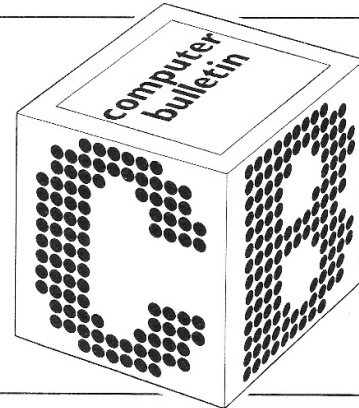

COMPUTER BULLETIN



ACIA6850

Serial input and output

H. J. C. Otten

This article introduces the ACIA6850. As a widely used IC for serial communication in microprocessor systems, the ACIA is an example of similar ICs. Besides the general characteristics of the ACIA and of serial communication, programming and a practical implementation are covered.

Important building blocks in a computer system serve for data communication. Data communication deals with the transport of data between computers and between computers and peripherals such as terminals, printers and the like. This requires a connection and an agreement about the data to be transported.

Two kinds of connection are possible:

- Parallel: each bit of the data to be transported needs its own wire.
- Serial: the bits are transported one after another over a single line.

We have already mentioned the elementary form of data: the bit. In general, data is organised as a group of bits: a character. The character is interpreted as a code by both devices connected to each other by data communication. A well-known example is the 7-bit ASCII code. The agreement in data communication consists first of this character code and then of the order in which the characters are sent. In data communication this agreement is called a protocol.

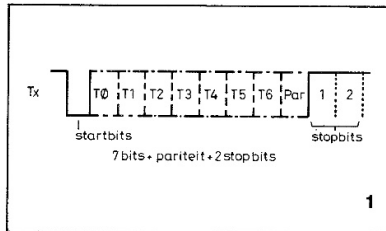
Asynchronous serial communication

In serial communication the bits follow one another. To be recognised by the receiver, receiver and transmitter must know when a bit has its valid value and how long it is present on the line. Two techniques can be used for this:- Synchronous: transmitter and

clock are synchronised by a common clock signal.

- Asynchronous: from the change on the line the receiver can see that a character follows.

The absence of a common clock signal in asynchronous serial communication does not mean that there are no fixed agreements about timing between transmitter and receiver. The time during which a bit is validly present on the line (the bit time) determines the transmission speed. The speed is measured in baud, defined as the number of bits per second that can be transmitted in this way.

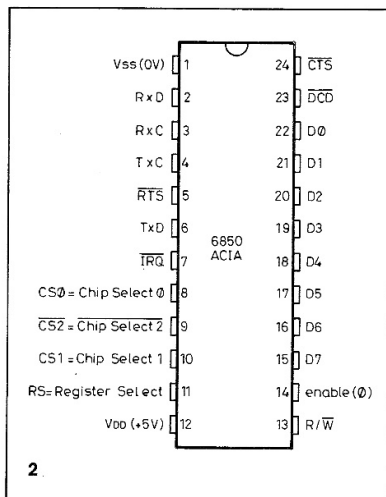


Character format

At rest, that is when no characters are being sent, the line is in the "MARK" state. The start of a character is formed by a so-called start bit, which begins with a transition to the "SPACE" state.

The start bit is followed by the data bits which together form the character. There can be 5 to 8 bits.

To detect whether transmission is proceeding correctly, a parity bit is often added. With even parity the parity bit is true ("1") if an odd number of bits in the character are true ("1"). Odd parity is the reverse. The end of a character is indicated by one or two stop bits, starting with a transition to the "MARK" state and thereby automatically passing into the rest state.



ACIA6850

The IC developed by Motorola, the ACIA6850 (Asynchronous Communications Interface Adapter), is able to handle the asynchronous serial data transport described above. The IC takes care of the parallel-to-serial conversion, of transmitting at the desired baud rate with start, parity and stop bits as required, and of receiving serially transmitted characters. In fact the IC consists of two almost independent parts: a transmitter and a receiver for serial asynchronous data communication.

As a member of the 6800 IC family, the ACIA is easy to use in microprocessor systems based on the 6502, 6800 and 6809. Although in this article we limit ourselves to these microprocessors, it is not difficult to use the ACIA in other systems as well.

The most common application of the ACIA is communication between terminal and computer, but computer-to-computer, printer and other peripheral communication is also realised with the ACIA.

Fig. 2 shows the pinout of the ACIA. The following signals are needed for

communication with the microprocessor system:

- 8-bit data bus, D0 through D7.
- E clock signal. In 6502 systems this is $\phi 2$: the system clock.
- IRQ output. This allows interrupts to the microprocessor on transmitting and receiving.
- Select inputs: CS0 and CS1 must be true and CS2 must be false to address the ACIA.

