U S E R M A N U A L

MIDI CONTROL CENTRES

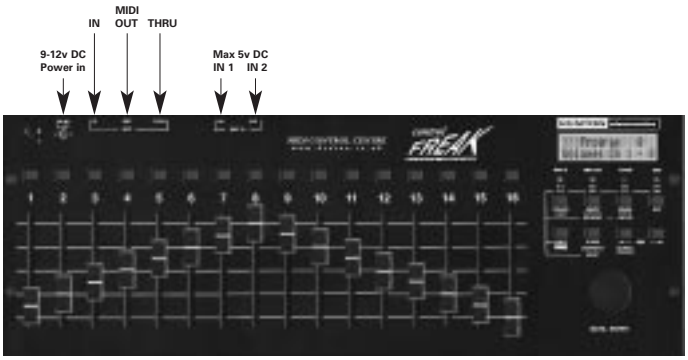
Control Freak STUDIO EDITION
&
Control Freak ORIGINAL

KENTON

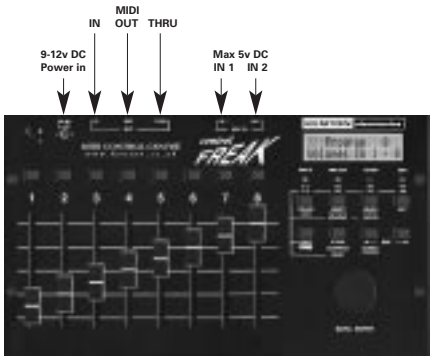
Foreword

Congratulations on your purchase of a Kenton Control Freak. Please take the time to read this manual thoroughly as it explains everything you will need to know in order to get the most from your unit.

Because of its ability to generate any type of MIDI message (including SysEx and NRPNs), the Control Freak allows you to unlock the full potential of any device that conforms to the MIDI standard. Of course, all this creative potential is great, but the depth and complexity of the MIDI control available can seem rather intimidating at first. Don't panic – read this manual carefully, try some examples, and in a short time enjoy total control of your MIDI equipment.



Control Freak STUDIO EDITION



Control Freak ORIGINAL

What is the Control Freak?

The Control Freak is a MIDI hardware controller which allows you to control any piece of MIDI equipment, be it a synth, soft-synth, soundcard, sequencer or anything else with a MIDI In port. The degree to which you can do this depends only on the amount of external MIDI control that particular piece of equipment has been designed to allow.

What are the benefits of using a Control Freak?

Many pieces of MIDI gear have no front panel controls of their own, others possess on-board controls which are frustrating and fiddly to operate. The Freak gives you the hands-on control you need, to access the various parameters on such pieces of equipment, instantly and in real time. The ability to tweak volumes, filters, envelopes and other popular effects with such ease can give a new lease of life to your existing MIDI equipment. Why pay out for new synth modules when you can unlock the full potential of the ones you already own?

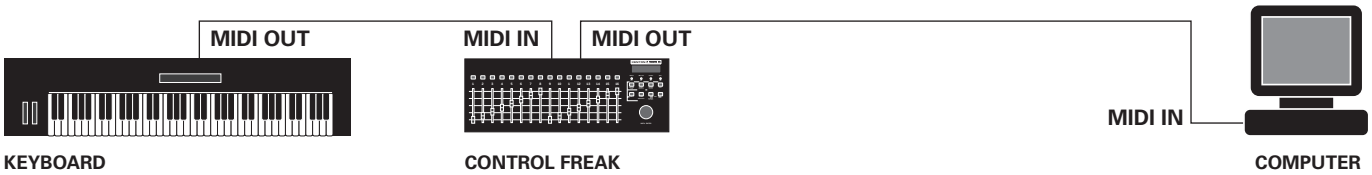
The Control Freak is also an excellent time saving device. One of the most popular uses for the unit is as a control surface for sequencers such as Cubase VST, Logic Audio and Cakewalk Pro-Audio, where the "virtual" faders on the computer screen can be manipulated from the Freak's real life faders. This removes the need for tedious adjustments by mouse, and is a far more intuitive way of working.

What is the principle of operation of the the Control Freak?

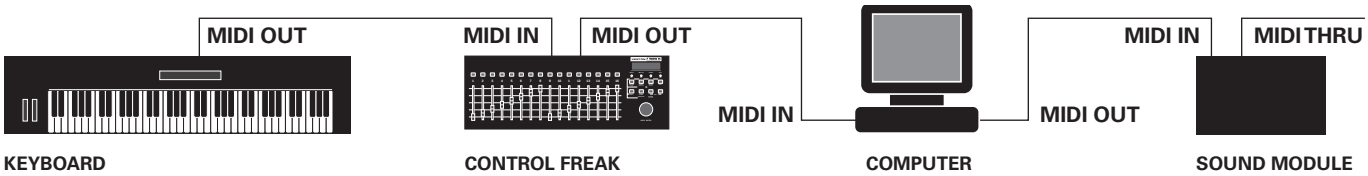
In most control strings used to control equipment be they SysEx messages, controller messages or NRPNs, there is usually one byte which actually specifies the new value required. With SysEx particularly, out of typically around 12 bytes, often only one or two bytes need to change to give you the change in sound. The rest are there to specify the manufacturer, the synth model, the type of message and which parameter you want to change. When setting up a slider or button to control something, most of the bytes have fixed values, when it comes to the variable data bytes, these can be set up to take their value from the slider.

Including the Control Freak in your MIDI setup - Connecting up

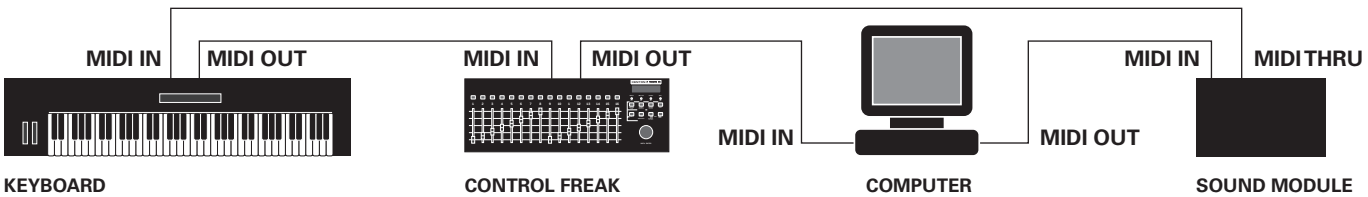
1) In the most basic setup where you have only a keyboard synthesiser, connect the MIDI out of the Freak to the MIDI in of your synthesiser.



2) In a simple setup, where you typically have a controlling keyboard connected to a computer which has a sound card installed, you would normally have the MIDI Out of your keyboard connected directly to the MIDI In of your computer. To include the Control Freak in your system, it should be inserted between the keyboard and the computer. To do this connect the MIDI Out of your keyboard to the MIDI In of the Control Freak and connect the MIDI Out of the Control Freak to the MIDI In of your computer. The Control Freak will automatically pass any MIDI it receives at its MIDI In, directly to its MIDI out. Whenever you move any sliders or press buttons on the Freak, this is merged with the MIDI from your keyboard and everything is sent directly to your computer.



3) If you have a sound module or other synthesisers connected to the MIDI Out of your soundcard, this is not a problem as your sequencer will also pass MIDI received at its MIDI In to its MIDI Out. This is often known as Soft-Thru.



4) Whatever your setup, if you want to record the slider and button movements you make on the Freak into your sequencer, you need the MIDI Out of the Freak connected to the MIDI In of your computer, and whatever you might previously have had plugged into the MIDI In of the computer, must now be plugged into the MIDI In of the Freak.

If you are using a keyboard that has its own internal sounds, you may want to use them in your setup too.

In more advanced setups, there may be situations where Thru boxes, merge boxes and switch boxes will be useful. Kenton maintain a stock of these items, together with a full range of MIDI leads.

Getting Started

In order to get you started, many of the programs on the Freak have been pre-loaded at the factory with useful features. For example program 0 contains a profile which is set up so the sliders control MIDI volumes on channels 1 –16 (1 – 8 for Original Control Freak) and the buttons control Mutes.

Connect the power supply to the Freak – if you are in the UK one will have been supplied, otherwise you must source one locally. For details see specifications on page 54.

Turn the power switch on, and the Control Freak will be in **PLAY MODE**. Probably the best way to familiarise yourself with the Freak is to experiment with some of the pre-loaded factory profiles. Turn the **Data Wheel** to see some of the profiles which are available.

If you are using a computer and a sequencer, load a song on the sequencer and start it playing. Select program 0 then move the sliders on the Freak and you will be able to change the volumes of the tracks on the sequencer. The buttons have also been set up so they alternately mute channels and then bring them back to the volume set on the slider. The Freak STUDIO EDITION has a slider for each MIDI channel, and the 8 slider Freak has volumes for MIDI channels 1 to 8 on Program #0 and volumes for MIDI channels 9 to 16 on program #1. To select program #1, turn the **Data Wheel** one click to the right.

You probably want to do more than just control the volumes of tracks. If you have a GS or XG compatible synth or soundcard, turn the datawheel a few clicks to the right, until you come to the program called GS/XG Stuff. This program has all the most useful things you need to start getting interesting sounds from your GS or XG equipment. For example slider 4 is set up to control Filter cutoff and Slider 5 resonance. With your sequencer still playing, move these sliders and hear the sound change. If this doesn't happen, it may be that the song you are using on your sequencer doesn't have anything playing on MIDI channel 1.

To change the channel the Freak is sending on, press the **[←]** button once (also labelled Global channel). The LCD screen should display:

```
Real-Time Edit
Global MIDI # 1
```

Turn the Data Wheel one click to the right so it says:

```
Global MIDI # 2
```

...and try the sliders again. Continue until you find a channel that works. Channel 10 is a good bet as it is usually the drum channel.

We will shortly be creating a simple demo sequence for download from the web. This will be a MIDI file which has parts that can be readily edited using the Freak.

Switching on

On power up, the LCD illuminates and displays the product name and current software version*1. After a couple of seconds the LCD changes to show:

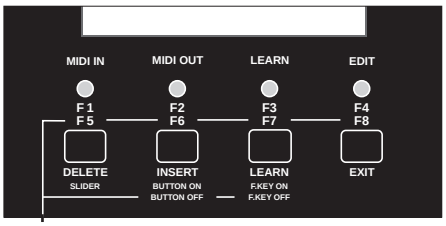
on the Freak **STUDIO EDITION**, or... on the Freak **ORIGINAL**

The Control freak is now in PLAY MODE

Introduction to controls in Play Mode

Sliders and buttons

Moving a slider or pressing a button will display its assigned name and its value as a decimal number on the LCD; the MIDI Out LED will also light up to show that it is sending data. The **Data Wheel** can be used to scroll through the 64 or 128 programs, and the 8 multi-function buttons beneath the LCD operate as follows.

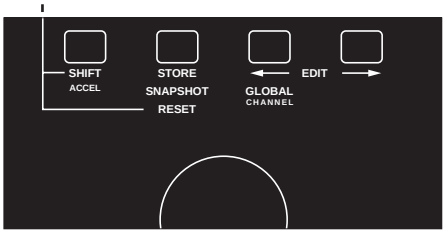


Function keys [F keys]

The four buttons directly below the LEDs can be used by themselves or in conjunction with the **[SHIFT]** button to act as programmable function keys and can generate any MIDI or internal data commands. Function key settings are Global and do not change with the programs. These **[F keys]** have been pre-programmed to provide Internal Clock controls and basic MMC control of sequencers and recording devices, however the factory defaults can be changed to provide other, user-defined operations.

The factory default settings for F keys are:

Without **[SHIFT]**	With **[SHIFT]**
1 Continue – internal clock	5 MMC Rewind
2 Stop – internal clock	6 MMC Stop
3 Start – internal clock	7 MMC Play
4 Internal prog – go to prog 0	8 MMC Fast Forward



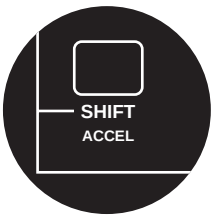
In various edit modes the function keys have other, preset uses. These however cannot be altered by the user.

Edit buttons

The four buttons above the **Data Wheel** are known as the Edit buttons but these are also useful in **PLAY MODE** as shown on the following page.

*1 You can program the Freak to display your own name here if you wish – see page 17.

What the edit buttons will do for you in PLAY MODE



What to expect from the [SHIFT] button in PLAY MODE

1 Pressing **[SHIFT]** will stop the sliders and buttons sending any MIDI until it has been released. This is useful if you want to know which slider or button does what. Press and hold **[SHIFT]**, then push one of the buttons above the sliders. This will display the button name on the top line of the LCD, and the slider name on the bottom line. The Button On name is displayed when pressed. The Button Off name, if used, is displayed when released.
Holding **[SHIFT]** also allows you to move a slider to a new position without actually sending any MIDI.

2 Pressing **[SHIFT]** while turning the **Data Wheel** will accelerate the **Data Wheel** by a factor of 4.

3 Pressing **[SHIFT]** and any of the 4 function keys below the LEDs will give you access to F keys 5-8.

4 Pressing **[SHIFT]** and **[STORE]** will send a 'Reset all controllers' message on the current Global MIDI Channel to MIDI Out. This is useful if you need to restore your synth to the original version of its present patch.

What to expect from the [STORE] button in PLAY MODE

Pressing **[STORE]** will send the current slider positions to MIDI Out as a **Snapshot**. This is great for sending data to a sequencer – just move the sliders until your synth sounds cool, press record on your sequencer and press **[STORE]** to send the snapshot.

What to expect from the left edit button [←] in PLAY MODE

Pressing the **[←]** button takes you into the Real-time menus. These all operate on MIDI in real-time. The **Data Wheel** can be used alone or with the **[SHIFT]** key to alter data. See the following page for an explanation of the Real-time menus.

What to expect from the right edit button [→] in PLAY MODE

Pressing the **[→]** button takes you into **EDIT MODE**. This causes the edit LED to light up and will disable MIDI transmission until you exit from **EDIT MODE**.

The first time you go to the edit menus after power up, the first page of the edit menu will show:

```
EDIT SLIDER NAME
Push → to Select
```

All subsequent visits to the edit menus will show whichever page you last accessed.

What to expect from the Data Wheel in PLAY MODE

In the home position, the **Data Wheel** is used to scroll through the available programs. The display will show the names of the profiles they contain.

