```
/*******************************************************************************
 Module
   SPIFollower.c

 Revision
   1.0.1

 Description
   This is a template file for implementing a simple service under the
   Gen2 Events and Services Framework.

 Notes

 History
 When           Who     What/Why
 -------------- ---     --------
 01/16/12 09:58 jec      began conversion from TemplateFSM.c
*******************************************************************************/
/*---------------------------- Include Files ----------------------------*/
/* include header files for this state machine as well as any machines at the
   next lower level in the hierarchy that are sub-machines to this machine
*/
#include "ES_Configure.h"
#include "ES_Framework.h"
#include "SPIFollower.h"
#include "PIC32_SPI_HAL.h"
#include "dbprintf.h"
#include "ES_CheckEvents.h"
#include "ES_DeferRecall.h"
#include "PIC32PortHAL.h"
#include <proc/p32mx170f256b.h>
#include <pic32m_builtins.h>
#include <sys/attribs.h>

/*---------------------------- Module Defines ----------------------------*/
//default values
#define NO_BEACON_COMMAND 0x1E
#define DISTANCE_FAR_COMMAND 0x22


#define DUMMY_VALUE 0xAA

//queries that can be received
#define BEACON_QUERY 0x10
#define DISTANCE_QUERY 0x20
//game start and end commands
#define GAME_START_COMMAND 0x40
#define GAME_END_COMMAND 0x41
//indicator light commands
#define LIGHT1_COMMAND_OFF 0x80
#define LIGHT2_COMMAND_OFF 0x90
#define LIGHT1_COMMAND_ON 0x81
#define LIGHT2_COMMAND_ON 0x91
//light pins
#define LIGHT1 LATBbits.LATB4
#define LIGHT2 LATBbits.LATB5
//servo angles
#define THETA_0 2700
#define THETA_180 20000
```

```
#define T3_PERIOD 0xFFFF


/*---------------------------- Module Functions ----------------------------*/
/* prototypes for private functions for this service.They should be functions
   relevant to the behavior of this service
*/
static void configSPI1Module(void);
static void configSPIInterrupts(void);
static void configOC4();

/*---------------------------- Module Variables ----------------------------*/
// with the introduction of Gen2, we need a module level Priority variable
static uint8_t MyPriority;

static uint8_t whichBeacon;
static uint8_t currentDistance;

static Response_t nextDataToSend;



/*---------------------------- Module Code ----------------------------*/
/****************************************************************************
 Function
     InitSPIFollower

 Parameters
     uint8_t : the priorty of this service

 Returns
     bool, false if error in initialization, true otherwise

 Description
     Saves away the priority, and does any
     other required initialization for this service
 Notes

 Author
     J. Edward Carryer, 01/16/12, 10:00
****************************************************************************/
bool InitSPIFollower(uint8_t Priority)
{
  //set default values for distances, beacons, and data to send


  //CONFIG SPI WITH INTERRUPTS
       //setup SPI
       //setup SPI interrupts
       //setup two light pins
       //set both lights initially off
 //setup OC4 for servo

 }


/****************************************************************************
 Function
    RunSPIFollower
```

```
 Parameters
   ES_Event_t : the event to process

 Returns
   ES_Event, ES_NO_EVENT if no error ES_ERROR otherwise

 Description
   add your description here
 Notes

 Author
   J. Edward Carryer, 01/15/12, 15:23
*****************************************************************************/
ES_Event_t RunSPIFollower(ES_Event_t ThisEvent)
{
      //if new beacon event
            //update beacon variable with event param
      //if new distance event
            //update distance variable with event param
}




/****************************************************************************
 private functions
 ****************************************************************************/
static void configSPI1Module(void){
  //------INITIALIZE SPI1-----
      //disable SPI

      //use PBCLK
  //clear the receive buffer
  //Follower mode, disable the analog function of B14, set B14 to an input // B14 = SCK
  //disable analog function of A0, set RA0 to an input, RA0 to the SS input, slave
select enabled
  //disable the analog function of A1, set A1 to an output, RA1 to the SDO1
  //disable the analog function of B11, set B11 to an input, RB11 to the SDI1
  //8 bit transfer
   //CKE to 1
  //CKP to 1
  //Enhanced buffer off
  //bit rate to 10kHz
  //clear the SPIROV bit
  //preload SPI1BUF
  //enable SPI
}

static void configSPIInterrupts(void){
      //ensure global interrupts disabled

      //enable multivector mode
     //set SPI1 priority
    //Clear any pending interrupt flags
    //enable receive and transmit interrupt
      //enable global interrupts
}

static void configOC4(void){
      //disable OC 4
```

```
	//select timer 3 as the source
	//cont operation in idle mode
	// set OC4 mode to PWM mode
	 //32 bit compare mode off

	//set initial duty cycle to zero position

	//map OC4 to B13
	//enable OC4
}

void __ISR(_SPI_1_VECTOR,IPL5SOFT) SPI1_data(void){
	//if receive flag is set
		//read received data

		//if received data has changed
			//if beacon query
				//return beacon next time
			//if distance query
				//return distance next time
			//if light 1 off command
				//turn light 1 off
			//if light 2 off command
				//turn light 2 off
			//if light 1 on command
				//turn light 1 on
			//if light 2 on command
				//turn light 2 on
			//if game start command
				//turn servo to on position
			//if game end command
				//turn servo to off position
			//otherwise
				//unrecognized command so send nothing
		//save last data received
		//continue until SPI buffer is empty

	//if transmit flag is set
		//if data requested is beacon
			//write beacon data to SPI buffer
		//if data requested is distance
			//write distance data to SPI buffer
		//data requested not valid/nothing
			//send dummy value

    //clear interrupt flags

}

/*------------------------------ Footnotes -------------------------------*/
/*------------------------------ End of file -------------------------------*/
```