


```

WRT_OFF & PLEN_OFF & STVREN_ON & LVP_OFF
);

// proto
void itostring(char , unsigned int , char* );
void mssp1_init(void);
void mssp2_init(void);
unsigned char i2c_scan(unsigned char);
unsigned char i2c2_scan(unsigned char);

// display data
unsigned char Msg1[17] = "i2c SCAN test ";
unsigned char Msg2[6] = "ADDR ";
unsigned char Msg3[5] = "ack ";
unsigned char Msg4[4] = " ";
unsigned char Msg5[10] = " ";

void main(void)
{
unsigned char i;

/* INITIALIZE REGISTER */
    OSCCON = 0b01111000;           // Set 16MHz

    LATA = 0b00000000;           // Clear
    TRISA = 0b00111100;         // LATA INPUT RA4,5:OSC,3:int,2:SW
    LATB = 0b11110000;         // LATB RESET RB7:LED

// MSSP2 will not work
    TRISB = 0b01110000;         // LATB INPUT RB6:SCL,5:SW,4:SDA
    LATC = 0b00000000;         // LATC RESET
    TRISC = 0b10000011;         // LATC INPUT RC7:SW,0,1:SHT

    ANSELA = 0b00000000;        // All digital
    ANSELB = 0b00000000;
    ANSELC = 0b00000000;

    CM1CON0 = 0b00000111;       // Not using compalator
    CM2CON0 = 0b00000111;       // Not using compalator

    OPTION_REG = 0b10100100;    // INTEDG 0,TOCS T0CKI 1,T0SE0,PSA
TIMERO,1/32
// not using INT
    INTCON=0;                    // INT off
    T1CON = 0;                   // Timer1 off

//CCP4 PWM initializing (83.8KHz on RB3 @16MHz)
    CCP4CON = 0b00001111;       // use PWM mode
    CCP4L = 0x0c;               // duty is 50%
    CCPTMRS = 0;                // Select TMR2
// TMR2 initilizing
    T2CON = 0b00000100;         // POSTSCALE 1:1 ,TMR2 ON , PRESCALE
1:1
    PR2 = 0x17;                 // resolution is 6.5bit mode
    TMR2ON = 1;                 // TMR2 start

```

```

    _delay_ms(200);
    _delay_ms(200);

    mssp1_init();                // MSSP initialize
    // mssp2_init();
    // MSSP2 will not work

    lcd_init();                 // LCD initialize continue
    HEART_BEAT = 1;            // LED off

    while(1)
    {
        _delay_ms(100);

        lcd_goto(0x00);        // Move cursor 1st line
        lcd_puts(Msg1);        // Display test message

        i=2;
        while(i){
            lcd_goto(0x40);    // Move cursor to 2nd line
            lcd_puts(Msg2);    // i2c addr message
            itostring(2,i,Msg4); // convert i to ASCII
            lcd_puts(Msg4);    // SCAN ADD

            if(i2c_scan(i)==0){ // If detect ack,displaying addr
                lcd_puts(Msg3);
                itostring(2,i,Msg4);
                lcd_puts(Msg4);
                while(UP_SW) ; // Waiting SW1 PUSH CHECK
                _delay_ms(400);
            }
            else lcd_puts(Msg5);

            i=i+2;
            _delay_ms(100);
            if(i&0x8)HEART_BEAT =1;// Heart beat LED
            else HEART_BEAT =0;
        }

        for(i=10;i>0;i--){    // 1s waiting
            _delay_ms(100);
        }
    }
}
/*****
* Converting 2/4 hex to ASCII
*****/
void itostring(char digit, unsigned int data, char *buffer)
{
    char i;                // digit:2 or 4
    buffer += digit;       // last data
    for(i=digit; i>0; i--){
        buffer--;
    }
}

```

```

        *buffer = (data & 0xF) + '0';    // ASCII code
        if(*buffer>0x39) *buffer=*buffer+7;
        data = data >> 4;                // next hex
    }
}

/*****
* MSSP1 initialize
*****/

void mssp1_init(void){
///    SSP2CON1 REGISTERS
    SSP1CON1bits.SSPEN = 1;    //Enables Serial Port Mode
    SSP1CON1bits.SSPM3 = 1;    ///////////////
    SSP1CON1bits.SSPM2 = 0;    //I2C Master Mode
    SSP1CON1bits.SSPM1 = 0;    // clock= Fosc/(4*(SSP2ADD+1))
    SSP1CON1bits.SSPM0 = 0;    ///////////////

//    SSP1CON2 REGISTERS
    SSP1CON2 = 0x00;
//    SSP1CON3 REGISTERS
    SSP1CON3 = 0x00;

//    SSP1STAT REGISTERS
    SSP1STATbits.SMP = 1;    // SLEW RATE non
    SSP1STATbits.CKE = 1;    //SMBus Specific Inputs Enabled

    SSP1ADD = 0x38;    //~75kHz
    // SSP1ADD = 0x26;    //~100kHz
    //SSP1ADD = 0x0E;    //~400kHz
//    SSP1ADD = 0xA0;
}

/*****
* MSSP2 initialize
*****/
// MSSP2 will not work
void mssp2_init(void){
///    SSP2CON1 REGISTERS
    SSP2CON1bits.SSPEN = 1;    //Enables Serial Port Mode
    SSP2CON1bits.SSPM3 = 1;    ///////////////
    SSP2CON1bits.SSPM2 = 0;    //I2C Master Mode
    SSP2CON1bits.SSPM1 = 0;    // clock= Fosc/(4*(SSP2ADD+1))
    SSP2CON1bits.SSPM0 = 0;    ///////////////

//    SSP2CON2 REGISTERS
    SSP2CON2 = 0x00;
//    SSP2CON3 REGISTERS
    SSP2CON3 = 0x00;

//    SSP2STAT REGISTERS
    SSP2STATbits.SMP = 1;    // SLEW RATE non
    SSP2STATbits.CKE = 1;    //SMBus Specific Inputs Enabled

    //SSP2ADD = 0x19;    //~75kHz for 8MHz

```

```

    SSP2ADD = 0x13;           //~100kHz
    //SSP2ADD = 0x07;        //~400kHz
    //SSP2ADD = 0x50;
}

/*****
* i2c address scanning test for M CCP1
*****/
unsigned char i2c_scan(unsigned char data){
    unsigned char ack_data;
    i2c_start();           // start condition
    i2c_write(data);      // addr & wite mode
    ack_data=i2c_readack(); // if ack:0
    i2c_stop();
    return (ack_data);
}

/*****
* i2c address scanning test for M CCP2
*****/
// MSSP2 will not work
unsigned char i2c2_scan(unsigned char data){
    unsigned char ack_data;
    i2c2_start();         // start condition
    i2c2_write(data);    // addr & wite mode
    ack_data=i2c2_readack(); // if ack:0
    i2c2_stop();
    return (ack_data);
}

```