

# Embedded Systems Labor

## 2. Laboratory

Autor: Roland Lezuo, Thomas Gadner

Last change: 24. April 2025

Eine Laboreinheit besteht aus mehreren Aufgaben. Für jede vollständig bearbeitete Aufgabe gibt es Punkte, die am Ende zur Gesamtnote beitragen. Für Fragen während des Labors stehen dir vier Betreuer zur Verfügung.

### Task 1 Drehgeber - Background

Ein Drehgeber ist ein elektromechanisches Bauteil, das die Drehbewegung eines Objekts in digitale Signale umwandelt. Diese Signale können dann von einem Mikrocontroller oder einer anderen Steuereinheit verarbeitet werden, um die Drehposition oder -bewegung zu überwachen. Drehgeber werden in vielen Anwendungen eingesetzt, z. B. in Maschinensteuerungen, Motoren und Joysticks, die eine präzise Positionserfassung erfordern. Ein Drehgeber besteht typischerweise aus einem rotierenden Element, das eine bestimmte Anzahl von Impulsen oder Schritten pro Umdrehung erzeugt. Diese Impulse werden durch die Bewegung einer rotierenden Welle oder eines Rades erzeugt und dann in digitale Signale umgewandelt. Ein Inkrementalgeber erzeugt Signale in Form von Impulsen, wenn sich der Geber dreht. Diese Impulse werden in zwei Phasen (Spur A und B) erzeugt, aus deren Phasenlage zueinander die Drehrichtung ermittelt werden kann. Durch Zählen der Impulse kann die Drehposition ermittelt werden. Es werden keine absoluten Positionen gespeichert, so dass nach einem Stromausfall die Position "neu"bestimmt werden muss.

DSO-X 2014A, MY56272087: Wed Apr 23 21:25:44 2025



## Task 2 Laboraufgaben

### Task: A: Datenblatt lesen und Bauteil verstehen (10 Punkte)

Suche das Datenblatt für den Drehencoder mit der Bauteilbezeichnung **EC12E1220407**. Was tut dieses Bauteil (Output Wave), was bedeutet **CW** sowie **CCW** in diesem Kontext? Wie muss ein Treiber, der die *CW Ereignisse* für dieses Bauteil erkennt, aussehen (Hinweis: ein GPIO Interrupt wird für die einfachste Variante benötigt)?

### Task: B: Implementierung eines EC12E Treibers (20 Punkte)

Implementiere und teste deinen Treiber. Debugge mittels breakpoints an folgenden Punkten im Program

- Trigger GPIO IRQ (alten Code verwenden)
- Entscheidung ob CW Ereignis
- Erkennen des CW Ereignisses in der main-loop

### Task: C: Ansteuerung Motor (35 Punkte)

Im nächsten Schritt verstehe den Baustein *DRV8210PDSGR* um den Motor in der Betriebsart *forward* zu betreiben. Erzeuge ein PWM-Signal (für *IN1*) und erhöhe den Duty-Cycle mit *CW*-Ereignissen, die am Rotary Encoder auftreten. Demonstriere, dass nach dem Einschalten des Systems

- der Motor steht,
- mit jedem *CW*-Ereignis schneller dreht (bis eine maximale Drehzahl erreicht ist),
- messe *IN1* mit dem Oszilloskop.

### Task: D: Erweiterte Ansteuerung Motor (35 Punkte)

Erweitere den Rotary Encoder Treiber so, dass er auch *CCW* Ereignisse erkennt und reduziere den Duty-Cycle des Motors bei jedem Auftreten. Erzeuge auch ein PWM-Signal für *IN2* und betreibe den Motor in der Betriebsart *backward*. Demonstriere, dass nach dem Einschalten des Systems

- der Motor steht,
- mit jedem *CW* Ereignis schneller dreht (bis eine maximale Drehzahl erreicht ist),
- mit jedem *CCW* Ereignis langsamer dreht (bis Stillstand),
- bei weiteren *CCW* Ereignissen schneller rückwärts dreht (bis maximale Drehzahl erreicht),
- *CW* Ereignisse das rückwärts Drehen verlangsamen (bis Stillstand),

- messe *IN1* und *IN2* mit dem Oszilloskop.

### Task: E: usatzaufgabe (25 Punkte)

Implementierung einer robusten und skalierbaren Lösung mit kontinuierlicher Abfrage (Timer-Interrupt alle 1 ms) unter Verwendung des Gray-Codes. Achtung: Für die Dekodierung von Drehgebern mit fliegenden Rastpunkten müssen Pegeländerungen erkannt und ausgewertet werden.

## Report

Wenn Sie eine oder mehrere Aufgaben abgeschlossen haben, rufen Sie einen der Betreuer im Labor zu dir. Wir werden die Aufgabe überprüfen.