

MIDI on the RS232 Port

music on the serial port!

Design by B. Bouchez

Early 1995, Yamaha introduced a new range of synthesizers with interesting interface capabilities. For the first time, these machines were compatible with three standards: MIDI, Macintosh and PC. To support the PC interface, the MU5, MU10, MU80 synthesizers and the SU10 sampler came with an RS232 serial port, providing connectivity with the PC. This spurred the author into designing an RS232-to-MIDI converter that's supported by a freeware driver from Yamaha.



Initially, these sound generators were designed to work with what Yamaha called 'HOST PORT' supporting software. In fact, only a few sequencers were actually capable of directly supporting this functionality. Fortunately, Yamaha designed a driver that permits the use of a serial port as a standard MIDI interface to almost any sequencer running under Windows.

An interesting spec of the Host Port mode of the MU5 is its ability to retransmit all messages arriving on the MIDI IN connector to the RS232 port. The same ability was available in the other direction, from RS232 to

MIDI OUT. In industry-speak, this is called a 'gateway'.

Actually, there are two problems for using an MU or SU sound unit in such a way. First, it is not possible to disconnect the sound generator from the serial port controller. The sound generator always responds to the incoming messages, giving the odd unexpected result. Second, apart the SU10, none of the other synthesizers are manufactured any more.

The interface described in this article allows you to emulate an MU or SU sound unit, as a transparent gateway, using the standard serial port on your PC. If you are using a

desktop PC, with a MIDI interface already available, like a MPU or Soundblaster, you can add a new MIDI port for all your applications, particularly under Windows. If you are using a portable computer with no MIDI capabilities, you can access the world of MIDI simply by connecting the interface to a free serial port.

System description

The interface schematic is given in **Figure 1**.

Thanks to a microcontroller, the interface schematic is really simple. Note that a microcontroller is mandatory, since the system needs to store data in a FIFO, due to the different baudrates used on the serial port and the MIDI port.

