TB3151

Expanding 8-Bit Digital Communications Using Peripheral Pin Select (PPS) Technical Brief

Author: *Christopher Best*
Microchip Technology Inc.

INTRODUCTION

The PIC® microcontrollers offer many types of digital communication. Modules, such as the Master Synchronous Serial Port (MSSP), which includes I^2C and SPI communication, are included in most devices; however, often there is only one module available. What happens if both I^2C and SPI are needed, but there is only one MSSP available? In the past, this typically meant migrating to a higher pin count device that contains two individual MSSP modules. It is now possible to take advantage of the Peripheral Pin Select (PPS) module to solve this problem.

The Alternate Pin Function (APF) and Peripheral Pin Select (PPS) module connects digital peripheral inputs and outputs to the device I/O pins. Rather than having a single fixed I/O pin dedicated to a peripheral input or output, APF and PPS allow rerouting of the signals to other pins.

Alternate Pin Function

The Alternate Pin Function is similar to PPS, but only allows the digital signal to be routed to either its default I/O pin or a secondary alternate pin.

For example, the PIC12(L)F1822 implements the Alternate Pin Function. The Alternate Pin Function Control (APFCON) register is the register used to configure the routing of the signal. In this example, the following signals can be rerouted via the APFCON register:

- RX/DT (UART Receive)
- TX/CK (UART Transmit)
- SDO (SPI Serial Data Output)
- $\overline{SS}$ (Slave Select)
- T1G (Timer1 Gate)
- P1B (CCP1 PWM Output)
- CCP1/P1A (Capture/Compare Inputs, PWM Output)

Upon Power-on Reset (POR), the APFCON register configures each signal to a default pin location. The user can then change the routing of one or more signals based on their needs (see Register 1 for specific routing information).

REGISTER 1: APFCON: ALTERNATIVE PIN FUNCTION CONTROL REGISTER[(1)]

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RXDTSEL	SDOSEL	SSSEL	—	T1GSEL	TXCKSEL	P1BSEL	CCP1SEL
bit 7							bit 0

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

bit 7 **RXDTSEL:** Pin Selection bit
For 8-Pin Devices (PIC12(L)F1822):
0 = RX/DT function is on RA1
1 = RX/DT function is on RA5
For 14-Pin Devices (PIC16(L)F1823):
0 = RX/DT function is on RC5
1 = RX/DT function is on RA1

bit 6 **SDOSEL:** Pin Selection bit
For 8-Pin Devices (PIC12(L)F1822):
0 = SDO function is on RA0
1 = SDO function is on RA4
For 14-Pin Devices (PIC16(L)F1823):
0 = SDO function is on RC2
1 = SDO function is on RA4

bit 5 **SSSEL:** Pin Selection bit
For 8-Pin Devices (PIC12(L)F1822):
0 = $\overline{SS}$ function is on RA3
1 = $\overline{SS}$ function is on RA0
For 14-Pin Devices (PIC16(L)F1823):
0 = $\overline{SS}$ function is on RC3
1 = $\overline{SS}$ function is on RA3

bit 4 **Unimplemented**: Read as '0'

bit 3 **T1GSEL:** Pin Selection bit
0 = T1G function is on RA4
1 = T1G function is on RA3

bit 2 **TXCKSEL:** Pin Selection bit
For 8-Pin Devices (PIC12(L)F1822):
0 = TX/CK function is on RA0
1 = TX/CK function is on RA4
For 14-Pin Devices (PIC16(L)F1823):
0 = TX/CK function is on RC4
1 = TX/CK function is on RA0

bit 1 **P1BSEL:** Pin Selection bit
For 8-Pin Devices (PIC12(L)F1822):
0 = P1B function is on RA0
1 = P1B function is on RA4
For 14-Pin Devices (PIC16(L)F1823):
P1B function is always on RC4.

bit 0 **CCP1SEL**: Pin Selection bit
For 8-Pin Devices (PIC12(L)F1822):
0 = CCP1/P1A function is on RA2
1 = CCP1/P1A function is on RA5
For 14-Pin Devices (PIC16(L)F1823):
CCP1/P1A function is always on RC5

Note 1: Register 1 applies to the PIC12(L)F1822 and PIC16(L)F1823 devices only and is used for example purposes. Refer to the applicable data sheet for device-specific information.

