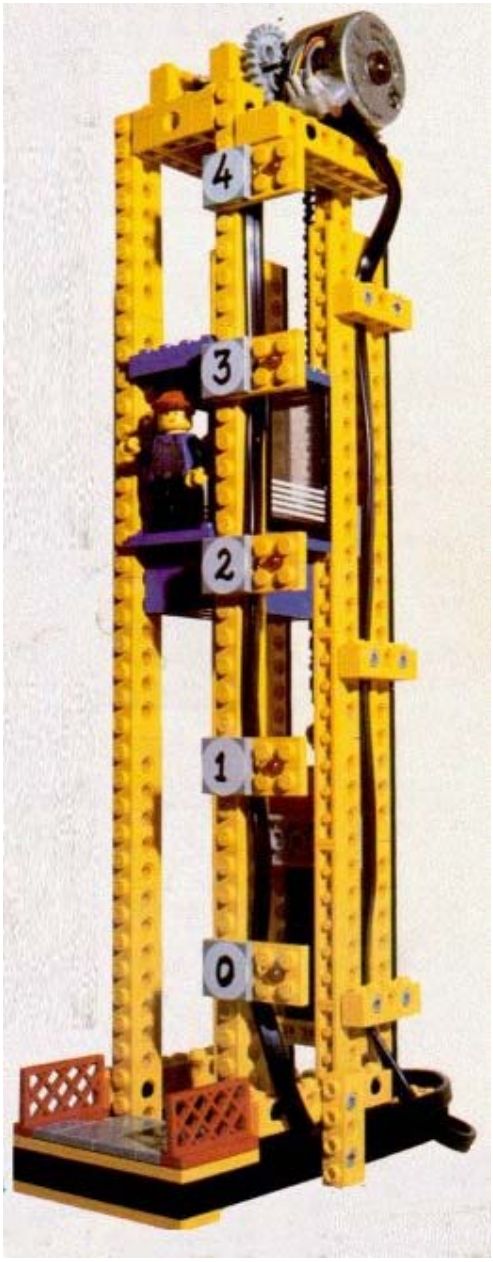
Legolift

Theo van Doorn

MSX Club Magazine 23 maart 1989

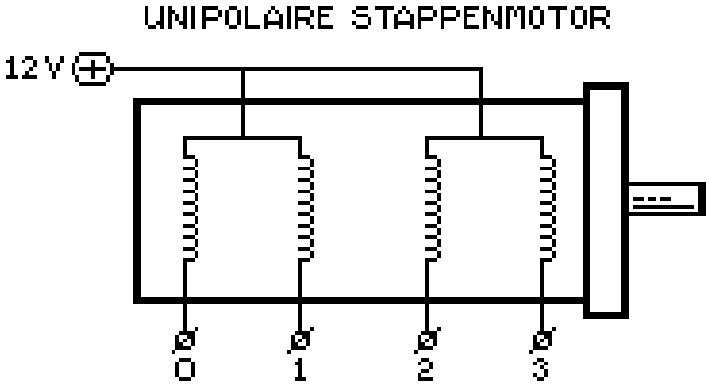


Een van de meest interessante toepassingen van een computer is, deze via een interface te verbinden met de buitenwereld, zodat men met behulp van de computer bijvoorbeeld metingen kan verrichten of modellen kan
besturen.

De MSX-computer, die minstens een, maar vaak twee cartridge-aansluitingen heeft is hiervoor uitermate geschikt. Zonder een interface is het evenwel mogelijk om verbindingen met de buitenwereld te maken, bijvoorbeeld via de joystick- en printerpoorten, deze mogelijkheden worden hier echter niet nader omschreven. De cartridge-poort van een MSX-computer beschikt standaard over de volledige data-, adres- en controlbus, bovendien is hierop ook de voedingsspanning van +5 Volt aanwezig, welke men met maximaal 300 mA mag belasten. Deze gegevens maken het voor de hobbyist erg interessant om voor de cartridge-poort allerlei interfaces te bouwen en aan te sluiten. Een van de meest flexibele interfaces is een ontwerp welk is gepubliceerd in Elektuur van februari 1986. Dit interface beschikt over 4 slots, waarin diverse soorten kaarten kunnen worden gestoken. De 4 slots hebben ieder een eigen adres, zodat deze makkelijk in basic, middels de "OUT"-en "lNP"-instructies aangestuurd kunnen worden. Zo zijn er bijvoorbeeld een 8-kanaals l/0-kaart, een A/D-omzet-ter-kaart en een klok-kaart. Hopenlijk worden hiervoor nog andere kaarten ontworpen.
Na het bouwen van dit interface en de 8-kanaals l/0-kaart, ben ik hiermee gaan experimenteren. De l/0-kaart heeft 8 ingangen welke in niet aangesloten toestand "hoog" (+ 5 Volt) zijn. Door op een ingang een schakelaar aan te sluiten, welke de ingang met de nul (GND) kan verbinden, kan men in een programma de stand, "AAN" of "UIT", van de schakelaar inlezen. Dit kan men bijvoorbeeld toepassen in een alarm-installatie of in een modeltrein-besturing voor het detekteren van wissels en treinen. Maar ook als extern te bedienen handschakelaar of bedieningspaneel. De 8 uitgangen van de kaart worden gevormd door een ULN 2803, welke 8 open-kollektor-darlington-uitgangen heeft, die maximaal een spanning van 50 Volt kan verdragen en een stroom van maximaal 500 mA per kanaal kan voeren. Bovendien heeft dit IC een beveiligingsdiode aan iedere uitgang, waardoor het ook mogelijk is induktieve belastingen zoals relais en kleine motoren te schakelen. Een van de eerste en meest makkelijke toepassingen is, om op de uitgangen 8 lampjes aan te sluiten. Men kan dan in een klein basicprogramma met behulp van de OUT-instruc-tie de lampjes in allerlei patronen aan en uit zetten waardoor een heuse lichtshow ontstaat. Als de lampjes echter te veel stroom opnemen of als de spanning hoger is dan 5 Volt, dan is het toepassen van een externe voedingsbron noodzakelijk. Men moet dan wel pen 10 van de ULN 2803 loskoppelen van de 5 Volt computerspanning en deze verbinden met de plus van de externe spanningsbron. De min moet verbonden worden met de massa (GND) van de l/O-kaart. Een volgende toepassing van de 8 kanaals l/O-kaart was het besturen van modellen gebouwd met LEGO-TECHNIC. Met de 8 beschikbare uitgangen kan men uiteraard acht motortjes schakelen, om deze gelijkstroommotortjes echter zowel rechts- als linksom te kunnen laten draaien, heb ik de schakeling volgens fig. 1 toegepast. Het is dan mogelijk om met twee uitgangen een motor zowel links- als rechtsom te laten draaien, bovendien wordt de motorstroom nu door powertransistors geschakeld, waardoor de ULN 2803 van de l/O-kaart wordt ontlast, hetgeen wel gewenst is, omdat de stroom bij aanlopen en in geval van blokkeren erg hoog kan oplopen. Een soortgelijke schakeling kan men ook vinden in een uitgave van Elektuur: "Robot-besturing met uw homecomputer", waarin overigens ook de MSX-interface en de 8-kanaals l/O-kaart alsmede andere motorbesturingen worden beschreven. Voor de hobbyist, erg aan te bevelen! Met de 8-kanaals l/O-kaart kan men op deze manier dan 4 motoren

besturen. Een van mijn projekten was een 3-assige robotarm van LEGO-TECHNIC, waarin 3 motoren voor de beweging van de arm en 1 motor voor de grijper waren toegepast. Op de ingangen werden de diverse eindschakelaars aangesloten, om te voorkomen dat de motoren te ver zouden doordraaien. De bediening geschiedde met twee joysticks.