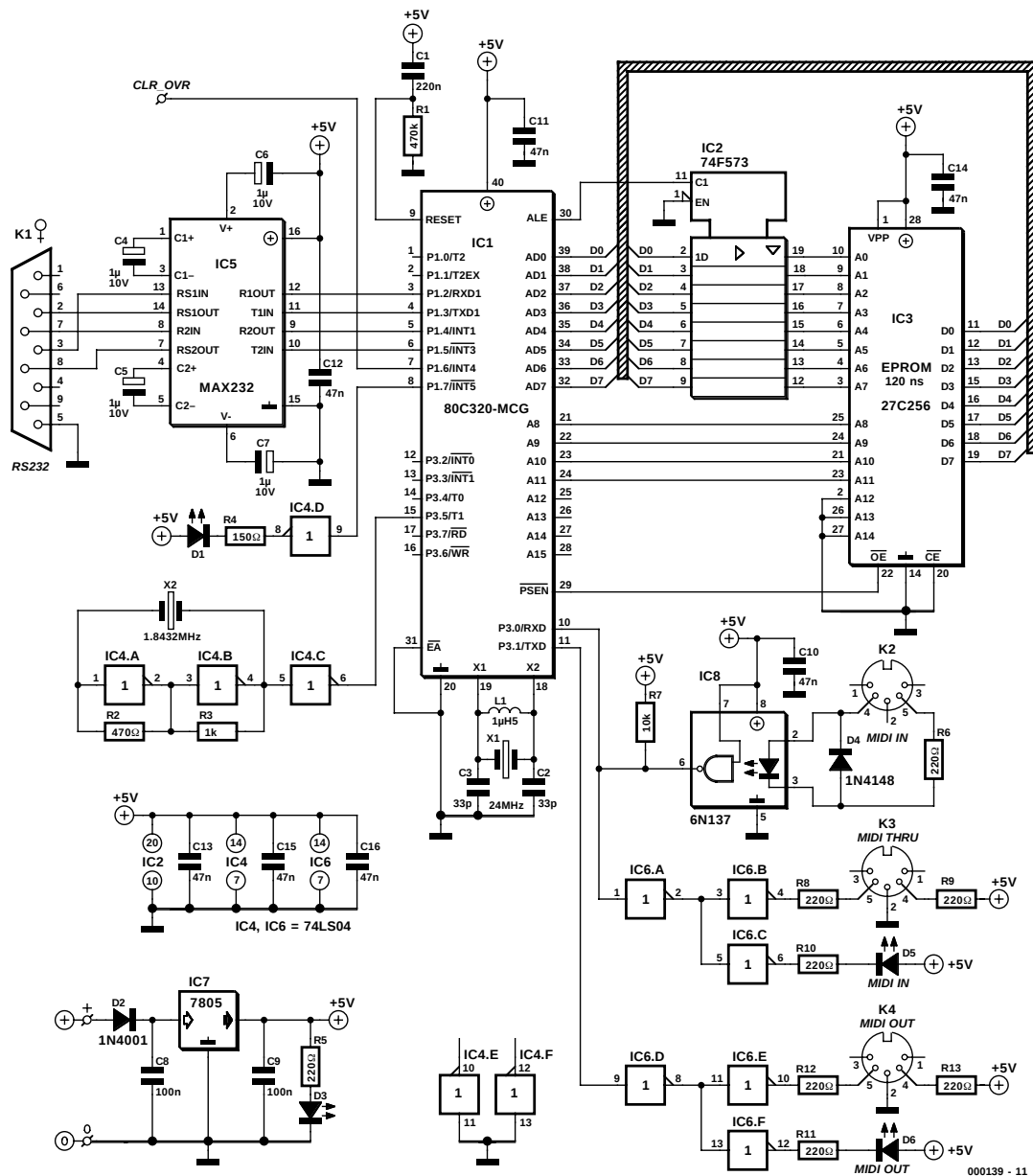


Figure 1. Circuit diagram of the RS232-to-MIDI Converter. At the heart of the circuit you find an 80C320 microcontroller from Dallas Semiconductor.