Peripheral Pin Select

PPS comes in a couple of forms. In low pin count devices (6 to 20 pins), the PPS module allows a digital signal to be routed to any pin, with no limitations. In high pin count devices (28 to 64 pins), the PPS module allows a digital signal to be routed to any pin within a PORT, and there are typically two PORTs designated for a particular signal. For example, the PIC16(L)F1719 device's PPS module routes the I^2C clock and data lines to any pin within either PORTB or PORTC, but not PORTA, PORTD or PORTE. Table 1 lists the peripheral identifier, the associated register name and the available PORTs for each digital peripheral found in PIC16(L)F1717/8/9 devices. Table 2 lists the output signal name, specific output code and available PORTs for each digital peripheral found in PIC16(L)F1717/8/9 devices.

Input/Output Selection Registers

PPS uses multiple control registers for both input and output signals.

Input routing is accomplished through the Peripheral Input Selection (xxxPPS) register, in which the 'xxx' placeholder is replaced by the peripheral identifier. For example, the CLC1 Input register would be appropriately named CLC1PPS.

Register 2 is the Peripheral Input Selection register for the low pin count PIC16(L)F18323 device. Register 3 is the Peripheral Input Selection register for the high pin count PIC16(L)F1717/8/9 devices and should be used in combination with Table 1 to determine the correct input routing.

REGISTER 2: xxxPPS: PERIPHERAL xxx INPUT SELECTION(1)

U-0	U-0	U-0	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u
—	—	—	xxxPPS<4:0>				
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **xxxPPS<4:0>:** Peripheral xxx Input Selection bits

11xxx = Reserved; do not use
...
10111 = Peripheral input is RC7
01110 = Peripheral input is RC6
10101 = Peripheral input is RC5
10100 = Peripheral input is RC4
10011 = Peripheral input is RC3
10010 = Peripheral input is RC2
10001 = Peripheral input is RC1
10000 = Peripheral input is RC0
...
01111 = Peripheral input is RB7
01110 = Peripheral input is RB6
01101 = Peripheral input is RB5
01100 = Peripheral input is RB4
...
0011x = Reserved; do not use
00101 = Peripheral input is RA5
00100 = Peripheral input is RA4
00011 = Peripheral input is RA3
00010 = Peripheral input is RA2
00001 = Peripheral input is RA1
00000 = Peripheral input is RA0

Note 1: Register 2 applies to the PIC16(L)F18323 only and is used for example purposes. Refer to the applicable data sheet for device-specific information.

REGISTER 3: xxxPPS: PERIPHERAL xxx INPUT SELECTION[1]

U-0	U-0	U-0	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u
—	—	—	xxxPPS<4:3>		xxxPPS<2:0>		
bit 7							bit 0

Legend:	W = Writable bit	q = Value depends on peripheral
R = Readable bit	u = Bit is unchanged	U = Unimplemented bit, read as ‘0’
‘0’ = Bit is cleared	‘1’ = Bit is set	-n/n = Value at POR and BOR/Value at all other Resets

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-3 **xxxPPS<4:3>:** Peripheral xxx Input PORTx Selection bits
See Table 1 for the list of available PORTS for each peripheral.
11 = Peripheral input is from PORTD (PIC16(L)F18323 only)
10 = Peripheral input is from PORTC
01 = Peripheral input is from PORTB
00 = Peripheral input is from PORTA

bit 2-0 **xxxPPS<2:0>:** Peripheral xxx Input PORTx Selection bits
111 = Peripheral input is from PORTx bit 7 (Rx7)
110 = Peripheral input is from PORTx bit 6 (Rx6)
101 = Peripheral input is from PORTx bit 5 (Rx5)
100 = Peripheral input is from PORTx bit 4 (Rx4)
011 = Peripheral input is from PORTx bit 3 (Rx2)
010 = Peripheral input is from PORTx bit 2 (Rx2)
001 = Peripheral input is from PORTx bit 1 (Rx1)
000 = Peripheral input is from PORTx bit 0 (Rx0)

Note 1: Register 3 and Table 1 apply to the PIC16(L)F1717/8/9 device family only and are used for example purposes. Refer to the applicable data sheet for device-specific information.

