

# CINTECH

**CINTECH B.V.**  
Leermiddelen voor het  
informatieonderwijs  
Computers en computer  
techniek

## Interfacing

Interfacing is het verbinden van een (micro)computer met de buitenwereld. Globaal dient de computer voorzien te worden van sensoren ("zintuigen") en signalen van buiten op te nemen en van apparatuur op naar buiten toe actie te ondernemen. Natuurlijk is er ook een interfacing-problematiek bij het koppelen van bijvoorbeeld een printer of een beeldscherm aan een computer. Hoewel in deze laatste gevallen ook vaak de nodige problemen kunnen optreden, denk maar eens aan een "verkeerde" printer bij een bepaalde computer, valt het hier in de praktijk nogal meer er zijn standaard oplossingen in de vorm van een complete set, computer - printer - disk drive - tape recorder - beeldscherm. In het onderwerp is de koppeling aan andersoortige omgevingen een veel minder ontwikkeld terrein. Toch is het juist daar van belang te laten zien dat een computer voor veel meer zaken te gebruiken is, dan voor het bewerken van een geheugen en een beeldscherm. Per saldo is een computer ook een veelzijdig programmeerbaar meet-, regel-, reken- en stuurinstrument. Tot voor kort waren deze toepassingen echter alleen bereikbaar voor handige knutselaars in de elektronica: PIA's, buffers, adresdecoding, handshaking, printplaten, een wirwar aan draden etc. Nu niet direct zaken die binnen het bereik liggen van de meeste leerkrachten die zich met burgerinformatica bezighouden. Toch is de techniek een element van onze cultuur geworden, men kan daar in de school niet omheen, ook niet bij de algemeen voorkomende vakken. Juist daar niet! De leerling kan best geconfronteerd worden met automatiseringsdenken, met meet- en regelproblemen en met eenvoudige robotica. Dit dient echter in een juiste omgeving te gebeuren: de elektronica zelf dient grotendeels "verborgen" te blijven, de snoertjes-cultuur moet beperkt blijven. Bovendien moet de programmeer-omgeving vriendelijk zijn. Signalen moeten bijvoorbeeld met INP en OUT ingelezen of weggestuurd kunnen worden, PEEK en POKE zijn al minder toegankelijk, vooral op "bizarre" geheugenplaatsen. Maar ook daar valt nog mee te werken omdat naar enkele adressen een rol spelen. Het programma moet met enkele ALS..... DAN..... constructies kunnen werken, een stroomdiagram moet simpel en overzichtelijk blijven.

Fabrikanten van leermiddelen beginnen deze markt te ontdekken, bekende leveranciers als Leybold, Phyx en Griffin bieden al het een en ander. Onlangs is de firma CINTECH (Heilige Geeststraat 5, 5301 CX ZALTBOMMEL, tel. 04180-5518) met een verrassend aanbod aangetreden. Het gaat hier om een geheel Nederlands product, dat ontwikkeld is door een leraar en op een aantal scholen grondig is uitgetest.

Centraal in de Cintech materialen staat de I/O-unit (Input/Output), die op iedere computer is aan te sluiten. Voor elke computer is er een afzonderlijke staker, deze is al ontwikkeld voor de ZX-Spectrum de Commodore 64, de P2000 en de Aster. De technische specificatie is aan het eind van dit artikel te vinden. De I/O-unit bevat een analoog-digitaal en een digitaal-analoog converter, zowel voor input als voor output, en een apart voedingsgedeelte voor de randapparatuur. Alle randapparatuur is bijzonder eenvoudig bij het aansluiten:

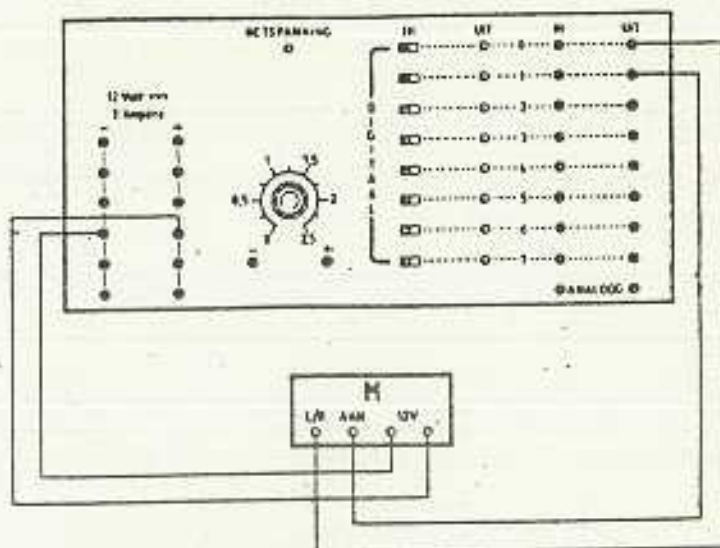
rood en zwart voor de voeding van het randapparaat en een groen snoertje voor het te ontvangen of weg te sturen signaal. De gehele communicatie tussen computer en buitenwereld verloopt via twee porten, één voor analoge en één voor digitale signalen.

De randapparatuur bestaat uit sensoren en energieregelingen voor 12 V en 220 V (goed beveiligd, continu regelbaar). Er zijn sensoren voor het meten van licht, temperatuur geleiding, vloeistofniveau en verplaatsing. Aan de energieregelingen is van alles te hangen: een bureaulamp, een aquariumpomp, een motor, een spoor-trein, een hijskraan, een electrisch zonnescherm etc. Een en ander zowel voor 0 - 12 V (gelijk) als 0 - 220 V (wissel). Met wat handig opstellen is een complete flessenvulinstallatie te realiseren, inclusief het transport en de beveiliging. Met aardige is dat dergelijke opstellingen gerealiseerd kunnen worden zonder

veel kennis van elektronica. Met noties als "er is stroom nodig om een apparaat te laten werken" en "elk apparaat geeft of vraagt een stuursignaal" kom je een heel eind. Dergelijke kennis ontbreekt ook niet vanzelf, maar is voor een breed publiek belangrijk en bereikbaar.

### Een voorbeeld: het digitaal sturen van een motor

Een 12 Volt (speelgoed) motor, gelijkstroom, wordt volgens onderstaand schema op de I/O-unit aangesloten. De motor kan linksom of rechtsom draaien ("verwisselen van plus en min"), dit gebeurt bij L/R. Aan en uit schakelen gebeurt bij AAN. In de beide rechterdraden loopt dus een stuursignaal, kies dus groene snoeren. De motor wordt gevoed door de 12 V, 3 A voeding op de I/O-unit (rode en zwarte draad), je zou hier ook een batterij voor kunnen nemen! Alle stekkerbussen hebben de overeenkomstige kleur, verglassen is praktisch rit-geeloten.



Met het volgende programma draait de motor enkele seconden linksom, vervolgens enkele seconden rechtsom en dat steeds opnieuw. Ik geef het programma voor de P2000, de digitale communicatie verloopt daar via poort 64:

```
10 OUT 64,2      2=00000010 d.v.z. "aan" en "linksom"
20 GOSUB 100     wachtuus
30 OUT 64,3      3=00000011 d.v.z. "aan" en "rechtsom"
40 GOSUB 100     wachtuus
50 GOTO 10       opnieuw
100 FOR I=1 TO 3000: NEXT I: RETURN
```

Het voorbeeld is wat flauw, maar hang aan de motor een katrol en daaraan een scherm voor een raan(pje), meet op de analoge ingang met de lichtsensoren de hoeveelheid zonlicht en het eerste automatiseringsproject is in de klas op leerlingniveau gerealiseerd..... De uitbreiding van het programma kan als volgt:

zonlicht en het eerste automatiseringsproject is in de klas op leerlingniveau gerealiseerd.... De uitbreiding van het programma kan als volgt:

```
5 MEET = INP (65)      analoge communicatie over
                       poort 65
6 IF MEET > 100 THEN ..... ELSE .....
```

Aan dergelijke programmatuur is ook te zien dat de aandacht gemakkelijk te richten is op de zaken waar het bij automatisering om gaat. Dit is een stukje informatica voor iedereen.

Het Cintech materiaal is binnenkort leverbaar. De basisset gaat circa f 1375,- kosten (I/O-unit, schakelunit, energieregeling, diverse sensoren). Er zijn enkele uitbreidingssets in de prijsklasse van een paar honderd gulden. Een complete set gaat ongeveer f 2500,- kosten. Prijzen exclusief BTW en exclusief

## INHOUD

Hoofdstuk 1: De computer	blz. 1
Hoofdstuk 2: De Input/Output-unit	blz. 7
hoofdstuk 3: <u>Digitaal</u> meten en sturen	blz. 12
A. Digitaal meten	blz. 13
B. Digitaal sturen	blz. 21
Hoofdstuk 4: <u>Analoog</u> meten en sturen	blz. 29
A. Analoog meten	blz. 30
B. Analoog sturen	blz. 41
Hoofdstuk 5: Eenvoudige regelingen	blz. 46
Bijlage : Binaire getallen	blz. 1/9

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

## Hoofdstuk 1.

### De computer.

Een computer is een machine die informatie bewerkt. Die informatie kan in de computer gevoerd worden, in de vorm van een programma of als losse gegevens. De computer bewerkt deze gegevens en voert het resultaat uit.

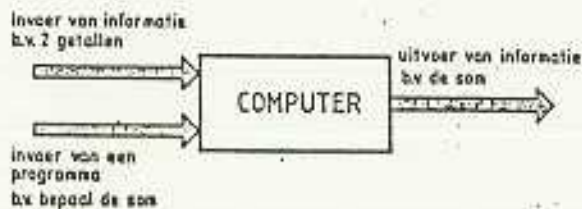


fig.1

De invoer van informatie gebeurt via INVOER-apparaten zoals bijvoorbeeld het toetsenbord, de magneetschijf (floppy-disc) of het cassette-bandje. De uitvoer van informatie geschiedt met UITVOER-apparaten zoals bijvoorbeeld het beeldscherm, de printer of de plotter (teken-machine). Deze in- en uitvoerapparaten zijn met de computer verbonden. We noemen ze RAND-APPARATUUR.

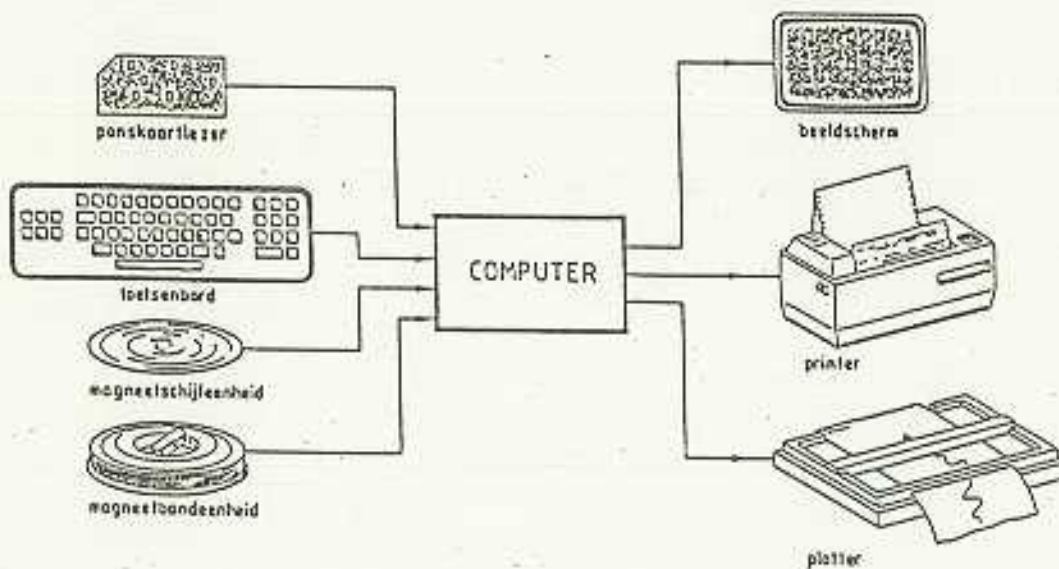
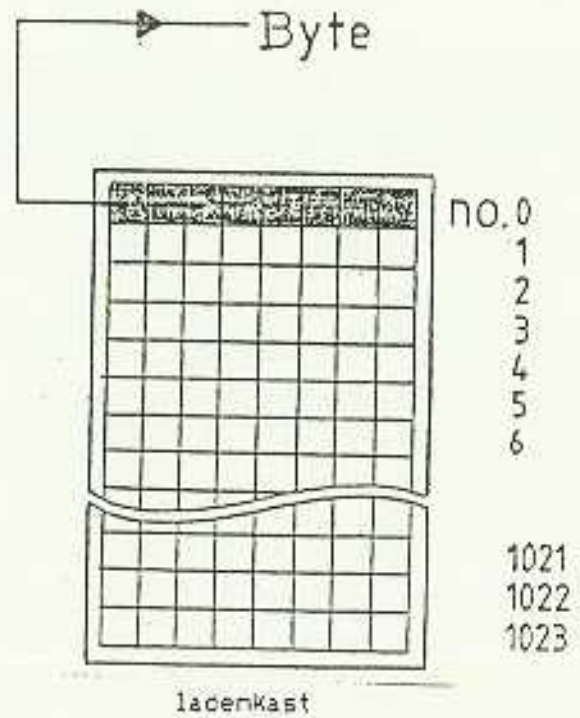


fig.2

Hoe we nu met de computer kunnen werken wordt misschien duidelijk als we het geheugen van de computer gaan bekijken. Een computer is een elektronisch apparaat. Dit wil zeggen; een apparaat dat met spanningen werkt. Het geheugen van een computer, met een geheugen van 16 K, kunnen we voorstellen als 16 ladenkisten met elk 1024 laden, die informatie kunnen bevatten. Zo'n lade noemen we een BYTE. Elke lade heeft een nummer (adres).

In figuur 3 is zo'n kist geschetst.



ladenkast

fig. 3

In elke lade zitten 8 vakjes.

Zo'n vakje is een schakelaar met 2 mogelijke standen: AAN of UIT. Staat de schakelaar aan, dan krijgt het spanning en geven we dat aan met het symbool [ 1 ].

Staat de schakelaar uit, dan krijgt het geen spanning en geven we dat aan met het symbool [ 0 ].

