

BESCHRIJVING I/O ROUTINES I065, versie 2.01 1986

Inhoudsopgave	Blz.
Inhoudsopgave.....	1
Inleiding.....	2
1. I/O devices.....	3
2. Functie toetsen met hun functies.....	5
2.01 ^I.....	6
2.02 ^O.....	6
2.03 ^S.....	6
3. De statusregel.....	7
4. Interrupts.....	9
5. De jumptabel.....	11
6. I/O routines.....	13
6.01 De beeldscherm output routine.....	13
6.02 De Viacom routines.....	15
6.03 RS232 interface.....	18
7 Variabelen.....	19

I065 is ontwikkeld door :

E.J.M. Visschedijk
Drakensteyn 299
7608 TR Almelo

Deze manual is geschreven door :

E.J.M. Visschedijk

Voor de programmatuur evenals deze manual geldt :
Copyright (c) 1986 Kim gebruikersclub Nederland

Inleiding

Het I065 gedeelte staat in de eprom. Dit is een 2764 exemplaar, dat echter voor de helft gevuld is. Dat komt ten eerste omdat de DOS65 gebruikers van het eerste uur een 2764 monitor eprom hebben en deze om kunnen laten programmeren, ten tweede omdat een 2764 goedkoper is dan een 2732 en ten derde omdat in de toekomst waarschijnlijk de andere helft weer gebruikt gaat worden.

I065 wordt gebruikt voor de elementaire routines als in- en output routines, IO device sturing, device initialisatie, status regel routines en de interrupt afhandeling. Ten opzichte van versie 1.01 is er nogal wat veranderd.

Wijzigingen ten opzichte van DOS 1.01 en MON65 1.01/2.:

Als het systeem aangezet wordt zal I065 zich melden. Dat gebeurt ook na een RESET. Het is dan de bedoeling dat DOS65 opgestart wordt door de bootstrap te starten. Dat gebeurt door de toets B (b) in te drukken. Dat is de enige toets buiten de functietoetsen, die op dat moment geaccepteerd wordt. De bootstrap wordt opgestart en DOS65 meldt zich in de DOS command mode. Op dat moment kunnen we eventueel MON65 van schijf halen om in de monitor te werken waar versie 1.01 automatisch in op kwam.

In I065 is ook een automatische screen-off opgenomen. Als gedurende een instelbare tijd geen toets meer is ingedrukt gaat automatisch het beeldscherm uit. Dit om te voorkomen dat het beeldscherm inbrandt.

De begin en eind adressen voor in- en output device 5 moeten nu met de hand gezet worden. Dat kan gebeuren via MON65 of via het DOS commando MEMFILL. Voordat in- of output device 5 geïntialiseerd wordt moeten eerst de variabelen WRBEG, WREND en/of REBEG, REEND gevuld worden.

Er zijn een aantal handige IO entries bijgekomen. (Zie hiervoor de jumptabel, hoofdstuk 5). Andere entries die ook in de DOS staan als PRBYT, CRLF etc. zijn vervallen. Het is nu mogelijk via de jumptabel een device te initialiseren en aan te roepen onafhankelijk van de op dat moment actieve IO devices.

1. Input en output devices

Ieder computersysteem heeft een bepaalde invoer nodig. De invoer wordt door het systeem verwerkt met een bepaald programma. Het resultaat van de invoer en dit programma moet zichtbaar worden gemaakt op een randapparaat dat voor de uitvoer zorg draagt. We noemen nu alles wat voor die invoer kan zorgen input devices en alles wat voor uitvoer kan zorgen output devices. Zo is het toetsenbord een input device en het beeldscherm of printer een output device. Iedere machine kent een of meerdere I/O devices die door de software ondersteund moeten worden. I065 houdt in principe rekening met 8 input en 8 output devices. Om bij te houden welk device nu actief is wordt voor zowel de input als de output een byte gereserveerd. In dit byte stelt ieder bit een device voor. Is een bit 1 dan is het bijbehorende device actief. Er moet altijd een (1) device actief zijn, zowel bij de input als bij de output. Met behulp van de functietoetsen worden de I/O devices geselecteerd. Als alle input en/of output devices worden uitgezet dan is dat iets wat niet mag voorkomen en dan worden automatisch de default I/O devices aangezet. De variabele waar de output device bits in staan is DEVMODOUT en de variabele waar de input device bits in staan is DEVMODINP. De default output device bits staan in de variabele OUTRET en de default input device bits staan in INPRET. De variabelen OUTRET en INPRET staan zo dat default het toetsenbord en het beeldscherm aan staan. OUTRET en INPRET zijn niet met de functietoetsen te wijzigen. Hieronder volgt een opsomming van de nummers van de I/O devices met het randapparaat dat ze aansturen.

OUTPUT DEVICE NUMMER	DEVICE OMSCHRIJVING	INPUT DEVICE NUMMER	DEVICE OMSCHRIJVING
1	beeldscherm	1	toetsenbord
2	printer centr.par.	2	niet gebruikt
3	RS232 output	3	RS232 input
4	VIACOM output	4	VIACOM input
5	memory output	5	memory input
6	vrij (DVO6VEC)	6	vrij (DVI6VEC)
7	vrij (DVO7VEC)	7	vrij (DVI7VEC)
8	vrij (DVO8VEC)	8	vrij (DVI8VEC)

Output device 1 is het beeldscherm dat meestal aan staat. In I065 staan alle routines die nodig zijn om het beeldscherm aan te sturen. Er wordt van uit gegaan dat als hardware voor het beeldscherm een Elektuur VDU kaart gebruikt wordt met 16 MHz kristal. Er is een andere karakter generator gebruikt, dit om invers video te krijgen.

Output device 2 wordt gebruikt om een printer aan te sturen. De printer moet een centronics parallel interface hebben. Als dit device wordt aangezet en de printer is niet aanwezig, wordt na een aantal seconden dit device weer uitgezet om te voorkomen dat er op een handshake wordt gewacht van een printer die niet aanwezig is.

Output device 3 is een seriele verbinding via de 6551 ACIA die op de CPU kaart aanwezig is. Deze ACIA staat default op 2400 baud. Na het selecteren van in of output device 3 worden de variabelen ACCTL en ACCMD in het control- en commandregister van de ACIA geladen. Door nu deze variabelen te wijzigen kan een andere baudrate geselecteerd worden. ACCTL staat default op \$BA en ACCMD staat default op \$05. Met het programma RS232 is het mogelijk deze waarden zeer snel in te stellen en het resultaat in een file weg te schrijven die automatisch tijdens opstarten wordt opgehaald. Op die manier wordt altijd tijdens opstarten de juiste waarde in de variabelen ACCTL en ACCMD geschreven.

Output device 4 is de zogenaamde VIACOM output. I065 is uitgerust met een speciaal communicatie programma. Deze communicatie gaat met behulp van 8 datalijnen en 2 handshakelijnen van een via. Hiermee kunnen 2 computers, die beide over VIACOM beschikken met elkaar communiceren. Als er enkele seconden geen handshake wordt ontvangen door de VIA, dan wordt automatisch dit device uitgeschakeld en het default output device ingeschakeld.

Output device 5 is het memory. I065 ziet ook het geheugen als een output device. Zo kan alle output die normaal naar het beeldscherm zou gaan, nu ook naar het geheugen gestuurd worden. Als dit device wordt aangezet wordt er eerst een geheugenbereik pointer gecopieerd waarbinnen de informatie neergezet moet worden. Wordt dit bereik overschreden, dan wordt automatisch output device 5 uitgezet.

