

Die Ballwurfmaschine

Die Aufgabe, die diesmal maschinell gelöst ist, besteht darin, eine Kugel über eine vorgeählte Strecke zu werfen. Hierzu gibt es generell viele verschiedene Lösungsansätze. Der hier beschriebene besteht darin, die Kugel mittels eines Schwungrads auf eine Anfangsgeschwindigkeit zu bringen, um sie dann im richtigen Winkel loszulassen.

1. Die Maschine

Bild 1 ergibt einen Überblick über die Maschine selbst. Die wesentlichen Baugruppen sind das große Schwungrad, der Antriebsmotor, der Schleifring für die Stromübertragung zum Elektromagneten und im Hintergrund der Viertelbogen für die Gabellichtschranke.

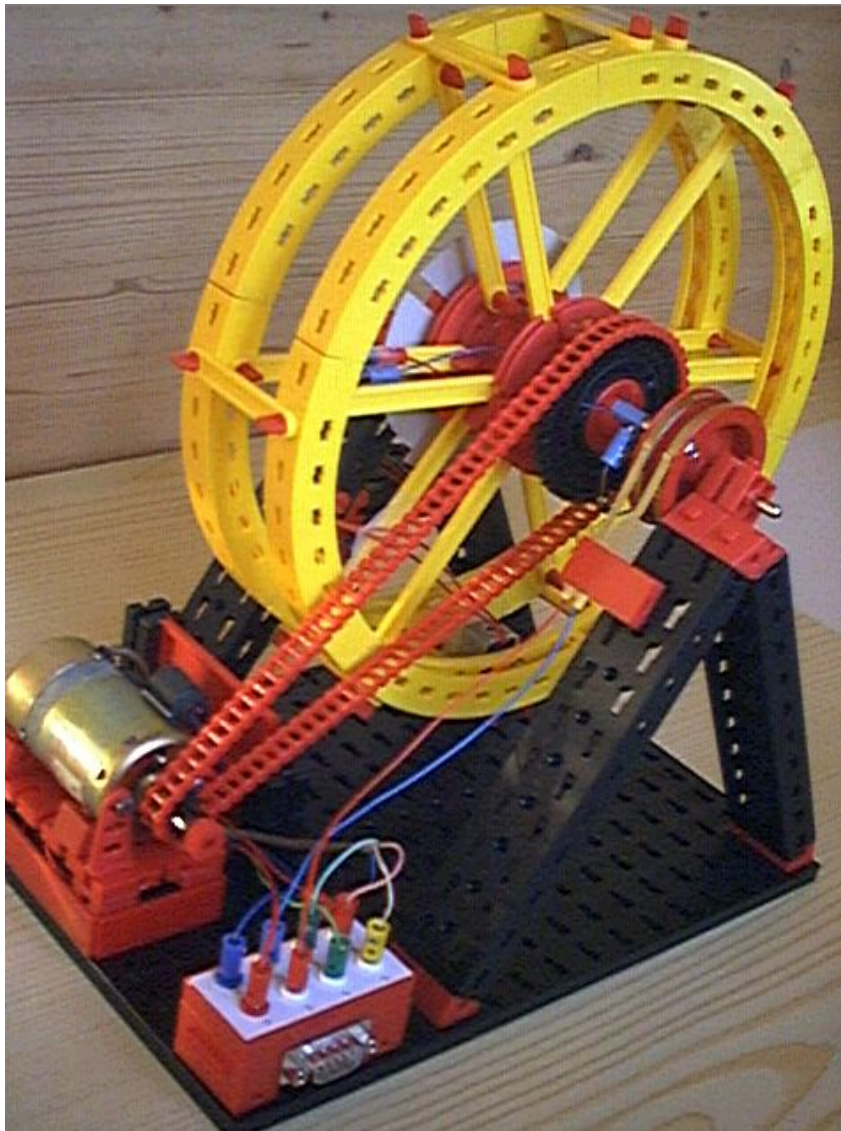


Bild 1: Die Ballwurfmaschine

Die Ballwurfmaschine ist für Wurfweiten bis über 5 Meter vorgesehen. Das erfordert Drehzahlen für das Schwungrad bis zu 600 U/min. Dabei entstehen sehr erhebliche Zentrifugalkräfte, die von dem Elektromagneten beherrscht werden müssen. Das Schwungrad ist nur statisch gewuchtet, so daß es mit der zu werfenden Kugel ausreichend ruhig umläuft. Sowie die Kugel ausgelöst ist, wird die Maschine sofort abgeschaltet.

