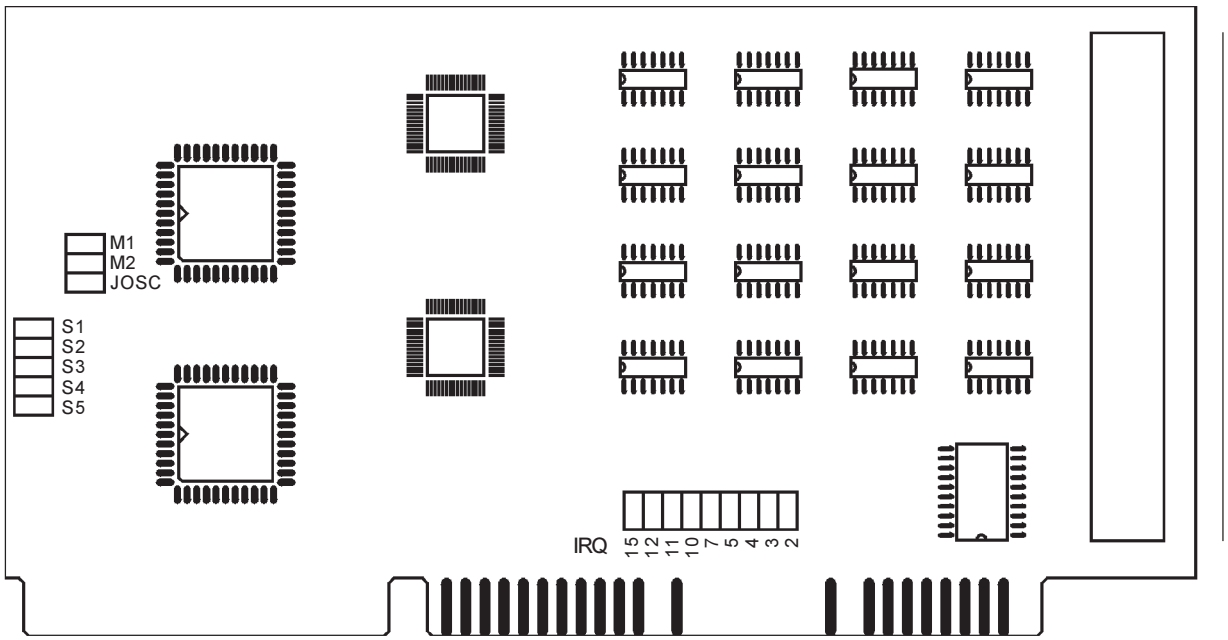
Async Adapter

Features

- 8 full duplex RS-232 channels
- Async protocol
- Full modem control
- Data rate up to 460.8 kbaud
- Deep FIFO 64 bytes
- Hardware compatible with ARNET and Digiboard PC/8 adapters
- Register compatible with 16750
- Octopus cable, compatible with Digiboard
- Up to 9 adapters per computer
- ISA bus
- Supported by virtually all operating systems

Contents



Description

The Cronyx-Omega adapter is the outstanding multiport serial communication board for PC ISA compatible systems. It provides 8 asynchronous RS-232 channels.

The Omega adapter has two jumper selectable hardware configuration modes, compatible with the most popular multiport architectures: ARNET and Digiboard. Due to it's full hardware compatibility with the traditional solutions, Omega is supported by all major operating systems:

- Windows NT, Windows 95
- MS-DOS, DesqView/X
- OS/2, QNX
- SCO Unix, SCO Xenix
- Unixware, Unix SVR4
- FreeBSD, OpenBSD, NetBSD, BSD/OS
- Linux

The adapter is based upon the state-of-the-art communication controller 16C654, that makes it unbeatable:

- High reliability
- Deep FIFO (64 bytes)
- High data rate up to 460.8 kbaud
- Small dimensions
- Low power

Jumpers

Jumper	*Description*
S1–S5	Base I/O Address
IRQ2–IRQ15	IRQ Number
JOSC	Turbo Mode (quadruple data rate)
M1–M2	Compatibility Mode ARNET/Digiboard

Base Address

The adapter occupies the range of 40h addresses in the I/O space of the ISA bus. When installing two and more adapters, the I/O space could be decreased by using "higher" addresses. It is recommended to install adapters by pairs, with the offset 400h, for example 180h and 580h, or 300h and 700h.

