

## Anforderungsanalyse "Hau den Lukas"

Guilherme Bufolo

Martin Oettinger

Dirk Pfeiffer

12.05.2009

# Inhaltsverzeichnis

- ① Aufgabenstellung
  - Übersicht
  - Konkrete Funktionen
- ② Hardware und Randbedingungen
  - Hardware
  - Randbedingungen
  - Nicht funktionale Eigenschaften
- ③ Funktionsbeschreibung
  - Pendeln
  - Kontinuierliches Heben
  - Schrittweises Heben
  - Gebremstes Fallenlassen
  - Schrittweises Fallenlassen
- ④ Einschränkungen und Annahmen
- ⑤ Fehlersituationen und Ideen

## Umfeld und Aufgabe

- im Rahmen der Veranstaltung “Echtzeitsystemlabor”
- Aufgabenstellung: Implementierung einer Steuersoftware
- Ziel: Eisenprojektil durch gezieltes an-/abschalten der Spulen kontrolliert anheben/fallen lassen

## Konkrete Funktionen

- Eisenkern
  - kontinuierlich/schrittweise anheben/fallen lassen
  - gebremst fallen lassen
  - pendeln lassen
- “Not-Aus-Funktion”
  - sofort alle Spulen abschalten
  - evtl. Bremsen des Kern bei unterster Spule

## Die “Hau den Lukas”-Hardware

- Plexiglasröhre
- 8 Spulen, 8 Lichtschranken
- Microcontroller (Infineon TriCore)



## Randbedingungen

- programmierbarer Programmablauf
- maximal eine Spule gleichzeitig aktiviert
- keine Spule länger als 4 Sekunden aktiviert lassen

## Nicht funktionale Eigenschaften

- System ist ereignisgesteuert
- Programm übernimmt Ausnahmeerkennung und -behandlung
- System verhält sich fail-safe im Ausnahmefall
- System versucht im Fehlerfall den Eisenkern ohne Hardwäreschäden in Ausgangsposition zu bringen

## Funktion “Pendeln”

- Eisenprojektil wird an *einer* Spule auf- und abbewegt

## Überlegungen

- möglicherweise nicht mit unterster/oberer Spule möglich
- jeweilige Lichtschranke wird durch Pendeln periodisch aktiviert/deaktiviert

## Vorgang

- Eisenkern mittig in Spule → Spule bereits aktiviert, Lichtschranke ausgelöst
- Spule deaktivieren bis Lichtschranke nicht mehr ausgelöst
- Spule aktivieren bis Lichtschranke auslöst
- Vorgang wiederholen bis Funktion abgebrochen wird

## Funktion “kontinuierliches Heben”

- Projektil wird von Spule  $n$  bis Spule  $n+x$  gehoben

### Vorgang

- Eisenkern mittig in Spule  $n$ , Spule aktiviert → Spule deaktivieren
- sobald Lichtschranke nicht mehr ausgelöst ist Spule aktivieren
- Deaktivierung der Spule  $n$  wenn Lichtschranke auslöst, aktivieren von Spule  $n+1$
- wiederholen bis Zielspule erreicht, diese erst bei Auslösen von Lichtschranke  $n+x$  aktivieren (=Projektil halten)

## Funktion “schrittweises Heben”

- Projektil wird von Spule  $n$  bis Spule  $n+1$  gehoben und verweilt dabei jeweils kurz an jeder Spule

### Vorgang

- Eisenkern mittig in Spule  $n \rightarrow$  Spule  $n$  deaktivieren
- sobald Lichtschranke nicht mehr auslöst Spule aktivieren
- bei Auslösen der Lichtschranke Spule deaktivieren
- bei Auslösen der Lichtschranke  $n+1$  Spule  $n+1$  aktivieren  $\rightarrow$  Projektil wird gehalten (Haltezeit noch festzulegen)
- Vorgang wiederholen bis Zielspule erreicht

## Funktion “gebremstes Fallenlassen”

- Projektil wird von Spule  $n$  bis Spule  $n-x$  fallengelassen und von Spule zu Spule abgebremst

### Vorgang

- Projektil ist mittig in Spule  $n \rightarrow$  Spule deaktivieren
- bei Auslösen von Lichtschranke  $n-1$  warten, bis diese nicht mehr ausgelöst
- Spule  $n-1$  aktivieren (Zeit noch festzulegen)
- Vorgang wiederholen bis Zielspule erreicht

## Funktion “schrittweises Fallenlassen”

- Projektil wird von Spule  $n$  bis Spule  $n-1$  fallen gelassen und verweilt dabei kurz an jeder Spule

## Vorgang

- ähnlich dem gebremsten Fallenlassen, aber Spulen sind länger aktiv

## Einschränkungen

- System funktioniert nur für die vorgegebenen Parameter (Längen, Abstände, Gewichte, etc.)
- konsistente Zustände zwischen Programmteilen (z.B. Spulen werden von Teilprogrammen mittig erwartet, müssen also dort hinterlassen werden)

## Annahmen

- Zeit zwischen Lichtschranken circa  $0.5\text{s} \rightarrow V_{\max}$  circa  $460\text{mm/s} \rightarrow 1.6\text{km/h}$
- daher max. Lichtschrankenaktivierungsdauer  $0.178\text{s} \rightarrow$  bei Abfragebetrieb Abtastung mind. alle  $0.089\text{s}$  nötig

## Fehlersituationen

- mehr als eine Lichtschranke gleichzeitig ausgelöst
- mehr als eine Spule gleichzeitig aktiviert
- Spule zu lang aktiviert

## Idee

- Fehler- bzw. Zustandsanzeige über LEDs
- Profiling (Spulenaktivierungsdauer, Projektilgeschwindigkeit, etc.)
- Einspielen eines neuen Programmablaufes über serielle Schnittstelle
  - dazu Software und geeignetes Protokoll entwerfen

Vielen Dank für Ihre Aufmerksamkeit