



Lösung 8051er

```
*****  
| CLASS:      P8051er  
| Compiler:   KeiluV3  
| PROGRAM:   Blinc_C.c  
| AUTHOR:    Gerhard Neumaier  
| DATE:      15.Okt 2010  
| DESCRIPTION: LEDs am Port1 zum Blinken zu bringen  
|  
| REQUIREMENTS: 8051er z.B. ATMEL89C5131  
| NOTES:      Ein komplettes Programm, keine Aenderungen notwendig  
|             Aufgaben: Aendern der Portpins, Aenderung der Geschwindigkeit  
|             Ausgabe am IIC-Bus am IC PCF8574  
*****
```

```
*****/  
#include <at89c5131.h>  
  
#define Blink P1 //Der Variablen Blink wird der Port1 zugewiesen  
  
//Fuer IIC-Bus:  
void start_i2c(void);  
void byte_out(unsigned char wert);  
  
void init(void) //Eine Funktion mit dem Namen init wird angelegt  
{  
//nichts zu tun  
}  
  
void zeit(unsigned int verzoegerung); //Funktion zeit() dem Compiler bekanntgeben  
//weil die Funktion nach der main-Funktion steht  
//Die Funktion Zeit hat hier einen Übergabewert "verzoegerung"  
unsigned char adresse = 0x72; //Adresse für IIC Baustein auf Platine Birk  
unsigned int i=0; //i 16bit pos Zahlenbereich 0..65535  
unsigned char j; //j 8bit pos Zahlenbereich von 0..255  
  
void main(void)  
{  
    init();  
  
    1 start_i2c(); //Startbedingung für den IIC-Bus  
    for (j = 20; j != 0; j--) //Zeitschleife ->Start wird in der Simulation besser sichtbar  
    {  
        2 byte_out(adresse); //Adresse für den IIC-Slave auf den Bus geben  
        while(1) // Endlosschleife  
        {  
            3 Blink = 0x32; // PORT 0011 0010 setzen  
            byte_out(Blink); // Ausgabe auf den Busleitungen SCL und SDA  
            zeit(60000); //Aufruf Funktion zeit Delay Übergabe hier 60000  
            Blink = 0; // PORT löschen  
            byte_out(Blink); // Ausgabe auf den Busleitungen SCL und SDA  
            zeit(30000); //Aufruf Funktion zeit Delay Übergabe hier 30000  
        }  
    }  
    //Ende main  
  
//Funktion zeit() zur Zeitverzögerung (Softwarezeitschleife)  
void zeit(unsigned int verzoegerung) //Funktion zeit  
{  
    for(i=0;i<verzoegerung;i++) //der Übergabewert verzoegerung wird in die for-Schleife eingesetzt  
    {  
        //keine Anweisung notwendig  
    }  
}
```

