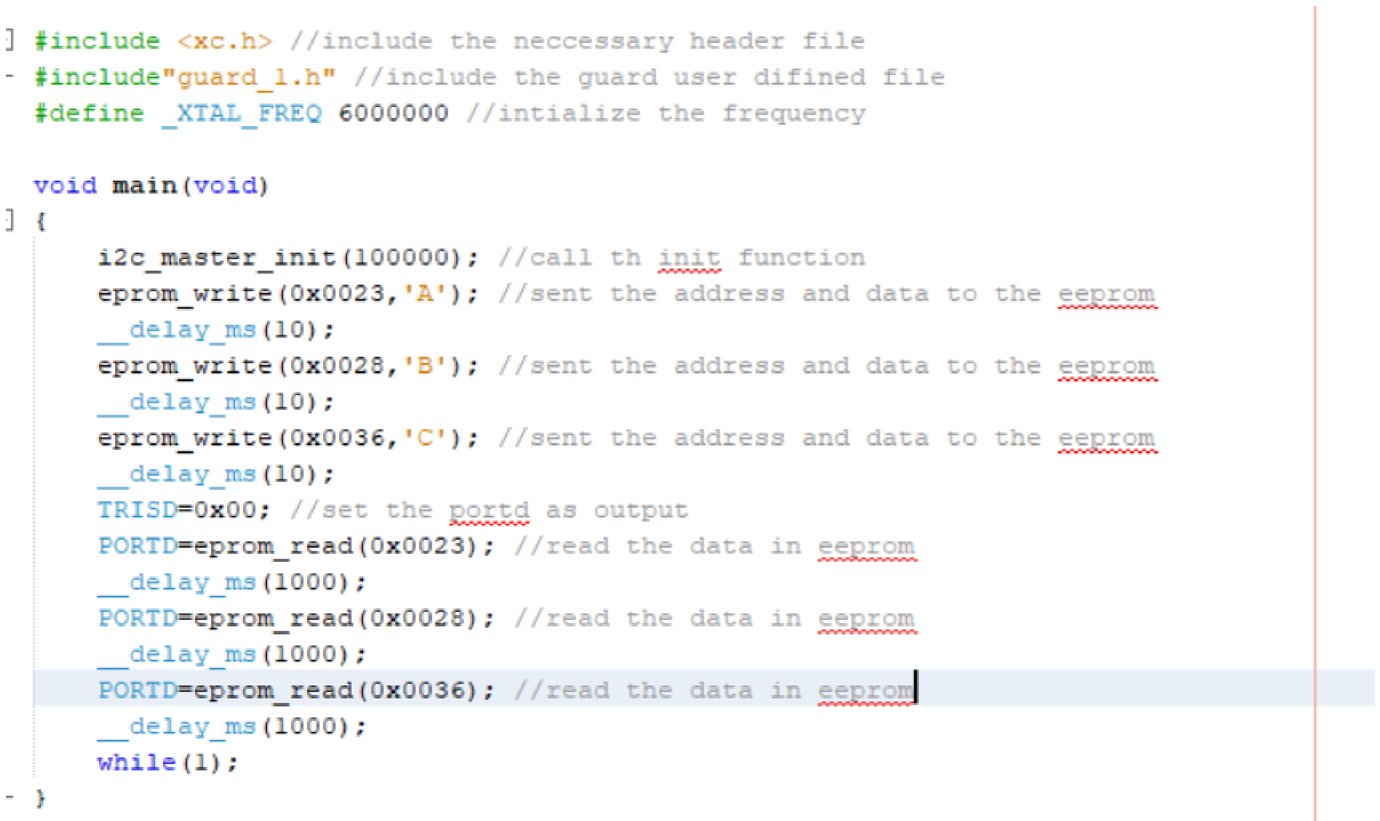
I2C

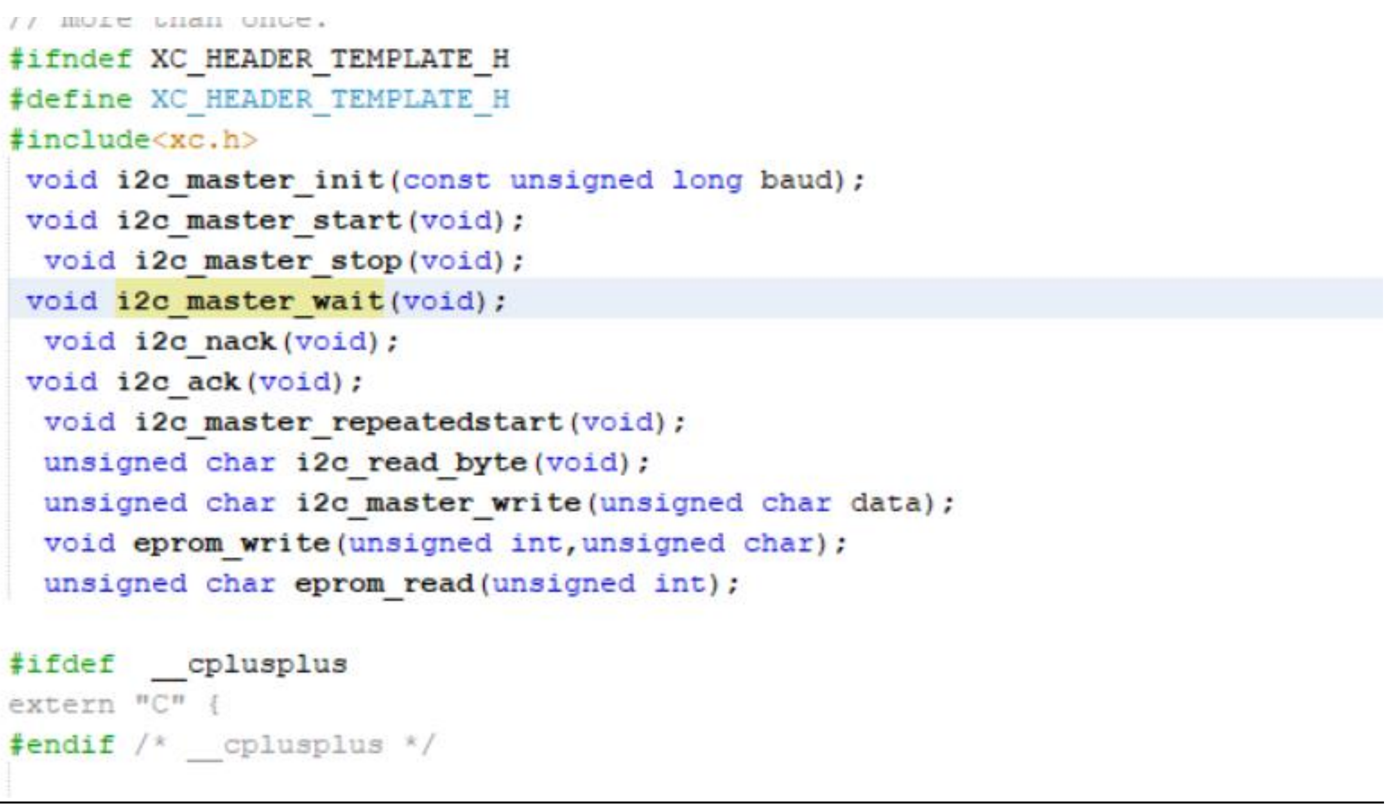
Task program

MAIN PROGRAM:-



```c
#include <xc.h> //include the neccessary header file
#include"guard_1.h" //include the guard user difined file
#define _XTAL_FREQ 6000000 //intialize the frequency

void main(void)
{
    i2c_master_init(100000); //call th init function
    eprom_write(0x0023,'A'); //sent the address and data to the eeprom
    __delay_ms(10);
    eprom_write(0x0028,'B'); //sent the address and data to the eeprom
    __delay_ms(10);
    eprom_write(0x0036,'C'); //sent the address and data to the eeprom
    __delay_ms(10);
    TRISD=0x00; //set the portd as output
    PORTD=eprom_read(0x0023); //read the data in eeprom
    __delay_ms(1000);
    PORTD=eprom_read(0x0028); //read the data in eeprom
    __delay_ms(1000);
