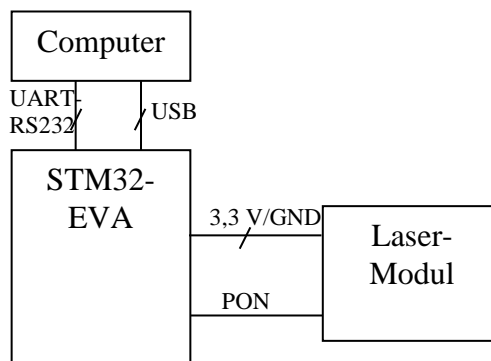


## Praktikum Mikrocontroller für Luft- und Raumfahrtanwendungen

Thema: Ansteuerung einer Laserdiode mithilfe einer Mikrocontrolleranwendung über die RS232-Schnittstelle des PCs

### Aufbau der Hardware



### Benötigte Baugruppen

- STM32F0-EVA
- Laser-Modul
- RS232-TTL-Converter

### Lernziele

- Analyse der Datenblätter und Überführung in einen Stromlaufplan
- Vertraut machen mit einem ST-Evaluationsboards (ARM CORTEX M0)
- Anwendung Interruptsteuerung und Schnittstellenkommunikation
- Umgang mit einem externen Modul
- ISP-Programmierung des STM32F0 mithilfe der STD-LIB

### Systembeschreibung

Mithilfe eines Lasermoduls soll die Funktionsweise einer Fernsteuerung durch den PC untersucht werden. Der 650nm-Laser ist über einen Schalttransistor mit dem Mikrocontroller an einem GPIO-Port verbunden. Der Schalttransistor ist notwendig, das die Maximalleistung von 5 mW nicht durch den  $\mu$ C getrieben werden darf. Erfolgt das Schaltsignal durch ein Terminalprogramm am PC (bspw. HTERM), wird der Laser aktiviert bzw. deaktiviert. Auch der Zustand der Laserdiode soll abfragbar sein.

### Aufgabenstellung

- Machen Sie sich mit der Funktionsweise des Lasermoduls vertraut.
- Zeichnen Sie einen Stromlaufplan Ihres Aufbaus.

- Verbinden Sie das Modul mit dem Evaluationsboard. Achten Sie darauf, die Versorgungsleitung PON an dem korrekten Pin (PA1) anzuschließen und ebenfalls 3,3V und GND mit den entsprechenden Pins zu verbinden. Nutzen Sie für die Kommunikation mit dem Board den RS232-Converter, schließen Sie jedoch keinesfalls die 5 V-Leitung an das Evaluationsboard an! Nutzen Sie für das USART-Peripheriemodul USART2 an Pins (PA2, PA3).
- Schreiben Sie ein Programm, welches beim Empfang des Befehls "L1" ("0x4C 0x31") das Lasermodul und gleichzeitig auch die LED LD4 (PC8) aktiviert. Empfängt das Board hingegen den Steuerbefehl "L0" ("0x4C 0x30"), so sind die beiden Ausgangspins zu deaktivieren. Ihr Board antwortet mit dem gleichen Befehl, welches es empfangen hat.
- Senden Sie am PC das Steuersymbol "L?" ("0x4C 0x3F"), so soll ihr µC-Programm den aktuellen Zustand des Ausgangs über UART liefern.
- Erweitern Sie Ihr Programm um eine DMA-Schnittstelle („direct-memory-access“), und nutzen Sie für Ihre Programmablaufsteuerung ein Interrupt basierend auf erfolgreichem DMA-Datentransfer!
- **Hinweise: Beachten Sie das mitgelieferte Dokument zum Laserschutz!**

*Benutzen Sie die folgende Konfiguration für Ihre Schnittstelle, sowohl im Quellcode als auch in HTERM:*

*Baudrate: 19200*

*8 Bit Wortlänge, 1 Stoppbit, keine Parität*

## **Protokollierung**

Stellen Sie Ihre Ergebnisse im Rahmen eines Protokolls dar. Eine Gliederung des Protokolls könnte wie folgend aussehen:

1. Einleitung und Beschreibung der Aufgabenstellung
2. Stromlaufplan zur Aufgabenstellung (Skizziert oder bspw. mit EAGLE erstellt)
3. Beschreibung der implementierten Funktionen und Schnittstellen
4. Zusammenfassung und Bewertung der erzielten Ergebnisse, Verbesserungsvorschläge
5. Quellenangaben
6. Anhang: Kommentierter Quelltext