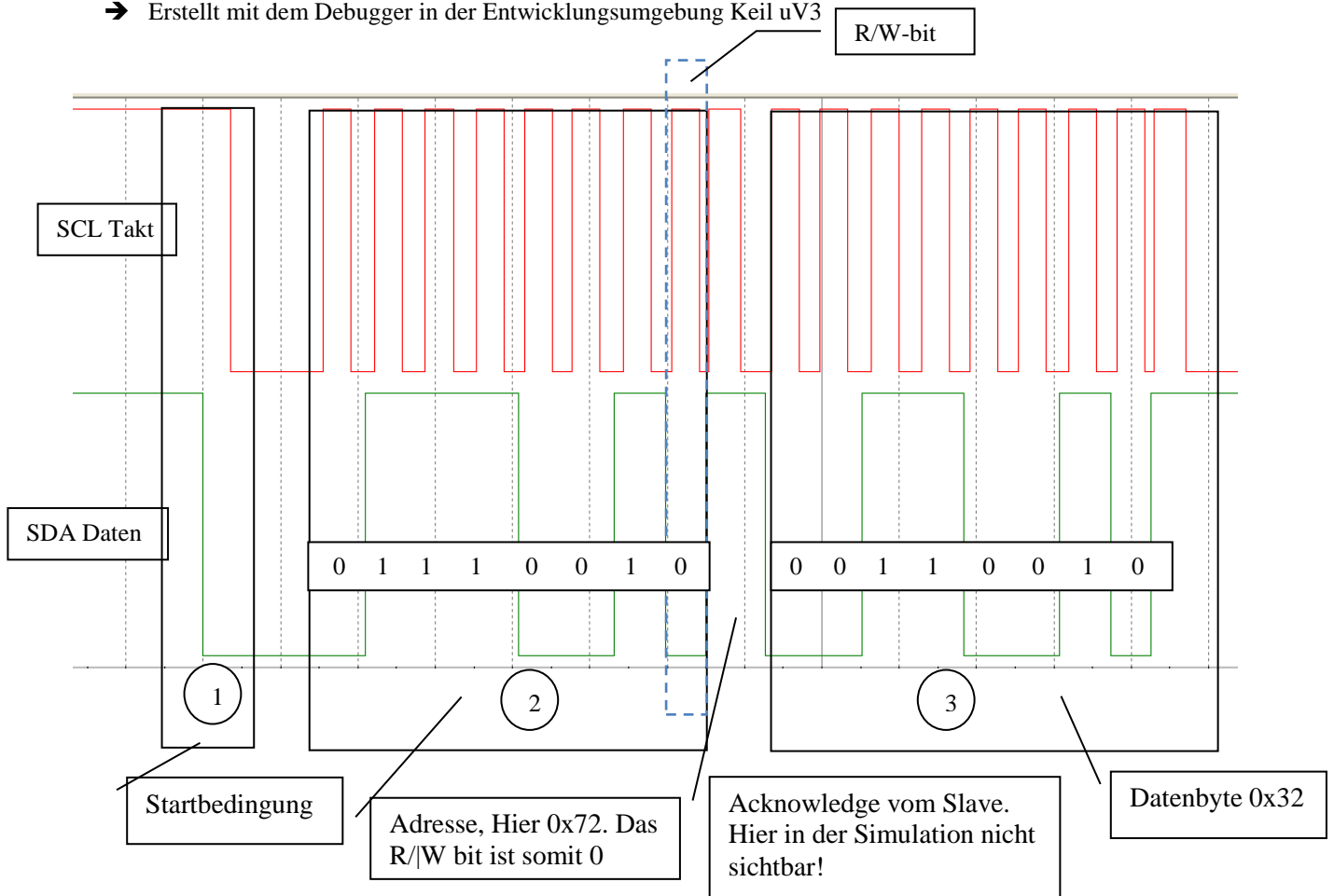


Hinweis:

Die **markierten Befehle** sind im Programm die Ergänzung für die Datenausgabe über IIC

Darstellung der Signale auf dem IIC-Bus

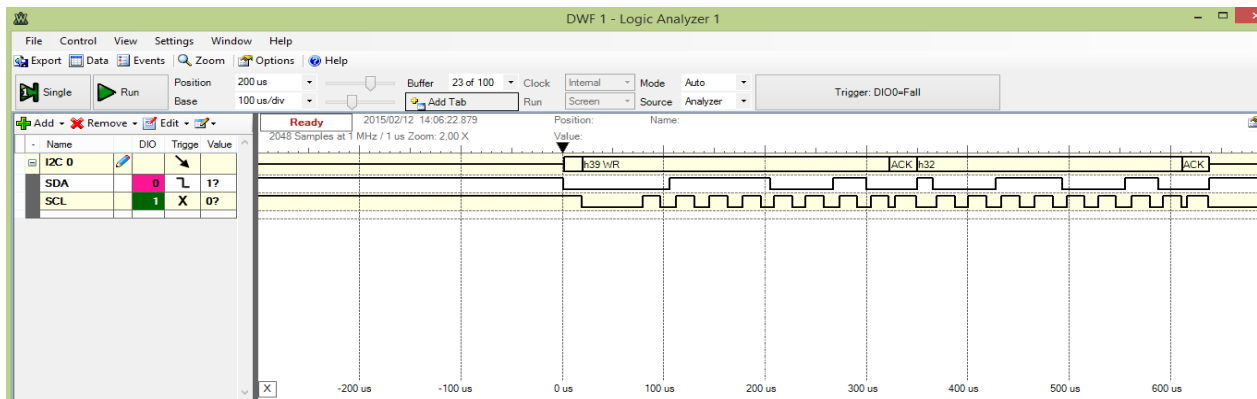
→ Erstellt mit dem Debugger in der Entwicklungsumgebung Keil uV3