De output devices 6, 7 en 8 zijn vrij. Als men zelf output routines wil schrijven dan kunnen die via deze output devices aangeroepen worden. Bij ieder output device hoort een vector die naar de output routine moet wijzen. De namen van deze vectoren zijn in de tabel weer te vinden.

Input device 1 is het toetsenbord. Dit is ook het default input device. Als alle input devices worden uitgezet wordt automatisch dit input device weer aangezet om te voorkomen dat de controle over het systeem wegvalt.

Input device 2 wordt niet gebruikt. Deze staat eigenlijk naast de centronics parallel output. Om hier een centronics parallel input van te maken heeft niet zoveel zin omdat ook VIACOM input als een centronics parallel input gezien kan worden.

Input device 3 is de RS232 input. Deze input gaat via dezelfde ACIA als de RS232 output. Ook deze baudrate staat default op 2400 baud. Om de baudrate en het aantal stopbits en dergelijke te wijzigen, moeten de variabelen ACCTL en ACCMD gewijzigd worden. Na een system reset staan deze waarden weer default en moeten dan opnieuw gevuld worden.

Input device 4 is de VIACOM input. De bedoeling van VIACOM is bij output device 4 al gegeven. Het gebruik van VIACOM wordt in hoofdstuk 6 uit de doeken gedaan. Als er enkele seconden geen handshake wordt ontvangen door de VIA, dan wordt automatisch dit device uitgeschakeld en het default input device ingeschakeld.

Input device 5 is het geheugen als input device. Uit het geheugen wordt gelezen en het systeem "denkt" dat de invoer gewoon van het toetsenbord komt.

De input devices 6, 7 en 8 zijn net als bij de output devices vrij door de gebruiker te definiëren.

2. Functietoetsen met hun functies

Het is mogelijk, zelfs als u een toetsenbord hebt zonder functietoetsen, functietoetsen te gebruiken. Het gebruik van deze toetsen is namelijk een bepaalde volgorde van gewone ascii waarden en niet het gebruik van toetsen met een vreemde code. Om bij een functietoets te komen moet eerst de ascii waarde \$1E weggestuurd worden. Op de meeste toetsenborden is dat control tegelijk met een circumflexe (^). Na deze \$1E wordt een gewone toets ingedrukt. Als \$1E niet gegenereerd kan worden door het toetsenbord, dan moet de variabele MONESC gewijzigd worden. De default waarde is \$1E. Omdat de control waarde van ^ volgens onze notatie ^^ is worden de functietoetsen in deze handleiding voorafgegaan door ^^.

De functietoetsen die altijd te gebruiken zijn als het keyboard het actieve input device is zijn :

- ^^I - active input devices laten zien en wijzigen
- ^^O - active output devices laten zien en wijzigen
- ^^S - aan en uitzetten van de statusregel

Let op dat de gewone toetsen van de functietoets volgorde hoofdletters zijn. Typt men kleine letters dan wordt dit niet als functietoets gezien.

2.01 Functie ^^I

Deze functietoets wordt gebruikt om te zien welke input devices actief zijn en om input device aan en uit te zetten. Nadat ^^I is ingetypt komt er een andere statusregel. Op deze statusregel staat dat we naar de input devices aan het kijken zijn. Er kan nu een getal ingetypt worden tussen de 1 en de 8. Op de statusregel zien we dat een eventuele 'on' achter een nummer verandert in een 'off'. Het device overeenkomend met het ingetypte nummer is dan uitgezet. Het is onzinnig alle input devices uit te zetten. De software zet dan ook het default input device weer aan. Als er twee input devices tegelijkertijd aangezet worden, dan wordt de meest linkse genomen. Dit in tegenstelling tot de output devices, hier kunnen er wel meerdere tegelijkertijd actief zijn. Input device 2 is zoals reeds eerder vermeld inactief.

Nadat men input device 4 heeft gekozen en 1 heeft uitgezet, moet de input van de viapoort komen volgens het VIACOM protocol. Is er echter niets aangesloten, dan wordt na ongeveer 10 seconden dit input device 4 wordt uitgezet, input device 1 gaat dan automatisch weer aan. Men moet er rekening mee houden dat wanneer er een ander input device wordt aangezet, er vaak nog op een karakter van het vorige input device gewacht werd, zodat er, als dit vorige input device b.v. het toetsenbord was, nog een dummy toets moet worden ingedrukt. De inhoud van deze dummy toets wordt wel naar het actieve output device gestuurd.

De input devices 6, 7 en 8 zijn vrij door de gebruiker te definiëren. De bijbehorende vectoren moeten naar de input routines wijzen, willen deze werken. Default wijzen deze vectoren naar een routine die het default input device weer aanzet en het zojuist aangezette device weer uitzet.

2.02 Functie ^^O

Deze functietoets wordt gebruikt om te zien welke output devices actief zijn en om output devices aan of uit te zetten. Deze functie werkt analoog aan de ^^I functietoets. Het is echter wel mogelijk meerdere output devices tegelijk aan te zetten.

2.03 Functie ^^S

Deze functietoets wordt gebruikt om de statusregel die onderin het beeld verschijnt uit te zetten. Door nog eens deze functietoets te gebruiken wordt de statusregel weer aangezet. Als de statusregel is uitgezet is het niet mogelijk toch een 25e regel op het beeldscherm te krijgen.

3. De statusregel

Het videoram op de VDU-kaart is 2 Kbyte groot. Het is hiermee mogelijk 25 regels van elk 80 karakters op te slaan. Omdat 24 regels van 80 karakters een standaard beeldscherm formaat is werd besloten de overige ruimte te gebruiken om een statusregel in het beeld te brengen. De statusregel is optioneel. Dat houdt in dat deze aan of uit gezet kan worden. Het is niet mogelijk een ander beeldscherm formaat te kiezen, bv. 25 regels van 80 karakters en geen statusregel. De statusregel kan men uitzetten met `^^S`. Normaal staat de statusregel in inverteerd video. Dit kan gewijzigd worden, door de variabele `INVST` die default op `$80` staat op `$00` te zetten. Als deze variabele gewijzigd is dan moet vanuit de DOS command mode het commando `CLEAR` gegeven worden, anders blijven enkele karakters nog inverteerd staan. Dat zijn de karakters die niet iedere seconde of na iedere toetsingave gewijzigd worden.

Er zijn drie soorten statusregels. De belangrijkste is de statusregel die verschijnt bij het aanzetten of resetten van het systeem. Van links naar rechts is er op die statusregel het volgende te zien.

- Time uu:mm:ss - Hier staat de tijd in uren, minuten en seconden. Met het DOS commando `TIME` kan deze tijd goed gezet worden.
- Date dd-mm-yy - De datum is zichtbaar in dag, maand en jaar. De datum kan gewijzigd worden met het DOS commando `TIME` of in geval van een real time clock met het commando `SETRTC`. Alleen om 00:00:00 uur wordt de datum door I065 gewijzigd. Als de bijbehorende geheugen adressen van de datum door een zelfgeschreven programma gewijzigd worden dan is dat niet direct zichtbaar op het beeldscherm. Men moet dan even de vlag `DATUPD` op `$FF` zetten. Dan wordt de datum goed gezet op het moment dat de monitor naar de keyboard input routine springt.
- Col: xx - Hier staat de kolom positie van de cursor. Dus de positie in horizontale richting. Deze waarde is 1 als de cursor geheel links staat. De maximale waarde is 80.
- Row: yy - Dit is de positie van de cursor in verticale richting. De bovenste regel is regel 1, de onderste regel is regel 24.

