

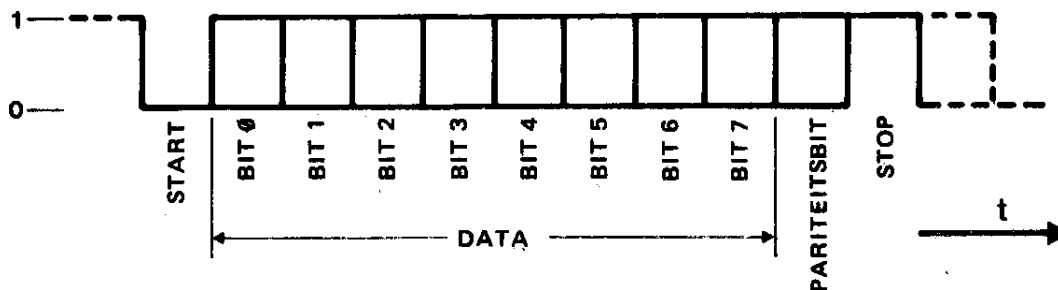
# RS-232-interface

## Seriële kommunikatie via de universele I/O-bus

RS-232-interface elektuur november 1985

*Scanned, ocr'ed and converted to PDF by HansO, 2001*

In de computerwereld komt de kreet "RS 232" vaak voor wanneer het gaat over serieel datatransport tussen computer en randapparatuur. RS 232 staat voor het geheel van gestandaardiseerde afspraken die ervoor zorgen, dat de kommunikatie heen en weer "in het gareel loopt", onafhankelijk van de individuele eigenschappen van de computer. Het gros van de computers bezit een seriële input/output-poort, waarop een printer, een modem of een andere computer kan worden aangesloten. Daar een dergelijke aansluiting toch wel tot de standaarduitrusting van een computersysteem moet behoren, hebben we een RS-232-kaart ontworpen die bestemd is voor de universele C64 I/O-bus (mei-nummer 1985). Men heeft de keuze uit TTL-lijnnivo of RS 232-lijnnivo. Wanneer we data moeten verplaatsen over niet al te lange afstanden, is parallelle transmissie de meest gebruikte methode. Het kenmerk hiervan is, dat een aantal databits tegelijkertijd verzonden worden. Voor elk bit van het over te zenden data-woord heeft men dus een draadverbinding nodig, met daarbij nog enkele verbindinglijnen die zorgdragen voor korrekte overdracht. Op zich is parallelle overdracht een erg betrouwbare kommunitatievorm. Willen we echter over langere afstanden informatie overdragen, dan wordt parallelle overdracht nogal kostbaar: elk bit vertegenwoordigt immers een lijn, terwijl ook tussenversterkers enz. meermaals uitgevoerd moeten worden. Voor grotere afstanden past men dan ook een andere vorm van data-overdracht toe: seriële transmissie. De databits worden hier achter elkaar over één lijn verzonden. In vergelijking met parallelle transmissie vergt dit systeem weliswaar een iets complexere zend- en ontvang-voorziening, maar de verbinding zelf — de kostenbepalende faktor — blijft eenvoudig. Bij draadloze data-overdracht is seriële transmissie de meest praktische oplossing. Laten we ons eens wat nader begeven op het pad van de seriële kommunikatie.



*Figuur 2. De algemene gedaante van een serieel datawoord, vergezeld van een start- en stopbit en (naar keuze) een pariteitsbit.*

