dsPIC30F Peripheral Module

dsPIC30F SPI Module

Session Agenda

- Module Overview
- SPI Transmission
- SPI Reception
- Framed SPI
- Additional Features

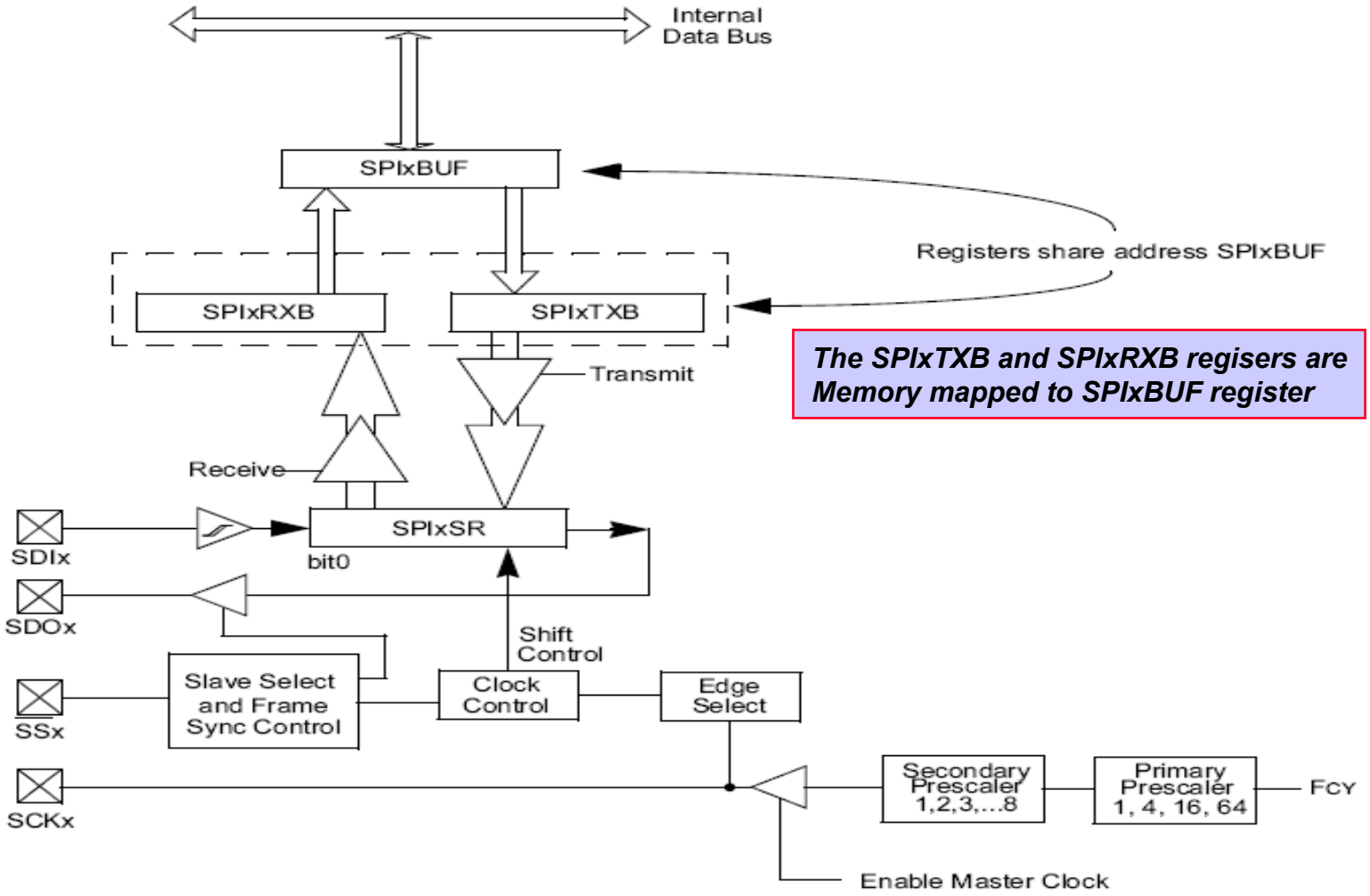
SPI - Overview

- Serial transmission and reception of 8-bit or 16-bit data
  - Full-duplex, synchronous communication
  - Compatible with Motorola's SPI and SIOP (simple. synchronous serial I/O port) interfaces
  - 3-wire interface
  - Supports 4 different clock formats and serial clock speeds up to 10 Mbps
  - Buffered Transmission and Reception

