Güralp DM24 MkIII

Operator's Guide

Document Number: MAN-D24-0004

Issue U – December, 2021

Designed and manufactured by
Güralp Systems Limited
3 Midas House, Calleva Park
Aldermaston RG7 8EA
England

Table of Contents

1 Preliminary Notes....8

1.1 Proprietary Notice....8
1.2 Cautions and Notes....8
1.3 Manuals and Software....8
1.4 Conventions....8

2 Introduction....9

2.1 Features....10

3 Installing the DM24....12

3.1 Power supplies....12
3.2 Connecting sensors....12
3.3 Connecting external hardware....13
3.3.1 FireWire disks....13
3.4 The DM24BH borehole digitiser....15

4 Using the DM24....16

4.1 Taps....16
4.1.1 Triggering....17
4.2 Streams....17
4.2.1 Digitisers for multiple instruments....18
4.3 Transmission modes....19
4.3.1 DIRECT....20
4.3.2 FILING....20
4.3.3 DUPLICATE....21
4.3.4 DUAL....22
4.3.5 FIFO (First In First Out)....22
4.3.6 Transmission mode summary....24
4.4 Buffering modes....24
4.4.1 RE-USE / RECYCLE....24
4.4.2 WRITE-ONCE....25

4.5 Downloading stored data....25
4.5.1 Downloading over FireWire....26
4.5.2 Reading DM24 disks....26
4.5.3 Downloading over the serial link....29
4.6 DM24 USB Host Operations....31
4.6.1 DM24 Mk3 USB host option....31
4.6.2 Formatting a USB device....32
4.6.3 Directory structure....32
4.6.4 System operation....33

5 Using Scream!....35

5.1 Configuring digitisers....35
5.1.1 System ID....36
5.1.2 Output control....37
5.1.3 Triggering....40
5.1.4 STA/LTA – Triggering by average values....41
5.1.5 Level triggering....46
5.1.6 External triggering....46
5.1.7 Pre-trigger and post-trigger recording....47
5.1.8 Mux Channels....47
5.1.9 Ports....49
5.2 Controlling digitisers....50
5.2.1 System....50
5.2.2 Triggering....51
5.2.3 Calibration....52
5.2.4 Mass Control....53
5.2.5 Data flow....54

6 Command line interface....56

6.1 Introduction....56
6.1.1 FORTH....56
6.2 Getting help....57
6.2.1 HELP....57
6.2.2 EXPLAIN....58
6.3 General configuration....58
6.3.1 SET-ID....58
6.3.2 SERIAL2....59
6.3.3 RESETLABELS....59
6.3.4 SENSOR-TYPE....59
6.3.5 AZIMUTH....59
6.3.6 CROSS....60
6.4 GPS, GNSS and timing....60
6.4.1 GPS-TYPE....60
6.4.2 HR-CYCLE....61

6.4.3 SET-CLOCK....61
6.4.4 TIME?....62
6.4.5 getWNROpivot....62
6.4.6 setWNROpivot....62
6.4.7 LEAPSECOND....62
6.4.8 SQPATCH....63
6.5 Port configuration....63
6.5.1 BAUD....63
6.5.2 STOPBITS....64
6.5.3 +MODEM....64
6.5.4 –MODEM....65
6.5.5 +GCFIN....65
6.5.6 –GCFIN....66
6.5.7 OPEN....66
6.5.8 USBPORT....66
6.6 Output configuration....67
6.6.1 SAMPLES/SEC....67
6.6.2 CONTINUOUS....67
6.6.3 SET-TAPS....68
6.6.4 COMPRESSION....68
6.6.5 LOWLATENCY....69
6.6.6 SETDSS....70
6.7 Variable Gain....71
6.7.1 *GAINS....71
6.7.2 *GAIN....71
6.8 Triggering....72
6.8.1 TRIGGERS....72
6.8.2 GTRIGGERS....72
6.8.3 S/WTRIGGER....72
6.8.4 TRIGGERED....72
6.8.5 STA....73
6.8.6 LTA....73
6.8.7 RATIOS....74
6.8.8 FRATIOS....74
6.8.9 BANDPASS....74
6.8.10 MICROG....75
6.8.11 HIGHPASS....75
6.8.12 PRE-TRIG....75
6.8.13 POST-TRIG....75
6.8.14 TRIGGERIN....76
6.8.15 TRIGGEROUT....76
6.8.16 POLARITY....76
6.9 Calibration....77
6.9.1 SINEWAVE....77
6.9.2 SQUAREWAVE....77
6.9.3 RANDOMCAL....78
6.9.4 MINUTE....78
6.9.5 %AMPLITUDE....79
6.9.6 AUTOCAL....79

