

UNIFACE, de buitenwereld aan de computer (2)

Klaas Robers

Wie heeft er nooit van gedroomd een grote modelspoorbaan te besturen met de computer? Of een uitgebreid meetnet in huis met sensoren in alle kamers en bestuurde kranen op alle radiatoren? Of de besturing van de verlichting van de plaat-

Het systeem

UNIFACE is een systeem. Dat betekent dat het niet zo maar een losse interface is, waarmee je een enkel randapparaat op de computer aansluit, nee, er kan veel meer en dat ook nog allemaal tegelijk. Vorige keer in PTC PRINT werden de grondbeginselen beschreven, waarbij slechts een beperkt aantal dingen kon worden bestuurd. Als voorbeeld werden daarbij de LEGO motortjes en lampjes gebruikt van de demonstratie op de open dagen. Er blijkt echter veel meer mogelijk te zijn met UNIFACE, door de buitenwereld-delen te voorzien van een selectie mogelijkheid. Maar het is misschien goed weer even met het begin

selijke toneelclub? Maar hoe sluit je dat dan wel allemaal aan op je computer? UNIFACE is het antwoord op die vraag en het kan net zo goed op de MSX als op de P2000 en hopelijk binnenkort ook op de P3100.

te beginnen.

Twee verschillende delen

UNIFACE bestaat altijd uit ten minste twee delen. Dat is het computer-deel en het buitenwereld-deel. Het computer-deel is een stukje electronica, dat op de computer wordt aangesloten. De interne elektrische signalen van de computer worden door het computer-deel zo aangepast, dat zij op een standaardmanier op een stekker beschikbaar komen. Voor de P2000T zit dit deel in een insteekdoos voor sleuf 2, voor de MSX is het een wat anders gevormde doos, die in één van de zogenaamde slots kan

*Artikel uit PTC Print 9
Copyright PTC en de auteur
Gescand en omgezet naar PDF door HansO, 2002*

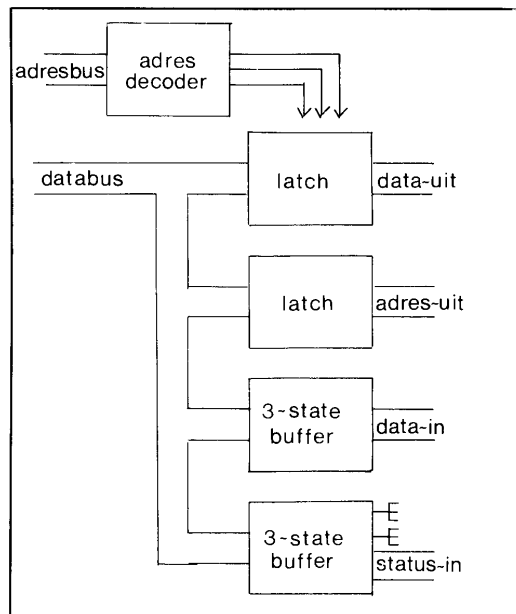
worden gestoken. In alle gevallen steekt er uit het computer-deel een 34-pennige connector, waarop een 34-aderige kabel kan worden aangesloten. De elektrische signalen op deze 34-aderige kabel zijn identiek voor het P2000-deel, het MSX-deel en het P3100-deel.

De buitenwereld

Op de 34-aderige kabel kunnen de buitenwereld-delen worden aangesloten. Wat die buitenwereld-delen doen? Ze kunnen lampjes of motortjes schakelen, maar een ander buitenwereld-deel kan een motortje in snelheid regelen. Weer een ander type kan sterkstroomlampen schakelen en dimmen. Er zijn dan nog de sensors. Die geven informatie uit de buitenwereld door naar de computer. Dat kan zijn het uit-of-aan-zijn van schakelaars, het al of niet belicht zijn van fotocellen, maar het kan ook zijn de waarde van een temperatuur of de kracht op een rekstrookje.

