```
#include <xc.h>
#define _XTAL_FREQ 20000000

#pragma config FOSC = HS        // Oscillator Selection bits (HS oscillator)
#pragma config WDTE = OFF       // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF      // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = ON       // Brown-out Reset Enable bit (BOR enabled)
#pragma config LVP = ON         // Low-Voltage (Single-Supply) In-Circuit Serial Programming Enable bit (RB3/PGM pin has PGM function; low-voltage programming enabled)
#pragma config CPD = OFF        // Data EEPROM Memory Code Protection bit (Data EEPROM code protection off)
#pragma config WRT = OFF        // Flash Program Memory Write Enable bits (Write protection off; all program memory may be written to by EECON control)
#pragma config CP = ON          // Flash Program Memory Code Protection bit (All program memory code-protected)

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#define DOOR_SENSOR_PIN PORTAbits.RA0
#define LIGHT_CONTROL_PIN PORTBbits.RB0
#define DOOR_LED_PIN PORTBbits.RB1
#define ROOF_SWITCH_PIN PORTBbits.RB1

void initSystem();
void initI2C();
void controlLighting();
void updateLEDStatus();
void transmitDoorStatus(unsigned char doorStatus);
void delay_ms(unsigned int ms);

void main() {
    unsigned char doorStatus = 0;

    initSystem();
    initI2C();

    while(1) {
        doorStatus = (DOOR_SENSOR_PIN == 0) ? 0 : 1;

        controlLighting();

        updateLEDStatus();

        transmitDoorStatus(doorStatus);

        __delay_ms(100);
    }
}

void initSystem() {
    TRISA = 0x01;
    TRISB = 0x00;
    ANSEL = 0x00;
    LATB = 0x00;
}

void initI2C() {
    SSP1STAT = 0x80;
    SSP1CON1 = 0x28;
    TRISCbits.TRISC3 = 1;
    TRISCbits.TRISC4 = 1;
}

void controlLighting() {
    if (DOOR_SENSOR_PIN == 0 && ROOF_SWITCH_PIN == 1) {
        PORTBbits.RB0 = 1;  // Set RB0 (Turn on light)
    } else {
        PORTBbits.RB0 = 0;  // Clear RB0 (Turn off light)
    }
}

void updateLEDStatus() {
    if (DOOR_SENSOR_PIN == 0) {
        PORTBbits.RB1 = 1;  // Set RB1 (Turn on LED)
    } else {
        PORTBbits.RB1 = 0;  // Clear RB1 (Turn off LED)
    }
}
```

```
void transmitDoorStatus(unsigned char doorStatus) {
    SSP1BUF = doorStatus;
    while (SSP1STATbits.BF);
}

void delay_ms(unsigned int ms) {
    while(ms--) {
        __delay_ms(1);
    }
}
```