Een ander type gelijkstroom-motor is de stappenmotor. De werking van de stappenmotor is heel verschillend ten opzichte van de normale gelijkstroom-motor. Door de twee polen van de normale gelijkstroom-motor aan te sluiten op een gelijkspanning, gaat de motor draaien. Worden de twee polen omgewisseld, dan keert ook de draairichting van de motor om. De stappenmotor echter heeft veelal vier of meer aansluitingen, afhankelijk van het type, hier ga ik in dit artikel niet verder op in. Sluit men deze polen aan op een gelijkspanning, dan zal dit geen draaiing tot gevolg hebben. In principe zal de stappenmotor bij elke spanningspuls op een volgende spoel, een stapje verder gaan. De meest gangbare motoren doen er 48 "stapjes" over om precies een omwenteling te maken. Er zijn ook andere typen. Door deze pulsbesturing is de stappenmotor uitermate geschikt om door een computer te worden bestuurd.



STAP 1	0	1	0	1	5
STAP 2	0	1	1	0	6
STAP 3	1	0	1	0	10
STAP 4	1	0	0	1	9

FIG.2: AANSTURINGS-VOLGORDE

Door de snelheid waarmee de pulsen worden toegediend, wordt de draai-snelheid bepaald. Dit is overigens wel aan een bepaalde grens gebonden omdat bij een te snelle puls-toediening, de motor als het ware niet meer kan "bijhouden". Ook kan het aantal stappen exact worden bestuurd, waardoor een uiterst nauwkeurig te bepalen verplaatsing van het aangedreven object kan worden verkregen. Men "weet" dan als het ware precies waar het aangedreven object zich bevindt. Door deze eigenschappen wordt de stappenmotor dan ook gebruikt in aandrijvingen van robots, printers, plotters, disk-drives, e.d.

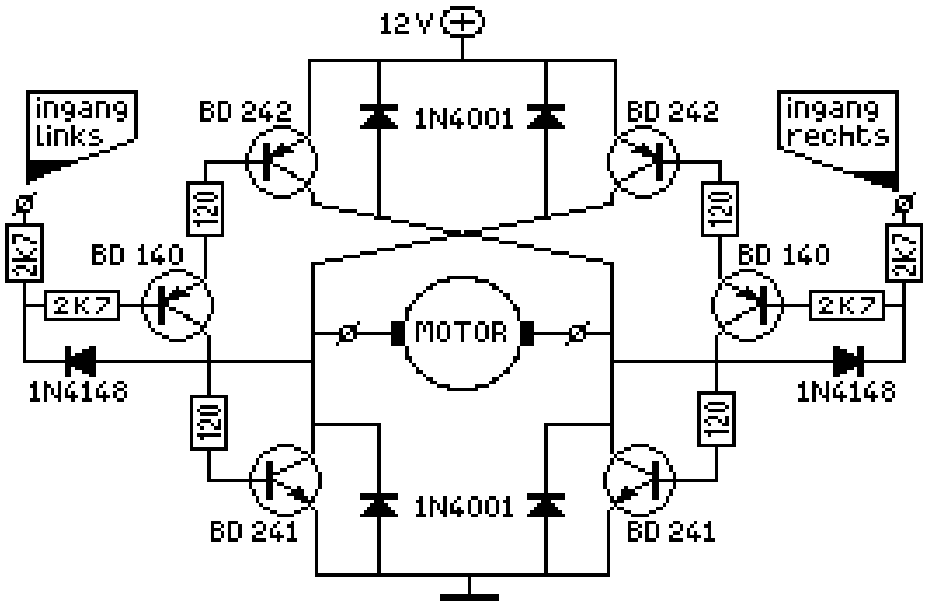


FIG 1: MOTORSTUURSCHAKELING

In de elektronica-dumphandel zijn stappenmotoren verkrijgbaar. Ook kan men ze wel eens uit oude, niet meer gebruikte apparatuur slopen, zoals in mijn geval. Hiermee ben ik wat gaan experimenteren. Deze stappenmotor is van het unipolaire type. De motor heeft vier spoelaansluitingen en een gemeenschappelijk massa-aansluiting. Door nu de vier spoelen in een bepaalde volgorde spanningspulsen te geven, kan men de motor laten draaien. Het toedienen van reeksen spanningspulsen, kan men verzorgen door een elektronische schakeling toe te passen, welke de pulsen genereert. Deze aanstuurlogica kan vooral bij hogere toerentallen, zeer complex worden. Voor de computer-hobbyist is het echter heel interessant om deze aansturing softwarematig, via een interface te doen. Daartoe heb ik de motor aangesloten op 4 uitgangen van de 8-kanaals l/0-kaart, en voorzien van een externe spanningsbron van 12 Volt. Ik heb me beperkt tot de aansturing in de zogenaamde "full-step mode", het is ook mogelijk om de motor halve stappen te laten doen. De volgorde van aansturing zien we in fig. 2 weergegeven. In deze figuur zien we in feite de binaire weergave van de waarden die achtereenvolgens naar de l/0-kaart gestuurd moeten worden. De l/0-kaart zit in het eerste slot van de interface, en is bereikbaar op adres 0. Met het volgende basic-programma kunnen we nu de motor laten draaien.

```
10 OUT 0,&B00000101 : GOSUB 60
20 OUT 0,&B00000110 : GOSUB 60
30 OUT 0,&B00001010 : GOSUB 60
40 OUT 0,&B00001001 : GOSUB 60
50 GOTO 10
60 REM tijds-vertraging
70 FOR T=0 TO 3 : NEXT T : RETURN
```

Voor de duidelijkheid zijn hier de binaire waarden gebruikt, men kan echter ook de decimale equivalenten invullen, achtereenvolgens dus 5, 6, 10 en 9. De draairichting kan men veranderen door de waarden in omgekeerde volgorde naar de l/0-kaart te sturen, dus 9,10,6 en 5. Detijds-vertrag-ing is noodzakelijk omdat anders de pulsen elkaar te snel opvolgen, waardoor de motor niet

in staat is deze te volgen en gewoon stil blijft staan. Door de waarde 3 in de tijdvertragingslus te wijzigen, is het mogelijk de snelheid van de motor te veranderen. Deze vertraging bepaalt namelijk de lengte van de puls. Hiermee kan men wat experimenteren om uit te zoeken wat de meest geschikte snelheid voor een bepaalde toepassing is. Men kan met behulp van een basic-programma de stappenmotor nu volledig besturen. Door bijvoorbeeld de tijdsvertraging variabel te maken kan men de snelheid regelen, en ook kan men een omkeer-routine in het programma opnemen.