SPI - Applications

- Interfacing with memory devices
  - Serial EEPROMs - e.g., 25xx256
- Interfacing with codecs
  - Control Ports or PCM Data
- Interfacing with communication chips
  - TCP/IP (ENC28J60)
- Interfacing with LCD controllers
  - PICmicro MCUs - e.g., 18F8490
- Boot Loader

SPI - Block Diagram



The SPIxTXB and SPIxRXB regisers are Memory mapped to SPIxBUF register

SPI Interface Pin

- SDI : Serial Data Input
- SDO : Serial Data Output
- SCK : Shift Clock input or output
- SS : Active low select or frame synchronization I/O pulse

The SPI module can be configured to operate using 3 or 4 pins, In the 3-pin mode the SS pin is not used.

SPIxCON Register

SPIxCON Register

-	FRMEN	SPIFSD	-	DISSDO	MODE16	SMP	CKE
bit15	14	13	12	11	10	9	bit8

- FRMEN : Framed SPI Support bit
- SPIFSD : Frame Sync Pulse Direction Control on SS pin
- DISSDO : Disable SDO pin
- MODE16 : Word/Byte Communication Select
- SMP : SPI Data Input Simple phase select
- CKE : SPI Clock Edge Select
  - The CKE bit is not used in the Framed SPI modes.

SPIxCON Register (cont.)

SPIxCON Register

SSEN	CKP	MSTEN	SPRME<2>	SPRME<1>	SPRME<0>	PPRE<1>	PPRE<0>
bit15	14	13	12	11	10	9	bit8

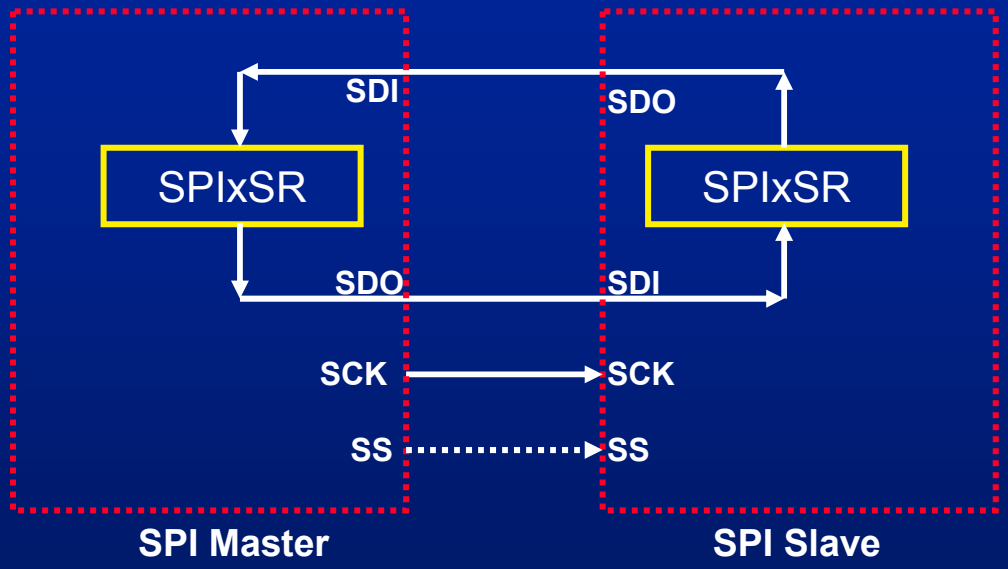
- SSEN : Slave Select Enable (Slave Mode)
- CKP: Clock Polarity Select
- MSTEN : Master Mode Enable
- SPRME<2:0> : Secondary Prescale (Master Mode)
  - 1:1 through 8:1
- PPRE<1:0> : Primary Prescale (Master Mode)
  - 1:1 , 4:1 , 16:1 and 64:1

