

```

/*****
/* Program      : 25CxxSPI.C
/* Function     : SPI EEPROM Utility Control Procedures
/* Author       : John F. Fitter B.E.
/* Reference    : XICOR Data Book, July 1995 X25160
/*
/* Rev No.     Rev date   Test date   Test platform   Description
/* -----
/*      00      6jun98
/*
/*      Copyright © 1998 Eagle Air Australia Pty. Ltd. All rights reserved
*****/

#define _25CXXSPI_C

#include <commdefs.h>
#include "main.h"
#include "25cxxspi.h"
#include "lcd44780.h"
#include "spi.h"
#include "delays.h"

// Procedure to return data at eeprom address. eeprom is left selected for fast reads.
unsigned char read_eeprom(unsigned int addr) {

    select_eeprom();
    spi_write(E2READ);
    spi_write(HIBYTE(addr));
    spi_write(addr);
    return spi_read(0);
}

// Procedure to return data at eeprom address and deselect eeprom when finished.
unsigned char read_eeprom_deselect(unsigned int addr) {
    unsigned char n;
    /*
    n = read_eeprom(addr);
    deselect_eeprom();
    return n;
    */
    select_eeprom();
    spi_write(E2READ);
    spi_write(HIBYTE(addr));
    spi_write(addr);
    n = spi_read(0);
    deselect_eeprom();
    return n;
}

// Procedure to write data to eeprom at addr
void write_eeprom(unsigned int addr, unsigned char data) {

    wren_eeprom();
    select_eeprom();
    spi_write(E2WRITE);
    spi_write(HIBYTE(addr));
    spi_write(addr);
    spi_write(data);
    do {
        toggle_eeprom();
        spi_write(E2RDSR);
    } while(!!(spi_read(0) & WIP));
    deselect_eeprom();
}

// Procedure to initialize the eeprom for reading and writing.
void init_eeprom() {

    e2_cs_dir = B_OUT;
    deselect_eeprom();
    select_eeprom();
    spi_write(E2RDSR);
    if(!!(spi_read(0) & (PROTECT_ALL))) {

```

```
wren_eeeprom(); // set the write enable latch
select_eeeprom();
spi_write(E2WRSR); // send status reg. write instruction
spi_write(PROTECT_NONE); // clear bank protect bits and WPEN
} // (WEL and WIP are read only
deselect_eeeprom(); // so this is ok
delay_ms(5);
}

// Procedure to set the eeeprom write enable latch.

void wren_eeeprom() {
    toggle_eeeprom();
    spi_write(E2WREN); // send write enable instruction
    deselect_eeeprom(); // need at least 2us wide toggle
} // pulse Tcsd

// Procedure to print a string of chars from eeeprom to the lcd. The string start is at addr.
// Printing terminates when a terminating null is encountered.

void print_eeestring(unsigned int addr) {
    unsigned char n;

    n = read_eeeprom(addr);
    while(!!n) {
        putchar(n);
        n = spi_read(0);
    }
    deselect_eeeprom();
}

// Procedure to write a string of characters to the eeeprom starting at addr. str is a pointer
// to the null terminated string to be saved in eeeprom.

void write_eeestring(unsigned int addr, const unsigned char *str) {
    while(*str) write_eeeprom(addr++, *(str++));
    write_eeeprom(addr, 0);
}

// Procedure to retrieve a floating point double value from the eeeprom and return the value.
// address is the location of the double.

double read_double_eeeprom(unsigned int address) {
    double dbl;
    *(unsigned char*)&dbl = read_eeeprom(address);
    *((unsigned char*)&dbl+1) = spi_read(0);
    *((unsigned char*)&dbl+2) = spi_read(0);
    *((unsigned char*)&dbl+3) = spi_read(0);
    deselect_eeeprom();
    return dbl;
}

// ***** EOF 25CXXSPI.C *****
```