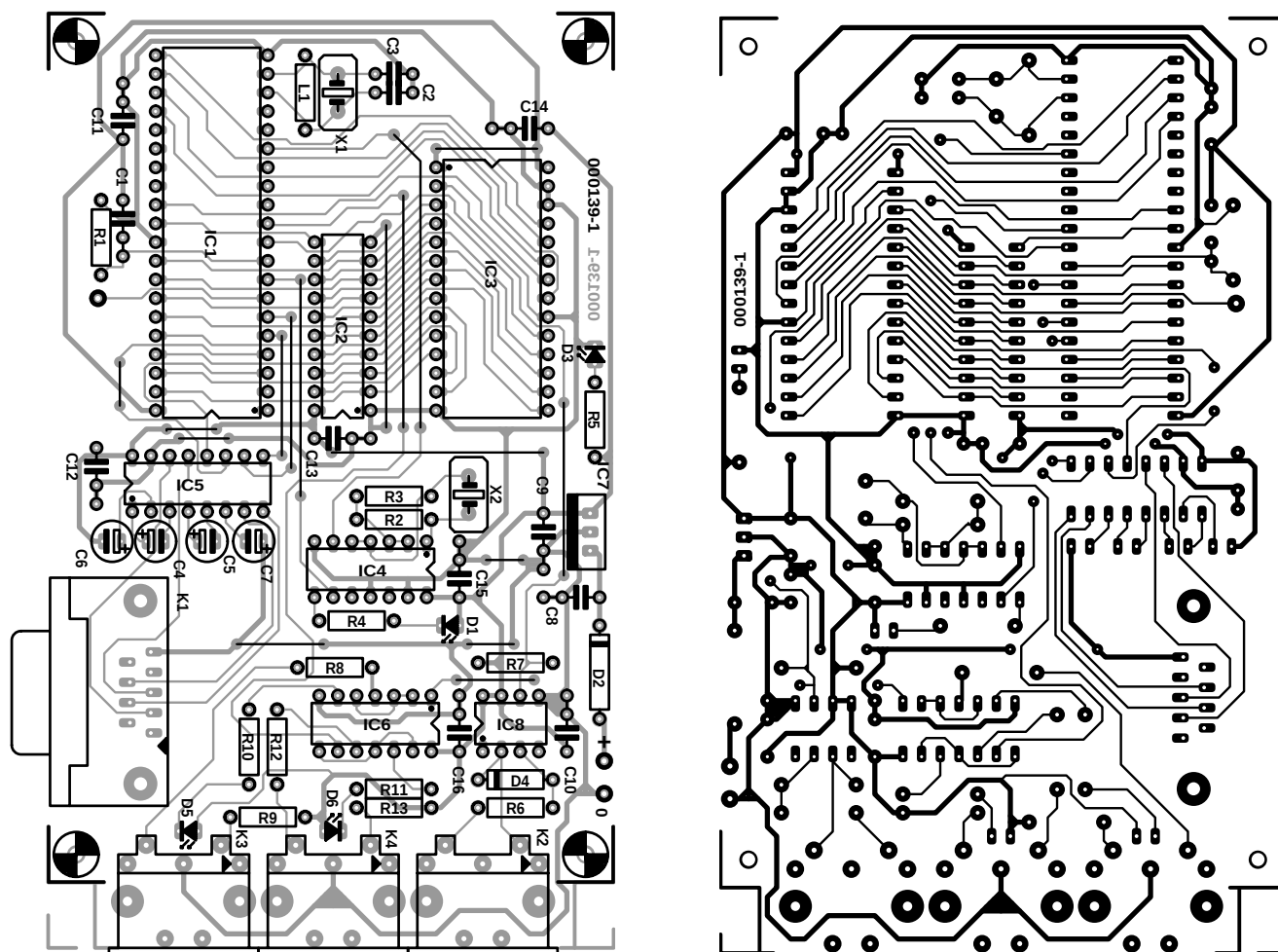


Figure 2. Copper track layout and component mounting plan of the PCB designed for the converter.

The processor is a Dallas type 80C320 microcontroller, which is slightly unusual in that it has two serial ports capable of running at different speeds. Besides, this family of microcontrollers employs a redesigned processor core, running code four times faster than a standard 8032.

Unfortunately, the standard 8032 processor is not compatible with the present application, because it includes only one serial port. The use of a Dallas 80C320 is therefore mandatory.

Here, the C320 does not employ external code memory since the on-chip RAM is sufficient for this application. As always, the address bus is demultiplexed externally by a 573 latch driven by the the processor's ALE signal. You will notice the use of a 27C256 EPROM, with some address lines tied to ground. The application program could easily be stored in a 27C64 EPROM, but these tend to be more expensive then higher capacity models, and they are not easily available in the 120ns speed range, which is mandatory in this case. Please also note that the 573

latch must be an F (fast) TTL device, for the same reasons of speed.

The main system clock is given by a 24 MHz crystal, associated with a small inductor to force it into overtone mode. The MIDI baudrate is derived from the system clock. Since it is not possible to derive the 38,400 baudrate from a 24 MHz clock, an external 1.8432MHz oscillator is used. This oscillator is built around three gates inside IC4. The output of the oscillator drives the T1 input of the processor.

Connected to processor port 1, the MIDI interface is built around IC6 in a classic way. The only difference between the standard MIDI schematic as given in the MIDI specifications is the use of two LEDs as communication indicators. Believe us, these LEDs are really useful in practice, nicely indicating whether a MIDI stream is present or not.

The RS232 interface is connected

to the serial port 0 of the processor, via the ubiquitous MAX232 level converter. Thanks to this chip, it is possible to use a single +5 V power supply. The DB9 connector allows the use of any standard serial cable for the link to the PC. Note that the RTS/CTS handshaking signals are used by the interface. A null-modem cable, with only wire links in the connectors, is **not** compatible with this application. It is necessary to use a fully wired-up cable.