    PORTD=eprom_read(0x0036); //read the data in eeprom
    __delay_ms(1000);
    while(1);
}
```

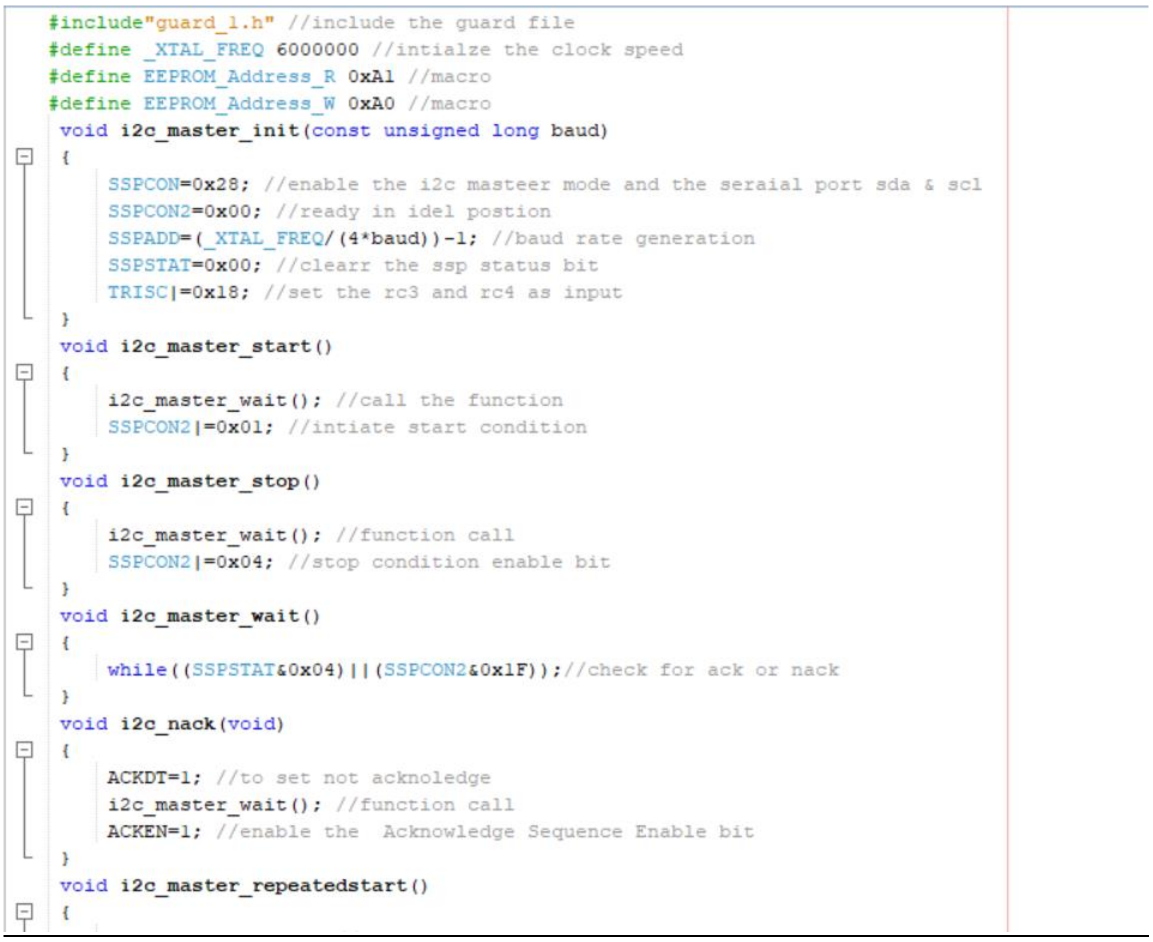
GUARD PROGRAM:-



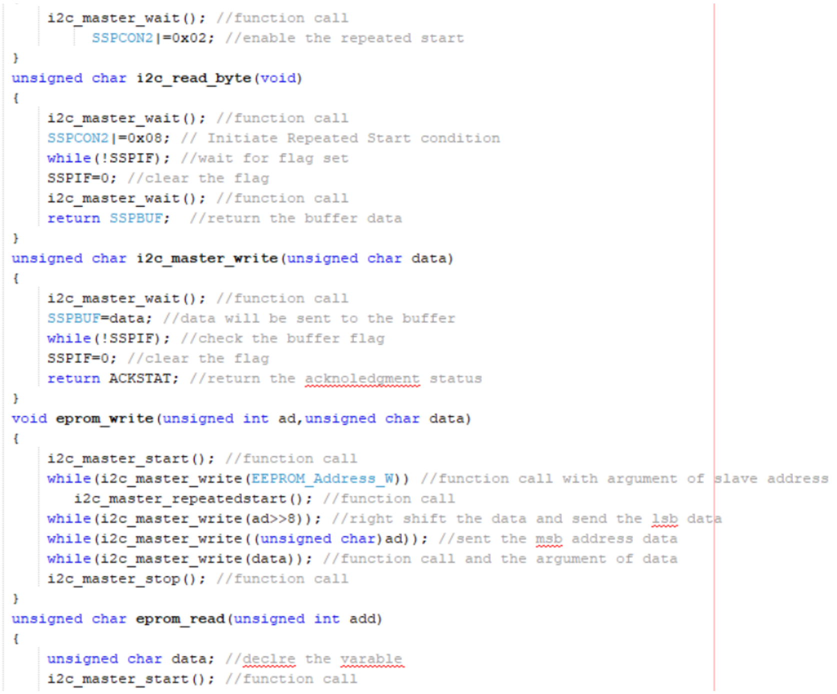
```c
// more than once.
#ifndef XC_HEADER_TEMPLATE_H
#define XC_HEADER_TEMPLATE_H
#include<xc.h>
 void i2c_master_init(const unsigned long baud);
 void i2c_master_start(void);
  void i2c_master_stop(void);
 void i2c_master_wait(void);
  void i2c_nack(void);
 void i2c_ack(void);
  void i2c_master_repeatedstart(void);
  unsigned char i2c_read_byte(void);
  unsigned char i2c_master_write(unsigned char data);
  void eprom_write(unsigned int,unsigned char);
  unsigned char eprom_read(unsigned int);

#ifdef  __cplusplus
extern "C" {
#endif /* __cplusplus */
```

Task program

FUNCTION PROGRAM:



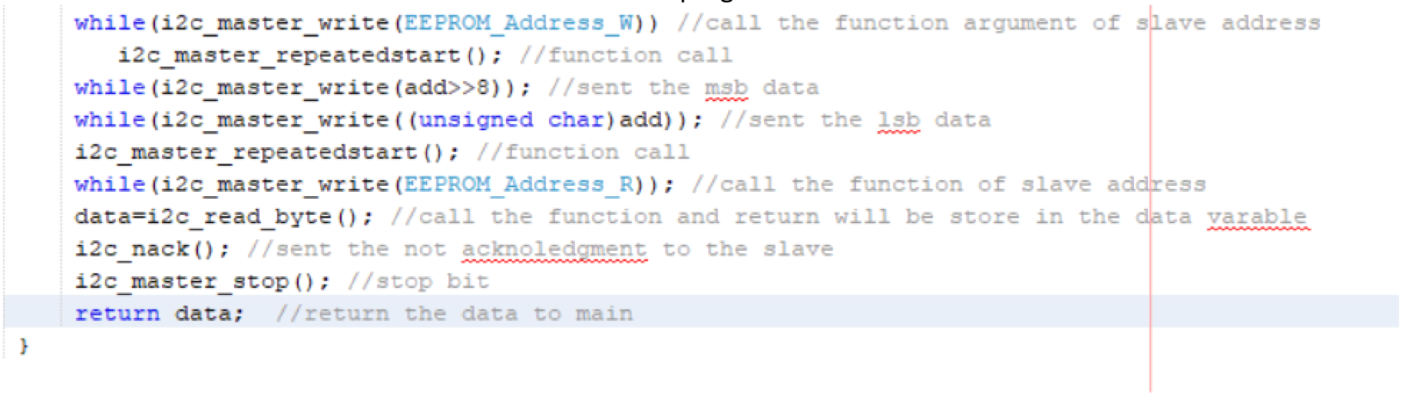
```c