Jumpers S *S5-4-3-2-1*	*I/O* *Addresses*	*Interrupt* *Register*
: : ▮ : :	100h–13Fh	1100h
: : ▮ : ▮	140h–17Fh	1140h
: : ▮▮ :	180h–1BFh	1180h
: : ▮▮▮	1C0h–1FFh	11C0h
: ▮ : : :	200h–23Fh	1200h
: ▮ : : ▮	240h–27Fh	1240h
: ▮ : ▮ :	280h–2BFh	1280h
: ▮ : ▮▮	2C0h–2FFh	12C0h
: ▮▮ : :	300h–33Fh	1300h
: ▮▮ : ▮	340h–37Fh	1340h
: ▮▮▮ :	380h–3BFh	1380h
▮ : ▮ : :	500h–53Fh	1500h
▮ : ▮ : ▮	540h–57Fh	1540h
▮ : ▮▮ :	580h–5BFh	1580h
▮ : ▮▮▮	5C0h–5FFh	15C0h
▮▮ : : :	600h–63Fh	1600h
▮▮ : : ▮	640h–67Fh	1640h
▮▮ : ▮ :	680h–6BFh	1680h
▮▮ : ▮▮	6C0h–6FFh	16C0h
▮▮▮ : :	700h–73Fh	1700h
▮▮▮ : ▮	740h–77Fh	1740h
▮▮▮▮ :	780h–7BFh	1780h

Interrupt Line

Jumpers IRQ *15-12-11-10-7-5-4-3-2*	*IRQ* *Number*
: : : : : : : : ▮	IRQ 2/9
: : : : : : : ▮ :	IRQ 3
: : : : : : ▮ : :	IRQ 4
: : : : : ▮ : : :	IRQ 5
: : : : ▮ : : : :	IRQ 7
: : : ▮ : : : : :	IRQ 10

: :▉: : : : : :	IRQ 11
:▉: : : : : : :	IRQ 12
▉: : : : : : : :	IRQ 15

Turbo Mode

Jumper JOSC	Data Rate
▉	300–115200 kbaud
:	1200–460800 kbaud

When the jumper JOSC is installed, the adapter implements the standard data rates in range from 300 to 115200 baud. Removing the jumper JOSC increases the data rates by 4 times relative to the standard ones. It makes it possible to use data rates 460.8, 230.4, 153.6 and 76.8 baud. No change in the software is required for using higher data rates. For example, for setting the rate 153600 baud, remove the jumper JOSC and initialize the driver for 38400 baud (38400 = 153600/4).

Compatibility Mode

The adapter is functioning in one of three compatibility modes: ARNET, ARNET with inversion, and Digiboard. The compatibility mode determines the format of an interrupt register.

Jumpers M M2-M1	Compatibility Mode
:▉	ARNET
: :	ARNET with inversion
▉:	Digiboard PC/8

Interrupt Register

The interrupt register contains the information about channels, requesting for the service. The interrupt register is placed in the I/O address space with the offset 1000h relative to the base adapter address. For example, if the adapter is installed at address 380h, then the interrupt register is at address 1380h.

The format of the interrupt register depends on the compatibility mode, set by jumpers M1-M2.

In ARNET mode every bit of the interrupt register corresponds to one channel. The least significant bit is related to channel 0, the most significant bit — to channel 7. When the channel is requesting for an interrupt, then the corresponding bit of the interrupt register is set to 1.

In ARNET with inversion mode the values of the bits of the interrupt register is changed to the opposite: if the channel is requesting for the service, then the bit is set to 0, otherwise 1.