SPI - Master / Slave

- SPI module can be configured as Master or Slave
  - In any SPI data transfer, there is a single Master and a single Slave
    - Selected by MSTEN bit, SPIxCON<5>
- Master generates serial clock pulse (on SCK pin)
  - SCK frequency determined by Primary Prescaler bits (PPRE) and Secondary Prescaler (SPRE) bits in SPIxCON register
  - **Fsck = Fcy / (PPRE * SPRE)**

SPI - Master / Slave Connection

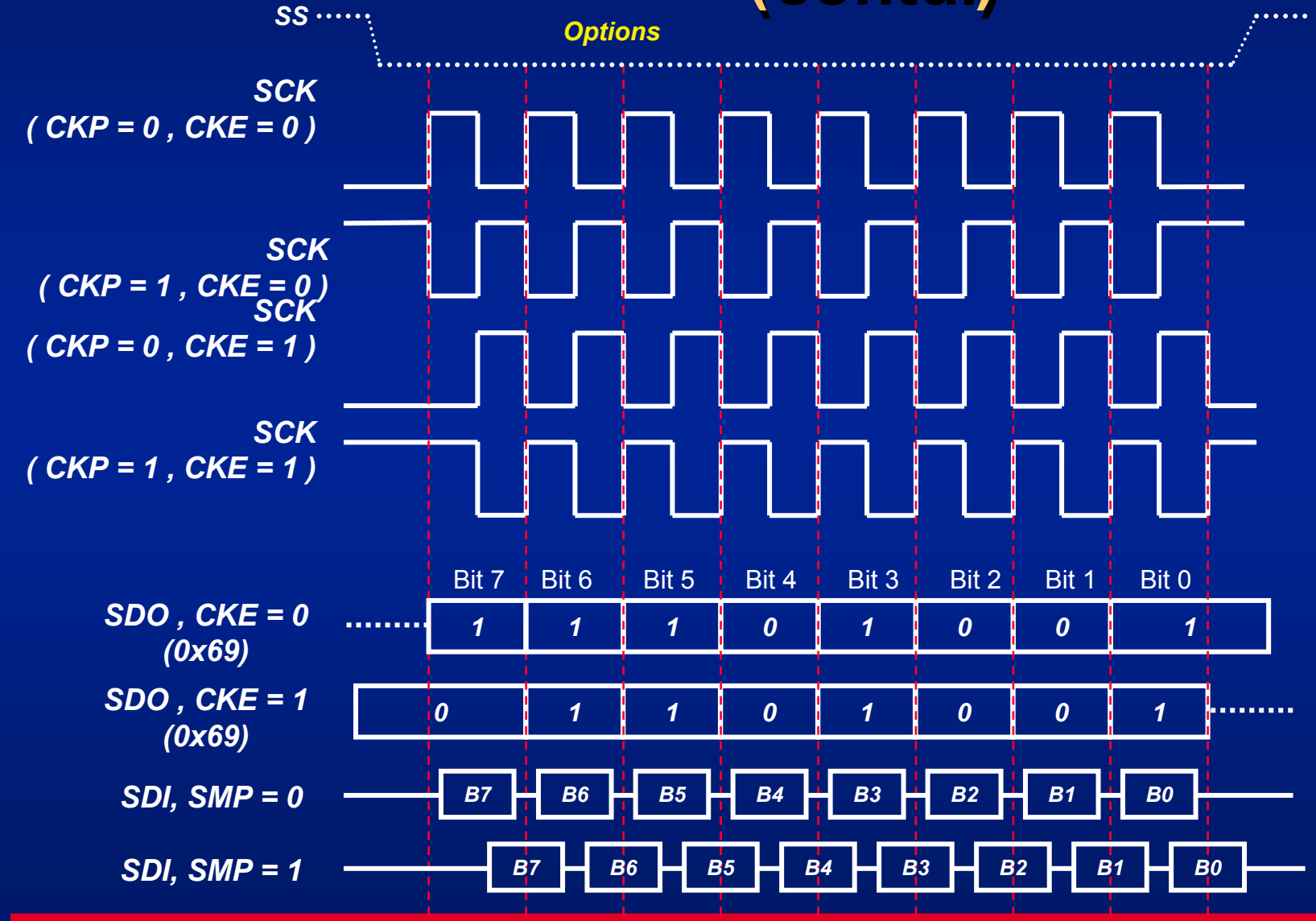
- As each transmitted bit gets shifted out though the SDO output pin, a received bit is simultaneously shifted in through the SDI pin
  - Synchronous full-duplex communications
  - Shift Register (SPIxSR) used for data transfer



