

Unbenannt

```
#include <REG936.H>
#define ACK 0x00
#define NACK 0x01
#define DATABYTE_FERTIG 0x02
#define NO_INFO 0xFF

unsigned char SlaveAdr, Temp_WERT;
unsigned char I2CStatus= NO_INFO;

main()
{
    unsigned char STCN75Val ;

    P1M1 |= 0x0C;           // SDA und SCL als Open-Drain
    P1M2 |= 0x0C;
    I2SCLH = 19;            // I2C-Geschwindigkeit ~100 KHz
    I2SCLL = 19;
    I2CON = 0x44;           // I2C-Hardware aktivieren
                                // (STO, STA, SI = 0, ACK =1)
    EI2C = 1;               // I2C Interrupt freigeben
    EA = 1;                 // allgemeine Interruptsperre aufheben

    SlaveAdr = 0x9F;        // Adresse Temperatursensor +R
    I2CON = 0x64;           // Startbedingung setzen

    while(I2CStatus == NO_INFO);
    if (I2CStatus == ACK)
    {
        I2CStatus = NO_INFO;    // Status zurücksetzen
        SlaveAdr += 1;          // Temperatursensor (STCN + R)
        I2CON = 0x64;           // Startbedingung setzen
    }
    while (I2CStatus == NO_INFO);
    if (I2CStatus == ACK)
    {
        while(I2CStatus == ACK);
        if (I2CStatus == DATABYTE_FERTIG)
            STCN75Val = Temp_WERT;
    }
    while(1);
}

void v_I2CInterrupt(void) interrupt 6
{
    switch(I2STAT) // Auswertung des Statusregisters
    {
        case 0x08: // Status: "START transmitted"
            I2DAT = SlaveAdr;
            I2CON = 0x44; // (STO, STA, SI = 0, ACK =1)
            break;
        case 0x18: // Status: "SLA+W transmitted; ACK received"
            I2CON = 0x54; // Stoppbedingung setzen (STO =1)
            I2CStatus = ACK;
            break;
        case 0x20: // Status: "SLA+W transmitted; NOT ACK received"
            I2CON = 0x54; // Stoppbedingung setzen (STO =1)
            I2CStatus = NACK;
            break;
        case 0x40: // Status: "SLA+R transmitted;
ACK received"
            I2CON = 0x44; // (STO, STA, SI = 0, ACK =1)
            I2CStatus = NACK;
            break;
        case 0x48: // Status: "SLA+R transmitted; NOT ACK received"
    }
}

Seite 1
```

```

                                Unbenannt
I2CON = 0x54;    // Stoppbedingung setzen (STO =1)
I2CStatus = NACK;
break;
case 0x50:        // Status: "Data byte received; ACK returned"
Temp_WERT = I2DAT;
I2CON = 0x54;    // Stoppbedingung setzen (STO =1)
I2CStatus = DATABYTE_FERTIG;
break;
}

```

```

}

```