Interface-nummers

U ziet dat er een verschil wordt gemaakt in output-delen, waarmee iets kan worden bestuurd, en input-delen, waarmee gegevens worden vergaard. Het bijzondere van UNIFACE is dat er niet maar een enkel buitenwereld-deel kan worden aangesloten, maar een hele reeks. Dat komt omdat er aan elk buitenwereld-deel een nummer moet worden gegeven. Dat nummer moet worden ingesteld met 8 kleine schakelaartjes op de print van het buitenwereld-deel. Daarmee zijn 255 verschillende standen mogelijk en zo zijn er 255 output-delen en 255 input-delen tegelijkertijd aan te sluiten. Dat kan heel eenvoudig, want de gebruikte lintkabel kan gemakkelijk worden voorzien van een heleboel stekkers. De kabel gaat dan vanaf het computer-deel stap voor stap langs de verschillende buitenwereld-delen. Omdat al die delen een ander nummer hebben (daar moet u wel voor zorgen) kan de computer alle prints apart aanspreken, ook al staat alles zo te zien aangesloten op dezelfde kabel.



Figuur 1: blokschema van het computerdeel van de UNIFACE.

Een stukje techniek

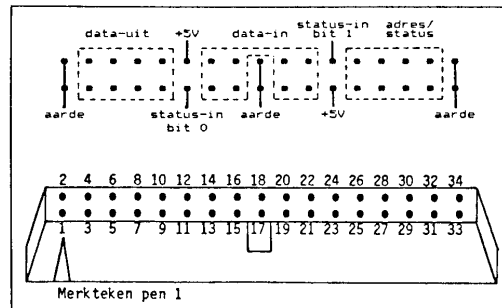
Eerst nu wat gegevens voor wie het naadje van de kous wil weten, daarna zullen we beschrijven hoe het computerprogramma de UNIFACE moet besturen. In figuur 1 staat het blokschema van het computerdeel. Het

komt in grote lijnen overeen met het principe-schema van het P2000-deel, zoals dat de vorige keer is afgedrukt. Er zijn twee output-poorten die we aanduiden als "data-uit" en "adres" en er zijn twee input-poorten die "data-in" en "status" zullen heten. Er zijn in het schema wel een paar kleine verschillen tussen de interfaces voor P2000, MSX en P3100, maar de overeenkomst is treffender. De verschillen komen hoofdzakelijk door het gebruik van andere poortnummers.

Belangrijk is de beschrijving van de UNIFACE-connector. Dit is een connector met 34 pennen, die de volgende functie hebben:

| | |
|-------------------------|-------------------------|
| pen 1: aarde | pen 2: aarde |
| pen 3: data-uit bit 0 | pen 4: data-uit bit 1 |
| pen 5: data-uit bit 2 | pen 6: data-uit bit 3 |
| pen 7: data-uit bit 4 | pen 8: data-uit bit 5 |
| pen 9: data-uit bit 6 | pen 10: data-uit bit 7 |
| pen 11: status-in bit 0 | pen 12: +5 Volt |
| pen 13: data-in bit 0 | pen 14: data-in bit 1 |
| pen 15: data-in bit 2 | pen 16: data-in bit 3 |
| pen 17: aarde | pen 18: aarde |
| pen 19: data-in bit 4 | pen 20: data-in bit 5 |
| pen 21: data-in bit 6 | pen 22: data-in bit 7 |
| pen 23: +5 Volt | pen 24: status-in bit 1 |
| pen 25: adres-uit bit 0 | pen 26: adres-uit bit 1 |
| pen 27: adres-uit bit 2 | pen 28: adres-uit bit 3 |
| pen 29: adres-uit bit 4 | pen 30: adres-uit bit 5 |
| pen 31: adres-uit bit 6 | pen 32: adres-uit bit 7 |
| pen 33: aarde | pen 34: aarde |

In figuur 2 wordt dit nog eens in een plaatje gegeven. Daaraan kunt u zien dat de connector mag worden omgedraaid zonder dat dit desastreus gevolgen heeft. Meestal kunnen de gebruikte connectors er maar op één manier in, omdat er een nokje aan een kant zit. Wat oudere typen hebben dit echter niet, alleen de bitjes en de poorten worden dan verwisseld, dus het werkt niet meer, maar nooit worden er uitgangen met uitgangen verbonden op de + en de - verwisseld.

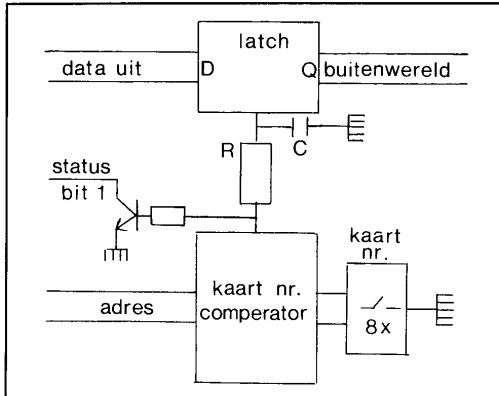


Figuur 2: schema van UNIFACE-connector.