The ACIA occupies two addresses in the memory map. The ACIA contains four registers. R/W (Read/Write) selects between two read-only registers and two write-only registers. The RS signal (Register Select) selects one of the two addresses.

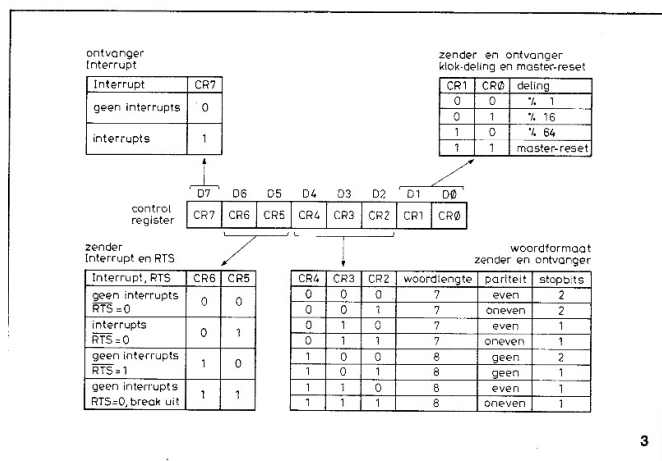
A notable feature is the absence of a RESET input on the ACIA. The IC has a built-in power-on reset circuit and, as will be discussed later, can be reset by software. The ACIA requires a supply voltage of +5 V and a ground connection.

Peripheral interface

The ACIA has three kinds of connections for communication with peripherals:

- Data transport. TxD is the output on which the data appears serially. RxD is the input to which the serial data must be applied.
- Clock signals. The baud rate at which data is transmitted can be set separately for transmitter and receiver. RxC is the clock input for the receiver and TxC the clock input for the transmitter. Under software control this clock signal is divided by a factor of 1, 16 or 64 to obtain the baud rate. Because on reception the data is sampled at the clock frequency, the divide ratio of 1 can only be used if the clock is taken from the transmitting side of the peripheral. With a divide ratio of 16 or 64, the clock signal need not come from the peripheral.
- Control. Two inputs and one output are available to control the communication. Clear To Send is an input with which transmission of a character can be held off, for example because the peripheral is still busy with the previously sent character. The CTS input must be high to be able to send a character. Data Carrier Detect is an input that controls the ACIA's receiver. When the DCD input is low, the ACIA can receive characters. A low-to-high transition can generate an interrupt if desired. If the peripheral is a modem this signal is used; otherwise it has few applications. Request To Send is an output that is set under program control.

The combination of CTS and RTS is usually used to establish a handshake between two devices. By connecting CTS of one device to RTS of the other peripheral and vice versa, sending and receiving can be fully controlled by both devices.



Registers

As already mentioned, the ACIA contains four registers. Two of them serve to control the data transfer: the Control Register (CR) and the Status Register (SR). The remaining two are the Receive Data Register (RDR) and the Transmit Data Register (TDR), which hold the character received or to be transmitted respectively.

Because only one input (Register Select) is available for selection, of the registers only RDR and SR can be read, and only TDR and CR can be written. The R/W signal distinguishes between registers that share the same address. Table 1 shows the register selection.

Transmit and Receive Data Registers

The character to be transmitted is written into the TDR register. A received character can be read from the RDR register. Because of the selection described above, both registers are at the same address, the RDR register being read-only and the TDR register write-only.

Control Register

The Control Register has the same address as the SR register; the registers are distinguished by reading, which selects the SR register, and writing, which selects the CR register.

The Control Register selects among the transmit and receive options offered by the ACIA. Fig. 3 shows the meaning of the bits in the CR register.

CR0 and CR1 together set the divide ratio of the transmit and receive clock signals, common to both transmitter and receiver. Master Reset (CR0 and CR1 high) initialises the ACIA.

CR2 through CR4 set the word length of 7 or 8 bits, the parity, which may be even, odd or absent, and the number of stop bits, one or two.

CR5 and CR6 set the transmitter interrupt option and the state of the RTS output. They allow an interrupt to be generated when a character has been sent. The interrupt condition is cleared by writing data into TDR.

CR7 sets the receiver interrupt option. The receiver can generate an interrupt when a character has been received, when DCD has gone from low to high, or when a character has been received while the previous one has not yet been processed (Overrun). The interrupt condition is cleared by reading RDR.

