



PRAKTIKUM EINGEBETTETE SYSTEME
SS2026
Termin 1
C-Programmierung für eingebettete Systeme, PIO

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V:Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

Lernziele:

Im ersten Termin werden wir uns mit dem Parallel Input Output (PIO) und dem Powermanagement-Controller (PMC) befassen. Wir werden diese nutzen, um die verbaute LED auf dem Arduino Due anzusteuern und auf Eingaben über die Matrix-Tastatur zu reagieren.

Aufgabe 1

In dieser Aufgabe möchten wir unsere zweite Menükomponente implementieren. Diese besitzt 4 Zustände:

- LED_ON - Die LED soll eingeschaltet sein.
- LED_OFF - Die LED soll ausgeschaltet sein.
- LED_BLINKING - Die LED soll blinken. Dies könnte später genutzt werden, um zu erkennen, ob das Programm ausgeführt wird oder an einer Stelle hängt, wie z.B. in einem Breakpoint.

Sie können zwischen den einzelnen Zuständen durch Betätigen der Taste "L" wechseln.

1. Initialisieren Sie die verbaute LED (PIOB27) auf dem Arduino Due. Implementieren Sie hierfür die Funktion `led_init`. Verwenden Sie hierfür die Register PER und OER.
2. Implementieren Sie die Funktion `led_get`. Diese soll den aktuellen Zustand der LED zurückgeben. Verwenden Sie hierfür das Register ODSR.
3. Implementieren Sie das An- und Ausschalten der LED durch die Funktion `led_set`. Dies können Sie realisieren, indem Sie die Register SODR und CODR verwenden. Den Zustand, welchen die LED haben soll, können sie über die globale Variable `state_led` auslesen.

Aufgabe 2

1. Wir wollen nun unser Programm erweitern, um zwischen den einzelnen Zuständen der LED durch betätigen der Tasten auf Tastatur-Matrix wechseln. (Von der Tastatur-Matrix sind nur die Tasten 1 bis 3 (PIOD0 - PIOD2) angeschlossen). Zum Wechseln zwischen den Zuständen können sie eine Taste oder alle drei Tasten verwenden. Über die Tasten soll zwischen dem Zustand LED_BLINKING, LED_ON und LED_OFF gewechselt werden können.
2. Erweitern sie die `led_init` Funktion:
3. Schalten sie im PMC die verwendeten PIO Controller ein. Nutzen sie hierfür die Register PCER0 und PCER1.
4. Initialisieren sie zunächst die verwendeten Tasten als Eingänge. Verwenden sie hierfür das ODR.
5. Welchen Unterschied hat ihre Implementierung auf das Zeitverhalten des Mikrocontrollers (Loop Frequenz)?

Leitfragen

1. Wofür stehen die einzelnen Registerabkürzungen und welchen Nutzen haben diese?
2. Wieso muss Peripherie im PMC aktiviert werden.
3. Welchen Unterschied gibt es zwischen der Nutzung von Busy Waiting und Polling? Was sind die Vor- und Nachteile?
4. Haben Sie weitere Fragen?