SPI - Serial Clock Formats

- 4 clock formats - set by CKP and CKE bits in the SPIxCON register
  - SCK is low when module is idle, SDO changes on clock going high (CKP=0, CKE=0)
  - SCK is low when module is idle, SDO changes on clock going low (CKP=0, CKE=1)
  - SCK is high when module is idle, SDO changes on clock going low (CKP=1, CKE=0)
  - SCK is high when module is idle, SDO changes on clock going high (CKP=1, CKE=1)

SPI - Serial Clock Formats (contd.)



SPI - Transmission

- Module is enabled by setting SPIEN bit in the SPIxSTAT register
- Transmission begins when data is written into the Master's Transmit Buffer
  - SCK pulses are generated by the Master only when SPIxSR contains data
- Transmission can be disabled by setting the DISSDO bit in the SPIxCON register

SPI - Transmission (contd.)

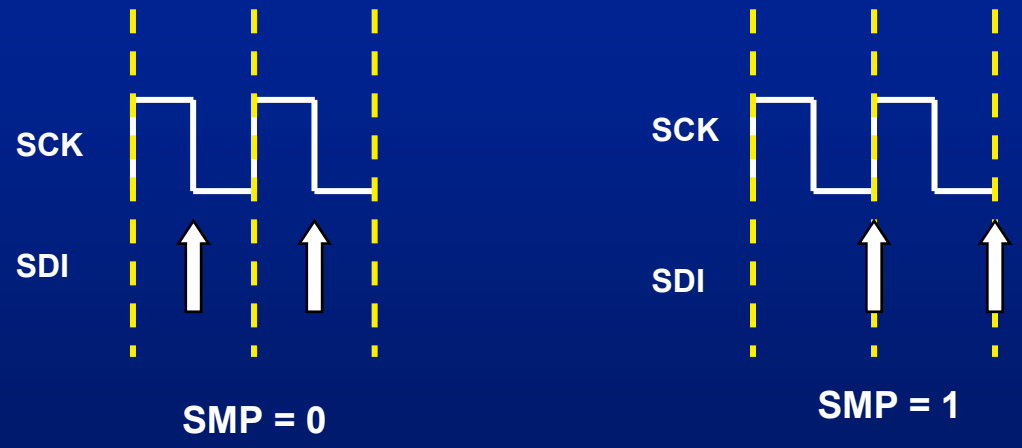
- SPIxBUF is buffered
  - You can write SPIxBUF while data is being shifted out through SPIxSR
  - SPITBF bit in the SPIxSTAT register indicates that the Transmit Buffer is full
    - Wait until SPITBF = 0 to write data
  - Transmission of the new data starts as soon as SPIxSR is idle

SPI - Reception

- Reception occurs concurrently with the transmission
  - When all bits of data have been shifted in through SPIxSR, SPIxSR contents are transferred to Receive Buffer
  - SPI interrupt (indicated by SPIIF bit and enabled by SPIIE bit) is generated so that buffer can be read

SPI - Reception (contd.)