De volgende informatie staat alleen op de statusregel als de screen editor `ED` actief is. Na verlaten van `ED` wordt dat gedeelte van de statusregel weer gewist.

- Ln: z - Tijdens een edit sessie is altijd de positie van de cursor gegeven in Col en Row. Dit is de positie op het beeldscherm. Men weet dan echter niet welke regel dat in de gehele file is. Als een edit file uit 200 regels bestaat en de cursor staat op de onderste regel, dan is z 200.
- Fn: filename - Als men een file opent om hierin te gaan editen, dan staat op deze plaats de filenaam. Als opnieuw een file wordt geopend wordt de naam gewijzigd.

Er bestaan nog twee statusregels. Als de functie ^^0 of ^^I wordt gekozen om out- of input devices te wijzigen dan verschijnt er op de statusregel informatie over de huidige situatie van die devices. Men kan dan zien welke devices actief zijn.

Het is echter ook mogelijk zelf een statusregel te definiëren. Hiervoor zijn speciale entries gemaakt in de jumptabel. Zo is het mogelijk om :

- De gehele statusregel te wissen.
- Een eigen gedefinieerde string in de statusregel te zetten.
- De oude statusregel weer terug te roepen.

4. Interrupts

- RESET - Wordt op de RESET-toets gedrukt, dan worden alle variabelen opnieuw in hun default toestand gezet. Het uurwerk blijft gewoon doorlopen en hoeft meestal (hangt af van de reden van de RESET) niet meer op tijd gezet te worden. Het toetsenbord en het beeldscherm worden weer geïntialiseerd zodat altijd de controle over het systeem terug is. Er moet opnieuw een B in worden getypt om de bootstrap op te starten.
- NMI - Na een NMI wordt er via de NMIVECTOR naar een dummy RTI gesprongen. Deze vector kan men zelf zetten voor allerlei doeleinden.
- IRQ - Er wordt via de vector IRQVECTOR naar de IRQ routine gesprongen. Hierin wordt gekeken van welke veroorzaker de interrupt kwam. Daarna wordt er uit een tabel het adres van de te runnen interrupt routine gehaald. In deze tabel staan de absolute adressen min \$1. Dit omdat via de stack naar deze adressen gesprongen wordt. Er zijn 16 vectoren in deze tabel opgenomen. Van beide VIA's kunnen 7 interrupts komen. Verder kan er een interrupt komen van de ACIA of kan het een software interrupt zijn. De software interrupt vector wijst naar een dummy RTS. In het programma MON65 wordt die vector ongelegd om een software BREAK te kunnen genereren. De adressen van de 16 vectoren worden beschreven bij de variabelen in hoofdstuk 7. Wordt er echter geen veroorzaker van de interrupt gevonden dan wordt er via de UNRINT vector naar een routine gesprongen die 'IRQ Ignored' op het beeldscherm zet. De oorzaak wordt echter niet opgeheven zodat deze informatie blijft komen en het systeem vaak alleen door een RESET te redden is. De UNRINT vector staat direct naar een absoluut adres te wijzen. Heeft men echter een extra randapparaat aangesloten oid. die een interrupt afgeeft, dan wordt er naar de UNRINT vector gesprongen. Tijdens initialisatie van dat apparaat (bv real time clock) moet de UNRINT vector worden ongelegd en moet wijzen naar een routine die kijkt of de interrupt van dat apparaat kwam, zo niet dan moet er naar IRQ Ignored gesprongen worden, zo ja dan kan de interrupt afhandeling van dat apparaat vervolgd worden.

De interrupt routines die door de tabel worden aangewezen moeten worden afgesloten met een RTS zodat de hoofd interrupt routine in I065 de op de stack gezette waarden terug kan halen en het hoofdprogramma vervolgd kan worden. Hieronder volgt de IRQ tabel. De adressen van de vectoren staan in hoofdstuk 7.

INTV1	- T1 Via 1	in gebruik door systeem clock
INTV2	- T2 Via 1	
INTV3	- CB1 Via 1	
INTV4	- CB2 Via 1	
INTV5	- SR Via 1	
INTV6	- CA1 Via 1	in gebruik door keyboard
INTV7	- CA2 Via 1	
INTV8	- T1 Via 2	
INTV9	- T2 Via 2	
INTV10	- CB1 Via 2	
INTV11	- CB2 Via 2	
INTV12	- SR Via 2	
INTV13	- CA1 Via 2	
INTV14	- CA2 Via 2	
INTV15	- Acia	in gebruik door IO device 3
INTV16	- Software interrupt	

5. De jumptabel

De jumptabel is aangelegd om vaste entries te houden voor belangrijke routines die vanuit de DOS of andere programma's worden aangeroepen. Hieronder volgt een opsomming en beschrijving van deze routines.

- \$F000 - Algemene output routine. Zet een karakter in de accu en roep deze routine aan. Alle actieve output devices krijgen dit karakter voor uitvoer. De registers A, X en Y blijven intact.
- \$F003 - Algemene input routine. Als naar deze routine wordt gesprongen keert de routine terug met in de accu de waarde van de invoer van het actieve input device. Als er meerdere input devices actief staan, wordt alleen de meest significante genomen. Het X en Y register blijven intact.
- \$F006 - Ook dit is een output routine. Echter hier wordt het output device genomen dat in het X register staat. Staat er bv \$02 in het X register dan wordt de printer genomen. De nummers die in het X register moeten staan komen overeen met de nummers voor de output devices bij functietoets ^^0. Geldig zijn de nummers 1 t/m 8. Als echter 9 wordt genomen dan wordt er iets op de statusregel gezet. In dat geval is ook het Y register van belang. Hierin staat dan de X positie op de statusregel. Wil met op de 25e positie van de statusregel een A schrijven, dan moet in de accu \$41, in X \$09 en in Y \$19 staan. Na afloop zijn de registers vernield.
- \$F009 - Dit is een input routine die het device neemt dat met het X register aangewezen wordt. De nummers 1 t/m 8 gelden en komen overeen met die van de functietoets ^^1. Neemt men echter \$09 in het X register dan wordt naar een keyboard input routine gesprongen die kijkt of er nog iets in het input buffer staat. Zo nee dan wordt er teruggekeerd met \$00 in de accu anders met de waarde van de ingetypte toets. Na afloop zijn de registers vernield.
- \$F00C - Met deze entry is het mogelijk devices te initialiseren. Het nummer van het device wordt in het X register gezet. Om onderscheid te maken tussen input of output devices wordt het meest significante bit gezet in geval van input devices. Wil men vanuit een applicatie programma de printer initialiseren, dan moet met \$02 in het X register naar deze entry worden gesprongen. Aan de carry is te zien of de initialisatie gelukt is. Was de printer namelijk niet aangesloten, of niet geselecteerd dan wordt er teruggekeerd met de carry gezet. Wil men VIACOM als input device initialiseren dan wordt met \$83 naar deze entry gesprongen.