2. Die Sensorik

Die Ballwurfmaschine ist mit 2 Sensoren bestückt, die es erlauben, die Winkelstellung des Schwungrads zu bestimmen und die aktuelle Drehzahl zu messen.

2.1 Inkremental-Drehgeber

Um die Winkelstellung des Schwungrads zu verfolgen, ist der Antriebsmotor mit einem Inkrementaldrehgeber ausgerüstet. Dieser hat eine Auflösung von 3 Grad (120 Impulse je Umdrehung). Die Auswerteelektronik erhöht die Auflösung um den Faktor 4. Die mechanische Untersetzung vom Antriebsmotor auf das Schwungrad ist ebenfalls 4, so daß sich insgesamt für die Winkelauflösung am Schwungrad 0,1875 Grad oder 1920 Impulse je Umdrehung ergeben.

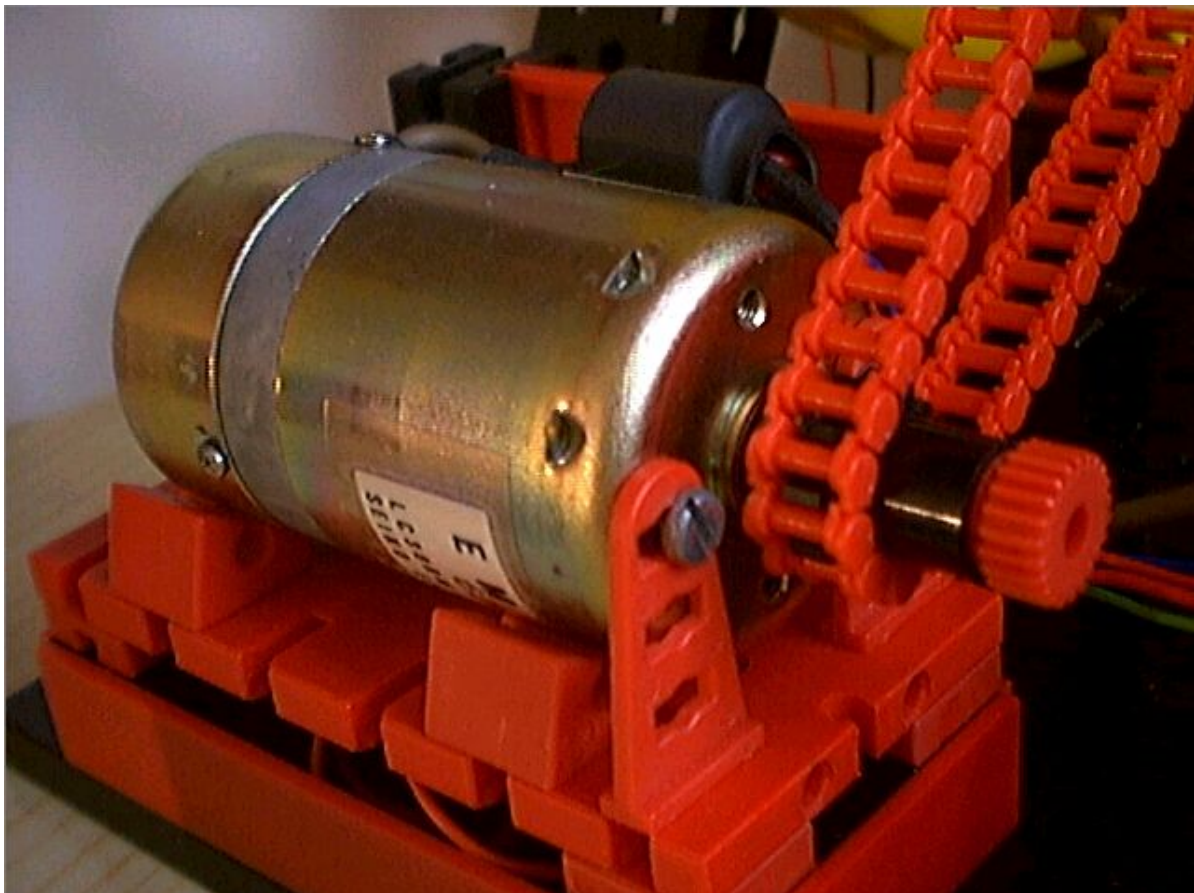


Bild 2: Der Antriebsmotor mit Inkremental-Drehgeber

Die Auswertung des Drehgebers am Motor geschieht in zwei Stufen. Der Drehgeber gibt zwei TTL-Digitalsignale ab, die 90° phasenverschoben sind. Aus der Phasenlage bestimmt sich die Drehrichtung und aus der Frequenz der Signale die Drehzahl des Motors. Die Phasenlage der Signale ist nicht weiter relevant, da die Maschine nur in einer Drehrichtung betrieben wird.

Die erste Auswertestufe besteht darin, daß die beiden Signale bei jedem Flankenwechsel in Einzelimpulse von 2 μ s Dauer zerlegt werden. Signal A liefert also bei jeder ansteigenden Flanke einen Impuls und bei jeder abfallenden Flanke. Ebenso Signal B. Daraus resultiert die Vervierfachung der Auflösung.

In einer nachfolgenden Gatterschaltung aus UND- und ODER-Gatter werden die Einzelimpulse in Abhängigkeit der Phasenlage der Signale A und B sortiert, so daß die Impulse je nach Drehrichtung entweder am Zählausgang ‚aufwärts‘ oder am Zählausgang ‚abwärts‘ erscheinen. In diesem Falle bleibt die Drehrichtung konstant, so daß nur ein Ausgang verwendet wird. Bild 3 zeigt den Schaltplan der ersten Auswertestufe und Bild 4 die dazugehörige elektrische Schaltung.