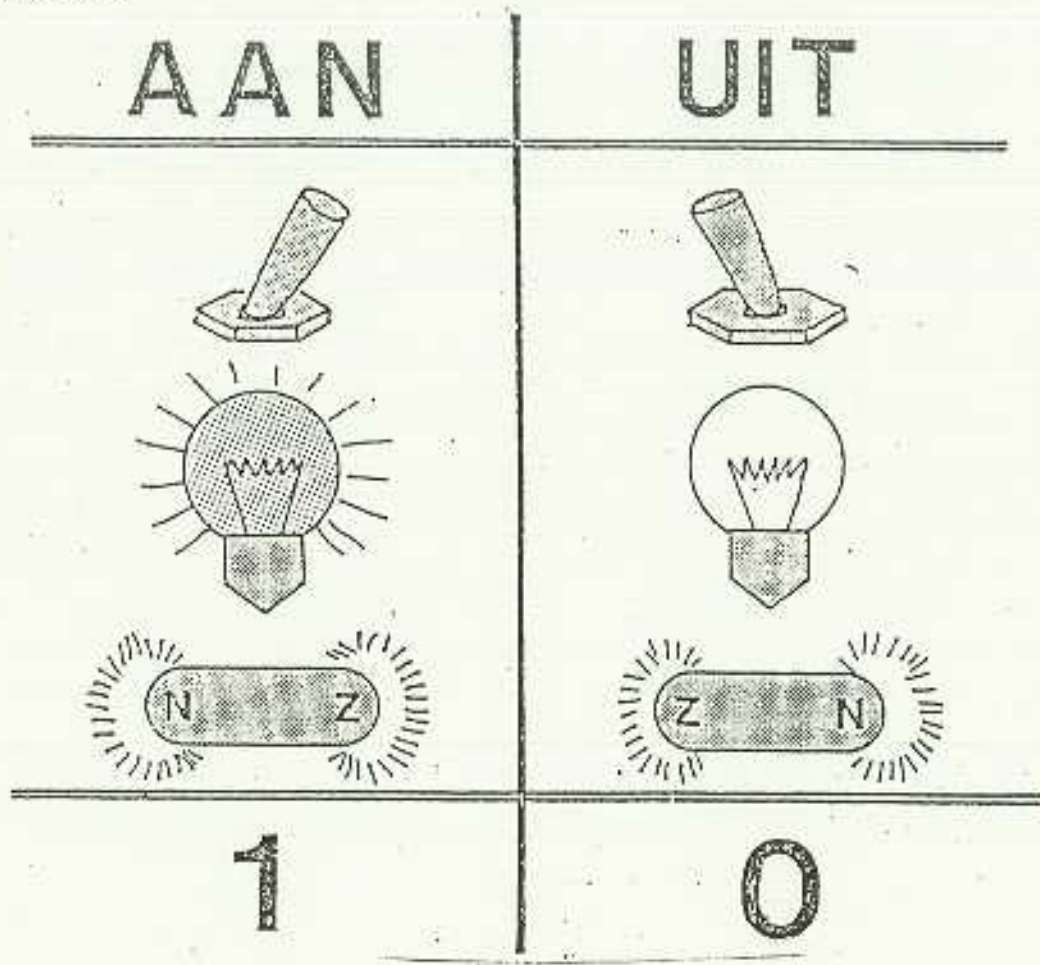


fig. 4

Samen bevatten de 8 vakjes, die we BIT's noemen, informatie. Die informatie kan bestaan uit een cijfer, een getal, een symbool of een commando. (Net zo als we gezien hebben bij de ASCII-tabel in het boekje Pascal, heeft elk symbool of commando een eigen vaste code. Die code heeft dezelfde waarde als het ASCII-nummer).

Het getal "7" zou, bijvoorbeeld, voorgesteld kunnen worden door de code 00000111, de letter "P" door 01010000 en vermenigvuldigen door 00101010.

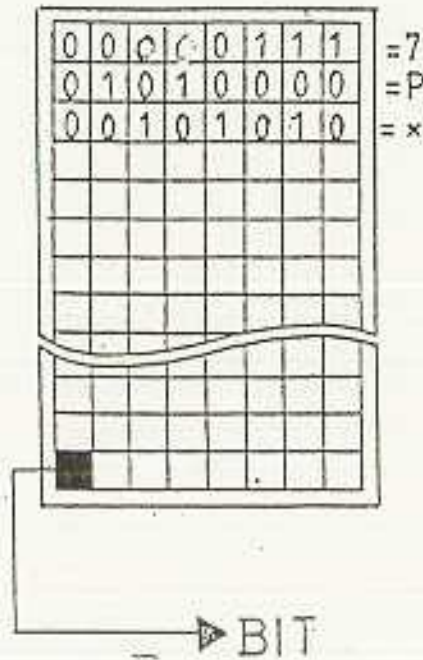


fig. 5

Willen we nu informatie in de computer brengen of veranderen, of informatie uit de computer halen, dan gebeurt dat door middel van spanningen. Zo zal het toetsenbord spanningen naar de computer sturen, terwijl de computer spanningen naar het beeldscherm stuurt.

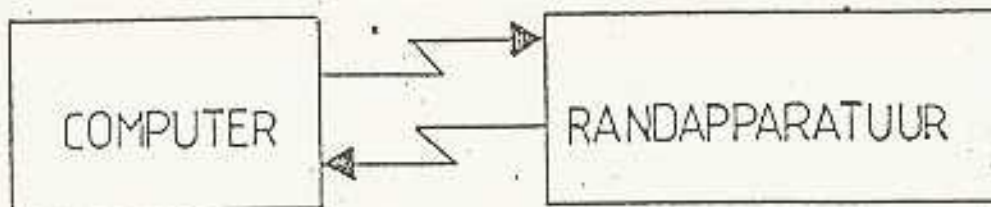


fig.6

In het volgende voorbeeld gaan we bekijken hoe een computer met de buitenwereld communiceert.

Voorbeeld:

In een brandstoftank, bij een tankstation, zit een brandstof-voeler. Deze voeler geeft zijn informatie, er is wel brandstof of er is geen brandstof, door aan de computer. Afhankelijk van die informatie "beslist" de computer, door middel van een programma, of hij een signaal naar buiten stuurt, zodat er een waarschuwings-lampje gaat branden. Zo weet de pompbediende wanneer de tank, bijna, leeg is.

In fig.7 wordt dit schematisch voorgesteld.

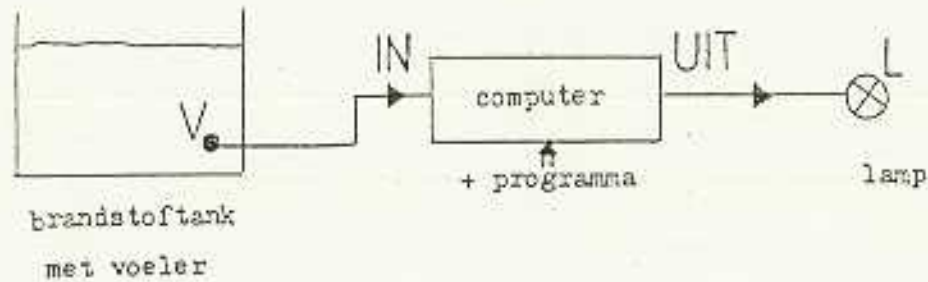


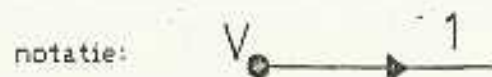
fig.7

In dit voorbeeld vormen de brandstof-voeler en de waarschuwings-lamp de buitenwereld.

VOELER [ V ] is een apparaat dat, d.m.v. spanningen, informatie aan de computer geeft. Zo'n apparaat noemen we een SENSOR. Als de voeler brandstof "voelt", geeft hij geen spanning aan de computer.



Voelt de voeler geen brandstof, dan geeft hij WEL een spanning aan de computer.



De voeler kan maar 2 toestanden onderscheiden; er is wel brandstof of er is geen brandstof. Daarom wordt zo'n voeler ook wel een DIGITALE voeler genoemd.

Omdat de voeler een spanning aan de computer geeft wordt hij op de ingang van de computer aangesloten.

LAMP [ L ] krijgt wel of geen spanning van de computer. Daarom wordt de lamp aangesloten op de uitgang van de computer.

We willen nu dat de lamp gaat branden als de brandstof bijna op is. De voeler moet dan, iets, boven de bodem van de tank gemonteerd worden.

\* Vraag 1: waarom?

-----  
1.  
-----

Ook moeten we, door een programma, aan de computer mededelen wat er moet gebeuren.

We gaan eerst bekijken wat de computer moet doen. Daarvoor stellen we een algoritme, een korte opsomming van wat de computer moet doen, op.

ALGORITME:

<1> Meet of er spanning op de ingang staat.

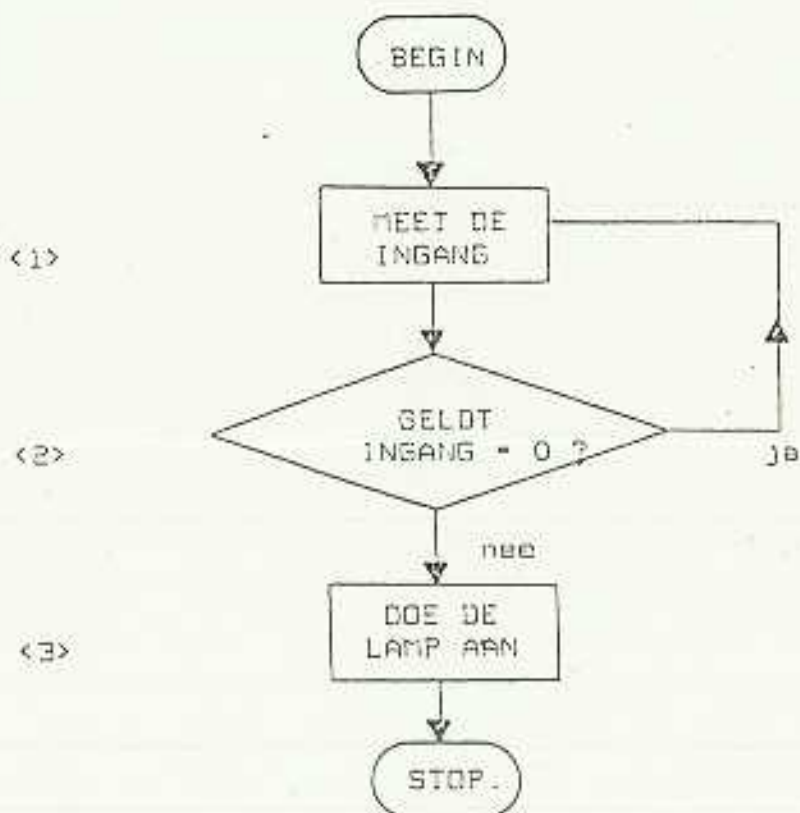
<2> Vergelijk en beslis of er opnieuw gemeten moet worden.

<3> Doe de lamp aan.



Uit dit algoritme maken we een stroomdiagram.

STROOMDIAGRAM:



We moeten dan het stroomdiagram vertalen in een voor de computer "begrijpbare" taal, in ons geval Exbasic en Turbo-Pascal. Die, voor de computer begrijpbare, opdrachten noemen we een programma.

Het "programma" wordt dan:

Exbasic:

```
10 Begin
20 meet v
30 while v=0
40 meet v
50 wend
60 doe de lamp aan
70 END
```

Turbo-Pascal:

```
Program meten;
var V: integer;

begin
  MEET V;
  while V=0 do MEET V;
  DOE DE LAMP AAN
end.
```

Natuurlijk kent de computer de commando's "MEET V" en "DOE DE LAMP AAN" niet. Hoe we dit oplossen zien we in het volgende hoofdstuk.

## HOOFDSTUK 2:

### De Input/Output-unit.

We kunnen de voeler en de lamp niet rechtstreeks op de computer aansluiten. We gebruiken daarom een kastje dat de Invoer (Engels: Input) en de Uitvoer (Engels: Output) van de informatie mogelijk maakt. Het zet signalen uit de buitenwereld om in spanningen voor de computer of omgekeerd.

Zo'n kastje noemen we een Input/Output-unit of korter I/O-unit. (fig 8<sup>a</sup> en 8<sup>b</sup>)

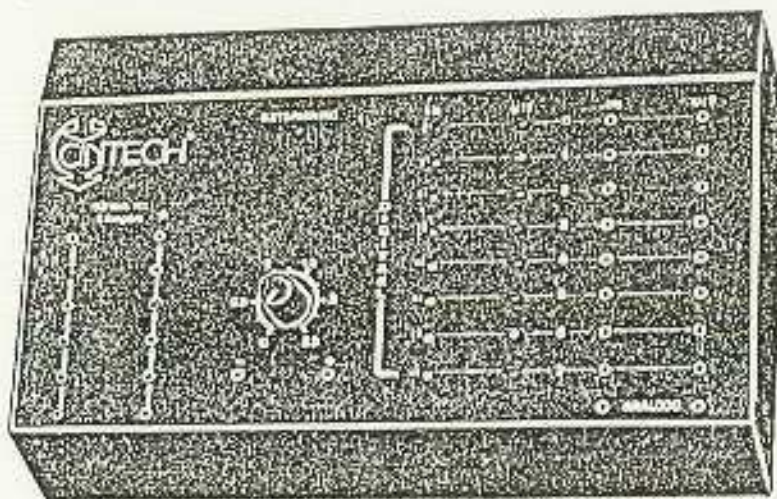
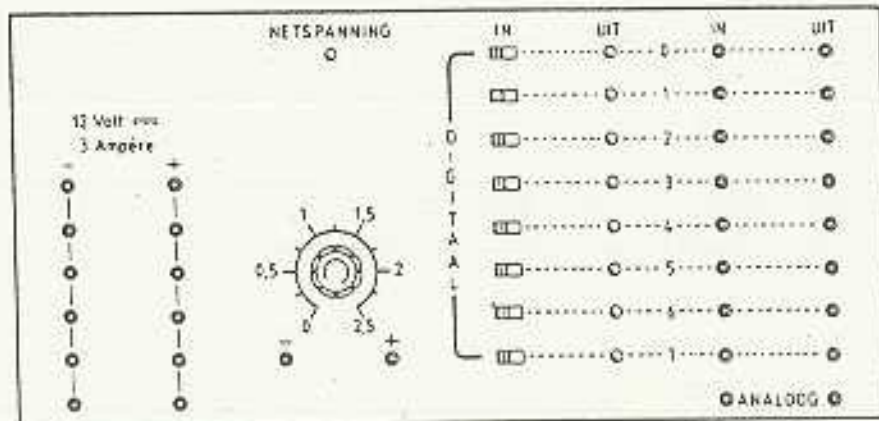


fig.8

Op de Ingang van de I/O-unit kunnen allerlei opnemers (sensoren) aangesloten worden. Ook kunnen we apparaten, die we willen sturen, zoals de lamp, op de Uitgang van de I/O-unit aansluiten.

Hoe de I/O-unit werkt gaan we in stapjes bekijken.

\* Het aanzetten van de I/O-unit.

Achter op de I/O-unit zit een schakelaar om het kastje aan en uit te zetten. Staat de I/O-unit aan dan zal het lampje boven op het kastje, waar "netspanning" staat, gaan branden.

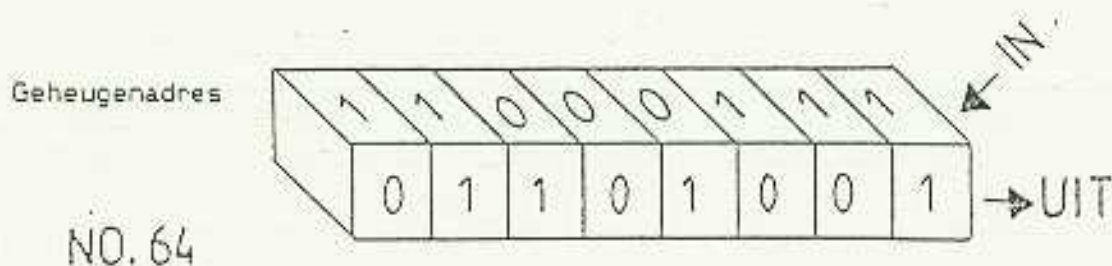


fig.9

\* Opdracht 1: Zet de I/O-unit aan.

De computer reserveert nu enkele geheugenadressen, die elk 2 geheugenplaatsen, laden met 8 vakjes, bevatten om informatie uit te wisselen met de I/O-unit. In ons geval zijn dat de geheugenadressen met nummer 64 en 65. (Adresnummer is afhankelijk van welke computer gebruikt wordt.)

In fig. 10 is zo'n geheugenadres geschetst.



Bi-directioneel uitgevoerd.

fig.10

Op de I/O-unit wordt geheugenplaats 64 (IN) voorgesteld door de 8 groene aansluitingen, genummerd 0 t/m 7 onder Digitaal IN. Zo wordt de geheugenplaats 64 (UIT) voorgesteld door de 8 groene aansluitingen, genummerd 0 t/m 7 onder Digitaal UIT.