- \$F00F - Deze entry wordt door de editor gebruikt om het rechter gedeelte van de statuslijn te initialiseren. Er wordt dan 'Ln:' en 'Fn:' op de statuslijn gezet.
- \$F012 - In de accu (high byte) en het Y register (low byte) staat de waarde die decimaal geprint moet worden achter 'Ln:' op de statusregel. Deze entry wordt gebruikt door de editor.
- \$F015 - In de accu (high byte) en het Y register (low byte) staat het adres van de string die de filenaam voorstelt die achter 'Fn:' gezet moet worden. Deze entry wordt door de editor gebruikt.
- \$F018 - Na aanroep van deze routine wordt het rechter gedeelte van de statusregel weer gewist. Ook deze entry wordt gebruikt door de editor.
- \$F01B - Na aanroep van deze routine wordt de gehele statusregel gewist. Een lege statusregel houdt in allemaal spaties in invers video.
- \$F01E - In de accu (high byte) en het Y register (low byte) staat het adres van de string die geprint wordt op de statusregel. Hiermee kan men zelf een statusregel definiëren. De string moet worden afgesloten met \$00 en mag niet meer dan 80 karakters bevatten. Na een clear screen moet deze routine opnieuw worden aangeroepen, omdat de I065 clear screen routine niet weet waar de user defined statusregel staat. Men kan ook gewoon een clear screen vermijden in dit soort gevallen.
- \$F021 - Met deze routine is het mogelijk de standaard statusregel weer terug te roepen.
- \$F024 - Door deze routine aan te roepen met in X en Y een waarde, dan wordt de cursor direct op positie X, Y op het beeldscherm gezet. De positie geheel links bovenin is 1,1. X loopt van minimaal \$01 t/m \$50 (is 80 decimaal). Y loopt van \$01 t/m \$19 (is 25 decimaal). Door nu de volgende routine aan te roepen wordt de cursor weer terug gezet op de positie van voor de aanroep.
- \$F027 - Als de routine op \$F024 is aangeroepen kan na afloop deze worden aangeroepen om de cursor weer op de positie van voor de aanroep te zetten. Denk erom deze routines zijn niet recursief aan te roepen. Dus bij tweemaal achter elkaar aanroepen van \$F024 zonder dat er ergens een aanroep van \$F027 tussen zit wordt de eerste terugkeer positie vernield.

6. IO routines

In dit hoofdstuk wordt nader ingegaan op bepaalde I065 routines. De werking van de routines wordt beschreven en tevens waarvoor de gebruiker deze routines kan gebruiken.

6.01 De beeldscherm output routine

Alle output gaat default naar het beeldscherm. Alleen als er output devices worden omgezet kan de output naar andere devices gaan. Het beeldscherm blijft echter het belangrijkste output device. Vandaar dat hiervoor een aantal zeer speciale mogelijkheden geschapen zijn.

- Cursor besturing.
- Invers video.
- Grafics.

Het is mogelijk de cursor via 'direct cursor addressing' te besturen. Dat wil zeggen dat de cursor door een bepaalde volgorde van karakters op iedere willekeurige plaats op het beeldscherm geplaatst kan worden. Dat kan gebeuren door gebruik te maken van de jumptabel entries \$F024 en \$F027 maar ook op de hieronder beschreven manier.

Door eerst het karakter \$14 (dat is control T) naar het beeldscherm te sturen wordt de routine er op geattendeerd dat de volgende twee karakters de positie van de cursor voorstellen. Het beeldscherm bestaat uit 80 kolommen en 24 regels. (De statusregel even niet meegeteld.) De positie die opgegeven wordt moet dus vallen binnen dat veld van 80 bij 24. De waarden moeten echter hexadecimaal worden opgegeven, zodat de grenzen van het veld worden: \$01...\$50 en \$01...\$18. (In de routine zelf wordt gerekend met waarden die met \$00 beginnen, dus 1 lager, maar daar merkt de gebruiker niets van.) Na de \$14 moet eerst de kolom coördinaat en daarna de rij coördinaat worden opgegeven. Als nu een '*' op de 40e kolom en de 10e regel geprint moet worden, moeten achter-eenvolgens de volgende karakters naar de beeldscherm output routine gestuurd worden: \$14 \$28 \$0A \$2A.

Het is zo natuurlijk ook mogelijk meerdere tekens te printen. Ook meerdere cursor besturingen mogen achter elkaar staan. Wil men iets op de statusregel printen dan moet de regel coördinaat \$19 zijn. Zorg er dan voor dat dit karakter niet wordt overschreven door bv. het uurwerk. Dat kan worden voorkomen door op variabele STATTOG de waarde \$FF te zetten. De statusregel wordt dan niet meer bijgehouden. Roep dan wel even een routine aan die de statusregel schoon maakt.

Er zijn ook nog andere speciale (cursor besturings) karakters of karakters die het wissen van het gehele scherm of gedeeltes hiervan tot gevolg hebben. Deze karakters met hun uitwerking staan hieronder opgesomd.

- #07 - Bell. Geeft een piepje als de juiste hardware aanwezig is.
- #08 - Back space. Zet de cursor een positie terug. Als de cursor op positie 1,1 stond volgt een scroll down.
- #09 - Horizontal tab. De cursor komt op de eerstvolgende tabpositie te staan. Dat is om de 8 posities. Als de laatste positie op de laatste regel geweest is volgt er een scroll up.
- #0A - Line feed. De cursor gaat een regel lager. Als de cursor op de laatste regel stond volgt er een scroll up.
- #0B - Vertical tab. De cursor gaat een regel omhoog. Als de cursor op de bovenste regel stond volgt een scroll down.
- #0C - Form feed. Het beeldscherm wordt gewist. De statusregel blijft onveranderd. De cursor gaat naar positie 1,1
- #0D - Carriage return. De cursor gaat naar het begin van de regel.
- #14 - Direct cursor addressing. Na dit karakter worden twee coördinaten verwacht. Indien een of beide coördinaten niet binnen de grenzen vallen, dan wordt de besturing niet uitgevoerd. Als de eerste waarde al fout was wordt toch de tweede waarde binnengelezen, maar er wordt niets mee gedaan.
- #19 - Clear to end of screen. Maakt schoon vanaf de cursor tot en met het eind van het beeldscherm. De cursor blijft staan.
- #1A - Clear to end of line. Maakt schoon vanaf de cursor tot en met het eind van de regel. De cursor blijft staan.
- #1B - Escape. Na dit karakter kan het beeldscherm in iners of grafics gezet worden. Ook met een escape sequence wordt die mode weer hersteld.
- #1C - Home. De cursor gaat naar positie 1,1. Het beeld blijft onveranderd.

Iners video is een andere mogelijkheid van het beeldscherm. Hiermee kunnen stukken tekst geaccentueerd worden. Dit is oa. gedaan met de statusregel. Om een karakter in iners op het beeldscherm te krijgen moet of het meest significante bit van dit karakter gezet zijn, of het karakter moet voorafgegaan worden door een escape sequence. Deze escape sequence is <ESCAPE> en i oftewel #1B #69. De karakters die hierop volgen staan allemaal iners. Na de escape sequence <ESCAPE> en n oftewel #1B #6E worden de karakters weer normaal afgedrukt.

Het is ook mogelijk een beperkt aantal grafic karakters op het beeldscherm te toveren. Door vooraf iners video in te schakelen kunnen deze karakters ook iners weergegeven worden. De escape sequence voor grafics is <ESCAPE> en F oftewel #1B #46. Om de grafics mode weer uit te zetten geldt de sequence <ESCAPE> en G oftewel #1B #47.