In Digiboard mode the three least significant bits of the interrupt register contain the channel number, requesting for interrupt (0..7). The most significant bits contain zeroes. If no channel is requesting for interrupt, then the register contains the value FFh.

After the hardware reset, the interrupt requests from all channels are disabled. To enable the interrupt from the channel, the bit D3 (OUT2) of the register MCR must be set to 1 (see 16654 data sheet).

Cable

The octopus cable of the Cronyx-Omega adapter is hardware compatible with the Digiboard PC/8 cables.

Pin DB-25	RS-232 Signal	Pins HDB-78 for channels 1..8
1	GND	ground (shield)
2	TXD	30,50,11,10,40,2,63,64
3	RXD	55,17,37,56,28,8,46,27
4	RTS	51,31,12,14,21,41,62,60
5	CTS	16,53,59,57,25,4,9,45
6	DSR	54,34,58,38,5,42,29,26

7	GND	68,69,70,71,72,73,74,75
8	DCD	35,33,39,18,43,23,48,6
20	DTR	49,32,13,52,22,3,61,1
22	RI	36,15,20,19,44,24,47,7

Warning: some cables could have the channels numbered starting from 0.

Software Installation

The adapter's delivery pack contains the disk with the software, that includes the diagnose utility and the recommended drivers for a number of operating systems. There is also the utility `omhelp.exe` for finding the free I/O base address. In this chapter you will find some guides and examples on testing the adapter, and installing the drivers.

Testing

To test the adapter, run the utility `diag.exe`. When called, it will search for all installed adapters. Check that I/O addresses and interrupt numbers, detected by the diag utility, are equal to the installed ones. From the menu "Test" run the "General Test". All channels of all installed adapters will be tested in the internal loopback mode.

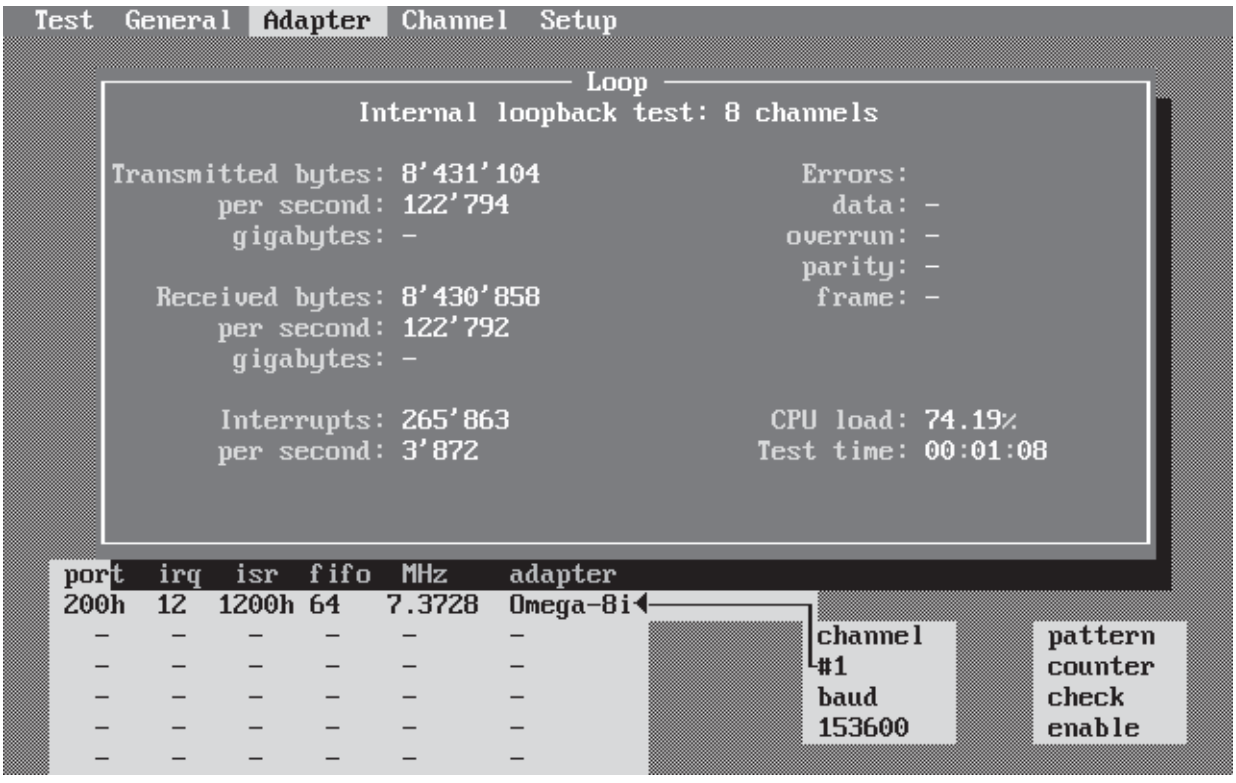
The required data rate and the test data pattern could be selected from the menu "Parameters" .

