


Arbeitsauftrag		
Fach:	AIIT 4JG Höhere Elektrotechnik	
Thema:	Timer Funktionen	
Lernziel:	Verwendung eines Timers Verwendung eines PWM Ausgangs	
Zeitdauer:	2 x 2 Einheiten	

Eine LED soll mit einer Frequenz von 1Hz blinken. Dies soll auf verschiedene Arten implementiert und die Lösungen miteinander verglichen werden:

1. Zählen der Durchläufe im Hauptprogramm.
Der SysTick Timer wird verwendet um einen Teil des Hauptprogramms alle 100ms auszuführen. Verwende diesen Teil des Programms um die LED blinken zu lassen.
2. Verwende den TIM3 um eine weitere LED mit der Frequenz von 1Hz blinken zu lassen. Verwende den Update Interrupt zu diesem Zweck.
3. Konfiguriere TIM3 nun für die Verwendung einer PWM. Verwende dafür Channel1 von TIM3, dieser kann mit GPIO PB4 gekoppelt werden. Setze den Duty Cycle der PWM auf 50%. Vergleiche die Blinkfrequenz dieser Konfiguration mit der Blinkfrequenz aus Punkt 2.
4. Konfiguriere TIM9/Channel 1 ebenfalls für die Verwendung als PWM Ausgang. Die PWM Frequenz soll 5kHz betragen, Duty Cycle 70%. Finde den zugehörigen GPIO.

Für die Lösung der Aufgaben sollen in der Datei blinky.c, die folgenden Funktionen implementiert werden:

```
static void GpioInit(void);
static void Timer3Init(void);
static void Timer3PwmInit(void);
static void Timer9Init(void);
static void Timer9PwmInit(void);
```

Beachte, dass alle Funktionen mit Kommentaren dokumentiert werden müssen!

Programmierung mit StdPeriphLib

Timer Konfiguration:

```
TIM_TimeBaseInitTypeDef Tim2Config;

/* TIM2 clock enable */
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
/* Time base configuration */
Tim2Config.TIM_Period = 400;
Tim2Config.TIM_Prescaler = 2300;
Tim2Config.TIM_ClockDivision = 0;
Tim2Config.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM2, &Tim2Config);
/* TIM2 enable counter */
TIM_Cmd(TIM2, ENABLE);
```

Timer Overflow Interrupt:

```
NVIC_InitTypeDef Tim2IrqConfig;
/* Enable the TIM2 global Interrupt */
Tim2IrqConfig.NVIC_IRQChannel = TIM2_IRQn;
Tim2IrqConfig.NVIC_IRQChannelPreemptionPriority = 0;
Tim2IrqConfig.NVIC_IRQChannelSubPriority = 1;
Tim2IrqConfig.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&Tim2IrqConfig);

TIM_ITConfig(TIM2, TIM_IT_Update, ENABLE);
```

PWM Config:

```
TIM_OCInitTypeDef PwmConfig;

PwmConfig.TIM_OCMode = TIM_OCMode_PWM2;
PwmConfig.TIM_OutputState = TIM_OutputState_Enable;
PwmConfig.TIM_OCPolarity = TIM_OCPolarity_Low;
PwmConfig.TIM_Pulse = 199; /* 50% duty cycle */
TIM_OC1Init(TIM2, &PwmConfig);
TIM_OC1PreloadConfig(TIM2, TIM_OCPreload_Enable);
```