7.02 De Viacom routines

De VIACOM I/O routines zijn ontwikkeld om twee systemen die over een dergelijk protocol beschikken snel met elkaar te kunnen laten communiceren. De communicatie gaat met 8 bits parallel met behulp van een via. Verder zijn er twee handshake lijnen nodig en een aarde zodat een 11 aderige kabel minstens nodig is om de twee systemen met elkaar te verbinden. De bedoeling van VIACOM is om programma's, stukken geheugen, enz. van het ene naar het andere systeem te sturen. De VIACOM routines zijn ondergebracht bij de I/O devices en wel nummer 4. Zet men output device 4 aan tegelijk met het beeldscherm dan zal de data die naar het beeldscherm gaat ook naar de output VIACOM routine gestuurd worden. Als er geen andere computer op de via is aangesloten dan zal het systeem ogenschijnlijk 'hangen'. Echter na ongeveer 10 seconden zal automatisch dit device worden uitgezet. Hetzelfde geldt voor input device 4. Dit kan echter niet tegelijk met het keyboard werken daar de input routine alleen het meest significante input device kiest en dat is het keyboard. Het keyboard moet dus uitgeschakeld worden. Dan zet men VIACOM input aan en weer lijkt het alsof het systeem 'hangt' als er geen ander systeem aan de andere kant van de via is aangesloten. De beide systemen moeten volgens fig. 1 op elkaar worden aangesloten. Bij gebruik van de CPU kaart van Elektuur is dat de A zijde van IC 3. Als de hardware in orde is moet volgens onderstaande beschrijvingen een programma vanuit DOS65 kunnen worden overgestuurd. De procedure is als volgt:

Zender:

- 1 - 'Type' de file met het commando TYPE op de volgende manier:
TYPE -N filenaam
- 2 - Voordat de <RETURN> achter de filenaam gegeven wordt moet eerst output device 4 aangezet worden. (^O 4)
- 3 - Geef dan de <RETURN>. Soms is nog een <RETURN> nodig.

Ontvanger:

- 1 - 'Create' een file met het CREATE commando op de volgende manier:
CREATE filenaam
- 2 - Er verschijnt een & teken als prompt. Zet nu input device 1 uit en input device 4 aan. (^I 1 4)
- 3 - Soms moet er een <RETURN> gegeven worden.

Als de file is overgestuurd duurt het 10 seconden en bij de ontvanger is het contact over het toetsenbord terug. Er moet dan nog een ^D en <RETURN> gegeven worden om de CREATE correct af te sluiten. Er moet nog wel in de file geedit worden om een ^J (Linefeed) die aan het begin van de file terecht is gekomen weg te halen, evenals een \$ teken aan het eind van de file. De zender heeft nog steeds contact over het toetsenbord en kan dus gelijk het output device 4 uitzetten. (^O 4)

De werking van VIACOM is gebaseerd op twee via's die met full auto handshake met elkaar communiseren. Fig. 2 moet hierover enige duidelijkheid verschaffen. De zender staat klaar om data te verzenden en kijkt of de CA1 lijn nul wordt, dat houdt in dat hij mag gaan zenden. Als deze lijn nul is zet de zender de data op de viapoort en maakt CA2 laag die dan de CA1 lijn in de ontvanger nul maakt. De data op de zender viapoort is nu valid. Een CA1 lijn die nul wordt zet automatisch de CA2 lijn weer hoog. Bij de ontvanger wordt CA1 nu laag die dus zijn CA2 en dus CA1 van de zender weer hoog maakt. De ontvanger kijkt ook naar zijn CA1 lijn en zodra deze laag wordt leest de ontvanger de data viapoort. Zodra de data bij de ontvanger gelezen wordt maakt deze zijn CA2 lijn laag wat tot gevolg heeft dat de CA1 lijn bij de zender ook laag wordt. Als de ontvanger nog niet klaar was met de vorige data te verwerken dan zal zijn CA1 lijn al een tijdje laag zijn voordat de nieuwe data gelezen kon worden. Als de CA1 bij de zender weer laag wordt mag de zender weer iets op de viapoort zetten. Als de zender nog bezig is met data ophalen dan is de CA1 lijn bij de zender al laag voordat deze kon zenden. Zodra de zender dus klaar is wordt direct de data verzonden.

Het bovenstaande werkt prima als alles een keer op gang is. De routine is heel simpel: Kijk naar CA1, is deze laag schrijf dan naar de viapoort of lees ervan. Is CA1 hoog blijf dan wachten, maximaal 10 seconden. Het grote probleem doet zich echter voor als de zender het eerste karakter moet versturen. Op dat moment heeft de ontvanger namelijk nog niets ontvangen en is CA1 bij de zender nog hoog. Het gevolg is dat het eerste karakter nooit verstuurd zal worden. De oplossing is echter vrij simpel. De output routine verloopt via een vector. Als het output device geïntialiseerd wordt, wordt de vector zo gezet dat het eerste karakter dat verstuurd moet worden direct, dus zonder dat er naar CA1 gekeken wordt, op de uitgangspoort van de via wordt gezet. Als dit karakter verstuurd is wordt de vector zo gezet dat voordat de volgende karakters verstuurd worden eerst naar CA1 wordt gekeken.