In ons voorbeeld met de voeler en de lamp zou de voeler, willekeurig, aangesloten kunnen worden op ingang nummer 5 en de lamp op uitgang nummer 2. De opstelling ziet er dan, schematisch, als volgt uit:

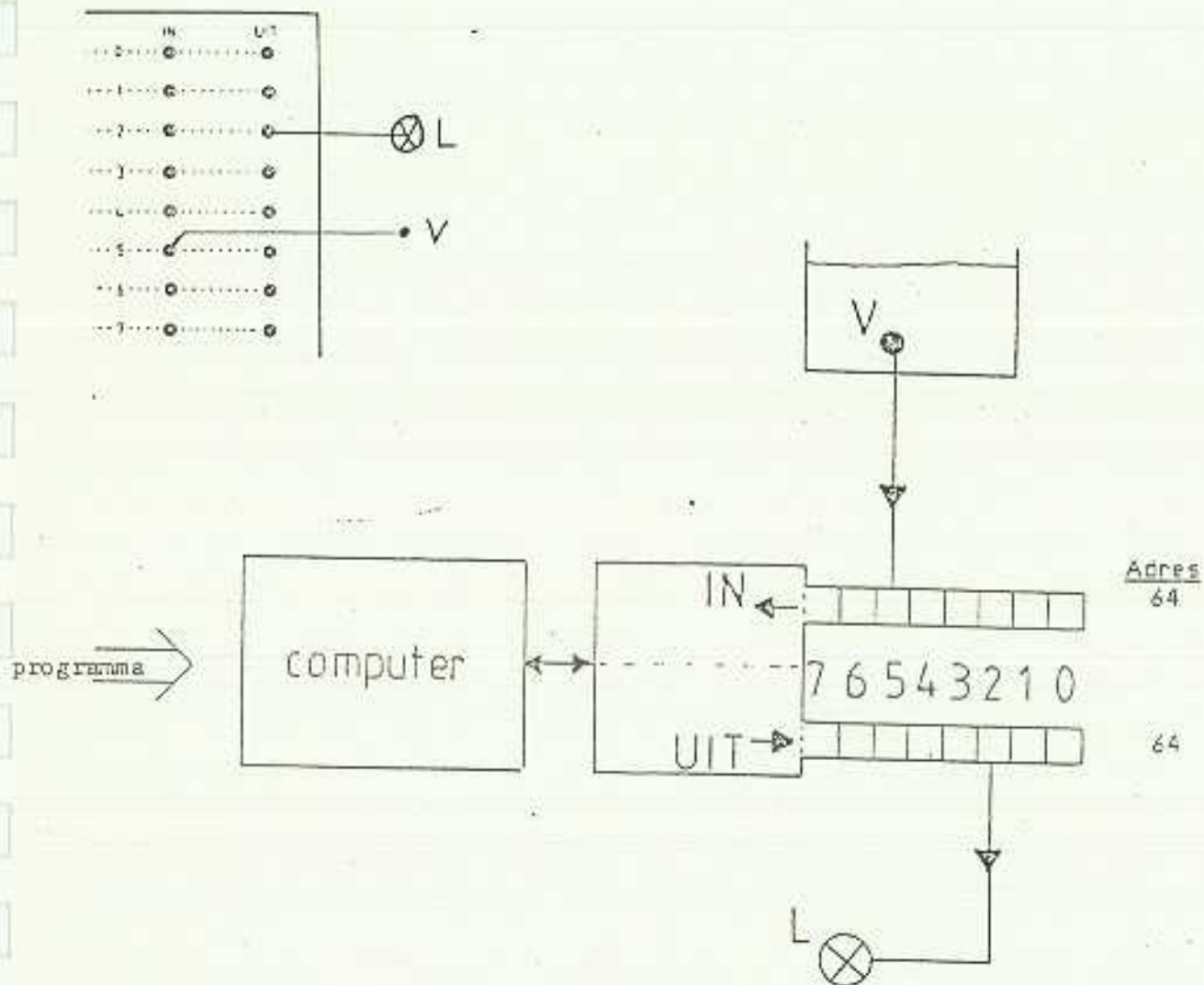


fig.11

Op dit moment schiet onze kennis te kort om dit voorbeeld verder uit te werken. Daarvoor is het nodig dat we de mogelijkheden, en onmogelijkheden, van het Cintech-materiaal beter kennen. Daarom gaan we eerst die mogelijkheden eens nader bekijken.

Er zitten twee voedingen op de I/O-unit, zie figuur 12.  
 Voeding (1) heeft 6 aansluitingen en geeft een constante spanning van 12 Volt.  
 Voeding (2) geeft een instelbare, spanning van 0 tot 2,55 Volt.  
 Met deze voedingen kunnen we de sensoren, en andere apparaten, een spanning geven.

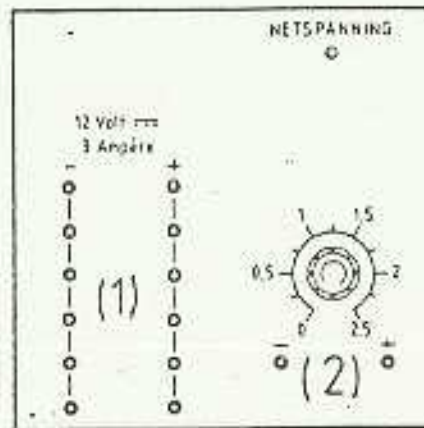
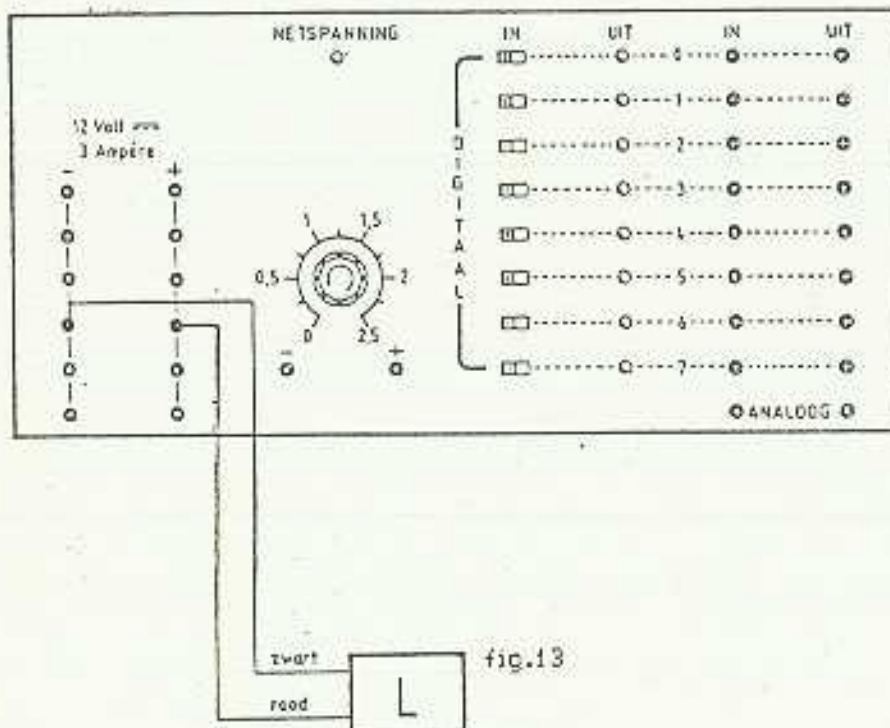


fig.12

Bij de voedingen zijn de RODE aansluitingen altijd de " + " en de ZWARTE de " \_ ".  
Deze kleuren vinden we ook op de sensoren en andere apparaten terug. Sluit altijd de RODE aansluitingen van de randapparaten aan op de RODE aansluitingen van de I/O-unit, met een ROOD snoetje. Zo ook zwart met zwart en groen met groen.

\* Opdracht 2: Sluit de lamp aan op de voeding van 12 Volt.



Op de I/O-unit zitten verder: een rij met 8 schakelaars  
( 2 standen-schakelaars; Aan/Uit of 1/0)

een rij met 8 lampjes (Leds)

de eerder genoemde 8 ingangen  
( om digitale sensoren op aan te sluiten)

de eerder genoemde 8 uitgangen  
( om digitaal te sturen apparaten op aan te sluiten)

Tenslotte is er nog een Analoge Ingang en Analoge Uitgang. (Hierover later meer).

De 8 schakelaars zijn doorverbonden met de overeenkomstige digitale ingangen.  
De 8 digitale uitgangen met de 8 overeenkomstige Led's.

In ons uitgewerkte voorbeeld (zie fig.11) <sup>worden</sup> betekent dit, dat de informatie die de voeler geeft aan de computer, *nagebootst kan* (simuleren) met schakelaar nr. 5. Staat schakelaar nr. 5 naar rechts (Aan = 1) dan komt dat overeen met: de Voeler geeft een signaal.  
Zo geldt ook: als de Lamp AAN is, dan brandt tevens Led nr. 2.  
De schakelaars en Led's samen, wordt ook wel het instructie gedeelte genoemd.

Hoofdstuk 3:

Digitaal meten en sturen.

Voor het digitaal meten en sturen gebruiken we of het instructiegedeelte of de 8 digitale In/Uit-gangaansluitingen.

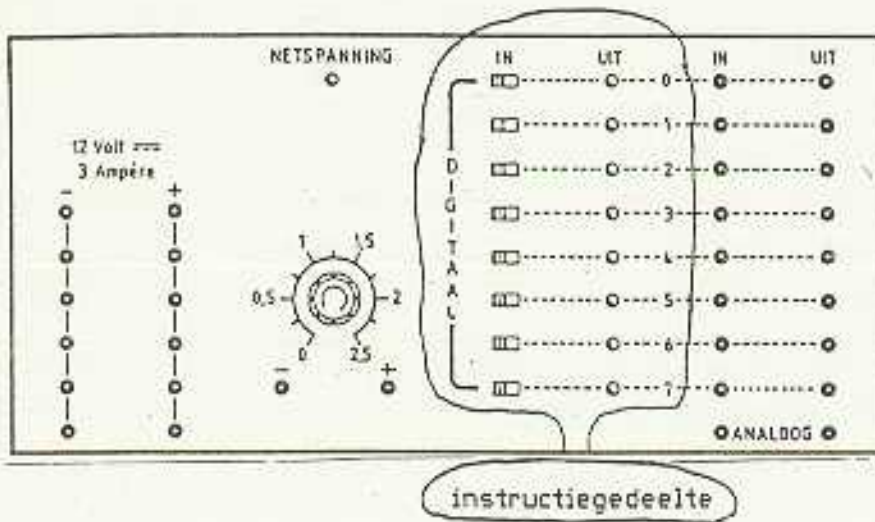


fig.14

De digitale ingangen " IN " kunnen elk maar twee toestanden onderscheiden; er staat wel spanning [ 1 ] op of er staat geen spanning [ 0 ] op.  
De digitale uitgangen " UIT " kunnen elk maar twee signalen afgeven; wel spanning of geen spanning.

De computer reserveert voor het digitale meten en sturen geheugenadres 64.

We zullen eerst eens digitaal gaan meten.

A. \*\* DIGITAAL METEN \*\*

De schakelaars op de I/O-unit zijn doorverbonden met de overeenkomstige ingangen.

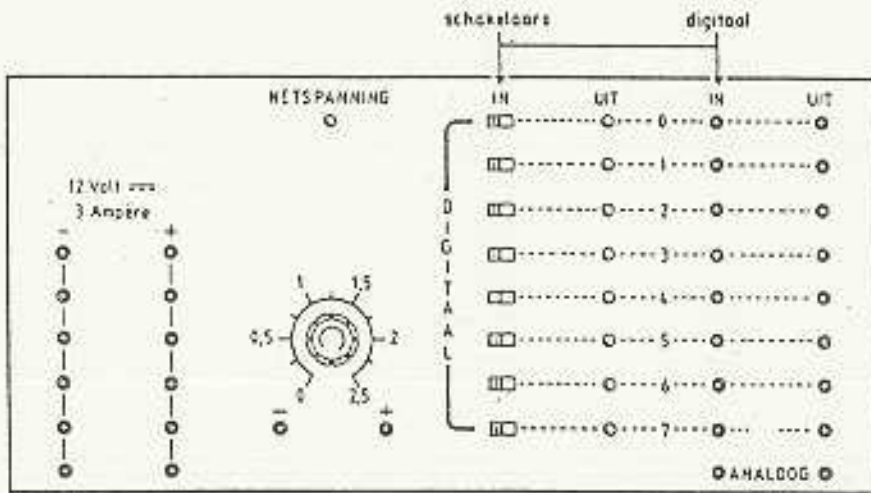


fig.15

Staat bijvoorbeeld schakelaar 4 naar links dan staat op ingang 4 geen spanning. Staat schakelaar 3 naar rechts dan wordt ingang 3 [ 1 ] gemaakt en staat er dus wel spanning op.

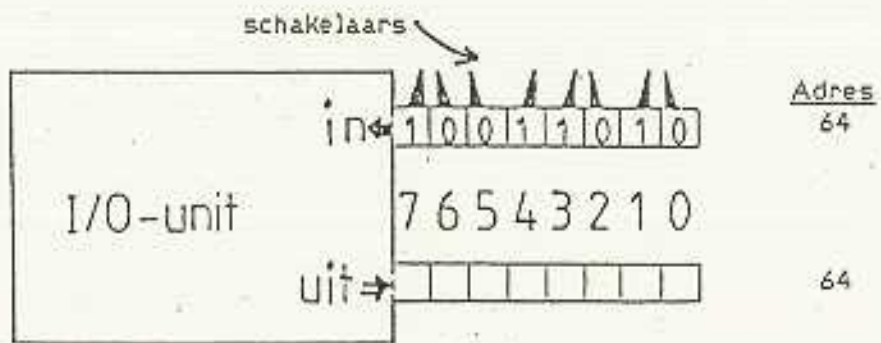


fig.16

\*\*\*\*\*  
 \*  
 \* ZORG ER DAAROM ALTIJD VOOR, DAT ALS JE GAAT \*  
 \* METEN, ALLE SCHAKELAARS NAAR LINKS STAAN !! \*  
 \*  
 \*\*\*\*\*



Voor het meten van de invoer gebruiken we het commando:

Pascal  
X := PORT [ 64 ]<sup>\*</sup>

Exbasic  
X = INP (64)

Dwz. De data op geheugenadres 64 wordt door de computer gelezen en toegekent aan X.

We gaan nu kijken welke waarden de computer leest als we met de schakelaars werken. Daarvoor zullen we een programma moeten maken. Bij het maken van programma's voor meten en sturen is het vaak erg handig om eerst een algoritme op te stellen. Dit is een korte opsomming van wat de computer zou moeten doen. Daarna wordt vaak nog een stroombiagram gemaakt waarin de handelingen van de computer in een schema worden gezet. Een voorbeeld van zo'n algoritme en stroombiagram hebben we al gezien in het voorbeeld met de brandstofvoeler.

Hoewel we hier met eenvoudige metingen en sturingen te maken hebben zullen we bij elk programma een algoritme en een stroombiagram geven/vragen.

\* spreek uit:

:= , krijgt de waarde van  
of  
, wordt

• Experiment 1

Digitaal meten met de schakelaars.

We willen een programma schrijven dat de waarde, die hoort bij de stand van de schakelaar, leest en ze op het beeldscherm afdrukt.

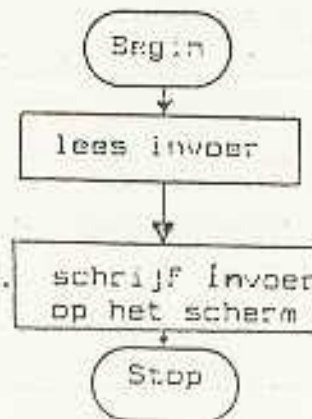
Algoritme

< 1 > lees de invoer en ken de waarde toe aan X

< 2 > schrijf X op het beeldscherm en

< 3 > stop

Stroomdiagram



Het programma wordt dan:

Pascal

```
program meetd;
var X: integer;

begin
  X := port [ 64 ];
  writeln (X)
end.
```

Exbasic

```
10 Begin
20 X= INP(64)
30 Print X
40 End
```

Je kunt het programma nog "versieren" door b.v. het scherm schoon te maken en: in Pascal "writeln(X)" te vervangen door "writeln('De gemeten waarde is : ',X)". Bij Exbasic "30 Print"De gemeten waarde is "; X

• Opdracht 3: Type het programma in.

