MIDI MATRIX 2

PROJECT

Neil Johnson describes the design and construction of a versatile eight-channel MIDI routing unit for the home or small studio.

Part 1

The world of electronic music consists of gadgets and cables. Gadgets are fun - synthesisers, mixers, effects, and a myriad of other weird and wonderful things. Cables, on the other hand, are not so fun, especially when it is time to rearrange the wiring of even the most basic studio.

Several solutions have been developed over the years. The most successful and easy to use are based on a patching arrangement, either using individual patch leads, or an electronic version using a variation of the "telephone exchange" principle - lots of inputs, lots of outputs, and many electronic or electro-mechanical switches between them.

In a typical small project or home studio there are two types of cable: audio cables, carrying audio signals between gadgets, and MIDI cables connecting all manner of computer-controlled gadgetry together. Unless considerable forethought and planning is undertaken, it is highly likely that mostly MIDI cables will, at some time or other, need to be moved. Most audio cables, however, once arranged to feed everything into the main mixer, tend to stay put.

The original MIDI Matrix had six channels and six rotary switches. Routing consisted of setting the switches to the desired positions. While simple and robust (the original unit has lasted over six years) it is not very easy to remember the switch settings for the various patch configurations - especially if, like the author, you have a poor memory!

So, as part of a studio upgrade plan it was decided to revamp the MIDI Matrix, giving the old idea a new spin and bringing it into the 21st century.

The Next Generation

The MIDI Matrix 2 (MM2) is an eight-channel programmable MIDI routing unit, with enough storage for sixty-four preset patches in an internal non-volatile EEPROM. Patches can be selected, edited and saved from the front panel. Patches can also be selected by Program Change commands received from a MIDI device (for example, a computer or a synthesizer).

Ease of use was a key consideration during the design stage. All the major functions have their own dedicated button, requiring only a small display for additional information. Also, an active LED display, rather than a passive LCD, greatly improves the readability of the display during use. Most studio gadgets these days have LCDs, often backlit, resulting in narrow viewing angles, fading contrast and backlights that last only a few years.

Blocks Away

Cast your eyes over the system block diagram shown in Figure 1. The system is split into two main sections, dealing with the user interface and the MIDI routing respectively. The heart of the system is the microcontroller. For reasons familiar with most constructors, the device chosen for this project is the Intel 8031 or compatible (i.e. 8032, 8051, Dallas 80C320, etc). The reason? Well, two in fact: the author had one at the bottom of his semiconductor junk box, and after a clear-out at work a Keil ANSI C compiler was made available for development.

The microcontroller starts with a reset from the on-board power-on reset circuit. The ROM contains the program code for the microcontroller, while the serial EEPROM stores the current MIDI channel number and the sixty-four patch memories, ready for instant recall by the user.

Moving up to the user interface, the LED display and button LEDs are driven by a four-way multiplexed display driver. This