Status Register

This register shows the status of both transmitter and receiver. It is at the same address as the CR register; the SR register is read-only (see table 2). Bit 0 is Receive Data Register Full (RDRF) and goes high when a received character is available in RDR. RDRF goes low again when the character has been read from RDR. DCD being high causes RDRF to go low.

Bit 1 is Transmit Data Register Empty (TDRE) and goes high when a new character to be sent can be placed in TDR. Doing so also immediately starts the transmission.

Bit 2 reflects the DCD input.

Bit 3 reflects the CTS input. As already mentioned, CTS can block the transmitter.

Bits 4, 5 and 6 reflect error conditions: respectively Framing Error (no valid stop bit found by the receiver), Receiver Overrun (previous character not yet read and the next already received; at least one character has been lost) and Parity Error (incorrect parity received). Reading the RDR register or a Master Reset clears these bits again.

Bit 7 indicates the state of the IRQ output. As is usual with 6800 and 6502 ICs, this bit is used to find the IC that generated the interrupt.

The interrupt bit goes low again when RDR is read or TDR is written, if the receiver or the transmitter respectively was the source of the interrupt.

Program examples

Listing 1 contains a number of practical example routines for using the ACIA for input and output.

The routines are written for the 6502, but can be used for the 6800 and 6809 without many essential changes.

Before the ACIA can be used for input and output, initialisation is required. First a Master Reset of the ACIA is performed with the following instructions:

```
ACIARS LDA `#$03
      ````STA `ACIACR
```

Here ACIACR is the address of the ACIA Control Register; it, like the Status Register, Transmit Data and Receive Data Register, is defined in lines 140 through 170.

Next the ACIA must be told what the character should look like: number of data bits, stop bits and parity bit.

The interrupt option and the state of the RTS output are also set. The divide ratio by which the transmit and receive clock is divided is set now as well. Lines 270 through 390 illustrate this with an example.

The routines INACIA and OUTACI are the simplest routines for doing input and output respectively with an ACIA. The accumulator holds the character received or to be sent. The first action is to test the flag and wait until it is set. Reading or writing a character makes the ACIA ready again for the next action.