Om nu dit experiment te doen moeten we een aantal malen het programma doorlopen.  
(runnen)

Zet eerst alle schakelaars naar links.

We beginnen met het bekijken van de waarden die de computer leest als we geen of een schakelaar naar rechtszetten. Van die resultaten maken we een tabel, waar het nummer van de schakelaar, die naar rechts staat, al staat en de gemeten waarde moet worden ingevuld.

\* Opdracht 4: Als je het programma runt krijg je de gemeten waarde te zien. Noteer die waarde in een tabel, zet de schakelaar terug en zet de volgende schakelaar naar rechts. Run het programma tot de tabel af is.

Schakelaar nr.	geen	1	2	3	4	5	6	7	8
gemeten waarde	0					32			

Tabel 1

Vraag 2: Wat valt je hierbij op?

2

Nu gaan we eens kijken welke waarden de computer leest als we 2 of meer schakelaars naar rechts zetten.

\* Opdracht 5: Vul tabel 2 in.

schakelaar nrs.	0,3	5,7	0,1,2	2,4,7	0,2,4,6	0,5,6,7	1,3,5,6,7	alle
gemeten waarde								

tabel 2

\* Vraag 3: Wat valt je nu op als je deze tabel vergelijkt met tabel 1?

3

\* Opdracht 6: Laat op het beeldscherm de waarde 211 afdrukken.

\* Vraag 4: Welke schakelaars moet je daarvoor naar rechts zetten?

4

Misschien is het je opgevallen dat in tabel 1 allemaal machten van 2 ingevuld moeten worden, terwijl in tabel 2 de, overeenkomstige waarden uit tabel 1, bij elkaar opgeteld moeten worden.

De computer leest de schakelaars, en dus ook de ingangen, als BINAIRE getallen en rekent ze om naar DECIMALE getallen. Een uitbreiding van tabel 1 zou dan ook worden:

Tabel 1a:

schakelaar nr.	1	0	1	2	3	4	5	6	7
binair getal	0000,0000	0000,0001	0000,0010	0000,0011	0000,1000	0000,1001	0000,1010	0000,1011	0000,1100
macht van 2	-	$2^0$	$2^1$	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	$2^7$
decimaal gemeten waarde	0	1	2	4	8	16	32	64	128

Als jouw conclusies, uit de tabellen, niet goed waren of als je iets meer wilt weten over binaire getallen, lees dan de bijlage BINAIRE GETALLEN eens door en maak de opgaven die daar in staan.

\* Experiment 2.

In de praktijk werken we niet met de schakelaars. We gebruiken dan SENSOREN, die wel of geen spanning geven. Een fotocel is zo'n sensor; hij geeft spanning [ 1 ] als er, voldoende, licht op valt en geeft geen spanning [ 0 ] als er "geen" licht op valt.

In onderstaande afbeelding is een fotocel aangesloten op de digitale ingang nr.3. Als lichtbron gebruiken we een infra-rood zender, die een, voor mensen, onzichtbaar licht uitzendt. Zo'n zender wordt ook vaak gebruikt als afstandbediening voor een T.V.

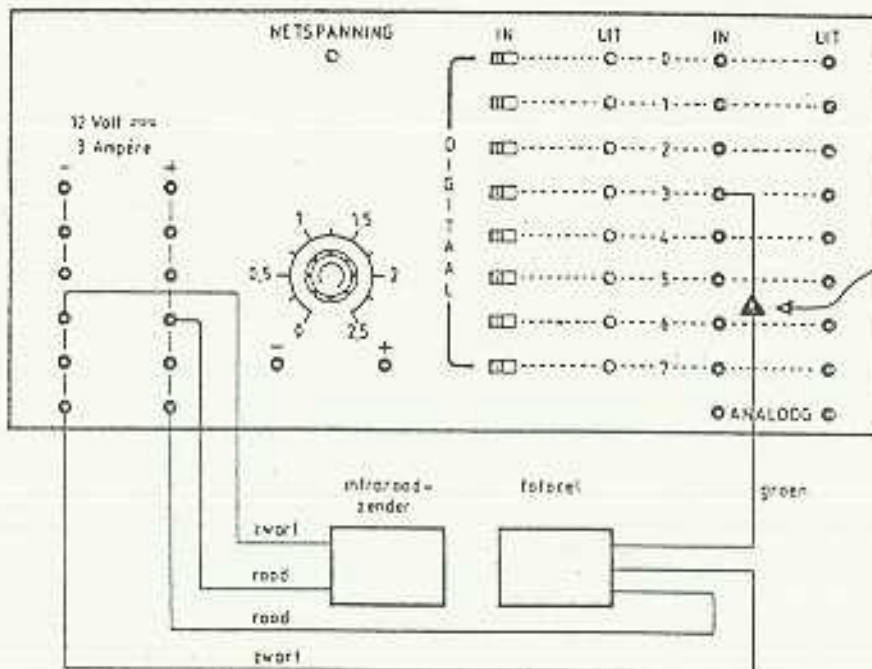


fig.17

\* Opdracht 7.: Bouw deze opstelling na en zorg er voor dat de schakelaars allemaal naar links staan.

\* Vraag 5.: Welke waarde, denk je, zal de computer lezen als er licht op de fotocel valt?

Wel licht.	Geen licht.
5	

Probeer dit eens uit, m.b.v. het programma uit exp.1.  
Gebruik ook eens een andere digitale ingang-aansluiting.

We zetten dit experiment voort:

We willen nu een programma schrijven, dat de gemeten waarde op het scherm schrijft en stopt als we schakelaar nr.7 naar rechts zetten.

\* Vraag 6.: Voor welke gemeten waarde(n) moet dit programma dus stoppen?

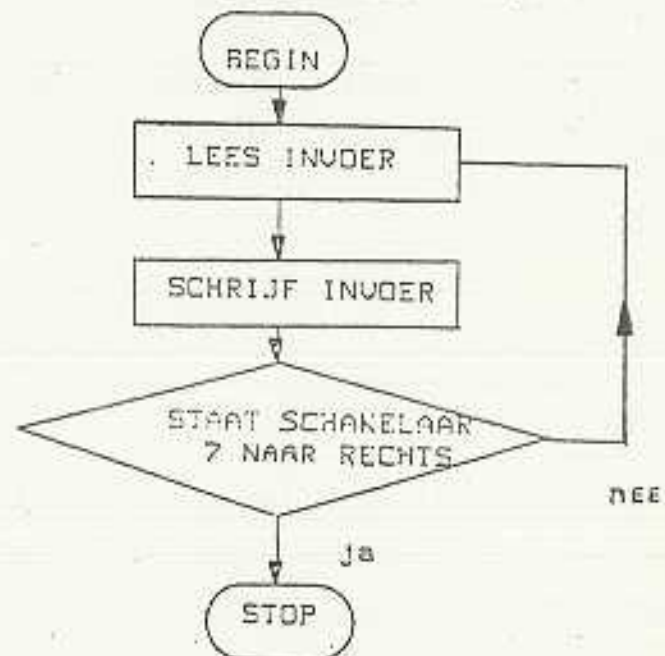
6.

Eerst stellen we weer een algoritme op en maken daarna een stroomdiagram.

Algoritme

- <1> Lees de invoer
- <2> Schrijf invoer op het scherm
- <3> Kijk of schakelaar nr.7 naar rechts staat en beslis of de invoer opnieuw gelezen moet worden
- <4> Stop

Stroomdiagram



\* opdracht 8.: Schrijf nu zelf het programma en type het in. Run het, terwijl er licht op de fotocel valt.

\* Vraag 7.: Welke waarde schrijft de computer op het scherm?

7

\* Opdracht 9.: Onderbreek de lichtstraal, door je hand tussen de infra-roodzender en de fotocel te houden.

\* Vraag 8.: Welke waarde wordt nu op het beeldscherm geschreven?

\* Vraag 9.: Op welke ingang moeten we de fotocel aansluiten als we het getal 64 op het scherm willen laten schrijven?

\* Opdracht 10.: Sluit de fotocel nu zo aan dat het getal 64 wordt afgebeeld. Run het programma en kijk wat de uitvoer is. Onderbreek de lichtstraal en bekijk de uitvoer. Stop daarna het programma door schakelaar 7 naar rechts te zetten.

B. \*\* DIGITAAL STUREN \*\*

Op de I/O-unit zijn de digitale-uitgangen doorverbonden met de 8 lampjes. Als er op uitgang nr. 5 een spanning, [ 1 ], staat dan zal lampje 5 branden. Staat er op uitgang 4 geen spanning, [ 0 ], dan zal lampje 4 niet branden. ( zie fig. 18 )



fig. 18

Voor het sturen van de uitvoer gebruiken we het commando:

Pascal

Exbasic

PORT [ 64 ] := getal

OUT 64, getal

Het 'getal' heeft een decimale uitdrukking.  
De computer stuurt dan de waarde van het getal, binair, naar de I/O-unit. De waarde die het getal kan hebben ligt tussen 0 en 255. ( Bij een 8-bits computer )

$$0 \leq \text{getal} \leq 255$$

\* Vraag 10. Waaron moet die waarde van het getal tussen de 0 en 255 liggen?

10



### \* Experiment 3

#### Digitaal sturen van de lampjes.

Eerst gaan we eens kijken welke lampjes, op de I/O-unit, gaan branden als we een getal intikken op het toetsenbord.

We gaan daarvoor een programma schrijven, door eerst een algoritme en een stroomdiagram te maken.

#### Algoritme

<1> Tik het getal in

<2> zet deze data op de uitgang

<3> stop

#### Stroomdiagram



Het programma wordt dan:

#### Pascal

```
program stuurdig;  
var getal : integer;
```

```
begin  
  readln(getal);  
  port[64] := getal;  
end.
```

#### Exbasic

```
10 Begin  
20 Input "voer getal in ";getal  
30 out 64,getal  
40 end
```

Het Pascal programma kun je verfraaien door voor "readln(getal)" de opdracht "WRITE ('Geef een getal tussen 0 en 255 ')" in te tikken.

\* Opdracht 11.: Type het programma in.

Willen we duidelijk zien welke lampjes de computer aan zet en waarom, dan zullen we dit programma een aantal malen moeten doorlopen.

Om nu een duidelijk beeld te krijgen hoe de uitvoer in zijn werk gaat zullen we een tabel bijhouden. Daarin schrijven we, onder het bijbehorende lampje, een 1 als het lampje brandt en een 0 als het niet brandt. De getallen die je in moet tikken staan voor de tabel.

\* Lezen d.w.z. de computer kijkt welke data er op adres 64 staat. (Input)

\* Schrijven d.w.z. de computer zet data op adres 64. (Output)

• Opdracht 12: Run het programma, voer de waarde in en noteer in de tabel welke lampjes branden.

LAMPJE BRANDT WEL : 1

LAMPJE BRANDT NIET : 0

Tabel 3.

lamp nr. getal	7	6	5	4	3	2	1	0
vb. 0	0	0	0	0	0	0	0	0
1								
2								
4								
8								
16								
32								
64								
128								
3								
24								
37								
48								
73								
111								
127								
150								
198								
250								
254								
255								

\* Vraag 11.: Wat valt je op als je de sturing vergelijkt met het meten?

11

\* Vraag 12.: Welke lampjes zullen gaan branden als je het getal 142 intikt?  
En welke als je 200 intikt?

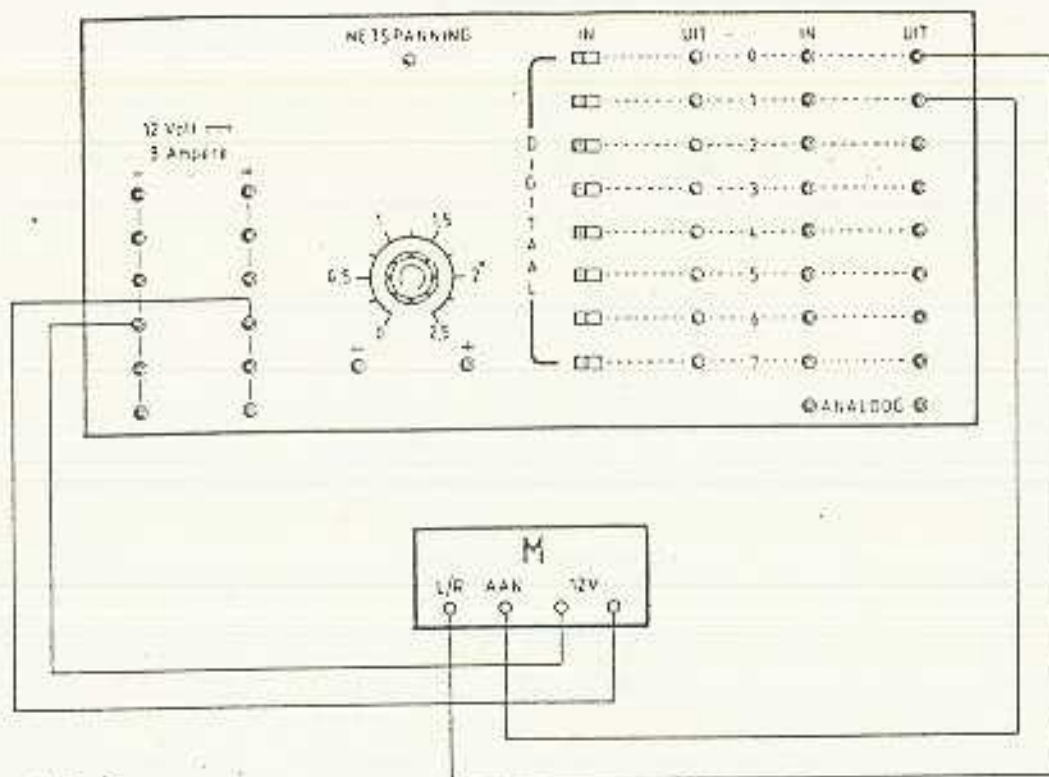
12

Misschien is het je opgevallen dat het sturen met de I/O-unit met dezelfde waarden gebeurt, als het meten met de I/O-unit. Die waarden kunnen we met het toetsenbord invoeren, maar ook in een programma verwerken. Een voorbeeld daarvan zullen we in het volgende experiment gaan bekijken.

## Experiment 4

### Digitaal sturen van een motor

Hiervoor gaan we eens kijken hoe we de motor van de draaischijf aan kunnen sturen. Sluit daarvoor de motor aan, zoals aangegeven in figuur 19.



Op het kastje van de draaischijf zitten 4 aansluitingen. De rechte twee aansluitingen 12V dienen om de voeding er op aan te sluiten. Verbind de rode aansluiting van de voeding, "+", en de zwarte met de "-". De tweede aansluiting van links, "AAN", dient ervoor om de draaischijf aan te sturen, terwijl de linkse aansluiting, "L/R", ervoor dient om de draaischijf zowel links-om, met de klok mee, als rechtson, tegen de klok in, te laten draaien.

\* Vraag 13: Welk getal moet je uitsturen om de draaischijf aan te zetten?

\* Opdracht 13.: Tik dit getal in, in het programma dat we gebruikt hebben bij het digitaal sturen van de lampjes. (exp.3)  
Kijk daarbij of de draaischijf rechts of links om draait.

Antwoord: De draaischijf draait RECHTS / LINKS om.