- Incoming data on the SDI pin is sampled either in the middle or end of each bit period, depending on the value of SMP bit in SPIxCON register
  - Module forces SMP = 0 for Slave



SPI - Reception (contd.)

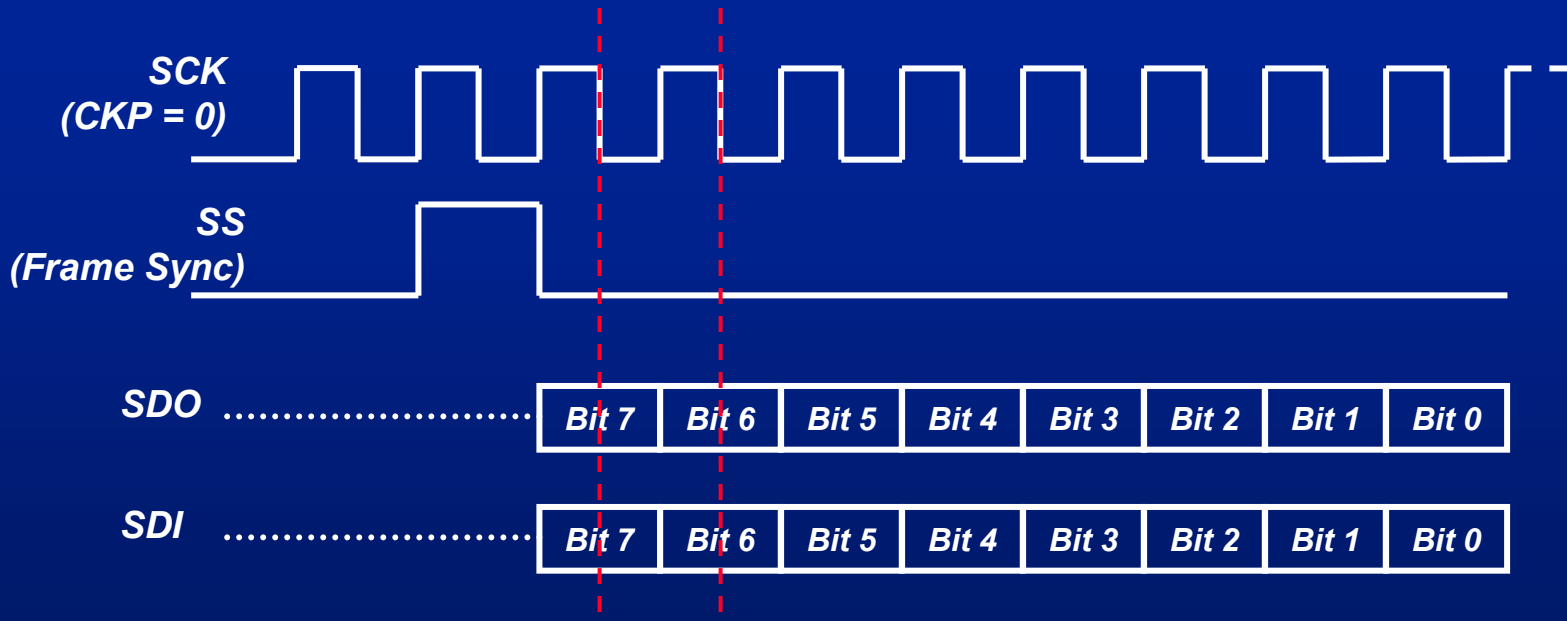
- SPIxBUF subject to Receive Overflow
  - SPIRBF bit in the SPIxSTAT register = 1 indicates that the Receive Buffer is full
  - SPIxBUF must be read before new data is completely shifted in
  - When receive overflow occurs...
    - New data not transferred to ReceiveBuffer
    - SPIROV bit in SPIxSTAT is set