## RS232

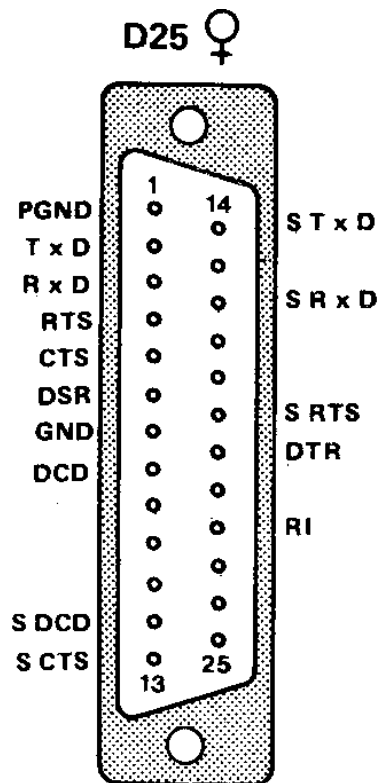
In het algemeen wordt bij communicatie-systemen een RS-232-verbinding toegepast tussen een "Data Terminal Equipment", afgekort DTE, en een "Data Communication Equipment", afgekort DCE. Met DTE wordt meestal een computer of terminal bedoeld, en met DCE bijvoorbeeld een modem. De computer zendt en ontvangt data naar en van de modem, welke zelf via een lijnverbinding met andere datasystemen verbonden is. Voor de communicatie tussen DTE en DCE moeten er eenduidige afspraken bestaan over de signalen en de afhandeling daarvan. Daar de databits één voor één verzonden worden, moet de timing van de bits en van de complete datawoorden bekend zijn. Allereerst wordt er onderscheid gemaakt tussen synchrone en asynchrone transmissie. Bij synchrone transmissie wordt er een continue datastroom overgezonden, waarbij de ontvanger gesynchroniseerd wordt op de zender. Meestal wordt de synchronisatie-klok uit het binnenkomend datasignaal zelf afgeleid, of er wordt een apart klok-signaal mee verzonden. Gangbaarder is echter asynchrone transmissie. Deze methode heeft zijn oorsprong gevonden in de vroegere telextechniek. Daar de aandrijfmotoren van de zenden- en ontvang-inrichtingen over langere perioden nooit geheel synchroon liepen, werd elke groep databits voorafgegaan door een startteken en afgesloten met een stop-teken. Voor de relatief korte periode van het datawoord is de synchronisatie dan toch voldoende. In figuur 2 zien we een voorbeeld van een dergelijk serieel signaal. Allereerst komt het startbit. Hierna volgen 8 databits (bijvoorbeeld een ASCII-teken). Het pariteitsbit dient voor foutcontrole en kan eventueel weggelaten worden. Het stopbit sluit het seriële datawoord af. Door middel van stopbit-controle kan nagegaan worden of zend- en ontvangsnelheid daadwerkelijk overeenkomen.

De korrekte afhandeling van de verzonden en ontvangen datastroom tussen DTE en DCE is gebaseerd op een protocol. Dit kan opgevat worden als een "spelreglement" om het dataverkeer in goede banen te leiden. In het algemeen gebruikt men hiervoor de officiële RS-232- of V24-aanbevelingen. Naast de dataverbinding bestaan er een aantal besturingslijnen voor het onderling besturen van DTE en DCE. In figuur 3 staan al deze signalen, samen met de aansluitingen op de standaard D-konnektor, aangegeven. DTR, DSR en DCD zijn typische signalen die gebruikt worden bij het tot stand brengen en afbreken van de verbinding.

De andere signalen komen aan bod als de verbinding er al is. Laten we eens nagaan hoe de loop van deze signalen is. We gaan uit van een full-duplex-verbinding, d.w.z. de RxD- en TxD-lijn beide in gebruik.

