Part 1

char flag;

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000011); //Read command

SPIReadWrite(0x20); //16 bit address (0x2000)

SPIReadWrite(0x00);

flag = SPIReadWrite(0); //Read value - data sent is dummy data

LATAbits.LATA3 = 1; //disable CS

sprintf(line2str, "Read %d %c", rx, flag);

if (count % 2 == 0) {

flag = 'E';

} else {

flag = 'O';

}

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000110); //WREN command

LATAbits.LATA3 = 1; //disable CS

Nop();

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000010); //Write command

SPIReadWrite(0x20); //16 bit address (0x2000)

SPIReadWrite(0x00);

SPIReadWrite(flag); //Write value

LATAbits.LATA3 = 1; //disable CS

\_\_delay\_ms(6);

Part 2

//Read address 0 of EEPROM

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000011); //Read command

SPIReadWrite(0); //16 bit address (0x0000)

SPIReadWrite(0);

rx = SPIReadWrite(0); //Read value - data sent is dummy data

rx <<= 8;

rx += SPIReadWrite(0); //Read value - data sent is dummy data

LATAbits.LATA3 = 1; //disable CS

//Write address 0 of EEPROM

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000110); //WREN command

LATAbits.LATA3 = 1; //disable CS

Nop();

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000010); //Write command

SPIReadWrite(0); //16 bit address (0x0000)

SPIReadWrite(0);

SPIReadWrite(count >> 8); //Write value

SPIReadWrite(count);

LATAbits.LATA3 = 1; //disable CS

\_\_delay\_ms(6);

Part 3

char str[] = "I am Groot";

char rdStr[16] = "Invalid";

//Read address 0 of EEPROM

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000011); //Read command

SPIReadWrite(0x03); //16 bit address (0x0300)

SPIReadWrite(0x00);

for (int i = 0; i < 11; i++) {

rdStr[i] = SPIReadWrite(0);

}

LATAbits.LATA3 = 1; //disable CS

sprintf(line2str, "Read %s", rdStr);

LCDClearLine(1);

LCDWriteLine(line2str, 1);

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000110); //WREN command

LATAbits.LATA3 = 1; //disable CS

Nop();

LATAbits.LATA3 = 0; //enable CS

SPIReadWrite(0b00000010); //Write command

SPIReadWrite(0x03); //16 bit address (0x0300)

SPIReadWrite(0x00);

for (int i = 0; i < 11; i++) {

SPIReadWrite(str[i]);

}

LATAbits.LATA3 = 1; //disable CS

\_\_delay\_ms(6);

sprintf(line2str, "Wrote %s", str);

LCDClearLine(1);

LCDWriteLine(line2str, 1);

INTCONbits.INT0IF = 0; //must clear the flag to avoid recursive interrupts