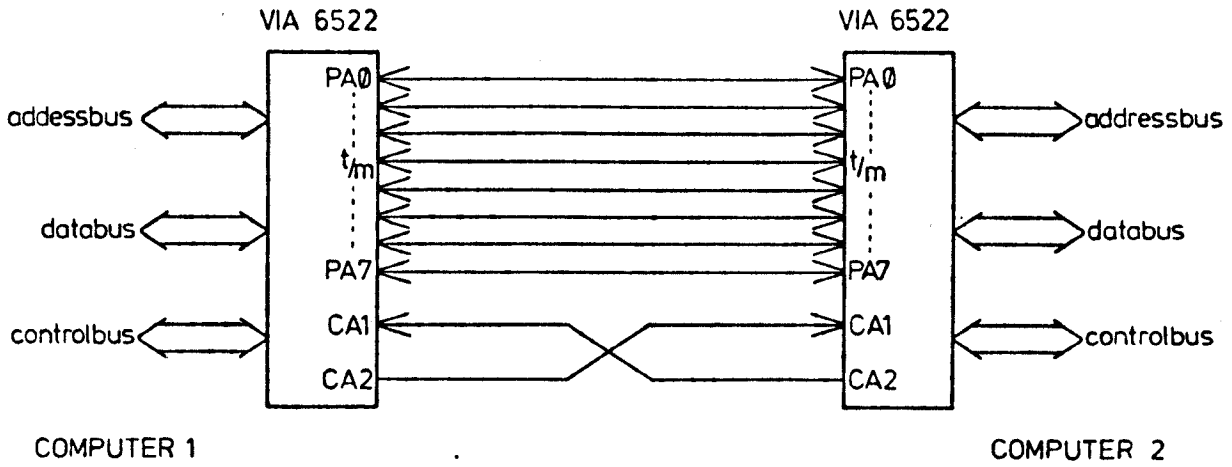


FIG. 1

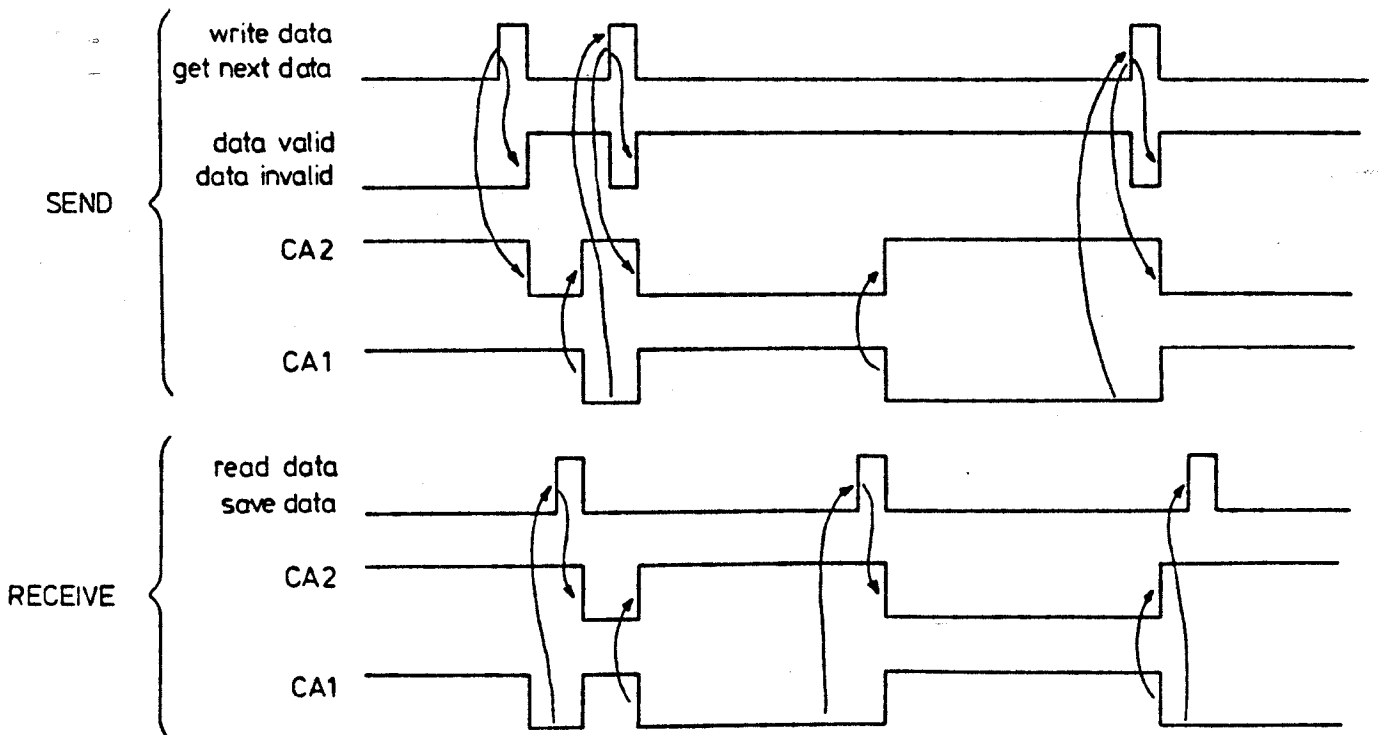


FIG. 2

6.03 RS232 Interface

Er is zowel een in- als output RS232 routine in de monitor opgenomen. Bij de input routine moet input device 3 aan- en 1 uitgezet worden. Bij het ontvangen van een receiver register full interrupt wordt het ontvangen karakter in het keyboard buffer gezet. De RS232 ontvang routine leest deze karakters uit het keyboard buffer. Het gevolg hiervan is dat er tijdens het ontvangen via RS232 ook de karakters worden ontvangen die via het toetsenbord worden gegeven. De RS232 input routine werkt zonder handshake procedure.

De RS232 output routine kan bv. voor het aansturen van een seriele printer gebruikt worden. Vanuit de DOS werkt deze printer dan niet met het PRINT commando daar dat commando de centronics uitgang neemt. De RS232 uitgang werkt wel met een handshake procedure. Als namelijk karakters worden verstuurd dan kan deze uitvoer geblokkeerd worden door een ^S naar de RS232 ingang te sturen. Na een ^Q gaat de uitvoer verder. Deze software handshake wordt door veel printers toegepast. Ook diverse modem verbindingen werken met dit protocol. Het uitzetten van dit device levert geen problemen op daar men normale controle heeft over het toetsenbord.

7. Variabelen

De bedoeling van dit hoofdstuk is om de DOS65 en I065 gebruiker duidelijk te maken welke variabelen waar staan en waarvoor hij deze eventueel zelf zou kunnen gebruiken. Hieronder volgt een opsomming van alle in I/065 gebruikte variabelen. Als eerste wordt het absolute adres genoemd van de variabele, daarna volgt de naam en de uitleg. De naam die gebruikt wordt is dezelfde als in de source listing van I065 die ook verkrijgbaar is.

Er worden een aantal zeropage variabelen gebruikt. Dat wil zeggen tijdelijk gebruikt. De variabelen op adres \$FB t/m \$FF worden door de bootstrap routine gebruikt. De variabelen PTA en PTB worden door I065 zelf gebruikt, maar na gebruik wordt de oorspronkelijke inhoud teruggezet. Deze adressen mogen door de gebruiker dus best ergens anders voor gebruikt worden. Dat houdt in, in gewone programma's. Het gaat fout als deze variabelen in interrupt routines gebruikt worden.

0000- PTA Wordt gebruikt om de statusregel te copieren bij de scroll routine, zowel bij scroll up als scroll down. (2 bytes)

0002- PTB Ook deze variabelen worden gebruikt bij de scroll routines. Er is hier voor zeropage adressen gekozen uit snelheids overwegingen. (2 bytes)

Onderstaande variabelen worden door de bootstrap routine gebruikt.

00FB- SECC TSL plaats.
00FC- RPOIN Pointer naar system blok (input buffer). (2 bytes)
00FE- ERCNT Aantal leesfouten.
00FF- ERC1 Density indikator.

De nu volgende variabelen worden door I065 gebruikt.