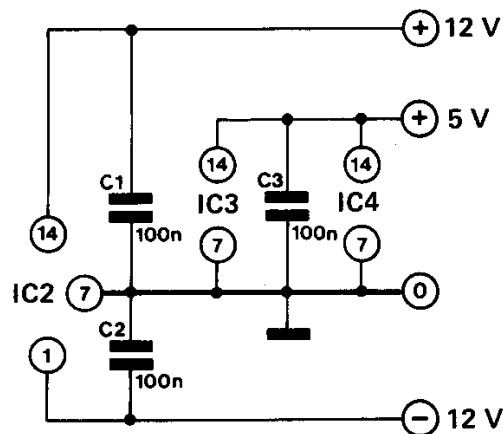
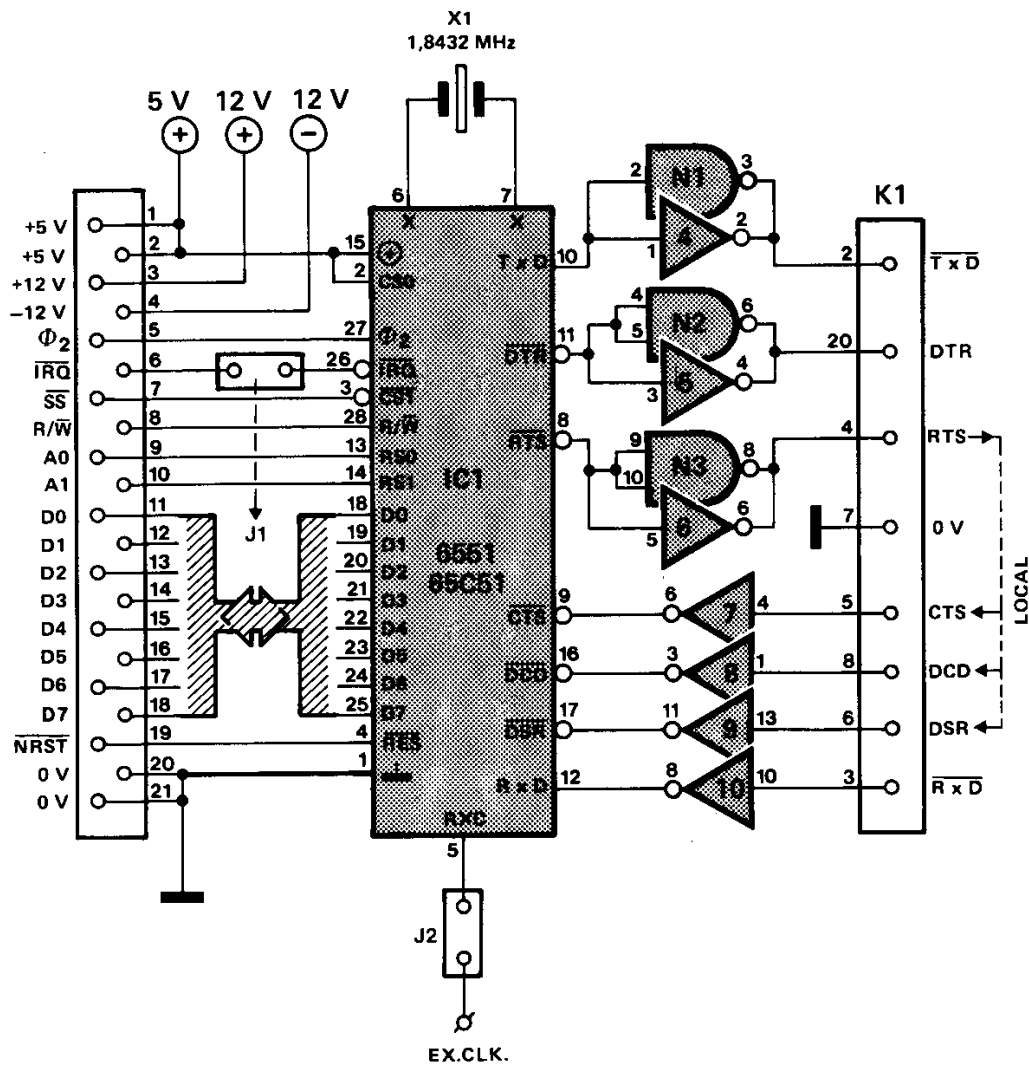
De DTE maakt de DTR-lijn actief, ten teken dat er een verbinding gevraagd wordt. De modem meldt dit terug door het aktiveren van de DSR-lijn (Data-Set = de DCE zelf). Wanneer de DTE wil zenden, maakt deze de RTS-lijn actief. De DCE geeft te kennen dat hij de data kan verwerken door het aktiveren van CTS. Bij full-duplex-verbindingen kan er altijd ontvangen worden door de DTE met als voorwaarde dat de modem het DCD-signaal actief heeft gemaakt. Dit gebeurt al bij het opstarten van de verbinding. (Zie ook Elektuur sept. '84: "data per telefoon" en "Telektor") De "secondary"-aansluitingen hebben te maken met de opsplitsing main-channel/back-channel bij modems. Om dit geheel van handelingen te vertalen naar een gebruiksklare seriële aansluiting op onze computer (of beter gezegd: op de universele I/O-bus) maken we gebruik van een intelligente periferie-chip: de 6551 ACIA (Asynchronous Communications Interface

Adapter). Kwa hardware én software is deze bouwsteen eenvoudig in een bestaand microprocessorsysteem onder te brengen, en levert een complete RS-232/V24-kompatibele verbinding met de buitenwereld op.



PGND	protective ground
T x D	transmit data
RxD	receive data
RTS	request to send
CTS	clear to send
DSR	Data-set ready
GND	Ground
DCD	Data carrier detect
DTR	Data-terminal ready
S DCD	Secondary data carrier detect
S CTS	Secondary clear to send
S TxD	Secondary transmitted data
S RxD	Secondary received data
S RTS	secondary request to send
RI	ring indicator

*Figuur 3. De gestandaardiseerde aansluitingen voor een RS-232 verbinding, uitgevoerd op een 25 polige D-konnektor. De "secondary" -aansluitingen en de RI-aansluiting worden alleen aan de modem-kant ge bruikt.*



N1 ... N3 =  $\frac{1}{4}$  IC2 = 75188; 1488  
 N4 ... N6 =  $\frac{1}{2}$  IC3 = 74LS04  
 N7 ... N10 = IC4 = 75189; 1489

Figuur 4. Het schema van de RS-232-kaart. Jumper J1 en J2 kunnen naar keuze worden aangebracht. Ook moet er een keuze gemaakt worden tussen het drietal N1/N2/N3 (RS232-drivers) of N4/N5/N6 (TTL-drivers).