REGISTER 4: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER[1]

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	RxyPPS<4:0>				
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits

11111 = Rxy source is DSM
11110 = Rxy source is CLKR
11101 = Rxy source is NCO1
11100 = Rxy source is TMR0
11011 = Rxy source is SDO2/SDA2
11010 = Rxy source is SCK2/SCL2
11001 = Rxy source is SDO1/SDA1
11000 = Rxy source is SCK1/SCL1
10111 = Rxy source is C2
10110 = Rxy source is C1
10101 = Rxy source is DT
10100 = Rxy source is TX/CK
10011 = Rxy source is CWG2D
10010 = Rxy source is CWG2C
10001 = Rxy source is CWG2B
10000 = Rxy source is CWG2A
01111 = Rxy source is CCP4
01110 = Rxy source is CCP3
01101 = Rxy source is CCP2
01100 = Rxy source is CCP1
01011 = Rxy source is CWG1D
01010 = Rxy source is CWG1C
01001 = Rxy source is CWG1B
01000 = Rxy source is CWG1A
00111 = Rxy source is CLC4OUT
00110 = Rxy source is CLC3OUT
00101 = Rxy source is CLC2OUT
00100 = Rxy source is CLC1OUT
00011 = Rxy source is PWM6
00010 = Rxy source is PWM5
00001 = Reserved
00000 = Reserved

Note 1: Register 4 applies to the PIC16(L)F18345 only and is used for example purposes. Refer to the applicable data sheet for device-specific information.

REGISTER 5: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER[(1)]

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	RxyPPS<4:0>				
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits
Selection code determines the output signal on the PORT pin.
(See Table 2 for supported PORTs and selection codes.)

Note 1: Register 5 and Table 2 apply to the PIC16(L)F1717/8/9 device family only and are used for example purposes. Refer to the applicable data sheet for device-specific information.

TABLE 1: PPS INPUT SELECTION CHART[1,2]

Peripheral	Register	PIC16(L)F1717/8/9		PIC16(L)F1718	PIC16(L)F1717/9	
		PORTA	PORTB	PORTC	PORTC	PORTD
PIN Interrupt	INTPPS	•	•			
Timer0 Clock	T0CKIPPS	•	•			
Timer1 Clock	T1CKIPPS	•		•	•	
Timer1 Gate	T1GPPS		•	•	•	
CCP1	CCP1PPS		•	•	•	
CCP2	CCP2PPS		•	•	•	
COG	COGINPPS		•	•		•
MSSP	SSPCLKPPS		•	•	•	
MSSP	SSPDATPPS		•	•	•	
MSSP	SSPSSPPS	•		•		•
EUSART	RXPPS		•	•	•	
EUSART	CKPPS		•	•	•	
All CLCs	CLCIN0PPS	•		•	•	
All CLCs	CLCIN1PPS	•		•	•	
All CLCs	CLCIN2PPS		•	•		•
All CLCs	CLCIN3PPS		•	•		•

Note 1: Example: CCP1PPS = 0x0B selects RB3 as the input for CCP1.

2: Register 3 and Table 1 apply to the PIC16(L)F1717/8/9 device family only and are used for example purposes. Refer to the applicable data sheet for device-specific information.

Output routing is performed using the Pin Rxy Output Source Selection Register (RxyPPS), in which the 'Rxy' placeholder is replaced by the pin identifier. For example, if the CCP1 output was to be routed to PORTC pin 3, the output register would be named RC3PPS.

The PPS output registers require a specific code to be entered into the RxyPPS<4:0> input field. Register 4 is the Output Source Selection register for the low pin count PIC16(L)F18345 device. Register 5 is the Output Source Selection register for the high pin count PIC16(L)F1717/8/9 devices and should be used in combination with Table 2 to determine the correct output routing.