The routine INTERM is useful when a so-called full-duplex terminal is used. The

The receive routine INACER makes better use of the error checking performed by the ACIA. When a protocol is used, for example, this routine will serve better than the simple routine INACIA.

```

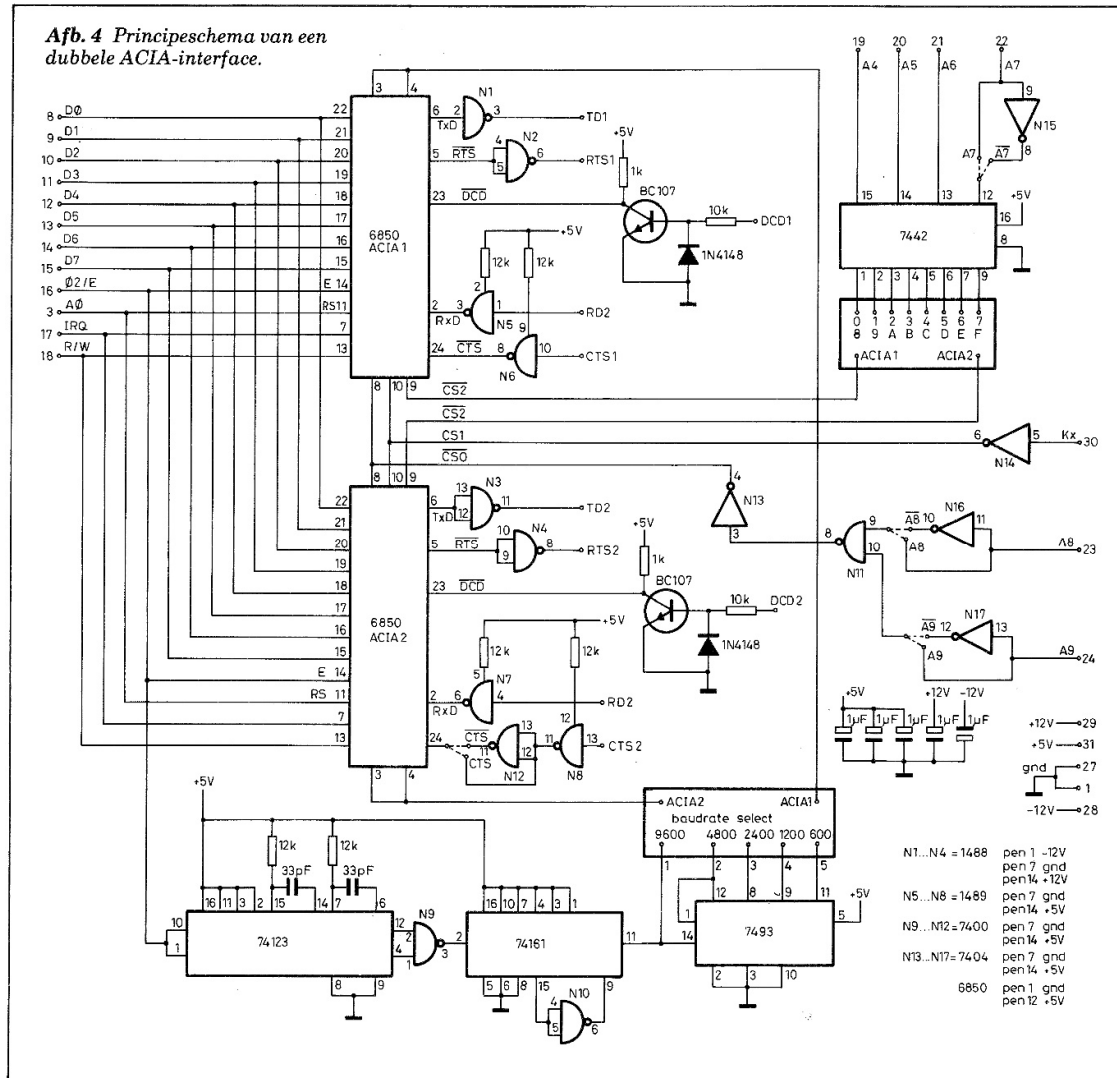
MICO 01 6502 ASSEMBLER V2.02 PAGE 01
MICO 01 6502 ASSEMBLER V2.02 PAGE 02

00100:
00200:
00300:
00400:
00500:
00600:
00700:
00800:
00900:
01000:
01100:
01200:
01300:
01400:
01500:
01600:
01700:
01800:
01900:
02000:
02100:
02200:
02300:
02400:
02500:
02600:
02700:
02800:
02900:
03000:
03100:
03200:
03300:
03400:
03500:
03600:
03700:
03800:
03900:
04000:
04100:
04200:
04300:
04400:
04500:
04600:
04700:
04800:
04900:
05000:
05100:
05200:
05300:
05400:
05500:
05600:
05700:
05800:
05900:
06000:
06100:
06200:
06300:
06400:
06500:
06600:
06700:
06800:
06900:
07000:
07100:
07200:
07300:
07400:
07500:
07600:
07700:
07800:
07900:
08000:
08100:
08200:
08300:
08400:
08500:
08600:
08700:
08800:
08900:
09000:
09100:
09200:
09300:
09400:
09500:
09600:
09700:
09800:
09900:
10000:
10001:
10002:
10003:
10004:
10005:
10006:
10007:
10008:
10009:
10010:
10011:
10012:
10013:
10014:
10015:
10016:
10017:
10018:
10019:
10020:
10021:
10022:
10023:
10024:
10025:
10026:
10027:
10028:
10029:
10030:
10031:
10032:
10033:
10034:
10035:
10036:
10037:
10038:
10039:
10040:
10041:
10042:
10043:
10044:
10045:
10046:
10047:
10048:
10049:
10050:
10051:
10052:
10053:
10054:
10055:
10056:
10057:
10058:
10059:
10060:
10061:
10062:
10063:
10064:
10065:
10066:
10067:
10068:
10069:
10070:
10071:
10072:
10073:
10074:
10075:
10076:
10077:
10078:
10079:
10080:
10081:
10082:
10083:
10084:
10085:
10086:
10087:
10088:
10089:
10090:
10091:
10092:
10093:
10094:
10095:
10096:
10097:
10098:
10099:
10100:
10101:
10102:
10103:
10104:
10105:
10106:
10107:
10108:
10109:
10110:
10111:
10112:
10113:
10114:
10115:
10116:
10117:
10118:
10119:
10120:
10121:
10122:
10123:
10124:
10125:
10126:
10127:
10128:
10129:
10130:
10131:
10132:
10133:
10134:
10135:
10136:
10137:
10138:
10139:
10140:
10141:
10142:
10143:
10144:
10145:
10146:
10147:
10148:
10149:
10150:
10151:
10152:
10153:
10154:
10155:
10156:
10157:
10158:
10159:
10160:
10161:
10162:
10163:
10164:
10165:
10166:
10167:
10168:
10169:
10170:
10171:
10172:
10173:
10174:
10175:
10176:
10177:
10178:
10179:
10180:
10181:
10182:
10183:
10184:
10185:
10186:
10187:
10188:
10189:
10190:
10191:
10192:
10193:
10194:
10195:
10196:
10197:
10198:
10199:
10200:
10201:
10202:
10203:
10204:
10205:
10206:
10207:
10208:
10209:
10210:
10211:
10212:
10213:
10214:
10215:
10216:
10217:
10218:
10219:
10220:
10221:
10222:
10223:
10224:
10225:
10226:
10227:
10228:
10229:
10230:
10231:
10232:
10233:
10234:
10235:
10236:
10237:
10238:
10239:
10240:
10241:
10242:
10243:
10244:
10245:
10246:
10247:
10248:
10249:
10250:
10251:
10252:
10253:
10254:
10255:
10256:
10257:
10258:
10259:
10260:
10261:
10262:
10263:
10264:
10265:
10266:
10267:
10268:
10269:
10270:
10271:
10272:
10273:
10274:
10275:
10276:
10277:
10278:
10279:
10280:
10281:
10282:
10283:
10284:
10285:
10286:
10287:
10288:
10289:
10290:
10291:
10292:
10293:
10294:
10295:
10296:
10297:
10298:
10299:
10300:
10301:
10302:
10303:
10304:
10305:
10306:
10307:
10308:
10309:
10310:
10311:
10312:
10313:
10314:
10315:
10316:
10317:
10318:
10319:
10320:
10321:
10322:
10323:
10324:
10325:
10326:
10327:
10328:
10329:
10330:
10331:
10332:
10333:
10334:
10335:
10336:
10337:
10338:
10339:
10340:
10341:
10342:
10343:
10344:
10345:
10346:
10347:
10348:
10349:
10350:
10351:
10352:
10353:
10354:
10355:
10356:
10357:
10358:
10359:
10360:
10361:
10362:
10363:
10364:
10365:
10366:
10367:
10368:
10369:
10370:
10371:
10372:
10373:
10374:
10375:
10376:
10377:
10378:
10379:
10380:
10381:
10382:
10383:
10384:
10385:
10386:
10387:
10388:
10389:
10390:
10391:
10392:
10393:
10394:
10395:
10396:
10397:
10398:
10399:
10400:
10401:
10402:
10403:
10404:
10405:
10406:
10407:
10408:
10409:
10410:
10411:
10412:
10413:
10414:
10415:
10416:
10417:
10418:
10419:
10420:
10421:
10422:
10423:
10424:
10425:
10426:
10427:
10428:
10429:
10430:
10431:
10432:
10433:
10434:
10435:
10436:
10437:
10438:
10439:
10440:
10441:
10442:
10443:
10444:
10445:
10446:
10447:
10448:
10449:
10450:
10451:
10452:
10453:
10454:
10455:
10456:
10457:
10458:
10459:
10460:
10461:
10462:
10463:
10464:
10465:
10466:
10467:
10468:
10469:
10470:
10471:
10472:
10473:
10474:
10475:
10476:
10477:
10478:
1
```