## De interface-schakeling

In figuur 4 zien we dat IC1 niet meer nodig heeft dan een kristal en een aantal poorten voor signaalnivo-aanpassing om zijn taak uit te voeren. Aan de linkerzijde van IC 1 herkennen we de typische aansluitingen op het computersysteem (hier zijn dat de slot-aansluitingen van de universele I/O-bus). Het 1,8432-MHz-kristal, dat aangesloten wordt op de pennen 6 en 7 van het IC, vormt de basis van een aantal, via software te kiezen standaard baud-rates. De R x C-aansluiting is een bi-direktionele in/uitgang voor resp. een externe ontvangklok voor afwijkende baudrates, of voor weergave van de ontvangklok van de interne baudrate-generator. De klofrequentie bedraagt in beide gevallen 16 x de baudrate. Aan de rechterzijde van IC1 komen we de bekende RS-232-lijnsignalen tegen met gebufferde uitgangen en ingangen. Vanuit het IC gezien worden alle signalen geïn-verteerd, wat inhoudt dat alle besturingssignalen actief hoog zijn (DTR, RTS, CTS, DCD en DSR) en de data-signalen actief laag zijn (T x D en R x D). Deze nivo's zijn standaard voor RS-232-verbindingen. Wat betreft de spanningswaarden die bij de signaalnivo's behoren, zijn er twee mogelijkheden: TTL (H = +5 V, L = 0 V) en RS-232-lijn (H = +3 V... +25 V, nominaal 12 V en L = -3 V.. -25 V, nominaal -12 V). In het schema staan NI, N2 en N3 parallel aan N4, N5 en N6 getekend. Het is de bedoeling, dat men één van deze sets kiest. N4, N5 en N6 zenden signalen op TTL-nivo uit, terwijl NI, N2 en N3 uitzenden op RS-232-nivo. De ontvang-buffers zijn geschikt voor zowel TTL- en RS-232-nivo's. Men moet dus even nagaan met welke nivo's het aan te sluiten apparaat werkt. Op de print (figuur 5) zijn zowel IC2 én IC3 aangeduid. Kiest men voor RS 232, dan wordt alleen IC3 in het voetje geprikt, en voor TTL alleen IC2. In het laatste geval is de + en -12-V-voeding niet nodig. Voor de opbouw van de print gelden de gebruikelijke regels, zoals bijvoorbeeld het gebruiken van goede IC-voetjes.

## Het gebruik: een duikje in de software

De opbouw van het interface-printje zult u ongetwijfeld zonder problemen geklaard hebben. De schakeling draait in feite om de ACIA, van buiten gezien een gewone 28-penner, maar van binnen heel wat mans. Laten we het inwendige van het IC eens verkennen. Figuur 6 toont ons de structuur van het geheel. De kommunika-tie aan de computerzijde geschiedt door manipulatie in een vijftal registers. In figuur 7a staat aangegeven hoe de registers gerangschikt zijn als functie van de ingangen RS0 en RS1. Daar deze ingangen verbonden zijn met de adreslijnen A0 en A1, liggen de registers opeenvolgend in het adresbereik van de I/O-bus. Wanneer men bijvoorbeeld de interface-kaart in slot 1 van de universele I/O-bus stopt, en stelt men het startadres van de I/O-range met behulp van de DIP-switches in op 4000hex, dan bevinden zich het Transmit en Receive Data Register op 4000hex, het Statusregister op 4001hex, het Commandregister op 4002hex en het Controlregister op 4003hex. We zullen de functie van al deze registers eens doornemen:

Transmit en Receive Data Register Bij zenden wordt bit 0 (LSB) het eerst uitgezonden. De niet-gebruikte bits (bijvoorbeeld bit 5, 6 en 7 bij een gekozen 5-bits-formaat) worden

als "don't care" beschouwd. In de ontvang-mode komt het eerst ontvangen databit op bitplaats 0, de volgende databits op bitplaats 1, 2 enz. De hoogste, niet gebruikte bitplaatsen worden met een 0 gevuld. Statusregister

