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

Christie

v1.2 - $\frac{3}{3}$

2024

Inhaltsverzeichnis

1 Allgemeines **2**
- 1.1 Laboraufbau . . . 3
- 1.2 Inbetriebnahme . . . 4
- 1.3 Logic Analyzer . . . 4

2 Hardware **5**
- 2.1 Mechanischer Aufbau . . . 5
- 2.2 Ansteuerung der Christie . . . 8
  - 2.2.1 Steppermotoren . . . 8
  - 2.2.2 Servomotoren . . . 10
- 2.3 Ränder des Displays - Errors . . . 10
  - 2.3.1 Soft-Error . . . 11
  - 2.3.2 Hard-Error . . . 11
  - 2.3.3 Randerkennung . . . 12
- 2.4 STM32 Pinbelegung . . . 12

3 *Empfohlener* Übungsablauf **13**
- 3.1 Aufbau des Programms . . . 13
- 3.2 Benotung . . . 13

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

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

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 der Christie besteht aus einem druckempfindlichen LCD Zeichenbrett und einem mechanischen Aufbau, welcher den Schreibkopf in x- und y-Richtung bewegen kann. Am Schreibkopf befindet sich eine mit einem Servomotor heb- und senkbare Halterung in welcher der eigentliche Zeichenstift eingespannt ist.
Ein weiterer Servomotor wird verwendet, um das Zeichenbrett zu löschen (durch mechanische Betätigung der Löschtaste).

1.1 Laboraufbau

Der Übungsaufbau setzt sich aus folgenden Komponenten zusammen:

- STM32F334R8-Mikrocontrollerboard mit angeschlossenem Logic Analyzer
- Christie

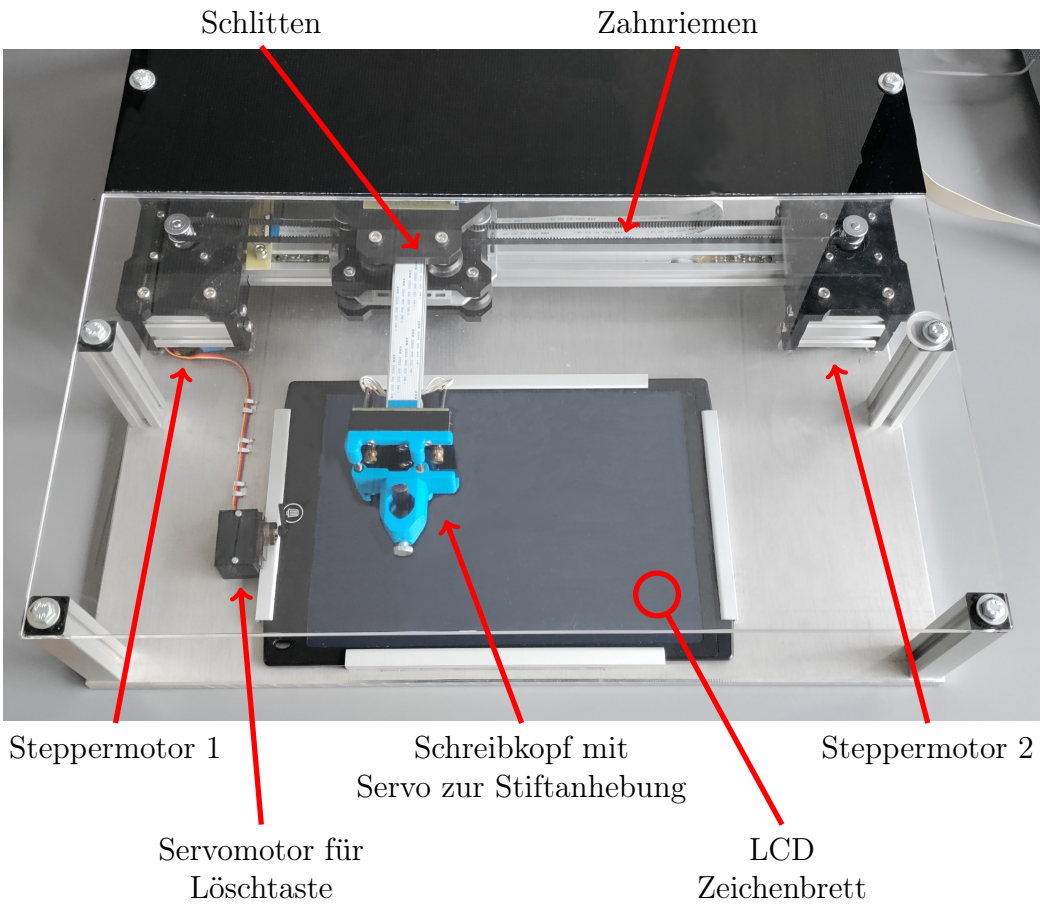

Abbildung 1: Übungsaufbau Christie

1.2 Inbetriebnahme

Schalten Sie zuerst die Christie mittels Schalter an der Steckdose ein und schließen Sie das NUCLEO-Board an.

Beachten Sie: Der STM32 der Christie und das NUCLEO-Board können sich über die Steuerleitungen teilweise gegenseitig mit Spannung versorgen und dadurch zu undefiniertem Verhalten führen. Nur wenn beide mit Spannung versorgt werden, ist ein fehlerloser Betrieb möglich.

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 LA-Kanälen ist in den Folgekapiteln und in Tabelle 4 ersichtlich.

2 Hardware

2.1 Mechanischer Aufbau

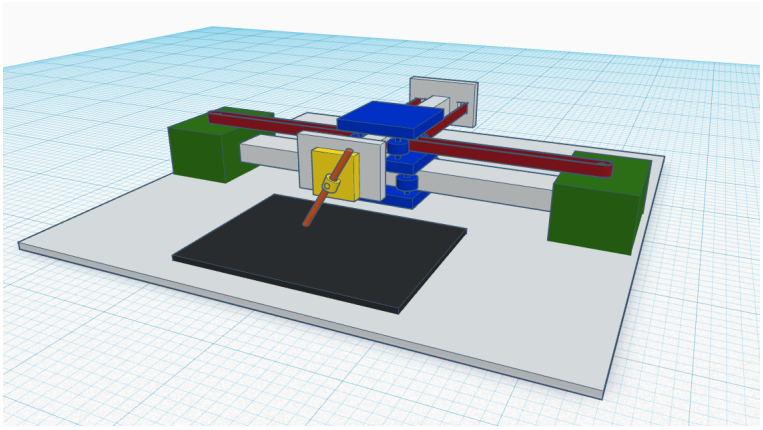

Abbildung 2: Christie - Schematische Darstellung das Aufbaus

Der Schreibkopf ist über zwei Schlitten in (x/y)-Richtung bewegbar. Er wird über zwei Steppermotoren und einen Zahnriemen angetrieben, dabei sind die Enden des Zahnriemens am Schreibkopf fixiert. Ein Schema des mechanischen Aufbaus finden Sie in Abbildung 3.

Die Bewegung der Anordnung kann als Superposition der zwei Motorbewegungen betrachtet werden: Steht zum Beispiel der linke Motor 1 still, so kann sich die Strecke (a+b) in der Länge nicht verändern. Der Motor 2 alleine kann also die Scheibkopfposition nur entlang einer diagonalen Linie in (+x/+y)-Richtung bewegen. In Abbildung 4 wird dies veranschaulicht (grüne Strecken werden länger, rote Strecken werden kürzer).
Die Bewegung, welche durch Motor 1 ausgelöst wird steht nach dem selben Prinzip normal zu der von Motor 2.

Für eine Bewegung **parallel** zu den kartesischen Koordinatenachsen (dh nur in x- oder nur in y-Richtung) müssen beide Stepper gleichzeitig und gleich schnell drehen.