De functie van de status-in signalen is dat daaraan gezien kan worden of een output actie of een input actie geslaagd is. We komen daar straks op terug bij het programmeren.

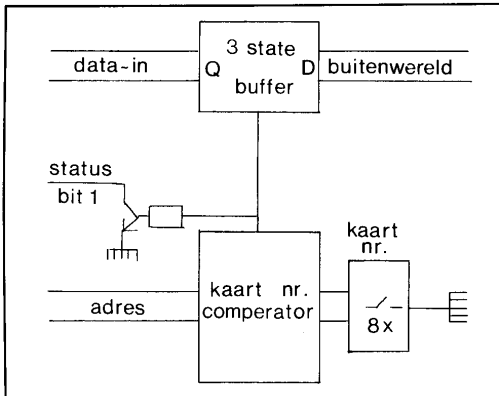
Buitenwereld-delen

In figuur 3 staat het blokschema van een buitenwereld output-deel. Te zien is dat het adres vergeleken wordt met de stand van de schakelaartjes op de print. Als die overeen komen wordt het getal op de draden "data-uit" overgenomen in een zogenaamde latch. Deze houdt het getal vast wanneer het adres veranderd wordt om een ander getal naar een ander output-deel te sturen. Merk op dat de vertraging met een weerstand en een condensator voorkomt dat een print zich heel even voelt aangesproken tijdens het wisselen van het adres. De transistor trekt de status lijn bit 0 omlaag zolang deze print zich voelt aangesproken.



Figuur 3: blokschema van een buitenwereld output-deel van UNIFACE.

In figuur 4 staat het blokschema van een buitenwereld input-deel. Dit komt grotendeels overeen met het schema van het output-deel. Het verschil is dat, wanneer de print zich voelt aangesproken, hij het getal uit de buitenwereld doorgeeft op de data-in draden. Ook nu trekt een transistor een status lijn, nu bit 1, omlaag. Een vertraging is bij deze delen niet nodig.



Figuur 4: blokschema van een buitenwereld input-deel van UNIFACE.

Statusbits

De statusbits zijn op twee manieren te gebruiken. Ten eerste geven zij aan of er een kaart is, die zich voelt aangesproken. Het programma kan zelfs in het begin alle kaartnummers een keer selecteren en kijken of de bits nul worden. Maar er kan meer mee. Stel dat er een kaart is met een UART, aangesloten op een modem. Op regelmatige tijdstippen kan er dan een nieuw byte naar toe worden gestuurd, dat vervolgens via de modem zal worden verstuurd. Zo'n UART heeft plaats voor 1 byte, we moeten dus weten wanneer er weer plaats is voor het volgende byte. Dat kan via het status-bit 0. Op de buitenwereld-print met de UART wordt het dan zo geregeld dat het status-bit 1 blijft als er geen plaats is. Het programma kan dan dus gewoon het nieuwe byte steeds opnieuw sturen totdat het status-bit 0 aangeeft dat het is geaccepteerd. Daarbij moet wel steeds het kaartadres nul gemaakt worden tussen de outputpogingen.

Met input gaat het net zo. Wanneer er geen nieuw byte is ontvangen blijft het status-bit 1 gelijk één. Het dan gelezen databyte zal in de praktijk gelijk zijn aan het vorige, maar in

principe is het ongeldige data. Pas wanneer een nieuw byte is ontvangen wordt gedurende 1 inputpoging het status-bit 1 gelijk aan nul. Het byte dat dan wordt afgegeven is het nieuw ontvangen byte.

Let op! Gedurende de outputpoging en de inputpoging, dus zolang de print geselecteerd is, mag het status-bit niet veranderen. Dus komt er toevallig net een byte binnen gedurende een inputpoging, dan blijft dat byte wachten op de volgende poging. Wanneer het status-bit dan aangeeft dat er geldige data aanwezig is mag eigenlijk die data niet veranderen zolang de kaart geselecteerd is. Dat is niet zo erg bij aangesloten schakelaars of lichtcellen, maar wel wanneer een compleet getal wordt aangeboden. Wanneer dat plotseling verandert kan het zijn dat net sommige bits van het oude en sommige van het nieuwe getal worden gelezen. En dan stelt het getal niets zinnigs voor. Tot zover de aanwijzingen voor wie zelf buitenwereld-delen wil ontwerpen.

