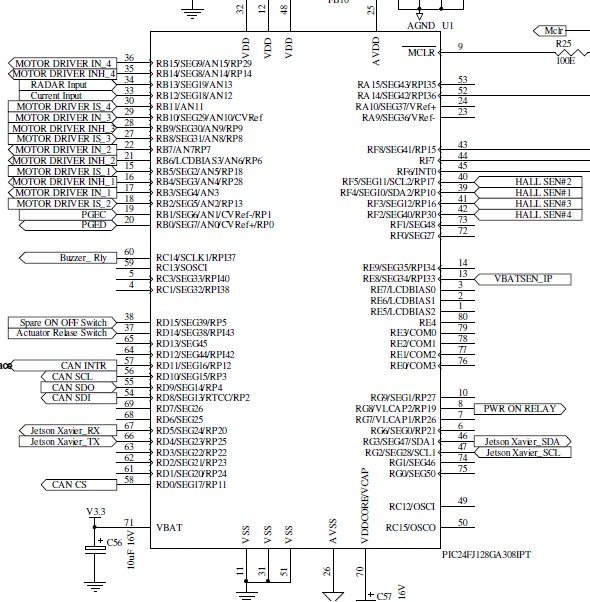
**Autonomous Tractor Safety System**

**Objective 8.0** To demonstrate SPI in ATSS PCB board.

****

**TRISC** DATA DIRECTION REGISTER C: 0x0000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TRISC15 | TRISC14 | TRISC13 | TRISC12 | TRISC11 | TRISC10 | TRISC9 | TRISC8 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |

**LATC OUTPUT** LATCH REGISTER C: 0x0000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LATC15 | LATC14 | LATC13 | LATC12 | LATC11 | LATC10 | LATC9 | LATC8 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |

**SP1CON1** SPI1 CONTROL REGISTER 1:0x0120

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | - | DISSCK | DISSDO | MODE16 | SMP | CKE |
|  |  |  | DISABLE SPI SCK1 (MASTER) | DISABLE SDO1 | WORD/BYTE | SPI1 DATA I/P SAMPLE (MASTER) | CLK EDGE SELECT |
|  |  |  | 0 | 0 | 0 | 0 | 1 |
|  |  |  | INT SPI CLK ENABLED | SDO1 CONTROLLED BY MODULE | BYTE | SAMPLED AT MIDDLE OF BIT | DATA CHG WHEN CLK ACTIVE TO IDLE |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SSEN | CKP | MSTEN | SPRE2 | SPRE1 | SPRE0 | PPRE1 | PPRE0 |
| SLAVE SEL EN | CLK POLARITY | MASTER EN | SECONDARY PRESCALE | | | PRIMARY PRESCALE | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| SSx UNUSED | LO IS IDLE, HI IS ACTIVE | MASTER | 8:1 | | | 64:1 | |

**SP1CON2** SPI1 CONTROL REGISTER 2:0x0000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| FRMEN | SPIFSD | SPIFPOL | - | - | - | - | - |
| FRAMED SPI1 | SYNC DIR ON SSX PIN | SYNC PULSE POLARITY |  |  |  |  |  |
| 0 | 0 | 0 |  |  |  |  |  |
| DISABLED | OUTPUT(MASTER) | ACT LOW |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | - | - | - | - | SPIFE | SPIBEN |
|  |  |  |  |  |  | SYNC EDGE SELECT | ENHANCED BUFFER EN |
|  |  |  |  |  |  | 0 | 0 |
|  |  |  |  |  |  | FRAME PRECEDED 1ST CLK BIT | DISABLED |

**SP1STAT** SPI1 STATUS AND CONTROL REGISTER:0x8000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SPIEN | - | SPISIDL | - | - | SPIBEC2 | SPIBEC1 | SPIBEC0 |
| SPI EN |  | SPI1 IN IDLE MODE |  |  | SPI1 BUFFER COUNT | | |
| 1 |  | 0 |  |  |  |  |  |
| ENABLED |  | CONTINUE |  |  | NO. SPI TR PENDING(M)/ COUNTED(S) | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SRMPT | SPIROV | SRXMPT | SISEL2 | SISEL1 | SISEL0 | SPITBF | SPIRBF |
| SR EMPTY | RX OVERFLOW | RX FIFO EMPTY | SPI1 BUFFER INTERRUPT MODE | | | TX BUFFER FULL | RX BUFFER FULL |
| 0 | 0 | 0 |  |  |  | 0 | 0 |
| NOT EMPTY | NO OVERFLOW | RX FIFO NOT EMPTY | INT WHEN LAST DATA IN RX BUFFER IS READ, BUFFER EMPTY | | | TX NOT STARTED, BUF EMPTY | RX INCOMPLETE, BUFFER EMPTY |

**Code Prg08.c**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Source code file: Prg08.c

Author, version, date: PSS ver 1.0 01.11.24

Program function: SPI

Simulation: PIC24FJ128GA308 MCU, MPLAB X IDE ver 6.05, XC16 ver 2.10

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

Read the 25LC256 EEPROM status register using SPI1

\*/

#include <xc.h>

#include <libpic30.h>

#define SEE\_WRSR 1 // Write status register, 25LC256 Serial EEPROM commands

#define SEE\_WRITE 2 // Write command

#define SEE\_READ 3 // Read command

#define SEE\_WDI 4 // Write disable

#define SEE\_STAT 5 // Read status register

#define SEE\_WEN 6 // Write enable

uint8\_t i**;**

void initializeSpi1**(**void**);**

int WriteSpi1**(** int data**);**

main**()**

**{**

initializeSpi1**();**

**while(**1**)**

**{**

LATCbits**.**LATC1 **=** 0**;** // Enable the EEROM, Send a Write Enable command

WriteSpi1**(**SEE\_WEN**);** // Write enable command

LATCbits**.**LATC1 **=** 1**;** // Disable the EEROM

LATCbits**.**LATC1 **=** 0**;** // Enable the EEROM, Check the Serial EEPROM status

WriteSpi1**(**SEE\_STAT**);** // Send a READ STATUS command

i **=** WriteSpi1**(**0**);** // Send/receive the data by sending a dummy command

LATCbits**.**LATC1 **=** 1**;** // Disable the EEROM

**}**

**}**

void initializeSpi1**(**void**)**

**{**

SPI1CON1 **=** 0x0120**;** // Select mode MODE16(0)-8 bit data, CKE(1)-serial

// outut changes when clock changes from active to idle, CKP(0)-0 is low 1 is high,

// MEN(1)-Master enable, SPRE(00)-8:1 prescale, PPRE(00)-64:1

SPI1STAT **=** 0x8000**;** // Enable the peripheral SPIEN(1)

TRISC **=** 0x0000**;**// RC1 is connected to CS (active low) of EEPROM and made an output

LATCbits**.**LATC1 **=** 1**;** // Disable the EEROM initially

**}**

int WriteSpi1**(** int data**)** // Send one byte of data and receive one back at the same

**{** // time

SPI1BUF **=** data**;** // Write to buffer for TX

**while(** **!**SPI1STATbits**.**SPIRBF**);** // Wait transfer completion

**return** SPI1BUF**;** // Read the received value

**}**