(streep het foutieve antwoord door)

\* Vraag 14.: WELK getal moeten we uitsturen om de draaischijf de andere richting op te laten draaien?  
Bedenk daarbij dat de motor natuurlijk aan moet blijven staan!!

14

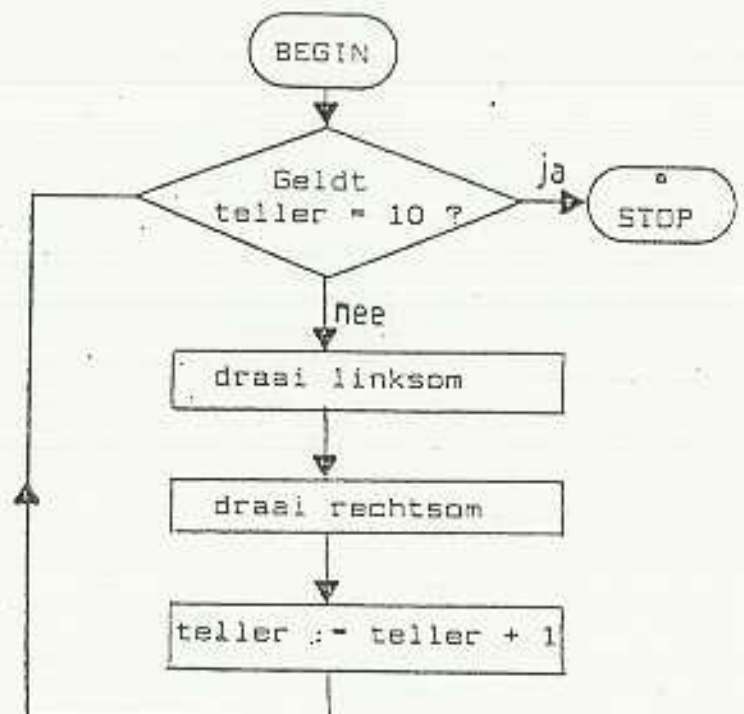
\* Opdracht 14.: Controleer dit getal door het weer in het programma in te tikken.

\* Opdracht 15.: Schrijf een programma waardoor de draaischijf, 10 keer, eerst enkele seconden linksom en dan enkele seconden rechtsom draait.  
Daarvoor gaan we eerst weer een algoritme en een stroomdiagram opstellen.

Algoritme

- <1> Kijk of de schijf 10 keer linksom en rechtsom heeft gedraaid en beslis of je stopt
- <2> Laat de schijf linksom draaien;
- <3> Laat de schijf rechtsom draaien;
- <4> Verhoog de teller met 1 en ga naar <1>.

Stroomdiagram



De teller moet in 't begin wel op 0 gezet worden.

Hieronder staat de opzet van het programma. Vul de open plaatsen in:

Exbasic

```
10 Begin
20 Teller=0
30 While teller < 11
40 out 64,....
50 for n=1 to 100:next n
60 out 64,0
70 out 64,....
80 for n=1 to 100:next n
90 out 64,0
100 TELLER=TELLER + 1
110 WEND
120 End
```

Pascal

```
program draaien;
var teller : integer;

begin
  TELLER :=0;
  while TELLER <....do
  begin
    for L :=0 to 5000 do PORT[64] :=....;
    for R :=0 to 5000 do PORT[64] :=....;
    TELLER :=TELLER +1
  end;
end.
```

\* Opdracht 16.: Tik het programma in en laat het runnen.

\* Experiment 5

Looplicht.

We gaan nu een looplicht maken op de I/O-unit. Daarbij doen we lampje 0 ongeveer 2 seconden aan en doen het daarna weer uit. Vervolgens lampje 1, lampje 2, lampje 3, enzovoorts.

\* Vraag 15.: We willen in het programma het commando OUT 64,G of PORT[64] :=G steeds gebruiken. Daarbij moeten we G een waarde geven, die we steeds moeten aanpassen.

Welke decimale waarde kan G hier zoal aannemen? Kun je deze decimale waarden ook schrijven als een macht van 2?

15

We willen de waarde van G ook gebruiken om het programma te laten stoppen.

\* Vraag 16.: Voor welke waarde van G moet het programma dan stoppen?

16

\* Opdracht 17.: Schrijf een programma voor het looplicht. Gebruik voor het getal dat we uitsturen de letter G en pas die steeds aan. Schrijf eerst het algoritme en maak een stroomdiagram, voordat je het programma gaat schrijven. Type het programma in en laat het runnen.

\* Opdracht 18.: a) pas opdracht 17 zo aan dat het looplicht 10 keer "loopt"

b) maak m.b.v. de leds (0,1,2) en de leds (5,6,7) een verkeerslichtregeling.  
Maak daartoe de tabel af:

Verkeerslicht 1	groen	oranje	rood	rood	rood	rood
Verkeerslicht 2	rood	rood	rood	groen	oranje	rood
Decimale waarde						

## Hoofdstuk 4

### Analoog meten en sturen

Bij het digitaal meten ging het er om, om 2 toestanden te kunnen onderscheiden: de voeler (V) geeft wel spanning [1] of geen spanning [0], er wordt wel licht [1] of geen licht [0] gemeten.

Als we bijvoorbeeld met een computer de temperatuur in een kamer willen regelen, zullen we de temperatuur natuurlijk moeten meten. Het is dan natuurlijk onzin om te zeggen: "Er is wel temperatuur" of "Er is geen temperatuur". Daarom kunnen we de temperatuur niet digitaal meten. Van een "temperatuursignaal" zullen we oneindig veel toestanden (waarden) moeten kunnen onderscheiden. Zo'n signaal wordt een analoog-signaal genoemd. Daarvoor zitten er op de I/O-unit 1 ingang en 1 uitgang, die geschikt zijn voor analoog meten en sturen.

Voor het analoog meten en sturen gebruiken we de onderste in- en uitgangen, waar "ANALOG" tussenstaat.



fig. 20

#### Analoog meten:

Met de analoge ingang kunnen we een spanning van 0 tot 2,55 Volt meten. De I/O-unit zet deze spanningen om in getallen: 0 tot en met 255.

#### Analoog sturen:

De computer geeft aan de I/O-unit een waarde af van 0 tot 255, die door de I/O-unit omgezet wordt in een evenredige spanning, op de analoge uitgang, die ligt tussen 0 en 2,55 Volt.

Voor het analoog meten en sturen reserveert de computer geheugenadres 65.

Eerst gaan we analoog meten.



## A. \*\* ANALOGG METEN \*\*

Voor het analoge meten gebruiken we het commando:

Pascal

X := PORT [ 65 ]

Exbasic

X = INP (65)

De computer leest dan de waarde van de analoge invoer en kent de waarde van die invoer toe aan X.

We gaan eerst eens kijken welke waarden de computer leest als we de analoge voeding ( 2 ) aansluiten op de analoge ingang. Daarvoor gaan we de opstelling maken, zoals geschetst in figuur 21.

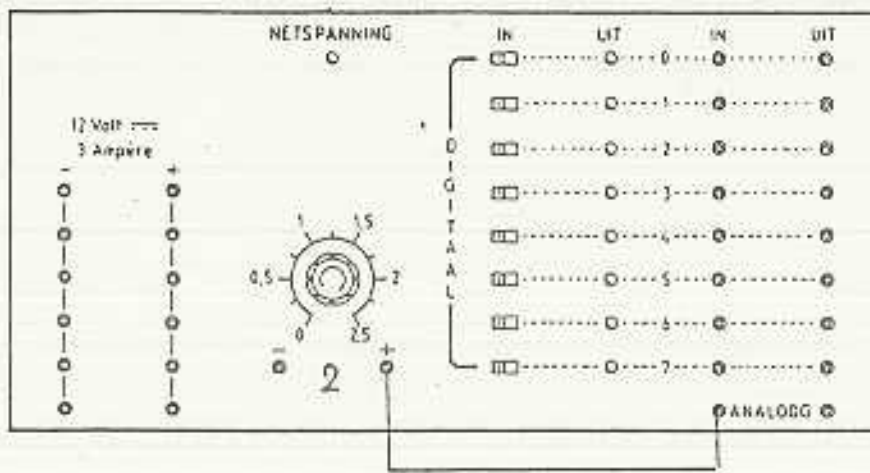


fig. 21

\* Oopdracht 19.: Sluit de analoge voeding aan op de analoge ingang, zoals getekend in figuur 21.

Het analoge signaal, de spanning, wordt via een Analoog-Digitaal-Omzetter omgezet in een binaire-code:

Een spanning van 0 Volt wordt omgezet in de code 00000000 en op het scherm afgebeeld als het decimale getal 0.

Een spanning van 2,55 Volt wordt omgezet in de code 11111111 en op het scherm afgebeeld als het decimale getal 255.

(\* AD-Converter)

```

*****
*
* spanning      binair      decimaal
* 0 V          00000000 - 0
* 2,55 V      11111111 - 255
*
*****

```

Als we nu willen weten welke waarde de computer leest, dan zullen we een programma aan de computer moeten geven. Voor dit programma gebruiken we het algoritme en het stroomdiagram, dat we ook voor het digitale meten hebben gebruikt. (zie blz.15)

- Het programma wordt dan echter toch anders, omdat we nu analoog meten, en dus een ander geheugenadres moeten lezen.

\* Opdracht 20.: Schrijf nu zelf een programma, dat 500 keer de analoge uitgang leest en de gelezen waarde op het scherm afbeeldt. (Maak eerst een algoritme en een stroomdiagram)

\* Opdracht 21.: Type het programma in en run het, terwijl je de instelling van de analoge voeding ( 2 ) verandert, door met de instelknop te draaien. Kijk welke waarden de computer afbeeldt, als je de knop, langzaam, van 0 Volt naar 2,55 Volt (= zover mogelijk rechtsonder) draait en terug.

Opm.: Het tussenvoegen van een tijdlus geeft een wat rustige uitvoer naar het beeldscherm.

Tijdlus: FOR N= 1 to 200:NEXT N

\* Vraag 17.: Welk getal wordt door de computer op het beeldscherm afgebeeld, als we de voeding op 1 Volt instellen?

17
----

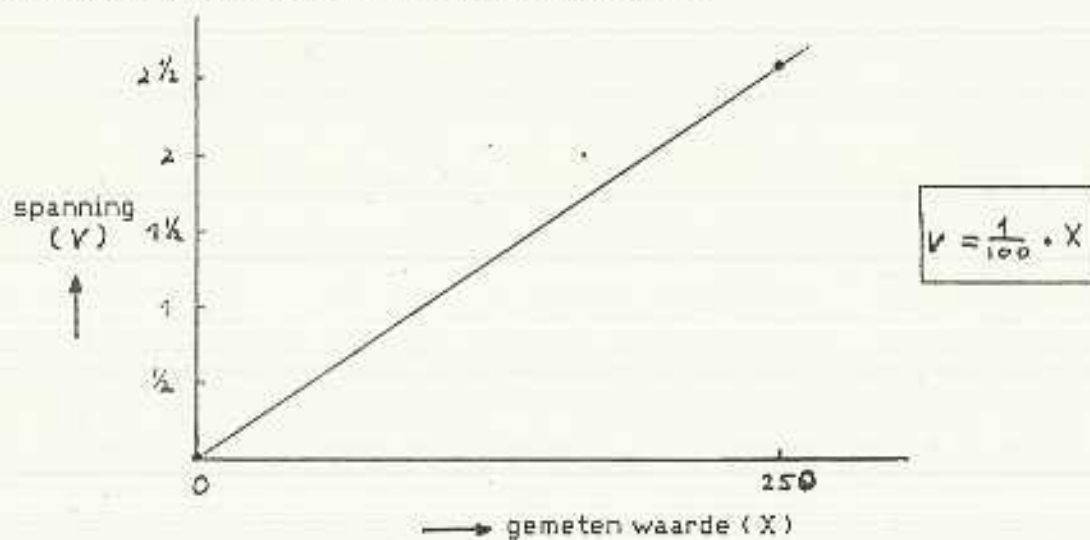
Wanneer nu de gemeten waarde (= het afgebeelde decimale getal [ x ]) wordt omgerekend naar spanning ( V ) dan hebben we een Voltmeter gemaakt.

Opm.: Het bepalen van het verband tussen de gemeten waarde ( X ) en een andere grootheid wordt ijken genoemd.

Hier geldt:

<u>Meetwaarde ( X )</u>	<u>Spanning ( V )</u>
255	2,55 Volt
0	0 Volt

We nemen aan dat het bedoelde verband linear is.



\* Opdracht 22.: Voeg aan het programma uit opdracht 20 de volgende regels toe:

Pascal

V:=0.01 \* X

"writeln ('De spanning is ',V)"

Exbasic

V=0.01 \* X

Print "De spanning is ";V;" Volt"

\* Opdracht 23.: Run het programma en controleer de ijking door de voeding op verschillende waarden in te stellen.

\* Experiment 7 ( Geautomatiseerde thermometer )

Zoals in de inleiding, op het analoog meten, al gezegd werd, zullen we de temperatuur ANALOOG moeten meten. Zouden we de temperatuur namelijk digitaal meten, dan zou de meting zijn: "De temperatuur is hoog genoeg" of "De temperatuur is te laag". Zo'n meting geeft ons vaak te weinig informatie.

Want als we de temperatuur meten, dan willen we graag weten hoe warm het is, hoeveel graden Celcius het is. Dit kunnen we met de I/O-unit doen, door de temperatuursensor op de analoge ingang aan te sluiten.

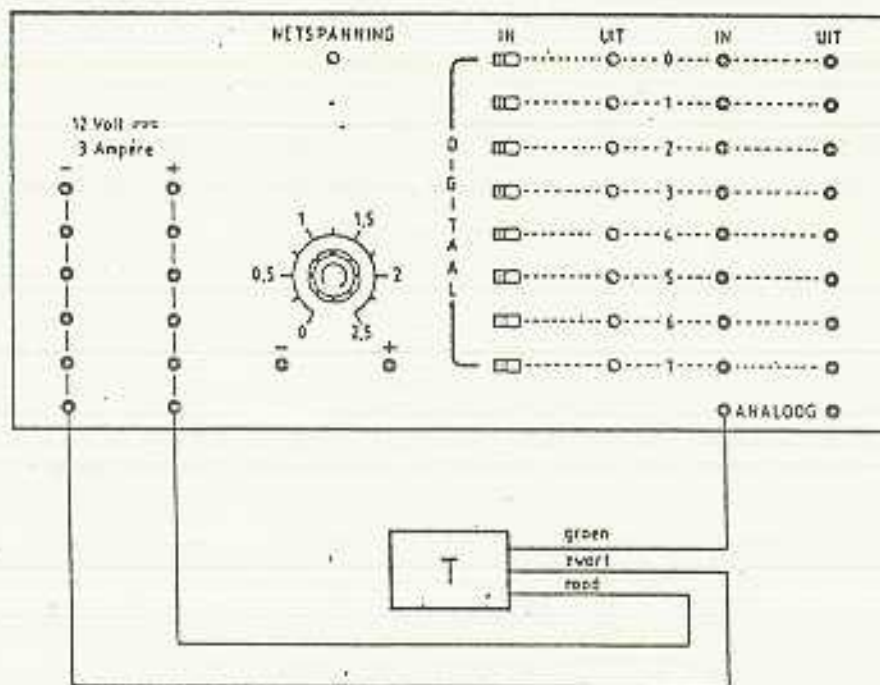


fig.22

\* Opdracht 24:- Sluit de temperatuursensor aan, zoals getekend in figuur 22.

Het gedeelte dat de temperatuur meet zit "in de punt van het staartje". We gaan nu kijken welke waarden de computer op het beeldscherm schrijft, als we dat "staartje" :

1. Op tafel leggen.
2. In de hand nemen.
3. In een bakje met warm water hangen.

Net zo als een normale thermometer heeft ook de temperatuursensor een tijd nodig om de juiste temperatuur aan te geven.