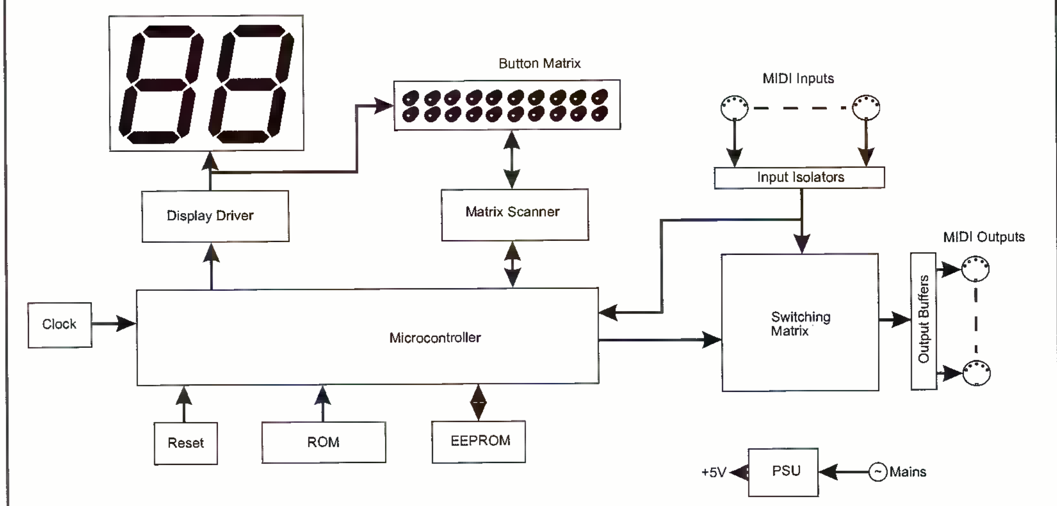


Figure 1. Overview of the MIDI matrix2

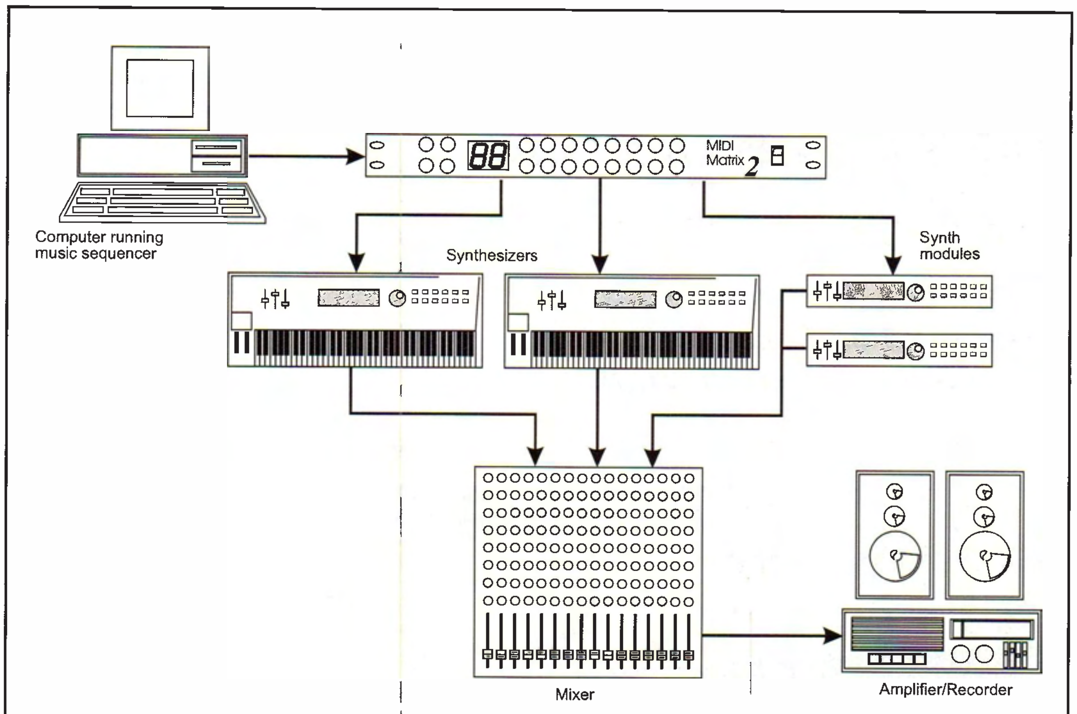


Figure 2. Typical MIDI based studio setup

method has two attractive features. Firstly, having one quarter of the LEDs on at a time reduces the current drawn by the display. Secondly, the number of wires from the driver to the display board is also reduced by a considerable margin. This project has thirty-two front panel LEDs: a direct drive method would need thirty-three wires, whereas the multiplexed method used here reduces this to twelve wires (four common-anode sources, eight cathode sinks).

The button scanning matrix works on a similar principle. There are twenty buttons to scan, in four banks of five, needing a total of nine wires between the front panel and the scanning circuit. Extra circuitry is included to ensure that multiple button presses are processed correctly.

The second main section of the MM2 is the MIDI routing and buffering section. Here, eight MIDI inputs are opto-isolated and buffered before being fed to the eight inputs of the routing matrix. Input channel eight is also taken to the microcontroller to allow remote control from a computer or keyboard (either a synthesizer of a master keyboard). The eight outputs of the routing matrix are buffered before being brought back out to the world through the back panel connectors.