#include"guard_1.h" //include the guard file
#define _XTAL_FREQ 6000000 //intialze the clock speed
#define EEPROM_Address_R 0xA1 //macro
#define EEPROM_Address_W 0xA0 //macro
 void i2c_master_init(const unsigned long baud)
 {
     SSPCON=0x28; //enable the i2c masteer mode and the seraial port sda & scl
     SSPCON2=0x00; //ready in idel postion
     SSPADD=(_XTAL_FREQ/(4*baud))-1; //baud rate generation
     SSPSTAT=0x00; //clearr the ssp status bit
     TRISC|=0x18; //set the rc3 and rc4 as input
 }
 void i2c_master_start()
 {
     i2c_master_wait(); //call the function
     SSPCON2|=0x01; //intiate start condition
 }
 void i2c_master_stop()
 {
     i2c_master_wait(); //function call
     SSPCON2|=0x04; //stop condition enable bit
 }
 void i2c_master_wait()
 {
     while((SSPSTAT&0x04)||(SSPCON2&0x1F));//check for ack or nack
 }
 void i2c_nack(void)
 {
     ACKDT=1; //to set not acknoledge
     i2c_master_wait(); //function call
     ACKEN=1; //enable the  Acknowledge Sequence Enable bit
 }
 void i2c_master_repeatedstart()
 {
```



```c
    i2c_master_wait(); //function call
       SSPCON2|=0x02; //enable the repeated start
}
unsigned char i2c_read_byte(void)
{
    i2c_master_wait(); //function call
    SSPCON2|=0x08; // Initiate Repeated Start condition
    while(!SSPIF); //wait for flag set
    SSPIF=0; //clear the flag
    i2c_master_wait(); //function call
    return SSPBUF;  //return the buffer data
}
unsigned char i2c_master_write(unsigned char data)
{
    i2c_master_wait(); //function call
    SSPBUF=data; //data will be sent to the buffer
    while(!SSPIF); //check the buffer flag
    SSPIF=0; //clear the flag
    return ACKSTAT; //return the acknoledgment status