RS	R/W	Register	Access
0	0	CR (Control Register)	write only
0	1	SR (Status Register)	read only
1	0	TDR (Transmit Data Register)	write only
1	1	RDR (Receive Data Register)	read only

D7	D6	D5	D4	D3	D2	D1	D0
IRQ	parity	OV RN	FE	CTS	DCD	TDRE	RDRF
	error	receiver overrun	framing error	not	not	transmitter data empty	receiver data register full

**Afb. 4** Principeschema van een dubbele ACIA-interface.



## Practical design

Fig. 4 shows a circuit for connecting two ACIAs to a microprocessor system. Around the ACIA, three parts of the circuit needed to use the ACIA can be identified:

- Connection to the microprocessor bus and selection of the ACIA address.
- Interface between the ACIA and the peripheral according to the RS232C standard.
- Generation of the transmit and receive clock signal.

## ACIA address selection

The circuit is designed for use with a 6502 or 6809 microprocessor system. For a 6800 microprocessor, VMA will have to be included in the selection.

Connected directly to the ACIA are: the data bus, address line A0, interrupt line IRQ, read/write line R/W and the system clock  $\phi 2$  or E.

It was decided to give the ACIA a space of 16 addresses by not including A1 to A3 in the selection. The ACIA select lines CS0 to CS2 must now be derived from address lines A4 to A15. The circuit shows only the selection from A4 to A9; the input line Kx is derived

from A10 to A15 and is a common line for a 1K space in which all I/O is placed. Generating this Kx signal has been covered before in Radio Bulletin, notably in the articles "Memory support for 6502 systems."