TABLE 2: PPS OUTPUT SELECTION CHART[1,2]

RxyPPS<4:0>	Output Signal	PIC16(L)F1717/8/9		PIC16(L)F1718	PIC16(L)F1717/9		
		PORTA	PORTB	PORTC	PORTC	PORTD	PORTE
`11xxx`	Reserved						
`10111`	C2OUT	•		•			•
`10110`	C1OUT	•		•		•	
`10101`	DT		•	•	•		
`10100`	TX/CK		•	•	•		
`10011`	Reserved						
`10010`	Reserved						
`10001`	SDO/SDA		•	•	•		
`10000`	SCK/SCL		•	•	•		
`01111`	PWM4OUT		•	•		•	
`01110`	PWM3OUT		•	•			•
`01101`	CCP2		•	•	•		
`01100`	CCP1		•	•	•		
`01011`	COG1D		•	•		•	
`01010`	COG1C		•	•		•	
`01001`	COG1B		•	•		•	
`01000`	COG1A		•	•	•		
`00111`	CLC4OUT		•	•		•	
`00110`	CLC3OUT		•	•		•	
`00101`	CLC2OUT	•		•	•		
`00100`	CLC1OUT	•		•	•		
`00011`	NCO1OUT	•		•		•	
`00010`	Reserved						
`00001`	Reserved						
`00000`	LATxy	•	•	•	•	•	•

Note 1: Example: RB3PPS = 0x16 selects RB3 as the Comparator 1 output.

2: Register 5 and Table 2 apply to the PIC16(L)F1717/8/9 device family only and are used for example purposes. Refer to the applicable data sheet for device-specific information.

PPS Locking

The PPS module includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit in the PPSLOCK register. Locking/unlocking PPS requires a special sequence of instructions as an extra precaution against unwanted changes. Example 1 shows the lock/unlock sequence for the I^2C clock and data lines for the PIC16(L)F1719.

EXAMPLE 1: LOCK/UNLOCK CODE SEQUENCE FOR I^2C CLOCK AND DATA LINES FOR PIC16(L)F1719

```
GIE = 0;                                    // Disable Interrupts
PPSLOCK = 0x55;                             // First unlock code
PPSLOCK = 0xAA;                             // Second unlock code
PPSLOCKbits.PPSLOCKED = 0x00;               // Unlock PPS

SSPDATPPS = 0x000C;                         // RB4 is MSSP SDA input
SSPCLKPPS = 0x000E;                         // RB6 is MSSP SCL input
RB4PPS = 0b10001;                           // RB4 is MSSP SDA output
RB6PPS = 0b10000;                           // RB6 is MSSP SCL output

PPSLOCK = 0x55;                             // First lock code
PPSLOCK = 0xAA;                             // Second lock code
PPSLOCKbits.PPSLOCKED = 0x01;               // Lock PPS
GIE = 1;                                    // Enable Interrupts
```

The PPS can also be permanently locked by setting the PPS1WAY Configuration bit. When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows the PPS selection registers to be configured during initialization and locked until the next device Reset event.

EXPANDING COMMUNICATION

As mentioned in the introduction, PPS allows for a single pin to be used by more than one peripheral. The following examples highlight the flexibility of the PPS module.

MSSP Switching

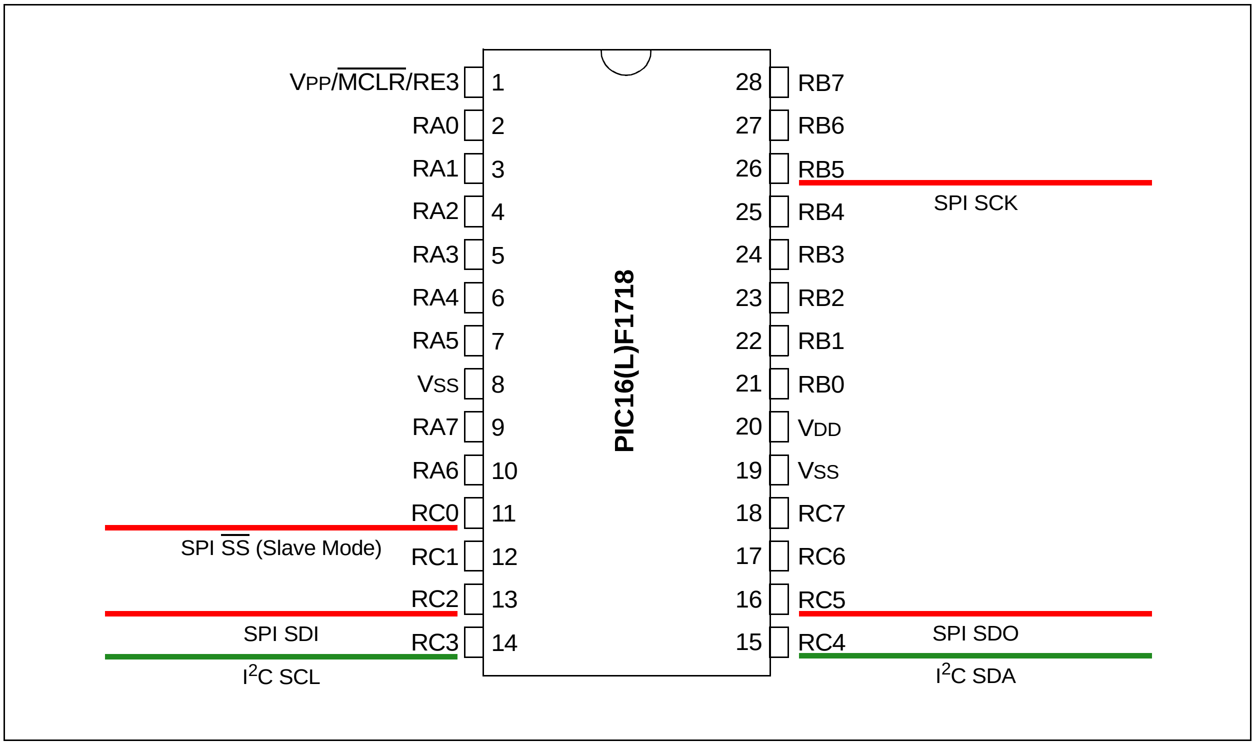
The Master Synchronous Serial Port (MSSP) contains both the Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I^2C) modules.