The menu "Channels" is used to test the individual channels in several modes: transmit only, receive only, internal loopback, external loopback.

An example of running the test at the data rate 230 kbaud is shown on page 4.

Setup under MS-DOS

Under MS-DOS, the FOSSIL drivers are commonly used for working with asynchronous data channels. The most popular FOSSIL drivers are BNU and X00. For example, to install the X00 driver for the Omega adapter on the I/O address 100h



and IRQ 5 (channels 1-6), add the following line to the start file autoexec.bat:

```
x00 e 2=100,irq5 3=108,irq5 4=110,irq5 \
      5=118,irq5 6=120,irq5 7=128,irq5
```

The utility `tty.exe` can be used to test the operability of the FOSSIL driver.

Setup under Windows 95

In the Windows 95 operating system the Cronyx-Omega adapter is supported by the standard driver of COM-ports. Open the Control Panel and run the utility "Add New Hardware". On the question "Do you want Windows to search for your new hardware?" press "No". Choose the adapter type "Ports (COM & LPT)". When selecting the manufacturer and device model choose "Standard port types/Communications Port". On the request of the I/O range and IRQ number, press the "Next" button. After this, the driver of the next available serial port will be installed (COM2, COM3 etc.). Repeat these actions several times, according to the number of serial channels. Note, that ports COM1-COM4 cannot be used for the Cronyx-Omega adapter, because they have the fixed I/O addresses and IRQ numbers, but they must be added to the system to get ports COM5, COM6 and so on.

Then run the utility "System" from the Control Panel, and select the "Device Manager" tab. In the device list, open the "Ports (COM & LPT)" subtree. Select the port COM5 and press the "Properties" button. On the "Resources" page set up the I/O address and the IRQ number of the Omega adapter. Repeat these actions for ports COM6 and others. The I/O addresses of channels should be set with the step 8, for example COM5–100h, COM6–108h, COM7–110h etc. All channels have the same IRQ number.

After setting the I/O addresses and IRQ numbers, the ports are ready for use.

Setup under Windows NT

Under Windows NT the Omega adapter is supported by the standard serial driver in Digiboard mode. Before installing the adapter into a computer, select the Digiboard compatibility mode by jumpers M1–M2.

From the Control Panel run the utility "Ports". Press button "Add..." to add the new port, set the I/O address and IRQ number, according to the adapter settings. Repeat this operation several times, for every port installed.

