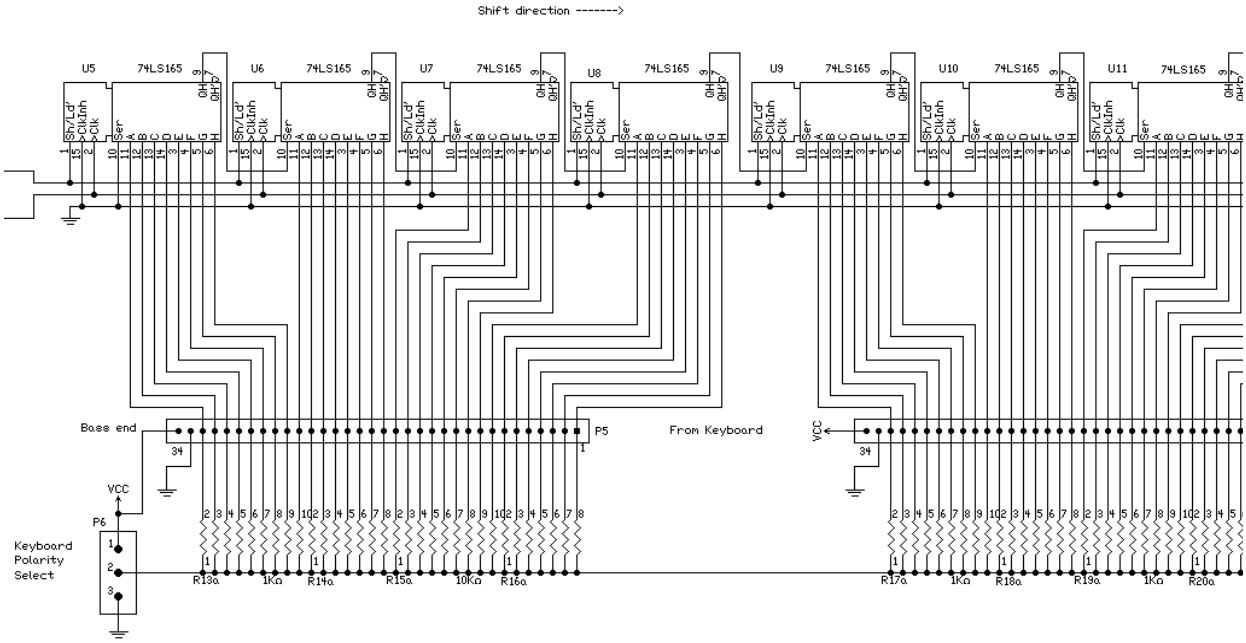
The MD-1 PC Board

The MD-1 board scans up to 64 key switches (which can be the 61 keys on a keyboard, or 32 switches on the pedals, or up to 64 stop tabs, or some combination thereof), and outputs MIDI note on and note off messages which tell when a switch goes on (key down) or off (key up). It has a few options which allow it to do other things as well, but these are just extras -- they don't affect the main purpose which is to provide a MIDI output from a keyboard.

The circuit consists of two main parts: the 74HC165 keyboard scanner circuits shown in Fig. 1, and the 68HC11 processor circuitry shown in Fig. 2.

74HC165 Keyboard Scanner

The circuit in Fig. 1 is really very simple - it has eight 74HC165 shift register IC's, connected to 64 pullup resistors and two connectors.

