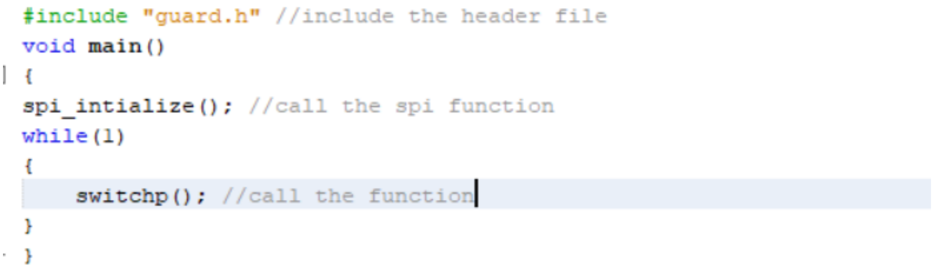
SPI

Task program

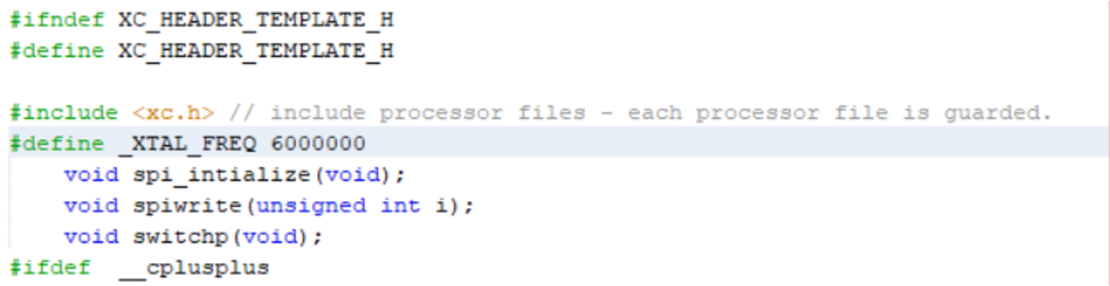
MASTER TX :-

MAIN PROGRAM:-



```
#include "guard.h" //include the header file
void main()
{
spi_intialize(); //call the spi function
while(1)
{
    switchp(); //call the function
}
}
```

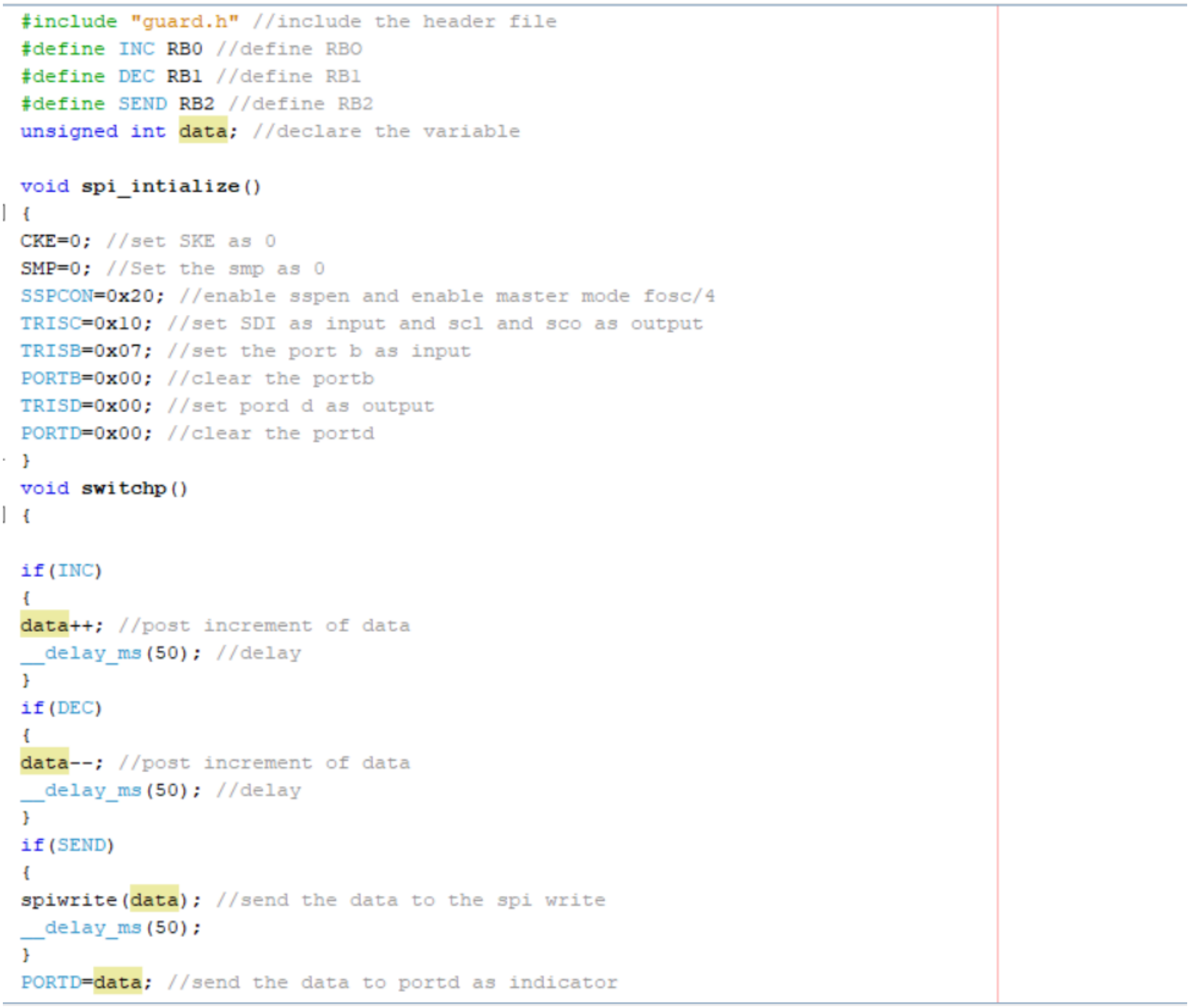
GUARD PROGRAM:-



```
#ifndef XC_HEADER_TEMPLATE_H
#define XC_HEADER_TEMPLATE_H

#include <xc.h> // include processor files - each processor file is guarded.
#define _XTAL_FREQ 6000000
    void spi_intialize(void);
    void spiwrite(unsigned int i);
    void switchp(void);
#ifdef  __cplusplus
```

