

```

//#####
// FILE:   Flax_DSP2833x_Spi.c
// TITLE:  DSP2833x SPI Initialization & Support Functions.
//#####

#include "DSP2833x_Device.h"    // DSP2833x Headerfile Include File
#include "DSP2833x_Examples.h"  // DSP2833x Examples Include File

/*****/
//-----
// InitSPI:
//-----
// This function initializes the SPI(A) to a known state.
void FlaxInitSpiParam(void)
{
    EALLOW;
    SpiaRegs.SPICCR.all = 0x0007;    // Reset on, rising edge, 16-bit char bits
    SpiaRegs.SPICTL.all = 0x0006;    // Enable master mode, normal phase,
                                     // enable talk, and SPI int disabled.
    SpiaRegs.SPIBRR = 0x0003;        // Set baud rate to LSPCLK/(3+1)
    SpiaRegs.SPICCR.all = 0x0097;    // Relinquish SPI from Reset & Loopback
    SpiaRegs.SPIPRI.bit.FREE = 1;    // Set so breakpoints don't disturb xmission
    EDIS;
}
/*****/
//-----
// InitSPIFifo:
//-----
// This function initializes the SPI(A) FIFO.
void FlaxInitSpiFifo(void)
{
    EALLOW;
    SpiaRegs.SPIFFTX.all=0xE040;
    SpiaRegs.SPIFFRX.all=0x204f;
    SpiaRegs.SPIFFCT.all=0x0;
    EDIS;
}
/*****/
//-----
// InitSPIFifo:
//-----
// This function transmit word by SPI(A) FIFO.
void FlaxSpiTx(Uint16 a)
{
    // Send a word to spi buff
    SpiaRegs.SPITXBUF=a;
}
/*****/
//-----
// Example: InitSpiGpio: Aviv Analytical Board
//-----
// This function initializes GPIO pins to function as SPI pins
//
// Each GPIO pin can be configured as a GPIO pin or up to 3 different
// peripheral functional pins. By default all pins come up as GPIO
// inputs after reset.
//
// Caution:
// For each SPI peripheral
// Only one GPIO pin should be enabled for SPISOMO operation.
// Only one GPIO pin should be enabled for SPISOMI operation.
// Only one GPIO pin should be enabled for SPICLKA operation.
// Only one GPIO pin should be enabled for SPISTEA operation.
// Comment out other unwanted lines.
void FlaxInitSpiGpio(void)
{
    EALLOW;

```

```

/* Enable internal pull-up for the selected pins */
// Pull-ups can be enabled or disabled by the user.
// This will enable the pullups for the specified pins.
// Comment out other unwanted lines.

    GpioCtrlRegs.GPAPUD.bit.GPIO16 = 0;    // Enable pull-up on GPIO16 (SPISIMOA)
    GpioCtrlRegs.GPAPUD.bit.GPIO17 = 0;    // Enable pull-up on GPIO17 (SPISOMIA)
    GpioCtrlRegs.GPAPUD.bit.GPIO18 = 0;    // Enable pull-up on GPIO18 (SPICLKA)
    GpioCtrlRegs.GPAPUD.bit.GPIO19 = 0;    // Enable pull-up on GPIO19 (SPISTEA)

/* Set qualification for selected pins to asynch only */
// This will select asynch (no qualification) for the selected pins.
// Comment out other unwanted lines.

    GpioCtrlRegs.GPAQSEL2.bit.GPIO16 = 3; // Asynch input GPIO16 (SPISIMOA)
    GpioCtrlRegs.GPAQSEL2.bit.GPIO17 = 3; // Asynch input GPIO17 (SPISOMIA)
    GpioCtrlRegs.GPAQSEL2.bit.GPIO18 = 3; // Asynch input GPIO18 (SPICLKA)
    GpioCtrlRegs.GPAQSEL2.bit.GPIO19 = 3; // Asynch input GPIO19 (SPISTEA)

/* Configure SPI-A pins using GPIO regs*/
// This specifies which of the possible GPIO pins will be SPI functional pins.
// Comment out other unwanted lines.

    GpioCtrlRegs.GPAMUX2.bit.GPIO16 = 1; // Configure GPIO16 as SPISIMOA
    GpioCtrlRegs.GPAMUX2.bit.GPIO17 = 1; // Configure GPIO17 as SPISOMIA
    GpioCtrlRegs.GPAMUX2.bit.GPIO18 = 1; // Configure GPIO18 as SPICLKA
    //GpioCtrlRegs.GPAMUX2.bit.GPIO19 = 1; // Configure GPIO19 as SPISTEA

    EDIS;
}
/*****
// This function initializes the SPI.
void FlaxInitSpi(void)
{
    FlaxInitSpiParam();
    FlaxInitSpiFifo();
    FlaxInitSpiGpio();
}
/*****

//=====
// End of file.
//=====

```