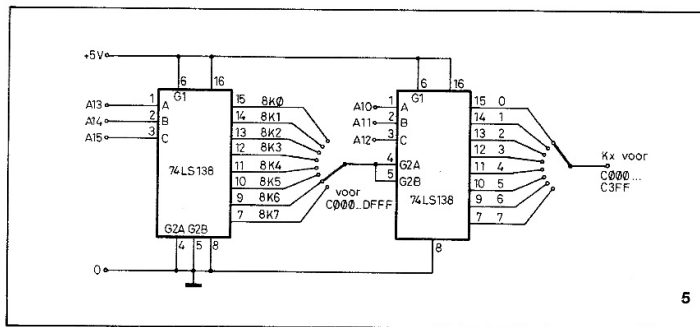


Fig. 5 shows how a Kx signal can be derived from A10 to A15 with two ICs. The position of the ACIA, to an accuracy of 16 addresses, within the 1 Kbyte space determined by Kx, is set by inverting or not inverting A7, A8 and A9 and by the three-to-ten decoder IC

type 7442.

An example will now make clear how a desired address for the ACIAs can be set in this way.

The desired address is \$C000 for the Control and Status Register and \$C001 for the Transmit and Receive Register. A0 and R/W select one of these registers, and the rest of the selection need only select \$C000 as the base address. For ACIA2 this is \$C010. The Kx signal, as generated in fig. 5, selects the range \$C000 to \$C3FF.

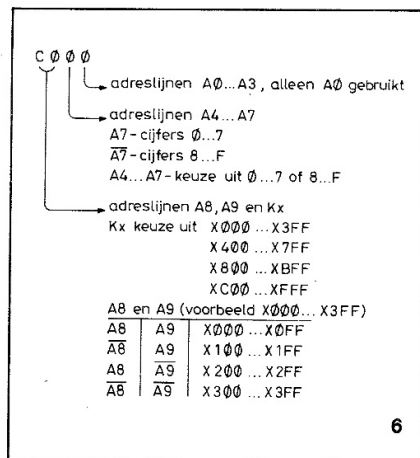


Fig. 6 shows how the selection is split up starting from the address.

A4 to A7 lead to CS2, by making the selection low. A4 to A6 and the 7442 IC lead to digits 0 to 7 or 8 to F. A7 selects the range 0 to 7 or 8 to F.

In the example, therefore, A7 must not be inverted.

ACIA1 is placed at address \$C000 by connecting CS2 to output 0/8; ACIA2 at address \$C010 by connecting CS2 of ACIA2 to output 1/A.

A8 and A9 lead to the select signal CS0, which must go low for selection. To obtain a zero as the third digit of the address, both A8 and A9 must not be

inverted. A8 and A9 lead to CS0 via the NAND function of the 7400 gate.

### Interface according to RS-232C

A widely used standard in asynchronous serial data communication is the RS-232C standard. The standard itself is discussed in more depth below; the circuit includes two ICs and a few other components to meet the electrical requirements of the RS-232C standard. According to the standard, -3 to -15 V is the logical true state (MARK) and +3 to +15 V the logical false state (SPACE). The ICs, types 1488 and 1489, are made

specifically to convert the signals of an ACIA, which are at TTL level, to RS-232C levels. The DCD input is converted in a simple way with a transistor and a few other components.

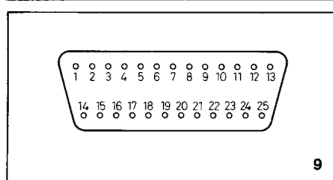
ACIA2 has the option of working with an inverted CTS input signal.

### **Transmit and receive clock**

The transmit and receive clock signal is generated in a way that has been published before in Radio Bulletin. Starting from the 1 MHz system clock, dividers produce the frequencies for the common baud rates: 9600 to 600 baud with the ACIA set to divide by 16, and 300 baud if we use the frequency marked 1200 baud with the ACIA set to divide by 64.