In typical applications using a PIC® MCU with only one MSSP module, the MSSP is configured for either SPI or I^2C, but not both. Previously, if both communication types were needed in an application, a PIC microcontroller that contained two individual MSSP modules could be used.

PPS allows the software to switch between the SPI and I^2C slave or master communication. When the application needs to communicate with a device on the I^2C bus, the software will reconfigure the MSSP registers for I^2C, and then reassign the I^2C clock (SCL) and data (SDA) lines via the PPS selection registers. When the application needs to use the SPI bus, the software reconfigures the MSSP for SPI, and then reassigns the SPI Clock (SCK), SPI Data In (SDI), SPI Data Out (SDO) and Slave Select ($\overline{SS}$) lines via the PPS selection registers.

Figure 1 shows a PIC16(L)F1718 with both I^2C and SPI bus lines. It is important to note that in this application, none of the bus lines are shared; SPI uses its own set of lines, as does the I^2C lines. Also, neither I^2C nor SPI can be used simultaneously; the MSSP can be configured for only one peripheral at a time.

FIGURE 1: I^2C AND SPI BUS LINES



Switching between the two peripherals can be accomplished by using the following steps (for this example, using the PIC16(L)F1718; the software starts with I^2C and switches to SPI):

1. Configure the MSSP for I^2C – Configure the appropriate MSSP registers to match the I^2C settings for the application.
2. Configure the PPS registers for I^2C – Configure the related MSSP PPS registers for I^2C usage and clear the unused SPI PPS registers. Both input and output registers must be configured so that both the SCL input and output, and SDA input and output are routed to the same pins. For example, Figure 1 shows the SCL line connected to RC3. In this case, SSPCLKPPS<4:0> = `0x13` and RC3PPS<4:0> = `0x10`, which route the SCL signal to RC3. Since the SPI SDO pin is not needed for I^2C, RC5PPS<4:0> should be cleared.
3. At this point, the I^2C bus lines are active, while the SPI lines are not. User software can now communicate via the I^2C bus.
4. Upon completion of the I^2C activity, user software clears the MSSP and PPS registers, disabling the MSSP and clearing all settings.
5. Configure the MSSP registers for SPI.
6. Configure the PPS registers for SPI.
7. Upon completion of the SPI communication, repeat Steps 1-7 as needed for communication.

Example 2 highlights the switching routine used to switch between I^2C and SPI in the PIC16(L)F1718.