Then call the resource editor (menu "Start/Run...", then enter "regedit"). Select the registry key HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/Serial/Parameters. There are several subkeys, for every serial port installed: Serial3, Serial4 etc. Every subkey contains the base port I/O address "PortAddress", IRQ number "Interrupt" and other parameters. For each subkey SerialN add three new values of type REG_DWORD: the address of interrupt register (InterruptStatus), compatibility mode Digiboard (Indexed) and relative number of the port in the adapter (PortIndex). For example, for the adapter on the base I/O address 100h (interrupt register at 1100h):

```
Serial3:
    InterruptStatus = 0x1100
    Indexed = 1
    PortIndex = 1
Serial4:
    InterruptStatus = 0x1100
    Indexed = 1
    PortIndex = 2
Serial5:
    InterruptStatus = 0x1100
    Indexed = 1
    PortIndex = 3
Serial6:
    InterruptStatus = 0x1100
    Indexed = 1
    PortIndex = 4
Serial7:
    InterruptStatus = 0x1100
```

```
        Indexed = 1
        PortIndex = 5
    Serial8:
        InterruptStatus = 0x1100
        Indexed = 1
        PortIndex = 6
    Serial9:
        InterruptStatus = 0x1100
        Indexed = 1
        PortIndex = 7
    Serial10:
        InterruptStatus = 0x1100
        Indexed = 1
        PortIndex = 8
```

After the reboot the adapter is ready for use.

Setup under Linux

To set up the parameters of the standard serial driver of the Linux operating system for working with the Cronyx-Omega adapter, add the following lines to the system start-up file /etc/rc.d/rc.serial (an example for the adapter at the I/O address 100h and IRQ number 5):

```
setserial -b cua2 port 0x100 irq 5 uart 16550A ^fourport
setserial -b cua3 port 0x108 irq 5 uart 16550A ^fourport
setserial -b cua4 port 0x110 irq 5 uart 16550A ^fourport
setserial -b cua5 port 0x118 irq 5 uart 16550A ^fourport
setserial -b cua6 port 0x120 irq 5 uart 16550A ^fourport
setserial -b cua7 port 0x128 irq 5 uart 16550A ^fourport
setserial -b cua8 port 0x130 irq 5 uart 16550A ^fourport
setserial -b cua9 port 0x138 irq 5 uart 16550A ^fourport
```

After the reboot the adapter is ready for use.

Setup under FreeBSD

For installing the adapter Cronyx-Omega in the FreeBSD operating system, the following lines should be added to the kernel configuration file (/sys/i386/conf/GENERIC). An example is for the adapter on the I/O address 100h and IRQ 5:

```
options COM_MULTIPORT
device sio2 at isa? port 0x100 tty flags 0x20205 \
    irq 5 vector siointr
device sio3 at isa? port 0x108 tty flags 0x20205
device sio4 at isa? port 0x110 tty flags 0x20205
device sio5 at isa? port 0x118 tty flags 0x20205
device sio6 at isa? port 0x120 tty flags 0x20205
device sio7 at isa? port 0x128 tty flags 0x20205
device sio8 at isa? port 0x130 tty flags 0x20205
device sio9 at isa? port 0x138 tty flags 0x20205
```

Then recompile and install the new kernel:

```
config GENERIC
cd /sys/compile/GENERIC
make depend all install
```

The device files for installed serial channels must be created:

```
cd /dev
sh MAKEDEV ttyd2 cuaa2 ttyd3 cuaa3
sh MAKEDEV ttyd4 cuaa4 ttyd5 cuaa5
sh MAKEDEV ttyd6 cuaa6 ttyd7 cuaa7
sh MAKEDEV ttyd8 cuaa8 ttyd9 cuaa9
```

After the reboot the adapter is ready for use.

Please, refer to manual page sio(4) and “Administration Handbook” for any additional information.

Setup under SCO Unix

In the SCO Xenix/Unix, Unixware, Interactive Unix and AT&T Unix SVR4 operating systems it is recommended to use the SAS (Streams Asynchronous Solution) and FAS (Fast Asynchronous Solution) drivers, available on the floppy disk, supplied with the Cronyx-Omega adapter. The driver’s distribution packages contain all needed instructions for installation under these operating systems.