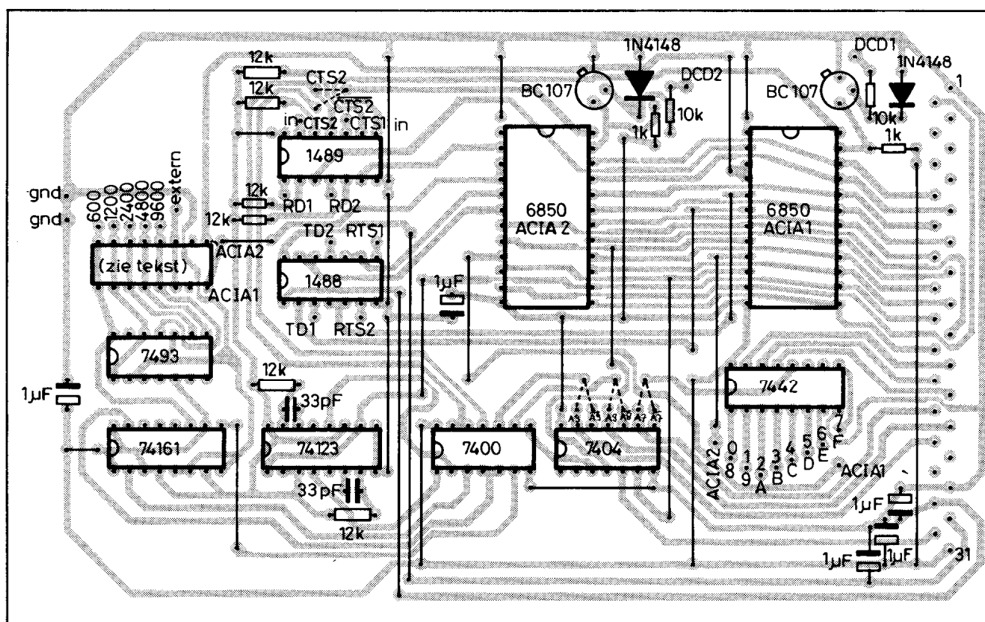
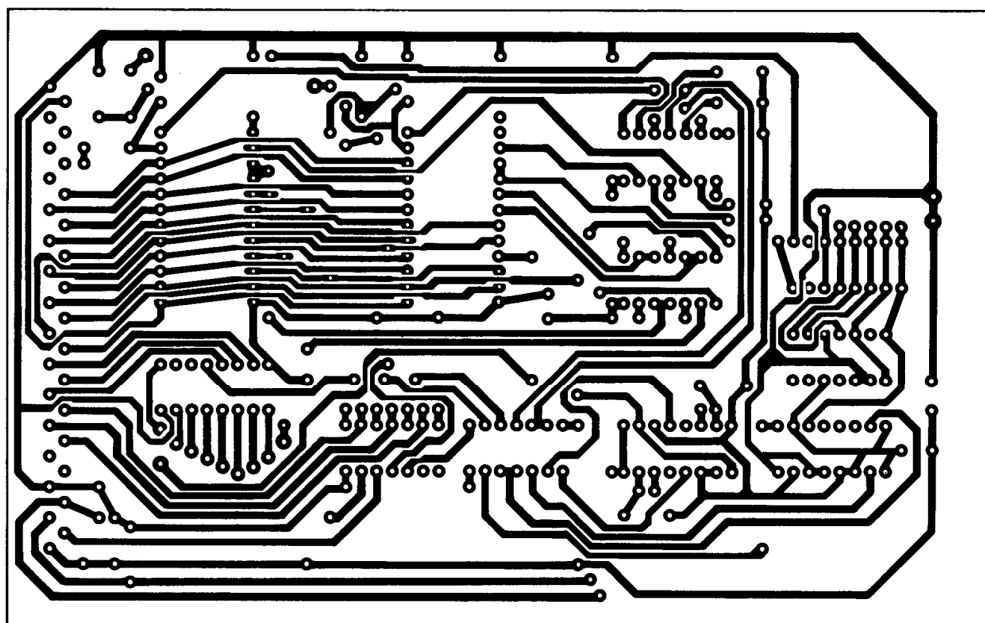
The 74123 IC turns the 1 MHz clock signal into 2 MHz by frequency doubling. The divider IC divides this 2 MHz frequency by 13, giving 153840 Hz. That is the frequency needed for communication at 9600 baud. The 7493 divider IC repeatedly divides this signal by 2.

# ACIA6850



**Afb. 7** Printontwerp, schaal 1 : 1.  
**Afb. 8** Componentenopstelling.  
**Afb. 9** Vooraanzicht van de connector met 25 pennen voor DTE (Data Terminal Equipment).  
**Tabel 3** Betekenis van de busconnector-aansluitingen van de print.

**Tabel 4** Meest geaccepteerde gebruikmaking van de RS-232C-connector.



## **PCB design**

A PCB layout in Eurocard format has been made from the circuit. The layout is shown in fig. 7 and the component layout in fig. 8. The board is single-sided, so a number of wire jumpers are needed.