Mijn wens was nu de stappenmotor toe te passen in een model, gebouwd met LEGO-TECHNIC en bovendien een grafische weergave daarvan op het scherm te laten zien, waarbij de bewegingen van het model en de schermweer-gave synchroon zouden lopen. Allereerst moest de stappen-motor aangepast worden om deze te gebruiken in combinatie met LEGO-TECHNIC. De asdiameter van het motortje is 2mm. Nu heb ik in het hart van een asje van LEGO-TECHNIC een gaatje geboord van 2mm. Dit moet erg nauwkeurig gebeuren, het beste gaat dit in een draaibank, maar met een vaste hand en een scherp oog is het voor de knutselaar best te doen. Vervolgens heb ik het asje afgezaagd op een lengte van 6mm, en op de motor-as vastgelijmd met 10 secondenlijm. Hiervan niet teveel gebruiken, daar anders de as in het lager vastgelijmd wordt. Vervolgens moet het motortje op een plat LEGO-blokje van 4x2 noppen worden bevestigd. Dit blokje moest door de afmetingen van de motor, aan de bovenkant met een vijl enigszins uitgehold worden. Het vastlijmen gaat uitstekend met een lijmpistool. Het is wel zaak om de motor zodanig te bevestigen, dat de as precies op de juiste plaats zit, zodat deze in een LEGO-TECHNIC-model kan worden toegepast. Dit centreren kan men met behulp van een aantal LEGO-TECHNIC-onderdelen zoals plaatjes en gatenbalkjes verwezenlijken, en moet voor een LEGO-bouwer geen probleem zijn. Nu nog een langer 5-aderig snoertje aan de motor solderen, en de bouw kan beginnen. Als model heb

ik een lift gekozen. Dit is een relatief eenvoudig model, en kan met een motor bestuurd worden. Het bouwen van het model is niet zo moeilijk, het programma om een en ander op het scherm te toveren had echter meer voeten in aarde. Ik koos voor screen 2 en een programma in basic. In eerste instantie heb ik een plaatje getekend op millimeterpapier, waarna men de coordinaten makkelijk in het basic-programma kan opnemen, om met de LINE- en DRAW-instructies de basis-tekening uit te voeren. Om de roterende beweging van de motor zichtbaar te maken, heb ik 4 sprites toegepast, die elkaar op hetzelfde transparant steeds afwisselen, waardoor een roterende beweging ontstaat. De lift bestaat uit twee sprites, een onder- en een bovendeel. Voor de etagenummers heb ik ook sprites toegepast, om ze te kunnen laten knipperen. De liftkabel wordt steeds met de LINE-instructie bijgesteld. Het programma (LIFTSCR2.BAS) zet na de initialisatie de sprites-data met de VPOKE-instructie in de SPRITE$-tabel in het videogeheugen (VRAM). Daarna wordt het scherm opgebouwd en worden de sprites geplaatst. Na het indrukken van het gewenste etagenummer op het toetsenbord, zet de lift zich dan in beweging, waarna men weer opnieuw een nummer kan ingeven. Gedurende de verplaatsing van de lift worden achtereenvolgens de stappenmotor aangestuurd, de sprites verplaatst en de liftkabel bijgesteld. Dit herhaalt zich totdat de lift op de gewenste etage is aangekomen. Omdat het programma LIFTSCR2.BAS geheel in basic is geschreven, draait de stappenmotor enigszins stoterig. Nadat de stappenmotor een aantal stappen heeft gedaan, moet er immers telkens een aantal sprites worden verplaatst, de liftkabel moet worden aangepast en er moet bovendien nog wat rekenwerk verricht worden, alvorens dan de stappenmotor weer aan de beurt is om een aantal stappen te zetten. In basic kosten deze schermhandelingen relatief veel tijd. Een oplossing hiervoor is dan, om een aantal machinetaal-routines in het basicprogramma op te nemen, die deze schermhandelingen voor hun rekening nemen. Dit opent dan ook de mogelijkheid om het aantal toegepaste sprites uit te breiden, bijvoorbeeld door een figuurtje in en uit de lift te laten stappen die zich met de lift verplaatst.

Na een aantal ontwikkelingsfase's ben ik dan uiteindelijk gekomen tot het huidige programma MUISLIFT.BAS. Dit programma is grotendeels geschreven in machinetaal met een klein gedeelte in basic, en maakt gebruik van screen 8 van de MSX-2 hetgeen een fraaier plaatje oplevert. Dit plaatje overigens is getekend met behulp van het tekenprogramma Designer Plus. Daartoe werd eerst in basic in screen 8 de basis tekening gemaakt, en vervolgens gesaved met BSAVE "LIFT.PIC",&HOOOO,&HD3FF,S . Hiermee saved men dat gedeelte van de data in het VRAM welke de hele scherm-inhoud van page 1 bevat. Deze tekening kan men dan, omdat de extensie ".PIC" toegevoegd is, inladen in Designer Plus voor verdere bewerking. Als met dit programma de tekening voltooid is en op de diskette is gesaved, kan men deze tekening in een basicprogramma weer inladen in screen 8 met de opdracht BLOAD "NAAM.PIC",S waarbij de toevoeging ",S" aangeeft dat de data voor het videogeheugen (VRAM) bestemd is. Op dezelfde manier kan men ook de sprites in het VRAM inladen. Dit gaat vele malen sneller dan met VPOKE van de sprites-data in een basicprogramma. De sprites-data moeten dan eerst in een basicprogramma of met behulp van een sprites-editor in de sprite$-tabel in het VRAM worden geplaatst, waarna men deze data op diskette kan saven met de opdracht BSAVE "SPRITES.BIN",&HFOOO,&HF800,S. De twee hexadecimale getallen geven het begin- en het ein-dadres van de te saven data ,de sprite$-tabel, aan. In dit geval is alleen het gedeelte van de sprites-tabel met de benodigde sprites gesaved (FOOOH-F25FH). De iiftbesturing kan zowel via het toetsenbord als met een muis geschieden. Bovendien kan de snelheid van de lift, op het scherm ingesteld worden, waarbij de stappenmotor echter deze snelheid, indien in net rode gebied is ingesteld, niet meer kan bijhouden, vanwege de te korte pulsen.

Deze instelling van de snelheid is in het basicgedeelte terug te vinden, hier wordt de ingestelde waarde met een POKE op een bepaalde geheugenplaats gezet, ten behoeve van de machinetaal-routine.