The software running on the micro-controller drives the entire system. It

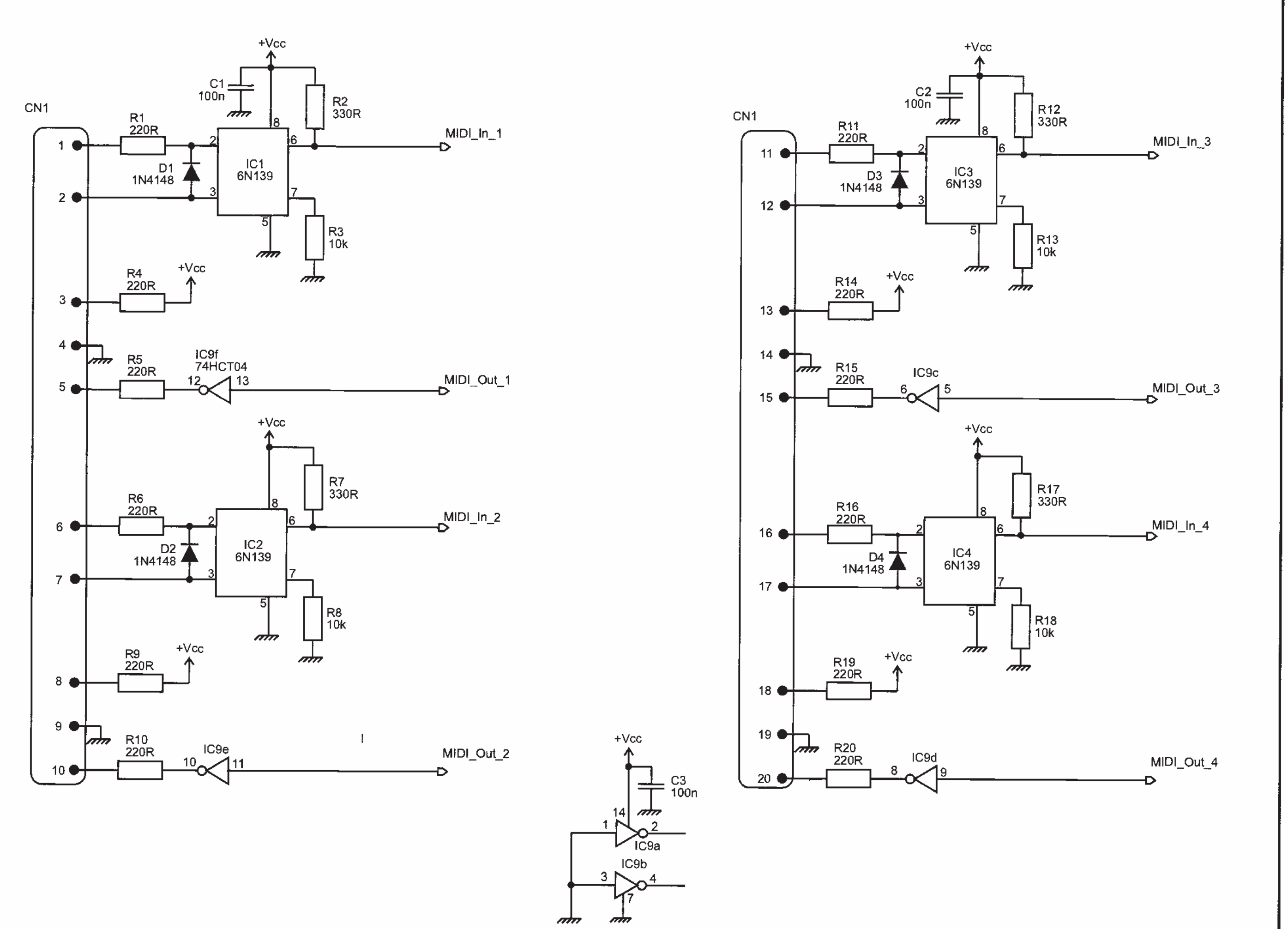


Figure 3. First half of MIDI Interface

A Brief Guide to MIDI

MIDI (Musical Instrument Digital Interface) is a communications system for passing control messages between electronic musical instruments, computers, recorders, lighting controllers, effects units, and so on. It was born in 1982, the child of a number of synthesizer manufacturers, and was helped into the marketplace with the launch of the immensely successful Yamaha DX7 keyboard.

MIDI uses a serial data format (one start bit, eight data bits, one stop bit, no parity) operating at 31,250 Baud. A '0' is represented by a minimum current of 5mA, and a '1' - the idle state - by 0mA. All MIDI inputs are opto-isolated (to avoid ground loops) and the current is specified to drive the input of a high-speed opto-isolator such as the HP 6N138 or Sharp PC-900.

The length of each message varies depending on what data it is carrying and the priority the message has over other messages. In general, MIDI carries "event" information - which key was played, how hard it was played, time events, status messages, etc. MIDI can also carry other application-specific data in SysEx messages, but the relatively slow transfer rate makes this prohibitively slow for large amounts of data (e.g. large samples from a sampler into a computer).

multiplexes the display, scans and debounces the buttons, drives the MIDI matrix, receives and interprets MIDI command messages, manages the data in the EEPROM, and provides the user with a selection of menus and facilities for operating the unit.