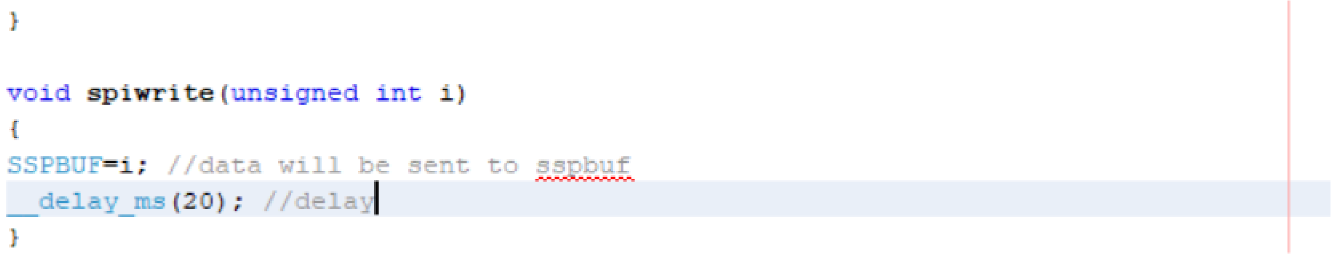
FUNCTION PROGRAM:



```
#include "guard.h" //include the header file
#define INC RB0 //define RB0
#define DEC RB1 //define RB1
#define SEND RB2 //define RB2
unsigned int data; //declare the variable

void spi_intialize()
{
CKE=0; //set SKE as 0
SMP=0; //Set the smp as 0
SSPCON=0x20; //enable sspen and enable master mode fosc/4
TRISC=0x10; //set SDI as input and scl and sco as output
TRISB=0x07; //set the port b as input
PORTB=0x00; //clear the portb
TRISD=0x00; //set pord d as output
PORTD=0x00; //clear the portd
}
void switchp()
{

if(INC)
{
data++; //post increment of data
__delay_ms(50); //delay
}
if(DEC)
{
data--; //post increment of data
__delay_ms(50); //delay
}
if(SEND)
{
spiwrite(data); //send the data to the spi write
__delay_ms(50);
}
PORTD=data; //send the data to portd as indicator
```

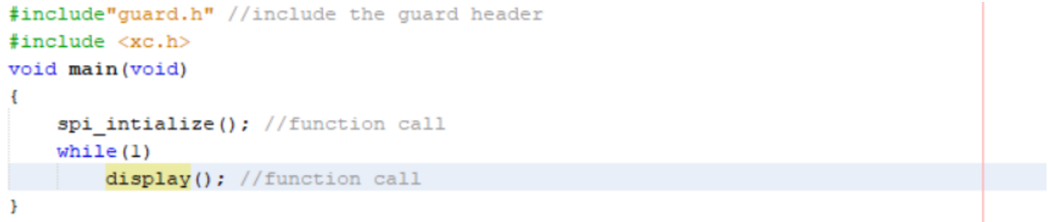


```
}

void spiwrite(unsigned int i)
{
SSPBUF=i; //data will be sent to sspbuf
__delay_ms(20); //delay
}
```

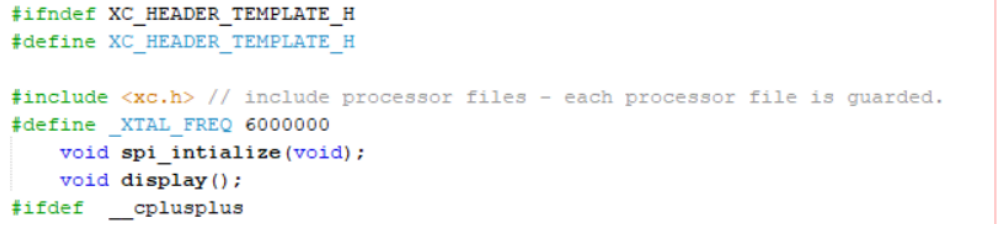
SLAVE RX :-

MAIN PROGRAM:-



```
#include"guard.h" //include the guard header
#include <xc.h>
void main(void)
{
    spi_intialize(); //function call
    while(1)
        display(); //function call
}
```