The power supply is made around a standard 5 V regulator, powered externally by a small 12 V DC adaptor. Because of the wide range of connectors available on these power supplies, we decided to solder pins only for the supply connections, rather than a complete PCB mount socket which may be impossible to get locally.

The last signal controlled by the processor is just a buffer overflow

COMPONENTS LIST

Resistors:

R1 = 470k Ω
 R2 = 470 Ω
 R3 = 1k Ω
 R4 = 150 Ω
 R5,R6,R8-R13 = 220 Ω
 R7 = 10k Ω

Capacitors:

C1 = 220nF
 C2,C3 = 33pF
 C4-C7 = 10 μ F 10V radial
 C8,C9 = 100nF
 C10-C16 = 47nF

Semiconductors:

D1,D3,D5,D6 = LED
 D2 = 1N4001
 D4 = 1N4148
 IC1 = 80C320-MCG
 (Dallas Semiconductor)

IC2 = 74F573
 IC3 = EPROM 27C256 100ns
 (programmed, order code **000139-21**)

IC4,IC6 = 74LS04
 IC5 = MAX232 (Maxim Integrated)
 IC7 = 7805
 IC8 = 6N137 (Atmel-Temic)

Inductors:

L1 = 1 μ H5

Miscellaneous:

X1 = 24MHz quartz crystal (third overtone)
 X2 = 1.843 2MHz quartz crystal
 K1 = 9-way sub-D socket (female), PCB mount
 K2,K3,K4 = 5-way DIN socket, pins at 180 degrees, PCB mount
 PCB, order code **000139-1**
 Disk, order code **000139-11** (driver, source code, hex file)

indicator, using the LED D4. If you want to reset this LED by an external command, you may connect a single pushbutton between P1.6 and ground. Note that receiving a SYSTEM RESET MIDI command (0xFF) on the RS232 line also causes the LED to go out.

Building the interface

Building the interface is quite easy, since there is only a handful components to install on the PCB. As usual, we recommend the use of sockets for the ICs, especially for the MIDI current loop inverters and the V24 adapter, since they are directly connected to the external world and may subject to electrostatic discharges when connecting or disconnecting cables.

The only difficulty that could possibly arise when you will build the interface may be encountered around the MIDI connectors. We chose PCB-mounted 5-pin DIN sockets, with a standard shape and PCB footprint. Some models however have slight differences in their housing and pin arrangement. If necessary, 'minor surgery' may be required to install them correctly on the board.

Having installed all components you may power up the board. The only LED that should light is POWER

ON. The two other LEDs must remain off. You can verify with an oscilloscope that you have a 1.8432MHz square wave on the T1 input of the processor. A test feature for the internal chip oscillator is provided. On the T2 pin of the processor, you should find a 250 kHz square wave.

If you find the 250 kHz signal, you can be confident that the program is running correctly, and the crystal oscillator runs at 24 MHz. If not, the first thing to check is the value of the

inductor near the crystal and check the value. If replacing the inductor does not remedy the fault, you will have to check your board for soldering errors or incorrectly fitted parts.

The interface does not need a lot of current to work. As we said before a small power adaptor (12V/200mA) is sufficient. Due to this, IC7 can manage without a heatsink, since it dissipates only little energy. Unfortunately, some inferior quality adaptors deliver much higher voltages than desired or selected (some even up to 18 V when set to 12 V), resulting in more heat to dissipate by IC7. If you note that the regulator becomes too hot, attach a small aluminium bracket to it (2 to 3 cm² is sufficient) to help it to cool. Your new MIDI interface is now ready for use.

Using and testing the interface

Since the interface is in fact nothing but a gateway, no particular processing is required to send and receive MIDI data. If you are using or creating software that can use the 'Host Port' capability (in the Yamaha jargon), the MIDI datastream just employs the serial port rather than a dedicated MIDI port since there is no information to add or remove. Data are exchanged with the serial interface using the 38,400 bits/s, 8, N, 1 format.