EXAMPLE 2: CODE SEQUENCE FOR I²C/SPI SWITCHING

```
void communicate_I2C(void)
{
    MSSP_Mode_Switch(I2C);                          // PPS switch routine
    get_I2C_data();                                 // Communicate via I2C
}
void communicate_SPI(void)
{
    MSSP_Mode_Switch(SPI);                          // PPS switch routine
    get_SPI_data();                                 // Communicate via SPI
}
void MSSP_Mode_Switch(bool Mode)                    // Switch between I2C and SPI
{
    if (Mode == 1)                                  // I2C mode
    {
        SSP1STAT = 0x80;                            // Sample end of data output
        SSP1CON1 = 0x28;                            // MSTR SCL = FOSC/4(SSPxADD+1)
        SSP1CON3 = 0x00;
        SSP1ADD = 0x13;                             // 0x13 = 100 kHz Clock
        SSPCLKPPS = 0x00;                           // Clear SCL input
        RB5PPS = 0x00;                              // Clear SCL output
        SSPDATPPS = 0x00;                           // Clear SDA input
        RC5PPS = 0x00;                              // Clear SDA output

        SSPCLKPPS = 0x13;                           // RC3-> MSSP:SCL input
        RC3PPS = 0x10;                              // RC3-> MSSP:SCL output
        SSPDATPPS = 0x14;                           // RC4-> MSSP:SDA input
        RC4PPS = 0x11;                              // RC4-> MSSP:SDA output
    }
    if (Mode == 0)                                  // SPI mode
    {
        SSP1CON1bits.CKP = 0;                       // Idle clock is a low level
        SSP1STATbits.CKE = 1;                       // Transmit active to idle
        SSP1STATbits.SMP = 0;                       // Input data sampled at
                                                    // Middle of data output time
        SSP1CON1bits.SSPM = 0b0000;                 // SPI master, clock = FOSC/4
        SSP1CON1bits.SSPEN = 1;                     // Enable MSSP module
        SSPCLKPPS = 0x00;                           // Clear SCL input
        RC3PPS = 0x00;                              // Clear SCL output
        SSPDATPPS = 0x00;                           // Clear SDA input
        RC4PPS = 0x00;                              // Clear SDA output

        SSPCLKPPS = 0x0D;                           // RB5-> MSSP:SCK output
        RB5PPS = 0x10;                              // RB5-> MSSP:SCK input
        SSPDATPPS = 0x12;                           // RC2-> MSSP:SDI input
        RC5PPS = 0x11;                              // RC5-> MSSP:SDO output
    }
}
```

