Task – 12 – I2C Communication

Write operation:

1.Write data 'A' to memory address 0x0023

2.Write data 'B' to memory address 0x0028

3.Write data 'C' to memory address 0x0036

Read Operation:

1.Read data 'A' from memory address 0x0023 and give that data to PORT register

2.Read data 'B' from memory address 0x0028 and give that data to PORT register

3.Read data 'C' from memory address 0x0036 and give that data to PORT register

Main Function:

```
#include <xc.h>
#include "I2C.h"
#define _XTAL_FREQ 16000000

void main() {

    TRISD = 0x00;  //setting PortD as output
    PORTD = 0x00;  // making PORTD as Low

    I2C_Master_Init(100000);            //calling function I2C_Master_Init with 100000 baudrate
    unsigned int Address = 0x0023;     //Initialize variable address as 0x0023
    unsigned char Data = 'A';          //Intialize variable Data as 'A'

    EEPROM_Write(0x0023,Data++);         //Writing in EEPROM the Data 'A'
    EEPROM_Write(0x0028,Data++);         //Writing in EEPROM the Data 'B'
    EEPROM_Write(0x0036,Data);           //Writing in EEPROM the Data 'C'
    __delay_ms(500);                     //delay for 500 ms

    Address = 0x0023;                    //assigning address as 0x0023
    PORTD = 0x00;                        //assigning PORTD as low
    PORTD = EEPROM_Read(0x0023);         //Reading Data from 0x0023 to PORTD
    __delay_ms(1500);                    //delay for 1500 ms
    PORTD = 0x00;                        //assigning PORTD as low
    PORTD = EEPROM_Read(0x0028);         //Reading Data from 0x0023 to PORTD
    __delay_ms(1500);                    //delay for 1500 ms
    PORTD = 0x00;                        //assigning PORTD as low
    PORTD = EEPROM_Read(0x0036);         //Reading Data from 0x0023 to PORTD
    __delay_ms(1500);                    //delay for 1500 ms


    while(1);    //while loop
}
```

I2C Header File:

```
#ifndef I2C_H
#define I2C_H


//declaring the required functions for I2C Operations

void I2C_Master_Init(const unsigned long baud);
void I2C_Master_Wait();
void I2C_Master_Start();
void I2C_Master_RepeatedStart();
void I2C_Master_Stop();
unsigned char I2C_Master_Write(unsigned char data);
unsigned char I2C_Read_Byte(void);
void I2C_ACK(void);
void I2C_NACK(void);
void EEPROM_Write(unsigned int add, unsigned char data);
unsigned char EEPROM_Read(unsigned int add);

#endif
```

I2C Function:

```
#include <xc.h>
#define _XTAL_FREQ 16000000                                 //Initializing frequency of 16000000
#define EEPROM_Address_R 0xA1                               //defining EEPROM_Address_R as 0xA1
#define EEPROM_Address_W 0xA0                               //defining EEPROM_Address_R as 0xA1

void I2C_Master_Init(const unsigned long baud)              //function defintion of I2C_Master_Init
{
    SSPCON = 0x28;              //00101000, Enabling serial port and setting as I2C as Master Device , Enabling SDA and SCL
    SSPCON2 = 0x00;             //initislizing SSPCON2 as 00000000
    SSPADD = (_XTAL_FREQ/(4*baud))-1;   //calculating the frequency and assigning to SSPADD
    SSPSTAT = 0x00;                     //Intializing SSPSTAT as 00000000
    TRISC3 = 1;                         //setting RC3,RC4 as SDA and SCL
    TRISC4 = 1;
}

void I2C_Master_Wait()                      //Master_Wait Function
{
   while((SSPSTAT & 0x04) || (SSPCON2 & 0x1F));    //Checking whether transmission is going on or not, checking all the bits are not setted
}


void I2C_Master_Start(){                    //I2C_Master_Start function definition
    I2C_Master_Wait();                      //calling Wait Function
    SEN = 1;                                //Enabling Start Bit
}

void I2C_Master_RepeatedStart(){            //I2C_Master_RepeatedStart function definition
    I2C_Master_Wait();                      //calling Wait Function
    RSEN = 1;                               //Enabling RSEN bit
}

void I2C_Master_Stop(){                     //I2C_Master_Stop function definition
    I2C_Master_Wait();                      //Calling Wait Function
    PEN = 1;                                //Enabling Stop Bit
}
```

```
unsigned char I2C_Master_Write(unsigned char data)      //I2C_Master_Write function definition
{
    unsigned char ack;                                  //declaring ack variable
    I2C_Master_Wait();                                  //calling Wait Function
    SSPBUF = data;                                      //Assigning SSPBUF the data
    while(!SSPIF);                                      //running loop till SSPIF is 1,checking data is transmitted or not
    SSPIF = 0;                                          //making SSPIF as 0
    ack = SSPCON2 & 0x40;                               //storing the value in ACKSTAT in ack variable
    return ack;                                         //returning ack
}

unsigned char I2C_Read_Byte(void)                       //I2C_Read_Byte function definition
{
    I2C_Master_Wait();                                  //calling Wait function
    RCEN = 1;                                           //making RCEN bit as 1
    while(!SSPIF);                                      //running loop till SSPIF is 1,checking data is transmitted or not
    SSPIF = 0;                                          //making SSPIF as 0
    I2C_Master_Wait();                                  //calling Wait again to check no other transmission is there
    return SSPBUF;                                      //returning the data in Buffer
}


void I2C_ACK(void)                                      //I2C_ACK function definition
{
    ACKDT = 0;                                          //making ACKDT as 0
    I2C_Master_Wait();                                  //calling Wait Function
    ACKEN = 1;                                          //Intiating acknowledge
}

void I2C_NACK(void)                                     //I2C_NACK function definition
{
    ACKDT = 1;                                          //making ACKDT as 1 , master sending NACK
    I2C_Master_Wait();                                  //Calling Wait Function
    ACKEN = 1;                                          //Intiating Acknowledge
}
```

```
void EEPROM_Write(unsigned int add, unsigned char data)         //EERPROM_Write Function
{
    I2C_Master_Start();                                         //Calling Start Function

    while(I2C_Master_Write(EEPROM_Address_W))                   //running till EEPROM is idle
        I2C_Master_RepeatedStart();

    I2C_Master_Write(add>>8);                                   //sending address
    I2C_Master_Write((unsigned char)add);                       //LSB first and MSB second
    I2C_Master_Write(data);                                     //writing Data
    I2C_Master_Stop();                                          //calling STOP function
    __delay_ms(1000);                                           //delay for 1000 ms
}

unsigned char EEPROM_Read(unsigned int add)                     //EEPROM Read Function
{
    unsigned char Data;                                     //declaring variable Data
    I2C_Master_Start();                                     //Calling Start Function


    while(I2C_Master_Write(EEPROM_Address_W))               //running till EEPROM is idle
        I2C_Master_RepeatedStart();

    I2C_Master_Write(add>>8);                                 //sending address
    I2C_Master_Write((unsigned char)add);                     //LSB first and MSB second
    I2C_Master_RepeatedStart();                               //Calling Repeated start to start again

    I2C_Master_Write(EEPROM_Address_R);                       //sending address with Read
    Data = I2C_Read_Byte();                                   //Storing Data in Data variable
    I2C_NACK();                                               //Master giving NACK Function
    I2C_Master_Stop();                                        //Calling STOP Function

    return Data;
}
```