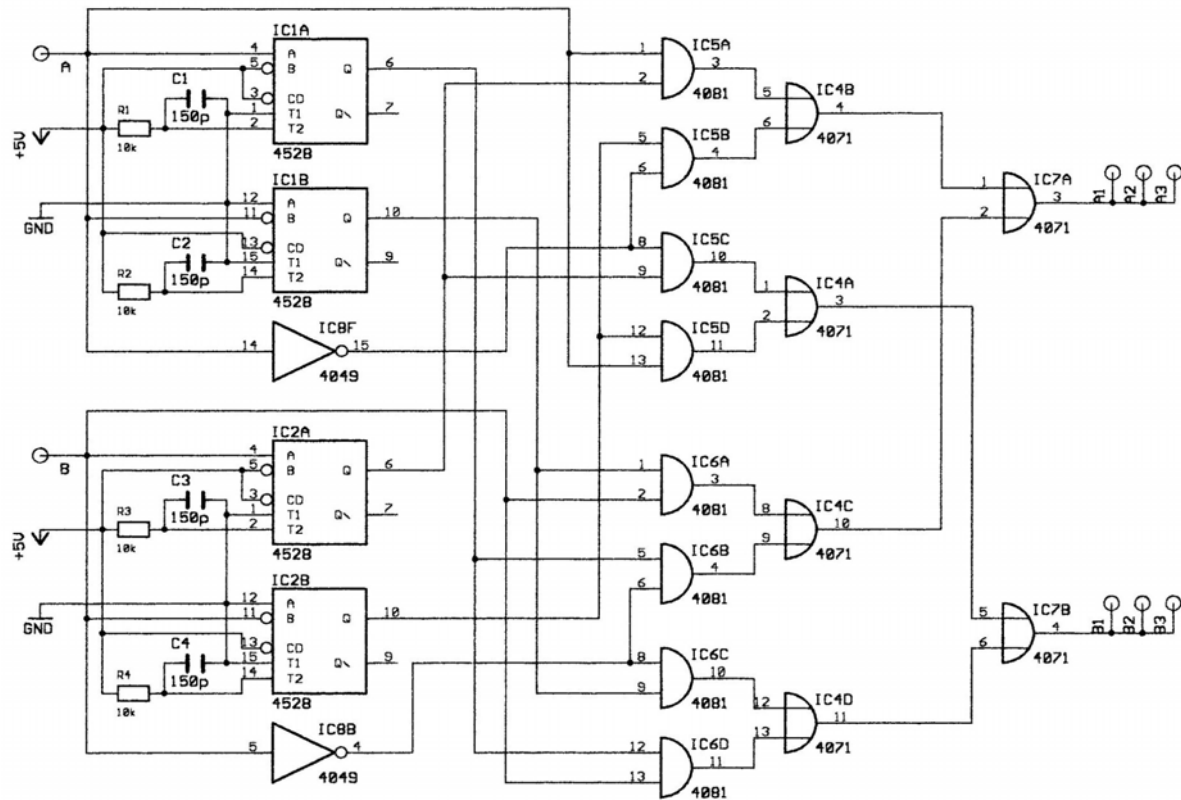


Bild 3: Schaltplan der ersten Auswertestufe...

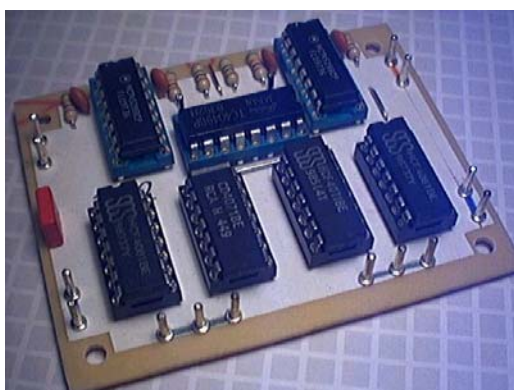


Bild 4: ...und ihre elektrische Ausführung

Am Ausgang der ersten Auswertestufe erscheint alle 0,1875 Grad des Schwungrads ein kurzer Impuls, der jetzt noch geeignet zu zählen ist.

Die zweite Auswertestufe zur Bestimmung der Winkellage des Schwungrads ist softwarebasiert. Jeder Impuls aus der ersten Auswertestufe löst am Steuerrechner einen Interrupt aus. Die Interruptroutine besteht lediglich darin, daß der Zähler um 1 erhöht wird. Erreicht der

Wert 1920, was genau einer Umdrehung des Schwungrads entspricht, stellt die Routine den Zähler auf Null zurück. Damit ist zu jedem Zeitpunkt in der Globalvariablen ‚Zähler‘ ein Maß für die Winkellage gespeichert. Noch ist der Zähler aber nicht auf eine exakte Position des Schwungrads bezogen.

2.2 Die Gabellichtschanke

Die Gabellichtschanke, die sich in der Nähe der Schwungradnabe befindet, liefert ein Signal, das direkt mit der Position des Schwungrads verbunden ist. In Bild 5 ist zu sehen, daß am Schwungrad ein Viertelkreisbogen befestigt ist. Mit jeder Umdrehung gibt die Gabellichtschanke ein Impulssignal im Tastverhältnis 1:3 ab. Ein Viertelbogen ist das Signal high und dann drei Viertelbögen low.

