

## Aufgabe 4 — Kugelfallversuch

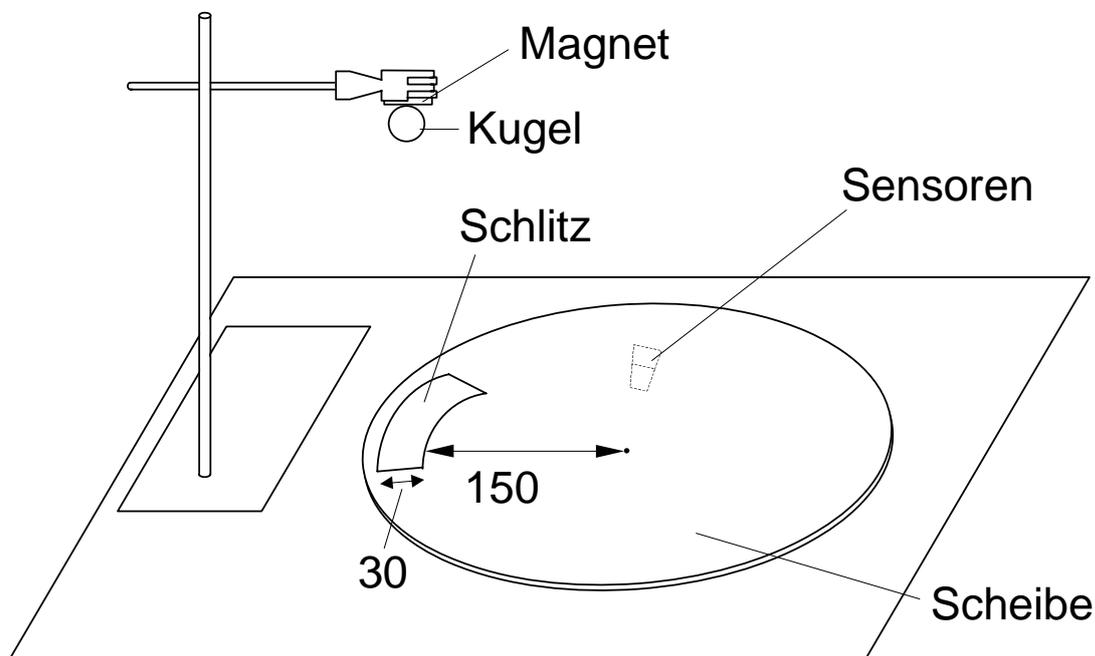
Lernziel: Echtzeitprogrammierung mit Ada

Unterlagen: Ada Language Reference Manual  
Vorgaben im Verzeichnis /home/pdv/pdv/lehre/ees/zu\_a4

### Aufgabenstellung

Es soll ein Programm geschrieben werden, welches eine magnetisch gehaltene Kugel rechtzeitig auslöst, so daß diese durch den Schlitz einer sich drehenden Scheibe fällt.

### Versuchsaufbau



Die Scheibe hat einen Durchmesser von 400 mm und ist 17 mm dick; der Durchmesser der Kugel beträgt 12 mm. Der Schlitz ist 30 mm breit und 150 mm vom Scheibenmittelpunkt entfernt. Sein Öffnungswinkel beträgt 39 Grad. Fallhöhe  $h$  (gemessen von der Kugelunterseite bis zur Scheibenoberseite) und Drehgeschwindigkeit sowie -richtung der Scheibe können von Versuch zu Versuch variiert werden.

### Theoretischer Aufgabenteil

Stellen Sie die Formel für den richtigen Auslöszeitpunkt in Abhängigkeit von Fallhöhe, Umdrehungszeit, Scheibendicke und Kugeldurchmesser auf! Lassen Sie in Ihre Berechnungen eine Sicherheitsreserve mit eingehen, so daß nicht die gesamte Schlitzlänge ausgenutzt wird.

Berechnen Sie die minimal mögliche Umdrehungszeit der Scheibe (die der maximal möglichen Drehgeschwindigkeit entspricht), um noch mit einer Kugel (Durchmesser  $D_k$ ) aus der Fallhöhe  $h$  durch den Schlitz der Scheibe zu treffen.

Die Ergebnisse Ihrer Berechnungen sind schriftlich als Teil der Aufgabenlösung abzugeben.

Hinweise:

- Der Luftwiderstand kann vernachlässigt werden.
- Erdbeschleunigung:  $g = 9,81 \text{ m/sec}^2$
- Es kann der Einfachheit halber davon ausgegangen werden, daß sich die Scheibe mit konstanter Geschwindigkeit bewegt. Wer Lust hat, kann als schönere Lösung implementieren, daß die die Scheibe verlangsamende Reibung (zumindest näherungsweise) mit beachtet wird.
- Vom Zeitpunkt des Abschaltens des Magneten bis zum Beginn des Kugelfalls vergeht eine kurze Zeit, die ebenfalls vernachlässigt werden kann.

