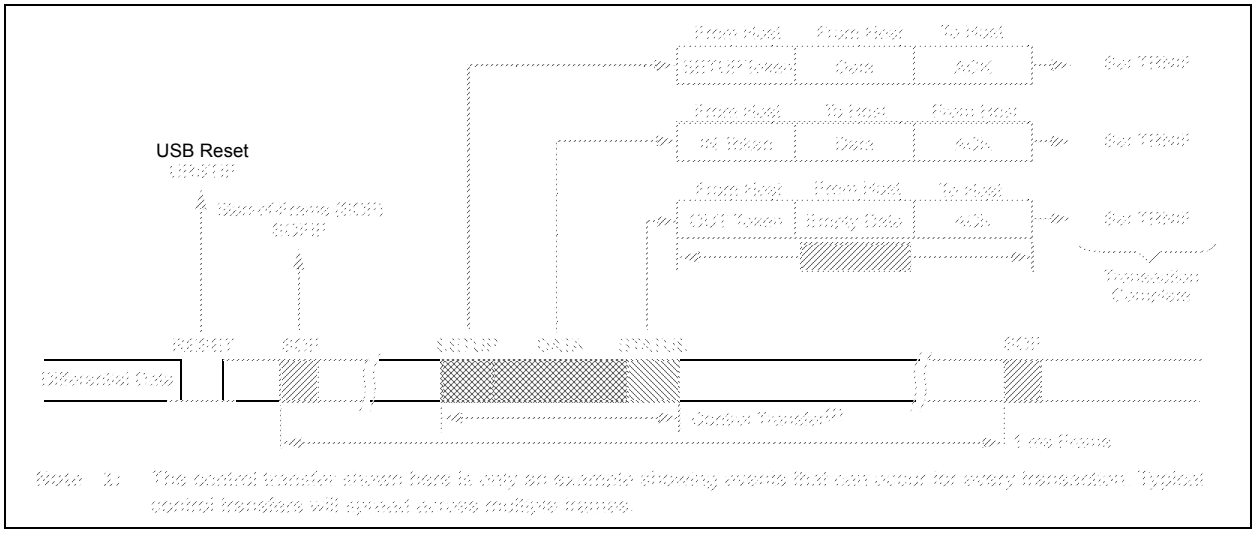


FIGURE 23-8: EXAMPLE OF A USB TRANSACTION AND INTERRUPT EVENTS



27.3.2 CAPACITANCE CALIBRATION

There is a small amount of capacitance from the internal A/D Converter sample capacitor as well as stray capacitance from the circuit board traces and pads that affect the precision of capacitance measurements. A measurement of the stray capacitance can be taken by making sure the desired capacitance to be measured has been removed. The measurement is then performed using the following steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT (= 1).
3. Wait for a fixed delay of time, *t*.
4. Clear EDG1STAT.
5. Perform an A/D conversion.
6. Calculate the stray and A/D sample capacitances:

$$C_{OFFSET} = C_{STRAY} + C_{AD} = (I \cdot t)/V$$

where *I* is known from the current source measurement step, *t* is a fixed delay and *V* is measured by performing an A/D conversion.

This measured value is then stored and used for calculations of time measurement or subtracted for capacitance measurement. For calibration, it is expected that the capacitance of $C_{STRAY} + C_{AD}$ is approximately known. C_{AD} is approximately 4 pF.

An iterative process may need to be used to adjust the time, *t*, that the circuit is charged to obtain a reasonable voltage reading from the A/D Converter. The value of *t* may be determined by setting C_{OFFSET} to a theoretical value, then solving for *t*. For example, if C_{STRAY} is theoretically calculated to be 11 pF, and *V* is expected to be 70% of V_{DD} or 2.31V, then *t* would be:

$$(4\ pF + 11\ pF) \bullet 2.31V/0.55\ \mu A$$

or 63 μs.

See Example 27-3 for a typical routine for CTMU capacitance calibration.

REGISTER 28-10: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F47J53 FAMILY DEVICES (BYTE ADDRESS 3FFFFFh)

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
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 7-0 **DEV<10:3>:** Device ID bits
These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.

DEV<10:3> (DEVID2<7:0>)	DEV<2:0> (DEVID2<7:5>)	Device
0101 1000	111	PIC18F47J53
0101 1000	101	PIC18F46J53
0101 1000	011	PIC18F27J53
0101 1000	001	PIC18F26J53
0101 1010	111	PIC18LF47J53
0101 1010	101	PIC18LF46J53
0101 1010	011	PIC18LF27J53
0101 1010	001	PIC18LF26J53

28.5.2 EXITING FAIL-SAFE OPERATION

The Fail-Safe Clock Monitor condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 2H (with any required start-up delays that are required for the oscillator mode, such as the OST or PLL timer). The INTRC oscillator provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The FSCM then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTRC oscillator. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

28.5.3 FSCM INTERRUPTS IN POWER-MANAGED MODES

By entering a power-managed mode, the clock multiplexer selects the clock source selected by the OSCCON register. FSCM of the power-managed clock source resumes in the power-managed mode.

If an oscillator failure occurs during power-managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTRC multiplexer. An automatic transition back to the failed clock source will not occur.

If the interrupt is disabled, subsequent interrupts while in Idle mode, will cause the CPU to begin executing instructions while being clocked by the INTRC source.

28.5.4 POR OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary device clock is either the EC or INTRC modes, monitoring can begin immediately following these events.

