

---

# Protokoll

## LEDs blinken

---

Systemtechnik Theorie  
4CHIT 2016/17

Martin Wölfer

Note:  
Betreuer: WEIJ

Version 0.2  
Begonnen am 7. September  
Beendet am 8. September

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Ziel . . . . .	1
1.2	Voraussetzungen . . . . .	1
1.3	Aufgabenstellung . . . . .	1
<b>2</b>	<b>Ergebnisse</b>	<b>2</b>
2.1	Code . . . . .	2
2.2	Probleme mit Interrupts . . . . .	2
2.3	Bilder Board . . . . .	3

# 1 Einführung

## 1.1 Ziel

LEDs am STM32F3Discovery-Board blinken

## 1.2 Voraussetzungen

- Grundwissen C
- STM32F3Discovery Board
- PDF mit Pin Belegung

## 1.3 Aufgabenstellung

Erzeuge eine blinkende LED mit Hilfe der Interrupt-Technik.  
Schreibe dazu ein kurzes Protokoll mit einem Foto.

## 2 Ergebnisse

### 2.1 Code

```

1  /**
   ****
3  * @file    main.c
   * @author  Woelfer Martin
5  * @version V1.0
   * @date    08-Oktobre-2016
7  * @brief    Default main function.
   ****
9  */

11 //Importiere GrundLibrary fuer das Board
#include "stm32f3xx.h"

13 //Definiere eine Variable welche alle Pins enthaelt
15 #define PINS GPIO_PIN_8 | GPIO_PIN_9 | GPIO_PIN_10 | GPIO_PIN_11 | GPIO_PIN_12 | GPIO_PIN_13 |
   GPIO_PIN_14 | GPIO_PIN_15

17 //Main Funktion
19 int main(void) {
   //Initialisiere die HAL Library
21   HAL_Init();
   //initialise die LEDs
23   LED_Init();
   //Endlosschleife
25   while(1){
       //Die Pins "aufdrehen", der erste Parameter GPIOE gibt an bei welchem Port die Pins
       //Der zweite Parameter PINS gibt an welche Pins genau angesteuert werden sollen
27       HAL_GPIO_TogglePin(GPIOE, PINS);

29       //Einen Delay Einbauen damit es blinkt => Statt Interrupt
31       HAL_Delay(500);
   }
33 }

35 void LED_Init(void){
37   //Eine Variable initialisieren welche man dann konfigurieren
   GPIO_InitTypeDef LED_Init;

39   //Auf dem Port E (GPIOE) aktivieren
41   __HAL_RCC_GPIOE_CLK_ENABLE();

43   //Sagen welche Pins angesteuert werden sollen (Alle)
   LED_Init.Pin = PINS;
45   //Modus auf Push Pull setzen
   LED_Init.Mode = GPIO_MODE_OUTPUT_PP;
47   //Pull Modus setzen auf NoPull
   LED_Init.Pull = GPIO_NOPULL;
49   LED_Init.Speed = GPIO_SPEED_FREQ_LOW;

51   //GPIO Initialisieren mit GPIO (Port E) und der "Konfiguration"
   HAL_GPIO_Init(GPIOE, &LED_Init);
53 }

```

### 2.2 Probleme mit Interrupts

Ich habe probiert das Programm mit Interrupts zu schreiben bin tatsächlich aber schon an den kleinsten Probleme wie das importieren von EXTI nicht mehr weitergekommen.

In der HAL Library existiert kein File um EXTI zu includen wie z.B. `#include stm32f3xx_hal_exti`.

Die Idee hinter Interrupts ist in der Methode in welcher die LEDs initialisiert werden auch gleich eine EXTI\_Line hinzuzufügen welche man anschließend mit `rising` und `falling` auf- und abdrehen kann. Dies wäre eine etwas eleganter Art um das Board blinken zu lassen

## 2.3 Bilder Board