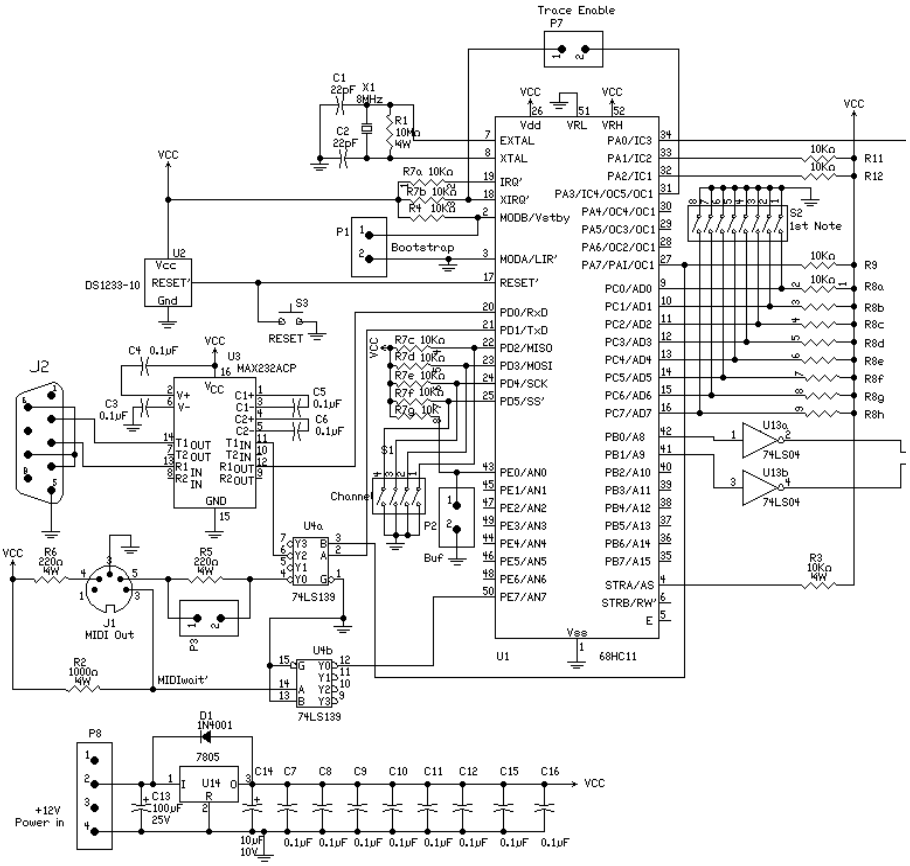


The 64 keys connect via two 34-conductor ribbon cables to connectors P4 and P5, 32 to each cable. The 64 resistors can be connected to either ground or +5 volts via a Keyboard Polarity Select jumper at P6 - if the keys go to ground, then the resistors go to +5 volts Vcc, and vice versa. A two-instruction change in the program in the microcomputer can accommodate either polarity, but I am using resistors to +5 volts and switches to ground.

The keys connect to a 64-stage shift register consisting of U5 through U12. The 68HC11 microprocessor alternately sends a LOAD' pulse to load key data into the register (pin 1 on all the 74165's), and then 64 SHIFT pulses to pin 2 on each 74165. The LOAD' pulse enters the data from the 64 keys into the 64 flip-flops in the shift register, while the 64 SHIFT pulses gradually "slide" the data, one bit at a time, into pin 34 of the 68HC11, where it is analyzed.

The 68HC11 circuitry

This circuitry, shown in Fig. 2, is much more complex. Let's explain the various parts, starting at the top left corner of the 68HC11.



(a) Top left is the 8 MHz crystal and attached circuitry, which sets the speed of the processor. The 68HC11 internally divides this by 4, so the main clock frequency of the processor is 2 MHz.

(b) Several resistors under that (and resistors elsewhere) connect various pins of the HC11 to +5-volt Vcc, to provide a pullup function - that is, to hold these pins at +5 volts when they are not used for other functions.

(c) Jumper P1 allows bootstrapping the HC11. In normal mode, this jumper is open. When pins 1 and 2 of the jumper are shorted together, the processor goes into a bootstrap mode, where it can be loaded with a program through the serial port. This is the method that the EEPROM software can initially be loaded.

(d) The DS1233-10 IC, U2, is a reset circuit. It monitors the Vcc power, and automatically resets the processor -- i.e., restarts it -- when the power goes below about 4.5 volts. Switch S3 also does a reset when pressed.

