The Roland PG1000

The PG-1000 is a dedicated programmer for the Roland D-50 keyboard synthesizer and the D-550 rack mounted version. It works with MIDI system exclusive, and requires an external 9V power supply. It has an impressive 56 faders used to set the four partials, two tones and one common parameter block for each D-50 patch. Programming was made easier by displaying parameter values on a backlit LCD screen along with dedicated partial/tone select buttons.

Its impressive number of faders and MIDI output immediately suggest a possible use as a generic MIDI controller device. When it came out in the late 1980's Midi Controller devices were rare. However, the MIDI output of the PG1000 is in System Exclusive form and each fader is tied to a specific D-50 function with a specific range of values; all of which does not easily translate to any use as a generic MIDI controller.

Upon opening up the PG1000, the internal circuitry turned out to be very simple and straightforward. It consists mainly of a NEC 78C10 Processor running code from a 32K PROM memory chip. The Processor includes eight 8-bit Analog to Digital Converters which are used to convert the 0 to 5 volt output from each slider. Each of the 8 ADCs handles 8 sliders routed through a 4051 CMOS Data Selector chip. Note that this scheme allows for a total of 64 sliders and the unit has only 56. One of the processor's ADC is not being used – a point that will turn out to be useful later.

The NEC 78C10 Processor, better described as a single chip microcomputer, has a number of other useful features put to good use in the PG1000: an internal serial I/O engine used for the MIDI I/O, internal clocks and counters used to set the MIDI and serial data rates, a small amount of RAM memory for storing setup data, and extra I/O lines dedicated to the LCD controller and several pushbuttons.

The Task

Seeing how simple and traceable the internal circuitry was, the task of reprogramming the PROM memory to turn the PG1000 into a MIDI slide controller became a viable possibility. To avoid as much assembly language programming as possible, I chose to set up a Forth Language operating system on the memory chip. This route was made possible by the development of the eForth system by C. H. Ting. The eForth system is a complete Forth system, designed to be small enough to fit on a memory chip. It requires that only 31 simple code words be built from assembly code for any particular processor. The remaining higher level eForth code words are then built from these 31 base words. With eForth, after programming the base words, I then had a complete higher-level programming language to facilitate building the main code that will turn the PG1000 into a MIDI Controller.

The rest of this paper will document how this was done. First, here is a description of the end product.

Modified Roland PG1000

This Roland PG1000 has been reprogrammed to put out conventional Midi command instead of System Exclusive.

The unit has 56 sliders numbered from 0 to 55. (There is also the capability for 8 external, zero to five volt, control voltage inputs numbered as sliders 56 to 63.) It also has 10 pushbuttons, 8 of which have taken on new functions as Cursor Left, Cursor Right, Field increment, Field Decrement, Slide increment, Slide Decrement, ENTER, and MIDI.

The unit has four modes of operation:

(1) Edit Mode.

Each Slider has an Edit Window on the LCD display with the following fields: Slider number, slider on/off, Midi channel number, Midi operation, Midi operation data, and Slider value.

An LCD cursor can be moved to any of the above fields using the Left/Right buttons. The selected field can then be edited using the Field Increment/Decrement buttons. The edited Slider Window is only loaded into memory when the Enter button is pressed. The Slide Increment/Decrement buttons enable you to step through the Slider Windows without moving the cursor.

No Midi data is sent while in the Edit Mode. The Slider value field provides a running display of the slider value.

The unit supports the following Midi Operations:

Key#
: Midi Key On is sent when a slider movement up from zero peaks out. The key value sent is programmed in the Midi data field and the key velocity sent is the peak value of the slider movement. A note off is sent when the slider returns to zero.

Key# AT
: Midi Key On with After-touch. Midi Key On/Off values are sent as described above in Key#. In addition, a continuous Midi

After-touch value is sent with any slider movement until it is returned to zero.

Control#
Midi Controller. A continuous controller value is sent with any slider movement. The Controller number is set in the Midi data field.

Program#
Midi Program Change. When the slider goes above a certain threshold value, the program change number as set in the midi data field is sent once. The data is not sent again unless the slider is returned below the threshold value.

Ch Press
Midi Channel Pressure. A continuous channel pressure value is sent with any slider movement. The midi data field is not used.

Ptch Whl
Pitch Wheel. A continuous pitch wheel value is sent with any slider movement. The midi data field is not used so that only a 7 bit value is sent.

(2) Midi Run Mode.

When the Midi button is pressed, the display will change to "Midi Running" and the enabled Sliders will start sending Midi data. To get back to Edit Mode press Enter, Increment, or Decrement.

(3) Setup Mode.

One problem with the unit is that when powered off, all the slider setting you set in the Edit Mode are lost. To help alleviate this inconvenience, a “Setup Mode” was programmed to allow you to easily setup the sliders with a number of fixed settings stored away in some available EPROM space.

To enter the Setup Mode move the cursor under the Edit field labeled "Slider#" and then press Up or Down. The display will change to read "Setup# nn". There are a total of 64 possible slider setups stored in ROM memory. To load a particular setup use the Up/Down buttons to select the setup number and then press the Enter Button. The display will then go back to Edit Mode.

Setup# 00 disables all 64 sliders. Setup# 01 was designed for a class. Most of the remaining setups act like Setup#00 but are empty, available for future customizing.

(4) Forth Mode.

The unit can have a serial input/output port which connects to any computer's serial I/O. With a terminal emulation program set for 9600 baud, 8-bits, 1 stop bit, you can access the Forth language operating system used in the unit. There is a small amount of Ram memory available on the processor chip for implementing your own programs.

The unit will exit its Edit program loop and enter the "Forth Mode" with any key action on the computer terminal when the serial port is connected. To re-enter the Edit program just type EDIT and return.