De temperatuursensor kan een spanning van 0 t/m 2,55 Volt aan de computer afgeven.

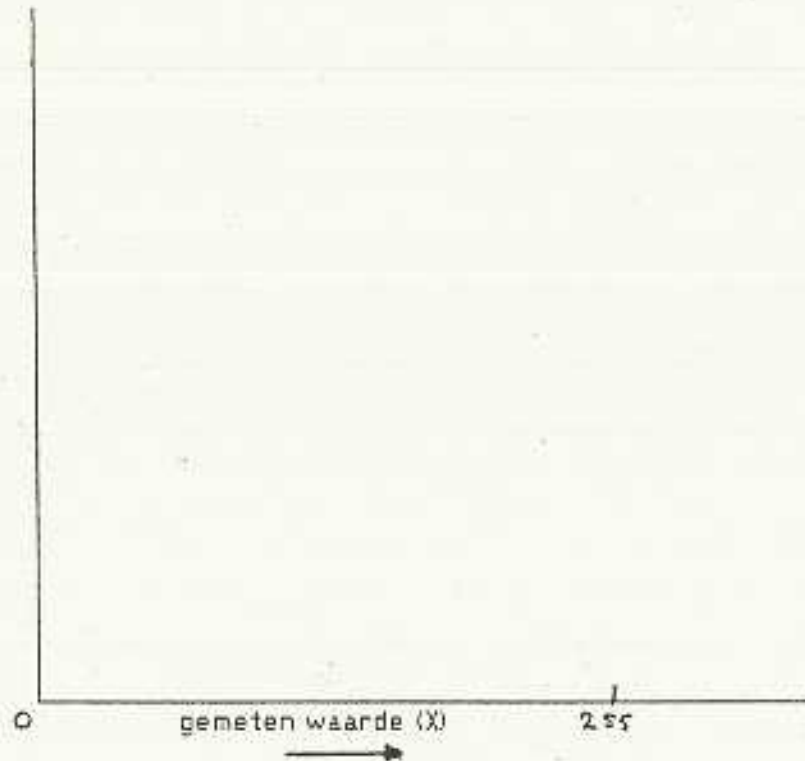
\* Opdracht 25:- Run het programma uit opdracht 20 en noteer, in de onderstaande tabel, welke waarden de computer op het beeldscherm schrijft, als we, achtereenvolgens, de temperatuursensor op tafel leggen, in de hand nemen of met het uiteinde in lauw water hangen.  
Vul de tabel daarbij in.

Tabel:

plaats	meetwaarde (X)
op tafel leggen	
in de hand nemen	
in het water hangen	

Omdat we willen weten met hoeveel graden Celsius deze meetwaarde overeenkomt, zullen we de waarden, die de computer leest, moeten omrekenen. Daarvoor moeten we eerst de temperatuursensor kalibreren, zodat we het verband tussen T en X kunnen vastleggen.

Temperatuur (T)  
in graden C.



### IJken van de temperatuursensor.

De temperatuursensor is lineair; dit wil zeggen dat er een lineair ( of eerste-graads ) verband is tussen de gemeten waarden (X) en de temperatuur (T).

Hoe ijken we de temperatuursensor?

Voor het ijken van een lineaire sensor hebben we twee meetwaarden nodig.

#### Voorbeeld:

Stel dat we twee metingen verricht hebben, door bij twee, bekende, temperaturen te meten met de temperatuursensor.

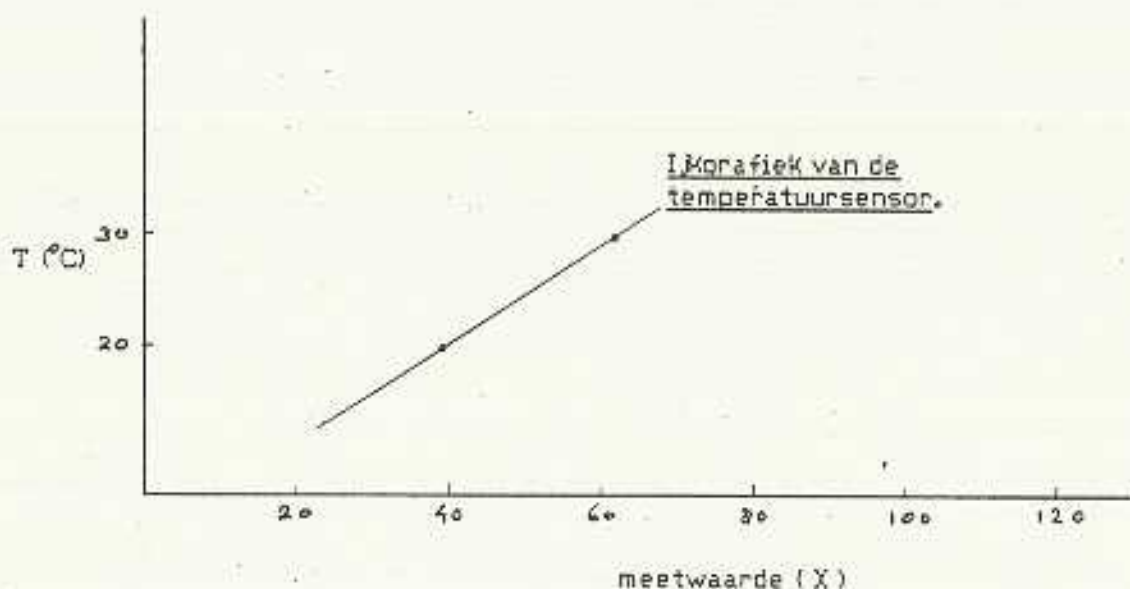
We meten 42 bij een temperatuur van  $20^{\circ}\text{C}$  en  
62 bij een temperatuur van  $30^{\circ}\text{C}$ .

Daarna zetten we de meetwaarden en temperaturen in een tabel:

Meetwaarde (X)	Temperatuur (T)
42	$20^{\circ}\text{C}$
62	$30^{\circ}\text{C}$

Uit die tabel kunnen we een grafiek tekenen, waarin we de temperatuur uitzetten tegen de meetwaarde

Omdat we weten dat het verband tussen de meetwaarden en de temperatuur lineair is kunnen we namelijk een rechte lijn door de twee meetpunten tekenen.



Doordat het verband dus linear is, kunnen we de lineaire IJK-FUNCTIE opstellen:

$$T = A \cdot X + B$$

We zien, in de tabel, dat als de meetwaarde (62-42=) 20 stijgt, de temperatuur (30-20=) 10° C stijgt.

De richtings-coëfficiënt, in de IJK-Functie " A ", wordt dan:

$$A = \frac{30 - 20}{62 - 42} = \frac{10}{20} = 0,5$$

De Functie wordt dan:  $T = 0,5 \cdot X + B$

Door nu een meetwaarde, met zijn bijbehorende temperatuur, in deze vergelijking in te vullen, kunnen we " B " uitrekenen.

Vullen we  $X=42$  en  $T=20$  in dan krijgen we:

$$20 = 0,5 \cdot 42 + B$$

oftewel:  $20 = 21 + B$  <----->  $B = -1$

Als we nu B invullen dan wordt de ijk-functie:

$$T = 0,5 \cdot X - 1$$

Nu we de ijk-functie kennen kunnen we van elke meetwaarde de bijbehorende temperatuur uitrekenen.

\* Vraag 18: Welke temperatuur geeft de temperatuursensor aan, als de meetwaarde 87 is?

18

Ook kunnen we bij een gemeten temperatuur zeggen welke meetwaarde de computer zal lezen.

\* Vraag 19: Welke meetwaarde zal de computer geven als de temperatuur 10° C is?

19

IJken van jouw temperatuursensor.

Nu gaan we "onze eigen" temperatuursensor ijken op de manier die we in het voorbeeld gezien hebben.

Voor het ijken van de temperatuursensor hebben we een aantal hulpmiddelen nodig:

- Een normale thermometer,
- Een bak met koud water en
- Een bak met warm water.

De thermometer dient er voor om de temperatuur van het water te meten,, zodat we de meetwaarden kunnen vergelijken met de werkelijke temperatuur.

\* Opdracht 26.: Laad het programma uit opdracht 20 in je computer. Neem de bak met koud water en plaats de thermometer en, hang het uiteinde van de temperatuursensor in het water. Wacht even zodat de thermometer en de temperatuursensor de juiste temperatuur aangeven. Als de thermometer de juiste temperatuur van het water aangeeft, lees deze temperatuur dan af, en RUN het programma.

\* Vraag 20.: Welke temperatuur geeft de thermometer aan? Welke gemeten waarde komt daarmee overeen?

X	T
20	

\* Opdracht 27.: Neem nu de bak met warm water en voer opdracht 26 opnieuw uit, maar dan met het warme water.

\* Vraag 21.: Welke temperatuur geeft de thermometer aan? Welke gemeten waarde komt daar nu mee overeen?

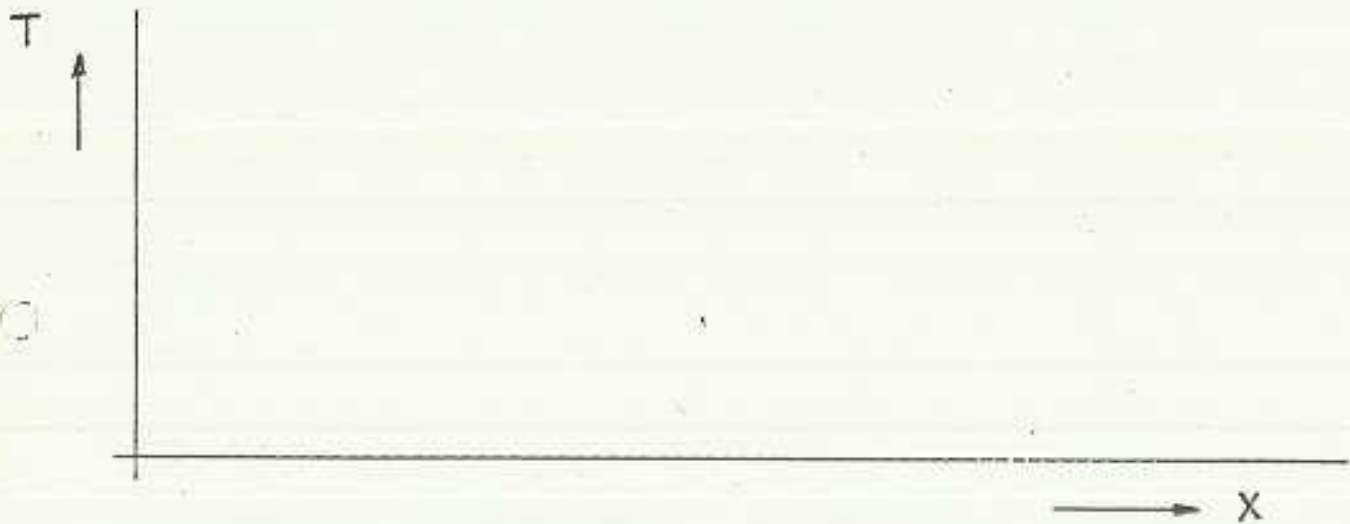
X	T
21	



\* Opdracht 28.: Zet de gemeten waarden, met hun overeenkomende temperaturen in onderstaande tabel.

meetwaarde (X)	temperatuur (T)
-----	-----
-----	-----
-----	-----

\* Opdracht 29.: Teken de IJK-grafiek.



\* Opdracht 30.: Bereken de IJK-functie door van de functie  $T = A \cdot X + B$  de waarden voor A en B uit te rekenen.

Dus  $A = \dots\dots\dots$

$B = \dots\dots\dots$

De IJK-functie wordt dan:

$T = \dots \cdot X + \dots$

We kunnen, nu we de IJK-functie kennen, deze vergelijking natuurlijk opnemen in een programma. Dan geeft de computer direct de uitvoer in graden Celsius. We moeten dan wel een nieuw algoritme en een nieuw stroomdiagram opstellen.

Het algoritme wordt dan:

- < 1 > Lees de analoge invoer en ken de waarde van de invoer toe aan X
- < 2 > Reken de invoer ( X ) om naar de temperatuur ( T )
- < 3 > Schrijf de temperatuur ( T ) op het beeldscherm en
- < 4 > STOP.

\* Opdracht 31.: Schrijf nu zelf het stroomdiagram en het programma, door gebruik te maken van het bovenstaande algoritme.

\* Opdracht 32.: Controleer of je omrekening goed is door op verschillende temperaturen te meten met een normale thermometer en met de temperatuursensor. Vul daarbij onderstaande tabel in.

thermometer	temperatuursensor	verschil

Uit deze tabel kun je nu concluderen of je omrekening van X naar T goed was, door naar de kolom verschil te kijken. Als dat verschil klein of 0 is, is jouw omrekening goed.

\* Vraag 22.: Was jouw omrekening goed of fout? Streep de foutive antwoorden door.

antwoord: Mijn omrekening was GOED / REDELIJK / FOUT.

Als je omrekening redelijk of fout was, doe dan de ijking van jouw temperatuursensor opnieuw.

Was je omrekening goed, dan ben je er in geslaagd om van je computer een THERMOMETER te maken:

PROFICIAT.

\* Opdracht 33.: Maak van de computer een weegschaal.

Benodigheden:

- \* draaihoeksensor
- \* statief, contragewicht met veer, schaal, klemmen.

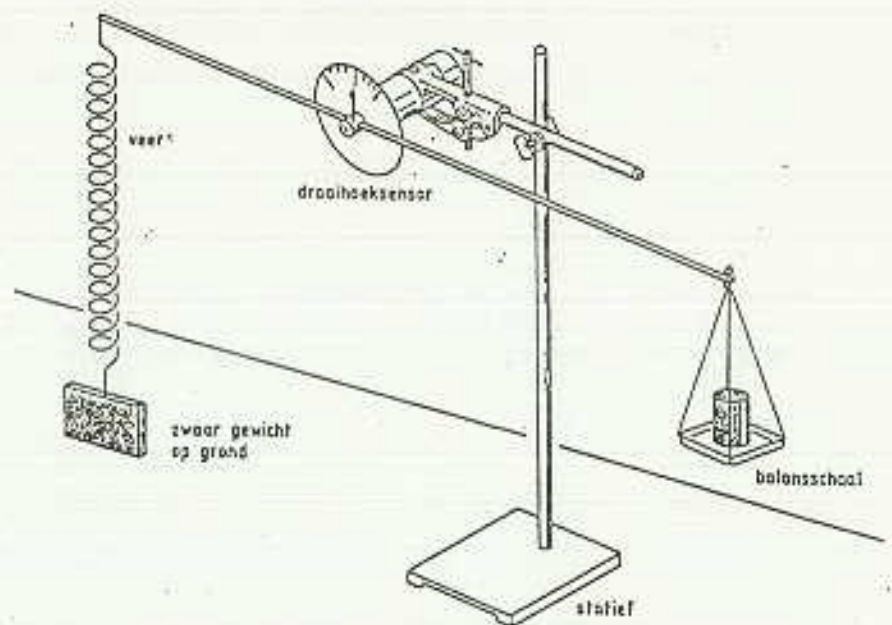


fig. 23

Werkwijze.

- \* Maak de opstelling
- \* Voer de ijking uit. (neem daartoe 2 bekende massa's)
- \* Bepaal m.b.v. programma uit opdracht 20 het verband tussen meetwaarde (X) en gewicht (G)
- \* Pas het programma aan opdat op het scherm het gewicht (G) wordt afgebeeld.

\* Opdracht 34.: Uitbreiding op opdracht 33.

Als op de weegschaal een brief wordt gelegd dient er op het scherm afgebeeld te worden:.....

Op deze brief moet een postzegel van ..... cent.