Programmeren

Het programmeren van output en input gaat voor de computers P2000, MSX en P3100 op precies dezelfde manier dankzij het feit dat zij allen met OUT en INP() overweg kunnen. Het enige verschil is de gebruikte poortnummers. Zoals al eerder vermeld zijn er twee poortnummers in gebruik, de poort voor data en de poort voor adres/status. De poortnummers zijn als volgt toegewezen:

| | data | Adres/status |
|-------|-------|--------------|
| P2000 | &H 60 | &H 61 |
| MSX | &H30 | &H 31 |
| P3100 | &H310 | &H311 |

Om te beginnen kan het programma kijken of er wel een UNIFACE computer-deel aanwezig is. Dit kan omdat bit 6 en bit 7 van de status-poort nul zijn wanneer de interface er is. Dus:

```
IF INP(status) AND &HC0 THEN PRINT
"interface niet aanwezig"
```

Voor status moet het poortnummer van de computer worden ingevuld.

Programmeren voor output

Om een getal naar een output-deel te sturen moeten we in deze volgorde handelen:

1. Stuur het getal naar de data-out poort
2. Stuur het adres naar de adres-poort
3. Lees bit 0 van de status-poort
4. Stuur 0 naar de adres-poort

Dit laatste is alleen nodig als er meer dan één buitenwereld-deel is aangesloten. Adres 0 wordt gebruikt om geen van alle printen aan te spreken, zodat het getal op de output-poort straks rustig kan worden veranderd zonder dat dit meteen door de nog geselecteerde print wordt overgenomen. In BASIC ziet het er zo uit:

```
100 OUT data, getal
110 OUT adres, kaartnummer
120 ST = INP(status)
130 OUT adres, 0
140 IF ST AND 1 THEN PRINT "output
mislukt"
```

Voor data, adres en status het overeenkomstige poortnummer invullen. Ja, adres en status is hetzelfde getal, maar voor de duidelijkheid benoemen we he even apart.

Programmeren voor input

Om een getal van een input-deel te lezen gaan we als volgt te werk:

1. Stuur het adres naar de adres-poort
2. Lees het getal van de data-poort

3. Lees bit 1 van de status-poort
 4. Stuur 0 naar de adres-poort
- Of in BASIC:

```

200 OUT adres, kaartnummer
210 DA = INP(data)
220 ST = INP(status)
230 OUT adres, 0
240 IF ST AND 2 THEN PRINT "input
    mislukt"

```

Nu staat in DA het getal dat vanuit de buitenwereld komt.

Programmeren voor output en input

Het is mogelijk om een output-kaart en een input-kaart hetzelfde nummer te geven. In dat geval moet er gecombineerd output en input worden gedaan, want zodra de output-kaart zijn adres ziet neemt hij de data op de data-out draden over. In BASIC gaat deze gecombineerde actie als volgt:

```

300 OUT data, getal
310 OUT adres, kaartnummer
320 DA = INP(data)
330 ST = INP(status)
340 OUT adres, 0
350 RETURN

```

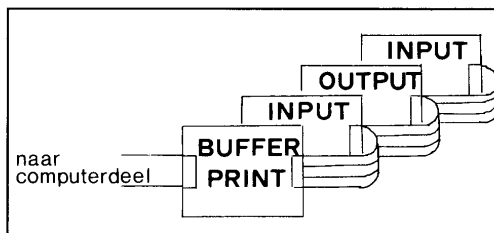
Deze actie is als subroutine uitgevoerd, want op deze manier kan er zowel output als input als beide gecombineerd worden gedaan. Deze vervangt beide bovenstaande programma-delen. Natuurlijk kan een andere programmeertaal gebruikt worden om de UNIFACE aan te sturen. Het omzetten van de instructies mag geen probleem zijn voor een enigszins ervaren programmeur.