Supporting the system is the power supply unit (PSU). This project is mains-powered, avoiding the use of wall-wart power supply blocks that are the bane of most studios. The author is proud of the fact that he has no external power supply blocks in his studio set up - although how long this desirable situation is likely to continue is another matter.

Figure 2 puts the MM2 into context with the rest of a small studio. On the left is a computer running some sort of sequencing software (Cakewalk, Cubase, etc). It provides one or more MIDI channels to control the rest of the studio, consisting of keyboards (top), synthesizers and samplers (right) and effects units (bottom). Finally, the mixer brings together the audio signals from the various units, producing the final output for listening or recording.

The mixer is shown connected to the MM2 to provide for mixer-automation control from the computer. This system is now available on many analogue mixers and all digital mixers, with the more expensive versions proving moving-fader automation.

Circuit Description

Figures 3 and 4 show the eight MIDI input opto-isolators and output buffers. MIDI uses a 5mA current loop to carry data between units (see separate box). Taking MIDI channel one as an example, the MIDI input current comes in through pin 1 of CN1, through R1 and the LED in IC1 (a high speed opto-isolator), and back out through pin 2. Diode D1 protects the LED in IC1 from reverse currents. R2 is the collector load resistor for IC1's open-collector output, while R3 provides bias for the Darlington output stage, reducing the turn-off time of the output.

The MIDI output circuit consists of R4, R5 and one-sixth of IC9. The hex inverting buffer provides the current switching capacity, while the resistors, together with the matching receiver resistor and LED, limit the current to the required 5mA.

The remaining seven MIDI channels operate in the exactly the same way. The unused inputs of IC9 and IC10 are tied low. Each has their own supply decoupling capacitor, as does each opto-isolator.

