```
/*****************************************************************************
 Module
   E128_SPI.c

 Revision
   1.0.0

 Description
   This module initializes the SPI port on the E128 and houses all functions
   necessary to recieve information from the forward artillery controller.

 Notes


 History
 When           Who     What/Why
 -------------- ---     --------
 02/01/2013     BMJ     Started Coding for ComGen for Lab 8
 02/21/2013     DYL     Edited SPI Hardware set up for FAC for Battleship project
*****************************************************************************/
/*----------------------------- 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 <stdio.h>
#include <stdlib.h>
#include <mc9s12e128.h>
#include <S12e128bits.h>
#include <Bin_Const.h>
#include <termio.h>
#include <hidef.h>
#include "S12eVec.h"

#include "E128_PWM.h"         //has all prescale definitions
#include "E128_SPI.h"
#include "E128_Servo.h"
#include "FAC_FSM.h"
#include "NavigationFSM.h"
#include "AlignPPService.h"
#include "DriveTrainService.h"
#include "ArtilleryFSM.h"


/*------------------------------ Module Code ------------------------------*/

void InitSPIHardware( void )
{
    //Initialize the Baud Rate to 6MHz
    //Baud Rate Divisor = (SPPR+1)*2^(SPR+1)
    //SPPR = 6, SPR = 6
    SPIBR |= _S12_SPPR0;
    SPIBR |= _S12_SPPR1;
    SPIBR |= _S12_SPPR2;

    SPIBR |= _S12_SPR0;
    SPIBR |= _S12_SPR1;
    SPIBR |= _S12_SPR2;

    /*************************************************************************
    Initialize the SPI Control Register to
    MODE 3 Communication - CPOL & CPHA = 1
    E128 as Master - MSTR = 1
    MSB First - LSBFE = 0
    DO NOT ENABLE Slave Select Output Enable
    DO NOT ENABLE SPI Interrupt Enable (incoming message) - SPIE = 1
    DO NOT ENABLE SPI Transmit Interrupt Enable (outgoing buffer) - SPITIE = 0
    Enable SPI - SPE = 1
```

```
    ************************************************************************/

    SPICR1 |= _S12_CPOL; //MODE 3 Communication - CPOL & CPHA = 1
    SPICR1 |= _S12_CPHA; //MODE 3 Communication - CPOL & CPHA = 1
    SPICR1 |= _S12_MSTR; //128 as Master - MSTR = 1
    SPISR; // Read SPISR Bit to clear interrupt flag directly after initialization
    SPICR1 |= _S12_SPE;  //Enable SPI - SPE = 1

    printf("SPI Initialized for comms with the Forward Artillery Controller\r\n");
}

/************************************************************************
 Function
    WriteMessage

 Parameters
   OutgoingMessage

 Returns
   Nothing

 Description
   This function writes the input OutgoingMessage into the transmit register of the SPI.

 Notes

 Author
   Debbie Li 2/21/2013
************************************************************************/

void WriteMessage ( char OutgoingMessage )
{
    SPISR; // clear SPTEF flag

    SPIDR = OutgoingMessage;
}


/************************************************************************
 Function
    ReadMessage

 Parameters
   None

 Returns
   character with CurrentMessage

 Description
   This function returns a 1-byte message from the FAC.

 Notes

 Author
   Debbie Li 2/21/2013
************************************************************************/

char ReadMessage ( void )
{
    static char CurrentMessage;

    SPISR; //clear SPIF flag

    CurrentMessage = SPIDR;
    return CurrentMessage;
}
```