}
void eprom_write(unsigned int ad,unsigned char data)
{
    i2c_master_start(); //function call
    while(i2c_master_write(EEPROM_Address_W)) //function call with argument of slave address
       i2c_master_repeatedstart(); //function call
    while(i2c_master_write(ad>>8)); //right shift the data and send the lsb data
    while(i2c_master_write((unsigned char)ad)); //sent the msb address data
    while(i2c_master_write(data)); //function call and the argument of data
    i2c_master_stop(); //function call
}
unsigned char eprom_read(unsigned int add)
{
    unsigned char data; //declre the varable
    i2c_master_start(); //function call
```

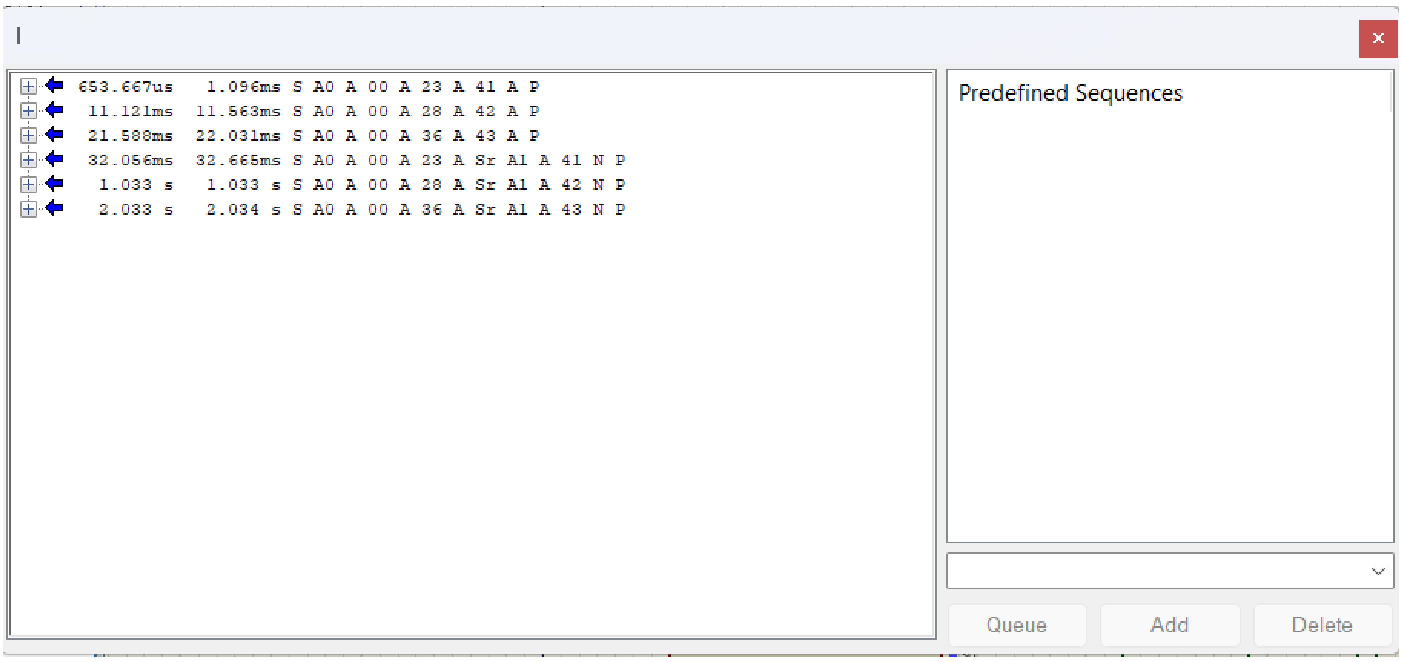
I2C

Task program



```
    while(i2c_master_write(EEPROM_Address_W)) //call the function argument of slave address
       i2c_master_repeatedstart(); //function call
    while(i2c_master_write(add>>8)); //sent the msb data
    while(i2c_master_write((unsigned char)add)); //sent the lsb data
    i2c_master_repeatedstart(); //function call
    while(i2c_master_write(EEPROM_Address_R)); //call the function of slave address
    data=i2c_read_byte(); //call the function and return will be store in the data varable
    i2c_nack(); //sent the not acknoledgment to the slave
    i2c_master_stop(); //stop bit
    return data;  //return the data to main
}
```

I^{2}C debugger:-



```
653.667us   1.096ms S A0 A 00 A 23 A 41 A P
 11.121ms  11.563ms S A0 A 00 A 28 A 42 A P
 21.588ms  22.031ms S A0 A 00 A 36 A 43 A P
 32.056ms  32.665ms S A0 A 00 A 23 A Sr A1 A 41 N P
  1.033 s   1.033 s S A0 A 00 A 28 A Sr A1 A 42 N P
  2.033 s   2.034 s S A0 A 00 A 36 A Sr A1 A 43 N P
```