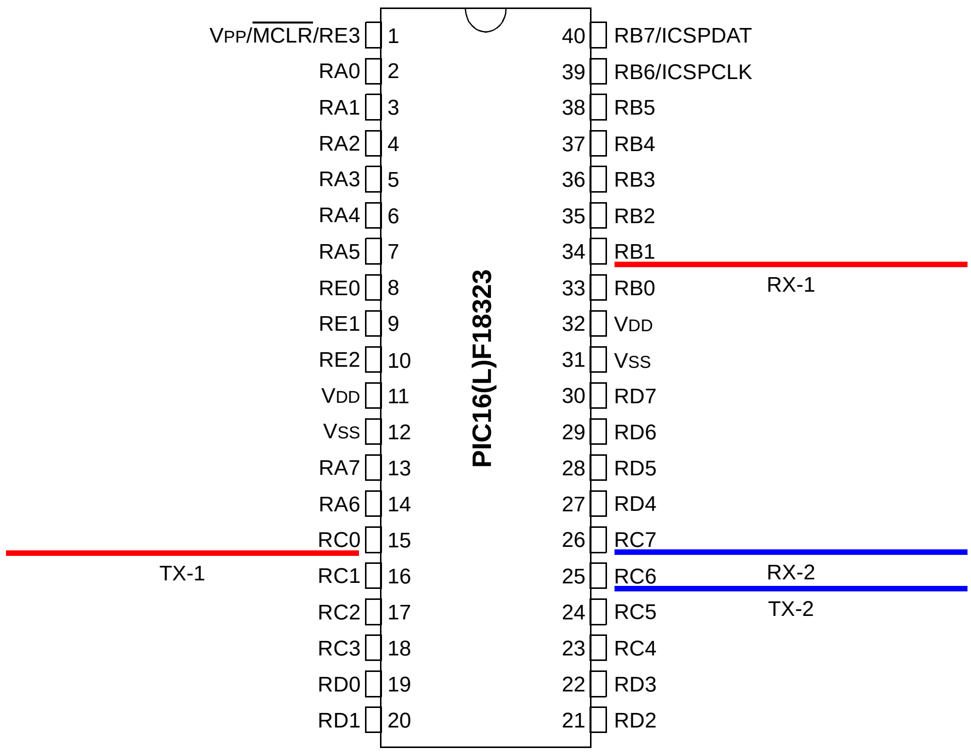
UART Switching

The same principle applies to switching between UART channels. UART bus lines are typically meant for a single master and a single slave. Multiple slave devices can receive simultaneous data streams from a master, but must have their RX inputs tri-stated when not receiving data from a master or another slave. Issues may arise since the user may not have control over the slave input configuration.

If the application requires more than one UART device on the bus, PPS can be used to switch between multiple bus lines.

Figure 2 shows a PIC16(L)F1719 with two sets of RX and TX lines.

FIGURE 2: UART RX/TX LINES



Switching between the two sets of RX/TX lines can be accomplished by using the following steps (for this example, using the PIC16(L)F1719, the software starts with RX/TX-1 and switches to RX/TX-2):

1. Clear all UART registers.
2. Clear all associated PPS registers (RXPPS, TXPPS, etc.)
3. Configure UART registers (setting Baud Rate, etc.)
4. Configure the RXPPS register to match the desired input pin.
5. Configure the RxyPPS register to match the desired TX output pin.
6. At this point, the UART is configured and ready for use.
7. Upon completion of communication with first UART device, user software clears all UART and PPS registers.
8. Configure UART registers.
9. Configure PPS registers to match new RX/TX pins.
10. Upon completion of communication, repeat Steps 1-9 as necessary for communication.

Example 3 highlights the switching routine used to switch between UART lines in the PIC16(L)F1719.

EXAMPLE 3: SWITCHING BETWEEN UART LINES CODE SEQUENCE

```
/* PIC16F1719 EUSART Code Snippet */

#define UART1 0
#define UART2 1

void  communicate_UART1(void)                        // Communicate with Device 1
{
      UART_Lines_Switch(UART1);                      // PPS switch to UART1 lines
      RX_TX_Device1();                               // Communicate
}
void  communicate_UART2(void)                        // Communicate with Device 2
{
      UART_Lines_Switch(UART2);                      // PPS switch to UART2 lines
      RX_TX_Device2();                               // Communicate
}
void  UART_Lines_Switch(uint8_t Module)              // PPS switching
{
      BAUD1CON = 0x00;
      RC1STA = 0x00;                                 // Clear UART registers
      TX1STA = 0x00;
      SP1BRGL = 0x00;
      TX1REG = 0x00;
      RC1REG = 0x00;
      LATCbits.LATC0 = 1;                            // Ensure output lines idle high
      LATCbits.LATC6 = 1;
      RXPPSbits.RXPPS = 0b00000;                     // Clear all related
      RC0PPSbits.RC0PPS = 0b00000;                   // PPS registers
      RC7PPSbits.RC7PPS = 0b00000;
      RB2PPSbits.RB2PPS = 0b00000;

      if(Module == 0)                                // Device 1 (UART1)
      {
            BAUD1CON = 0x40;                         // Set Baud Rate
            RC1STA = 0x90;                           // Enable UART, cont. receive
            TX1STA = 0x22;                           // Enable transmit, async
            SP1BRGL = 0x19;                          // 19.2K @ 32MHz
            TX1REG = 0x00;
            RC1REG = 0x00;

            RXPPSbits.RXPPS = 0b01010;               // PIC RX ON RB2
            RC0PPSbits.RC0PPS = 0b10100;             // PIC TX ON RC0
      }
      if(Module == 1)                                // Device 2 (UART2)
      {
            BAUD1CON = 0x40;
            RC1STA = 0x90;                           // Enable UART, cont. receive
            TX1STA = 0x22;                           // Enable transmit, async
            SP1BRGL = 0x19;                          // 19.2K @ 32MHz
            TX1REG = 0x00;
            RC1REG = 0x00;

            RXPPSbits.RXPPS = 0b10111;               // PIC RX ON RC7
            RC6PPSbits.RC6PPS = 0b10100;             // PIC TX ON RC6
      }
}
```

NOTES:

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: 978-1-5224-0485-9

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

07/14/15