The address selection also requires a number of wire jumpers to be fitted, for inverting or not inverting A7 to A9 and for choosing the output of the 7442 IC, namely 0/8 to 7/F. For ACIA2 a wire jumper must also be used to choose between an inverted or non-inverted CTS signal. The baud rate requires a wire jumper for each ACIA or, if an IC socket is fitted, a header plug with a wire jumper on it.

The board has a 31-pole connector to DIN41617; the pin assignment is given in table 3.

## **RS-232C standard**

The well-known RS-232C standard was set up by the EIA, the Electrical Industry Association, to bring uniformity to the jumble of connections and specifications of modems. Modems are devices that can transport digital signals over telephone lines by modulation and demodulation.

RS-232C is a standard for the connection between Data Terminal Equipment, usually abbreviated to DTE, and Data Communication Equipment, abbreviated to DCE, which use a serial signal as their interface.

An almost identical standard was developed by the CCITT, a part of the United Nations.

The standard consists of the following parts:

- Electrical characteristics of the signals.
- Mechanical construction of the interface.
- Functional description of the circuits and a list of subsets for special purposes.

The RS-232C standard is excellently described in the book mentioned at the end of this article, and the interested reader is referred to it for many aspects of and practical guidance on data communication.

The ACIA circuit and interface meet the electrical requirements of the RS-232C standard, and besides the input and output lines the ACIA has enough control lines to connect most devices working to the RS-232C standard.

Unfortunately, many devices whose manufacturers claim they are RS-232C-compatible cannot simply be connected to each other. The use of the control lines in particular varies considerably.

For simple cases, a little experimenting and study of (hopefully complete) documentation will usually reveal how things should be connected.

## **RS-232C connector**

A 25-pin connector with the layout of fig. 9 is the most commonly used connector for RS-232C. A female connector is used for DTE equipment and a male connector for DCE equipment. DTE equipment is usually a terminal, teletype etc. DCE equipment is a modem etc. For a DCE peripheral the computer is a DTE, and for a DTE the computer is

a DCE.

Table 4 contains the most widely accepted use of the connector. The other pins can be used for all kinds of purposes, ranging from test points to a 20 mA current-loop interface.

Two examples will hopefully illustrate the use.

#### **Printer H14**

Besides the data transmit and receive lines, the H14 printer has an RTS handshake line which, on the first models, has the reverse of the usual polarity. The ACIA2 interface is suitable for this.

The H14 printer is regarded as a DCE device, and a female connector is connected to the ACIA. The following pins are used:

1 = frame ground.

2 = Tx data sent by the computer.

3 = Rx data sent to the computer.

4 = RTS of the printer, connected inverted to CTS of the ACIA.

7 = signal ground.

20 = DTR of the printer, connected to DCD of the ACIA, indicates printer on.

## Modem

A standard 300 baud modem, being a DCE, requires a male connector on the computer as a DTE device. The following connections were used:

1 = frame ground.

2 = Tx data from the modem to the computer.

3 = Rx data to the modem from the computer.

4 = RTS from the computer to the modem.

5 = CTS from the modem to the computer.

7 = signal ground.

8 = DCD from the modem to the computer.

20 = +12 V via a 1 k $\Omega$  resistor to the modem: DTR (Data Terminal Ready).

DSR (Data Set Ready) from the modem is ignored here.

The two examples above were arrived at by carefully studying the documentation and experimenting. A handy aid for this is a small box, available from several manufacturers, with indicator lamps and easily changed connections, which is placed between the two devices.

## Literature

A handbook on data communication offering a wealth of information is the English-language book "Technical Aspects of Data Communication" by John E. McNamara, published by Digital Press, Digital Equipment Corporation.

Table 4 - Most widely accepted use of the RS-232C connector.

Pin	Meaning	Source
1	protective ground	n/a
2	transmitted data	DCE
3	received data	DTE
4	RTS (Request To Send)	DCE
5	CTS (Clear To Send)	DTE
6	DSR (Data Set Ready)	DCE
7	signal ground	n/a
8	Data Carrier Detect (DCD)	DCE
20	Data Terminal Ready	DTE

**Table 3 - Pin assignment of the bus connector of the board.**

<b>Pin</b>	<b>Meaning</b>
1	ground
2	NMI
3	A0
4	A1
5	A2
6	A3
7	RESET
8	D0
9	D1
10	D2
11	D3
12	D4
13	D5
14	D6
15	D7
16	$\phi 2/E$
17	IRQ
18	R/W
19	A4
20	A5
21	A6
22	A7
23	A8
24	A9
25	A10
26	- 5 V
27	analog ground
28	- 12 V
29	+12 V
30	Kx
31	+5 V