LVAs 384.996 & 384.174
'Mikrocomputer LU'
'Mikrocomputer für Informatiker_innen'

Logan

v0.4

2025

Inhaltsverzeichnis

Hinweis zur Laborübung:
Bitte machen Sie sich *vor der Übung* mit folgenden Dokumenten vertraut:

- Diese Angabe (`Logan.pdf`).
- Die entsprechenden Abschnitte der im TISS hochgeladenen Unterlagen zum verwendeten NUCLEO-Board, konkret:
  - RCC
  - GPIO (inkl. der 'Alternate Functions')
  - SPI*
  - USART*
  - TIMER*
  - Interrupts (von Peripherieeinheiten)
  - EXTI* (inkl. SYSCFG)
  - ADC

  *) Überlegen Sie, welche der gekennzeichneten Peripherieeinheiten (zB welcher Timer, welche SPI) für die Anwendung in Frage kommen (das ergibt sich auch aus der Pinbelegung, siehe Tabelle 1).

Bei auftretenden Fragen während Ihrer Vorbereitungen wenden Sie sich *vor* Ihrem Übungstermin an die Tutoren.

1 Allgemeines

Ziel dieser Aufgabe ist es, den Umgang mit externer Peripherie zu erlernen. Der Aufbau besteht aus fünf Linearantrieben, welche Murmeln auf einem Spielfeld positionieren können. Die Linearantriebe sind über einen zentralisierten Seilzug miteinander verbunden, welcher durch zwei Schrittmotoren angetrieben wird. Dabei werden die Positionen der fünf Aktuatoren durch den Duty-Cycle von je einem PWM-Signal (125Hz) übermittelt. Die Steuerung erfolgt über SPI-Befehle und einem Step-Signal für die Motoren.

1.1 Laboraufbau

Abbildung 1 zeigt Logan. Der gesamte Laboraufbau setzt sich aus folgenden Komponenten zusammen:

- STM32F34R8-Mikrocontrollerboard auf einer Adapterplatine
- Logan

1.2 Inbetriebnahme

Schalten Sie zuerst Logan ein. Nützen Sie dazu den Schalter der angeschlossenen Verteilersteckdose. Logan ist so konfiguriert, dass nach dem Einschalten eine Grundposition angefahren wird. Schließen Sie anschließend den Mikrocontroller am PC an und beginnen Sie mit der Bearbeitung Ihrer Aufgabe.

ACHTUNG: Während der Startsequenz reagiert Logan auf keine Eingänge.

1.3 Logic Analyzer

Es sind einige relevante digitale Pins an einem fix verbauten 8-Kanal Logic Analyzer angeschlossen. Mit dessen Hilfe kann das zeitliche Ein-/ Ausgangsverhalten der Pins überprüft und zur Fehlerbehebung genutzt werden. Eine kurze Anleitung zum Umgang mit dem Logic Analyzer wird es zu Beginn des Labors geben, außerdem finden Sie eine ausführliche Dokumentation in den hochgeladenen Unterlagen. Die Zuteilung der angeschlossenen Pins zu den acht Kanälen (LA CH0 .. CH7) ist in den Folgekapiteln und in Tabelle 1 ersichtlich.