Practische uitvoering

Over de uitvoering van de computerdelen hebben we het al gehad, voor de P2000 is het een insteekdoos voor sleuf 2, voor de MSX een slot-doos en voor de P3100 een print, die binnenin de processor-kast moet worden geplaatst. Er zijn tot nu toe twee buitenwereld-delen klaar, een 8 bits output-print en een 8 bits input-print. Op de output-print zitten 8 stroomversterkertjes volgens het schema dat de vorige keer is afgedrukt. De voedingsspanning van T3 (zie PTC PRINT nr.8, pagina 15, figuur 4) is apart op de print aan te sluiten en mag liggen tussen 5 en 24 Volt. Daarmee zijn relais, lampen, motoren, enz. te schakelen, die maximaal 500 mA trekken. De input-print bevat 8 ingangsversterkertjes zoals op dezelfde bladzijde beschreven. Hierop kunnen foto-transistoren, lichtsluizen, schakelaars, reeds-witches, enz. worden aangesloten.

Beide buitenwereld-delen zitten op een printje van 8 x 10 cm. Deze printjes passen precies in een VERO KMT-rekje. In zo'n rekje kunnen ongeveer 10 van deze

printjes. In de achterkant is ruimte voor een voedingsapparaat. Met een 34-aderige kabel met veel connectoren kunnen alle printen worden aangesloten op het computerdeel. Daartoe is het gebruik van een bufferprint erg gemakkelijk.



Figuur 5: zo wordt de bufferprint gebruikt.

Bufferprint

In principe kunnen er 255 output-delen en 255 input-delen tegelijk worden aangesloten, kaartnummer 0 wordt immers niet gebruikt. In de praktijk is dit aantal beperkt door het feit dat de gebruikte TTL-poorten niet meer dan 20 andere poorten kunnen sturen. Dus bij 20 prints houdt het op. Om deze beperking te kunnen omzeilen is er de zogenaamde bufferprint ontworpen. Aan de ene kant kunnen hierop tot 20 buitenwereld-delen worden aangesloten en aan de andere kant gedraagt deze print zich electrisch als een enkele kaart. Ook deze buffer-print kan in het rekje worden geschoven. Het hele rekje met prints gedraagt zich dan weer als een enkele kaart, dus er kunnen zo 20 rekjes met prints parallel worden aangesloten op een computer-deel. En met nog weer een extra buffer... Ja, zo komen we wel op de 255.

En dan is er natuurlijk een enkeling voor wie dat aantal te weinig is. Daarvoor maken we nog eens een selectorkaart, die net als de bufferprint tussen de kabel wordt geschakeld en die het hele achterliggende deel kan afschakelen. Zo kunnen er 256 achterliggende delen worden gecreëerd. Kortom, UNIFACE is vrijwel onbeperkt uitbreidbaar.

Wat er nog meer komt

UNIFACE staat nog maar aan het begin. Wij verwachten nog een stroom aan nieuwe buitenwereld-delen. Maar wat er komt hangt ook voor een deel af van u. Hebt u ideeën voor nieuwe buitenwereld-delen, laat het ons dan weten. En hoe u meer u er zelf al aan hebt kunnen doen, hoe gemakkelijker het is en hoe sneller de nieuwe print beschikbaar kan zijn. Zo helpt u mee een universeel bruikbaar interface-systeem op te bouwen, dat te vergelijken is met professionele systemen van veel, veel hoger prijsniveau.

Het UNIFACE ondersteuningsteam

De Universele Interface van de PTC biedt zoveel mogelijkheden dat een groot aantal enthousiastelingen bezig is deze te inventariseren. Wat is er zo allemaal bedacht? Een overzicht:

- Treinbesturing
- Weerstation
- Huisbeveiliging
- Radio-amateur toepassingen
- Maken van pianola-rollen
- Diaprojectorbesturing
- Toneelverlichting
- Overregeling

- Regeling centrale verwarming
- Robotbesturing
- Ontwikkeling analoog/digitaal omzetter
- Carillon besturing
- DOKA toepassingen
- Begeleiding sportevenementen
- etc.

Een indrukwekkende lijst! De bedoeling is dat over al deze onderwerpen in de komende maanden kleine boekjes met praktische voorbeelden, aansluitschema's en programma's verschijnen zodat gebruikers van de Universele Interface direct aan de slag kunnen.

Heeft u meer suggesties? Laat dat dan weten aan Lizet van Os, Bureau PTC, Postbus 67, 5600 AB Eindhoven.