For example:

```
  Program   0
Volumes Ch 1 -16
```

```
  Program   6
 GS/XG Stuff
```

Real-time Edit Each press of the **[←]** button displays a different real-time edit item. When any of these is displayed, the operation of the **Data Wheel** changes to control the displayed item.

```
 Real-Time Edit
Global MIDI #  1
```

The **Data Wheel** changes the Global MIDI channeL.

```
 Real-Time Edit
Int Clock 120bpm
```

The **Data Wheel** controls the tempo of the MIDI clock generator built into the Freak.

```
 Real-Time Edit
R/T Value #1=  0
```

The **Data Wheel** makes changes to R/T Value #1.

```
 Real-Time Edit
R/T Value #2=  0
```

The **Data Wheel** makes changes to R/T Value #2.

General When you move a slider or press a button, a value is displayed on the LCD in decimal format – this is suffixed with a d (for decimal).

When you are in edit mode and are required to enter data values, the Freak also displays a hexadecimal number (suffixed h) as well as its decimal equivalent. This is because values (particularly SysEx) are often given in hexadecimal format in the manufacturer's literature.

```
↑↑ Program  62
Vols Ch1-8 S1Ext
```

Ext. In #2
Ext. In #1

When the screen is showing the program name and number , you may see one or two up arrows in the top left corner. These indicate that the external inputs are active for this program.

Understanding how the Freak uses MIDI channels

In the simplest case you don't need to set any MIDI channel at all, in which case the Freak will use the Global MIDI channel (factory default #1) – You can change this at any time whilst you are in play mode by going to the Real-Time Edit section and using the **Data Wheel** to change the Global MIDI channel.

To change the Global MIDI channel from the real-time edit section, press the **[←]** key once and you will see the following screen:

```
Real-Time Edit
Global MIDI #  1
```

Use the **Data Wheel** to select the required Global MIDI channel. You can then press the **[→]** key to return to the home position or continue using the sliders/buttons whilst still in the real-time edit mode. This Global MIDI channel will remain in force until the unit is turned off. You can change the default for the Global MIDI channel from the Edit Global data section, see page 17.

There are however two other ways of setting a MIDI channel, both of which override the Global MIDI channel. The MIDI channel can be assigned at PROGRAM level in which case any data which would have used the Global MIDI channel, will use this channel instead.

```
EDIT PROGRM DATA
Push → to Select
```

```
MIDI Chan Assign
Use MIDI Chan  1
```

The default for this however is for it to be unassigned, in which case the display will show the

```
MIDI Chan Assign
Use GLOBAL Chan
```

The MIDI channel can also be assigned at slider/button level. This is the highest priority and will override any previous MIDI channel. At any point whilst editing slider/button data, where you are prompted to enter a COMMAND, you can force a MIDI channel.

```
     COMMAND
Set MIDI Chan  1
```

All MIDI from that point on will use the designated MIDI channel. You can even set a MIDI channel more than once within the same slider/button. See the example on page 29.

MIDI channels set in this way remain in effect until another Set MIDI Channel command is issued, or all the data for that particular slider/button has been sent. In other words a MIDI channel set in this way does not get carried over to another slider/button.

We recommend you familiarise yourself with the next section before you edit the Freak.
If however you are in a hurry to get going you can skip this section and jump straight to the examples on page 23.

Editing Overview

The main editing commands are grouped together here for easy reference.

Defining objects to edit

Moving a slider before entering edit mode will define it as the Slider to Edit. This also works with Push buttons but not the F keys.

The [←] and [→] edit buttons

The **[←]** and **[→]** buttons are used to navigate through the individual edit pages.

In any edit menu

Pressing **[STORE]** will save any edits at any time.

Slider 16/8 operates as a second **Data Wheel**.

Pressing **[SHIFT]** whilst moving the **Data Wheel** will accelerate the **Data Wheel** by a factor of four.

Pressing **[EXIT]** will return you to **PLAY MODE**. However, if you have made any changes and they have not been stored, you will see the prompt: `Lose changes?` You are then given the choice between **[SHIFT] + [EXIT]** or **[STORE]** options, or pressing the **[→]** button which will re-enter the relevant edit pages.

Holding **[SHIFT]** while pressing **[EXIT]** will leave edit mode at any time. All current edits will be lost, the user will not be prompted.

Editing names

Use the **Data Wheel** to select lower case letters and numbers.
Use **[SHIFT] + Data Wheel** to select capitals and symbols.

The top left two buttons act to delete characters and insert spaces.

The left and right arrow keys act as cursor buttons.

Pressing **[STORE]** while holding **[SHIFT]** jumps directly from edit name to edit data pages.

Editing data

When editing slider, button or function key data, pressing **[LEARN]** will activate the learn mode and light the Learn LED. The Freak will then be ready to receive incoming MIDI data. Pressing **[LEARN]** again will allow you to check or further edit the received MIDI data prior to **[STORE]**.

When editing MIDI data strings, the top left two buttons act to delete and insert bytes.

Clearing data

When editing slider, button or function key data, pressing **[LEARN]** twice acts as a quick way to erase unwanted MIDI strings prior to entering new data.

Abbreviations

PB = Push button (one of the buttons above the sliders)
FK = Function key
F key = Function key
RX = Receive
R/T = Real-time

When you have pressed the **[→]** key to take you into **EDIT MODE,** the **Data Wheel** can be used to scroll through the various edit menus. At present there are 21 edit menu options:

Menu	What you see on the top line of the Freak's screen	Description
1	EDIT SLIDER NAME	**16/8 sliders and 16/8 buttons**
2	EDIT SLIDER DATA	These items relate to editing the 16/8 sliders and 16/8 buttons directly above
3	EDIT PB ON NAME	them. The Button Off items are optional and will only be used if the button
4	EDIT PB ON DATA	state is set to on/off or toggle. The sliders and buttons, along with various
5	EDIT PB OFF NAME	other related data, are grouped together in a program (see below).
6	EDIT PB OFF DATA	
7	EdIT FK ON NAME	**F keys**
8	Edit FK ON DATA	These items relate to editing of the four buttons below the LEDs, which,
9	EDIT FK OFF NAME	together with the **[SHIFT]** button, gives you Function keys 1-8. Function key
10	EDIT FK OFF DATA	data is global, and does not change with the programs. Function key off items are optional as described above.
11	EDIT PROGRM NAME	**Programs**
12	EDIT PROGRM DATA	These items relate to program name, program MIDI channel, external input jacks 1 & 2, Realtime bytes 1 & 2 and Program Sysex Header for each program..
13	EDIT GLOBAL NAME	**Global settings**
14	EDIT GLOBAL DATA	The global data section relates to power on name, Global MIDI channel, internal Clock tempo, SysEx device number and Global SysEx Header.
15	SYSX RX PROG MAP	**Program mapping** Allows you to choose destination programs for incoming profiles. You may want to use this when you download profiles from our website.
16	SYSX DUMP SINGLE	**SysEx dump utilities**
17	SYSEX DUMP PROG	These allow you to dump individual sliders/buttons, programs or blocks of
18	SYSEX DUMP BLOCK	data. Note: a block = all sliders or all buttons, for example.
19	LIBRARY MODE	**Library mode** Currently unimplemented.
20	COPY SINGLE	**Copy utilities**
21	COPY PROGRAM	Copy single allows you to copy any slider/button to any other slider/button. Copy program allows you to copy any program to any other program location.

These 21 edit menu items are explained in more detail on the following pages.

PLAY MODE					EDIT MODE							
EDIT [←]	**EDIT [←]**	**EDIT [←]**	**EDIT [←]**	**[HOME]**	**EDIT [→]**	**EDIT [→]**	**EDIT [→]**	**EDIT [→]**	**EDIT [→]**	**EDIT [→]**	**EDIT [→]**	**EDIT [→]**
Realtime value #2	Realtime value #1	Internal Clock Speed	Global MIDI Channel	**Program number select**	**1 Edit Slider Name**	Slider to edit	Character #1	Character #2	Character #3	Character #4	Character #5	Character #6
					2 Edit Slider Data	Slider to edit	Minimum Level	Maximum Level	Byte #1	Byte #2	Byte #3	Byte #4
					3 Edit PB ON Name	Button to edit	Character #1	Character #2	Character #3	Character #4	Character #5	Character #6
					4 Edit PB ON Data	Button to edit	Minimum Level	Maximum Level	Button mode select	Byte #1	Byte #2	Byte #3
					5 Edit PB OFF Name	Button to edit	Character #1	Character #2	Character #3	Character #4	Character #5	Character #6
					6 Edit PB OFF Data	Button to edit	Minimum Level	Maximum Level	Button mode select	Byte #1	Byte #2	Byte #3
					7 Edit FK ON Name	Function key to edit	Character #1	Character #2	Character #3	Character #4	Character #5	Character #6
					8 Edit FK ON Data	Function key to edit	Minimum Level	Maximum Level	Button mode select	Byte #1	Byte #2	Byte #3
					9 Edit FK OFF Name	Function key to edit	Character #1	Character #2	Character #3	Character #4	Character #5	Character #6
					10 Edit FK OFF Data	Function key to edit	Minimum Level	Maximum Level	Button mode select	Byte #1	Byte #2	Byte #3
					11 Edit Program Name	Character #1	Character #2	Character #3	Character #4	Character #5	Character #6	Character #7
					12 Edit Program Data	MIDI Channel assign	Ext In #1 sldr/butn assign	Ext In #1 operating mode	Ext In #2 sldr/butn assign	Ext In #2 operating mode	Reatime value #1 minimum	Reatime value #1 maximum
					13 Edit Global Name	Character #1	Character #2	Character #3	Character #4	Character #5	Character #6	Character #7
					14 Edit Global Data	Global MIDI Channel	Internal clock rate BPM	Device Number	Sysex Byte #1	Sysex Byte #2	Sysex Byte #3	Sysex Byte #4
					15 Sysex Rx Program Map	Remap to Program ##						
					16 Sysex Dump Single	From Program ##	Slider, Button or F key ##	Push Store key to send				
					17 Sysex Dump Program	From Program ##	Push Store key to send					
					18 Sysex Dump Block	Block select	Push Store key to send					
					19 Library Mode	Library select	Library position ##	To Program ##	Slider Button or F key ##	Push Store key to copy		
					20 Copy Single	From Program ##	Slider, Button or F key ##	To Program ##	Slider Button or F key ##	Push Store key to copy		
					21 Copy Program	From Program ##	To Program ##	Push Store key to copy				

Editing overview

Editing is accomplished by use of the left and right arrow buttons and the **Data Wheel**.

On power up the Control Freak is at the HOME position, where use of the **Data Wheel** changes the program number.

The left arrow edit key takes you to four realtime positions, where use of the **Data Wheel** will adjust the Global MIDI channel, the Freak's clock speed or the Realtime value #1 or Realtime value #2. These are independent of the program selected. Realtime values 1 & 2 can be affected by min max + default values if they are set in Edit program data.

A single press of the right arrow edit key from the home position allows you to select the parameter you wish to edit, where use of the **Data Wheel** will scroll through the options.

Further presses of the right arrow edit key allow you to change the parameter indicated, again using the **Data Wheel** to change the value as desired.

When you have finished editing, press **[STORE]** then **[EXIT]** to return to the HOME position. You will be prompted to save your changes if you haven't already saved them.

EDIT MODE

EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]	EDIT [→]
Character #7	Character #8	Character #9	Character #10	Character #11	Character #12	Character #13	Character #14	Character #15	Character #16			
Byte #5	Byte #6	Byte #7	Byte #8	Byte #9	Byte #10	Byte #11	Byte #12	Byte #13	Byte #14	Byte #15	etc.	…until Byte #45
Character #7	Character #8	Character #9	Character #10	Character #11	Character #12	Character #13	Character #14	Character #15	Character #16			
Byte #4	Byte #5	Byte #6	Byte #7	Byte #8	Byte #9	Byte #10	Byte #11	Byte #12	Byte #13	Byte #14	etc.	…until Byte #45
Character #7	Character #8	Character #9	Character #10	Character #11	Character #12	Character #13	Character #14	Character #15	Character #16			
Byte #4	Byte #5	Byte #6	Byte #7	Byte #8	Byte #9	Byte #10	Byte #11	Byte #12	Byte #13	Byte #14	etc.	…until Byte #45
Character #7	Character #8	Character #9	Character #10	Character #11	Character #12	Character #13	Character #14	Character #15	Character #16			
Byte #4	Byte #5	Byte #6	Byte #7	Byte #8	Byte #9	Byte #10	Byte #11	Byte #12	Byte #13	Byte #14	etc.	…until Byte #45
Character #7	Character #8	Character #9	Character #10	Character #11	Character #12	Character #13	Character #14	Character #15	Character #16			
Byte #4	Byte #5	Byte #6	Byte #7	Byte #8	Byte #9	Byte #10	Byte #11	Byte #12	Byte #13	Byte #14	etc.	…until Byte #45
Character #8	Character #9	Character #10	Character #11	Character #12	Character #13	Character #14	Character #15	Character #16				
Realtime value #1 default	Reatime value #2 minimum	Reatime value #2 maximum	Realtime value #2 default	Sysex Byte #1	Sysex Byte #2	Sysex Byte #3	Sysex Byte #4	Sysex Byte #5	Sysex Byte #6	Sysex Byte #7	Sysex Byte #8	
Character #8	Character #9	Character #10	Character #11	Character #12	Character #13	Character #14	Character #15	Character #16				
Sysex Byte #5	Sysex Byte #6	Sysex Byte #7	Sysex Byte #8									

Slider operations

EDIT MENU 1 **Slider name**

This allows you to enter a name of up to 16 characters for each individual slider.
Pressing the **[→]** button will move to the next screen.

This allows you to choose the slider to edit by turning the **Data Wheel.** If the required slider is moved before entering the editing pages, it will automatically become the initial slider to edit.
Pressing the **[→]** button will move to the next screen.

Use the **Data Wheel** to select lower case letters and numerals.
Use **[SHIFT]** + **Data Wheel** to select capital letters and symbols.
The two top left buttons act to delete characters and insert spaces respectively.The **[←]** and **[→]** buttons work as cursor controls.

Normal: abcdefghijklmnopqrstuvwxyz0123456789
With **[SHIFT]** pressed: ABCDEFGHIJKLMNOPQRSTUVWXYZ*/+-.,&@!':

Press **[STORE]** to save the name entered then **[EXIT]** to return to **PLAY MODE or use [SHIFT]/[STORE]** to jump to

EDIT MENU 2 **Slider data**

Pressing the **[→]** button will take you into the slider edit pages.

This allows you to select the slider to edit by turning the **Data Wheel**.
Pressing the **[→]** button will move to the next screen:

This is the minimum value setting for the slider. The **Data Wheel** can be used to change the value if required.
Pressing the **[→]** button will move to the next screen:

This is the maximum value setting for the slider. The **Data Wheel** can be used to change the value if required.
Pressing the **[→]** button will move to the next screen:

The initial end of data setting can be changed using using the **Data Wheel** to any of the commands listed on page 21. There is space for up to 45 command or data bytes to be entered here.