Dit register kan alleen uitgelezen worden. Bit 0, 1 en 2 geven aan, of er tijdens ontvangst een pariteitsfout, een framing-fout (stopbit-kontrol) of een overrun-fout is opgetreden. Het belangrijkste zijn echter bit 3 en 4. Deze geven aan of er een volledig karakter verzonden of ontvangen is. Deze bits moeten in het besturingsprogramma afgevraagd worden om te beslissen of een volgend karakter aan de beurt is. Via de laatste drie bits worden de DCD- en DSR-lijnen en de interrupt-status uitgelezen. Wanneer men voor de besturing van de interface-kaart een interrupt-struktuur toepast, moet men na het detecteren van een interrupt als eerste het statusregister testen om te beslissen hoe er verder gehandeld gaat worden. Wanneer men afziet van een dergelijke interrupt-struktuur, wordt jumper J2 niet aangebracht, d.w.z. de interrupt wordt wel gegenereerd, maar wordt niet doorgegeven naar het computersysteem. Wanneer een (willekeurige) schrijf-operatie naar het statusregister plaatsvindt, treedt een "geprogrammeerde reset" op. In figuur 7 staat bij elk register aangegeven wat het effect is van deze reset. Een streepje betekent "onbepaald". Command Register

Bit 0 bepaalt het DTR-sigitaal en de ontvangerstatus. Via bit 1 wordt beslist of er al dan niet een interrupt ten gevolge van een vol Receive Data Register wordt gegeven. Bit 2 en 3 besturen het RTS-sigitaal en daarmee de zender-operatie. Ook wordt hiermee bepaald of een interrupt ten gevolge van een leeg Transmitter Data Register wordt gegeven. Bit 4 is normaliter 01 Bit 5, 6 en 7 hebben te maken met pariteitsbesturing en -kontrol van zender en ontvanger.

