

Dokument NR.: ZB-005-U-04

# LED ALS SIGNALGEBER 1



LED als Signalgeber  
Timer 0, Mode 2  
im Polling-Betrieb



# 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 0 im Mode 2**
- Polling-Betrieb

## 1.1 VERWENDETE MODULE:

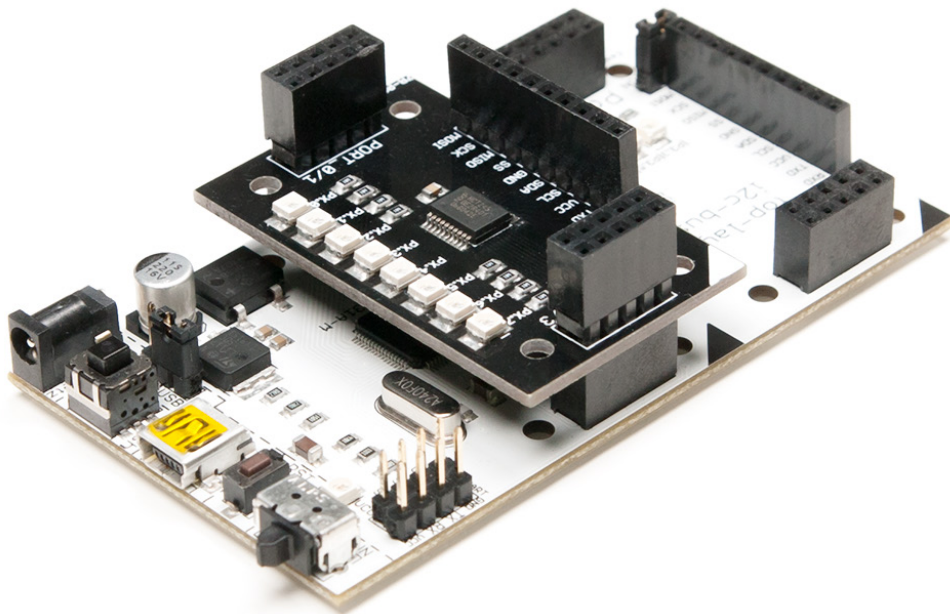
8051 Mikrocontroller Board mit AT89C5131A-RDTUM mit einem 24MHz Quarz (SB-001/SB-002) und LED Board (ZB-005).

## 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-04). Es wird kein Quellcode gezeigt oder erklärt.

[„Target Mission“ auf YouTube](#)



## 2. LÖSUNG AUFGABE 1

```

/*****
Datei:      ZB-005-U-04_Timer0_Polling.c
Produkt:    ZB-005-U-04
Projekt:    Timmer0/Mode2/Polling

Datum:      Jan.2013 - first release
Version     1.0
Kompiler:   Keil V9.05

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 0 im Mode 2
- Polling-Betrieb

*****/

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

    /* Timer0 init-----*/

    TCON = 0x10; /* TR0 (2^4) = 1; Timmer0 ein */
    TMOD = 0x02; /* M10 (2^1) = 1; Timmer0 Mode2 */

    /* Berechnung Zeit t = 250ms =>
    250ms:(12/24MHz) = 500.000d muss man zaehlen um 250ms zu erreichen.
    250d ist ein Probewert (zwischen 1 ... 2^8) -> Wichtig! Teilung ohne Rest!
    500.000d:250d = 2.000d;

    Um die Zeit von 250ms zu erreichen braucht man 2.000d Ueberlaeufe.
    Den Wert den man in die beiden Register laden muss, berechnet man
    (2^8 - 250d) = 5d = 0x05h
    Da wir 250 Maschinenzyklen zaehlen moechten, schreiben wir in den Register 0x05h.
    */

```

```
    TL0 = 0x00;
    TH0 = 0x05;

//Timmer0 Overflow Flag auf 0 setzen, Register TCON

    TF0 = 0;

/* end of Timer0 init-----*/

while(1){
    while(TF0 != 1);    /* warten auf TF0 */
    uiCount++;
    TF0 = 0;
    if(uiCount == 2000){
        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](mailto:info@i2c-bus.de).

Vielen Dank!