Button operations

EDIT MENU 3

Button On name

This allows you to enter a name of up to 16 characters for each Button On. Button On names are edited in the same way as slider names.
Press **[STORE]** then **[EXIT]** or use [**SHIFT]/[STORE]** to jump to

EDIT MENU 4

Button On data

Pressing the **[→]** button will take you into the Button On edit pages:

This allows you to select the button to edit by turning the **Data Wheel**. If the required button is pressed before entering the editing pages, it will automatically become the initial button to edit.
Pressing the **[→]** button will move to the next screen:

This is the minimum value setting for the button. The **Data Wheel** can be used to change the value if required.
Pressing the **[→]** button will move to the next screen:

This is the maximum value setting for the button. The **Data Wheel** can be used to change the value if required.
Pressing the **[→]** button will move to the next screen:

Use the **Data Wheel** to select the button mode you require:

Sends the PB ON DATA message only when the button is pressed.

Sends the PB ON DATA message when the button is pressed and the PB OFF DATA message when the button is released.

Sends the PB ON DATA message and PB OFF DATA message on alternate button presses.
Pressing the **[→]** button will move to the next screen:

The initial end of data setting can be changed with the **Data Wheel** to any of the commands listed on page 21. There is space for up to 45 command or data bytes to be entered here.

EDIT MENU 5

Button Off name

This is edited in exactly the same way as Button On name.

EDIT MENU 6

Button Off data

Pressing the **[→]** button will take you into the Button Off edit pages, which are edited in exactly the same way as the Button On edit pages.

F key operations

EDIT MENU 7

F key On name

```
EDIT FK  ON NAME
Push → to Select
```

This allows you to enter a name of up to 16 characters for each F key On. F key On names are edited in the same way as slider names.
Press **[STORE]** then **[EXIT]** or use [**SHIFT]/[STORE]** to jump to

EDIT MENU 8

F key On data

```
EDIT FK  ON DATA
Push → to Select
```

Pressing the **[→]** button will take you into the F key On edit pages:

```
Func Key to Edit
Functn Key #   1
```

This allows you to select the F key to edit by turning the **Data Wheel**.

Pressing the **[→]** button will move to the next screen:

```
  Header  Data
Minimum Levl   0
```

This is the minimum value setting for the F key. The **Data Wheel** can be used to change the value if required.
Pressing the **[→]** button will move to the next screen:

```
  Header  Data
Maximum Levl 127
```

This is the maximum value setting for the F key. The **Data Wheel** can be used to change the value if required.
Pressing the **[→]** button will move to the next screen:

```
  Header  Data
Button ON only
```

Use the **Data Wheel** to select the F key mode you require:

Sends the FK ON DATA message only when the F key is pressed.

```
Button On + Off
```

Sends the FK ON DATA message when the F key is pressed and the FK OFF DATA message when the F key is released.

```
Button Toggle
```

Sends the FK ON DATA message and FK OFF DATA message on alternate F key presses.
Pressing the **[→]** button will move to the next screen:

```
   COMMAND
End of DATA
```

The end of data setting can be changed with the **Data Wheel** to any of the commands listed on page 21. There is space for up to 45 command or data bytes to be entered here.

EDIT MENU 9

F key off name

```
EDIT FK OFF NAME
Push → to Select
```

This is edited in exactly the same way as F key On name.

EDIT MENU 10

F key Off data

```
EDIT FK OFF DATA
Push → to Select
```

Pressing the **[→]** button will take you into the F key Off edit pages, which are edited in exactly the same way as the F key On edit pages.

Program operations

EDIT MENU 11

```
EDIT PROGRM NAME
Push → to Select
```

Program name

This allows you to enter a name of up to 16 characters for each individual program. Program names are edited in the same way as slider names.
Press **[STORE]** then **[EXIT]** or use **[SHIFT]/[STORE]** to jump to

EDIT MENU 12

```
EDIT PROGRM DATA
Push → to Select
```

Program data

Pressing the **[→]** button will take you into the program edit pages:

```
MIDI Chan Assign
Use Global Chan
```

This allows the user to define a MIDI channel for this program. Any data which would have used the Global MIDI channel will use this channel instead. See page 8 for a detailed look at channel assignments.
Pressing the **[→]** button will move to the next screen:

```
Ext Contrl In #1
   Unassigned
```

This allows you to assign Ext #1 input signals received at the ¼" jack socket to control any slider, button or F key. Use the Data Wheel to select a slider, button or F key destination for the signal from the Ext #1 input.
Press the **[→]** key to move to the next screen.

```
Ext Contrl In #1
Input = Voltage
```

If a slider was selected at the previous screen, you now have the choice of:

or

```
Input = Pedal
```

Alternatively, if a button or F key was selected at the previous screen, you now have the choice of:

```
Sw On=BREAK (Hi)
```

or

```
Sw On=MAKE (low)
```

For a fuller explanation of using the external inputs see page 34.

```
Ext Contrl In #2
   Unassigned
```

This screen and the following one are identical in use to the Ext control In #1 section above but are for Ext contrl In #2
Press the **[→]** key to move to the next screen.

```
  R/T Value #1
Minimum Val=   0
```

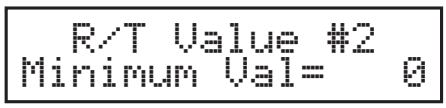
This allows the user to define a minimum value for real-time byte #1. When in real-time edit mode the data wheel will not let you select a value less than this.
Press the **[→]** key to move to the next screen.

```
  R/T Value #1
Maximum Val= 127
```

This allows the user to define a maximum value for real-time byte #1. When in real-time edit mode the data wheel won't let you select a value greater than this.
Press the **[→]** key to move to the next screen.

```
  R/T Value #1
Default Val= Off
```

This allows the user to define a default value for real-time byte #1. This is the value that real-time byte #1 will take up every time you select this program, however if Off is selected, the byte will just stay at the value it was when you last adjusted it, or another program set a default value there.
Note: to select Off, use the **[DELETE]** key (top left button).
Press the **[→]** key to move to the next screen.



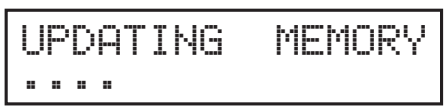
These three screens are identical in function to those above but apply to real-time byte #2.
Pressing the **[→]** button will move to the next screen:

Pressing the **[→]** button will move to the next screen:

Pressing the **[→]** button will move to the next screen:

This screen shows the first byte of an 8 byte Program SysEx header. You can enter a string of up to 8 bytes here, which can be used in any SysEx string within this program by selecting 'Progrm Sysex Hdr' from the data variables list.

When you are in play mode and the data is being sent, the bytes are read starting with byte #1 and will stop either when all 8 bytes have been sent or when an End of DATA is encountered.



Press **[STORE]** to save any edits
then **[EXIT]** to return to **PLAY MODE.**

Global operations

EDIT MENU 13

Global name

This allows you to enter a startup name. This name will be seen on power up, and is edited in the same way as slider names.
Press **[STORE]** then **[EXIT]** or use [**SHIFT]/[STORE]** to jump to

EDIT MENU 14

Global data

Pressing the **[→]** button will take you into the Global edit pages:

This allows you to set a default for the Global MIDI channel, which will be remembered on power off/on.*1
Pressing the **[→]** button will move to the next screen:

This allows you to set a default tempo which will be remembered on power off/on.*1
Pressing the **[→]** button will move to the next screen:

This allows you to set a SysEx device number for the Freak itself, and also to set the Control Freak STUDIO EDITION to accept Control Freak ORIGINAL profiles. You then use SysEx RX prog map to choose destination programs for incoming profiles.

Range: 1-16 and "Dev =8 ch Freak". You can usually leave this at 1 unless you want to load 8 channel Freak data, in which case turn the **Data Wheel** one click anticlockwise to show 'Dev=8 ch Freak'.
Then press **[STORE]** to activate.
Pressing the **[→]** button will move to the next screen:

Even numbered 8-channel profiles will go into the left sliders (1-8).
Odd numbered profiles will go into the right sliders (9-16).

This screen shows the first byte of an 8 byte Global SysEx header. You can enter a string of up to 8 bytes here, which can be used in any SysEx string within any program by selecting Global Sysex Header from the data variables list.

When you are in **PLAY MODE** and the data is being sent, the bytes are read starting with byte #1 and will stop either when all 8 bytes have been sent or when an End of DATA is encountered.

Press **[STORE]** to save any edits
then **[EXIT]** to return to **PLAY MODE.**

*1 The MIDI channel and Int clock tempo settings can be altered in real-time by accessing the Real-time edit menu **[←]**. Any changes made will be temporary and on power off/on will default to the settings made in Global data as described above.

RX/TX Dump operations

EDIT MENU 15 **SysEx RX prog map**

This allows you to redirect incoming profile dumps to any internal program.
Pressing the **[→]** button will move to the next screen:

The **Data Wheel** can be used to select a destination program for the incoming profile.Then just send the data to the Freak from your computer.
Note: this function is *only* active while this screen is displayed.

EDIT MENU 16 **SysEx dump single**

This allows you to dump individual slider, button or F key information to an external SysEx storage device.
Pressing the **[→]** button will move to the first screen:

The **Data Wheel** can be used to change the source program if required.
The bottom line of the display will show the name of the selected slider, button or F key.
Pressing the **[→]** button will move the cursor across to the right.

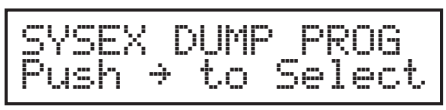
The initial setting sldr x can be changed with the top left three buttons in combination with **[SHIFT]** and the **Data Wheel** to select any slider, button or F key.

	F1	F2	F3
Without **[SHIFT]**	Slider	Button On	F key On
With **[SHIFT]**	Slider	Button Off	F key Off

The bottom line of the display will show the name of the selected slider, button or F key.
Pressing the **[→]** button will move to the next screen:

Pressing **[STORE]** will dump the chosen data from MIDI Out.

EDIT MENU 17 **SysEx dump prog**



This allows you to dump individual programs to an external storage device.
Pressing the **[→]** button will move to the next screen:

The **Data Wheel** can be used to select the required program.
Pressing the **[→]** button will move to the next screen:

Pressing **[STORE]** will dump the chosen data from MIDI Out.

EDIT MENU 18

SysEx dump block

This allows you to dump various different data blocks to an external storage device. Pressing the **[→]** button will move to the next screen:

The **Data Wheel** can be used to select from the available blocks.
For example Slider block dumps every slider in every program.

'All 64 Programs' sends a very large SysEx dump of the entire 64 programs to allow you to clone one Freak from another. Note however that you will need to clear the memory of the target Freak first. To do this, you need to turn it off, then turn it on again whilst you hold all four F keys pressed. Caution – this will wipe any profiles you have in the machine.
Pressing the **[→]** button will move to the next screen:

Pressing **[STORE]** will dump the chosen data from MIDI Out.

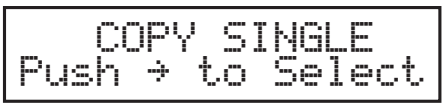
EDIT MENU 19

Library mode

Reserved for future use.

Copy operations

EDIT MENU 20

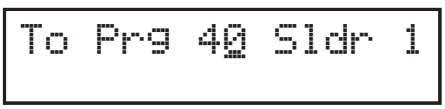


Copy single

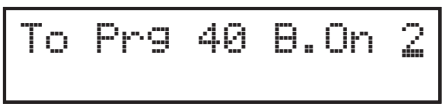
This allows you to copy any single slider, button or function key data to any other slider, button or function key destination.
Pressing the **[→]** button will move to the next screen:

The **Data Wheel** can be used to change the source program if required.
The bottom line shows the name of the slider, button or F key, not the program.
Pressing the **[→]** button will move the cursor across to the right.

The initial setting sldr x can be changed using the **Data Wheel** and the top left three buttons in combination with **[SHIFT]**.
Pressing the **[→]** button will move to the next screen:



The **Data Wheel** can be used to select a destination program.
Pressing the **[→]** button will move the cursor across to the right.



Select a destination slider/button/F key. The initial setting sldr x can be changed as previously described.
Pressing the **[→]** button will move to the next screen:

Pressing **[STORE]** will then copy the source data to the chosen destination.

EDIT MENU 21

Copy program

This allows you to copy any program to any other program location. Pressing the **[→]** button will move to the next screen:

The **Data Wheel** can be used to select the source program.
The bottom line of the display will show the name of the selected program.
Pressing the **[→]** button will move to the next screen:

The **Data Wheel** can be used to select a destination program.
The bottom line of the display will show the name of the selected program.
Pressing the **[→]** button will move to the next screen:

Pressing **[STORE]** will then copy the source program to the chosen destination.

Commands

When you see the

prompt you have the option of the following commands. The table below lists all currently available commands and their associated data bytes. As may be seen, the majority of MIDI and Internal Commands use a command followed by one or two variables.

MIDI Command	Data Byte # 1	Data Byte # 2
• End of data		
Controller	Controller number	Controller value
F0 Begin SysEx	"See ""SysEx Explained"""	
Note-on	Note number	Velocity
Note-off	Note number	Velocity
Program change	Program number	
NRPN MSB only	See below and NRPN/RPNs explained	
NRPN MSB + LSB		
RPN MSB only		
RPN MSB + MSB		
Pitchbend	Pitchbend LSB	Pitchbend MSB
Mono aftertouch	Aftertouch amount	
Poly aftertouch	Note number	Aftertouch amount
Controller Hi/Lo	Controller number	Controller value MSB/LSB
MTC Quarter frame	Message type/value	
F2 Song position pointer	SPP Data LSB	SPP Data MSB
F3 Song select	Song number	
• Internal program	Internal program number	
• Set global MIDI channel	Global MIDI channel	
• Set internal clock rate	Internal clock rate BPM	
• Set real-time value # 1	Real-time value # 1	
• Set real-time value # 2	Real-time value # 2	
• Internal clock start		
• Internal clock stop		
• Internal clock continue		
F6 Tune request		
F8 Timing clock		
FA MIDI Clock start		
FB MIDI Clock continue		
FC MIDI Clock stop		
FE Active sensing		
FF System reset		
Set MIDI channel 1		
↓		
Set MIDI channel 16		
• Invert data		

• = internal Freak data commands
F0-FF = MIDI system data commands
All others are MIDI channel commands

NRPNs/RPNs

Note that where MSB only and MSB + LSB are referred to in the command box below, this refers to the ***Data*** MSB or ***Data*** MSB + LSB. ***Parameter*** MSB and LSB must always be defined and sent. For further information see NRPNs explained on page 38.