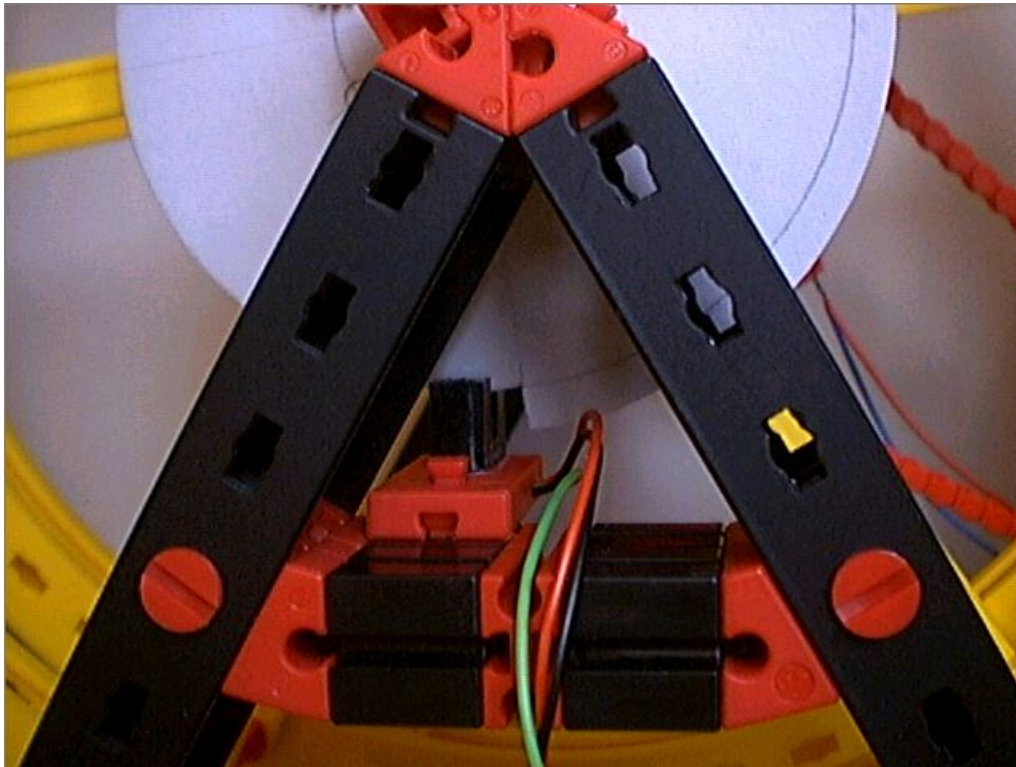


Bild 5: Gabellichtschanke mit Viertelkreis

Die Gabellichtschanke ist mit einer sehr feinen Schlitzblende versehen und gibt ein höchst präzises Signal ab.

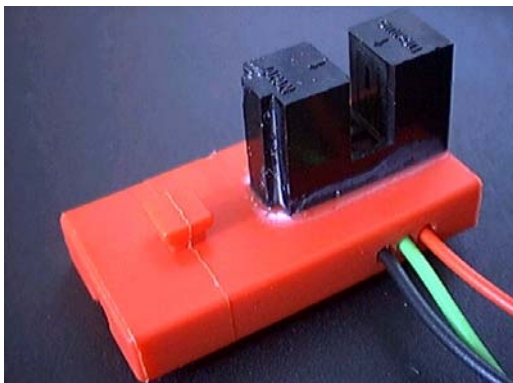


Bild 6: Gabellichtschanke im Detail

Damit der Motorzähler einen definierten Bezugspunkt zur Maschine hat, wird innerhalb der Prozedur zur Messung der Drehzahl (Abwurfgeschwindigkeit) auf das Gabellichtschrankensignal gewartet und zur abfallenden Flanke der Motorzähler gesetzt.

3. Die Bestimmung der Abwurfgeschwindigkeit

Die oben beschriebene Sensorik reicht noch nicht hin, die Drehzahl zu bestimmen, denn dazu kommt noch die Zeit. Der verwendete Steuerrechner erlaubt den Zugriff auf den Bustakt, der quartzgenau 8 MHz zur Verfügung stellt. Zu diesem Zweck ist der Rechner mit einem Multi-IO-Board ausgerüstet, das den Zählerbaustein 82C54 enthält.

Der 82C54 ist ein dreifach Abwärts-Zähler, der auf unterschiedliche Weisen programmierbar ist. Einer der drei Zähler ist direkt mit dem Bustakt verbunden. Er wird als ‚Rate-Generator‘ so programmiert, daß an seinem Ausgang permanent 100 kHz Taktfrequenz erscheint.

Ein weiterer Zähler nimmt diese Frequenz auf. Dessen ‚Gate‘ wird jedoch durch die Gabellichtschranke freigegeben oder gesperrt. Während einer Viertelumdrehung ist das Gate frei und der Zähler zählt Impulse bekannter Frequenz. Danach sperrt die Gabellichtschranke den Eingang und der Zähler kann ausgelesen werden. Die tatsächliche Drehzahl berechnet sich dann aus der Anzahl der gezählten Impulse.

4. Der Steuerungsablauf

Im ersten Schritt initialisiert die Software die wichtigsten Variablen und die Interruptroutine. Sie schaltet den Elektromagneten ein und wartet auf die Bestätigung, daß die Kugel daran haftet. Der Antriebsmotor startet.

Ist das Gabellichtschrankensignal gesetzt, dann wartet die Software darauf, daß es abfällt oder wenn es null ist, daß es erscheint. In beiden Fällen ist in diesem Augenblick die Winkellage klar und der Zähler wird auf entsprechende Werte gesetzt.

