



Dokument NR.: ZB-005-U-08

# LED ALS SIGNALGEBER 5



LED als Signalgeber  
Timer 2, Auto-Reload Mode  
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	02.2013	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 2 im Auto-Reload Mode**
- Polling-Betrieb

## 1.1 VERWENDETE MODULE:

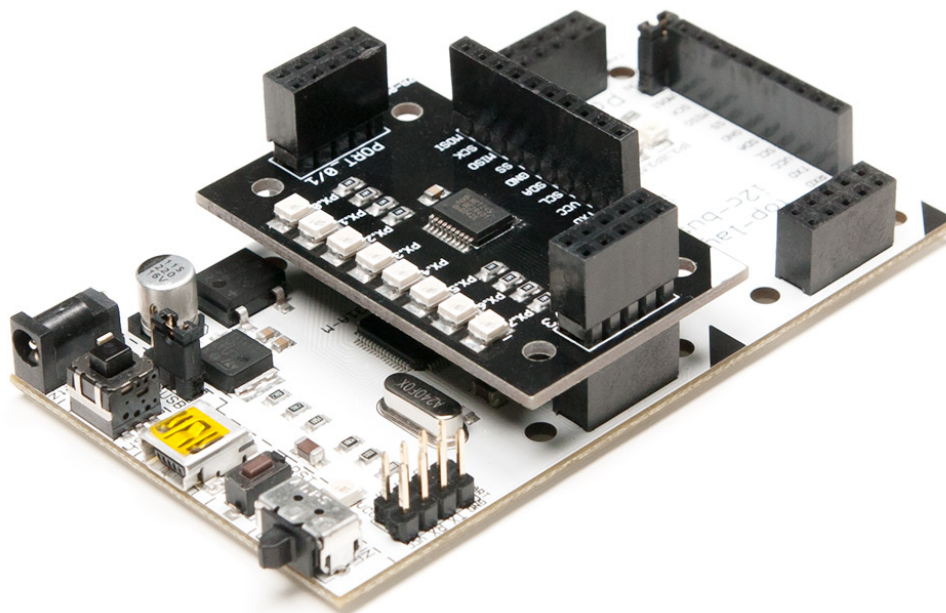
8051 Mikrocontroller Board mit AT89C5131A-RDTUM mit einem 24MHz Quarz (SB-001/SB-002) und LED Board (ZB-005).

## 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-08). Es wird kein Quellcode gezeigt oder erklärt.

[„Target Mission“ auf youTube](#)



## 2. LÖSUNG AUFGABE 1

```

/*****
Datei:      ZB-005-U-08_Timer2-auto-relod-mode-polling.c
Produkt:    ZB-005-U-08
Projekt:    Timmer2/Auto-Relod-Mode

Datum:      Feb.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 2 im Auto-Reload Mode
- 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 char uchCount = 0x00;

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

    /* Timer2 init-----*/

    T2MOD = 0x00;
    T2CON = 0x04;

    /* 250m : (12/24MHz) = 500.000d => Zaehlschritte
    50.000d ist eine Zahl zwischen 1...2^16
    2^16 - 50.000d = 15.536d = 0x3CB0h werden in die Register geladen => 10 Ueberlaeufe mues-
    sen gezaehlt werden
    */

    TH2 = 0x3C;
    TL2 = 0xB0;
    RCAP2H = 0x3C;
    RCAP2L = 0xB0;

```

```
/* Timmer2 Overflow Flag auf 0 setzen, Register T2CON */
TF2 = 0;
/* end of Timer2 init-----*/

while(1){
    while ((T2CON & 0x80) == 0); /* entspricht while(TF2 == 0);*/
    uchCount++;
    TF2 = 0;
    if(uchCount == 10){
        uchCount = 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!