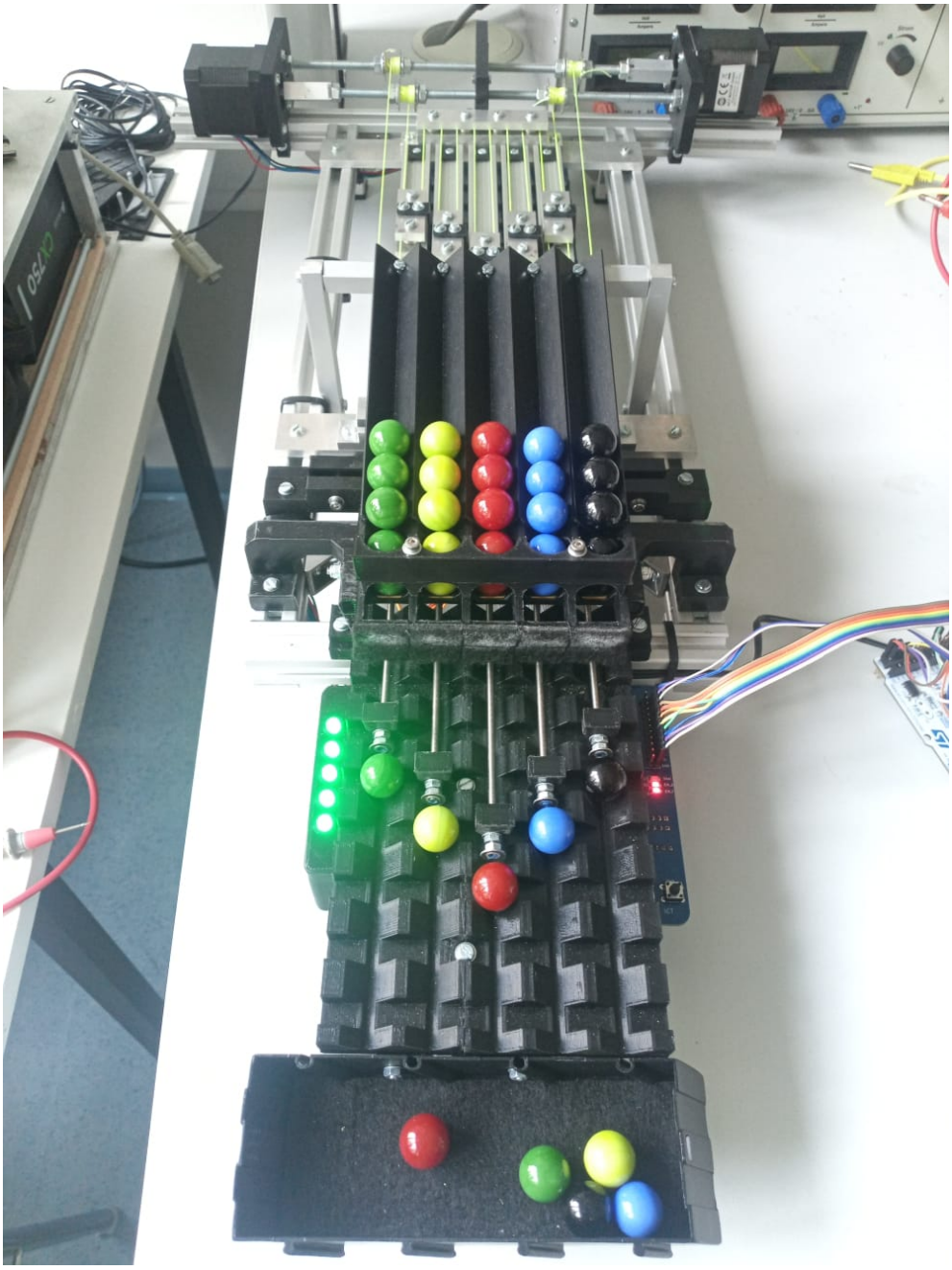

Abbildung 1: Laboraufbau

2 Hardware

Der Versuchsaufbau besteht aus fünf Linearantriebe, welche miteinander über einen Seilzug verbunden sind. Diese können Murmeln von einem Magazin hohlen und auf der Spielfläche positionieren.

2.1 Antriebsstrang

Der Antriebsstrang besteht aus zwei Schrittmotoren und fünf Sperren. Abbildung 2 zeigt den Seilzug, welcher die Linearantriebe bewegt. Dabei zieht jeweils ein Schrittmotor am Seil für den Back Antrieb und Front Antrieb. Die Steuerung vom Versuchsaufbau sorgt dafür, dass immer nur maximal ein Motor aktiv ist und der Zweite auf Freilauf geschaltet wird. Dadurch werden die Linearantriebe in Bewegung versetzt. Um gezielt einen Linearantrieb steuern zu können, sollten alle Anderen mittels der Sperren blockiert werden. Für die Ansteuerung siehe Kapitel 3.1.1.