If you want to use Windows (95 or 98) software that does not include the 'Host Port' support, you have to get the Yamaha driver that often goes by the name CBXT3. This driver is available from the download pages at <http://www.yamaha.co.uk>

This driver is freeware — the only limitation is that it is supplied 'as is', i.e., with no war-

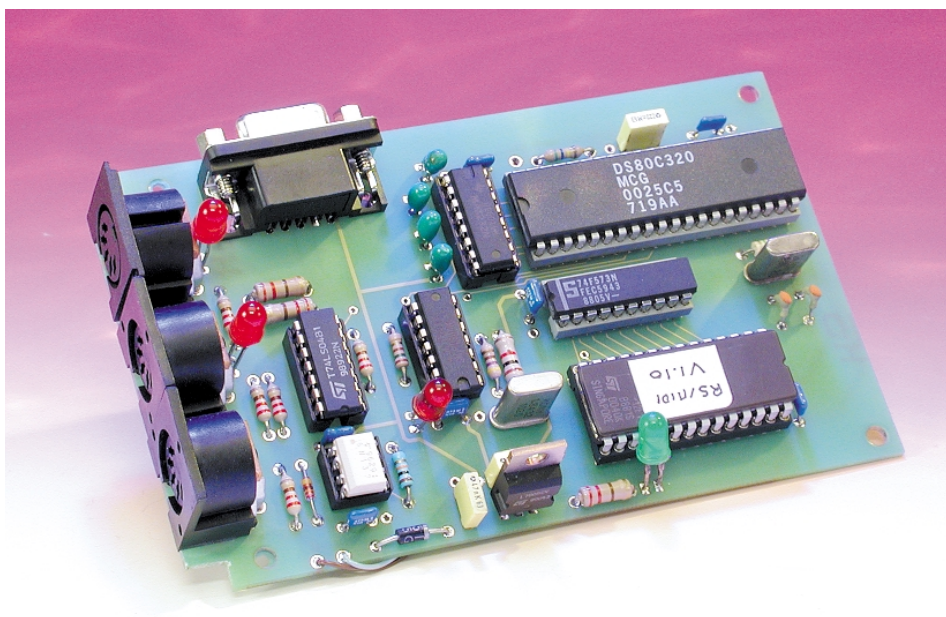


Figure 3. Finished prototype of the RS232-to-MIDI Converter.

ranty or assistance from Yamaha. The driver files are available on a disk with order number **000139-11** through the Publishers' readers Services, and as a ZIP compressed archive from the Free Downloads page, section April 2001, of the *Elektor Electronics* website at www.elektor-electronics.co.uk. The archive includes the source code, hex file and the Yamaha driver, version 1.85.

Under Win95 and Win98 the author tested the CBXT3 driver Version 1.80b2 with full satisfaction. Version 1.85, copied from disk or extracted using Winzip, may be used also. The files will be automatically decompressed into a subdirectory called MIDIDRV.

Launch the Setup program and follow the instructions on the screen. When the software displays a configuration window, choose the serial port you want to use for the converter. When in doubt, don't panic as you will be able to return to this window at all times. Do not check the 'Multiport' box because the present converter does not support this functionality. As a matter of course, you should not check the 'Driver Disable' box because then the driver will not work. This box, by the way, simply serves to free up serial port 'resources' which may be necessary for other applications in heavily loaded PC systems.

After the last window, the software will ask you if you want to reboot the system. Do so to enable the new driver.