Command	Parameter MSB	Parameter LSB	Data MSB	Data LSB
NRPN MSB only	NRPN Cntrllr MSB	NRPN Cntrllr LSB	NRPN Data MSB	
NRPN MSB + LSB	NRPN Cntrllr MSB	NRPN Cntrllr LSB	NRPN Data MSB	NRPN Data LSB
RPN MSB only	RPN Cntrllr MSB	RPN Cntrllr LSB	RPN Data MSB	
RPN MSB + LSB	RPN Cntrllr MSB	RPN Cntrllr LSB	RPN Data MSB	RPN Data LSB

Data variables

Most commands require some data to be entered at the subsequent screens. The following data variables can be used at any of these screens.
The table lists the data types in the order you see them as you turn the **Data Wheel** to the left (anti-clockwise).

Data Val 00h 0	Fixed numeric value of zero
Data from Slider	Data is taken from local slider
Use Data frm S01	Data is taken from another slider – in this case slider #1
	Other sliders S02 to S15 (Freak ORIGINAL only goes up to S08)
Use Data frm S16	Data is taken from slider #16
Use R/T Value #1	Byte definable by user in real-time
Use R/T Value #2	Byte definable by user in real-time
Data Val 7Fh 127	Fixed numeric value of 127 – note the hexadecimal value given too
	Fixed numeric values 126 to 2
Data Val 01h 1	Fixed numeric value of 1

Note that if you are entering a controller number at the screen following a controller command, the Freak gives you the standard names for the controllers as well as the numeric value.

There are some additional data variables which are only available when you are entering data following a SysEx command. These additional variables appear in-between some of the variables listed above and are as follows, again in the order you see them as you turn the **Data Wheel** to the left (anti-clockwise):

End Sysex [F7]	Obligatory end byte for SysEx message
Sx 14b signed LH	See SysEx explained page 40
Sx 14b signed HL	See SysEx explained page 40
Sx Ascii M/I/LSB	Format only used by Korg Wavestation (as far as we know)
Sx Data 0Hh/0Lh	See SysEx explained page 40
Sx Data 0Lh/0Hh	See SysEx explained page 40
Sx 4x4 nibl HhLl	See SysEx explained page 40
Sx 4x4 nibl lLhH	See SysEx explained page 40
Prog rm Sysex Hdr	String definable at program level
Global Sysex Hdr	String definable at global level
Checksum	Automatically calculates the checksum (mainly Roland).

Editing examples – a step by step guide

In all the following examples the naming of the slider/button has been omitted as it is common to all examples and is described fully on page 12.
Further editing examples will be made available on our website.

Example of making a slider control filter cutoff on the Global MIDI channel

In this case we are assuming that filter cutoff can be controlled with controller #74. This is by no means always the case, but the method holds good for any other controller message such as MIDI volume or pan etc. just by selecting a different controller number when prompted.

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the Data Wheel to the right (clockwise) to display:

```
EDIT SLIDER DATA
```

Press the **[→]** key and use the **Data Wheel** to select the Slider to Edit
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – the screen will display:

```
    COMMAND
 End of DATA
```

Turn the **Data Wheel** one click to the right to display:

```
   Controller
```

Press the **[→]** key – the screen will display:

```
Controller Numbr
0  Bank Select
```

Turn the **Data Wheel** to the right (clockwise) to display:

```
74 Sound Cntrl 5
```

You can speed up the process by holding the **[SHIFT]** key down whilst you turn the **Data Wheel**

Press the **[→]** key – the screen will display:

```
Controller Value
Data Val 00h   0
```

Turn the **Data Wheel** one click to the left to display:

```
Data from Slider
```

Press the **[→]** key – the screen will display:

```
    COMMAND
 End of DATA
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Press the **[EXIT]** key to go back to **PLAY MODE**

Example of making a button send note on & off on the Global MIDI channel – taking the note number from the local slider (the one directly beneath the button) at a fixed velocity of 64

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the **Data Wheel** to the right (clockwise) to display:

```
EDIT PB  ON DATA
```

Press the **[→]** key and use the **Data Wheel** to select the Button to Edit
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – the screen will show

```
  Header  Data
Button ON only
```

Turn the Data Wheel one click to the right – the screen will display:

```
Button On + Off
```

Press the **[→]** key – the screen will display:

```
   COMMAND
End of DATA
```

Turn the Data Wheel three clicks to the right – the screen will display:

```
   Note-On
```

Press the **[→]** key – the screen will display:

```
   Note Number
Data Val 00h   0
```

Turn the Data Wheel one click to to the left – the screen will display:

```
Data from Slider
```

Press the **[→]** key – the screen will display:

```
    Velocity
Data Val 00h   0
```

Turn the Data Wheel to the right (clockwise) until the screen displays:

```
Data Val 40h  64
```

You can speed up the process by holding the shift key down whilst you turn the **Data Wheel.**

Press the **[→]** key – the screen will display:

```
   COMMAND
End of DATA
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Press the **[EXIT]** key to go back to **PLAY MODE**

When you select Button On + off above, this tells the button to send a message not only when you press it, but when you release it too. The default mode is for the button to send a message only when pressed. There is a third mode which sends alternate messages each time you press the button, this is called toggle.

At this point you have set the button so that it turns a note on when you press it, but now you also need to make it send a note off when you release it, so continue as follows:

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the Data Wheel to the right (clockwise) until the display shows:

```
EDIT PB OFF DATA
```

Press the **[→]** key – Button to Edit should still be the correct one
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – the screen will show

```
  Header  Data
Button ON only
```

Turn the Data Wheel one click to the right – the screen will display:

```
Button On + Off
```

Press the **[→]** key – the screen will display:

```
   COMMAND
End of DATA
```

Turn the Data Wheel four clicks to the right – the screen will display:

```
Note-Off
```

Press the **[→]** key – the screen will display:

```
   Note Number
Data Val 00h   0
```

Turn the Data Wheel one click to to the left – the screen will display:

```
Data from Slider
```

Press the **[→]** key – the screen will display:

```
   Velocity
Data Val 00h   0
```

Turn the Data Wheel to the right (clockwise) until the screen shows:

```
Data Val 40h  64
```

Again, you can speed up the process by holding the shift key down whilst you turn the **Data Wheel**

Press the **[→]** key – the screen will display:

```
   COMMAND
End of DATA
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Press the **[EXIT]** key to go back to **PLAY MODE**

To use this example, first select a note number by using the slider – this is shown in the display. 60 is a good number to start with as this is middle C. Press the button and a note-on will be sent, release the button and a note-off will be sent.

In the above example you could easily have used a fixed value for the note and a variable value for the velocity instead.

Example of making a button send a chord on the Global MIDI channel – taking the velocity from the local silder (the one directly beneath the button)

For this example we will send middle C and the major third and fifth above it. These correspond to MIDI note numbers 60, 64 and 67

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the Data Wheel to the right (clockwise) until the display shows

```
EDIT PB  ON DATA
```

Press the **[→]** key and use the Data Wheel to select the Button to Edit
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – the display will show:

```
  Header  Data
Button ON only
```

Turn the Data Wheel one click to the right – the display will show:

```
Button On + Off
```

Press the **[→]** key – the display will show:

```
   COMMAND
 End of DATA
```

Turn the Data Wheel three clicks to the right and the display will show:

```
   Note-On
```

Press the **[→]** key – the display will show:

```
   Note Number
Data Val 00h   0
```

Turn the Data Wheel to the right (clockwise) until the display shows:

```
Data Val 3Ch  60
```

Press the **[→]** key – the display will show:

```
    Velocity
Data Val 00h   0
```

Turn the Data Wheel one click to the left to display:

```
Data from Slider
```

Press the **[→]** key – the display will show:

```
   COMMAND
 End of DATA
```

Turn the **Data Wheel** three clicks to the right to display:

```
   Note-On
```

Press the **[→]** key – the display will show:

```
   Note Number
Data Val 00h   0
```

Turn the **Data Wheel** to the right (clockwise) to display:

```
Data Val 40h  64
```

Press the **[→]** key – the display will show:

```
    Velocity
Data Val 00h   0
```

Turn the **Data Wheel** one click to to the left to show:

```
Data from Slider
```

Press the **[→]** key – the display will show:

```
   COMMAND
 End of DATA
```

Turn the **Data Wheel** three clicks to the right to display:

```
   Note-On
```

Press the **[→]** key – the display will show:

```
Note Number
Data Val 00h   0
```

Turn the Data Wheel to the right (clockwise) until the display shows:

```
Data Val 43h  67
```

Press the **[→]** key – the display will show:

```
Velocity
Data Val 00h   0
```

Turn the **Data Wheel** one click to to the left – the display will show:

```
Data from Slider
```

Press the **[→]** key – the display will show:

```
COMMAND
End of DATA
```

Press the **[STORE]** key to save your data
Press the **[EXIT]** key to go back to **PLAY MODE**

At this point you have set the button so that it turns the chord on when pressed.
To make it turn the notes off again when released continue as follows:

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
EDIT PB OFF DATA
```

Press the **[→]** key – Button to Edit should still be the correct one
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – the display will show:

```
Header  Data
Button ON only
```

Turn the **Data Wheel** one click to the right - the display will show:

```
Button On + Off
```

Press the **[→]** key – the display will show:

```
COMMAND
End of DATA
```

Turn the **Data Wheel** four clicks to the right - the display will show:

```
Note-Off
```

Press the **[→]** key – the display will show:

```
Note Number
Data Val 00h   0
```

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
Data Val 3Ch  60
```

Press the **[→]** key – the display will show;

```
Velocity
Data Val 00h   0
```

Turn the **Data Wheel** one click to to the left - the display will show:

```
Data from Slider
```

Press the **[→]** key – the display will show:

```
COMMAND
End of DATA
```

Turn the **Data Wheel** four clicks to the right - the display will show:

```
Note-Off
```

Press the **[→]** key – the display will show:

```
Note Number
Data Val 00h   0
```

Turn the **Data Wheel** to the right (clockwise) until the display shows:	Data Val 40h 64
Press the **[→]** key – the display will show:	Velocity Data Val 00h 0
Turn the **Data Wheel** one click to to the left - the display will show:	Data from Slider
Press the **[→]** key – the display will show:	COMMAND End of DATA
Turn the **Data Wheel** four clicks to the right and the display will show:	Note-Off
Press the **[→]** key – the display will show:	Note Number Data Val 00h 0
Turn the **Data Wheel** to the right (clockwise) until the display shows:	Data Val 43h 67
Press the **[→]** key – the display will show:	Velocity Data Val 00h 0
Turn the **Data Wheel** one click to to the left - the display will show:	Data from Slider
Press the **[→]** key – the display will show:	COMMAND End of DATA

Press the **[STORE]** key to save your data
Press the **[EXIT]** key to go back to **PLAY MODE**

To use this example, first select a velocity by using the slider – this is shown in the display. 64 is a good number to start with. Note that a velocity of zero is treated as a note-off so if you have the slider at the bottom of its travel, you will not hear anything. Press the button and a chord will be sent, release the button and the notes will be released.

Shortcut: after you have programmed the PB On data, you could simply copy the PB On data to the PB Off data for the same button and change the notes on to notes off.

Example of sending on multiple MIDI channels from one slider

How to make a slider send MIDI volume on two MIDI channels at the same time
In the example below, we're using MIDI Channels 1 & 2.

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
EDIT SLIDER DATA
```

Press the **[→]** key and use the Data Wheel to select the Slider to Edit
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – the display will show:

```
    COMMAND
End of DATA
```

Turn the **Data Wheel** to the left (anti-clockwise) until the display shows:

```
Set MIDI Chan 1
```

Press the **[→]** key – the display will again show:

```
    COMMAND
End of DATA
```

Turn the **Data Wheel** one click to the right and the display will show:

```
   Controller
```

Press the **[→]** key – the display will show:

```
Controller Numbr
0  Bank Select
```

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
7  Main Volume
```

Press the **[→]** key – the display will show:

```
Controller Value
Data Val 00h 0
```

Turn the **Data Wheel** one click to the left - the display will show:

```
Data from Slider
```

At this point you have set the slider to send MIDI volume (controller #7) on MIDI Channel #1.
Continue as follows to make it also send on Channel #2.

Press the **[→]** key – the display will show:

```
    COMMAND
End of DATA
```

Turn the **Data Wheel** to the left (anti-clockwise) until the display shows:

```
Set MIDI chan  2
```

Press the **[→]** key – the display will again show:

```
    COMMAND
End of DATA
```

Turn the **Data Wheel** one click to the right - the display will show:

```
   Controller
```

Press the **[→]** key – the display will show:

```
Controller Numbr
0  Bank Select
```

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
7  Main Volume
```

Press the **[→]** key – the display will show:

```
Controller Value
Data Val 00h   0
```

Turn the **Data Wheel** one click to the left and the display will show:

```
Data from Slider
```

Press the **[→]** key – the display will show

```
COMMAND
End of DATA
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Press the **[EXIT]** key to go back to **PLAY MODE**

As you can see from the above, you have to send the message twice. Setting the channel then sending the message, then setting a new channel and sending the message again.

You can send any combination of messages up to the maximum the Freak allows – that is until you can't press the **[→]** key any more (45 bytes).

NB: the 'Set MIDI Channel command' must come before a MIDI message if you want it to apply to that message. You can reset MIDI Channels as many times as you like in any slider / button.

Example of sending increasing MIDI volume message on one MIDI channel and sending decreasing MIDI volume message on another MIDI channel – all from one slider

How to make a slider send MIDI volume on two MIDI channels at the same time with one increasing whilst the other decreases. In other words, a pan from one channel to another.

In the example below, we're using MIDI Channels 1 & 2 – it is very similar to the previous example, but with the addition of an 'Invert Data' command.

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the **Data Wheel** to the right (clockwise) until the the display shows:

```
EDIT SLIDER DATA
```

Press the **[→]** key and use the Data Wheel to select the Slider to Edit
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – the display will show:

```
COMMAND
End of DATA
```

Turn the **Data Wheel** to the left (anti-clockwise) until the display shows:

```
Set MIDI Chan  1
```

Press the **[→]** key – the display will again show:

```
COMMAND
End of DATA
```

Turn the **Data Wheel** one click to the right - the display will show:

```
Controller
```

Press the **[→]** key – the display will show:

```
Controller Numbr
0  Bank Select
```

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
7  Main Volume
```

Press the **[→]** key – the display will show:

```
Controller Value
Data Val 00h   0
```

Turn the **Data Wheel** one click to the left - the display will show:

```
Data from Slider
```

At this point you have already set the slider to send MIDI volume (controller #7) on MIDI Channel #1. Continue as follows to make it also send inverted data on Channel #2.