2.2 Pinbelegung

Der Mikrocontroller ist über eine SPI-Schnittstelle und sechs Datenleitungen mit Logan verbunden. Zusätzlich ist ein Potentiometer am STM32 angebracht. Die genaue Pinbelegung finden Sie in Tabelle 1. Die genannten SPI-Pins müssen ihrer Verwendung entsprechend in den dafür vorgesehenen GPIO- und SPI-Registern konfiguriert werden. Zusätzlich muss der Pin PB4 als logischer Ausgang, Pin PC0, PC2, PA1, PB10 und PB14 als Eingang und PA7 als analoger Eingang für den ADC konfiguriert werden. Die Konfiguration der UART-Schnittstelle ist frei wählbar, solange Sie im Terminalprogramm und auf dem Mikrocontroller identisch konfiguriert ist.

2.3 ADC

Mit Hilfe des an PA7 angeschlossenen Potentiometers (Spannungsbereich VDD - GND) soll die Geschwindigkeit der Linearantriebe beeinflusst werden.

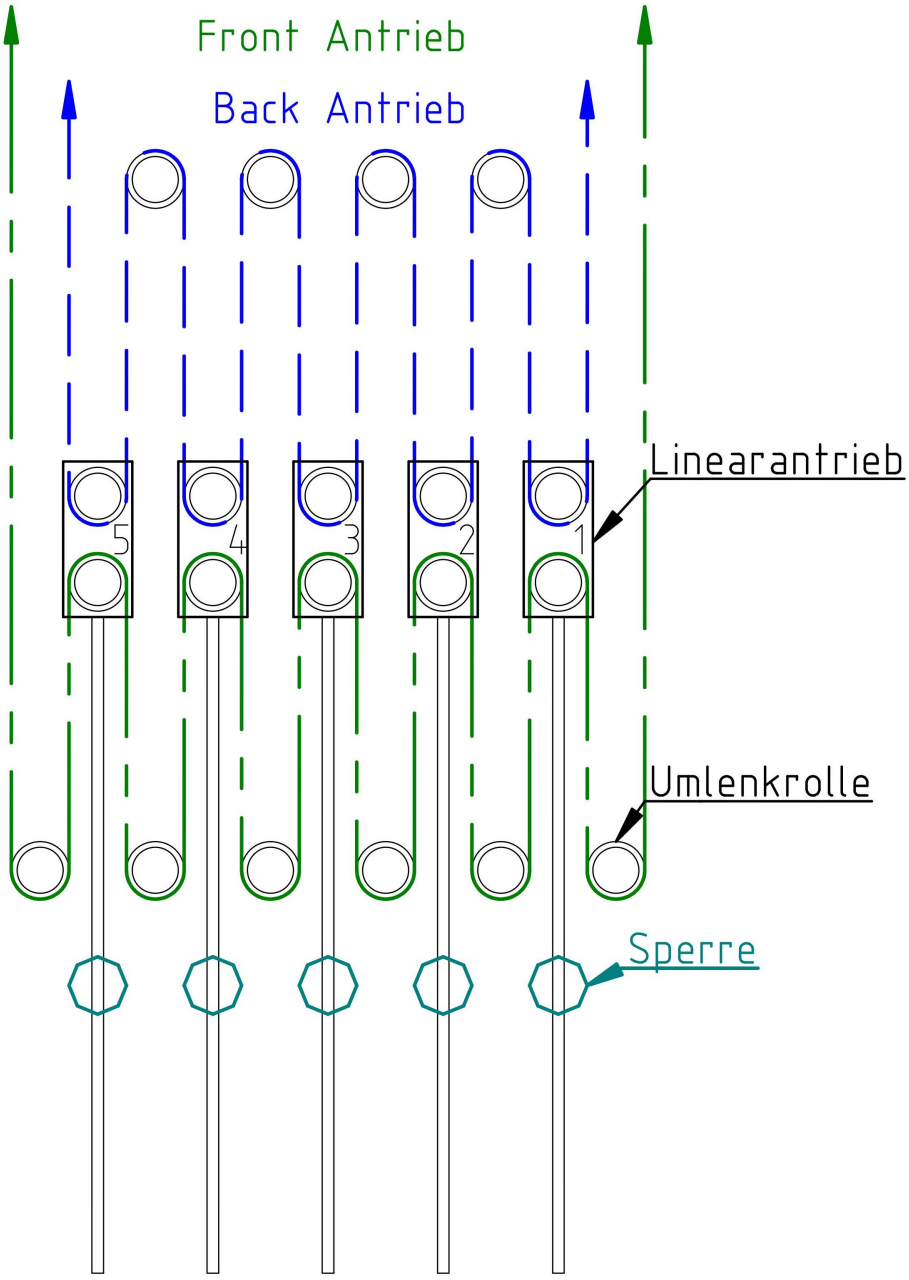

Abbildung 2: Antriebsstrang Logan

STM32-Pin	Beschreibung	Funktionseinheit	LA CHx
PB3	SCLK		LA CH2
PA6	MISO	SPI-	LA CH4
PB5	MOSI	Kommunikation	LA CH3
PA15	NSS		LA CH1
PB4	Step	Step Signal	LA CH5
PC0	Sig1		LA CH0
PC2	Sig2		
PA1	Sig3	Position PWM	
PB10	Sig4		
PB14	Sig5		
PA7	ADC Eingang	Potentiometer	
PA2	TX	UART-	
PA3	RX	Kommunikation	

Tabelle 1: Pinbelegung des STM32F334R8 für den Versuchsaufbau Logan

3 Software

Die Kommunikation mit Logan erfolgt über die SPI-Schnittstelle. Diese besteht aus den vier SPI spezifischen Pins (MOSI, MISO, NSS und SCLK). Zusätzlich muss ein Step-Signal für die Schrittmotoren angelegt werden, welches durch jeden angelegten Puls den Motor ein kleines Stück weiter bewegt. Dieses darf maximal eine Frequenz von 5kHz haben (dies ist die Maximalgeschwindigkeit vom Antrieb). Als Eingangssignale für den STM32 sind fünf asynchrone PWM-Signale der Positionen sowie das Signal des externen Potentiometers verfügbar.

3.1 SPI Kommunikation

Die SPI Schnittstelle muss wie folgt konfiguriert werden:

- Mode: Master Mode Two-lines Full-Duplex
- Datengröße: 8 Bit
- Endian: MSB first
- Baudrate: 1 MBaud

- Idle clock line ist LOW
- Stabile Daten sollen bei steigender Flanke der clock line anliegen
- Während der Übertragung muss das NSS Signal auf LOW liegen

3.1.1 Kontrollbyte

Die Sperren und Bewegungsrichtung werden mittels einem Kontrollbyte, welches über die SPI-Schnittstelle gesendet wird, gesteuert. Dieses soll maximal nur alle 4ms gesendet werden. In Tabelle 2 ist die Zusammensetzung dargestellt. Dabei werden beim Aktivieren vom Freilauf beide Antriebe deaktiviert. Es macht Sinn immer nur eine Sperre freizugeben, aber es ist auch möglich mehrere Linearantriebe gleichzeitig zu aktivieren.

bit 7							bit 0
X	Frw	Dir	Sp5	Sp4	Sp3	Sp2	Sp1

bit 7	**X:**	Nicht verwendet
bit 6	**Frw:**	Freilauf(Freewheel)
	1=	Freilauf an
	0=	Freilauf aus
bit 5	**Dir:**	Richtung(Direction)
	1=	Nach Vorne(Front)
	0=	Nach Hinten(Back)
bit 4-0	**Sp:**	Sperre Linearantrieb 5-1
	1=	gesperrt
	0=	freigegeben

Tabelle 2: Kontrollbyte

3.1.2 Position PWM

Logan kommuniziert die Positionen der fünf Linearantriebe über jeweils ein PWM-Signal. Dabei ist zu beachten, dass die Signale nicht miteinander synchronisiert sind und dadurch jedes bei einem anderen Zeitpunkt starten kann. Jedes PWM-Signal hat eine Frequenz von 125Hz und eine Duty-Cycle Auflösung von 5 bit (0 = 0%, 31 = 100%). In Abbildung 3 ist die Positionsverteilung zu sehen. Das PWM-Signal stellt die Bereiche 1 bis 17 da (grün/blau)

und die Positionen auf dem Spielfeld (rot) entsprechen der Übergänge zwischen dieser Bereiche. Der verwendete Duty-Cycle geht von 2 bis 18 (Duty-Cycle 2 = Bereich 1), da 1 für einen Errorzustand der Hardware vorbehalten ist. Position 1 bewirkt, dass eine Murmel auf das Spielfeld geladen wird. Das Erreichen von Position 16 befördert Diese in die Auffangbox.