Het programma bestaat dus uit een basic- en een machinetaalgedeelte. Deze combinatie maakt snelle schermbewerkingen mogelijk, terwijl in basic bepaalde zaken, waarbij de snelheid niet van belang is, eenvoudiger geprogrammeerd kunnen worden. Het machinetaalgedeelte staat vanaf adres DOOOH in het geheugen en heeft een lengte van 521 bytes. De source-listing staat op de diskette onder de naam LIFT.FLS en is in basicregels geschreven. Als zodanig is deze tekst met behulp van de FLASH-assembler, in machinetaal vertaald, en op diskette gesaved met BSAVE "LIFT.BIN",&HDOOO,&HD20F . In de machinetaal wordt gebruik gemaakt van diverse BIOS-routines, deze vindt men vooraan in de sourse-listing met erachter het adres waar deze aan te roepen zijn en een korte uitleg ervan. De meeste, in dit programma gebruikte, BIOS-routines bevinden zich in de SUBROM van de MSX-2, welke in een ander slot zit. Deze kan men in tegenstelling tot de MSX-1 BlOS-routines niet rechtstreeks aanroepen. Dit kan dan door eerst het adres van de BlOS-routine in register IX van de Z80-microprocessor te laden, waarna de BlOS-routine EXTROM op adres 015FH wordt aangeroepen. Deze zorgt dan op zijn beurt voor de interslot- call naar de SUBROM. Er zijn ook andere, snellere methodes met een van de routines CALSLT of SUBROM, de hier toegepaste is de meest eenvoudige, omdat deze routine zelf uitzoekt in welk slot de subrom zich bevindt, en men alleen het adres van de gewenste BlOS-routine dient in te geven. Een meer uitgebreide omschrijving van de BlOS-routines vindt men in gespecialiseerde literatuur op dit gebied. Het machinetaalgedeelte is modulair opgebouwd en bestaat uit twee hoofdlussen, lift omhoog en lift omlaag, en een routine voor de muisbesturing. In het basicprogramma worden deze aangeroepen met respectievelijk USRO, USR1 en USR2. Bij de twee eerstgenoemden wordt bovendien een parameter meegegeven HO% en LA%. Dit is een integer tellerwaarde voor het aantal keren dat de aangeroepen hoofdlus moet worden doorlopen, en wordt bepaald door de huidige positie van de lift en het gewenste etagenummer, de liftverplaatsing dus. In de hoofdlussen wordt telkens een andere subroutine aangeroepen, die ieder Ã©Ã©n onderdeel voor hun rekening nemen. Zo zijn er subroutines voor het besturen van de stappenmotor, het verplaatsen van de diverse sprites (lift, figuur en rotor), het aanpassen van de liftkabel en het knipperen van het gewenste etagenummer.

Deze modulaire opbouw van het programma maakt het overzichtelijk en makkelijker te wijzigen of uit te breiden met andere subroutines.

Het verplaatsen van de sprites inde subroutine gebeurt door het rechtstreeks wijzigen van de data in de transparanttabel. In deze tabel, die in screen 8 begint op adres FAOOH, zijn voor ieder van de 32 transparanten, 4 bytes gereserveerd. De eerste 3 bytes bevatten achtereenvolgens de waarden voor het Y-coordinaat, het X-coordinaat en het nummer van de sprite. Het vierde byte bevat in de screenmode O t/m 3 (MSX-1), het kleurnummer. In screenmode 4 t/m 8 heeft dit byte geen funktie, voor het bepalen van de kleur van de sprites dient dan de spritecolor-tabel. Zo wordt het knipperen van een etagenummer teweeggebracht door in de spritecolor-tabel, de data van de betreffende sprite te wijzigen. In deze knipper-subroutine wordt overigens ook het aansturen van de etage-led's in het model, via een tweede l/0-kaart, verzorgd. Omdat er vijf led's aangesloten moesten worden, kon dit aantal niet meer op de eerste l/0-kaart, zodat uitbreiding met een tweede noodzakelijk was. In de

subroutine voor het aanpassen van de kabellengte wordt via de BlOS-routine RDVRM de positie van de lift (Y-coordinaat) bepaald, waarna met MAPXYC het adres in de VRAM van de aan te passen pixel wordt vastgesteld. Met behulp van de BlOS-routines SETATR en SETC wordt vervolgens de kleur van de pixel gewijzigd. De subroutine voor de muisbesturing tenslotte maakt o.a gebruik van de BlOS-routine NEWPAD om de status van de muis en de X- en Y-offset uit te lezen. Waarna de offset-waarde bij de huidige waarde wordt opgeteld, en wordt bijgesteld op de betreffende adressen, in het VRAM, van de, X- of Y-coordinaten van de twee elkaar overlappende sprites van het aanwijshandje. Dan wordt met GTTRIG de status van de vuurknop opgevraagd. Is deze ingedrukt, dan worden de X- en Y-coordinaten op een bepaald adres gezet en wordt teruggekeerd naar basic. Met een peek-instructie op deze adressen worden in basic de X- en Y-coordinaten opgevraagd, om vervolgens te bepalen naar welke etage de lift zich moet begeven, of op welke snelheid de lift moet worden ingesteld. Het regelen van de snelheid wordt verwezenlijkt, door het wijzigen van de tellerwaarde in de subroutine DELAY (vertragingslus).

Met dit artikel hoop ik voor sommigen een aanzet te hebben gegeven naar een erg interessante toepassing van de MSX-computer, het besturen van modellen en andere randapparatuur. Ook hoop ik dat de beginnende machinetaalprogrammeur meer inzicht heeft gekregen in het toepassen van bepaalde routines. Wellicht kan men sommige routines in een eigen programma gebruiken. Het programma leent zich voor verdere wijzigingen of uitbreidingen. Het steeds weer zoeken naar nieuwe mogelijkheden in het programmeren en besturen van zelfbouw-modellen blijft een boeiende bezigheid. En de MSX-computer is hiervoor bij uitstek geschikt. Tenslotte wens ik iedere MSX-er veel plezier met zijn of haar computerhobby. Reakties, vragen en suggesties betreffende dit artikel zie ik gaarne tegemoet.