Press the **[→]** key – the display will show:

```
   COMMAND
End of DATA
```

Turn the **Data Wheel** one click to the left - the display will show:

```
Invert Data
```

Press the **[→]** key – the display will show:

```
   COMMAND
End of DATA
```

Turn the **Data Wheel** to the left (anti-clockwise) until the display shows:

```
Set MIDI Chan  2
```

Press the **[→]** key – the display will again show:

```
   COMMAND
End of DATA
```

Turn the **Data Wheel** one click to the right - the display will show:

```
   Controller
```

Press the **[→]** key – the display will show:

```
Controller Numbr
0  Bank Select
```

Turn the **Data Wheel** to the right (clockwise) until the the display shows:

```
7  Main Volume
```

Press the **[→]** key – the display will show:

```
Controller Value
Data Val 00h   0
```

Turn the **Data Wheel** one click to the left and the display will show

```
Data from Slider
```

Press the **[→]** key – the display will show:

```
   COMMAND
End of DATA
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Press the **[EXIT]** key to go back to **PLAY MODE**

Example of programming a button to send a System Exclusive Message (SysEx)

For this example we are going to send a General MIDI System On message which comprises six bytes as follows: (in hexadecimal) F0h, 7Eh, 7Fh, 09h, 01h, F7h.

Press the **[→]** key from home position to start edit and display the edit menu options.

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
EDIT PB  ON DATA
```

Press the **[→]** key and use the **Data Wheel** to select the Button to Edit
Press the **[→]** key – leave Minimum Levl at 0
Press the **[→]** key – leave Maximum Levl at 127
Press the **[→]** key – leave Button ON only
Press the **[→]** key – the display will show:

```
    COMMAND
  End of DATA
```

Turn the **Data Wheel** two clicks to the right and the display will show:

```
Begin Sysex [F0]
```

Press the **[→]** key – the display will show:

```
Byte #  1  Sysex
Data Val 00h   0
```

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
Data Val 7Eh 126
```

You can speed up the process by holding the shift key down whilst you turn the **Data Wheel**.

Press the **[→]** key – the display will show:

```
Byte #  2  Sysex
Data Val 00h   0
```

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
Data Val 7Fh 127
```

Press the **[→]** key – for Byte #3

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
Data Val 09h   9
```

Press the **[→]** key – for Byte #4

Turn the **Data Wheel** to the right (clockwise) until the display shows:

```
Data Val 01h   1
```

Turn the **Data Wheel** two clicks to the left and the display will show:

```
 End Sysex  [F7]
```

Press the **[→]** key – the display will show:

```
    COMMAND
  End of DATA
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Press the **[EXIT]** key to go back to **PLAY MODE**

NB: the byte number indication on the top line begins with the byte after the F0h.

Example of programming a button to learn Note On/Off messages

Press the **[–>]** edit button to enter E**DIT MODE**

Use the Data Wheel to select…
Enter a suitable name as previously described

```
EDIT PB  ON NAME
Note On 36 C1
```

Press **[SHIFT] + [STORE]** to jump directly to PB on data.

```
Minimum Levl   0
```

Press the **[LEARN]** button

```
Learning
WAITING FOR DATA
```

Press and hold a key on your keyboard

```
    Note-On
```

Press the **[LEARN]** button again

```
    COMMAND
Set MIDI Chan 1
```

Press the **[<–]** button to view the button state*1

```
  Header  Data
Button ON only
```

Use the **Data Wheel** to select

```
Button On+OFF
```

Press the STORE button to save data.

```
UPDATING MEMORY
....
```

Press the **[EXIT]** button to return to **PLAY MODE**

Press the **[–>]** edit button to enter **EDIT MODE**

```
EDIT PB  ON DATA
```

Use the **Data Wheel** to select*2

```
EDIT PB OFF NAME
```

Enter a suitable name

```
 Note Off 36 C1
```

Press and hold a key on your keyboard
Press **[LEARN]**

```
    Learning
Waiting for data
```

Release the key on your MIDI controller

```
    Note-Off
```

Press **[LEARN]** again

```
    COMMAND
Set MIDI Chan  1
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Press the **[EXIT]** key to go back to **PLAY MODE**

*1 Learn will always insert a 'Set MIDI Chan' command for the MIDI channel the data was received on. If you want to use the Global channel at this point, press the **[DELETE]** key.

*2 You could use 'Copy Single' instead. (See page 19). Copy the PB On to PB Off for the same button. Then edit the PB Off data and change the Note On command to Note Off.

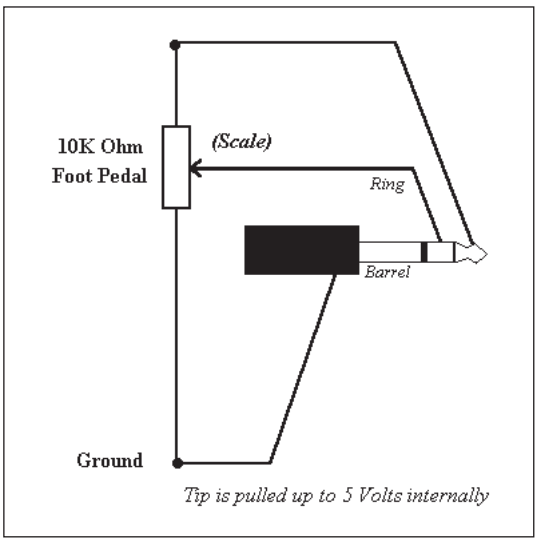
Note: some keyboards/controllers send a 'Note On' message with a Velocity of 0 instead of a 'Note Off' message. They are functionally identical but the Control Freak will of course show a 'Note On' on these cases, instead of the 'Note Off' in the example above.

Using the external inputs

Each external input can be used to connect either a variable or switch type foot pedal to replace the operation of a slider, button or F key. The external input settings are selected in 'Edit program data' – see page 15 for details of how to assign an input.

When an external input is configured to operate as a slider, the input replaces the chosen slider (which is then rendered non-operational, although only in the chosen program). When an input is configured to work as a button it works in parallel with the button, so either can be used.

When a slider is selected, the tip of the jack is pulled up internally to +5v (through about 470 ohms) and the ring of the jack becomes the variable input. The main body of the jack is the ground connection.



When a button or function key is selected you can use a mono jack plug since the ring connection is ignored and the switching is detected by the tip, which is pulled up internally to +5v (through about 5k ohms).

When either of the external inputs is assigned to a particular program, two arrows in the top left hand corner of the LCD denote which is active; for example:

on the Control Freak STUDIO EDITION or

on the Control Freak ORIGINAL

When both external inputs are set to replace the same slider, Ext #1 takes precedence.

Depending on whether you select a slider or a button/function key, you are given different options in the Edit Program Data pages. If a slider is selected, you are given the option of 'Input = Pedal' or 'Input = Voltage'. 'Pedal' should be used if you are using a variable foot pedal (as in the example below), and 'Voltage' should be used if you are supplying an external voltage (eg: for CV-to-MIDI conversion).

If a button/function key is selected, you are given the option of 'Sw On = Make (low)' or 'Sw On = Break (Hi)'. This is because there are two ways in which switch pedals can work – either they make a connection when pressed, pulling the voltage down ('low'), or they break connection when pressed, pulling the voltage up ('Hi'). Alternatively you can supply an external switching voltage instead of the switching pedal. The voltages supplied must be zero = on / 5v = off – or vice versa depending on the setting above.

Using a variable pedal to control MIDI volume (using In #1)

Although this example uses MIDI volume (controller #7) , you could substitute another continuous controller (eg: controller #1 – modulation) or a different type of MIDI message entirely.

Remember, all you are actually doing is replacing the operation of a slider with a foot pedal, therefore anything you can do with a slider you can also do with a pedal.

Choose a slider to replace – eg Program #0 Slider #1 which already sends MIDI volume on Channel #1. This slider already does what you want – now you need to make the external variable pedal do it instead of the slider.

Select 'Edit Program Data' – screen reads:

```
EDIT PROGRM DATA
```

Press the **[–>]** button twice – screen reads:

```
Ext Contrl In #1
   Unassi9ned
```

Use the **Data Wheel** to select slider #1 – screen reads:

```
Slider Number  1
```

(the slider you are replacing with a variable pedal plugged into the In #1 jack socket)

Press the **[–>]** button to move to the next screen
Use the **Data Wheel** to select 'Input = Pedal' – screen reads:

```
Input = Pedal
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Note the pedal will now control MIDI volume on channel #1 and the slider will be inoperative whilst you are using Program 0. The pedal assignments only apply to a single program so if you want them to apply to several programs, you need to set them up in each one.

Using a switch pedal for control of volume mute and reset (using In #2)

Choose a button to replace – eg Program #0 Button #1 which already sends mute and reset on Channel #1
This button already does what you want – now you need to make the external pedal do it instead of the button.

Select 'Edit Program Data' – screen reads:

```
EDIT PROGRM DATA
```

Press the **[–>]** button four times – screen reads:

```
Ext Contrl In #2
   Unassi9ned
```

Use the **Data Wheel** to select button #1 (the button you are operating with a switch pedal plugged into the In #2 jack socket)

Press the **[–>]** button to move to the next screen
Use the **Data Wheel** to select:

```
Sw On=Break (Hi)
```

Press the **[STORE]** key to save your data

```
UPDATING MEMORY
```

Setting a button to act as a sustain pedal

This is most useful if this button is then assigned to an external switch pedal.

Select 'Edit PB On Name'– screen reads

```
EDIT PB ON NAME
```

Use the **Data Wheel** to select a button and give the button a name in the usual way (eg:'Sustain On')

Hold **[SHIFT]** and press **[STORE]** – this takes you directly to:

```
  Header Data
Minimum Level  0
```

Press the **[→]** button twice – screen reads:

```
 Header Data
 Button ON only
```

Use the **Data Wheel** to select 'Button On + Off'

```
Button On + Off
```

Press the **[→]** button – screen reads:

```
    Command
  End of Data
```

Use the **Data Wheel** to select:

```
   Controller
```

Press the **[→]** button – screen reads:

```
0  Bank Select
```

Use the **Data Wheel** to select:

```
64 Sustain Pedal
```

Press the **[→]** button – screen reads:

```
Controller Value
Data Val 00h   0
```

Use the **Data Wheel** to select:

```
Data Val 7Fh 127
```

Press the **[STORE]** key to save your data:

```
UPDATING MEMORY
```

Use the **Data Wheel** to select:

```
Edit PB Off Name
```

Use the **Data Wheel** to select the same button as before and edit the name in the usual way (eg:'Sustain Off')

Hold **[SHIFT]** and press **[STORE]** – this takes you directly to:

```
  Header  Data
Minimum Levl   0
```

Press the **[→]** button twice – screen reads

```
Header Data
Button ON only
```

Use the **Data Wheel** to select:

```
Button On + Off
```

Press the **[→]** button – screen reads:

```
    Command
  End of Data
```

Use the **Data Wheel** to select:

```
   Controller
```

Press the **[→]** button – screen reads:

```
0  Bank Select
```

Use the **Data Wheel** to select:

```
64 Sustain Pedal
```

Press the **[→]** button – screen reads
Leave this setting as it is

```
Controller Value
Data Val 00h   0
```

Press the **[STORE]** key to save your data

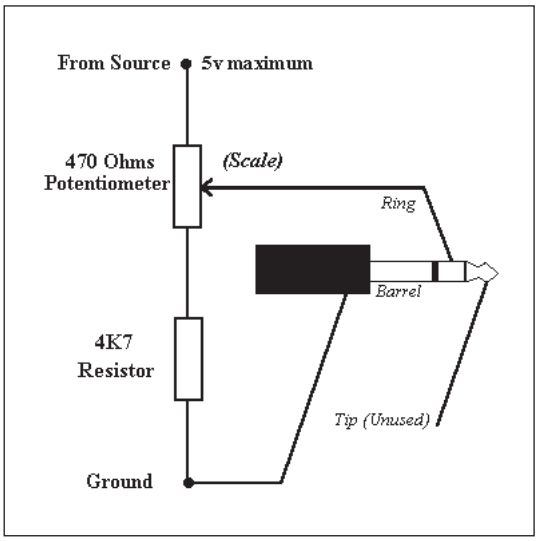
```
UPDATING MEMORY
```

CV-to-MIDI conversion

The external inputs may also be used as CV/Gate inputs in order to perform CV-to-MIDI conversion. As the set up procedure is rather extensive, Program 60 (125 for Control Freak ORIGINAL) has been pre-loaded with a CV-to-MIDI demo patch.

An example of how to program this will be made available on our website.

When the inputs are configured for CV-to-MIDI conversion, the following network is recommended in order to obtain satisfactory results:



To set up, first play bottom 'C' on your analogue monosynth and adjust it to play in tune. Next, play top 'C' two octaves up and adjust the trimmer (scaling) until this 'C' is also in tune. The synth will then accurately convert 1V/Oct CV into MIDI.

Notes

CV-to-MIDI conversion will not work using a Hz/V synth as the master.

Scaling – each individual slider may be scaled between 0 and 255, which is particularly useful when using the external inputs. However, where a slider's value is set above 127, the output MIDI message will not go higher than 127 (the data is truncated).

Playing style – to avoid hanging notes, care must be taken with the articulation of notes when playing.

Although the Control Freak sends an 'All notes off' message when each key is released in order to avoid this effect, it should be noted that some older synths (eg:Yamaha DX-7) do not respond to this standard MIDI message [123] but to message [127] instead.

IMPORTANT:When using the inputs for voltage to MIDI conversion, the input range must be restricted to the operational range of 0 to +5v DC.

Non-Registered Parameter Numbers (NRPNs) explained

It is worth mentioning that you really don't need to know how NRPNs actually work in order to program the Freak to send them. Because of the Freak's friendly editing system, all you need is the Non Registered Parameter Numbers given in the manufacturer's product literature, just enter these numbers when prompted and the Freak will do the rest. You might have the information printed in your manual or you may need to get it from the manufacturer's website.

The easiest way to explain how to go about programming NRPNs is with an example. For instance to change filter cutoff on a GS/XG instrument uses an NRPN, this is listed in the product literature as follows:

Non Registered Parameter Number for Filter Cutoff
NRPN MSB = 01
NRPN LSB = 08

Go into edit for the slider or button you want to program, (as described in the editing section of this manual).

Select NRPN MSB only from the command menu, then press the **[→]** key.

You will be prompted with:
Use the Data Wheel to select the MSB value given in the product literature - e.g. 1

```
NRPN Cntrllr MSB
Data Val 01h   1
```

Press the **[→]** key again and you will be prompted with:
Use the Data Wheel to select the LSB value given in the product literature - e.g. 8

```
NRPN Cntrllr LSB
Data Val 08h   8
```

press the **[→]** key again and you will be prompted with:
Use the **Data Wheel** to select:

```
NRPN Data MSB
data from slider
```

Press the **[→]** key again and then press **[store]** and then press **[exit]**

Actually, the above has already been done for you in one of the supplied profiles. If you want to see how this has been done, go to the program containing GS/XG stuff and look at slider #4 which is called GS Filter Cutoff. Go into edit slider data and keep pressing the **[→]** key while you look at the data presented in the display.