The microcontroller heart of the system

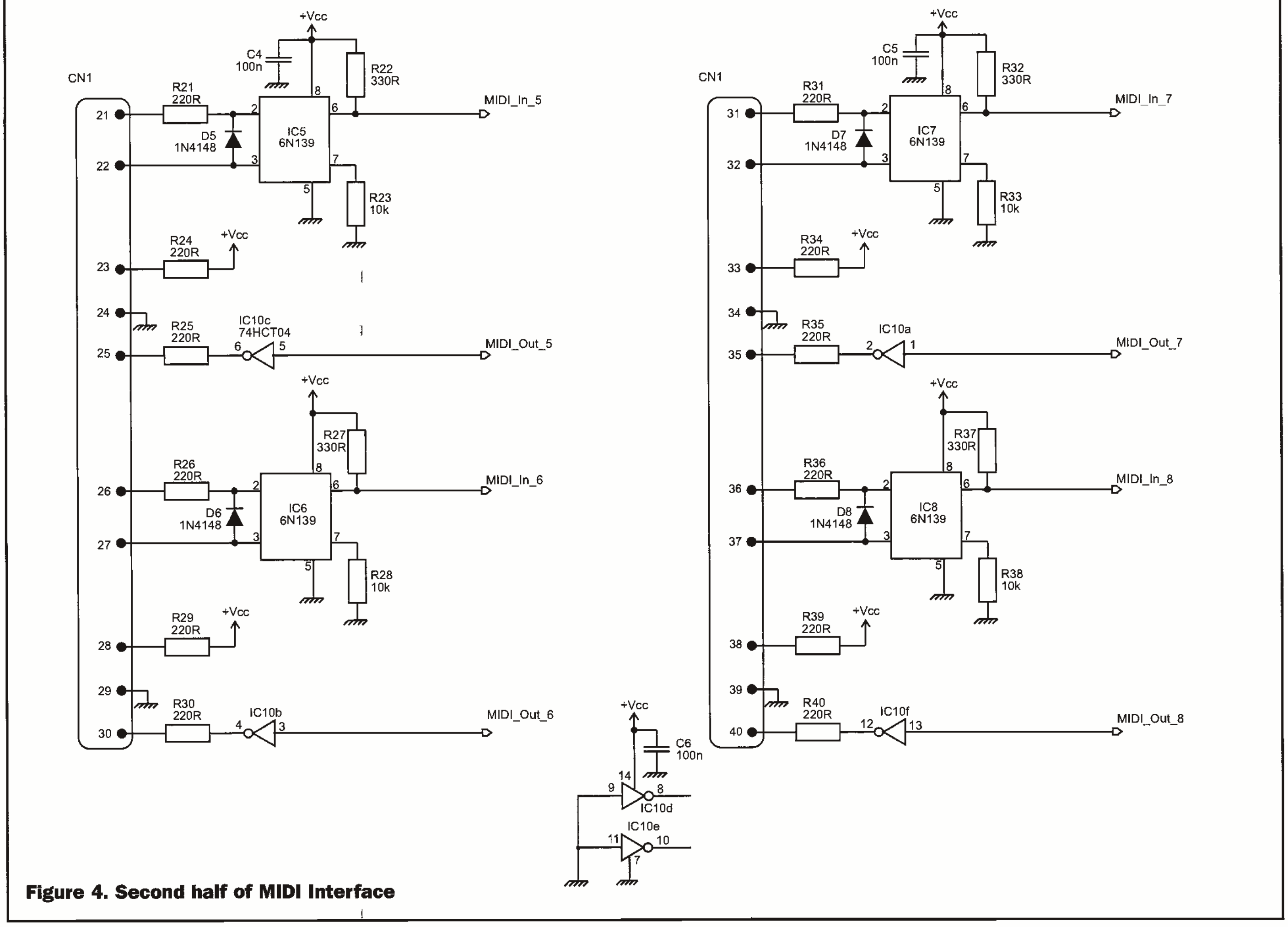


Figure 4. Second half of MIDI Interface

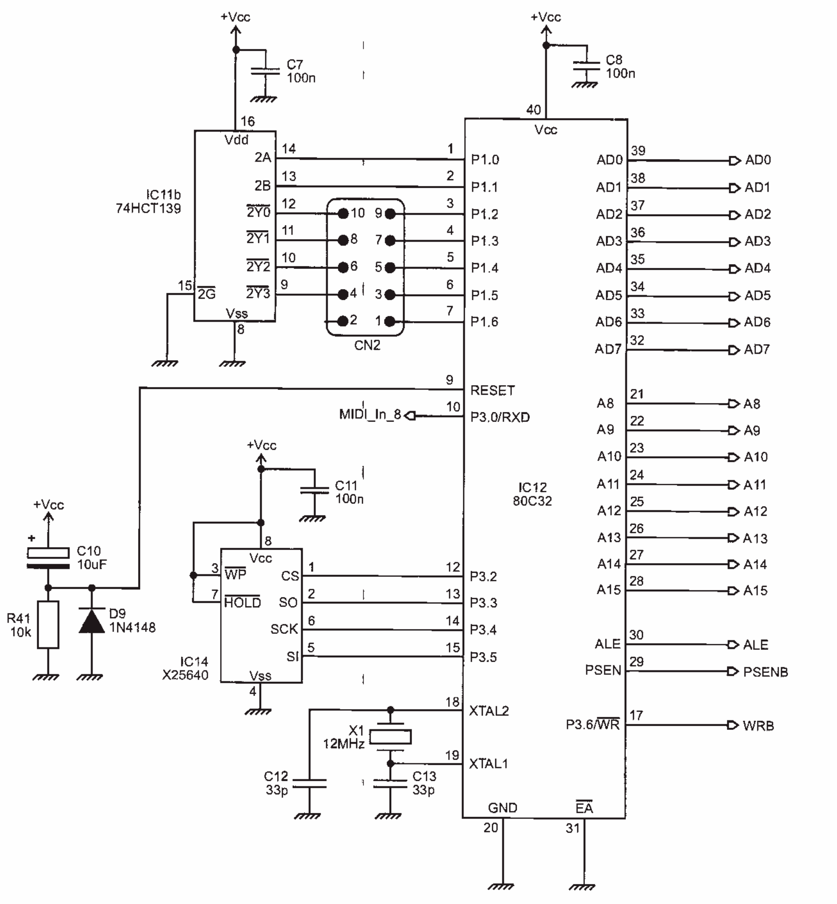


Figure 5. The brain of the MIDI Matrix- The microcontroller

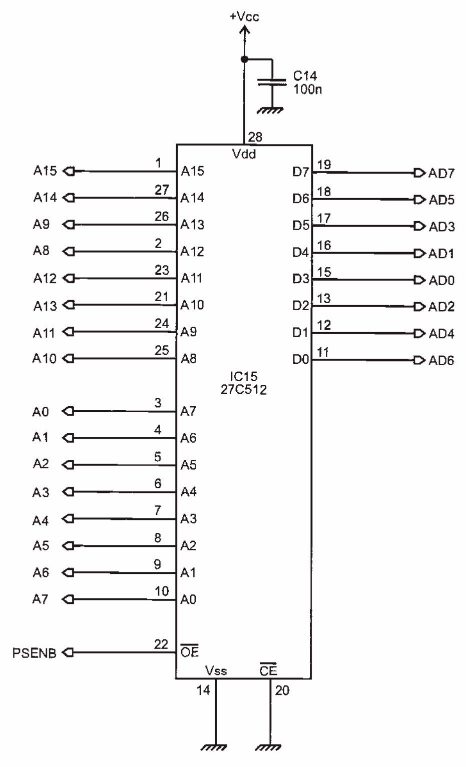
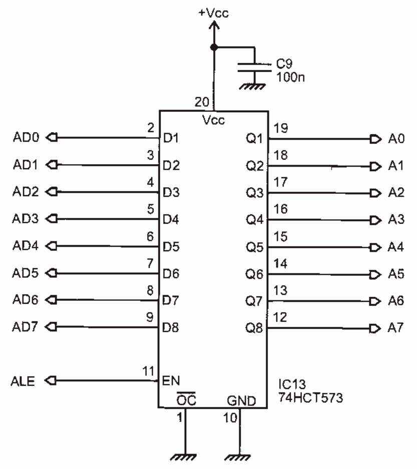


Figure 6. EPROM connections. Note the mismatch between device pin functions and the connections to the microcontroller. See text for explanation.



is shown in Figure 5. R41, D9 and C10 provide a power-on reset pulse to the reset input of the microcontroller, IC12. The non-volatile EEPROM connects to the microcontroller through four dedicated pins. Both the Write Protect and Hold functions are not used in this circuit, and are wired high. The processor's clock is generated by X1, with C12 and C13 providing the correct loading for the crystal, ensuring stable oscillation and guaranteed start-up behaviour.

The keyboard connects directly to CN2. Scanning is implemented with one half of IC11 (to provide the four column select lines) with the five row sense lines returning to the microcontroller through pins 3 to 7.

The 80C32 family of microcontrollers multiplexes the lower half of the address bus with the data bus. This extra complication arises from the designer's efforts to fit the device into a 40-pin package. This scheme was also used on the once-popular 8085 microprocessor, and latterly on the 8086. During memory accesses the lower half of the address bus is latched by IC13. The ALE (Address Latch Enable) signal holds the latch during the data read or write half of the memory access cycle.

