

Name: **Marco Söllinger s2410306011**

Gruppe: 1 | 2

Punkte: / 24

ERSTE SCHRITTE: LAUFLICHT

AUFGABE

Programmieren Sie ein Lauflicht mit LED3, LED4 und LED5 auf dem APUS Board. Die LED-Ports und die entsprechenden Port-Pins sind aus dem Schaltplan ersichtlich.

Das Lauflicht soll von links nach rechts nach links usw. laufen. Die Richtung muss daher bei der linken bzw. rechten LED umgekehrt werden. Die LEDs sollen ca. alle 200ms wechseln.

Als Basis kann das Demo-Projekt inkl. der Funktionen aus dem GPIO-Treiber verwendet werden.

ANLEITUNG

1. Machen Sie sich mit dem EvalBoard, der Programmierumgebung und den Datenblättern vertraut. Mit welcher Geschwindigkeit läuft der Prozessor?
2. Programmieren Sie die Anpassungen zum bestehenden Projekt.
3. Testen Sie das Programm auf dem Eval Board.

HINWEISE

- Das Programm muss immer in einer Endlosschleife enden.
- Programmieren Sie das Programm in den Flash-Speicher, damit es nach dem Power-On sofort startet.

ABGABE (elektronisch)

- Kommentierter Sourcecode (20 Punkte)
- Dokumentation (4 Punkte)

0.1 Aufgabe 1:

In dieser Aufgabe wurde, dass Laufflicht implementiert.// Dabei ist darauf zu achten, dass die LED's bei einem Low-Signal leuchten.

Diese Aufgabe wurde auf dem Registerlevel mit direkten Zugriff auf die Register der GPIO's implementiert. Das APUS-Board benutzt extern fuer den Mikrocontroller einen 8MHz Quarz.

Wobei der Mikrocontroller intern mit einer Taktfrequenz bis zu 48MHz arbeiten kann.

led.c

```

1  /**
2  ****
3  * @file    led.c
4  * @author  Marco Sllinger
5  * @version V1.0
6  * @date    10.10.2025
7  * @brief   Main program body
8  ****
9  */
10
11 /* Includes -----*/
12 #include "stm32f0xx_conf.h"
13 #include "sysdelay.h"
14
15 /* Private typedef -----*/
16 /* Private define -----*/
17 /* Private macro -----*/
18 #define LED3_ON GPIO_BSRR_BR_0
19 #define LED3_OFF GPIO_BSRR_BS_0
20 #define LED4_ON GPIO_BSRR_BR_1
21 #define LED4_OFF GPIO_BSRR_BS_1
22 #define LED5_ON GPIO_BSRR_BR_2
23 #define LED5_OFF GPIO_BSRR_BS_2
24
25 /* Private variables -----*/
26 /* Private function prototypes -----*/
27 /* Private functions -----*/
28
29 /**
30 * @brief Initialize the sys tick timer (M0 core) which is used for delay.
31 * @param None
32 * @retval None
33 */
34 void SysTick_Init(void)
35 {
36     /* init the sys tick timer to be called every 1ms */
37     SysTick_Config(SystemCoreClock / 1000);
38 }
39
40 /**
41 * @brief This is the SysTick interrupt handler which is called every 1ms.
42 *        We have to increment the HAL tick counter which is used for SysDelay
43 * @param None
44 * @retval None
45 */
46 void SysTick_Handler()
47 {
48     SysDelay_IncTicks();
49 }
50
51 /**
52 * @brief This function advances the Led cycle by one step
53 *        The operation in this functions are atomic.
54 * @param None
55 * @retval None
56 */
57 void cycleLed(){
58     static uint8_t pos = 1;
59
60     switch(pos) {
61         case 0:
62             GPIOC->BSRR = (LED3_ON | LED4_OFF | LED5_OFF);

```

```

63         break;
64     case 1:
65         GPIOC->BSRR = (LED3_OFF | LED4_ON | LED5_OFF);
66         break;
67     case 2:
68         GPIOC->BSRR = (LED3_OFF | LED4_OFF | LED5_ON);
69         break;
70     case 3:
71         GPIOC->BSRR = (LED3_OFF | LED4_ON | LED5_OFF);
72         break;
73
74     default:
75         GPIOC->BSRR = (LED3_ON | LED4_ON | LED5_ON);
76         pos = 0xFF;
77         break;
78
79     }
80
81     pos = (pos+1)%4;
82 }
83
84 /**
85  * @brief Setup the initial state for the Led cycle
86  *          Doesnt setup the clock of the Port
87  * @param None
88  * @retval None
89  */
90 void initCycleLed(){
91     GPIOC->BSRR = (LED3_ON | LED4_OFF | LED5_OFF);
92     GPIOC->MODER |= GPIO_MODER_MODER0_0 | GPIO_MODER_MODER1_0 | GPIO_MODER_MODER2_0;
93 }
94
95 /**
96  * @brief Main program
97  * @param None
98  * @retval None
99  */
100 int main(void)
101 {
102     SysTick_Init();
103
104     // enable clock for GPIO Port C
105     RCC->AHBENR |= RCC_AHBENR_GPIOCEN;
106
107     initCycleLed();
108
109     while (1)
110     {
111         SysDelay_Delay(200);
112         cycleLed();
113     }
114 }
115

```