Daran schließt sich die Phase der Drehzahlmessung an. Je drei Messungen werden miteinander verglichen. Ist die Abweichung weniger als ein Prozent, dann wird sofort der dazugehörige Abwurfwinkel berechnet. Die Software wartet dann, bis der Winkel erreicht ist und schaltet sofort den Magneten und den Motor aus.

5. Technische Daten

Radius des umlaufenden Kugelschwerpunkts:	121 mm
Maximale Drehzahl des Schwungrads:	600 U/min
erreichbare Abwurfgeschwindigkeit:	7,6 m/s
erreichbare Wurfweite:	6 m
Durchmesser der Stahlkugel	10 mm
Gewicht:	4,08 gr
maximale Fliehkraft der Kugel:	1,94 N
Zeitbedarf für vollständigen Wurfvorgang:	12 s

6. Software in kommentierter Form

```
program Wurf;

uses crt,dos;

const IRQNr = 5;

var MZ,WI      : integer;
    ISRold     : procedure;
    V0,A,B,C,E,H : real;

(*-----*)
(* Erläuterung zu den Variablen und Konstanten: *)
(* *)
(* IRQNr = 5: Nummer des verwendeten Interrupts. Nr. 5 ist bei DOS frei *)
(* MZ : Motorzähler. MZ zählt die Impulse des Inkrementalgebers 0..1919 *)
(* Wi : theoretischer, idealer Abwurfwinkel *)
(* V0 : Abwurfgeschwindigkeit, entspricht der Tangentialgeschwindigkeit *)
(* A,B,C : Speicher für drei Messungen von V0 *)
(* E : gewünschte Wurfweite aber der Maschinenvorderkante *)
(* H : Höhendifferenz zwischen Abwurfpunkt und Maschinenaufstandfläche *)
(*-----*)

{$F+}
procedure ISRnew; interrupt;
begin
    (* Nach dem Aufruf des Interrupts muß der nächste Interrupt über das *)
    (* Board DAS08 erneut freigeschaltet werden *)
    port[$302] := 8;
    (* Motorzähler um einen erhöhen *)
    inc(MZ);
    (* und den Wertebereich auf eine Umdrehung beschränken *)
    if MZ = 1920 then MZ := 0;
    (* Interruptroutine als beendet melden *)
    port[$020] := 32;
end;
{$F-}

procedure Init;
begin
    (* Bildschirm löschen und die Variablen mit Dummies und Nullen belegen*)
    clrscr;
    MZ := 0;
    V0 := 0.0;
    A := 1.0;
    B := 2.0;
    C := 3.0;
    (* Abschalten aller Ausgangsleitungen des Parallelport Data-Registers *)
    port[$378] := 0;
    (* Zähler 2 als Rate-Generator schalten. Ausgabe ist danach 100 kHz *)
    port[$307] := 180; (* Betriebsart *)
    port[$306] := 40; (* Low-Byte *)
    port[$306] := 0; (* HighByte *)
    (* Aktivieren der Interruptroutine durch verbiegen des Zeigers *)
    getintvec(IRQNr+8,addr(ISRold));
    setintvec(IRQNr+8,addr(ISRnew));
    port[$302] := 8; (* erlaubt IRQ über das Board DAS08 *)
    (* neuen Interrupt aktivieren *)