The program code is stored in an EPROM (Figure 6). The strange connection pattern results from remapping the pinouts to help simplify the PCB layout. During software development, a program was written to automatically convert the programming files to suit the non-standard connections. This bizarre non-ideal system was used extensively during the early days of microprocessors, with double-sided PCBs and hand-laid track routes. Today, with multi-layer PCBs, advanced CAD software and surface mount devices, it is much easier to design systems with ideal circuit configurations.

The MIDI routing matrix itself, together with the multiplexed LED driver, will be shown in Figure 7 next month. All MIDI inputs and outputs connect to the large CPLD (complex programmable logic device) IC16, a Lattice Semiconductor ispLSI 1016E. Figure 8 (shown next month) the top-level structure of the CPLD design.

Each MIDI output is connected to one of the MIDI inputs through an 8-input multiplexer. In side the CPLD are eight such multiplexers, together with the latches and address decoding logic for mapping the registers into the microcontroller's external data memory.

The CPLD also contains the display registers for driving the multiplexed display. There are three 4-bit display registers - two for the eight cathode drivers, and one for the anode drive selector (the other half of IC11). The cathode drive circuit (IC17 and resistors R42-R49) sinks the current from one cathode on each of its eight lines. The four anode drive transistors (Q1-Q4) supply current to one of the four common-anode lines. This circuit drives all thirty-two LEDs in four banks of eight, split as: left digit and mode button, right digit and mode button, top select row, and bottom select row.

The PSU (Shown in Figure 9 next month) is a basic transformer-rectifier-voltage regulator unit. The regulator is protected from reverse voltages by D14. This is good practice, especially as these low-cost 3-terminal regulators do not include this degree of protection, and is very susceptible to reverse voltages. If the suggested size of case is used it is recommended that the specified transformer is used.

TO BE CONTINUED