## B. \*\* ANALOG STUREN \*\*

Voor het analog sturen gebruiken we het commando:

Pascal

PORT [ 65 ] := GETAL

Exbasic

OUT 65, GETAL

met  $0 \leq \text{GETAL} \leq 255$

### \* Experiment 8.

De 'energie' voor een lamp kan d.m.v. de analoge uitgang in 255 stapjes toegevoerd worden.

We sluiten daartoe de energieregel-doos + een lamp van 220V aan op de analoge uitgang. (zie fig.24)

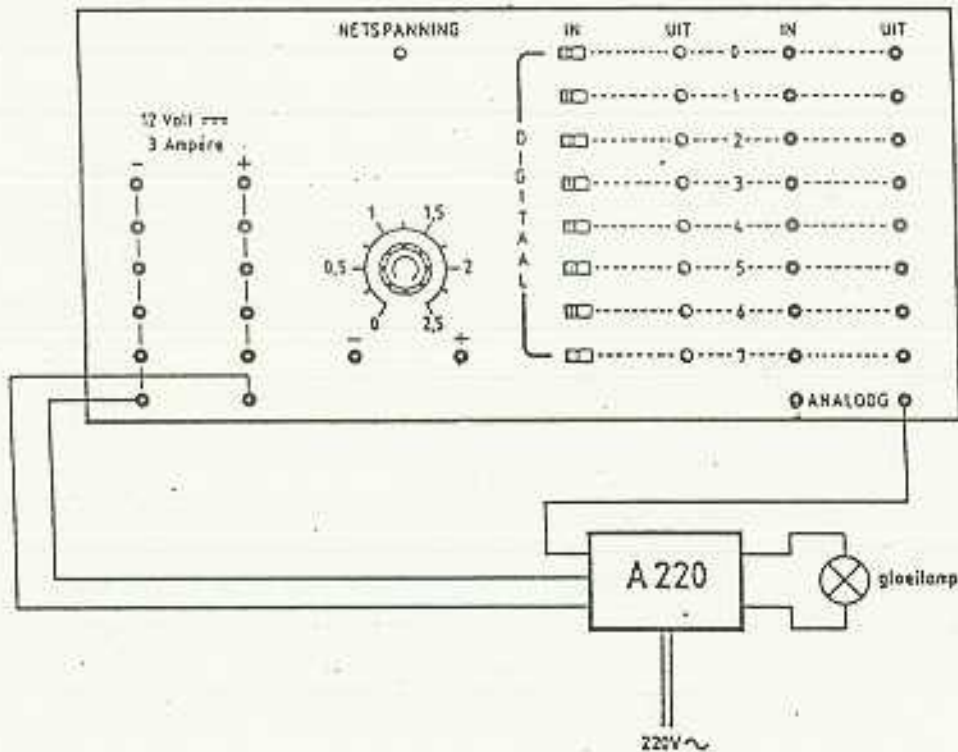


fig. 24

We gaan eerst "het bereik van de lamp" bepalen d.w.z. bij welk getal dat we uitsturen brandt de lamp maximaal en bij welk getal minimaal?

Het programma uit experiment 3 (blz. 22) kan in aangepaste vorm gebruikt worden:

Pascal

```
program stuuran;  
var getal : integer ;  
  
begin  
  readln (getal)  
  port [ 65 ] := getal  
end.
```

Exbasic

```
10 Begin  
20 Input "Voer getal in"; Getal  
30 Out 65 , Getal  
40 End
```

\* Opdracht 35.: Type het programma in.

\* Opdracht 36.: Run het programma, voer voor het 'getal' de waarden uit de tabel in, en onderzoek waar de lamp nog juist brandt.

Tabel.

<u>getal</u>	<u>lamp brandt</u>	
20	niet	
30	....	
40	....	
50	....	
60	....	
70	....	waar minimaal?
80	....	
90	....	
100	....	
110	....	
120	....	
130	....	
140	....	
150	....	
160	....	
170	....	
180	....	
190	....	waar maximaal?
200	....	
210	....	
220	....	
230	....	
240	....	
250	....	

\* Opdracht 37.: Wat gebeurt er na het runnen van het volgende programma?

Pascal

```

program stuuran 1;
var getal:integer;

begin
  getal :=0 ;
  while getal <200 do
  begin
    Port [ 65 ]:=getal
    getal:=getal + 10
    for N:=0 to 200:next N
  end;
end.

```

Exbasic

```

10 Begin
20 Getal = 0
30 While Getal < 200
40 Out 65 , Getal
50 Getal := Getal + 10
60 For N=1 to 200: Next N
70 Wend
80 Out 65 , 0
90 END

```

\* Opdracht 38.

Verander het programma zodanig opdat de lamp van maximaal branden, steeds minder fel gaat branden.

\* Opdracht 39.

Maak een programma dat 10 keer de volgende cyclus uitvoert:  
De lamp is uit, gaat steeds feller branden tot maximaal en tenslotte weer steeds minder fel branden totdat de lamp uit is.

\* Opdracht 40.

Sluit de motorunit + draaischijf aan zoals in fig. 25 is aangegeven.

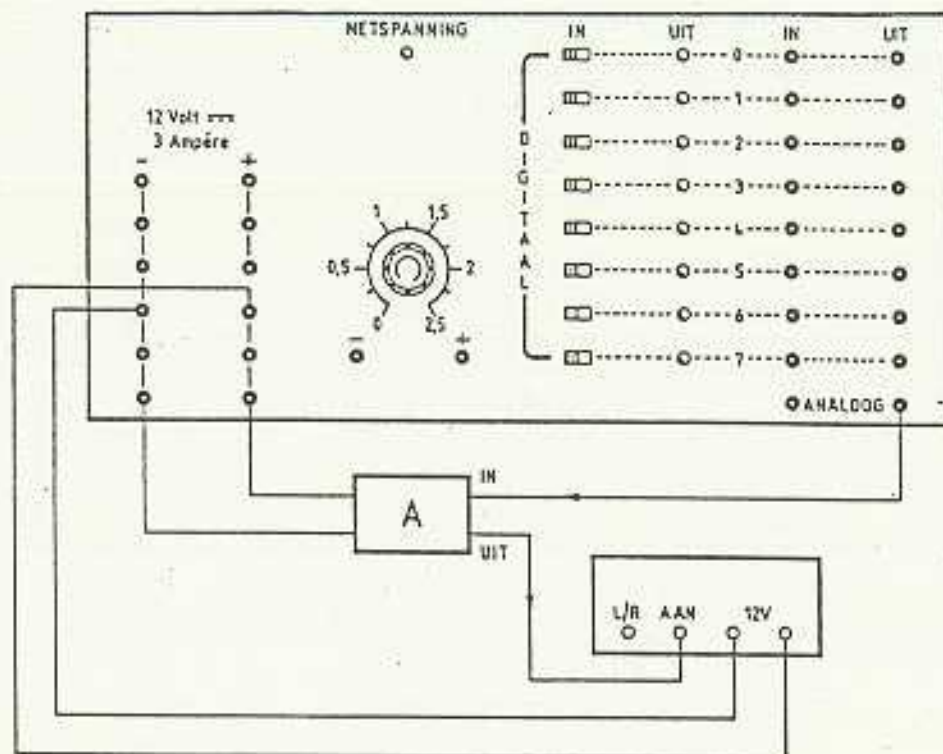


fig. 25

De box A in figuur 25 is niets meer dan een versterker van het analoge signaal. ( De computer kan op de analoge uitgang maximaal 2,55 Volt zetten ) De unit A wordt energieregeling (12V) genoemd.

Vraag: Wat is het "bereik" van de draaischijf?  
(Zie exp.8)

\* Opdracht 41.

Maak een programma opdat de draaischijf vanuit stilstand steeds harder gaat ronddraaien.

\* Opdracht 42.

Maak een programma opdat de draaischijf vanuit stilstand steeds sneller gaat ronddraaien en daarna steeds langzamer en tot stilstand komt.

\* Opdracht 43.

Dezelfde opdracht als in 42, doch met een andere draairichting.  
Aanwijzing.: sluit L/R aan.

## HOOFDSTUK 5:

### Eenvoudige regelinen.

We hebben in ons allereerste voorbeeld (blad 4) gezien hoe de computer afhankelijk van de gemeten waarde (= het signaal van de voeler) wel of niet een signaal uitstuurt (het signaal naar de lamp).

Als ingangsignaal (meetsignaal) en uitgangsignaal (stuursignaal) via een programma aan elkaar gekoppeld zijn spreken we van een regeling.

### Voorbeeld.

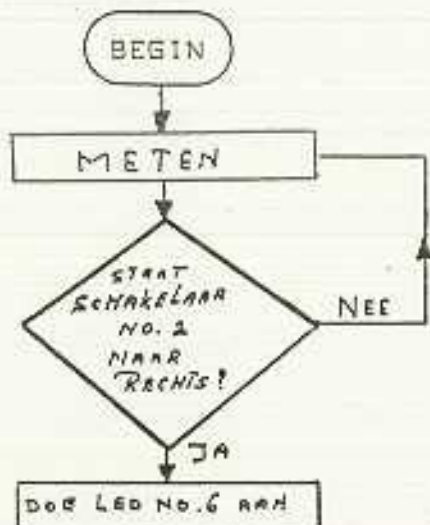
We zetten alle schakelaars op het instructie-gedeelte op uit (0) en gaan het volgende uitvoeren:

Als alleen schakelaar no. 2 naar rechts (1) wordt gezet dient lampje no. 6 te gaan branden.

### Algoritme:

- < 1 > Lees de ingang en ken de waarde toe aan X
- < 2 > zolang  $X < > 4$ , doorlezen
- < 3 > Doe lampje no. 6 aan

### Stroomdiagram



### Exbasic

```
10 Begin
20 X = INP ( 64 )
30 WHILE X < > 4
40 X = INP ( 64 )
50 WEND
60 OUT 64, 64
70 END
```

### \* Opdracht 44.

Run dit programma.

### \* Opdracht 45.

Pas het programma zodanig aan opdat als alleen schakelaar no. 2 naar rechts wordt gezet alle leds gaan branden m.w.v. led no. 0.

### \* Opdracht 46.

Maak een programma waarmee het volgende gebeurt:

Als een schakelaar(s) naar rechts gezet wordt dan gaat de overeenkomstige led(s) branden.



### Experiment 9.

We willen in een ruimte de hoeveelheid licht konstant houden d.w.z. als de lichtinval van buiten meer wordt, mag de lamp minder fel gaan branden en als de lichtinval van buiten minder wordt, dient de lamp juist feller te gaan branden.

#### Benodigheden:

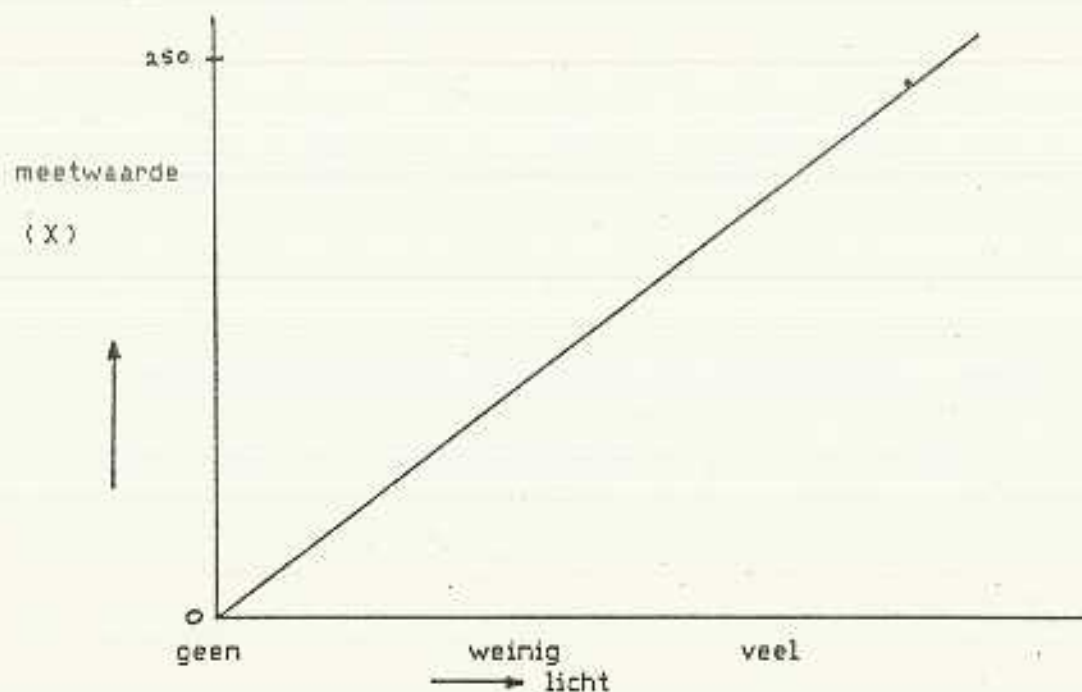
- \* Lichtgevoelige-sensor (L.D.R.) om de hoeveelheid licht van buiten te meten.
- \* Lamp + energieregelingunit (220V) zie exp.8 (blz.41)

#### Opstelling.

Zie exp.8 (blz.41) en figuur 24.

De L.D.R. moet aangesloten worden op een analoge ingang.

Ga na, dat de volgende grafiek geldt voor de L.D.R.



#### \* Opdracht 47.

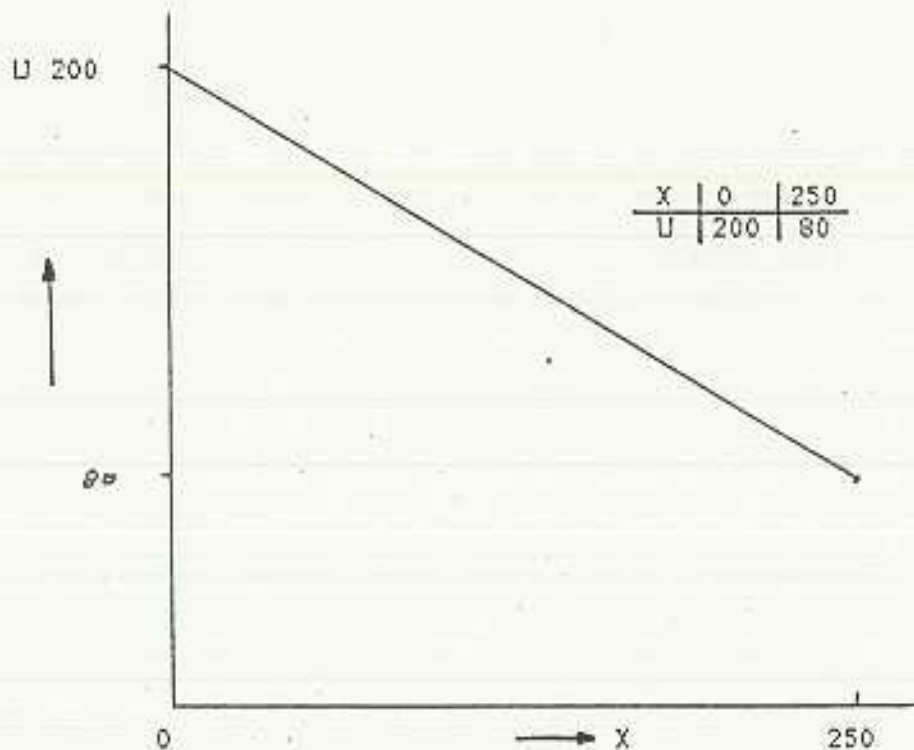
Gebruik het programma uit opdracht 20 en run het.

\* Opdracht 48.

Bepaal het bereik van de lamp. (zie opdracht 36)

We veronderstellen:  $U_{\min} = 80$  en  $U_{\max} = 200$

Het verband tussen de meetwaarde  $X$  (=lichtintensiteit van buiten, gemeten door de L.D.R.) en de waarde  $U$  voor de te regelen lampintensiteit is grafisch:



\* Opdracht 49.