Beachten Sie: Wenn nur ein Motor bewegt wird, muss der andere Motor

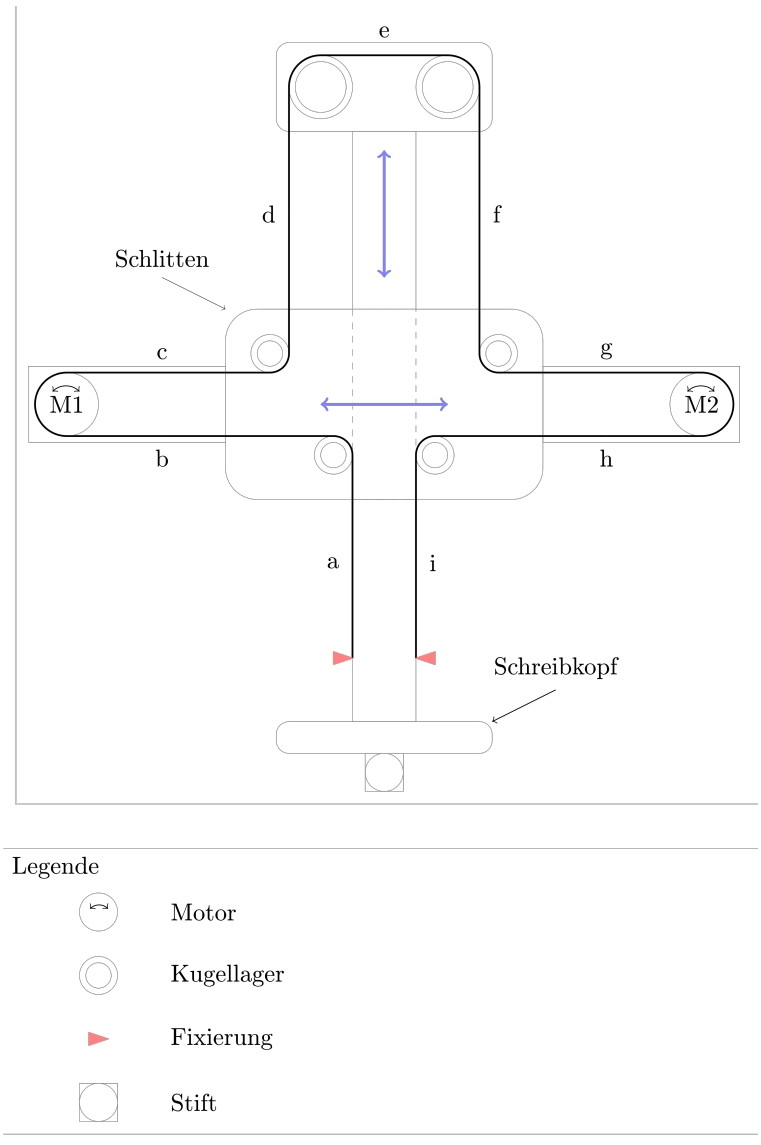

Abbildung 3: Christie - Mechanischer Aufbau (Schema)

dennoch bestromt werden, um dessen Achse festzuhalten. Andernfalls könnte die Achse frei mitdrehen und die Bewegung somit verzerren.