    port[$021] := port[$021] and not(1 shl IRQNr);
```

```
end;
```

```
procedure V0_holen;
```

```
var I : word;
```

```
begin
```

```
  (* Falls das Signal der Lichtschranke gesetzt ist, dann erst warten *)  
  repeat until (port[$302] and 16) = 0;
```

```
  (* Das Signal ist abgefallen, jetzt die Zählerregister setzen, da *)  
  (* jetzt der Zähler blockiert ist *)  
  port[$307] := 116; (* Betriebsart Zähler 1*)  
  port[$305] := 255; (* Low-Byte *)  
  port[$305] := 255; (* HighByte *)
```

```
  (* jetzt wieder warten, bis das Lichtschrankensignal kommt *)  
  repeat until (port[$302] and 16) = 16;  
  (* jetzt ist Zähler 1 frei und zählt im Takt von Zähler 2 abwärts *)
```

```
  (* warten, bis das Signal der Lichtschranke wieder abfällt *)  
  repeat until (port[$302] and 16) = 0;  
  (* die Zählung ist beendet. Jetzt den Motorzähler setzen, weil in *)  
  (* diesem Augenblick die Stellung des Schwungrads bekannt ist. *)  
  MZ := 1740;  
  (* und Zähler 1 auslesen *)  
  I := port[$305];  
  I := 65535 - (I + port[$305] * 256);  
  (* daraus jetzt die Abwurfgeschwindigkeit errechnen. Ergebnis ist m/s *)  
  if I <> 0 then V0 := 0.1963/(I/100000);  
  (* die Zahl 0.1963 berücksichtigt diverse Modelleigenschaften *)
```

```
end;
```

```
procedure Spannung(E,H : real);
```

```
(* Diese Prozedur schätzt die notwendige Mindestmotorspannung ab *)
```

```
var Ink,E1,U,X : real;
```

```
  Ch      : char;
```

```
begin
```

```
  (* Die Wurfentfernung wird um 0,17 m verlängert, da sie normalerweise *)  
  (* ab der Maschinenvorderkante gezählt wird und nicht ab dem Abwurfort*)  
  E := E + 0.17;  
  (* Das Inkrement für das Iterationsverfahren wird auf 1 gesetzt *)  
  Ink := 1.0;  
  (* Die Spannung U ist anfangs null *)  
  U := 0.0;
```

```
repeat
```

```
  (* Hochzählen der Spannung um das Inkrement *)  
  U := U + Ink;  
  (* X ist vorläufige Spannung durch 2 * Erdbeschleunigung *)  
  X := U*U/19.62;  
  (* daraus kann eine vorläufige Gesamtwurfweite gerechnet werden *)  
  E1 := X + U/1.414 * sqrt((X+2*H)/9.81);  
  (* Ist die vorläufige Gesamtwurfweite zu groß und das Inkrement *)  
  (* größer Null bzw. die vorläufige Wurfweite zu klein und das *)  
  (* Inkrement kleiner Null, dann wird das Inkrement durch -10 geteilt*)  
  if ((E1>E) and (Ink>0)) or ((E1<E) and (Ink<0)) then Ink := Ink/-10.0;  
  (* solange, bis die vorläufige Gesamtwurfweite und die gewünschte *)  
  (* Wurfweite übereinstimmen *)
```

```
until abs(E-E1)/E < 0.001;