Met vriendelijke groeten,

Theo van Dooren

Weverstraat 26

5524 BE Steensel (Nederland) telefoon 04970-15181

LIFT.FLS

(Scanned and ocr'ed but not guaranteed to be error free! MSXHans)

```
10    ; LIFT.FLS (Lego-lift op screen 8)
15    ; Source-list ing in basic voor assembleren met FLASH.
20    ; Machinetaal-routine voor het verplaatsen van sprites e.d.
30    ; Met muisbesturing van hand-aanwijzer.
35    ; Snelheid is op het scherm in te stellen.
41    ; ==  Een programma van Theo van Dooren   ==
42    ; ==                  Weverstraat 26
43    ; ==                  5524 BE Steensel
50    ORG &HDOOO ;beginadres
60    EXTROM EQU &H015F ;aanroepen van MSX2-routines.   IX=adres.
70    WRTVRM EQU &H0109 ;schrijven naar VRAM.   A=data, HL=adres.
80    RDVRM EQU &H010D  ;lezen van VRAM.       A=data, HL=adres.
90    BIGFIL EQU &H016B ;vullen VRAM. A=data, HL=adres, BC=lengte.
100   MAPXYC EQU &H0091 ;omzetten coord. naar adres.
110                     ;invoer: BC=x, DE=y.  uitvoer: HL=x, A=y.
120   SETATR EQU &H0099 ;veranderen van kleur, A=kleurno.
130   ATRBYT EQU &HF3F2 ;adres voor werkkleur
140   SETC   EQU &H009D ;veranderen van pixelkleur.     **********
150   READC  EQU &H0095 ;bepalen van pixelkleur.
160   NEWPAD EQU &H01AD ;inlezen van muis-status.
170   GTTRIG EQU &HOOD8 ;status vuurknop. inv: A=no. uitv: A=data.
180   XC09   EQU &HFA25 ;x-coord. sprite op transp. 9 (handwijzer).
190   YC09   EQU &HFA24 ;y-coord. sprite op transp. 9 (handwijzer).
200   XC010  EQU &HFA29 ;x-coord. sprite op transp.10 (handwijzer).
210   YC010  EQU &HFA28 ;y-coord. sprite op transp.10 (handwijzer).
220   TELLER EQU &HF7F8 ;teller, aantal hoofdlussen.
230   ETANMR EQU 54000  ;adres voor gewenst etagenummer.
240   LEDBIT EQU 54001  ;adres v bit tbv leds etage-aanduiding.
250   YC09BA EQU 54002  ;store y-coord. handwijzer tbv basicprogr.
260   XC09BA EQU 54003  ;store x-coord. handwijzer tbv basicprogr.
270   SNELH  EQU 54004  ;data voor delay, bepaalt snelheid v. lift.
280   ;
290   ;***********  hoofdlus, lift omhoog  **********************
300   ;
310   LIFTHO LD A,(TELLER)     ;aantal keer de hoofdlus.
320          LD B,A            ;teller in B
330   WEERHO LD C,3            ;per 3 pixels knipperen etageno.
340   HOGER  CALL SPRIHO       ;zet sprites v lift l pixel hoger.
350          CALL ROTOR        ;verplaatst rotor v liftmotor.
360          CALL KABLHO       ;kort liftkabel l pixel in.
370          CALL MOTORH       ;laat motor xx stappen draaien.
380          DEC C             ;teller-l
390          JR NZ,HOGER       ;teller<>0: l pixel verder.
400          CALL KNIPPR       ;laat etageno. knipperen.
410          DJNZ WEERHO       ;tellerB<>0: weer.
420          RET               ;terug naar basicprogr.
430
440   ***********  hoofdlus, lift omlaag  **********************
450
460   LIFTLA LD A,(TELLER)     ;aantal keer de hoofdlus.
470          LD B,A            ;teller in B
```

```
480   WEERLA LD C,3           ;per 3 pixels knipperen etageno.
490   LAGER  CALL SPRILA      ;zet sprites v lift l pixel lager.
500          CALL ROTOR       ;verplaatst rotor v liftmotor.
510          CALL KABLLA      jverlengt liftkabel met l pixel.
520          CALL MOTORL      ;laat motor xx stappen draaien.
530          DEC C            ;teller-l
540          JR NZ,LAGER      ;teller<>0: l pixel verder.
550          CALL KNIPPR      ;laat etageno. knipperen.
560          DJNZ WEERLA      ;tellerB<>0: weer.
570          RET              ;terug naar basicprogr.
580     ;
590     ;********** besturen van stappenmotor (omhoog)  ************
600   ;
610   MOTORH PUSH BC          ; store registers
620          LD B,4           ;teller, 4x4 stappen/subr.
630   STAPHO LD A, 9          ;data stap l
640          OUT (0),A        ;aansturing stap l
650          CALL DELAY       ;vertragingslus
660          LD A,10          ;
670          OUT (0),A        ;Idem stap 2
680          CALL DELAY       ;
690          LD A, 6          ;
700          OUT (0),A        ;Idem stap 3
710          CALL DELAY       ;
720          LD A,5           ;
730          OUT (0),A        ;Idem stap 4
740          CALL DELAY       ;
750          DJNZ STAPHO      ;teller-l, als toO.dan nog 4 stappen
760          POP BC           ;
770          RET  ;
780   ;
790   ;********** besturen van stappenmotor (omlaag)  ************
800   ;
810          OTORL PUSH BC    ; store registers
820          LD B,4           ;teller, 4x4 stappen/subr.
830   STAPLA LD A,5           ;data stap l
840          OUT (0),A        ;aansturing stap l
850          CALL DELAY       ;vertragingslus
860          LD A,6           ;
870          OUT (0),A        ;Idem stap 2
880          CALL DELAY       ;
890          LD A,10          ;
900          OUT (0),A        ;Idem stap 3
910          CALL DELAY       ;
920          LD A, 9          ;
930          OUT (0),A        ;Idem stap 4
940          CALL DELAY
950          DJNZ STAPLA      ;teller-l, als toO.dan nog 4 stappen
960          POP BC           ;
970          RET              ;
980   ;
990   ;*************** delay  (vertragingslus)  ******************
1000 ;
1010  DELAY  LD DE,(SNELH)    ; teller, bepaalt vertragingstijd.
1020  TREKAF DEC E
1030         JR NZ,TREKAF
1040         DEC D
```

```
1050         JR NZ,TREKAF
1060         RET
1070  ;
1080  ;*******************  spriho ****************************
1090  ;  (sprites van lift en manneke , l pixel hoger)
1100  ;
1110  SPRIHO LD IX,RDVRM       ;Lees in VRAM
1120         LD HL,&HFA04      ;adres y-coord. lift boven-deel,
1130         CALL EXTROM       ;(sprite op transp.l)
1140         DEC A             ;en verminder deze met 1.
1150         LD IX,WRTVRM      ;Schrijf nieuwe waarde terug in
1160         CALL EXTROM       ;VRAM, zodat sprite, l pixel omhoog
1170         LD IX,RDVRM       ;schuift.
1180         LD HL,&HFA08      ;
1190         CALL EXTROM       ;
1200         DEC A             ;Idem voor lift onder-deel.
1210         LD IX,WRTVRM      ;(sprite op transp.2)
1220         CALL EXTROM       ; Idem voor stilstaand manneke
1230         LD IX,RDVRM       ;sprite op transp.8)
1240         LD HL,&HFA20
1250         CALL EXTROM
1260         DEC A
1270         LD IX,WRTVRM
1280         CALL EXTROM
1290         RET
1300   ;