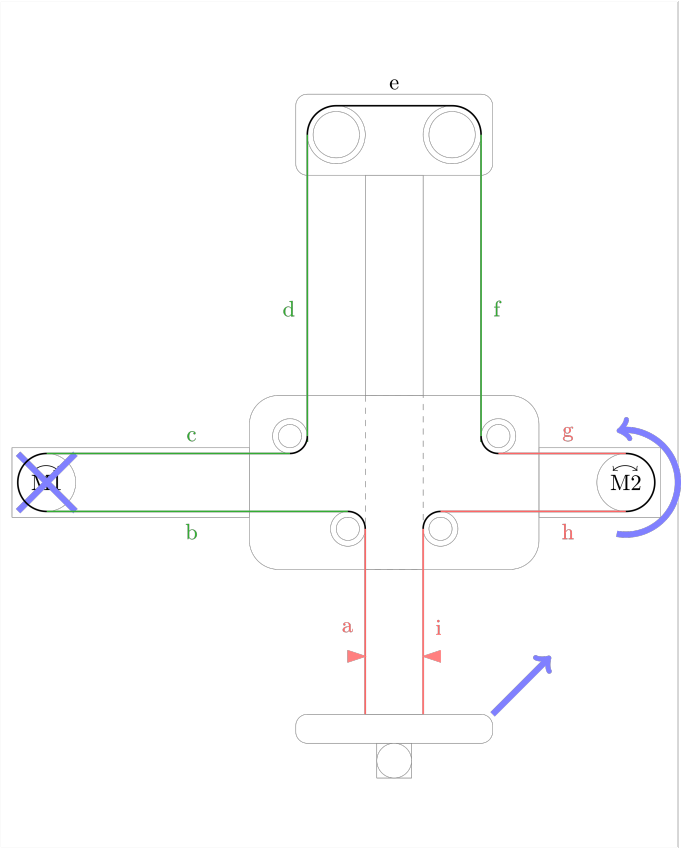

Abbildung 4: Christie - Verhältnisse, wenn Motor 2 bewegt wird

2.2 Ansteuerung der Christie

2.2.1 Steppermotoren

Beachten Sie: Sollten Sie nicht mit Steppermotoren vertraut sein, ist es empfehlenswert, sich vor der Laborübung in die Funktionsweise und Ansteuerung von Steppermotoren einzulesen.

Die beiden Steppermotoren M1 und M2 haben je zwei Spulen (L1 und L2), die in richtiger Abfolge der Polarität bestromt werden müssen, um eine Drehung zu bewirken. Über die in der Christie verbaute H-Brücke kann in jeder der beiden Spulen die Stromrichtung bestimmt werden, indem pro Spule einer der beiden GPIOs (A oder B) auf HIGH gesetzt wird. In Abbildung 5 sehen Sie die möglichen Positionen eines Steppermotors und die elektrischen Anschlüsse. Die H-Brücken sorgen auch dafür, dass der Logikpegel des STM32 auf die für die Motoren erforderliche Spannung von 12V umgesetzt wird.

Wichtig: Die Zustände der Stepper Pins werden von der Christie erst durch eine steigende oder fallende Flanke am Strobe-Pin übernommen!

Hinweis: Die maximal erreichbare Geschwindigkeit sind 250 Vollschritte oder 600 Halbschritte pro Sekunde. Bei höheren Frequenzen könnten Schritte übersprungen werden ('verloren gehen').