- E700- SAVSTACKP Dit is een software stackpointer voor een routine die A, X en Y register saved.
- E701- STATTOG Variabele die bijhoudt welke statusregel actief is. Het meest significante bit is aan of uit. 1=aan.
- E702- CRPX Variabele waarin de inhoud van CURPX wordt gezet voordat er naar de routine wordt gesprongen die Col en Row print in de statusregel.
- E703- CRPY Variabele waarin de inhoud van CURPY wordt gezet voordat er naar de routine wordt gesprongen die Col en Row print in de statusregel.
- E704- CURPX De positie in de X (horizontale) richting van de cursor. De waarde van deze variabele is 1 lager dan wat Col aangeeft. Deze waarde loopt van \$00 tot max. \$4F
- E705- CURPY De positie in de Y (verticale) richting van de cursor. De waarde van deze variabele is 1 lager dan wat Row aangeeft. Deze waarde loopt van \$00 tot max. \$17 Als er iets op de statusregel wordt geprint, dan is deze variabele \$18.
- E706- CURP Vlag die bijhoudt of er direct cursor adresssing plaatsvindt. Ook is hieraan te zien of de eerste binnengekomen coördinaat binnen de grenzen viel.
- E707- SCURPX Tijdelijk bewaar adres van CURPX.
- E708- SCURPY Tijdelijk bewaar adres van CURPY.
- E709- XTMP Tijdelijk bewaar adres voor x richting pointer.
- E70A- XPSOLD Oude X positie voor aanroep POSIT routine
- E70B- YPSOLD Oude Y positie voor aanroep POSIT routine
- E70C- OUTCHR Save adres voor te printen karakter
- E70D- MAXSTL Maximale stringlengte in routine die karakters op de statusregel zet.
- E70E- HSFLG Vlag die voor de handshake procedure ^S en ^Q zorgt in de RS232 routines.
- E70F- DEVCHOIC Als functie ^^0 wordt gekozen is deze variabele tijdelijk \$00. Bij ^^I is deze variabele \$01.
- E710- DEVMODOUT Hierin staan de active output device bits.
- E711- DEVMODINP Hierin staan de active input device bits.
- E712- SDEVMODOUT Save adres voor DEVMODOUT
- E713- SDEVMODINP Save adres voor DEVMODOUT
- E714- SDVOUT Save adres voor DEVMODOUT bij output routine.
- E715- SDVINP Save adres voor DEVMODINP bij input routine.
- E716- OUTRET Default setting voor DEVMODOUT. Als VIACOM te lang op een handshake moet wachten, als het memory output device niet meer actief is of wanneer output device 6, 7, of 8 wordt geactiveerd zonder dat deze geïnitieerd is, dan wordt deze waarde in DEVMODOUT gezet en de initiële toestand is terug.
- E717- INPRET Default setting voor DEVMODINP. Hiervoor geldt hetzelfde verhaal als voor OUTRET maar dan met betrekking tot de input devices en DEVMODINP.
- E718- CLSEC Alleen tijdens de input routine die wacht op het indrukken van een toets wordt op het beeldscherm het uurwerk bijgehouden. Deze variabele krijgt de waarde van de seconden als het uurwerk "op tijd gezet" is. Er wordt daarna gekeken of CLSEC nog gelijk is aan de seconden teller. Is dit het geval dan wordt nog even gewacht voordat het uurwerk weer gelijk wordt gezet.

- E719- FUNENA Vlag die gezet wordt als ^^ ingedrukt wordt. Oftewel er wordt een functietoets verwacht.
- E71A- FNCEN Als hier een waarde ongelijk aan \$00 in staat zijn de functietoetsen uitgeschakeld.
- E71B- VIAVRA Als er tien seconden geen handshake is geweest wordt er uit VIACOM gesprongen. VIAVRA houdt het aantal verstreken seconden bij.
- E71C- GRFLG Grafics vlag. Hierin staat \$46 als grafics aan staat. Op het beeldscherm worden dan de grafics karakters in plaats van de normale ascii karakters geprint. Deze vlag wordt gezet door de opeenvolging van de karakters <ESCAPE> en F naar de beeldscherm output routine te sturen, dus \$1B \$46. De grafics mode wordt uitgezet door <ESCAPE> en G naar de beeldscherm routine te sturen, dus \$1B \$47. Dan worden er weer gewone ascii karakters geprint. In de variabele GRFLG staat dan \$00.
- E71D- PNTXLSAVE Save adres voor PNTXL
- E71E- PNTXSAVE Save adres voor PNTXH
- E71F- PNTX Doordat de ontwerper zo weinig mogelijk gebruik wilde maken van zeropage adressen vervallen bepaalde adresseringen mogelijkheden. Bv. STA (ADR),Y. Om nu toch dat soort adresseringen te simuleren worden elders in het geheugen self modifying programma's toegepast. Op adres PNTX wordt een opcode neergezet. PNTXL en PNTXH zijn dan absolute adressen die gewijzigd kunnen worden. Op het eerste adres na PNTXH staat een RTS. De opcodes die op PNTX worden gezet hangen af van het gebruik.
- E720- PNTXL Low byte van het absolute adres van het self modifying programma PNTX.
- E721- PNTXH Low byte van PNTX.
- E723- DMPRAM Self modifying programma dat gebruikt wordt om karakters op het beeldscherm te zetten. Dit is het opcode adres, er staat altijd STA. (\$8D)
- E724- DUMPL Low byte van VDU ram adres dat gevuld gaat worden bij het eerstvolgende karakter dat naar de beeldscherm routine gaat.
- E725- DUMPH High byte van boven genoemde.
- E727- LDALET Self modifying programma dat gebruikt wordt om een hele rij van karakters naar de actieve output devices te sturen. Hier staat altijd LDA. (\$AD)
- E728- LETADRL Low byte van het absolute adres van het karakter dat deze routine ophaalt.
- E729- LETADRH High byte van boven genoemde.
- E72B- FRWR Self modifying programma dat gebruikt wordt om een karakter ergens in het geheugen te zetten bij output device 5, het memory als output device.
- E72C- FRWRL Low byte van absolute adres van FRWR.
- E72D- FRWRH High byte van absolute adres van FRWR.
- E72F- FRRE Self modifying programma dat gebruikt wordt om een karakter ergens uit het geheugen te lezen bij input device 5, het memory als input device.
- E730- FRREL Low byte van absolute adres van FRRE.
- E731- FRREH High byte van absolute adres van FRRE.
- E733- MONESC Hierin staat default \$1E. Oftewel het karakter waarna de monitor het volgende karakter als functie toets ziet.

- E734- ACCTL Deze waarde wordt bij het aanzetten van I/O device 3 de RS 232 routines in het acia control register gezet. Default staat deze variabele op \$BA wat inhoudt dat er twee stop bits en 7 data bits gebruikt worden. Tevens wordt de interne baudrate generator gebruikt die ingesteld staat op 2400 baud. Na een reset wordt de default waarde weer in ACCTL gezet.
- E735- ACCMD Bij het aanzetten van I/O device 3 wordt de inhoud van deze variabele in het command register van de acia gezet. Default staat deze waarde op \$05. Dat houdt in dat er geen parity wordt verstuurd en dat er niet naar wordt gekeken, tevens wordt de echo mode gedisabled. Er komt een transmitter control interrupt en het RTS level is laag actief. Ook is de IRQ interrupt van de acia ingeschakeld. Bit 0 is ook 1 om de gehele acia te laten werken.
- E736- TIMDAT Als de tijd op de statusregel geprint wordt is de inhoud ':', bij de datum is dat '-'.
E737- SEC1/20 Er loopt een interrupt via een timer in de VIA. Deze timer geeft iedere 1/20e seconde een interrupt. In deze variabele worden de 1/20ste delen van 1 seconde geteld. Deze waarde loopt tot maximaal \$13. De minimale waarde is 0. Dus deze teller kan tot 20 tellen.
- E738- TOUTF Vlag die normaal op \$FF staat en door de timer op \$00 wordt gezet als de screen time-out verstreken is.
- E739- TREMP Als het transmit register van de acia empty is wordt er een interrupt gegeven. Dan wordt er \$00 in deze vlag geschreven zodat de acia output routine hiaraan kan zien of het transmit register leeg is.
- E73A- DECI1 Deze drie variabelen worden gebruikt om om te rekenen van hex naar decimaal. Het uiteindelijke decimale getal staat als decimale bytes in deze variabelen.
- E73B- DECI2 Gedeelte van decimaal getal. (Zie boven)
- E73C- DECI3 Gedeelte van decimaal getal. (Zie boven)
- E73D- LNRL Low byte van een 16 bit getal dat omgerekend wordt en als decimaal getal op de statusregel wordt gezet achter Ln: De screen editor maakt hier gebruik van om aan te geven in welke regel van de file de cursor staat.
- E73E- LNRH High byte van boven genoemde.
- E73F- COMP Door de ascii waarde \$19 naar de beeldscherm routine te sturen wordt het beeldscherm schoon gemaakt vanaf de cursor tot het eind van het scherm. In deze 2 byte variabele staat op dat moment het adres van het laatste karakter op het beeldscherm dat gewist moet worden.
- E741- ESCFLG Deze vlag wordt gezet als er een <ESCAPE> naar de beeldscherm routine is gestuurd. (\$1B)
- E742- STATFG Vlag intern voor I065 die aangeeft dat er op de statusregel geprint moet worden.
- E743- INVERS Om karakters in invers video op het beeldscherm te krijgen moet deze vlag gezet zijn. Dat gebeurt door een <ESCAPE> en i naar de beeldscherm routine te sturen. De zogenaamde escape sequence wordt dus \$1B \$69. Om invers weer uit te zetten wordt <ESCAPE> en n, dus \$1B \$6E naar de beeldscherm routine gestuurd.
- E744- INVERSS Dit is het save adres voor INVERSS tijdens bepaalde routines.
- E745- INVST Hierin staat default \$80 omdat de statusregel default op invers staat. Wil men de statusregel gewoon, dan moet hier \$00 staan. Geef naderhand wel even het commando CLEAR om het beeldscherm opnieuw op te maken.