(e) Switch S1 is a four-pole DIP switch which sets the channel number when using the MIDI output. MIDI signals can be assigned a channel from 1 to 16; switch S1 sets that.

(f) The 68HC11 has a built-in serial I/O port which can run at the standard baud rates, as well as at 31.25 kHz, the MIDI rate. It uses pins 20 and 21. This serial port connects both to 9-pin RS-232 connector J2, and also to MIDI OUT jack J1.

The MD-1 board has two serial ports, though only one of them can be used at a time:

- There is 9-pin DB9F connector, which can be connected to the comm port of a PC. It is mainly there to provide a means of programming (bootstrapping) the microcontroller, and running tests, but it can also be used for other purposes -- for example, the two boards used as keyboard controllers (cards A and C) input the note-on velocity through this connector. The MAX232 IC, U3, is used as the converter between the low-level TTL voltages used by the HC11, and the positive and negative larger voltages used by typical RS-232 ports at the DB9F connector. It includes a charge pump which provides the necessary +10 and -10 volt power for output).

- There is also a 5-pin MIDI connector, which is the main MIDI output.

The MIDI output to J1 pins 4 and 5 is pretty standard, but the circuit has a few additions which were originally planned for other uses, such as handshaking on the MIDI jack via the MIDIwait' signal (buffered by half of the 74LS139).

Serial output from U1 pin 21 goes through U4a, a 74LS139. I'm using it to steer the output to either the MAX232 or the MIDI jack. It is controlled by its B input pin, which comes from pin 27 of the HC11 - it is normally held high on CPU reset, so the normal connection is to the 9-pin connector. If the 68HC wants to output MIDI, it grounds this B input (via U1 pin 27), and changes the baud rate to 31.25 kHz.

(g) Jumpers P2 and P7 are usually left open in normal use, but they allow this board to run with the Motorola Buffalo version of the 68HC11 for debugging purposes. P2 selects whether the Buffalo program will run, or whether the HC11 will instead jump to location B600 on bootup. P7 is used to enable tracing and breakpoints. As mentioned, these two jumper headers are here strictly for debugging and other applications.

(h) Power supply circuitry at the bottom includes a 7805 regulator and filtering, so the board can be fed from an unregulated DC source, or even from a "wall wart" converter. If you use a +5-volt supply, then the 7805 can be left out and its terminals 1 and 3 shorted together.

(i) Two 74LS04 inverters are used to buffer the LOAD' and SHIFT signals to the shift registers.

(j) Switch S2 is an 8-pole SPST DIP switch which is not used by the current software, and need not be installed. (In software versions before 1.0, it was used to select the first or lowest note on the keyboard. For example, suppose the lowest key on a given keyboard is the C two octaves below middle C. Since middle C is defined as note number 60 in MIDI, two octaves below that is 24 keys lower, so this switch would be set to the binary equivalent of 60 minus 24, or 36, which is

hex 24, or binary 00100100. The current software versions default to this value without using the switch.)

(k) The diagram shown is for revision B of the pc board. The photo below shows a revision A board, which lacked the reset switch S3. The latest board is revision C, which has an LED which flashes when there is serial output. All three revisions are otherwise identical, and any of the three works as well as any other; the changes are strictly for debugging purposes, and are useless otherwise.

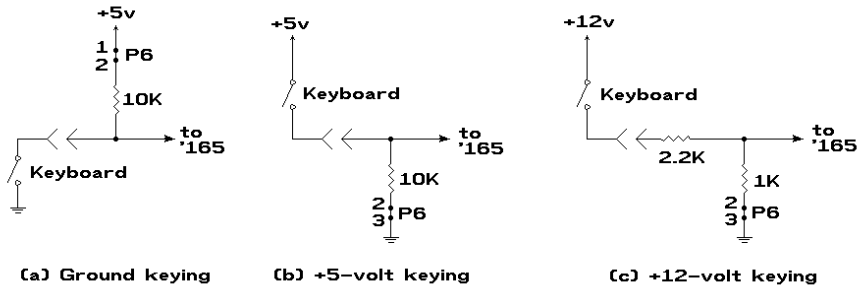
Keyboard Connection

Fig. 3 shows how the keyboard contacts connect to the rest of the system. There are essentially three possible ways to do it:

Fig. 3 (a) is how I do it on my organ. Pins 1 and 2 of P6 are jumpered, so the bunch of 10K resistors connected to the 74165 IC's bring +5 volts to the shift register inputs. The keyboard contacts then all go to ground. Hence the IC input is at +5 volts when the key is up, and goes to ground when the key goes down to be played.

Fig. 3 (b) is an alternative connection; it would work just as well, but it requires that the keyboard frame go to +5 volts instead of ground. In this case, connecting pins 2 and 3 on P6 makes the IC inputs normally grounded, but pressing a key pulls the corresponding input to +5 volts. Since the inputs now work opposite to the previous circuit, the program also needs to be told to interpret the voltages differently. This is done by changing a pair of instructions in the program.

Fig. 3 (c) shows how it would be done in organs that use +12 volts for keying. The same software change is needed as in (b) above, but in addition we need to modify the circuit to reduce the +12 volts down to under +5 volts, to make it safe for the ICs.
This change is actually very simple: (1) for the eight resistor packs R13 through R20, use 1K packs instead of 10K packs, and (2) instead of using connectors P4 and P5 (each of which has 34 pins, for a total of 68 pins), replace them with 68 quarter-watt resistors. Pins 1 through 32 each get replaced with 2.2K resistors, while pins 33 and 34 get replaced with 1 ohm or less (or better yet, just a piece of leftover wire from one of the other resistors.) In each case solder the resistor flush with the board on the component side, standing up. Then trim all the leads sticking up from the board so they stand straight up about 1/4 inch. In other words, the top ends of the resistors stand up like a new connector, and the ribbon cable connector will slip directly on them.

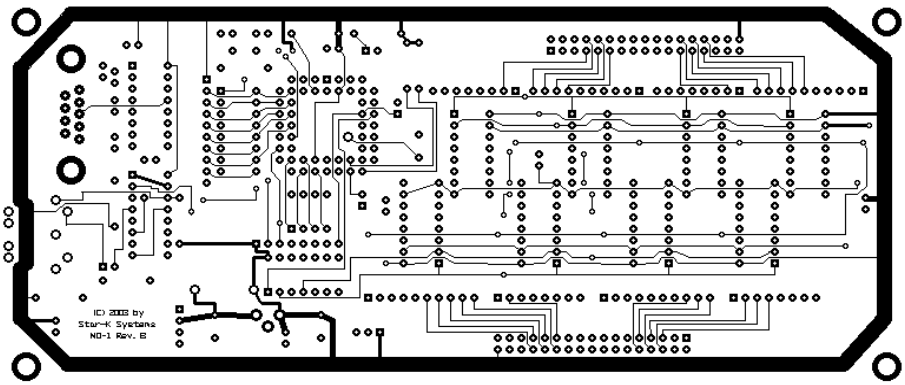


The resistor values shown are for 74HC ICs. If using 74LS, change the 10K resistors in (b) to 1K.