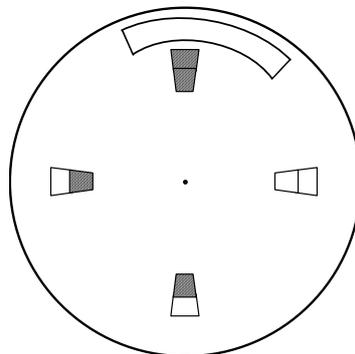
### Praktischer Aufgabenteil

Schreiben Sie ein Ada-Programm, welches die Kugel so auslöst, daß diese ohne anzustoßen durch den Schlitz der sich drehenden Scheibe fällt. Die Fallhöhe muß zu Beginn des Versuchs gemessen und vom Programm eingelesen werden. Berechnen und zeigen Sie die sich daraus ergebende minimale Umdrehungszeit! Danach wird die Scheibe in Drehung versetzt. Ist eine konstante Geschwindigkeit erreicht, muß die Kugel zum vorher berechneten Zeitpunkt ausgelöst werden und durch den Schlitz fallen. Das Programm soll eine entsprechende Fehlermeldung ausgeben, wenn die Umdrehungszeit zu gering ist. Danach kann die Scheibe von Hand gebremst werden oder das Programm einfach abwarten, bis durch die Reibung die Geschwindigkeit gering genug ist. Beachten Sie außerdem, daß die Kugel zu einem geeigneten Zeitpunkt am Magnet befestigt werden muß.

Strukturieren Sie Ihr Programm übersichtlich! Beispielsweise können die Funktionen für Sensoren und Magnet in einem Paket implementiert werden, welches dann vom Hauptprogramm aus importiert wird.

### Hinweise zur Hardware

Der Versuchsaufbau ist an einem der Echtzeitrechner angeschlossen. Zur Verfügung stehen zwei optische Sensoren zur Lage- und Drehrichtungserkennung der Scheibe.



Über den Sensoren sind an der Unterseite der Scheibe Markierungen in Abständen von 90 Grad angebracht (siehe Abbildung). Wie im Bild skizziert, sind die Markierungen so angebracht, dass sich ein Rand der Doppelmarkierung genau in der Mitte des Schlitzes befindet.

Der aktuelle Sensorzustand kann über Baustein X, Port A, (Ada-) Bits 7 und 6 (Eingabe) abgefragt werden. Befindet sich eine in der Skizze dunkel dargestellte Markierung über dem Sensor, ist der ausgelesene Wert 0.

Bit 5 (Ausgabe) von Port A steuert den Haltemagnet. Er ist eingeschaltet, wenn das Bit den Wert 1 hat. Zur einfacheren Fehlersuche zeigen die drei Leuchtdioden des Steuergerätes den aktuellen Zustand der beiden Sensoren sowie des Magneten an. Unabhängig vom Steuerrechner kann der Magnet aktiviert werden, indem der Schalter am Steuergerät in die obere Position gebracht wird. Mit dem Drehknopf kann die Spannung am Magnet und damit die Haltekraft eingestellt werden.

Für die korrekte Übertragung von Sensor- und Magnetsignalen zwischen Echtzeitrechner und Steuergerät muß der verwendete Schnittstellenbaustein entsprechend programmiert werden! Beachten Sie in diesem Zusammenhang u. a. die Richtung der einzelnen Port-Bits (Ein- bzw. Ausgabe).

Für die notwendigen Zeitmessungen steht ein Timer-Baustein zur Verfügung. Die zu seiner Ansteuerung notwendigen Funktionen sind bereits im Paket TIMER realisiert, welches nur aus dem Vorgabenverzeichnis kopiert und übersetzt werden muß.

Verwendbare Funktionen:

```
procedure TIMER_STARTEN; Setzt den Zähler auf Null und startet den Timer
procedure TIMER_STOPPEN; Hält den Zähler an
function TIMER_LESEN return FLOAT; Gibt gemessene Zeit zwischen
    TIMER_STARTEN und TIMER_STOPPEN in Millisekunden zurück
procedure WARTEN (ZEIT : in FLOAT); Wartet angegebene Zeit in Millisekunden
```

*Zum Schluß noch ein allgemeiner Hinweis: Bitte behandeln Sie den vorhandenen Aufbau so, daß alle Gruppen die Möglichkeit haben, die Aufgabe zu bearbeiten! Seien Sie bitte vorsichtig bei den Kugelfallversuchen und setzen Sie die Scheibe nicht in zu schnelle Drehung, um Unfälle zu vermeiden.*