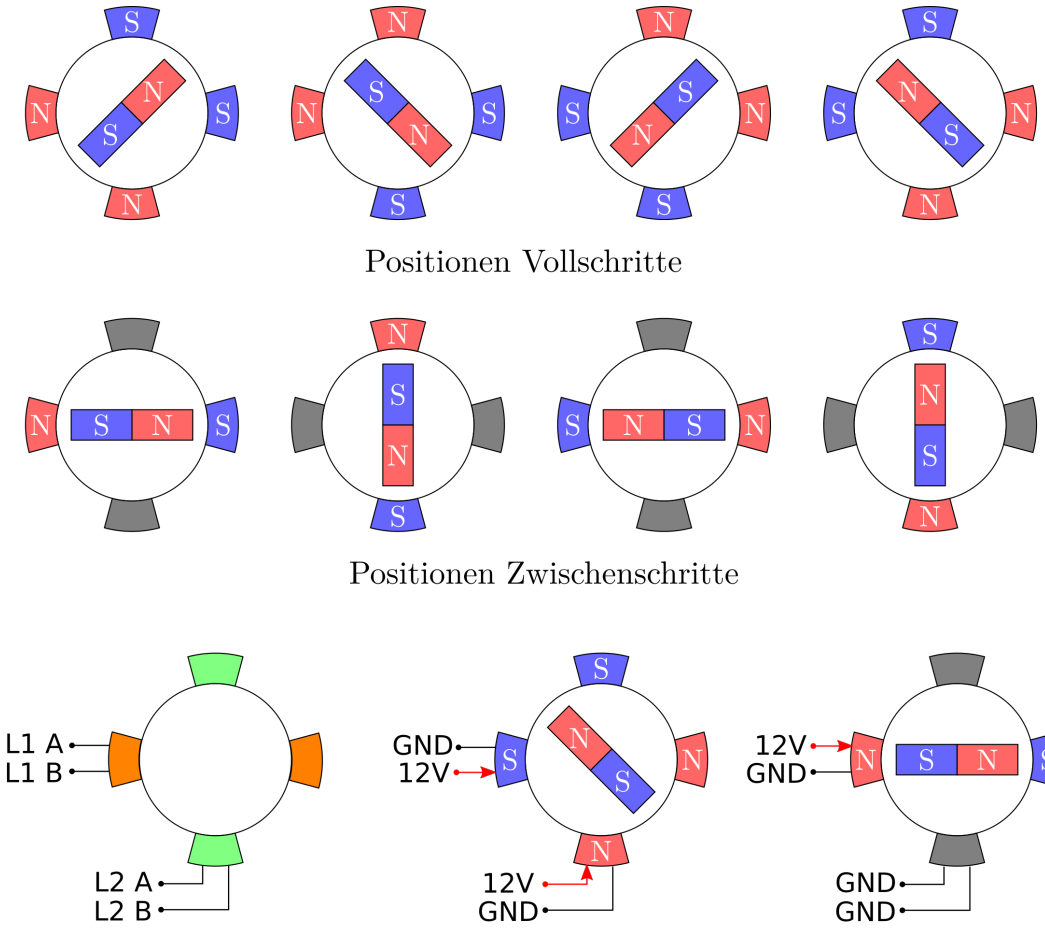

Abbildung 5: Steppermotoren: Positionen und elektrische Ansteuerung

2.2.2 Servomotoren

Die Servomotoren der Christie werden über den Logikpegel an den Pins PEN und DEL gesteuert und werden ebenfalls erst bei einer steigenden oder fallenden Strobe-Flanke übernommen.
Solange sich einer der beiden Servos noch aktiv in Bewegung befindet, wird der BUSY-Pin auf HIGH gesetzt und weitere Änderungen werden von der Christie ignoriert.
Beachten Sie: In diesem Fall wird das BUSY-Signal erst ca. 0,1ms nach der Strobe-Flanke auf HIGH gesetzt.

PEN	DEL	Funktion
	0	Löschtaste nicht drücken
	1	Löschtaste drücken
0		Stift auf Zeichenbrett absenken
1		Stift anheben

Tabelle 1: Ansteuerung Servomotoren

2.3 Ränder des Displays - Errors

Christie hat Sensoren auf dem mechanischen Aufbau um zu verhindern, dass der Stift das Zeichenbrett verlässt oder sogar beschädigt. Für jeden Rand des Displays gibt es zwei Sensoren, jeweils einen für den Soft-Error (innere Begrenzung) und einen für den Hard-Error (äußere Begrenzung):

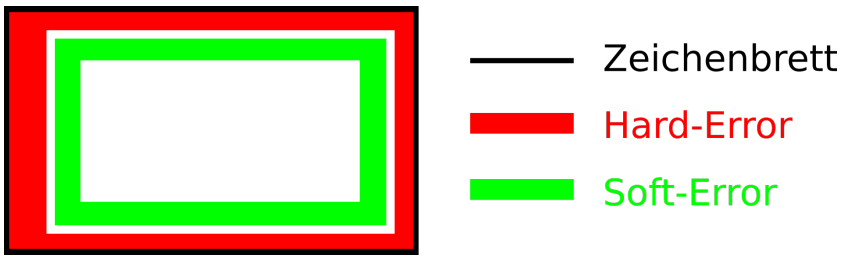

Abbildung 6: Christie-Zeichenbereich

Durch insgesamt vier Error-Pins wird die jeweilige Situation angezeigt (siehe Tabelle 2 und Tabelle 3).

ERROR	ERR_H_S	Stiftposition
0	0	im Zeichenbereich
0	1	im Zeichenbereich
1	0	im Soft-Errorbereich
1	1	im Hard-Errorbereich

Tabelle 2: Pinzustand bei Soft- oder Hard-Error

2.3.1 Soft-Error

Beim Soft-Error wird nur angezeigt, dass der Stift sich dem Rand des Zeichenbretts nähert. Das geschieht einerseits durch Setzen der jeweiligen Error-Pins, zusätzlich wird die Farbe der Beleuchtung von Grün auf Blau geändert.

Der Soft-Error ist rein indikativ und verursacht keinen Eingriff in die Ansteuerung der Motoren.