SPI - Data Sizes

- 8-bit and 16-bit data communication
  - SPI operation is identical for both data sizes, except number of bits transmitted
    - For 8-bit data, Master generates 8 SCK pulses
    - For 16-bit data, Master generates 16 SCK pulses
- 16-bit operation is selected by setting the MODE16 bit in the SPIxCON register

SPI - Framed SPI

- SPI supports Frame Synchronization
  - Enabled by setting FRMEN bit in the SPIxCON register
  - SCK pulses are continuous in this mode



SPI - Framed SPI (contd.)

- Frame Master generates Frame Sync pulses
- Frame Master or Slave mode is selected by clearing or setting the SPIFSD bit in the SPIxCON register
- Shifting of data starts only after a Frame Sync pulse is generated on the SS pin
- 4 possible Framed SPI modes
  - SPI Master, Frame Master
  - SPI Master, Frame Slave
  - SPI Slave, Frame Master
  - SPI Slave, Frame Slave

SPI - Additional Features

- Slave Select (SS) pin functionality
  - In this mode, the Slave functions only as long as the SS pin is driven low
  - Enabled by setting SSEN bit in the SPIxCON register
- Slave Wake-up from SLEEP
  - Since SCK pulses are provided by the Master, SPI Slave can function in SLEEP
  - Slave Reception wakes up the device from SLEEP

SPI Master Mode operation

- If using interrupt
  - Clear the SPIxIF in respective IFSn register
  - Set the SPIxIE bit in the respective IECn register
  - Write the SPIxIP in the respective IPCn register
- Write the desired setting to the SPIxCON register with MSTEN (SPIxCON<5>) =1
- Clear the SPIROV bit (SPIxSTAT<6>)
- Enable SPI operation by setting the SPIEN bit
- Write the data to be transmitted to the SPIxBUF register, Transmission will start as soon as data is written to the SPIxBUF register.

SPI Slave Mode Operation

- Clear the SPIxBUF register
- If using interrupt
  - Clear the SPIxIF in respective IFSn register
  - Set the SPIxIE bit in the respective IECn register
  - Write the SPIxIP in the respective IPCn register
- Write the desired setting to the SPIxCON register with MSTEN (SPIxCON<5>) = 0
- Clear the SMP = 0
- If the CKE is set, then the SSEN bit must be set, thus enabling the SSx pin
- Clear the SPIROV bit
- Enable SPI operation by setting the SPIEN bit

Jumper Setting for Lab1

- Caption !!
  - I2C, SPI, UART and PGC/PGD share both pin 25 & 26
  - Need to use the jumper to change the debug pin to EMUC1 and EMUD1 (default are PGC & PGD)
  - On the Configuration Bits select “Use the EMUC1 and EMUD1)
- So,
  - Program Mode : Set DSW2 1&2 to ON, 3&4 are OFF position
  - Debug Mode : Set DSW2 1&2 to OFF, 3&4 are ON position

SPI Lab1 (Master Mode)

- Initialize the dsPIC30F4011 to an SPI Master
- Set 1 Second time base using Timer1 to read the ADC result form both VR1 and VR2
- Put ADC Value on the LCD module (MSB Only)
- Use the SPI Write Command to write the ADC value to the external EEPROM (25LC160A) at location 0x10 (VR1) and 0x20 (VR2) address
- Read 25LC160A with SPI Read Command to read EEPROM data from 0x10 and 0x20 then display on the LCD Module at Second Line

SPI Lab2 (Slave Mode)

- Initialize the dsPIC30F4011 to a SPI Slave Device
- Use the _SPI1_Interrupt to process the SPI input data
- Master send out both value for VR1 and VR2 to Slave, Slave side has to receive the date with 64 bytes receiver buffer (simulate EEPROM data area)
- Slave display received data on the LCD line 2
- Slave has capability simulate the 25LC160A that cab be Read/Write thorough SPI command