Sometimes the product literature gives the parameter numbers in hex (hexadecimal). Don't worry because the Freak shows both a decimal and a hexadecimal value in the display when you are in edit mode. The hexadecimal value has h after it, eg 0Eh is decimal 14, just use whichever is applicable.

For those of you who still want to know more, read on…
NRPNs use two numbers to specify the parameter that you wish to modify (eg filter cutoff) and then use data entry, which is controller #6 (or sometimes #6 & #38) to set the value for the specified parameter.

There are two commands on the Freak for NRPN, one is called NRPN MSB only and the other is called NRPN MSB+LSB. These names are possibly a bit misleading because what they refer to is how the data is sent, i.e. controller #6 only (data MSB) or both controller #6 (data MSB) & controller #38 (data LSB). The ***parameter*** MSB & LSB are always sent in both types of command.

NRPN MSB only is the one you would normally use, the MSB only part meaning it sends data values on controller #6 only. The other called NRPN MSB+LSB is for equipment which requires both controller #6 and controller #38 messages to be sent. However the product literature will tell you if you need to use this option.

Non-Registered Parameter Numbers, or NRPNs, are actually made up of multiple continuous controller messages. There are only 128 continuous controllers (0-127) of which the last few are reserved for system messages (eg #121 = reset all controllers). However by combining several controllers into an NRPN the potential number of parameters available for control is greatly increased.
Using an NRPN needs three or four ordinary continuous controller messages:

1) Controller #99 with a value of xxx where xxx is the NRPN MSB
2) Controller #98 with a value of yyy where yyy is the NRPN LSB

The above specifies the Non-Registered Parameter Number you want to modify. You then actually set the value of that NRPN by sending a value for controller #6 (data entry msb) or a combination of controllers #6 (data msb) and #38 (data lsb) as follows:

3) Controller #06 (data msb) with the value that you want the NRPN to be.
   (typically 'Data from slider' on the Freak)
4) Controller #38 (data lsb) - not required for most pieces of equipment
   (if used, probably also 'Data from slider')

If you assign both Data MSB and Data LSB to 'Data from slider', the Freak will automatically split the slider data it sends into MSB / LSB as required, it won't go and send the same value for both.

Notes
MSB = Most Significant Byte LSB = Least Significant Byte

Registered Parameter Numbers (RPNs) explained

These are essentially the same as NRPNs except that the parameters they control have been agreed upon by the MMA (MIDI Manufacturers Association) and JMSC (Japan MIDI Standards Committee). This means that they are the same for all products that respond to RPNs, although there are only a handful of RPNs which have been assigned at present:

Registered Parameter Numbers

MSB	LSB	Function
00	00	Pitchbend sensitivity
00	01	Fine tuning
00	02	Coarse tuning

Using an RPN needs three or four ordinary continuous controller messages as with the NRPN described above:

1) Controller #101 with a value of xxx where xxx is the RPN MSB
2) Controller #100 with a value of yyy where yyy is the RPN LSB
3) Controller #06 (data msb) - always required - usually assigned in the Freak to data from slider
4) Controller #38 (data lsb) - not required for most pieces of equipment.

System Exclusive (SysEx) Explained

SysEx messages contain data that is specific to one manufacturer and usually to one individual product. Each manufacturer wanting to use SysEx messages has been assigned a Manufacturer ID number. This is always the first byte after the begin SysEx byte [F0H], as in the example below:

F0H	Manufacturer ID	data	data	data ⟶	F7H
Begin SysEx					End SysEx

In some cases, the Manufacturer ID number may be three bytes instead of one. If the first byte after [F0] is 00H, then it is a three byte ID number – the two bytes following the 00H define the manufacturer. For example Yamaha is one byte: 43H, whereas Kenton Electronics is three bytes: 00H, 20H, 13H.

Apart from 'Begin SysEx' [F0H], ID number and 'End SysEx' [F7H] the format of the data in the rest of the message is entirely at the manufacturers discretion as long as they all have a zero in the most significant bit, i.e. all values must be between 0 and 127. Manufacturers are free to use the data in any way they see fit. The string below can be seen as a typical example:

F0H	Manufacturer ID	SysEx device no.	Command	Parameter	Data	F7H
Begin SysEx	XX	XX	XX	XX	XX	End SysEx

All SysEx bytes are usually represented in hex (hexadecimal). Don't worry because the Freak shows both a decimal and a hexadecimal value in the display when you are in edit mode. The hexadecimal value has h after it, eg 0Eh is decimal 14.

Three SysEx ID numbers have been set aside for special purposes:
7DH for non-commercial use. e.g. schools, research etc. where the product will not be released to the public.
7EH for Non Real-Time use.
7FH for Real Time use. e.g. MIDI machine control (MMC) MIDI show control (MSC)

For you to be able to program SysEx strings into the Freak to control any piece of equipment, you will definitely need the appropriate product literature for that equipment. This can often be found at the back of the product manual, or may be available either on request from the manufacturer or from their website. From then on it depends on the individual product literature how easy or difficult they have made it to program the Freak. In the ideal case, manufacturers give the whole string you need to enter, this can easily be entered into the Freak just as it is. Unfortunately, some manufacurers make it more difficult than it needs to be by splitting the information up so you have to collate the stuff you need from various parts of the product manual.

A typical example of a SysEx string is that which the Kenton Pro-2000 MIDI-CV converter uses to change the speed of LFO1:

F0H 00H 20H 13H 02H 20H 01H 05H data data F7H

...where the two data bytes in this case are actually an 8 bit value (0-255) split into two bytes of 4 bits each because this is the way the Pro-2000 wants it. The two bytes are entered on the Freak by entering Sx Data 0Lh/0Hh once only. In other words setting Sx Data 0Lh/0Hh automatically sends two bytes of data, the first containing the low four bits and the other the high. The Pro-2000 happens to want the bytes sent in this order. There is another similar function available on the Freak called Sx Data 0Hh/0Lh which sends the two bytes in the revcrse order, as some other manufacturers need them this way round instead.

Dealing with SysEx variable data bytes

There are several ways in which different manufacturers want their variable data sent. The Freak has got several data options available for dealing with these. When you are in edit mode entering SysEx data, you are offered several new choices for variable data bytes in addition to the options available for controller messages and the like. These are mentioned in the reference section but the following explains what these are, and where you would use them.

Displayed on LCD	No of bytes sent	What it is
Data from Slider	1	Data from local slider 0-127
Sx 4x4 nibl lLhH	4	16 bit value sent, split over 4 bytes. The 4 hi bits of each byte are zero and the lo 4 bits contain the data – sent least significant byte first – as follows: 0000 1111 , 0000LLLL, 0000hhhh, 0000HHHH
Sx 4x4 nibl HhLl	4	16 bit value sent, split over 4 bytes. The 4 hi bits of each byte are zero and the lo 4 bits contain the data – sent most significant byte first – as follows: 0000HHHH, 0000hhhh, 0000LLLL, 0000 1111
Sx Data 0Lh/0Hh	2	8 bit value sent, split over 2 bytes. The 4 hi bits of each byte are zero and the lo 4 bits contain the data – sent least significant byte first – as follows: 0000 1111, 0000HHHH
Sx Data 0Hh/0Lh	2	8 bit value sent, split over 2 bytes. The 4 hi bits of each byte are zero and the lo 4 bits contain the data – sent most significant byte first – as follows: 0000HHHH, 0000LLLL
Sx Ascii M/ I / Lsb	3	8 bit value sent, split over 3 bytes. eg: 123 sent as ascii for 1 then ascii for 2 then ascii for 3 sent most significant byte first. Unique to the Korg Wavestation as far as we know.
Sx 14b signed HL	2	14bit value sent, split over 2 bytes. The hi bit of each byte is zero and the lowest 7 bits contain the data – sent most significant byte first – as follows: 0HHHHHHH, 0LLLLLL
Sx 14b signed LH	2	14 bit value sent, split over 2 bytes. The hi bit of each byte is zero and the lowest 7 bits contain the data – sent least significant byte first – as follows: 0LLLLLL, 0HHHHHHH
Checksum	1	Automatically calculates the checksum required for a lot of the newer Roland stuff.

If you want to get the maximum available data range in the above multiple byte data formats, you will need to change the maximum value setting in the header data section of editing, from 127 to 255.

The **[Learn]** function of the Control Freak will recognise SysEx but please note the following:

When learning SysEx from Roland equipment which needs a checksum byte, the byte will be seen as a variable byte by the Freak and consequently will be shown as `data from slider`. You need to manually change this to `checksum`.

When learning SysEx from any equipment which needs data sent in any of the above multiple byte formats, each variable byte will be shown as `data from slider`. You will need to manually change the first one of these to the correct data type and you will have to delete one or more further bytes. Refer to the product manual for guidance on this.

What is the difference between profiles and programs?

A program is any one of the 64 memory locations on a Control Freak STUDIO EDITION (128 on Control Freak ORIGINAL) whereas a profile is a collection of pre-programmed names and data used for controlling a piece of MIDI equipment.

Put another way, profiles are what go into program locations. For example, the factory preset profile in program 0 is MIDI volumes/mutes, but it could be put into any of the available programs.

Downloading Profiles

How do I download profiles from the web and load them in?

Kenton are continually adding to the hundreds of profiles already on our website. These profiles are available for download free from our website at www.kenton.co.uk. We also welcome profiles submitted by our customers for the benefit of other users. You can email these to us at feedback@kenton.co.uk.

Click on the download button in the left hand frame. This brings up a page in the right hand frame offering various options. Click on Control Freak profiles. Follow the instructions on-screen to view the contents of the various profiles on offer or download them. You are presently offered the choice of downloading profiles one at a time, or downloading all the profiles in section as a .zip file. The .zip file contains the .mid files you require to load into your sequencer.

If you download the individual profiles, you will need to change the extension from .bin to .mid because they have been changed in order to stop your browser from trying to play them! You should be offered the choice of saving the .bin files and at the same time you should have the facility to rename them as they are being saved.

Use a MIDI lead to connect the MIDI output of your sequencer to the MIDI input of the Freak. Open your MIDI sequencer program ,then load the .mid file you want. Always start with an empty song otherwise you may end up with two or more files loaded.

Select RX prog map on the Freak (see page 18) and select the destination program. Hit the play Button On your sequencer – it's as simple as that.

16 Channel Freak loading 16 channel Freak profile

If you do not use Rx Prog Map, the profile will by default go into prog #63. Ensure the Device Number (in Edit Global Data) is set to #1.

16 Channel Freak loading 8 channel Freak profile

If you do not use Rx Prog Map, the profile will by default go into prog #62 left hand section – sliders/buttons 1-8.

Note: to load 8 channel Freak profiles into the 16 channel Freak you must set the Device Number (in Edit Global Data) to say 'Dev=8 channel Freak'. You must then save the change for it to become operational.

8 Channel Freak loading 8 channel Freak profile

If you do not use Rx Prog Map, the profile will by default go into prog #124.

It is not currently possible to load 16 channel Freak profiles into the 8 channel Freak.

Cubase VST users

Controlling Cubase VST from the Kenton Control Freak

Since 3.7 PC / 4.1 Mac, the ability to control various real-time parameters within VST has been available using a hardware remote device. In the current versions it is only possible to control the volumes, pans, mutes and solos on the audio mixer, plus transport functions – this is due to limitations of VST itself, not of the Freak. However, later versions of VST may offer a greater degree of control.

All new Control Freak units come pre-programmed with profiles to control the VST audio mixer. Control Freak owners with an earlier software revision than 4501.289 (4501.422 for Original Freak) will not have these profiles but can download them from our website www.kenton.co.uk, where there is also a profile for controlling transport.

Configuring VST to recognise the Control Freak

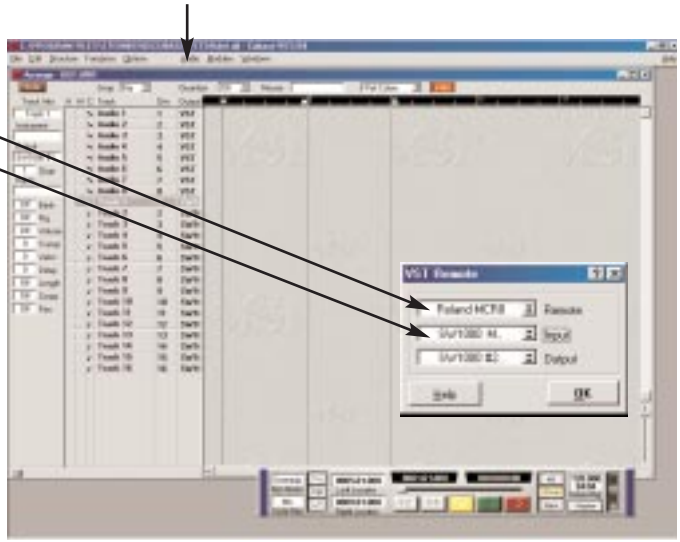
Firstly, it is essential that in VST you select the Roland MCR8 as your remote controller. Go to the audio drop down menu and select VST remote.

In the VST remote window select Roland MCR8.

Set input to be from your soundcard.

Note: Your Freak must be set to MIDI channel 16 or VST will not see it (Kenton profiles are already set to the correct channel).

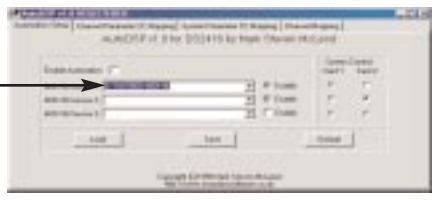
Plug-in instruments such as the Pro-Five and Model-E can also be controlled with the Control Freak. Profiles will be available from our website shortly.



Yamaha DSP Factory

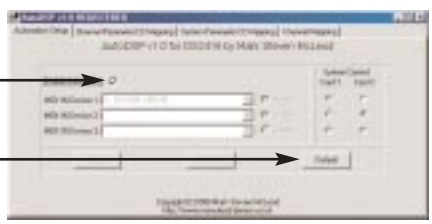
DSP Factory owners can also control the DSP mixing desk inside VST from the Control Freak. This is done by using a program called Auto DSP, which can be downloaded from www.msmcleod.demon.co.uk.

Before you start VST, run the Auto DSP program and select your MIDI In device.



Then click on enable automation.

Finally, click on default.



Start VST and you will find that the DSP mixer now responds to program 0 (volumes) on the Freak. All other parameters on the DSP Factory card can be controlled with the Freak – you just have to set them up how you want.