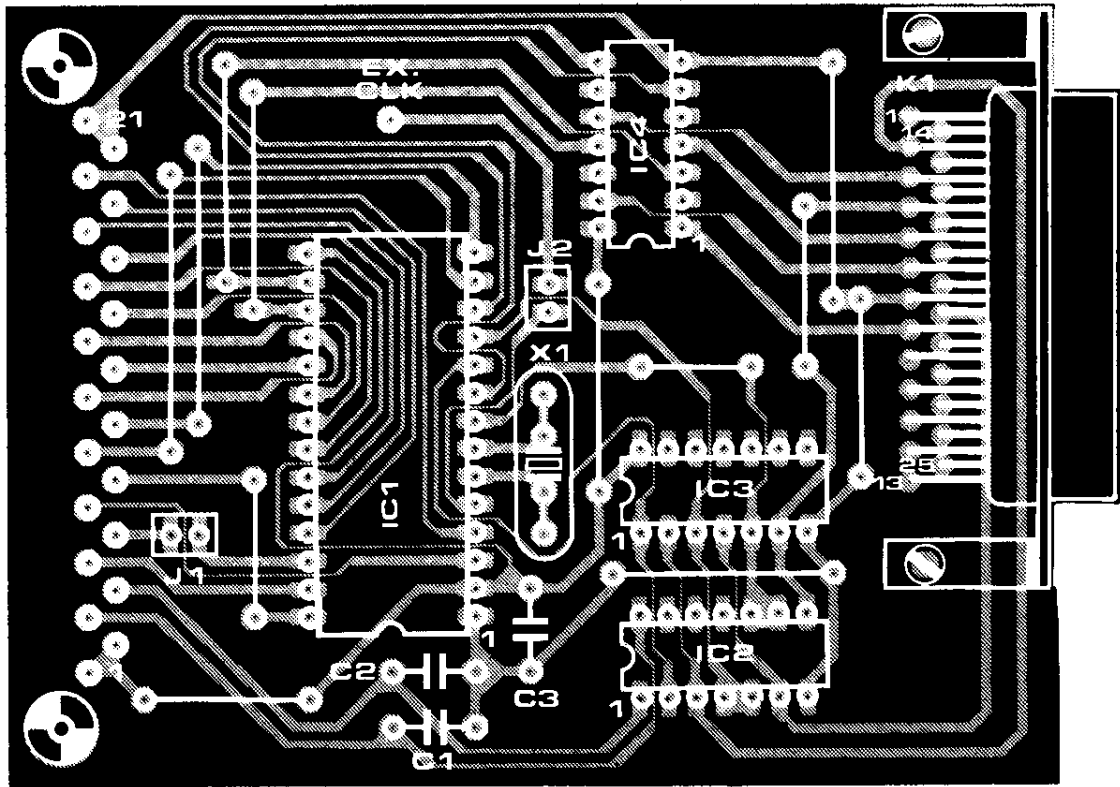
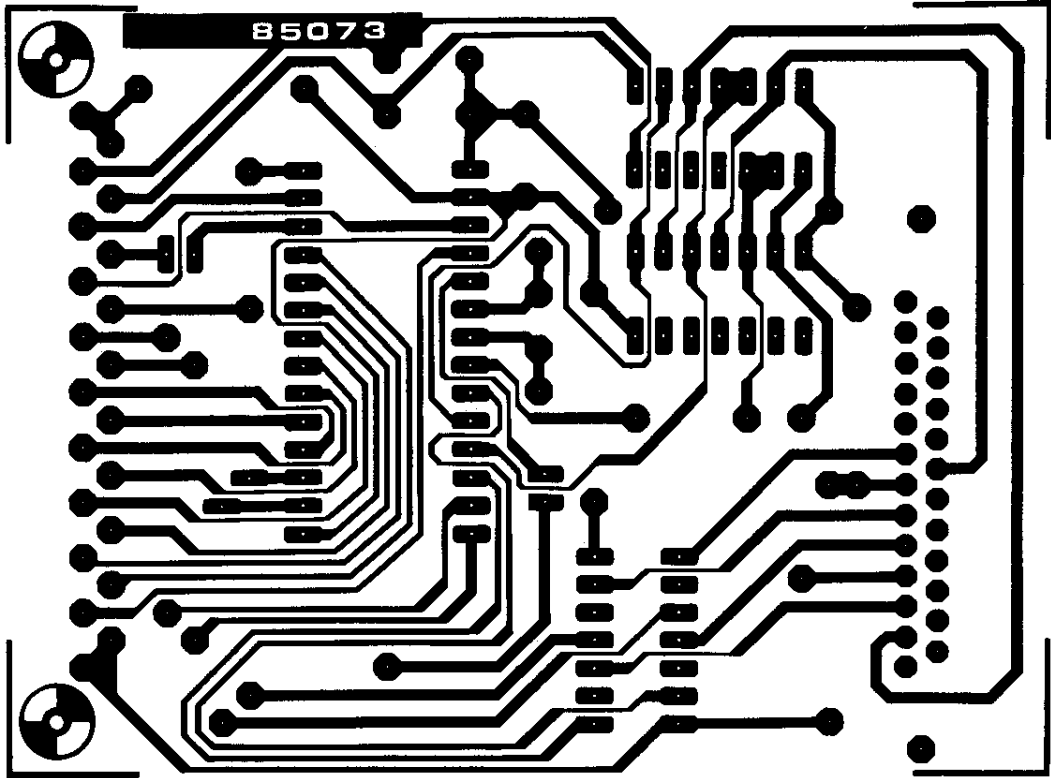
## Control Register

Via dit register wordt het formaat van de seriële data bepaald. Bit 0... 3 bepalen de baudrate: er kan gekozen worden uit een aantal standaard-baudrates, afgeleid uit de kristalfrekwentie. In het geval "16 x external clock" kan men een apart kloksigitaal aan pin 6 van IC1 aanbieden (pin 7 open). De baudrate bedraagt 1/16 deel van deze frekwentie. De interne baudrate-generator wordt naar de ontvanger doorgeschakeld als bit 4 "1" is. Zender en ontvanger werken dan met dezelfde baudrate en pin 5 van IC1 (RxC) doet dan dienst als uitgang van de interne generator (frekwentie = 16 x de ingestelde baudrate). Men kan zo meerdere ACIA's aan elkaar koppelen. Wanneer bit 4 "0" is, doet de Rx C-aansluiting dienst als ingang voor de ontvanger-clock via jumper J2. Bit 5 en 6 bepalen de lengte van het datawoord. Met bit 7 kunnen er 1, IVz of 2 stopbits gekozen worden.

U ziet dat er heel wat geprogrammeerd kan worden. Met wat kennis van machinetaal en een bescheiden portie kunst- en vliegwerk kan ieder aan de hand van de bovenstaande beschrijving zijn specifieke seriële verbinding software-matig in elkaar knutselen.