Op de volgende adressen staan de vectoren die wijzen naar de interrupt routines. Denk erom dat de adressen die in de tabel staan 1 lager zijn dan het startadres van een interrupt routine. De interrupt routine moet worden afgesloten met een RTS en niet met een RTI. De registers die op de stack zijn gezet door het hoofd interrupt programma kunnen er nu ook weer af worden gehaald door dat programma.

E750-	INTV1	interrupt 1 vector, T1 Via 1, System clock
E752-	INTV2	interrupt 2 vector, T2 Via 1
E754-	INTV3	interrupt 3 vector, CB1 Via 1
E756-	INTV4	interrupt 4 vector, CB2 Via 1
E758-	INTV5	interrupt 5 vector, SR Via 1
E75A-	INTV6	interrupt 6 vector, CA1 Via 1, Keyboard
E75C-	INTV7	interrupt 7 vector, CA2 Via 1
E75E-	INTV8	interrupt 8 vector, T1 Via 2
E760-	INTV9	interrupt 9 vector, T2 Via 2
E762-	INTV10	interrupt 10 vector, CB1 Via 2
E764-	INTV11	interrupt 11 vector, CB2 Via 2
E766-	INTV12	interrupt 12 vector, SR Via 2
E768-	INTV13	interrupt 13 vector, CA1 Via 2
E76A-	INTV14	interrupt 14 vector, CA2 Via 2
E76C-	INTV15	interrupt 15 vector, Acia, RS232
E76E-	INTV16	interrupt 16 vector, Software break, MON65
E770-	SVAINT	Accu save adres in interrupt routine.
E771-	COUD	Als de keyboard routine om input vraagt, dan wordt de cursor aangezet. De soort cursor wordt bepaald door een register in de CRTC. De waarde COUD wordt in de CRTC gezet. Als de cursor van mode wordt veranderd, dan wordt die nieuwe CRTC waarde ook in COUD gezet.
E772-	TOUTIL	I065 is uitgerust met een automatische screen-off utility. Dat houdt in dat wanneer er een bepaalde tijd geen toets is ingedrukt het beeldscherm automatisch dooft. Zodra er een toets ingedrukt wordt, gaat het beeldscherm weer aan. Voor deze timer wordt een 16 bits variable gebruikt. Deze heet TOUTL en TOUTH. De default waarde is 1800 seconden. (Half uur) Zodra er een toets ingedrukt wordt, moet de default waarde weer in TOUTL en TOUTH gezet worden. Deze default waarde wordt uit TOUTIL en TOUTIH gehaald. Om nu de time-out tijd te kunnen instellen wordt de waarde uit een variabele gehaald. Het DOS programma DPTIME stelt deze waarden in. Vanuit dat programma is het mogelijk de screen-off tijd in te stellen tussen de 0 en de 65535 seconden.
E773-	TOUTIH	Zie TOUTIL
E774-	TOUTL	Zie TOUTIL
E775-	TOUTH	Zie TOUTIL

- E776- WRBEG Om het memory als output device te gebruiken, moet voor het gebruik eerst deze pointer naar het begin van het vol te schrijven adresgebied wijzen. WREND wijst naar het eind van het gebied. Deze waarden moeten gevuld worden voordat output device 5 wordt geïnitieerd. Dat kan gebeuren door een hulpprogramma te schrijven of vanuit de monitor, of vanuit de DOS commandmode met het commando MEMFILL.
- E778- WREND Zie WRBEG
- E77A- REBEG Om het memory als input device te gebruiken, moet voor het gebruik eerst deze pointer naar het begin van het adresgebied wijzen waaruit gelezen moet worden. REEND wijst naar het eind van het gebied. Deze waarden moeten gevuld worden voordat input device 5 wordt geïnitieerd. Dat kan gebeuren door een hulpprogramma te schrijven of vanuit de monitor, of vanuit de DOS commandmode met het commando MEMFILL.
- E77C- REEND Zie REBEG
- E77E- DATUPD Als er een interrupt van de timer voor het uurwerk komt, worden bijbehorende registers bijgehouden. Alleen tijdens de keyboard input routine wordt de tijd in de statusregel ingevuld. De datum wordt niet telkens vanuit zijn registers gecopieerd naar de statusregel. Dat gebeurt alleen als er een dag-overgang is, dus 's nachts om 12 uur. Op dat moment wordt de vlag DATUPD op \$FF gezet en de routine die de tijd op de statusregel zet weet dan dat ook de datum veranderd is.
- E77F- DAY Variabele waarin de datum wordt bewaard. Deze waarde wordt om middernacht door de interrupt routine gewijzigd. Na een RESET blijft deze waarde ongewijzigd.
- E780- MONTH Variabele waarin de maand wordt bijgehouden. Deze waarde wordt door de interrupt routine om middernacht en alleen op de laatste dag van de maand gewijzigd. Er wordt rekening gehouden met 30 of 31 dagen en met schrikkeljaren. Na een RESET blijft deze waarde ongewijzigd.
- E781- YEAR Variabele waarin het jaartal wordt opgeslagen. Wordt alleen op middernacht op 31 december door de interrupt routine gewijzigd. Na een RESET blijft deze waarde ongewijzigd.
- E782- HOURS Hierin worden de uren opgeslagen van het uurwerk dat op interrupt basis loopt. Na een RESET blijft deze waarde ongewijzigd.
- E783- MINUTES Hierin worden de minuten opgeslagen van het uurwerk dat op interrupt basis loopt. Na een RESET blijft deze waarde ongewijzigd.
- E784- SECONDS Hierin worden de seconden opgeslagen van het uurwerk dat op interrupt basis loopt. Na een RESET blijft deze waarde ongewijzigd.
- E785- DVO4VEC Output device 4 gooit nadat het eerste karakter verzonden is zijn vector om. Dit is omdat er na het eerste karakter gewacht moet worden op een handshake voordat het volgende karakter verzonden wordt. Anders zou dit output device wachten op een handshake voordat er een karakter verzonden is.
- E786- DVI4VEC Voor de uniformiteit loopt ook het input device 4 via een vector.