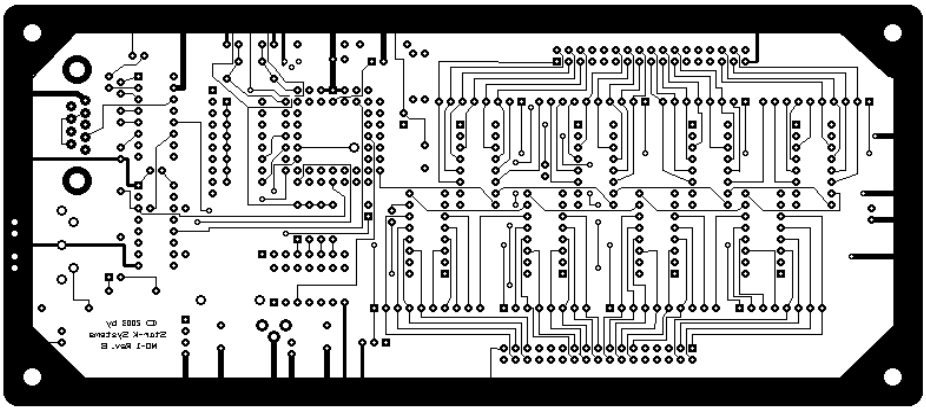
PCB Layout

The following figures show the MD-1 PC board layout (revision B). The board is a two-sided board with plated through holes, and the size is approximately 3.5 by 8 inches. While I do not sell any blank PC boards, kits, or completed boards, you can make your own boards from the layouts shown below, or there are a number of PC board manufacturers who can make a board for you if you send them the Gerber files available here.

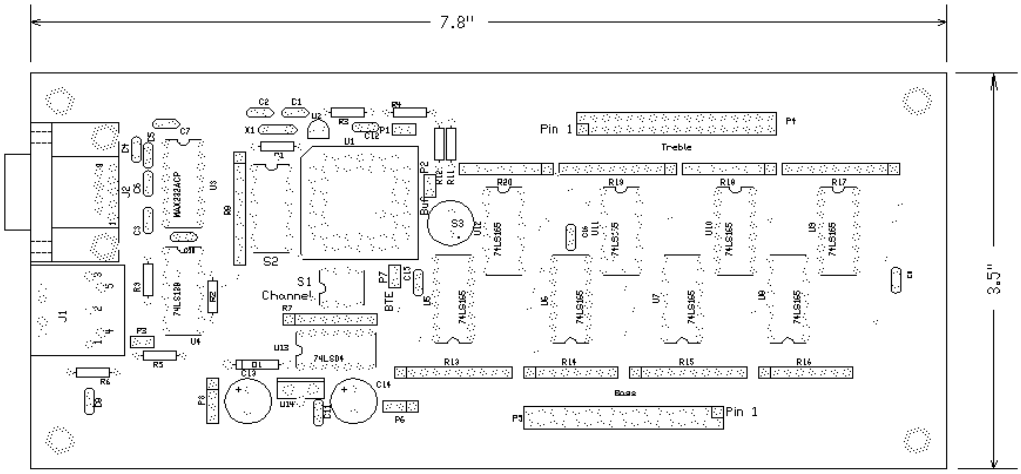
The top solder side:



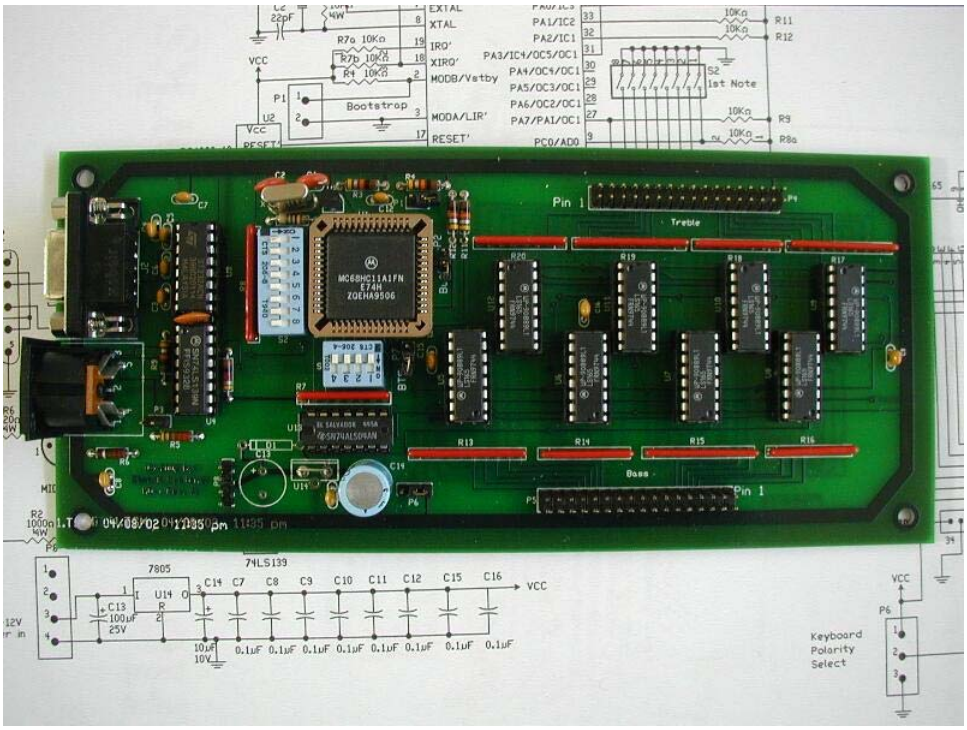
The bottom solder side (as viewed from the top, through the board):