Wat betreft de verbinding met de randapparatuur kan er onderscheid gemaakt worden in de combinatie DTE-DCE (interface-kaart-modem) of de combinatie DTE-DTE (interface-kaart met een andere computer, terminal of printer). In het laatste geval moet de verbindingkabel zodanig uitgevoerd worden, dat de functie van de DCE zoveel mogelijk wordt gesimuleerd. In figuur 8 zien we een aantal verbindingen tussen interface-kaart en rand-apparatuur. In figuur 8a is het rand-apparaat een modem, in de

andere gevallen wordt de modemfunctie van het aangesloten apparaat in meer of mindere mate gesimuleerd door de kabelverbindingen. 8b en 8c zijn de eenvoudigste versies van de zogenaamde kruisverbindingen of nul-modem-verbindingen. De zend- en ontvangdatalijnen zijn kruislings met elkaar verbonden, terwijl beide systemen hun eigen controle-signalen opwekken. Eventuele control-karakters worden over de data-lijnen over en weer verzonden. Speciaal voor onze 6551-kaart is de "local-mode" - verbinding, als in figuur 8c getekend, het meest geëigend. In figuur 8d worden er wat meer controle-signalen gebruikt. Bij het opstarten wordt onderling het DTR-DSR-mechanisme verzorgd. Wanneer het ene apparaat een RTS geeft (en dus zijn eigen CTS verzorgt), wordt aan de andere kant DCD geaktiveerd, zodat deze op ontvangen overschakelt. Het ligt dus geheel aan de toepassing, hoe de onderlinge verbinding gemaakt moet worden. Tot slot geven we u nog een toepassingsvoorbeeldje voor onze interface-kaart: een RTTY-ontvang-programma van eenvoudige opzet, het programma is oorspronkelijk bedoeld voor de Acorn Atom, maar kan met enkele wijzigingen eenvoudig aangepast worden op elke andere 6502-computer (die natuurlijk wel met de universele I/O-bus én deze RS-232-kaart uitgerust moet zijn).





*Figuur 5. Komponentenzijde en koper-layout van de RS-232-kaart. Aan de ene zijde een konnektor die in de slots van de universele I/O-bus past, en aan de andere kant de standaard RS-232-konnektor.*

### **Onderdelenlijst**

Kondensatoren:

C1,C2,C3 = 100 n

Halfgeleiders:

IC1 = 6551, 65C51

IC2 = 75188, 1488

IC3 = 74LS04

IC4 = 75189, 1489

Diversen:

X1 = kristal 1,8432 MHz

K1 = 25-polige haakse sub-

D-konnektor, female 21-polige haakse konnektor,male, volgens DIN 41617

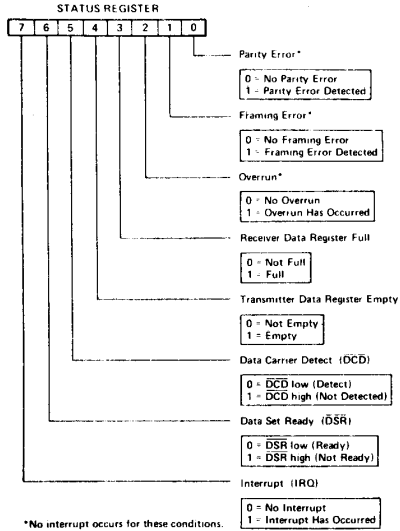
2 jumpers met bijbehorende konnektorstrip

Geschatte bouwkosten: f 115,— (inkl. konnektors)

7a

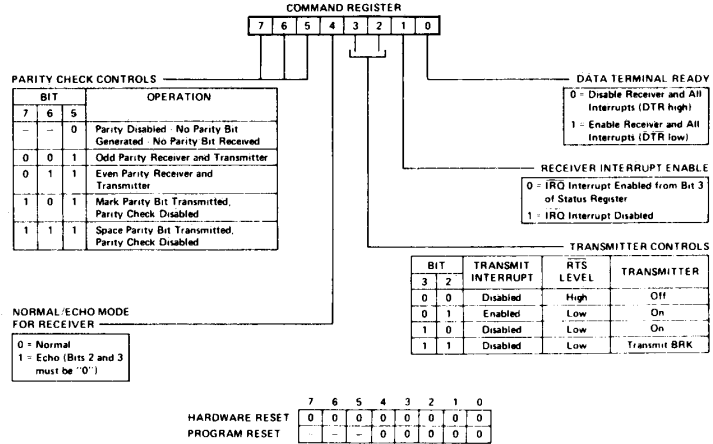
RS <sub>1</sub>	RS <sub>0</sub>	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