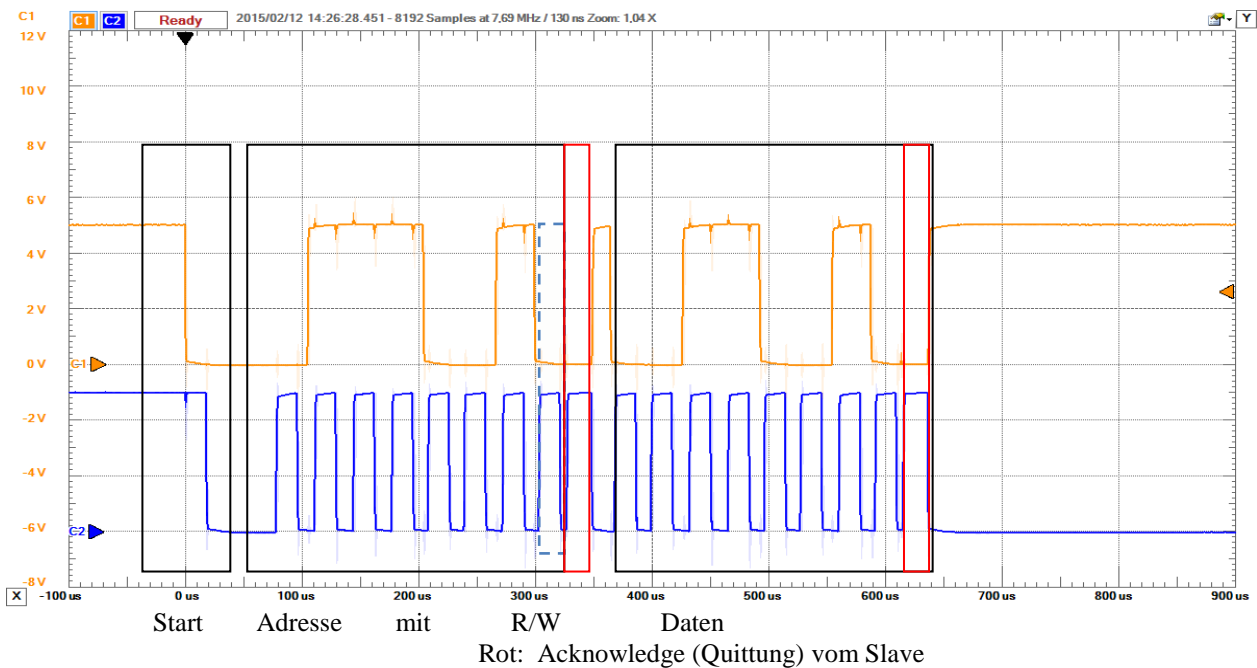
In der Simulation mit dem Logik Analyser von Keil kann man nur die selbst erzeugten Signale vom Master anzeigen. Ein eingehendes Signal vom Slave leider nicht. Deshalb sieht man nach der Adresse das eingehende Acknowledge vom Slave nicht. Die SDA Leitung ist bei gültigem Acknowledge 0.



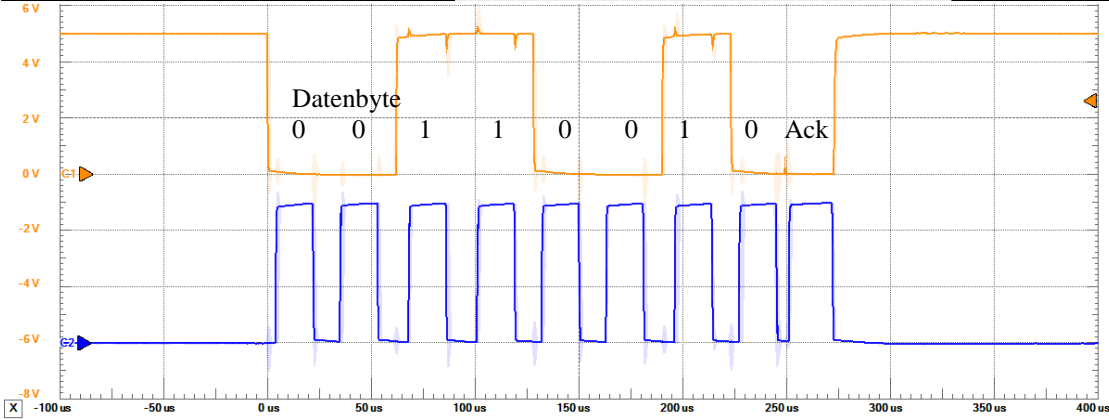
Messung mit „Analog Discovery“ von Digilent mit dem Logic-Analyser