```

```
(* Die Spannung in Volt ergibt sich jetzt aus einer gemessenen *)
```

```
(* Regressionsgeraden *)
```

```
U := ((U + 2.2) / 0.76) * 1.05;
```

```

(* ausgeben *)
writeln('Spannung vorwählen : ',U:2:1,' Volt');
(* warten bis das OK von der Tastatur kommt *)
repeat until keypressed;
Ch := readkey;
(* Wurfweite wieder auf den ursprünglichen Wert zurücksetzen *)
E := E - 0.17;
end;

procedure Winkel(E,H : real);
var AWM,Ink,X,E1,E2 : real;
begin
  AWM := 0.7506; (* AbwurfWinkelMaximal = 45 Grad in Bogenmaß *)
  Ink := -0.085; (* 5 Grad Inkrement in Bogenmaß *)
  repeat
    (* Iterativ zum tatsächlichen Abwurfwinkel *)
    AWM := AWM + Ink;
    (* X ist die Wurfweite bis zum höchsten Punkt der Wurfparabel *)
    X := V0*V0*sin(2*AWM)/19.62;
    (* von da aus kommt noch der waagerechte Wurf hinzu. Fallhöhe ist *)
    (* die eingegebene Höhendifferenz H und die Höhe vom Abwurfpunkt *)
    (* bis zum Scheitelpunkt der Wurfparabel. Das ergibt zusammen E1 *)
    E1 :=X+V0*sin(AWM)*sqrt(((V0*V0*cos(AWM)*cos(AWM))/19.62+H)*2)/9.81);
    (* die Gesamtwurfentfernung ist auch noch vom Abwurfwinkel abhängig *)
    E2 := E + 0.082 + 0.125*cos(AWM);
    (* geht der Wurf zu weit und ist das Inkrement auch noch größer Null*)
    (* oder ist der Wurf zu kurz und das Inkrement kleiner Null, dann *)
    (* wird das Inkrement umgepolt und halbiert *)
    if ((E1>E2) and (Ink>0)) or ((E1<E2) and (Ink<0)) then Ink := Ink/-
2.0;
    (* solange, bis Sollwurfweite und Istwurfweite gleich groß sind *)
    (* oder der Abwurfwinkel über 90 Grad beträgt *)
  until (abs(E2-E1)/E2 < 0.001) or (AWM>1.6);
  (* Ist der Abwurfwinkel ok, dann... *)
  if A1<1.6 then
    begin
      (* wird die Gesamtwurfweite ausgegeben *)
      writeln('Gesamtwurfweite ',E2:1:2,' m');
      (* und der Winkel in Motorzählerschritte umgerechnet *)
      WI := round(AWM*960/3.14159);
      (* ansonsten wird er zu null gesetzt *)
    end else WI := 0;
end;

(*-----*)
(*                H A U P T P R O G R A M M                *)
(*-----*)

begin

  Init;
  (* Eingabe, wie weit die Maschine werfen soll. Die Entfernung ist ab *)
  (* Maschinenvorderkante gezählt *)
  write('Gewünschte Wurfweite in m : ');
  read(E);
  (* der Auftreffpunkt muß nicht unbedingt auf der Tischplatte sein, wo *)
  (* der die Maschine steht. Falls ja, dann sind hier 0,25 m einzugeben *)
  (* bzw. 0,25 m minus die Höhe des Auftreffpunkts *)
  write('Höhendifferenz zum Abwurfort in m : ');
  read(H);