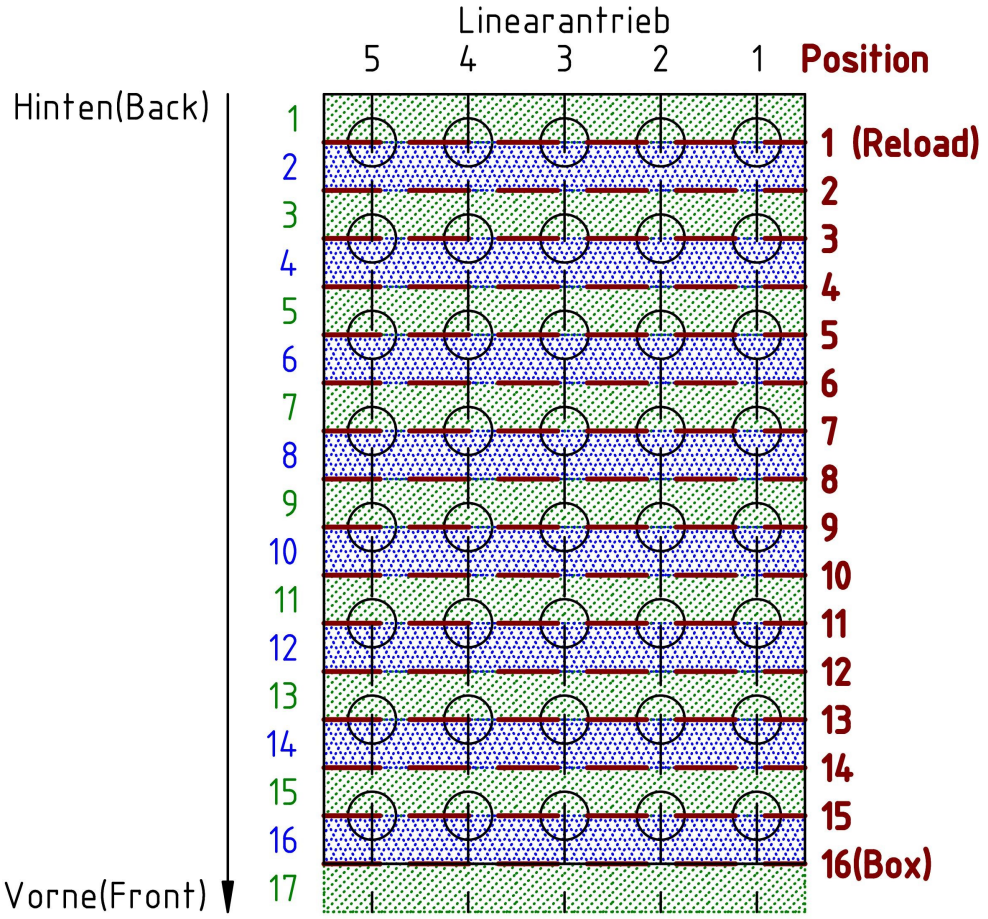

Abbildung 3: Positionsverteilung

3.1.3 Schutzfunktionen und LEDs

Der Versuchsaufbau Logan verfügt über einige Schutzfunktionen, welche eine Beschädigung verhindern. Auf der rechten Seite sind drei rote LEDs und ein Taster zu finden. Die LED EN_F und EN_B signalisieren, welcher Antrieb deaktiviert ist(On = Antrieb Off). Die Stat LED leuchtet, wenn ein Kontrollbyte automatisch abgeändert wurde. Ein langsames Blinken bedeutet, dass der Aufbau gerade eine Demofahrt macht und ein Schnelles, dass der Softstop aktiv ist. Mit dem Taster SW1 wird durch kurzes Drücken (<500ms) der Softstop aktiviert. Dabei werden alle Sperren freigegeben und alle Signale vom STM32 ignoriert. Um wieder in den Normalbetrieb zu gelangen, muss die Taste lange gedrückt werden (>500ms). Zusätzlich können Demofahrten mit einem langen Drücken aktiviert werden. Dabei fährt der Versuchsaufbau selbstständig eingespeicherte Positionen an. Auf der linken Seite sind grüne Signal LEDs, welche den Zustand der Sperren darstellen(On = Sperre freigegeben). Die Nummerierung geht dabei von Hinten nach Vorne. Diese grünen Signal LEDs reagieren nicht sofort auf Änderungen.

