

Fachgruppe Technische
Informatik

C-Programmierung für eingebettete Systeme,
Timer-Counter-Baustein, Wave-Modus, Puls-
Weiten-Modulation



PRAKTIKUM EINGEBETTETE SYSTEME

SS2026

Termin 4

C-Programmierung für eingebettete Systeme, Timer-Counter-Baustein,
Wave-Modus, Puls-Weiten-Modulation

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

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

Lernziele:

Wir befassen uns mit der Motoransteuerung des Fahrzeugs. An jedem Arbeitsplatz der Versuchsaufbauten befindet sich ein Motortreiber mit einem entsprechenden Motor. Im Fahrzeug ist derselbe Motortreiber mit einem leistungsfähigeren Motor verbaut. Es ist wichtig, bereits jetzt das Ansprechverhalten des Motors im Fahrzeug vom festen Versuchsaufbau unterscheiden zu können. Die Geschwindigkeit des Motors wird durch den Duty Cycle eines PWM-Signals gesteuert. Die Fahrtrichtung kann über die PIO-Ausgänge (PIOC23 und PIOC24) gesteuert werden, die durch entsprechende Pegel vorwärts oder rückwärts fahren ermöglichen.

Aufgabe 1

Schreiben/Ergänzen Sie, zur Initialisierung eines PWM-Signal (ENA) an TIOA6/PIOC25 und der für die Richtung notwendigen Signale (IN1, IN2) an PIOC23 und PIOC24, den benötigten Code in `drive_init()` und `drive_loop()`.

Aufgabe 2

Implementieren Sie die Motorsteuerung in der Funktion `drive_loop`. Denken Sie daran, dass Ihr Regelalgorithmus (im nächsten Termin zu implementieren) zur Ansteuerung des Motors verwendet werden soll. Die Implementierung erfolgt dann in den Dateien `motion.h` und `motion.c`.

Leitfragen

- Wie könnte sich das Ansprechverhalten des Motors zwischen dem Fahrzeug und dem festen Versuchsaufbau unterscheiden?
- Welche anderen Möglichkeiten gäbe es, den Treiberbaustein des Motors anzusprechen?
- Für welche Frequenz für das PWM-Signal haben Sie sich entschieden?
- ..