1310   ;*************** rotor, roteren van liftmotor  ************
1320   ;
1330   ROTOR LD IX,WRTVRM      ;Verander spriteno. op transp.O
1340         LD HL,&HFA02      ;mbv teller in C.
1350         LD A,C            ;Achtereenvolgens worden de sprites
1360         SLA A             ;no. 0,1,2,3 op transp.O geplaatst,
1370         SLA A             ;waardoor draai-effect ontstaat.
1380         CALL EXTROM       ;
1390         RET               ;
1400   ;*******************  sprila  ****************************
1410   ;
1420   ; (sprites van lift en manneke, l pixel lager)
1430   ;
1440  SPRILA LD IX,RDVRM       ;Lees in VRAM
1450         LD HL,&HFA04      ;adres y-coord. lift boven-deel,
1460         CALL EXTROM       ;(sprite op transp.l)
1470         INC A             ;en vermeerder deze met 1.
1480         LD IX,WRTVRM      ;Schrijf nieuwe waarde terug in
1490         CALL EXTROM       ;VRAM, zodat sprite, l pixel  omhaag
1500         LD IX,RDVRM       ;schuift.
1510         LD HL,&HFA08      ;
1520         CALL EXTROM       ;
1530         INC A             ;Idem voor lift onder-deel.
1540         LD IX,WRTVRM      ;(sprite op transp.2)
1550         CALL EXTROM       ;
1560         LD IX,RDVRM       ;
1570         LD HL,&HFA20      ;Idem voor stilstaand manneke.
1580         CALL EXTROM       ;(sprite op transp.8)
1590         INC A             ;
1600         LD IX,WRTVRM      ;
1610         CALL EXTROM       ;
```

```
1620   ;
1630   ;*********  kablho, kort liftkabel l pixel in  *************
1640   ;
1650  KABLHO PUSH BC           ;Bepaal achtergrondkleur van (78,80)
1660         LD BC,78          ;x-coord.
1670         LD DE, 80         ;y-coord.
1680         LD IX,MAPXYC      ;
1690         CALL EXTROM       ;
1700         LD IX,READC       ;
1710         CALL EXTROM       ;
1720         LD (ACHKLR),A     ;store achtergr.kir.
1730         LD HL,&HFA04      ;Bepaal y-coord. lift boven-deel
1740         LD IX,RDVRM       ;
1750         CALL EXTROM       ;
1760         LD E,A            ;Y-coord. onderste pixel v kabel is:
1770         INC E             ;Y-coord. lift +1.
1780         LD D,O            ;Y-coord. in DE
1790         LD BC,80          ;X-coord. in BC
1800         LD IX,MAPXYC      ;
1810         CALL EXTROM       ;
1820         LD A,(ACHKLR)     ;Verander onderste pixel van kabel in
1830         LD (ATRBYT),A     ;achtergrondkleur. (wissen)
1840         LD IX,SETATR
1850         CALL EXTROM
1860         LD IX,SETC
1870         CALL EXTROM
1880         POP BC
1890         RET
1900
1910   *********  kablla, verlengt liftkabel met l pixel  *********
1920
1930  KABLLA PUSH BC
1940         LD HL,&HFA04     ; Bepaal y-coord. lift boven-deel
1950         LD IX,RDVRM
1960         CALL EXTROM
1970         LD E,A           ;Y-coord. onderste pixel v kabel is:
1980         LD D,O           ;Y-coord. in DE
1990         LD BC,80         ;X-coord. in BC
2000         LD IX,MAPXYC
2010         CALL EXTROM
2020         LD A,l           ; Kleur l pixel onder liftkabel zwart,
2030         LD  (ATRBYT),A
2040         LD IX,SETATR
2050         CALL EXTROM
2060         LD IX,SETC
2070         CALL EXTROM
2080         POP BC
2090         RET
2100  ;
2110  ;***************** knipperen etageno. ***************
2120  ;
2130  KNIPPR PUSH BC           ;
2140         LD HL,&HF830      ;Beginadr trp.3 spritecolortab.VRAM
2150         LD A,(ETANMR)     ;
2160         ADD A,L           ;Beginadr.+ etageno.= transp.no. van
2170         LD L,A            ;gewenste etage.
2180         LD IX,RDVRM       ;
```

```
2190         CALL EXTROM      ;Lees momentele kleurno.
2200         AND &HOF         ;maskeer 4 rechter kleurenbits
2210         CP 15            ;kleur = wit (15)?,
2220         JR  Z,KLR6*J     ;dan andere kleur.
2230         LD A,O           ;
2240  LEDUIT OUT (4),A        ;
2250         LD A,15          ;
2260         JR KLEUR
2270  KLR6   LDA,(LEDBIT)
2280  LEDAAN OUT (4),A
2290         LD A,10          ;
2300  KLEUR  LD BC,0016       ;Vul spritecolortab. met nw kleurno.
2310         CALL BIGFIL      ;
2320         POP BC           ;
2330         RET              ;
2340   ;
2350   ; ***************** muisbesturing ***************
2360   ;                  (tbv handwijzer
2370   ;
2380  MUIS   LD IX,NEWPAD     ;
2390         LD A,&HOC        ;
2400         CALL EXTROM      ;Muis op poort l, aanwezig?
2410         LD A,&HOD        ;
2420         CALL EXTROM      ;Geef X-offset van muis.
2430         LD B,A           ;Store in B.
2440         LD IX,RDVRM      ;
2450         LD HL,XC09       ;Lees X-coord. in van sprite op
2460         CALL EXTROM      ;transparant 9, en tel de X-offset
2470         ADD A,B          ;van de muis hierbij op.
2480         LD (XCOORD),A    ;Store nieuwe X-coord.
2490         LD IX,WRTVRM     ;en schrijf deze weer terug in trans-
2500         CALL EXTROM      ;parant-tabel no.9.
2510         LD HL,XC010      ;en in transp.no.10.
2520         CALL EXTROM      ;
2530  ;
2540         LD IX,NEWPAD     ;
2550         LD A,&HOE        ;
2560         CALL EXTROM      ;Geef Y-offset van muis.
2570         LD B,A           ;Store in B.
2580         LD  IX,RDVRM     ;
2590         LD HL,YC09       ;Lees Y-coord. in van sprite op
2600         CALL EXTROM      ;transparant 9, en tel de Y-offset
2610         ADD A,B          ;van de muis hierbij op.
2620         LD (YCOORD),A    ;Store nieuwe Y-coord.
2630         LD IX,WRTVRM     ;en schrijf deze weer terug in trans-
2640         CALL EXTROM      ;parant-tabel no.9,
2650         LD HL,YC010      ;en in transp.no.10.
2660         CALL  EXTROM     ;
2670   ;
2680         LD A, l          ;
2690         CALL GTTRIG      ;Vuurknop l (=linkermuisknop) in?
2700         CP  O            ;
2710         JR Z,TERUG       ;Nee, dan return.
2720         LD A,(YCOORD)    ;Ja, dan zet Y-coord. op peek-adres
2730         LD (YC09BA),A    ;voor basic-programma.
2740         LD A,(XCOORD)    ;Idem voor X-coord.
2750         LD  (XC09BA),A   ;
```

```
2760   TERUG RET              ;
2770   ;
2780   ;******************  opslag variabelen  *********************
2790   ;
2800   ACHKLR DB 0            ;achtergrondkleur liftschacht
2810   XCOORD DW 0            ;tijd. opslag voor X-coord.
2820   YCOORD DW 0            ;tijd. opslag voor Y-coord.
2830   ;
2840   END
```