Cakewalk Pro Audio users

Cakewalk users can control the mixing desk from the Freak.

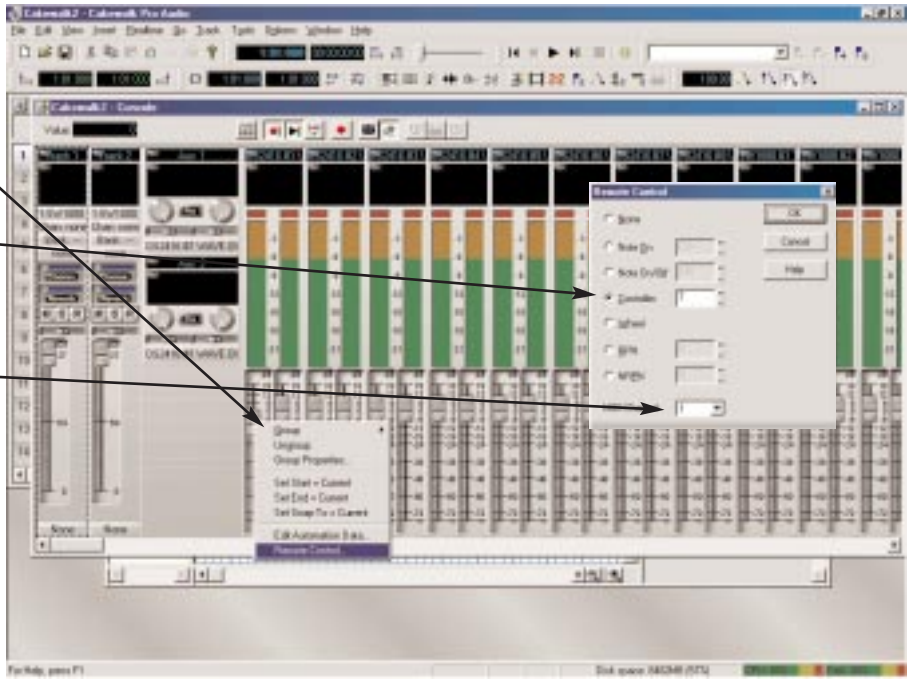
Just follow the simple procedure that follows:

Open up the mixing desk inside cakewalk and right click on slider 1; this will bring up a sub-menu.

Click on remote control. This opens up the remote control toolbox where you can set up your controller.

For now select controller and set it to number 7, this is volume control. Set the MIDI channel to channel 1. Press OK.

This has set up your slider. The slider in cakewalk will now respond to slider 1 on the Control Freak (program 0 volumes ch 1-16) for additional channels repeat procedure as above and increase the MIDI channel as you go: MIDI ch 2,3,4, etc.



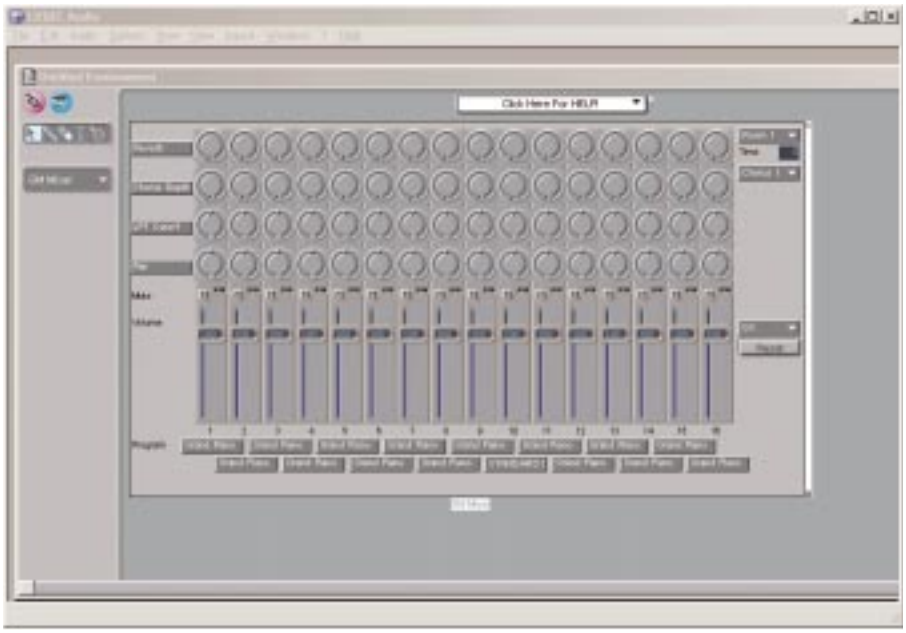
For people using Yamaha's DSP factory, the same procedure as above applies for setting up volumes on the audio mixing desk.

Logic Audio users

Logic Audio users can control the midi desk from the Kenton Control Freak.

Bring up the GM mixing desk inside Logic and the mixer is ready to go. This is done by using the Freak's factory default profile Program 0 volumes ch 1-16 for volume control and the Freak's factory default profile Program 1 pans ch 1-16) for pan controls.

For control on the chorus and reverb you can switch to other Freak factory defaultprograms where you can assign sliders to operate the reverb and chorus with controller data.

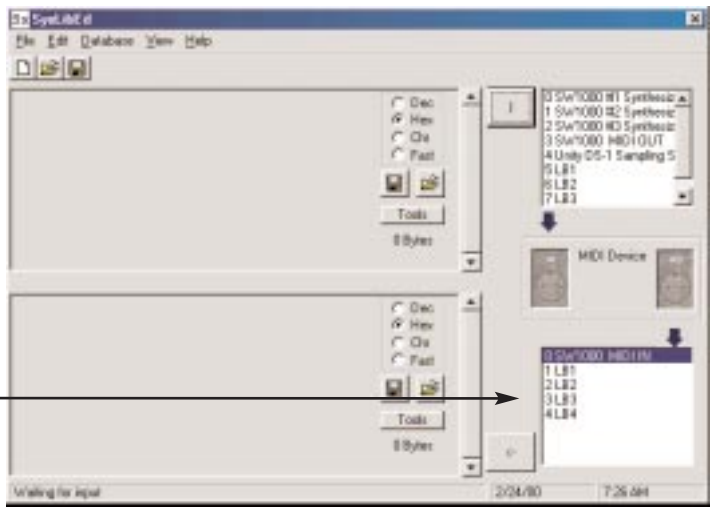


Yamaha dsp factory owners can use the audio mixer inside logic Audio in exactly the same way.

Saving a SysEx File

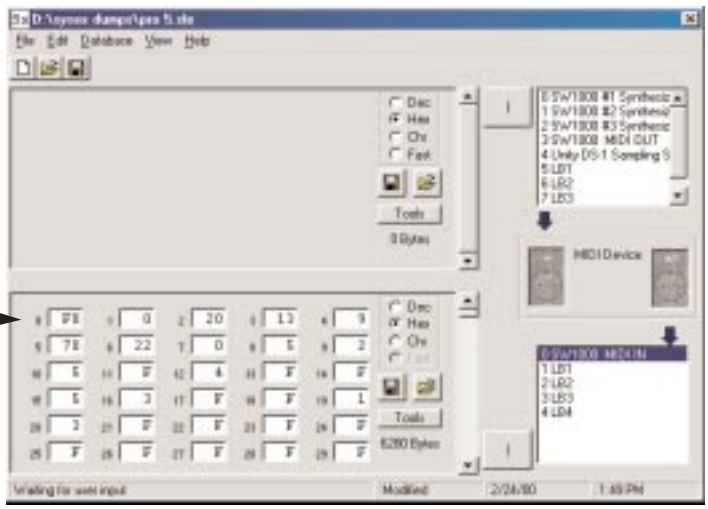
How to save a SysEx file without a sequencer

It is possible to save your profiles without using a sequencer. Instead you can use a dedicated SysEx editor.



Set the Freak to the type of SysEx dump you would like to do (single, program or block) before you press store.

Open up your SysEx editor and make sure the input is set to your sound card.



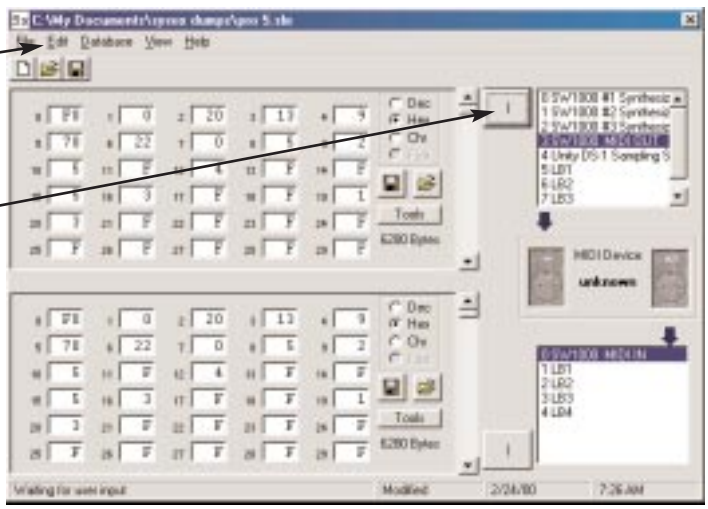
Press store on the Control Freak and you will see the data passed to the SysEx editor.

Save the set up. Your profile is now saved and can be restored to the Freak at any time.

To restore a Control Freak profile, load your saved SysEx file from Edit on the menu bar.

Copy in to out. Now select the output of your sound card and select send on your SysEx editor.

You will now see the green light on the Control Freak flash and then update its memory. The profile has been put back into the Freak.



The above screenshots are from the PC shareware product SysLibEd, which you can download from http://www.october28.com. You don't need SysEx knowledge to use it and it works well with the Freak.

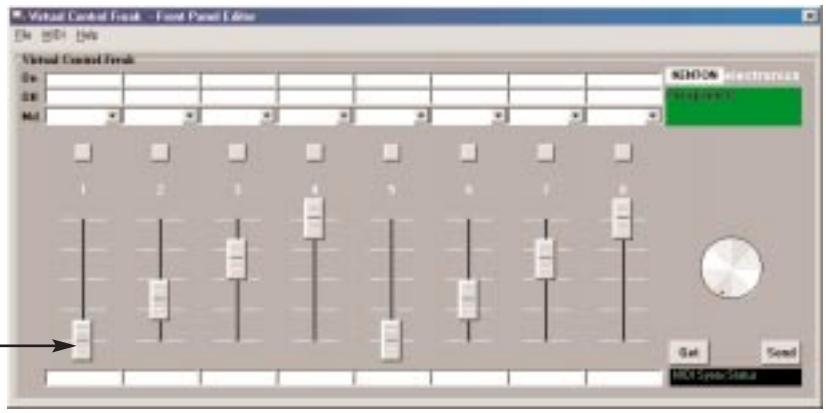
The Virtual Control Freak

The virtual Control Freak is a freeware program that allows the user to set up or edit a profile without using the Freak. This is useful as you can then save and edit till you are happy with it before you send it to your Freak.

At the moment it is only an 8 slider program but the 16 slider version will be released soon. You can download this program from:
http://www.dheard.dial.pipex.com

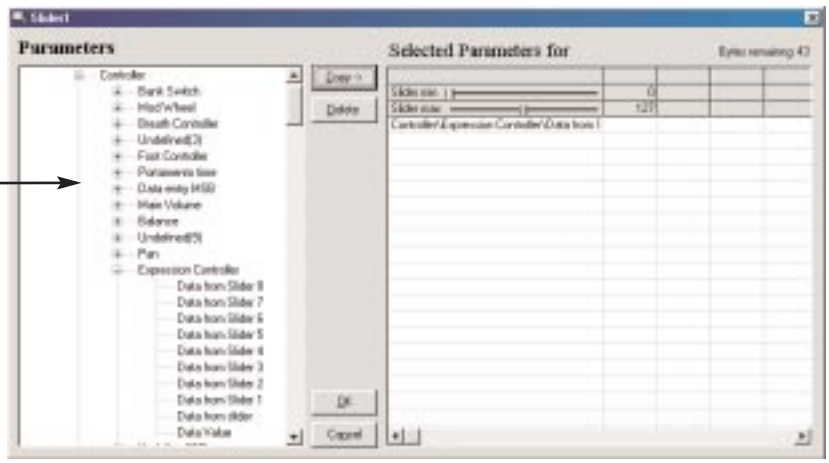
This program saves as a MIDI file so you use your sequencer as you did earlier in the manual for SysEx dumping and receiving to and from your sequencer.

This shows the startup screen. Select a slider or button.



A sub-menu window down drops where you can set up the parameters for that button or slider.

Read the 'readme' file that comes with the program for setting the program up to receive and send data to and from the Virtual Control Freak.



Details of the Control Freak STUDIO EDITION System Exclusive Specification

The first five bytes of the Freak's own SysEx messages are common to all data types.
N.B. the h suffix means hexadecimal

[1] 0F0h – Begin SysEx command
[2] 00h – Company ident first byte
[3] 20h – Company ident second byte
[4] 13h – Company ident third byte
[5] 09h – Product code – Control Freak Studio (16 slider)

64 byte Single data dump

F0h, 00h, 20h, 13h, 09h, 40h, prog_num , slider_num , data.....data, F7h

where prog_num is a (decimal) number between 0 and 127
where slider_num is as follows (decimal) :

0 – 15	Sliders 1 to 16
16 – 31	Button-on 1 to 16
32 – 47	Button-off 1 to 16
48 – 55	Function key-on 1 to 8
64 – 71	Function key-off 1 to 8
96 – 103	Program (all 8 values return the same program data)
112 – 119	Global (all 8 values return the same global data)
data...data	64 bytes of data sent low 4 bits, then high 4 bits, so 128 bytes are actually sent. data bytes will have their high 4 bits set to zero – i.e. 0000LLLL , 0000HHHH

n.b. values must always be sent even if not required – i.e. prog_num for function keys

64k Block data dump

F0h, 00h, 20h, 13h, 09h, 50h, block_num , data.....data , checksum_lo , checksum_hi , F7h

where block_num is as follows (decimal)

0	All Slider data (data from all 1024 sliders – 64 x 16)
1	All Button-on data
2	All Button-off data
3	All Program data/Funtion keys-on/off/Global
data...data	64k bytes of data sent low 4 bits, then high 4 bits, so 128k bytes are actually sent. Data bytes will have their high 4 bits set to zero – i.e. 0000LLLL , 0000HHHH
checksum_lo/hi	14 bit additive checksum sent as 2 bytes each having bit 7=0 so original data – 00HHHHHH LLLLLLL is sent as 0LLLLLLL, 0HHHHHHL

64k Library data dump

F0h, 00h, 20h, 13h, 09h, 60h, library_num , data.....data , checksum_lo , checksum_hi , F7h

as above except library_num can be 0 or 1 only to select library 1 or 2.