```



```

(* sind die Parameter drin, wird erst einmal die mindestens notwendige*)
(* Motorspannung abgeschätzt, die zum Erreichen der Weite notwendig *)
(* ist. Höhere Spannungen ergeben steilere Würfe *)
Spannung(E,H);
writeln;

(* Motor und Magnet werden eingeschaltet *)
port[$378] := 3;
(* eine Sekunde warten, damit der Motor auch läuft. Sonst kann es pas-*)
(* sieren, daß bei der Messung der Wurfgeschwindigkeit durch Null ge- *)
(* teilt wird. *)
delay(1000);

repeat
  (* In V0_holen wird die Abwurfgeschwindigkeit gemessen. *)
  (* aufeinanderfolgende Messungen rücken durch A, B und *)
  (* C durch *)
  V0_holen;
  C := B;
  B := A;
  A := V0;
  (* bis sich die drei Messungen praktisch nicht mehr voneinander *)
  (* unterscheiden. Die Drehzahl ist jetzt konstant und bekannt *)
until (abs(A-C)+abs(B-C))/(2*C) < 0.001;

Winkel(E,H);
(* zu der vorhin gemessenen Abwurfgeschwindigkeit wird jetzt der dazu *)
(* passende Abwurfwinkel berechnet *)
if WI = 0 then
  begin
    (* Tja, die Drehzahl des Schwungrads reicht nicht. *)
    (* Motor aus *)
    port[$378] := 2;
    (* Meldung machen *)
    writeln('Drehzahl wird nicht erreicht. ');
    repeat
      (* warten, bis das Schwungrad zum Stillstand gekommen ist *)
      Wi := MZ;
      delay(100);
    until Wi = MZ;
    (* jetzt erst Magnet ausschalten, die Kugel hängt ja noch dran *)
    port[$378] := 0;
  end else
  begin
    (* ansonsten warten, bis der Abwurfwinkel erreicht ist. *)
    repeat until MZ = Wi-round(V0*10.1);
    (* Magnet aus, die Kugel fliegt *)
    port[$378] := 0;
    (* Ausgabe der echten Wurfparameter *)
    writeln('Abwurfwinkel theor. ',Wi/5.333:2:1,' Grad');
    writeln('Abwurfwinkel prakt. ',(Wi-round(V0*10.1))/5.333:2:1,'
Grad');
    writeln('Geschwindigkeit ',V0:1:2,' m/s');
  end;

  (* Fertig, Interrupt abschalten und wieder zurückbiegen *)
  port[$021] := port[$021] or (1 shl IRQNr);
  setintvec(IRQNr+8,addr(ISRold));

  repeat until keypressed;

end.

```