Messung mit „Analog Discovery“ von Digilent mit dem Oszilloskop



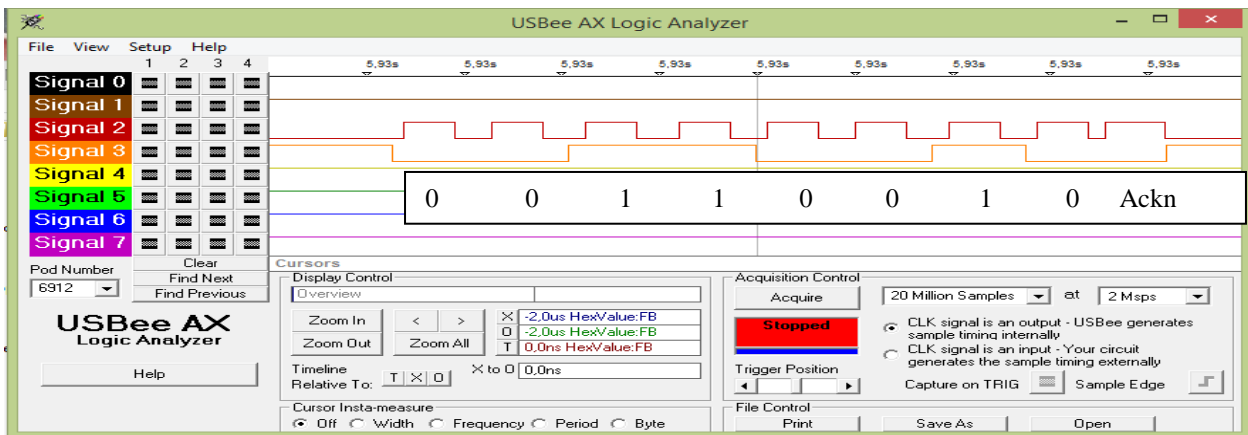
Gesendet wird Startbedingung, dann die Adresse 0x72 (Das letzte bit is R/W in diesem Fall 0)
Danach meldet der Slave (IC 8574a) das Acknowledge-bit. Danach sendet der Master das Datenbyte 0x32.
Der Slave quittiert wieder mit 0 auf der Datenleitung.

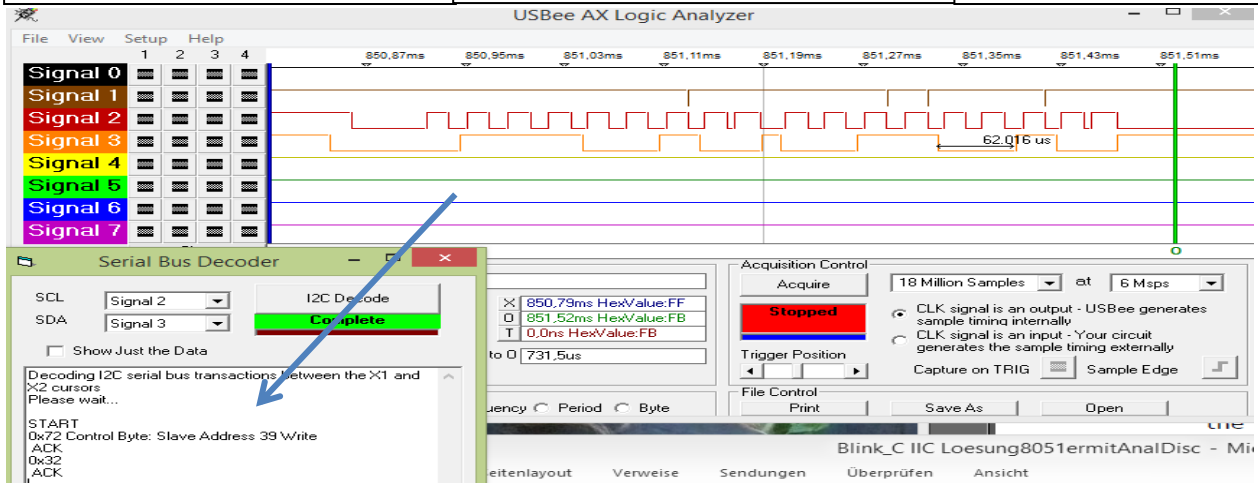


Messung mit „USBee AX PRO“:
Gesamtüberblick

20 mill Samples mit einer Abtastrate von 2Mega-Samples ergibt eine Messung über 10 sec. Man erkennt in der Übersicht die Signale am IIC-Bus.

Im Bild unten eine Datenübertragung am IIC Bus:





Auswertung mit Software. Dargestellt die Startbedingung, Slaveadresse 0x 72 -> 0x39 1bit nach rechts geschoben