LIFTSCR2.BAS

(taken from the basic file)

```
10 ' LIFTSCR2.BAS
20 ' legolift met unipolaire stappenmotor
30 ' aangesloten via 8-kan. I/O-kaart  (adres 0)
40 ' besturing via toetsenbord
50 '=============================================================================
60 ' een programma van Theo van Dooren
70 '                   Weverstraat 26
80 '                   5524 BE  Steensel (Nederland)
90 '                   tel. 04970-15181
100 '============================================================================
110 '  initialisatie  (F1=noodstop-toets)  ----------------------------------
120 COLOR 1,7,7 : SCREEN 2,2 : KEY1,"out 0,0"+CHR$(13) : EN=0 :'EN=etage nu
130 OPEN "GRP:" AS1 : YNU=146 : M=0 : C=9
140 '  inlezen sprites  ------------------------------------------------------
150 FOR I=0 TO 11*32-1 :' aantal sprites x 32
160 READ A: VPOKE BASE(14)+I,A: NEXT I
170 '  schermopbouw   -------------------------------------------------------
180 LINE (0,175)-(256,192),14,BF
190 DRAW"bm70,175c1l20u145l5u5r20u15l5u5r40d5l5d15r105d5l5d145l5"
200 DRAW"bm50,175c1r20u147r8u3l8u15r20d15l8d3r8d2r100d25l100d5r100d25l100d5r100
d25l100d5r100d25l100d5r100d25"
210 PAINT(61,6),1,1
220 FOR Y=40 TO 160 STEP 15 :                                    'raampjes
230 CIRCLE (60,Y),4,7 :CIRCLE (60,Y),2,7: NEXT Y
240 CIRCLE (80,17),5,1 : DRAW"bm74,25e3r1g3" : DRAW"bm86,25h3l1f3" :'liftmotor
250 CIRCLE(20,160),10,3 : PAINT(20,160),3   :                    'boompjes
260 CIRCLE(30,150),23,12,,,7: PAINT(30,150),12
270 CIRCLE(10,165),12,12: PAINT(10,165),12
280 CIRCLE(40,155),13,10,,,3 : PAINT(40,155),10
290 CIRCLE(230,145),30,12,,,7: PAINT(230,145),12
300 CIRCLE(250,155),20,3,,,.3: PAINT(250,155),3
310 CIRCLE(218,165),15,10,,,.7: PAINT(218,165),10
320 '  sprites plaatsen   ----------------------------------------------------
330 S=10
340 FOR Y=35 TO 155 STEP 30
350 PUT SPRITE S,(170,Y),6,S : S=S-1 : NEXT Y :'              etage nummers ***
360 PUT SPRITE 0,(73,9),13,0 :'                                    motor ***
370 PUT SPRITE 1,(73,YNU),4,4 : PUT SPRITE 2,(73,YNU+15),4,5 :'      lift ***
380 LINE(80,23)-(80,YNU),1 :'                                   liftkabel ***
390 PRESET (120,15): PRINT #1,"LEGOLIFT"
400 ' ingave gewenste etage  ------------------------------------------------
410 COLOR 1,14: PRESET(20,182) : PRINT #1," Naar welke etage wil je ? "
420 EW$=INPUT$(1) : EW=VAL(EW$): IF EW>4 OR EW=EN THEN 410
430 IF EW>EN THEN EH=EW-EN: EN=EN+EH : M=0 :GOSUB 450: 'omhoog
440 IF EW<EN THEN EL=EN-EW: EN=EN-EL : M=3 :GOSUB 570: 'omlaag
450 'subr. omhoog  ----------------------------------------------------------
460 COLOR 14: PRESET(20,182) : PRINT #1," Naar welke etage wil je ? "
470 FOR YL=1 TO 30*EH :' lift aantal pixels omhoog
480 YNU=YNU-1
490 FOR O=1 TO 4 :' motor doet 4x4 stappen
500 OUT 0,9 : FOR T=0 TO 0:NEXT T : OUT 0,5: FOR T=0 TO 0:NEXT T : OUT 0,6 : FOR
```

```
T=0 TO 0:NEXT T : OUT 0,10 : FOR T=0 TO 0:NEXT T,O
510 PUT SPRITE 1,(73,YNU),4,4 : PUT SPRITE 2,(73,YNU+15),4,5 :' veplaatsen lift
520 IF YNU MOD 3=0 THEN PUT SPRITE EW+6,(170,155-EW*30),C,EW+6 :IF C=9 THEN C=6
ELSE C=9 :'knipperen etage nummers
530 PUT SPRITE 0,(73,9),13,M : M=M+1 :IF M>3 THEN M=0 :' draaien v. motor
540 LINE(80,164)-(80,YNU+1),7 :' aanpassen liftkabel
550 NEXT YL
560 FOR T=0 TO 1000 :NEXT T : RETURN 410
570 'subr. omlaag  ------------------------------------------------------------
580 COLOR 14: PRESET(20,182) : PRINT #1," Naar welke etage wil je ? "
590 FOR YL=1 TO 30*EL :' lift aantal pixels omlaag
600 YNU=YNU+1
610 FOR O=1 TO 4 :' motor doet 4x4 stappen
620 OUT 0,9 : FOR T=0 TO 0:NEXT T : OUT 0,10: FOR T=0 TO 0:NEXT T : OUT 0,6 :
FOR T=0 TO 0:NEXT T : OUT 0,5 : FOR T=0 TO 0:NEXT T,O
630 PUT SPRITE 1,(73,YNU),4,4 : PUT SPRITE 2,(73,YNU+15),4,5 :'verplaatsen lift
640 IF YNU MOD 3=0 THEN PUT SPRITE EW+6,(170,155-EW*30),C,EW+6 :IF C=9 THEN C=6
ELSE C=9 :' knipperen etage nummers
650 PUT SPRITE 0,(73,9),13,M : M=M-1 :IF M<0 THEN M=3 :' draaien v. motor
660 LINE(80,23)-(80,YNU+1),1 :' aanpassen liftkabel
670 NEXT YL
680 FOR T=0 TO 1000 :NEXT T : RETURN 410
690 ' ************************  data sprites  ************************
700 '  rotor 0  ------------------------------------------------------
710 DATA 000,000,000,001,001,001,001,001,003,007,003,001,000,000,000,000
720 DATA 000,000,000,000,128,192,128,000,000,000,000,000,000,000,000,000
730 '  rotor 1  ------------------------------------------------------
740 DATA 000,000,000,001,001,001,001,001,003,007,003,001,000,000,000,000
750 DATA 000,000,000,000,128,192,128,000,000,000,000,000,000,000,000,000
760 '  rotor 2  ------------------------------------------------------
770 DATA 000,000,000,000,000,000,014,015,012,000,000,000,000,000,000,000
780 DATA 000,000,000,000,000,000,096,224,224,000,000,000,000,000,000,000
790 '  rotor 3  ------------------------------------------------------
800 DATA 000,000,000,001,003,007,003,001,001,001,001,001,000,000,000,000
810 DATA 000,000,000,000,000,000,000,000,128,192,128,000,000,000,000,000
820 '  lift boven ----------------------------------------------------
830 DATA 001,002,002,255,255,255,192,192,192,192,192,192,192,192,192,192
840 DATA 128,064,064,255,255,255,003,003,003,000,000,000,000,000,000,000
850 '  lift onder  ---------------------------------------------------
860 DATA 192,192,192,192,192,192,192,192,192,192,192,192,255,255,255,000
870 DATA 000,000,000,000,000,000,000,000,000,000,000,000,255,255,255,000
880 '  etage no. 0  --------------------------------------------------
890 DATA 003,007,014,012,012,012,012,012,012,012,012,012,012,014,007,003
900 DATA 192,224,112,048,048,048,048,048,048,048,048,048,048,112,224,192
910 '  etage no. 1  --------------------------------------------------
920 DATA 001,003,007,001,001,001,001,001,001,001,001,001,001,001,007,007
930 DATA 128,128,128,128,128,128,128,128,128,128,128,128,128,128,224,224
940 '  etage no. 2  --------------------------------------------------
950 DATA 003,007,014,012,012,000,000,000,000,001,003,007,014,012,015,015
960 DATA 192,224,112,048,048,048,048,112,224,192,128,000,000,048,240,240
970 '  etage no. 3  --------------------------------------------------
980 DATA 007,007,006,000,000,001,003,003,000,000,000,012,012,014,007,003
990 DATA 240,240,048,112,224,192,128,224,112,048,048,048,048,112,224,192
1000 '  etage no. 4  -------------------------------------------------
1010 DATA 000,000,000,000,001,001,003,003,006,006,006,007,007,000,000,000
1020 DATA 096,096,192,192,128,128,000,000,000,096,096,240,240,096,096,096
```