For HS or HSPLL modes, the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

Note:	The same logic that prevents false oscillator failure interrupts on POR, or wake-up from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 28.4.1 "Special Considerations For Using Two-speed Start-up"**, it is also possible to select another clock configuration and enter an alternate power-managed mode while waiting for the primary clock to become stable. When the new power-managed mode is selected, the primary clock is disabled.

28.6 Program Verification and Code Protection

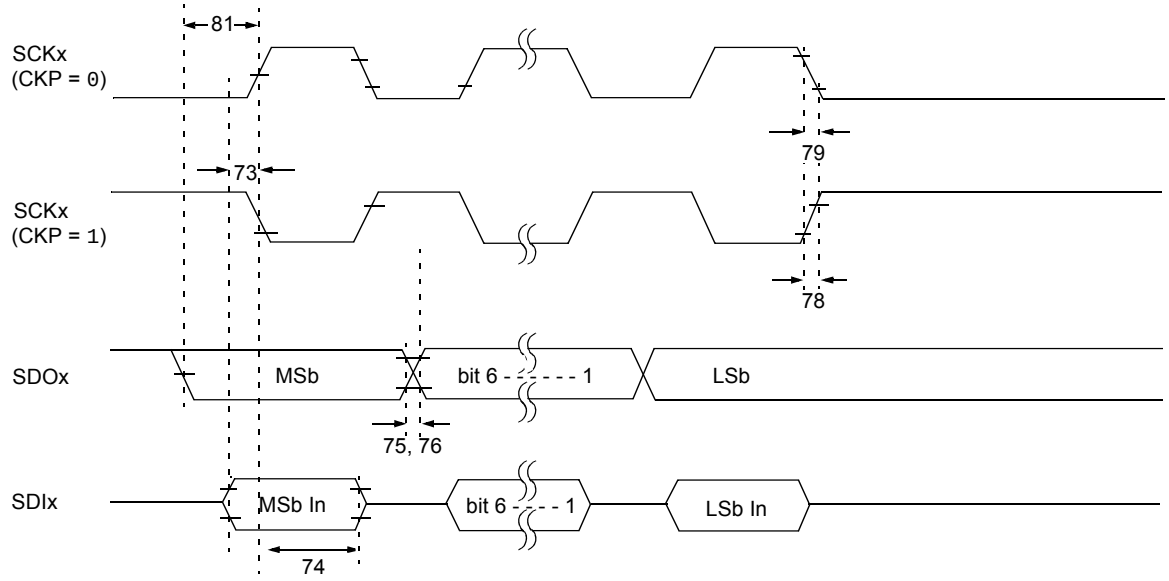
For all devices in the PIC18F47J53 family of devices, the on-chip program memory space is treated as a single block. Code protection for this block is controlled by one Configuration bit, CP0. This bit inhibits external reads and writes to the program memory space. It has no direct effect in normal execution mode.

28.6.1 CONFIGURATION REGISTER PROTECTION

The Configuration registers are protected against untoward changes or reads in two ways. The primary protection is the write-once feature of the Configuration bits, which prevents reconfiguration once the bit has been programmed during a power cycle. To safeguard against unpredictable events, Configuration bit changes resulting from individual cell level disruptions (such as ESD events) will cause a parity error and trigger a device Reset. This is seen by the user as a Configuration Mismatch (CM) Reset.

The data for the Configuration registers is derived from the FCW in program memory. When the CP0 bit is set, the source data for device configuration is also protected as a consequence.

FIGURE 31-13: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)



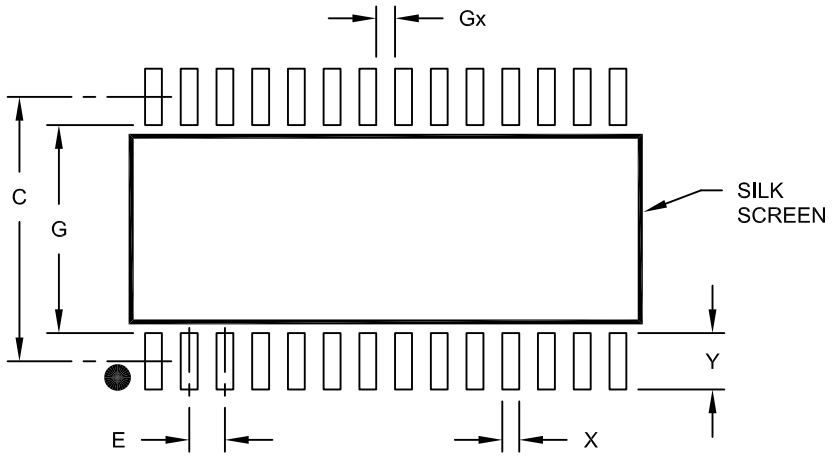
Note: Refer to Figure 31-4 for load conditions.

TABLE 31-21: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)

Param. No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
73	TDIV2SCH, TDIV2SCL	Setup Time of SDIx Data Input to SCKx Edge	35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			100	—	ns	VDD = 2.15V, VDDCORE = 2.15V
74	TSCH2DIL, TSCL2DIL	Hold Time of SDIx Data Input to SCKx Edge	30	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			83	—	ns	VDD = 2.15V
75	TDOR	SDOx Data Output Rise Time	—	25	ns	PORTB or PORTC
76	TDOF	SDOx Data Output Fall Time	—	25	ns	PORTB or PORTC
78	TSCR	SCKx Output Rise Time (Master mode)	—	25	ns	PORTB or PORTC
79	TSCF	SCKx Output Fall Time (Master mode)	—	25	ns	PORTB or PORTC
81	TDOV2SCH, TDOV2SCL	SDOx Data Output Setup to SCKx Edge	TCY	—	ns	

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A

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

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

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

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-0936-6