Bepaal het verband tussen  $U$  en  $X$ .

(Aanw.:  $U = A \cdot X + B$ )

Algoritme.

- < 1 > Meet de lichtintensiteit en ken de waarde toe aan  $X$
- < 2 > Reken  $X$  om naar  $U$
- < 3 > Stuur  $U$  uit
- < 4 > Opnieuw

Om het programma ooit te laten eindigen gebruiken we de voorwaarde: Zolang de lichtintensiteit ( $X$ ) groter is dan 5 wordt er geregeld.

## Programma:

### Pascal

```
Program regeling;
var A,X: integer;
    U: real;
Begin
while X > 5 do
begin
X:= PORT [ 65 ]
U:= A . X + B
A:= round ( U )
PORT [ 65 ]:= A
end;
end.
```

### Exbasic

```
10 Begin
20 X = INP ( 65 )
30 While X > 5
40 X = INP ( 65 )
50 U = A . X + B
60 OUT 65,U
70 WEND
80 End
```

### \* Opdracht 50.

Run dit programma. Varieer m.b.v. gordijnen de lichtintensiteit in de ruimte.

Maak nu zelf: opstelling, algoritme, stroomdiagram, programma voor

#### 1. Brandalarm.

Benodigheden: temperatuursensor  
zoemer (+ eventueel lamp(en))

#### 2. Inbraakalarm.

Benodigheden: fotocel + infraroodzender  
zoemer (+ eventueel lamp(en))

#### 3. Deurgong.

Benodigheden: fotocel + infraroodzender  
zoemer (gong)

De gong gaat alleen als er iemand groter dan 1.40 m binnen komt.  
Van alle anderen wordt een telling bijgehouden.

#### 4. Geautomatiseerde Keukenweger. (tot 500 gram)

Benodigheden: draaihoeksensor, veer + contragewicht, statief,  
weegschaaltje

Op een beeldscherm wordt het gewicht afgebeeld.

#### 5. Display.

#### 6. Temperatuurregeling.

Benodigheden: temperatuursensor  
ventilator (met schakelunit 220V)

#### 7. Aansturen stappenmotor.

#### 8. Draaischijf.

Benodigheden: fotocel + infraroodzender, draaischijf,  
3 voorwerpen met verschillende afmetingen.

Plaats de 3 voorwerpen op de draaischijf. Schijf draait.

Na de 2e rondgang stopt de draaischijf bij het kleinste

voorwerp. Op het scherm wordt afgebeeld: Dit is het kleinste voorwerp.

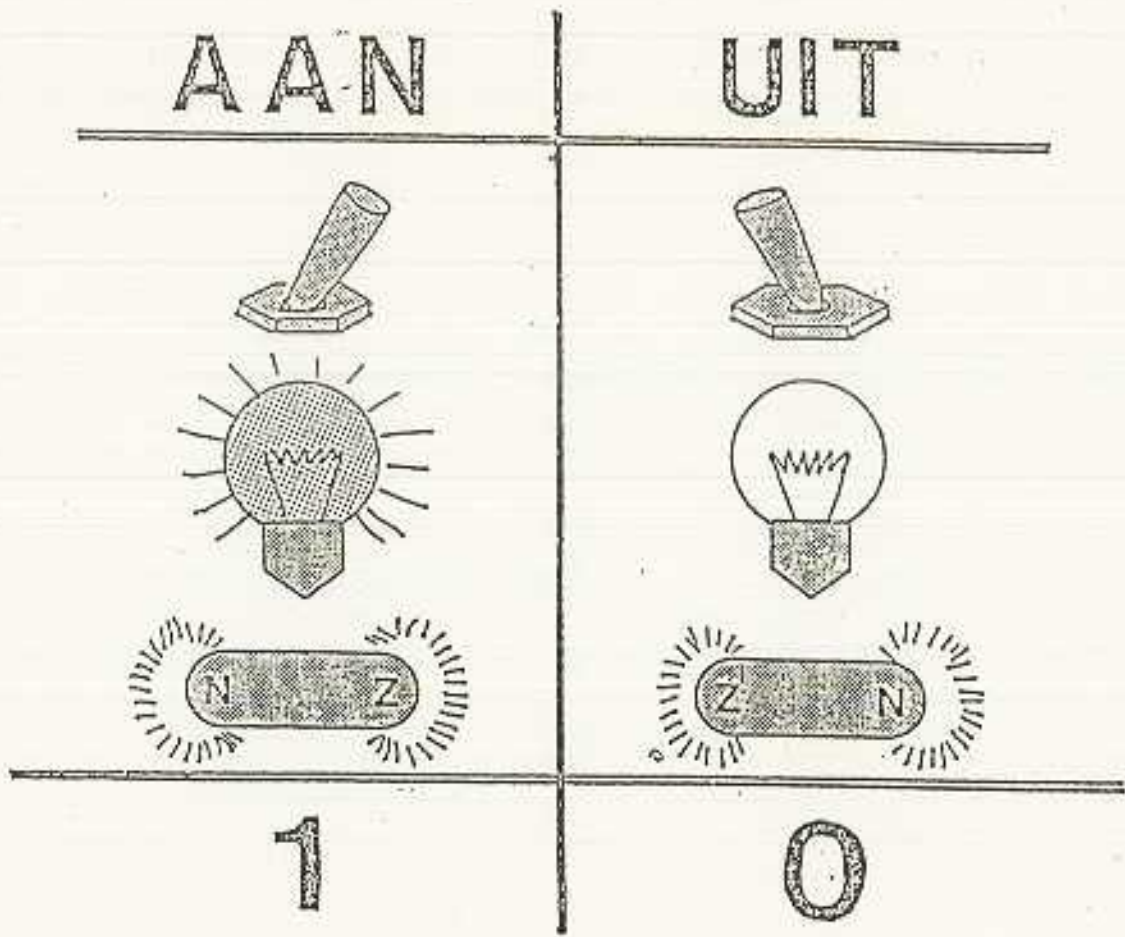
9. Aansturen van auto's, treinen, hijskranen, liften etc. etc.

Extra benodigheden: fotocellen + infraroodzenders  
ompoolrelais, magneet, zoemer,  
schakelunit ( 220 V )

10. Reeds eerder besproken:

- verkeerslichtenregeling
- lichtregeling
- looplicht ( discolicht )
- thermometer
- voltmeter
- brievenweger

BIJLAGE BINAIRE GETALLEN



binair

## Inleiding.

We kennen natuurlijk allemaal het DECIMALE of tien-tallige talstelsel. Dit talstelsel bestaat uit tien symbolen, namelijk: 0, 1, 2, 3, 4, 5, 6, 7, 8 en 9. Deze symbolen krijgen pas een waarde als ze in een bepaalde volgorde worden gezet. Zo hebben de symbolen 2, 3 en 4 een heel andere waarde als ze genoteerd worden als 234 dan als ze genoteerd worden als 432.

De positie van de symbolen bepaald dan ook de macht van 10 waarmee dit symbool vermenigvuldigd moet worden. Het decimale talstelsel is daarom een POSITIESTELSEL.

## Voorbeeld 1.

$$\begin{aligned}234 &= 200 + 30 + 4 \\ &= 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0\end{aligned}$$

## Voorbeeld 2.

$$\begin{aligned}345,03 &= 300 + 40 + 5 + 0,0 + 0,03 \\ &= 3 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 + 0 \cdot 10^{-1} + 3 \cdot 10^{-2}\end{aligned}$$

Het grondtal van het decimale stelsel is dus 10.

Een computer is, zoals je weet, een electronisch apparaat. Dit wil zeggen dat de computer werkt met elektrische spanningen. Nu is de computer zo gebouwd dat hij slecht twee verschillende situatie's kan waarnemen, namelijk: Er is wel spanning en er is geen spanning.

Omdat de computer maar twee situatie's kent, wordt er in de computertechniek met het twee-tallige of BINAIRE talstelsel gerekend.

Dit stelsel bestaat uit twee symbolen, die gebruikt worden om de situatie aan te geven, namelijk 0 en 1.

0 geeft dan aan: er is geen spanning, en

1 geeft dan aan: er is wel spanning.

## Het binaire tallenstelsel.

Het binaire talstelsel is, evenals het decimale talstelsel, een positiestelsel. De plaats van de symbolen bepaalt daarom de waarde van het getal.

Het binaire talstelsel bestaat uit twee symbolen; 0 en 1.

Het grondtal in het binaire talstelsel is 2.

### Voorbeeld 3.

Een voorbeeld van een binair getal is 10110.

Het stelt, omgerekend in het decimale stelsel, het getal 22 voor.

$$\begin{array}{r} 10110 = \\ \begin{array}{l} \rightarrow 0 * 2^0 = 0 * 1 = 0 \\ \rightarrow 1 * 2^1 = 1 * 2 = 2 \\ \rightarrow 1 * 2^2 = 1 * 4 = 4 \\ \rightarrow 0 * 2^3 = 0 * 8 = 0 \\ \rightarrow 1 * 2^4 = 1 * 16 = 16 \end{array} \\ \hline 22 \end{array}$$

Om aan te geven in welk talstelsel een getal gegeven is, wordt het grondtal, tussen haakjes, als index aan het getal toegevoegd.

Een verkorte notatie van de zin: "Het binaire getal 10110 stelt in het decimale stelsel het getal 22 voor.", is dan:

$$10110 (2) = 22 (10)$$

Staat er na een getal geen index, dan nemen we aan dat we in het decimale stelsel werken.

Het omrekenen van het ene talstelsel naar het andere talstelsel noemen we converteren.

In het decimale stelsel heeft elk getal een naam, bijvoorbeeld twee, zes, vierentwintig, honderd.

Het binaire talstelsel kent zulke namen niet. Het binaire getal 10110 wordt daarom uitgesproken als:

een - nul - een - een - nul (binair).

## CONVERTEREN ; BINAIR NAAR DECIMAAL.

De positie van het symbool, 0 of 1, in een binair getal, bepaalt de macht van 2 waarmee we dit getal moeten vermenigvuldigen.  
Het systeem werkt hetzelfde als in het decimale talstelsel.

voorbeeld 4.

$$\begin{aligned} 1101(10) &= 1 \cdot 10^3 + 1 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0 \\ &= 1000 + 100 + 0 + 1 \\ &= 1101 \end{aligned}$$

Voorbeeld 5.

$$\begin{aligned} 1101(2) &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 \\ &= 8 + 4 + 0 + 1 \\ &= 13 \end{aligned}$$

In de computerkunde werk je vaak met 8 positie's. De volgende tabel is daarbij erg handig:

Tabel:

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	64	32	16	8	4	2	1

Deze tabel is een hulpmiddel om snel de conversie van binair naar decimaal uit te voeren. Daarvoor zetten we de binaire waarde uit in de tabel, op de desbetreffende plaatsen. We lezen dan vrijwel direct de decimale waarde af, door de kolommen waar een 1 in staat bij elkaar op te tellen.

Voorbeeld 6.

Converteer het binaire getal 10010110 (2) naar het decimale talstelsel. De oplossing vinden we door gebruik te maken van de tabel.

7	6	5	4	3	2	1	0
2	2	2	2	2	2	2	2
128	64	32	16	8	4	2	1
1	0	0	1	0	1	1	0

Uit deze tabel lezen we nu af:  
 $10010110(2) = 128 + 16 + 4 + 2$   
 $= 150(10)$



Je ziet nu misschien ook dat je een minimale en een maximale waarde kunt krijgen, als we gebruik maken van 8 positie's.

minimaal:  $00000000_2 = 0_{10}$

maximaal:  $11111111_2 = 255_{10}$

\* Opdracht 1.

Reken de gegeven decimale getallen zelf na door de binaire getallen te converteren naar decimale getallen, door gebruik te maken van de tabel.

\* Opgave 1.

Converteer de volgende binaire getallen naar het decimale talstelsel:

a. 10

d. 1011001

b. 101

e. 1011

c. 101010

f. 11001100

## CONVERTEREN : DECIMAAL NAAR BINAIR.

Het converteren van decimaal naar binair kunnen we op twee manieren doen:

Manier 1.:

Voorbeeld 7

Schrijf het getal 87 (10) binair.

Oplossing: We maken weer gebruik van de tabel.

7	6	5	4	3	2	1	0
2	2	2	2	2	2	2	2
-----							
128	64	32	16	8	4	2	1

Met behulp van de tabel zoeken we de hoogste macht van 2 die we van 87 af kunnen trekken, zodat de rest nog positief is. In dit geval is dat  $2^6$ . Dan gaan we opzoek naar de hoogste macht van 2 die we van de rest af kunnen trekken, enzovoorts....

$$\begin{array}{r} 87 \quad 6 \\ 64 = 2 \\ \hline \text{1e rest} \rightarrow 23 \quad 4 \\ 16 = 2 \\ \hline \text{2e rest} \rightarrow 7 \quad 2 \\ 4 = 2 \\ \hline 3 \quad 1 \\ 2 = 2 \\ \hline 1 \quad 0 \\ 1 = 2 \\ \hline 0 \end{array}$$

$$\text{Dus } 87 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$87 (10) = 1010111 (2)$$

Voorbeeld 8.

Schrijf 26 (10) binair

Oplossing:

$$\begin{array}{r} 26 \quad 4 \\ 16 = 2 \\ \hline 10 \quad 3 \\ 8 = 2 \\ \hline 2 \quad 1 \\ 2 = 2 \\ \hline 0 \end{array}$$

1 1 1

$$\text{Dus } 26 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

$$26 (10) = 11010 (2)$$

Opgave 2.

Converteer de volgende, decimale, getallen op dezelfde manier van decimaal naar binair.

a. 22

c. 117

e. 250

b. 43

d. 192

f. 23

Opgave 3.

Converteer de, decimale, getallen 0 tot en met 20 van decimaal naar binair.

Manier 2.:

Voorbeeld 9.

Schrijf het decimale getal 87 binair.

Oplissing:

$87 : 2 = 43 \text{ rest } 1$	want $87 = 43 \cdot 2^1 + 1 \cdot 2^0$
$43 : 2 = 21 \text{ rest } 1$	$43 \cdot 2^1 = 21 \cdot 2^2 + 1 \cdot 2^1$
$21 : 2 = 10 \text{ rest } 1$	$21 \cdot 2^2 = 10 \cdot 2^3 + 1 \cdot 2^2$
$10 : 2 = 5 \text{ rest } 0$	$10 \cdot 2^3 = 5 \cdot 2^4 + 0 \cdot 2^3$
$5 : 2 = 2 \text{ rest } 1$	$5 \cdot 2^4 = 2 \cdot 2^5 + 1 \cdot 2^4$
$2 : 2 = 1 \text{ rest } 0$	$2 \cdot 2^5 = 1 \cdot 2^6 + 0 \cdot 2^5$
$1 : 2 = 0 \text{ rest } 1$	$1 \cdot 2^6 = 0 \cdot 2^7 + 1 \cdot 2^6$

In de kolom "rest" zien we nu het binaire getal verschijnen:

$$87 (10) = 1010111 (2)$$

Voorbeeld 10.

Schrijf het decimale getal 26 binair.

Oplissing:

$$26 : 2 = 13 \text{ rest } 0$$

$$13 : 2 = 6 \text{ rest } 1$$

$$6 : 2 = 3 \text{ rest } 0$$

$$3 : 2 = 1 \text{ rest } 1$$

$$1 : 2 = 0 \text{ rest } 1$$

$$\text{Dus } 26 (10) = 11010 (2)$$

Opgave 4.

Converteer de volgende, decimale, getallen op manier 2 van decimaal naar binair.

a. 22  
b. 47

c. 81  
d. 65

e. 203  
f. 191

Opgave 5.

Converteer de getallen 21 tot en met 40 van decimaal naar binair.