- Die Schrittmotoren werden nur freigegeben wenn ein Linearantrieb sich auch mechanisch bewegen kann.(Entsperrt und nicht auf Endanschlag)
- Der Antrieb wird unterbrochen, wenn eine Sperre in Bewegung ist.
- Wenn eine Murmel vom Magazin geholt wurde, muss mindestens der Bereich 4 erreicht werden, bevor sich wieder nach hinten bewegt werden darf. (Unerlaubte Bewegungen werden automatisch gesperrt durch eine Abänderung des empfangenen Kontrollbytes)

4 *Empfohlener* Übungsablauf

Bauen Sie Ihr Programm modular auf! Implementieren und testen Sie die Teilaufgaben so weit wie möglich separat und führen Sie diese erst dann zur Gesamtlösung zusammen. Ein *Vorschlag* für die Herangehensweise und Separierung der Teilaufgaben:

1. Implementieren Sie die SPI-Verbindung zwischen dem STM32 und dem Aufbau. Testen Sie die korrekte Funktion der SPI-Schnittstelle zunächst, indem Sie die Sperren ansteuern. Es empfiehlt sich, parallel dazu eine USART-Verbindung mit dem PC herzustellen. So können Sie mittels des Terminal-Programms Befehle an den STM32 senden, der Diese dann über die SPI-Schnittstelle an den Aufbau senden kann.

2. Legen Sie ein PWM-Signal an den Step-Pin um die Motoren zu bewegen.

3. Fahren Sie mittels den Sperren und der Richtungsteuerung zwischen den Endstopps.

4. Implementieren Sie nun das Einlesen von einem Positions PWM-Signal und testen dies mit der Demofahrt oder mittels der Fahrt zwischen den Endstopps.

5. Konfigurieren Sie den ADC und verwenden Sie die eingelesenen Daten des Potentiometers (z.B. Geschwindigkeit des Antriebs).

6. Steuern Sie einen Linearantrieb, um gezielt Positionen zu erreichen.

7. Implementieren Sie eine Logik um mit allen Linearantrieben Positionen anfahren zu können.

8. Erweitern Sie Ihren Algorithmus, um mit den Murmel etwas darstellen zu können.

5 Benotung

Die Note für das Labor setzt sich aus dem Abgabegespräch sowie den Funktionalitäten, die Sie implementiert haben, zusammen. Im Folgenden befindet sich ein Richtwert, welche Funktionalitäten zum Erreichen einer bestimmten Note erfolgreich implementiert werden müssen. Dabei ist die Erfüllung aller Minimalanforderungen für die 'schlechteren' Noten die Voraussetzung für eine 'bessere' Note. Die Gesamtnote hängt jedoch zusätzlich von dem Abgabegespräch ab, d.h. wie gut Sie den Code erklären können und ob Sie in der Lage sind, kleine Änderungen vorzunehmen.
Die gesamte Steuerung soll über Interrupts erfolgen, wenn Sie glauben, an irgendeiner Stelle eine Warteschleife zu benötigen, fragen Sie bei den Tutoren nach, ob Sie das dürfen. Unnötige Warteschleifen führen zu Punkteabzügen.

Genügend:	Die Sperren können angesteuert werden und die Linearantriebe fahren in beide Richtungen bis zu den Endstopps (keine Positionsregelung benötigt).
Befriedigend:	Das Potentiometer wird für die Geschwindigkeitsregelung verwendet und es kann ein einziger Linearantrieb gezielt Positionen anfahren (Position Input über UART). Der Duty-Cycle von einem Linearantrieb soll über UART ausgegeben werden.
Gut:	Es können mit jedem Linearantrieb gezielt Positionen angefahren werden (Position Input über UART).
Sehr gut:	Es können Symbole mittels der Murmeln am Spielfeld dargestellt werden (z.B. Buchstaben/Zahlen). Das Symbol kann dabei über UART ausgewählt werden und die Murmeln werden danach automatisch platziert.