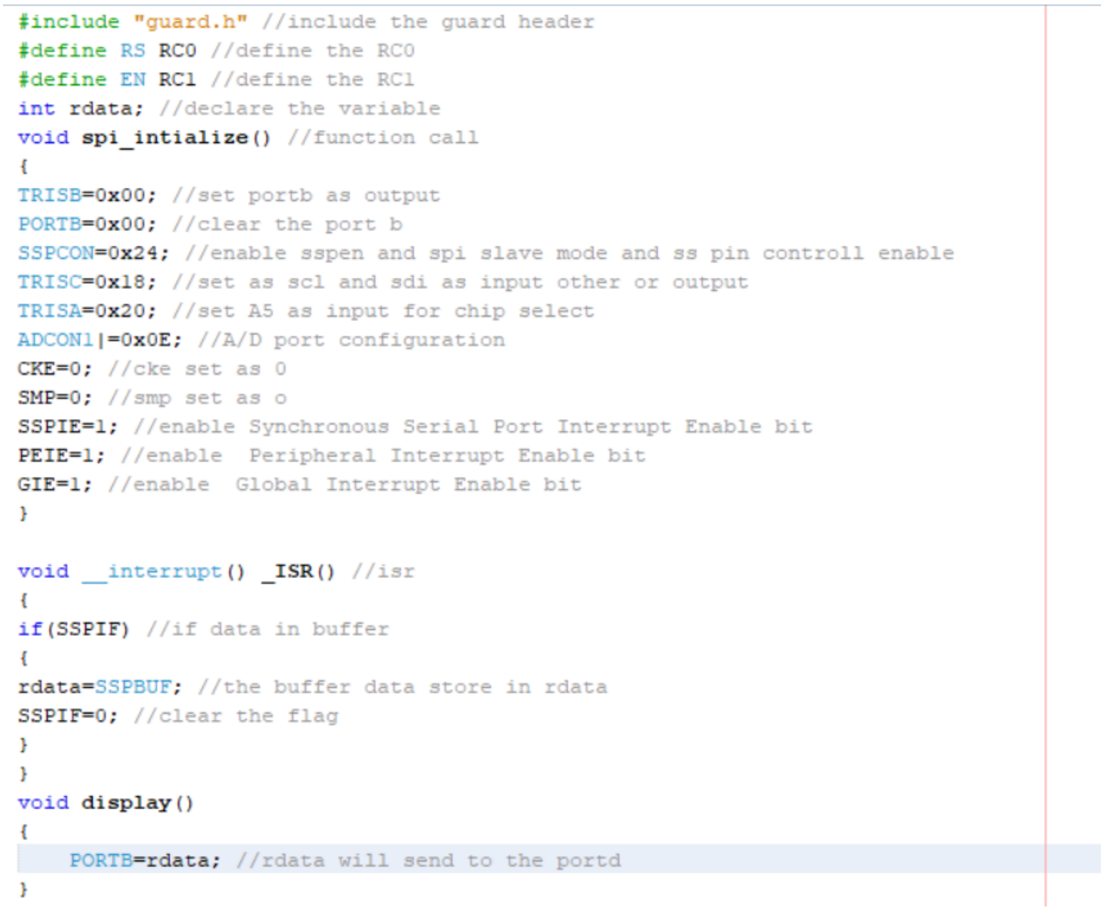
GUARD PROGRAM:-



```
#ifndef XC_HEADER_TEMPLATE_H
#define XC_HEADER_TEMPLATE_H

#include <xc.h> // include processor files - each processor file is guarded.
#define _XTAL_FREQ 6000000
    void spi_intialize(void);
    void display();
#ifdef  __cplusplus
```

FUNCTION PROGRAM:



```
#include "guard.h" //include the guard header
#define RS RC0 //define the RC0
#define EN RC1 //define the RC1
int rdata; //declare the variable
void spi_intialize() //function call
{
TRISB=0x00; //set portb as output
PORTB=0x00; //clear the port b
SSPCON=0x24; //enable sspen and spi slave mode and ss pin controll enable
TRISC=0x18; //set as scl and sdi as input other or output
TRISA=0x20; //set A5 as input for chip select
ADCON1|=0x0E; //A/D port configuration
CKE=0; //cke set as 0
SMP=0; //smp set as o
SSPIE=1; //enable Synchronous Serial Port Interrupt Enable bit
PEIE=1; //enable  Peripheral Interrupt Enable bit
GIE=1; //enable  Global Interrupt Enable bit
}

void __interrupt() _ISR() //isr
{
if(SSPIF) //if data in buffer
{
rdata=SSPBUF; //the buffer data store in rdata
SSPIF=0; //clear the flag
}
}
void display()
{
    PORTB=rdata; //rdata will send to the portd
}
```