The top silk screen, which shows the location of components:



The completed PC board:



The drawings above show the revision B PC board, whereas the photograph is of a slightly earlier revision A board which worked fine, but lacked reset switch S3.

THE SOFTWARE

The 68HC11A1 has 512 bytes of EEPROM and 256 bytes of RAM, and only about 50% of each is used in the program. The board layout permits the EEPROM to be written right on the board.

To help you understand the program, here is a brief synopsis:

The program resides in EEPROM, and is started automatically by the Buffalo monitor in the processor. Buffalo handles the reset and interrupt vectors, so they do not appear in the program. The program starts by setting up the serial port for 31.25K baud operation as required for MIDI, and initializes the various ports. Then it reads the desired MIDI channel number from the 4-position DIP switch on port D, and the "lowest note" from the 8-position DIP switch on port C, and stores these values in RAM. Then it erases its data area and sends out 128 note-off messages to make sure everything is quiet.

The RAM contains a 64-byte table called DATA, one byte for each key. The bit pattern in each byte is like this:

Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0

Bit 7 is the current state of that note: 1 means the note is currently playing, 0 means it is currently quiet.
Bit 0 is the latest key position detected: 0 is up, 1 is down.
Bits 1 through 6, in that order, are the 6 previous key positions as detected on the 6 previous scans.

The program has two loops, of which the outer loop sends a LOAD pulse to the shift register to load all key data into the shift register, and initializes some pointers to the data.

Then comes the inner loop. It takes a quick look at the current bit coming in from the shift register, and then steers the program to one of four points: it goes to

- SILEDN if the note is silent but the key is down. In this case it immediately sends out a note-on MIDI message and stores that info in the DATA array.
- SILEUP if the note is silent but the key is up. In this case it just stores that info in the DATA array.
- PLAYDN if the note is playing and the key is down. Here too it just stores that info in the array.
- PLAYUP if the note is playing but the key is up. This condition could be due to a dirty switch contact, so the program checks to see the past history of the key -- this is where bits 6 through 1 of the DATA array location come into play. If all of these are also 0, indicating that the key was up on the previous six scans, then the program issues a note-off message to really turn off the note. Otherwise, it merely updates the data, but leaves the note playing. This procedure is normally called "debouncing" the switch.

As shown, the program is set up for switches connected to ground, as in Fig. 3 (a) above. To use a keyboard which generates either +5 or +12 volts (as in (b) or (c) above), the program has to invert the input. This is done by a minor change in the program: in the two instructions BNE SILEUP and BNE PLAYUP, the BNE must be changed to BEQ (which changes the two "branch if not equal" instructions to "branch if equal".).