Setup under OS/2

For OS/2 operating system it is recommended to use the SIO driver, available on the floppy disk, supplied with the Cronyx-Omega adapter. All the needed instructions are inside the driver’s package.

Programming Examples

The programming examples shown below take into account the extended capabilities of the 16654 serial controller. There are the samples of an initialization, servicing the interrupts in the ARNET and Digiboard modes, and servicing the interrupts without using the interrupt register. It is assumed that the adapter has the base I/O address 300h and the jumper JOSC is installed. The following declarations are used:

```
#define BASE      0x300
#define DATA      0+BASE
#define IER       1+BASE
#define IIR       2+BASE
#define FCR       2+BASE
#define LCR       3+BASE
#define MCR       4+BASE
#define LSR       5+BASE
#define MSR       6+BASE
#define SCR       7+BASE
#define DLL       0+BASE
#define DLH       1+BASE
#define EFR       2+BASE
#define INTREG    BASE+0x1000
```

Initialization

```
#define FQ        1843200
#define BD        9600
#define DIVISOR   ((FQ/8+BD)/BD/2)

for (chan=0; chan<8; ++chan) {
    /* Set 16654 mode. */
    outb (chan*8+LCR, 0xbf);
    outb (chan*8+EFR, 0x10);

    /* Baud rate. */
    outb (chan*8+LCR, 0x80);
    outb (chan*8+DLH, DIVISOR >> 8);
    outb (chan*8+DLL, DIVISOR);
    outb (chan*8+LCR, 0x03);

    /* Mode 8-N-1, FIFO level 32 bytes
     * on transmit, 56 bytes on receive */
    outb (chan*8+IER, 0);
    outb (chan*8+FCR, 0xa7);
    outb (chan*8+MCR, 0);
    outb (chan*8+SCR, 0);

    /* Enable interrupts. */
    outb (chan*8+MCR, 0x08);
}
```

Channel Interrupt Handler

```
intr_chan (chan, iir)
{
    switch (iir & 0x0f) {
    case 0x04:      /* receive */
    case 0x0c:      /* timeout */
    case 0x06:      /* receive error */
        lsr = inb (chan*8+LSR);
        if (lsr & 0x0e)
            error ();
        while (lsr & 0x01) {
            inb (chan*8+DATA);
            lsr = inb (chan*8+LSR);
            if (lsr & 0x0e)
                    error ();
        }
        if (! (lsr & 0x20))
            break;
    case 0x02:      /* transmit */
        while (inb (chan*8+LSR) & 0x20)
            outb (chan*8+DATA, data);
        break;
    case 0x00:      /* modem signal */
        inb (chan*8+MSR);
        break;
    }
}
```

Servicing Interrupts In ARNET Mode

```
intr_arnet ()
{
    int isr, chan;
    while ((isr=inb(INTREG)) != 0) {
        for (chan=0; chan<8; ++chan)
            if ((isr >> chan) & 1)
                    intr_chan (chan,
                        inb (chan*8+IIR));
}
```

Servicing Interrupts In ARNET Inverted Mode

```
intr_arnet_inverted ()
{
    int isr, chan;
    while ((isr=inb(INTREG))!=0xff) {
        for (chan=0; chan<8; ++chan)
            if (! ((isr >> chan) & 1))
                    intr_chan (chan,
                        inb (chan*8+IIR));
}
```

Servicing Interrupts In Digiboard Mode

```
intr_digi ()
{
    int isr, ch;
    while ((isr=inb(INTREG))!=0xff) {
        ch = isr & 7;
        intr_chan (ch, inb (ch*8+IIR));
    }
}
```

Servicing Interrupts Without The Interrupt Register

```
intr_poll ()
{
    int idle, chan;
    do {
        idle = 1;
        for (chan=0; chan<8; ++chan) {
            iir = inb (chan*8+IIR);
            if (! (iir & 1)) {
                    intr_chan (chan, iir);
                    idle = 0;
            }
        }
    } while (! idle);
}
```