b

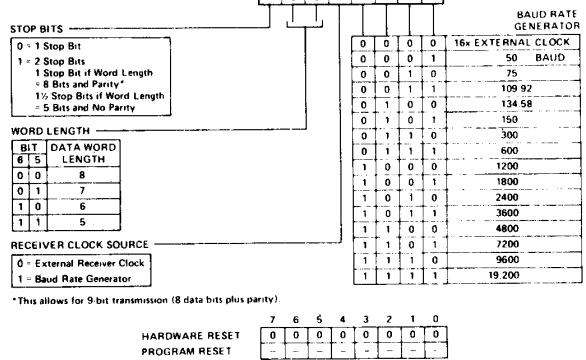


7	6	5	4	3	2	1	0
HARDWARE RESET							
0	-	-	1	0	0	0	0
PROGRAM RESET							
-	-	-	-	0	-	-	-

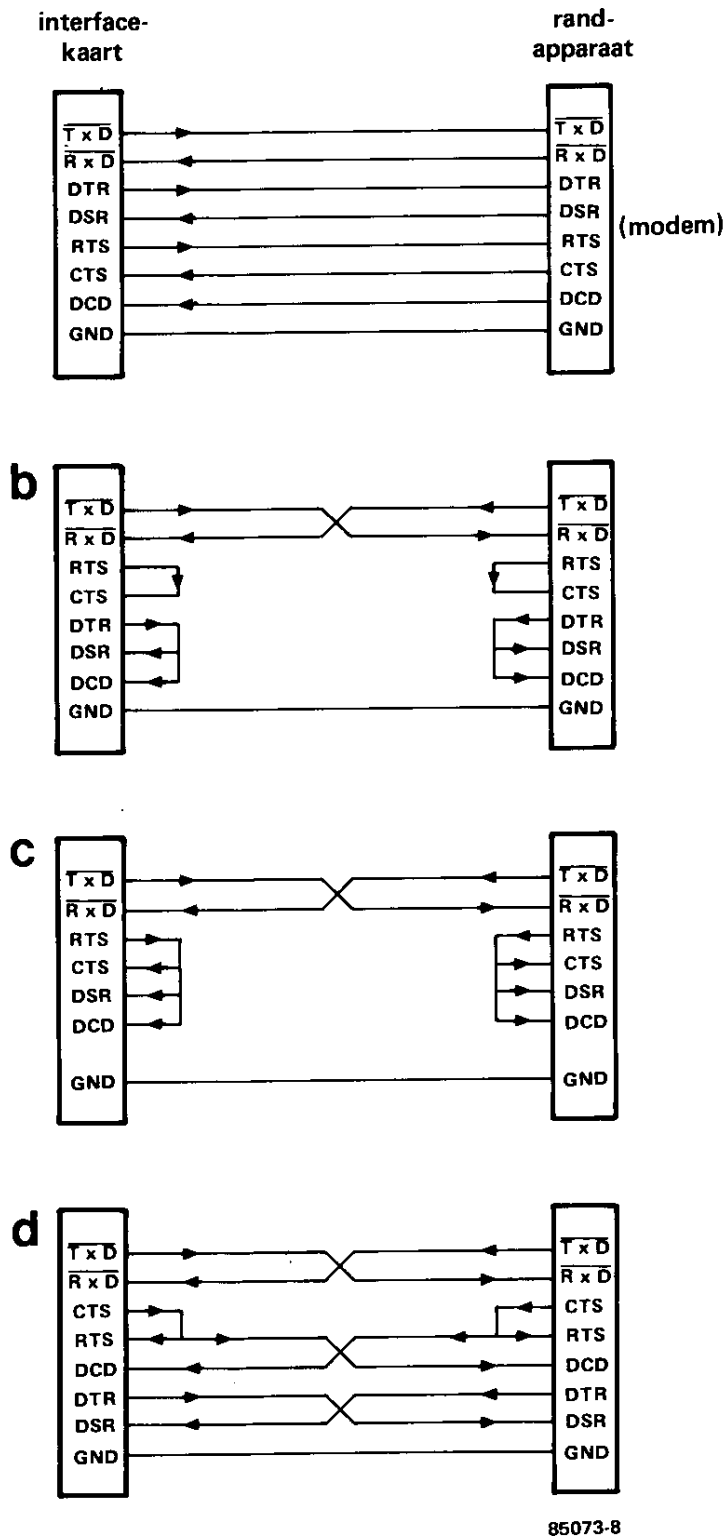
c



d



Figuur 7. Een totaaloverzicht van de registers waarmee de ACIA geprogrammeerd kan worden.



Figuur 8. Een viertal voorbeelden van RS-232-verbindingen. De laatste drie voorbeelden zijn bedoeld voor communicatie tussen twee DTE's (bijvoorbeeld tussen twee computers onderling).