MUISLIFT.BAS

(taken from the basic file)

```
10 ' ------  MUISLIFT :   LEGOLIFT IN SCREEN 8  ---------------------------
20 '
30 ' legolift met unipolaire stappenmotor
40 ' motor aangesloten via 8-kan. I/O-kaart (out-0)
50 ' met knipperende etage-leds op (out-4)
60 '
70 ' Machinetaal-routine MUIS.BIN
80 ' Scherm-data in LIFT.PIC
90 ' Sprites-data in SPRITES.BIN
100 ' besturing via toetsenbord en muis.
110 '=============================================================================
120 ' Een programma van Theo van Dooren
130 '                   Weverstraat 26
140 '                   5524 BE Steensel (Nederland)
150 '                   telefoon: 04970-15181
160 '=============================================================================
170 SCREEN 8,2 : KEY1,"out 0,0"+CHR$(13) : EN%=0 :'EN=etage nu
180 BLOAD"LIFT.PIC",S :'inladen scherm
190 BLOAD"SPRITES.BIN",S :'inladen sprites
200 BLOAD"LIFT.BIN"   :'inladen ML
210 OPEN "GRP:" AS1 : YNU%=146 : M=0 : C=9: DEFUSR1=53275!: DEFUSR0=53248!:
DEFUSR2=53668! :'resp.  hoog- laag- en muis-ML-routines
220 '  inlezen sprites-kleuren  ---------------------------------------------
230 RESTORE 870:A$=""
240 FOR I=0 TO 15: READ A: A$=A$+CHR$(A): NEXT
250 COLOR SPRITE$(8)=A$
260 '  sprites plaatsen   -------------------------------------------------
270 S=10 : MX=110 : MY=157 : MA=11
280 FOR Y=35 TO 155 STEP 30
290 PUT SPRITE S-3,(52,Y),10,S : S=S-1 : NEXT Y :'            etage nummers ***
300 PUT SPRITE 9,(225,100),1,17 : PUT SPRITE 10,(225,100),8,18:' handwijzer ***
310 PUT SPRITE 0,(73,9),14,0 :'                                      motor ***
320 PUT SPRITE 8,(MX,MY),,MA :'                                    manneke ***
330 PUT SPRITE 1,(73,YNU%),9,4 : PUT SPRITE 2,(73,YNU%+16),9,5 :'      lift ***
340 LINE(80,23)-(80,YNU%),1 :'                                   liftkabel ***
350 POKE 54004!,0: POKE 54005!,3 :'                            beginsnelheid ***
360 OUT 4,1    ' LED etage 0 aan
370 LET EW$=INKEY$: IF EW$="" THEN 380 ELSE 620
380 IF NOT STRIG(1) THEN PUT SPRITE 8,(MX,MY),,11:FORT=0TO15:A=USR2(0):NEXT
390 IF NOT STRIG(1) THEN PUT SPRITE 8,(MX,MY),,12:FORT=0TO15:A=USR2(0):NEXT:GOTO
370 :'zwaaiend manneke
400 YCO=PEEK(54002!)
410 IF YCO>30  AND YCO<52  THEN GOTO 480 ELSE GOTO 420
420 IF YCO>60  AND YCO<82  THEN GOTO 490 ELSE GOTO 430
430 IF YCO>90  AND YCO<112 THEN GOTO 500 ELSE GOTO 440
440 IF YCO>120 AND YCO<142 THEN GOTO 510 ELSE GOTO 450
450 IF YCO>150 AND YCO<172 THEN GOTO 520 ELSE GOTO 460
460 IF YCO>198 AND YCO<212 THEN GOTO 530 ELSE GOTO 470
470 A=USR2(0): GOTO 380
480 EW%=4 : GOTO 630
490 EW%=3 : GOTO 630
500 EW%=2 : GOTO 630
```

```
510 EW%=1 : GOTO 630
520 EW%=0 : GOTO 630
530 '------------ instellen snelheid lift ----------------------
540 A=USR2(0): XCO=PEEK(54003!): IF XCO<213 OR XCO>247 THEN 370
550 LINE(213,204)-(XCO,204),255 : LINE(248,204)-(XCO+1,204),1
560 IF XCO>212 AND XCO<220 THEN POKE 54004!,0 :POKE 54005!,8 :GOTO 370
570 IF XCO>219 AND XCO<227 THEN POKE 54004!,0 :POKE 54005!,5 :GOTO 370
580 IF XCO>226 AND XCO<232 THEN POKE 54004!,0 :POKE 54005!,3 :GOTO 370
590 IF XCO>231 AND XCO<238 THEN POKE 54004!,0 :POKE 54005!,2 :GOTO 370
600 IF XCO>237 AND XCO<244 THEN POKE 54004!,0 :POKE 54005!,1 :GOTO 370
610 IF XCO>243 AND XCO<248 THEN POKE 54004!,1 :POKE 54005!,1 :GOTO 370
620 EW%=VAL(EW$)
630 IF EW%=EN% OR EW%>4 THEN 370
640 POKE 54000!,16*EW% : POKE 54001!,2^EW%:'resp. ETANMR en LEDBIT
650 OUT 4,2^EW% : MA=12 : GOSUB 760 :'manneke in lift
660 IF EW%>EN% THEN EH%=EW%-EN%: EN%=EN%+EH% :HO%=EH%*10: GOSUB 680:'omhoog
670 IF EW%<EN% THEN EL%=EN%-EW%: EN%=EN%-EL% :LA%=EL%*10: GOSUB 720:'omlaag
680 'subr. omhoog  ------------------------------------------------------------
690 A=USR(HO%)
700 MY=MY-EH%*30 : MX=75 : GOSUB 810:'manneke uit lift
710 OUT 0,0: RETURN 370
720 'subr. omlaag  ------------------------------------------------------------
730 A=USR1(LA%)
740 MY=MY+EL%*30 :MX=75 : GOSUB 810:'manneke uit lift
750 OUT 0,0: RETURN 370
760 ' subr. manneke in lift ---------------------------------------------------
770 PUT SPRITE 8,(MX,MY),,MA : MX=MX-1 : IF MX<=75 THEN 800
780 FORT=0TO1 :NEXT:MA=MA+1 : IF MA=15 THEN MA=13
790 GOTO 770
800 PUT SPRITE 8,(75,MY),,12 :RETURN
810 ' subr. manneke uit lift --------------------------------------------------
820 PUT SPRITE 8,(MX,MY),,MA : MX=MX+1 : IF MX>=110 THEN 850
830 FORT=0TO10:NEXTT:MA=MA+1 : IF MA=17 THEN MA=15
840 GOTO 820
850 PUT SPRITE 8,(110,MY),,11 :RETURN
860 ' *************  data sprites-kleuren  ************************
870 DATA 8,8,8,8,4,4,4,4,4,4,10,10,10,10,10,2
```