

The MIDI ViewportTM Atmel Version

The MIDI Viewport is a data analyzer for MIDI messages. But it is also much more. Handheld, battery powered, and compact, it displays MIDI data in the order it is received, one byte at a time--using an LCD 2 line by 16 character display. Connected to a keyboard or sequencer it will tell you, in English, which MIDI events are being sent by the device. You can scroll through the 64-byte buffer to see what is really going on in your MIDI system.

There is no power switch on the unit, so you can't leave it on and run down the batteries. Pressing either of the two buttons on the unit will wake it up. The Viewport will stay on as long as it is receiving MIDI data. If nothing is received for about 5 minutes, the Viewport saves its batteries by going back to sleep.

Cables bad? Not sure? Connect a cable between the Viewport's MIDI input and output and press both buttons. The unit will test the cable by sending MIDI events out the MIDI output port and listening for them to come back into the MIDI input. This is more rigorous test than a simple DC continuity check. You'll know that your cable can actually carry MIDI information.

Need to quickly find out if a MIDI instrument or sequencer is able to receive? Connect the Viewport's output to the device's (or computer's) input port. When you press both buttons on the Viewport it will play a *beacon*--Middle C notes on all 16 channels. If the receiving device makes sound (or your sequencer shows it is recording the events), you *know* MIDI is working.

The MIDI Viewport is an intermediate level project. These instructions describe how to construct the Viewport using an Atmel 89C2051 microprocessor. Similar instructions for a 40-pin Intel 87C51-based project are [here](#). A pre-programmed microprocessor is available for \$20 from the author for the Atmel version, but not for the 80C51 version. [Write me](#) for info.

For this project, you'll need a Atmel 89C2051 microprocessor, and a device programmer that can program it. And, you'll need a few other parts, including a suitable LCD display and a sturdy plastic or metal project case.

Parts, Parts, Parts

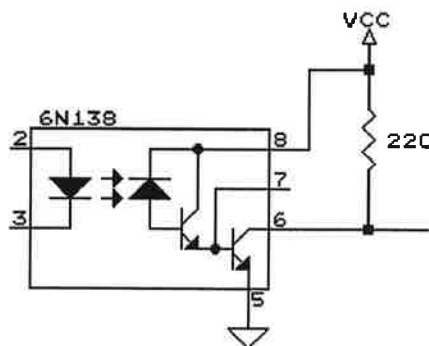
Here is the list of parts that you'll need to build the MIDI Viewport. Most of these parts are available from electronics distributors such as [Digi-Key](#) (especially good source for the LCD display), [Marshall Electronics](#), [Jameco](#), [B.G. Micro](#), and others.

<u>Qty</u>	<u>Part Number</u>	<u>Part Description</u>	<u>Reference Des.</u>
1	Optrex DMC-16207 or equiv.	LCD Display, 2 line by 16 characters	A1
1		4AA Battery Holder	BT1
1		Cap, 0.1 uF, Cer.	C1
3		Cap, 1 uF, Elect.	C2, C5, C6
2		Cap, 22 pF, Cer.	C3, C4
2	1N4148	Diode, Silicon	CR1, CR2
2		Connector, DIN, 5 pin, 180 Deg., PCB mount	P1, P2
1	2N2907	Transistor, PNP, TO-92	Q1
2	2N2222A	Transistor, NPN, TO-92	Q2, Q3
1		Potentiometer, 20K, 3/4 Turn	R1
4		Resistor, 220, 1/8W	R2, R3, R4, R5
2		Resistor, 10K, 1/8W	R6, R7
1		Resistor, 22K, 1/8W	R8
2		Switch, SPST, Momentary	SW1, SW2
1	89C2051-12PC	Atmel microprocessor, 128 byte RAM, 2K EEPROM	U1
1	PC-900	Opto-isolator, Sharp	U2
1	CD4011	Quad NAND Gate, CMOS	U3
1		Crystal, 12 MHz, HC-49	X1
1		Socket, 20-pin, low profile, for U1	

You'll also need a case of some sort. A small PacTec box is ideal. There are even some with built-in battery compartments. Select a box that is large enough to hold the LCD display, batteries, and other parts--preferably one that is palm-sized.

There are a wide range of 2x16 LCD displays that will work; most use the same Hitachi controller chips and interface pinouts. Make sure the display is not backlit--battery life would be far too short with the extra current drain. A suitable display should cost \$15-20.

If you can't find the Sharp PC-900 opto-isolator, substitute the more common 6N138. It uses a different pinout, so you'll need to connect it like this:



Construction

Position the two momentary switches on the sides of the case. When you hold the Viewport in your palm you should be able to push one button with your thumb and the other with another finger of the same hand. Alternatively, put the two buttons on the top surface of the Viewport, hold the unit with both hands, and push the buttons with your thumbs.

