```
#include <avr/pgmspace.h>
#include <avr/io.h>
#include <stdlib.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <util/sbit.h>



#include "i2c.h"
#include "ADCauslesen.h"
#include "globalextern.h"

uint8_t reg00 = 0;
uint8_t reg01 = 0;


void adcinit(void)
{
        TWI_START();                    // Sende Startbedingung
        TWI_MT_SLA_ACK();               // Sende Master Transmit Slave Address mit
          ACK
        TWI_MT_DATA_ACK(0b00000110);    // Sende Master Transmit Data mit ACK
          ( 0000 011x = Reset)
        TWI_STOP();

        TWI_START();                    // Sende Startbedingung
        TWI_MT_SLA_ACK();               // Sende Master Transmit Slave Address mit
          ACK
        TWI_MT_DATA_ACK(0b01000000);    // Sende Master Transmit Data mit ACK
          ( 0100 01xx = Schreibe auf Register 00)
        TWI_MT_DATA_ACK(0b00000000);    // Sende Master Transmit Data mit ACK
          ( 0000 0010 = AIN0+ AIN1- / Gain = 2)
        TWI_MT_DATA_ACK(0b01000100);    // Sende Master Transmit Data mit ACK
          ( 0100 01xx = Schreibe auf Register 01)
        TWI_MT_DATA_ACK(0b00001010);    // Sende Master Transmit Data mit ACK
          ( 0000 1010 = Externe Referenz + Continous mode)
        TWI_STOP();

        TWI_START();                    // Sende Startbedingung
        TWI_MT_SLA_ACK();               // Sende Master Transmit Slave Address mit
          ACK
        TWI_MT_DATA_ACK(0b00100100);    // Sende Master Transmit Data mit ACK
          ( 0010 01xx = Lese Register 01)
        TWI_R_START();                  // Sende Repeated Start
        TWI_MR_SLA_ACK();
        reg01 = TWI_READ_DATABYTE_NACK();
        TWI_STOP();



        TWI_START();                    // Sende Startbedingung
        TWI_MT_SLA_ACK();               // Sende Master Transmit Slave Address mit
          ACK
```

```
        TWI_MT_DATA_ACK(0b00100000);    // Sende Master Transmit Data mit ACK
          ( 0010 01xx = Lese Register 00)
        TWI_R_START();                  // Sende Repeated Start
        TWI_MR_SLA_ACK();
        reg00 = TWI_READ_DATABYTE_NACK();
        TWI_STOP();

        TWI_START();                    // Sende Startbedingung
        TWI_MT_SLA_ACK();               // Sende Master Transmit Slave Address mit
          ACK
        TWI_MT_DATA_ACK(0b00001000);    // Sende Master Transmit Data mit ACK
          ( 0000 100x = Starte Wandlung)
        TWI_STOP();
}


uint16_t Auslesen(void)
{
    uint16_t Temp = 0;
    uint16_t high_byte, low_byte;
    uint8_t MSByte = 1;
    uint8_t LSByte = 1;

//  _delay_ms(100);

    TWI_START();                    // Sende Startbedingung
    TWI_MT_SLA_ACK();               // Sende Master Transmit Slave Address mit ACK
    TWI_MT_DATA_ACK(0b00010000);        // Sende Master Transmit Data mit ACK
      ( 0001 xxxx = Lese Datenregister)
    TWI_R_START();                  // Sende Repeated Start
    TWI_MR_SLA_ACK();               // Sende Slave Adrdress in Master Recieve Mode
      mit ACK
    MSByte = TWI_READ_DATABYTE_ACK();
    LSByte = TWI_READ_DATABYTE_NACK();
    TWI_STOP();

    high_byte = MSByte << 8;
    low_byte = LSByte;
    Temp = high_byte + low_byte;
    return Temp;
}
```