


```

#include "LPC213x.h"

unsigned char SlaveRcv = 0xAA;
unsigned char SlaveSnd;

void I2C0_Isr(void) __irq
{
    unsigned char st;

    st = I2C0STAT;
    I2C0CONCLR = 0x2C;           // clear STA, AA and SI
    switch(st)
    {
        case 0x60:               // own SLA+W received, Ack returned (slave receiver)
        case 0x68:               // Addressed as slave
            I2C0CONSET = 0x04;   // set AA, return ACK on first byte
            break;
        case 0x80:               // Data received, ACK returned
            SlaveRcv = I2C0DAT;   // read and store data, NACK on next byte
            IOCLR1 = 0x00FF0000;  // Turn off LEDs P1.16..23
            IOSET1 = SlaveRcv << 16; // Turn on LED
            break;
        case 0x88:               // data received, NACK returned
        case 0xA0:               // STOP or REP.START received while addressed as slave
        case 0xC0:               // Data transmitted, NOT ACK received
        case 0xC8:               // Last data transmitted, ACK received
            I2C0CONSET = 0x04;   // set AA, switch to not addressed slave mode
            break;
        case 0xA8:               // own SLA+R received, Ack returned (slave transmitter)
        case 0xB8:               // Data transmitted, ACK received
            I2C0DAT = SlaveSnd;  // Transmit last data AA = 0
            break;
        default:
            break;
    }
    VICVectAddr = 0;           // reset VIC
}

void I2C0_Init(void)
{
    PINSEL0 |= 0x50;           // P0.3 = SDA, P0.2 = SCL
    I2C0ADR = 0x20;           // set I2C slave address
    I2C0CONSET = 0x44;        // enable I2C hardwar and set AA (ack)

    VICVectAddr0 = (unsigned int) &I2C0_Isr;
    VICVectCnt10 = 0x29;     // Channell on Source#9 ... enabled
    VICIntEnable |= 0x200;   // 9th bit is the I2C
}

```