Drill two holes for the MIDI input and output connectors. In the interest of sanity, permanently mark which connector is the input and which is the output on the top of the case.

[illegible]

Click on this image to see a larger, printable version.

When this project is constructed properly, a fresh set of Alkaline batteries will last 6 months or more (or about 100 hours of continuous use).

Program the microprocessor using [this firmware](#). Install the chip in its socket and connect the LCD display to the board. Notice that this firmware is different from that used in the Intel 80C51 version of this project.

Testing and Final Assembly

Install 4 fresh AA batteries in the battery holder. The unit should turn on and display a signon screen and copyright notice. After a couple of seconds you will see the display below. You may have to adjust the contrast pot in order to see the image on the LCD display.

```
00 00 <00>
MIDI DATA = 00
```

Connect a known good MIDI cable between the MIDI input and output jacks and press and hold both buttons. The display should change to:

```
00 00 *00*
CABLE TESTS OK
```

If you see these displays, congratulations! You have a working Viewport.

If not, check your wiring. Measure the voltage at U1-20 and U3-14. It should read between 5 and 6 volts. Check to be sure that the microprocessor is properly programmed and installed correctly.

Once the Viewport is working, clean the circuit board using a commercial PCB degreasing spray. Battery life is greatly affected by the presence of solder rosin and dirt. Any rosin left on the board acts as a discharge path for the batteries. Instead of lasting 6 months or more, the batteries will need to be replaced in only a month or two.

Reassemble the unit in its case and reinstall the batteries. The unit should turn on normally. One final test: disconnect the Viewport from any MIDI devices and let it sit idle. It should turn off in about 5 minutes and turn back on when either button is pressed.

Using the MIDI Viewport

Internally, the Viewport stores the last 64 bytes of MIDI data that it has received. The top line of the display shows up to 5 consecutive bytes at a time, in hexadecimal. The two buttons on the unit serve to scroll the display. The right button moves the display earlier in time (moving the bytes to the right) while the left button moves the display later in time. The center-most byte, surrounded by brackets, indicates the current location in the 64 byte buffer that is being analyzed.

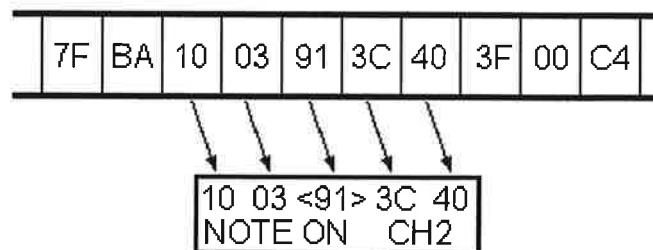


Fig. 2. Displaying the buffer data

Figure 2 shows a typical sequence of bytes received by the Viewport. Notice that the byte in the middle

of the display, inside the brackets, is the status byte for a Note On message on channel 2. The Viewport decodes the hex value for you, displaying "NOTE ON CH2". If the left button is pressed once, moving to the next byte in time, the display would read:

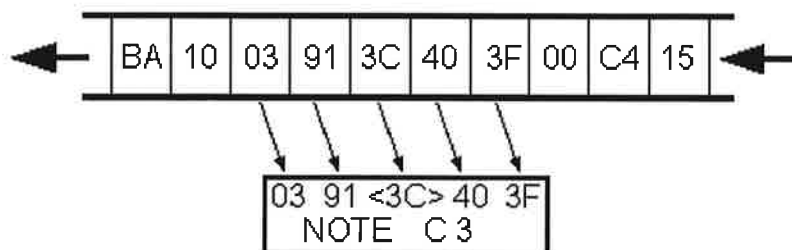


Fig. 3. Scrolled left one byte

Two MIDI events never appear in the Viewport buffer: MIDI Timing Clock (F8) and Active Sensing (FE). These events repeat at a rapid pace when sent by a MIDI device. If stored in the buffer they would cause other events to scroll out of sight before they could be viewed and analyzed. Instead, the Viewport displays these events when received by showing a small 'C' (for MIDI Clocks) or a small 'A' (for Active Sensing). These enunciators appear to the left and right of the brackets in the display.

The Viewport will echo received events to its MIDI output, so it can be put inline in a MIDI system. However, don't leave the unit connected for long periods. The battery will run down more quickly (although it should last for 100 hours or more). Alternatively, an idle unit will turn itself off and cease echoing events.

Once they've used a Viewport, most people find that they can't live without one. Performing musicians, touring groups, software developers, hobbyists, and technicians all rely on this little data analyzer to keep them sane. Hopefully, you'll find the MIDI Viewport indispensable too.

The MIDI Viewport™ is a trademark of Paul Messick and John Battle.

The MIDI Viewport firmware, display screens and circuit design are Copyright (c) Paul Messick and John Battle, 1989-1998. All rights reserved.