2.3.2 Hard-Error

Bewegt man sich trotz Soft-Errors weiter in Richtung des Randes, erfolgt ein Hard-Error: Die Error-Pins werden entsprechend gesetzt und Beleuchtung schaltet auf Rot um.

Zusätzlich wird der Zugriff auf die Motoren unterbunden und der Scheibkopf bewegt sich selbstständig aus dem Hard-Error wieder in den Soft-Error Bereich.

Beachten Sie: Durch das automatische Bewegen des Schreibkopfes beim Hard-Error wird eine Diskrepanz zwischen der realen Position und jener Position, die Ihr Programm annimmt, auftreten und eine Re-Initialisierung erforderlich machen.

2.3.3 Randerkennung

Über den Zustand der ERR_C1 und ERR_C2 Pins kann bestimmt werden, an welchem Rand der Schreibkopf in den Errorbereich ragt.

ERR_C1	ERR_C2	Rand
0	0	oberer Rand
0	1	rechter Rand
1	0	linker Rand
1	1	unterer Rand

Tabelle 3: Randerkennung

2.4 STM32 Pinbelegung

Signal	Pin STM32	Beschreibung	Logic Analyzer
Stepper 1 L1 A	PB1	Motor 1 - Spule 1	LA CH3
Stepper 1 L1 B	PB2	Motor 1 - Spule 1	LA CH4
Stepper 1 L2 A	PA10	Motor 1 - Spule 2	LA CH5
Stepper 1 L2 B	PB3	Motor 1 - Spule 2	LA CH6
Stepper 2 L1 A	PB5	Motor 2 - Spule 1	
Stepper 2 L1 B	PB4	Motor 2 - Spule 1	
Stepper 2 L2 A	PB10	Motor 2 - Spule 2	
Stepper 2 L2 B	PA8	Motor 2 - Spule 2	
Strobe	PB6	Flanke zur Übernahme der Daten	LA CH1
PEN	PA9	Stiftposition	
DEL	PC7	Löschtaste	
BUSY	PA11	Busy-Signal	LA CH2
ERROR	PA0	Error-Signal	
ERR_H_S	PA1	Hard- bzw. Soft-Error	
ERR_C1	PB0	Bit zur Randerkennung	
ERR_C2	PA12	Bit zur Randerkennung	
ADC	PA7	Potentiometer "X"	

Tabelle 4: Pinbelegung

Tabelle 4 zeigt die Pinbelegung aus der Sicht des STM32.

3 *Empfohlener* Übungsablauf

3.1 Aufbau des Programms

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:

- Bewegen des Schreibkopfes in kartesischen Koordinaten
- Auslesen der vier Error-Pins, um die Ränder und somit die Nullposition zu finden
- Bewegung des Schreibkopfes auf eine beliebige Position
- Zeichnen von einfachen Formen
- Auslesen der UART zum PC
- Sieben-Segment-Logik
- Auslesen des ADCs und Algorithmus zur Rauschunterdrückung

3.2 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: Orientierung auf dem Zeichenbrett:
Ihr Programm muss nach einem Reset zunächst die Anzeige löschen, die Mitte finden und diese dann mit einem Symbol (z.B. Kreuz) kennzeichnen.

Befriedigend: Zeichnen:
Es kann eine konkrete Form (z.B. Kreis, Sinusfunktion) an beliebiger, im Code definierter Stelle, korrekt geplottet werden.

Gut: Sieben-Segment-Uhr:
Ihr Programm kann eine per UART vom PC empfangene Uhrzeit vierstellig darstellen (z.B. in einer üblichen Sieben-Segment-Darstellung bestehend aus linienförmigen Segmenten).

Sehr gut: Zeitzoneneinstellung:
Ihr Programm kann zusätzlich die per UART übertragene Uhrzeit um eine Verschiebung von +/- drei ganzen Stunden anpassen. Bei Drehung am Potentiometer "X" soll die angezeigte Zeit gelöscht und die neue Uhrzeit angezeigt werden. Das ADC-Rauschen wird durch geeignete Maßnahmen (z.B. Mittelung) ausgefiltert.