256k All 64 programs data dump – inc function keys & global (approx 293k sent)

F0h, 00h, 20h, 13h, 09h, 50h, 70h, data.....data , F7h

256k bytes of data are sent in a packed format where every 7 bytes of 8 bit data are sent as 8 bytes of 7 bit data.The 7 MSBs are sent in the first byte and the following 7 bytes are the original 7 bytes sent with their MSB set to zero. Can be used to clone Freaks.

3136 byte Complete Program dump

program data + 16 sliders +16 button-on +16 button-off (=49x64)

F0h, 00h, 20h, 13h, 09h, 70h, prog_num , data.....data , F7h

where prog_num is a (decimal) number between 0 and 127

data...data	3136 bytes of data sent low 4 bits, then high 4 bits, so 6272 bytes are actually sent. data bytes will have their high 4 bits set to zero – i.e. 0000LLLL , 0000HHHH

Request for 64 byte single data dump (also Button/Fkey etc.)

F0h, 00h, 20h, 13h, 09h, 10h , 40h, prog_num , slider_num , F7h

where prog_num is a (decimal) number between 0 and 127
where slider_num is as follows (decimal) :

0 – 15	Sliders 1 to 16
16 – 31	Button-on 1 to 16
32 – 47	Button-off 1 to 16
48 – 55	Function key-on 1 to 8
64 – 71	Function key-off 1 to 8
96 – 103	Program (all 8 values return the same program data)
112 – 119	Global (all 8 values return the same global data)

The Control Freak responds with the specified 64 bytes of data

For example: to request a dump of program 6 button-on 3 you would send the following:

F0h, 00h, 20h, 13h, 09h, 10h , 40h, 06h , 12h , F7h
Note that 12h is the hexadecimal equivalent of decimal 18 for selecting button-on 3
Request for 64k Block data dump

F0h, 00h, 20h, 13h, 09h, 10h , 50h, block_num , F7h

where block_num is as follows (decimal)

0	All Slider data (data from all 1024 sliders)
1	All Button-on data
2	All Button-off data
3	All Program data / Function keys-on/off / Global

The Control Freak responds with the specified 64k bytes of data

For example: to request a dump of all button-off data you would send the following:
F0h, 00h, 20h, 13h, 09h, 10h , 50h, 02h , F7h

Request for 64k Library data dump

F0h, 00h, 20h, 13h, 09h, 10h , 60h, library_num, F7h

as above except library_num can be 0 or 1 only to select library 1 or 2

Request for 256k all 64 programs data dump

F0h, 00h, 20h, 13h, 09h, 10h , 50h, 06h, F7h

The Control Freak responds with 256k bytes of data in a packed format where approx 293k bytes of data are actually transmitted. Note: to actually respond to this dump the memory of the Freak must be completely wiped first.To do this turn the Freak on with all four F keys pressed.

Request for 3136 byte Complete Program dump

prog data + 16 sliders + 16 button-on + 16 button-off (= 49x64)

F0h, 00h, 20h, 13h, 09h, 10h , 70h, prog_num, F7h

where prog_num is a (decimal) number between 0 and 127

The Control Freak responds with the specified 3136 bytes of data.

For example: to request a dump of program #33 you would send the following:
F0h, 00h, 20h, 13h, 09h, 10h , 70h, 21h , F7h

Note that 21h is the hexadecimal equivalent of decimal 33 for selecting the program

Details of the Control Freak ORIGINAL System Exclusive Specification

The first five bytes of the Freak's own SysEx messages are common to all data types.
N.B. the h suffix means hexadecimal

[1] 0F0h – Begin SysEx command
[2] 00h – Company ident first byte
[3] 20h – Company ident second byte
[4] 13h – Company ident third byte
[5] 08h – Product code – Control Freak Original (8 slider)

64 byte Single data dump

F0h, 00h, 20h, 13h, 08h, 40h, prog_num , slider_num , data.....data , F7h

where prog_num is a (decimal) number between 0 and 127
where slider_num is as follows (decimal) :

0 – 7	Sliders 1 to 8
16 – 23	Button-on 1 to 8
32 – 39	Button-off 1 to 8
48 – 55	Function key-on 1 to 8
64 – 71	Function key-off 1 to 8
96 – 103	Program (all 8 values return the same program data)
112 – 119	Global (all 8 values return the same global data)
data...data	64 bytes of data sent low 4 bits, then high 4 bits, so 128 bytes are actually sent. data bytes will have their high 4 bits set to zero – i.e. 0000LLLL , 0000HHHH

n.b. values must always be sent even if not required – i.e. prog_num for function keys

64k Block data dump

F0h, 00h, 20h, 13h, 08h, 50h, block_num , data.....data , checksum_lo , checksum_hi , F7h

where block_num is as follows (decimal)

0	All Slider data (data from all 1024 sliders – 128 x 8)
1	All Button-on data
2	All Button-off data
3	All Program data / Funtion keys-on/off / Global
data...data	64k bytes of data sent low 4 bits , then high 4 bits, so 128k bytes are actually sent. data bytes will have their high 4 bits set to zero – i.e. 0000LLLL , 0000HHHH
checksum_lo/hi	14 bit additive checksum sent as 2 bytes each having bit 7=0 so original data – 00HHHHHH LLLLLLL is sent as 0LLLLLLL, 0HHHHHHL

64k Library data dump

F0h, 00h, 20h, 13h, 08h, 60h, library_num , data.....data , checksum_lo , checksum_hi , F7h

as above except library_num can be 0 or 1 only to select library 1 or 2

256k All 128 programs data dump – inc function keys & global (approx 293k sent)

F0h, 00h, 20h, 13h, 08h, 50h, 70h , data.....data , F7h

256k bytes of data are sent in a packed format where every 7 bytes of 8 bit data are sent as 8 bytes of 7 bit data.The 7 MSBs are sent in the first byte and the following 7 bytes are the original 7 bytes sent with their MSB set to zero. Can be used to clone Freaks.
1600 byte Complete Program dump

program data + 8 sliders + 8 button-on + 8 button-off (=25x64)

F0h, 00h, 20h, 13h, 08h, 70h, prog_num , data.....data , F7h

where prog_num is a (decimal) number between 0 and 127

data...data	1600 bytes of data sent low 4 bits, then high 4 bits, so 3200 bytes are actually sent. data bytes will have their high 4 bits set to zero – i.e. 0000LLLL , 0000HHHH

Request for 64 byte single data dump (also Button/F key etc.)

F0h, 00h, 20h, 13h, 08h, 10h , 40h, prog_num , slider_num , F7h

where prog_num is a (decimal) number between 0 and 127
where slider_num is as follows (decimal) :

0 – 7	Sliders 1 to 8
16 – 23	Button-on 1 to 8
32 – 39	Button-off 1 to 8
48 – 55	Function key-on 1 to 8
64 – 71	Function key-off 1 to 8
96 – 103	Program (all 8 values return the same program data)
112 – 119	Global (all 8 values return the same global data)

The Control Freak responds with the specified 64 bytes of data

For example: to request a dump of program 6 button-on 3 you would send the following:

F0h, 00h, 20h, 13h, 08h, 10h , 40h, 06h , 12h , F7h

Note that 12h is the hexadecimal equivalent of decimal 18 for selecting button-on 3

Request for 64k Block data dump

F0h, 00h, 20h, 13h, 08h, 10h , 50h, block_num , F7h

where block_num is as follows (decimal)

0	All Slider data (data from all 1024 sliders)
1	All Button-on data
2	All Button-off data
3	All Program data / Funtion keys-on/off / Global

The Control Freak responds with the specified 64k bytes of data

For example: to request a dump of all button-off data you would send the following:
F0h, 00h, 20h, 13h, 08h, 10h , 50h, 02h , F7h

Request for 64k Library data dump

F0h, 00h, 20h, 13h, 08h, 10h , 60h, library_num , F7h

as above except library_num can be 0 or 1 only to select library 1 or 2

Request for 256k all 128 programs data dump

F0h, 00h, 20h, 13h, 08h, 10h , 50h, 06h, F7h

The Control Freak responds with 256k bytes of data in a packed format where approx 293k bytes of data are actually transmitted. Note to actually respond to this dump the memory of the Freak must be completely wiped first.To do this turn the Freak on with all four F.keys pressed.

Request for 1600 byte Complete Program dump

prog data + 8 sliders + 8 button-on – 8 button-off (=25x64)

F0h, 00h, 20h, 13h, 08h, 10h , 70h, prog_num, F7h

where prog_num is a (decimal) number between 0 and 127

The Control Freak responds with the specified 1600 bytes of data.

For example: to request a dump of program #33 you would send the following:
F0h, 00h, 20h, 13h, 08h, 10h , 70h, 21h , F7h

Note that 21h is the hexadecimal equivalent of decimal 33 for selecting the program

MIDI Analyser Mode

The Control Freak also has a MIDI analyser function. This feature allows you to see what types of MIDI messages are being transmitted by your master keyboard, or sequencer. In this way, the Control Freak may be used as a powerful diagnostic tool if you have a problem with your MIDI set-up.

To enter analyser mode, you must power on the Control Freak whilst holding the SHIFT button. The Firmware version will then be displayed until the shift key is released. The display will then show 'CHANnel', 'NOTE#' and 'VELOCity', and will display the relevant data, below the heading.

Using the **[F1]**, **[F2]**, **[F3]** and **[F4]** function keys, different types of received messages may be displayed:

[F1] Short press
CLock, STATUS and SYSEX data received
for checking MIDI clock.

[F2] Short press
CHANnel, PROG# and BANK# –
displays program change as received.

[F3] Short press
CHANnel, CONTroller# and VALUE
displays controller information.

[F3] Long press
CONTROLLER IDENTity –
displays the name and number of the received controller.

[F4] Short press
CHANnel,NOTE4# and VELOCity as above, note information.

* Although pitchbend and after-touch are not strictly controllers, when controller number is selected, they will also be displayed when received.

* If 'controller values' is selected, and pitchbend or after-touch are received, their values will also be displayed.

Overview of the MIDI message format

To make it easier to understand, MIDI data is usually represented in either decimal (base 10) or hexadecimal (base 16) numbers. On the Freaks, decimal is suffixed with [d]. Hex is suffixed with [h].

The MIDI standard uses 0-255 in the following way:

00 - 7F = 0-127 variable for data

	Channel Commands		Data 1 (0-127)	Data 2 (0-127)
80 - 8F =	Note-on	(channels 1-16)	Note number	Velocity
90 - 9F =	Note-off	(channels 1-16)	Note number	Velocity
A0 - AF =	Poly aftertouch	(channels 1-16)	Note number	Aft. amount
B0 - BF =	Controllers	(channels 1-16)	Controller number	Controller value
C0 - CF =	Programs	(channels 1-16)	Program number	-
D0 - DF =	Mono aftertouch	(channels 1-16)	Aftertouch amount	-
E0 - EF =	Pitchbend	(channels 1-16)	LSB	MSB

	System Common Commands		
F0 =	Begin System Exclusive*	Company ID	data... data... F7
F1 =	MIDI time code quarter frame	Type/Value	-
F2 =	Song position pointer	LSB	MSB
F3 =	Song select	Song number	-
F4 =	Undefined	-	-
F5 =	Undefined	-	-
F6 =	Tune request	-	-
F7 =	End System Exclusive*	-	-

	System Real-time Commands		
F8 =	Timing clock	-	-
F9 =	Undefined	-	-
FA =	Clock start	-	-
FB =	Clock continue	-	-
FC =	Clock stop	-	-
FD =	Undefined	-	-
FE =	Active sensing	-	-
FF =	System reset	-	-

* See System Exclusive explained on page 40.

Specifications

		Freak STUDIO EDITION	Freak ORIGINAL
User interface	Controls:	16 x 60mm sliders 16 x data push buttons 8 x editing push buttons Data entry wheel Power switch	8 x 60mm sliders 8 x data push buttons 8 x editing push buttons Data entry wheel Power switch
	Displays:	LCD 2x16 back-lit 4 LED indicators	LCD 2x16 back-lit 4 LED indicators
Connections	Inputs:	Power socket d.c. 2.1mm centre positive MIDI In, 5 pin DIN type Ext1, 1/4" stereo jack socket Ext2, 1/4" stereo jack socket	Power socket d.c. 2.1mm centre positive MIDI In, 5 pin DIN type Ext1, 1/4" Stereo jack socket Ext2, 1/4" Stereo jack socket
	Outputs:	MIDI Out,Thru 5 pin DIN type	MIDI Out,Thru 5 pin DIN type
Accessories		Operating manual Mains power adaptor (UK Only)	Operating manual Mains power adaptor (UK Only)
Physical data	Dimensions: Weight:	432mm x 163mm x 58mm 3Kg	269mm x 163mm x 58mm 2.5Kg

Power Supply 9V dc. @ 250mA
The Control Freak must be powered from a suitable power adaptor.
This should be 9 volts DC regulated or unregulated at 250mA or greater, terminated with a 2.1mm DC jack connected centre to positive.

UK models are supplied with a mains adaptor, but overseas customers must source locally.

Warranty The Control Freak comes with a 12 month back to base warranty (from purchase date).
Customer must arrange and pay for carriage to and from Kenton Electronics.

Support Control Freak profile updates for a wide range of devices are available from our website:
www.kenton.co.uk.

Comments Email your comments and ideas to: feedback@kenton.co.uk

Features

- 16 or 8 fully programmable sliders
- 16 or 8 fully programmable buttons
- 8 fully programmable function keys (4 x 2 with the shift button)
- 4 multi-function edit buttons
- Data entry wheel
- 2 line LCD screen
- MIDI In / Out / Thru (with merge function)
- 2 x 1/4" external input jacks (capable of CV-to-MIDI conversion)
- 64 user memories for STUDIO EDITION / 128 for ORIGINAL
- MIDI clock generator
- MIDI analyser mode
- Advanced learn mode
- Individual names for every slider, button, function key and program
- Sliders and buttons can send any MIDI message, including multiple and inverted messages.
- Buttons can function in three different ways – Button On, button toggle or Button On/Off
- SysEx dump of presets, blocks and individual sliders, buttons, etc. to external devices
- Full Roland checksum support
- Snapshot function – sends all current slider positions to MIDI Out
- Reset all controllers function
- Hundreds of presets are already available from www.kenton.co.uk

Further Information

Further information and additional examples will be made available on our website.

Contact

Kenton Electronics
Brookfarm House
Station Road
South Wimbledon
London SW19 2LP
UK

Tel: +44 (0)20 8544 9200
Fax: +44 (0)20 8544 9300

Written by

Paul Gregson
Dave Millichamp
Anthony Barden
Peter Herman
John Smout
John K. Price