



Dokument NR.: ZB-005-U-06

LED ALS SIGNALGEBER 3



LED als Signalgeber
Timer 1, Mode 1
im Polling-Betrieb



Bitte denken Sie an die Umwelt,
bevor Sie diese Datei ausdrucken

INHALTSVERZEICHNIS

- 1. Aufgabe 1 3
 - 1.1 Verwendete Module: 3
 - 1.2 „Target Mission“ 3
- 2. Lösung Aufgabe 1 4

Modification History:

Version	Date	Comments
1.0	01.2013	first release

1. AUFGABE 1

Programmieren Sie einen LED-Signalgeber.

- LED soll exakt 250ms an und 250ms ausbleiben.
- Benutzen Sie die LED am Port 2 Pin 0
- Benutzen Sie **Timer 1 im Mode 1**
- Polling-Betrieb

1.1 VERWENDETE MODULE:

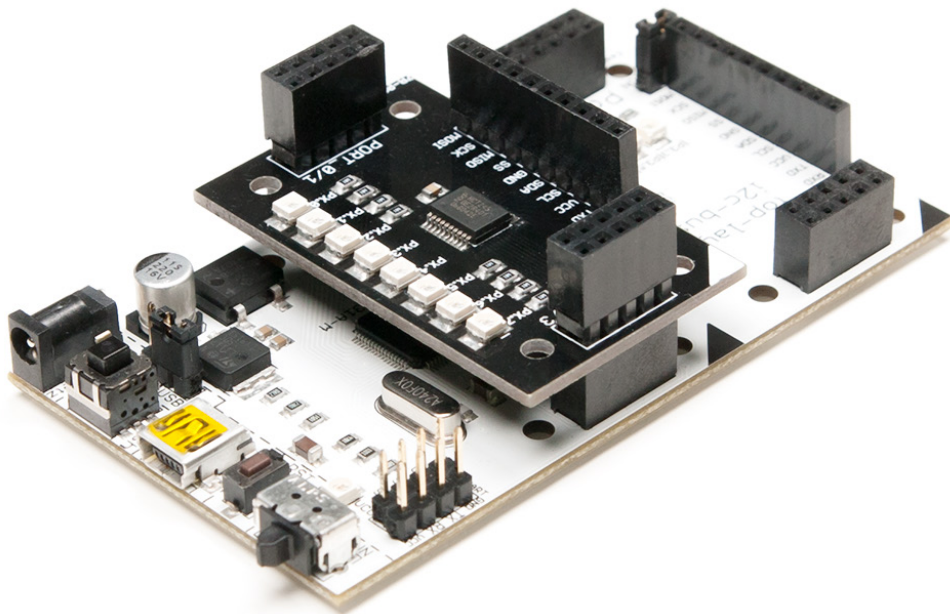
8051 Mikrocontroller Board mit AT89C5131A-RDTUM mit einem 24MHz Quarz (SB-001/SB-002) und LED Board (ZB-005).

1.2 „TARGET MISSION“

Ein Video sagt mehr, als tausend geschriebene Worte.

In diesem Video wird gezeigt wie das Ergebnis aussehen soll (ZB-005-U-06). Es wird kein Quellcode gezeigt oder erklärt.

[„Target Mission“ auf YouTube](#)



2. LÖSUNG AUFGABE 1

```

/*****
Datei:      ZB-005-U-06_Timer1-Model-polling.c
Produkt:    ZB-005-U-06
Projekt:    Timmer1/Model/Polling

Datum:      Jan.2013 - first release
Version     1.0
Kompiler:   Keil V9.05
MC:         AT89C5131A (8051)
Quarz:      24MHz

Autor:      Viktor Schabelski info@i2c-bus.de.de
Lizenz:     Creative Commons Lizenz
            Namensnennung - Keine kommerzielle Nutzung - Keine Bearbeitungen
            www.i2c-bus.de/cc

Programmieren Sie einen LED-Signalgeber.
- LED soll exakt 250ms an und 250ms ausbleiben.
- Benutzen Sie die LED am Port 2 Pin 0
- Benutzen Sie Timer 1 im Mode 1
- Polling-Betrieb

*****/

/*****
Includes
*****/
#include <at89c5131.h>

sbit bitLED = 0xA0; /* Port 2 Pin 0 */

/*****
Funktionen
*****/

/*=====
* Name : main
* Input: none
* Output : none
* Descr. : Main Funktion
-----*/
void main(void) {
    unsigned int uiCount = 0x00;

    P2 = 0x00; /* Port2 auf 0 setzen */

    /* Timer1 init-----*/

    TCON |= 0x40; // TR1 (2^6) = 1; Timmer1 ein
    TMOD |= 0x10; // M01 (2^4) = 1; Timmer1 Model ein

    /* 250m : (12/24MHz) = 500.000d => Zählschritte
    50.000d ist eine Zahl zwischen 1...2^16
    2^16 - 50.000d = 15.536d = 0x3CB0h werden in die Register geladen => 10 Überläufe müssen
    gezählt werden
    */

    TH1 = 0x3C;
    TL1 = 0xB0;

```

```
/* Timmer1 Overflow Flag auf 0 setzen, Register TCON */
TF1 = 0;

/* end of Timer1 init-----*/

while(1){
    while(TF1 != 1);
    uiCount++;
    TF1 = 0;

    /* kein Autoreload !!! Timer 1 Register neu laden */
    TH1 = 0x3C;
    TL1 = 0xB0;

    if(uiCount == 10){
        uiCount = 0;
        bitLED =~ bitLED;
    }
}
}
```

Haben Sie einen Fehler entdeckt?

Wir sind dankbar für Ihren Hinweis.
Schicken Sie uns bitte diesen Hinweis
einfach per E-Mail: info@i2c-bus.de.

Vielen Dank!