The above text describes **software version 1.0, 1.1, and below**, used on the MD-1 pedal board and most stop switch boards (boards B and D)..
Software version 1.0C (no longer current) contains the following changes:
1. Instead of issuing 128 note-off messages at the beginning, the software outputs an "all-notes-off" message.
2. 8-position DIP switch S2 is not used; instead, the lowest note number 36 or $24 is defaulted into the program as an EQUate at LOWNOT. This is valid since all known keyboards start on the C two octaves below middle C.
3. The NOTEON subroutine has been modified so that it recognizes special key combinations to change program/patch or channel numbers. The operation is described further below.
Software version 1.0Cvel (used on the two keyboard decoders, boards A and C) contains the above changes plus the velocity is input from the DB9F connector at 31.25 kbaud from the corresponding MD-2 board. If nothing is received, then the velocity defaults to hex $70, as before.
Software versions 1.1C and 1.1Cvel contains all of the features of versions 1.0C or 1.0Cvel as above, plus they operate the COP timer. Some of the 68HC11A1FN chips sold on Ebay are used, and have the COP circuit enabled, with the result that these circuits will not work with the above programs. Rather than disable the COP timer (which takes some effort), these software versions simply 'tickle' the COP timer periodically to keep it running. (The COP or 'Computer Operating Properly" timer on the 68HC11 does an automatic reboot if the running program does not periodically 'tickle' it.)
Software version 1.0S is unique to my system, and is modified to do program changes for the Roland CM32P in addition to Hauptwerk and/or Kloria MyOrgan. It is not described in these pages.

A Note on the DIP Switches, Jumpers, and Operation

DIP Switches: The two DIP switches set the operating parameters as follows:

DIP switch S1 sets the channel number *minus 1*. That is, MIDI channels are numbered 1 through 16, but the binary code used is 0 through 15. For example, if I want to play on channel 13, I actually set S1 to 1100, which is the binary number for 12. The way the switches are wired, an open switch generates a 1 while a closed switch is a 0, so the four switches are set to open-open-closed-closed.

DIP switch S2 sets the number of the lowest key on the keyboard, which is usually key number 36 (hex 24), the C two octaves below middle C. It is used with version 1.0 software, but not with version 1.0C software and later, where the number 36 (hex 24) is automatically set in the software code since all full-size keyboards to our knowledge start at that same note, and hence it is not necessary to make this changeable.

Jumpers: There are several header strips for jumpers. P6 has been explained above -- depending on whether the keyboard switches go to +5 volts or ground, it connects the pullup/pulldown resistors to the opposite voltage. P3 is normally open -- it would only be used if you wanted to use the MIDI output connector for some other purpose, such as directly outputting TTL levels.

When using a 68HC11A1 that has Motorola's Buffalo debugger programmed into its ROM, three others (P1, P2, and P7) determine the operating mode for the software, as follows:

1. To run Buffalo, place a shorting jumper on P2 and P7. This allows you to do various debugging functions via the serial port; if you have programmed the MD-1 software into the chip, you can do a manual jump to it at location B600.
2. To have Buffalo start, but then right away automatically jump to the MD-1 software at location B600, remove all the shorting jumpers. This is the normal operating mode once the board is installed into the organ.
3. To go into bootstrap mode, so you can program the chip via the serial port, place a shorting jumper on P1.

Construction Information

* Click here for a list of parts, prices, and suggested sources.
* Click here to download the "Gerber" computer files that may be used to order blank pc boards from any normal pc board manufacturer (see the next paragraph for information).
* Click here for information on programming the 68HC11

Although the circuit could be wired in many different ways, it is easiest to order a bare printed circuit board to hold all the parts. The Gerber files above are what a pc board manufacturer uses to build the board. There are a number of such manufacturers who deal via the internet in small "prototype" quantities, and who can make boards for you. For example, www.pcbnet.com will make five of these boards for $110 total. www.BareBonesPCB.com charges 50 cents per square inch plus $25 per lot. The MD-1 board is 28 square inches, so the first bare board would cost $39, and each succeeding board would be $14. Before using any of these suppliers, check the web -- there are probably even cheaper ones by now.