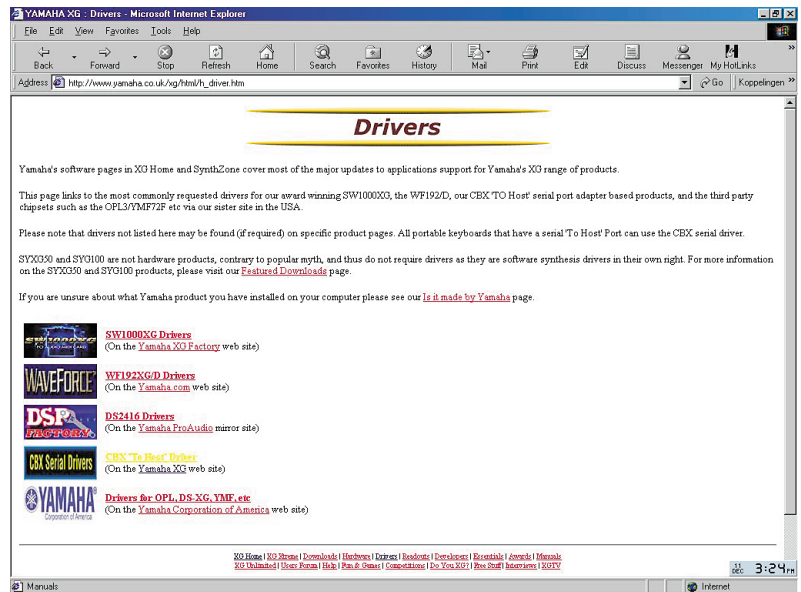
Once the PC is up and running again, you should be able to find a new icon called Yamaha CBX Driver in Control Panels. Clicking on this icon will take you to the configuration window mentioned above.

On some PCs, for reasons unknown to the author, the Yamaha driver will be difficult to install. To be precise: although it will be fully installed, the driver will not be active, and the Yamaha CBX icon will not appear in Control Panels. If that happens, do not panic, just follow the instructions below, literally.

- Install the driver as described above.
- Locate the file CBXT3.DRV installed by the Yamaha program (in principle the file should be in C:\Windows\System).
- Copy the file into a temporary subdirectory created for the occasion.
- Create a text file and name it OEMSETUP.INF in this same directory. Include these lines in the .INF file (stick to syntax and order):

```
[disks]
1=., "YAMAHA CBX Driver for
Windows95", disk1
```

```
[Installable.Drivers]
CBXT3 = 1:CBXT3.drv, "MIDI", "YAMAHA CBX
Driver for Windows95",
```



- Open the Windows Control Panel
- Launch 'Add New Hardware'
- Click on 'Next' until the system wants to start an automatic search for new hardware. In Windows 98, you'll have to allow the system to do a search for Plug and Play Devices. Next, choose 'select hardware from a list'.
- Select the category 'sound, video and game controllers'.
- Click on 'Have Disk'.
- Browse until you find the files CTBXT3.INF and OEMSETUP.INF. The .INF file should appear automatically in the window.
- Respond 'OK' to the questions that follow, then click on 'Finish'.
- The software should now appear in the Control Panel window as described above. If so, you may restart your PC (if it has not already prompted you to do so).

Normally, at this point, you will have a new MIDI interface listed in the list of available devices under Windows. To test your interface, you can use any sequencer running under Windows, that permits interface selection (the majority of sequencers like Cubase, MIDI Workshop, etc., have this functionality). If you do not know how to set up your sequencer for other MIDI hardware, you can simply use the Media Player, after configuring the driver as the standard MIDI port for Windows. To do so, open the Control Panel and start the Multimedia control. Then select the MIDI folder and pick 'Yamaha serial

driver' from the driver list.

Note that it is possible to adapt the driver behaviour (in some respects), especially for SYSEX messages. This functionality is not available for all drivers, but if you experience some problems with System Exclusive messages, take a moment to look at text files to know if adaptations should be necessary.

If you notice that your serial port seems not to work correctly for some other Windows applications (it happens mostly with some portable computers), you just need to open the 'YAMAHA CBX Driver' icon in the Control Panel, and then check the 'Driver Disable' box.

A final interesting point to note: the Yamaha driver frees the serial port when there is no multimedia application using it. This port is then available for other applications when MIDI is not useful.

And now, have fun with your new interface...

(000139-1)