6.9.7 AUTOCAL?....79
6.9.8 -AUTOCAL....80
6.10 Actions....80
6.10.1 LOCK....80
6.10.2 UNLOCK....80
6.10.3 CENTRE....80
6.10.4 MASSES?....81
6.10.5 %AUTO-CENTRE....81
6.10.6 O/SNULL....81
6.10.7 RE-SYNC....81
6.10.8 ?LITHIUM....82
6.10.9 RE-BOOT....82
6.11 Data storage and transmission....82
6.11.1 SHOW-FLASH....82
6.11.2 RESET-FLASH....83
6.11.3 ERASEFILE....83
6.11.4 EVENTSLIST....84
6.11.5 RESET-LIST....84
6.11.6 DIRECT....85
6.11.7 FILING....85
6.11.8 DUPLICATE....85
6.11.9 DUAL....85
6.11.10 ADAPTIVE....85
6.11.11 FIFO....86
6.11.12 HEARTBEAT....86
6.11.13 MS-GAP....86
6.11.14 RE-USE....87
6.11.15 WRITE-ONCE....87
6.11.16 MODE?....87
6.12 Downloading over the serial port....87
6.12.1 DOWNLOAD....87
6.12.2 ALL-FLASH....88
6.12.3 ALL-DATA....88
6.12.4 STREAM....89
6.12.5 STATUS-ONLY....89
6.12.6 S/S....89
6.12.7 ALL-TIMES....89
6.12.8 FROM-TIME....89
6.12.9 TO-TIME....90
6.12.10 EVENTS....90
6.12.11 GO....91
6.12.12 END-DOWNLOAD....91
6.13 Downloading over FireWire....91
6.13.1 DISKMENU....91
6.13.2 MBTRANSFER....92
6.13.3 DIR....94

7 Inside the DM24 95

7.1 State of health information 95
7.1.1 Heartbeat messages 95
7.1.2 Status blocks in continuous mode 97
7.2 GPS and Synchronisation 99
7.2.1 Overview 99
7.2.2 Leap-seconds 100
7.2.3 Week-number roll-over 100
7.2.4 Multiple re-synchronisation events 101
7.3 Updating firmware 103
7.3.1 Updating firmware using an EAM, Affinity or NAM 103
7.3.2 Updating firmware with Scream! 103
7.4 Setting up external triggering 106
7.4.1 Overview 107
7.4.2 Triggering your own equipment 108
7.4.3 Using an external trigger source 109
7.4.4 Triggering several Güralp instruments simultaneously 110
7.5 Technical details 112

8 Güralp Compressed Format 114

8.1 GCF blocks 114
8.1.1 The GCF header 114
8.1.2 The GCF body 116
8.2 Sending GCF streams over a network 116
8.2.1 Initiating a connection 117
8.2.2 Data packets 117
8.2.3 Recovering missing data 118
8.3 Sending GCF streams over a serial link 118
8.3.1 The transport layer 119
8.3.2 ACK and NACK signals 119

9 Connector pin-outs 121

9.1 SENSOR A and B ports 121
9.2 AUXILIARY port 122
9.3 GPS Port 123
9.4 DATA IN port 124
9.5 FIREWIRE port 125
9.6 USB port 126
9.7 DATA OUT port 127

10 Digitiser specifications.....128

11 Revision history.....130

1 Preliminary Notes

1.1 Proprietary Notice

The information in this document is proprietary to Güralp Systems Limited and may be copied or distributed for educational and academic purposes but may not be used commercially without permission.

Whilst every effort is made to ensure the accuracy, completeness and usefulness of the information in the document, neither Güralp Systems Limited nor any employee assumes responsibility or is liable for any incidental or consequential damages resulting from the use of this document.

1.2 Cautions and Notes

Cautions and notes are displayed and defined as follows:



Caution: A yellow triangle indicates a chance of damage to or failure of the equipment if the caution is not heeded.



Note: A blue circle indicates indicates a procedural or advisory note.

1.3 Manuals and Software

All manuals and software referred to in this document are available from the Güralp Systems website: www.guralp.com unless otherwise stated.

1.4 Conventions

Throughout this manual, examples are given of command-line interactions. In these examples, a fixed-width typeface will be used:

```
Example of the fixed-width typeface used.
```

Commands that you are required to type will be shown in bold:

```
Example of the fixed-width, bold typeface.
```

Where data that you type may vary depending on your individual configuration, such as parameters to commands, these data are additionally shown in italics:

```
Example of the fixed-width, bold, italic typeface.
```

Putting these together into a single example:

```
System prompt: user input with variable parameters
```

2 Introduction

The Güralp DM24 multi-channel digitiser is a state-of-the-art digital module for seismic data. It can be supplied in several different formats:

A *stand-alone* DM24 can be installed with existing or new analogue sensors.



Conversely, a *modular* digitiser is fixed to, and becomes a physical part of, a Güralp seismometer. These can be supplied with new instruments, or retrofitted to Güralp 1T, 3T, 3ESP or 40T seismometers at the factory. A Güralp 5TD is also available, as shown on the right. A modular digitiser benefits from the sensor's isolated, clean and stable environment and allows it to operate at optimum noise levels and precision.

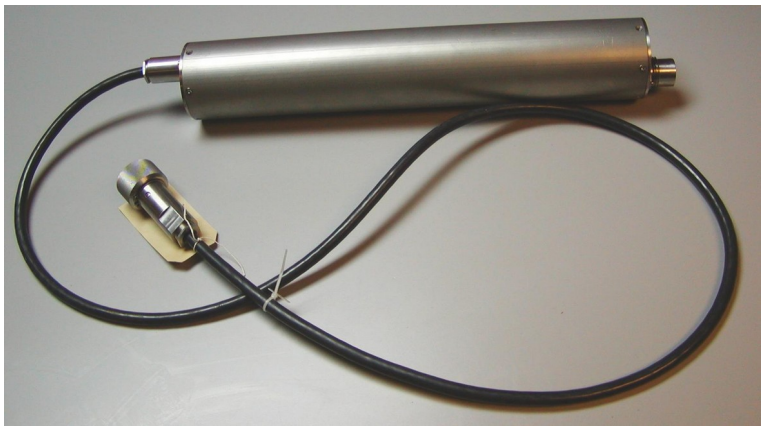


DM24s are also supplied with integrated EAMs (Embedded Acquisition Units) in plastic-cased and metal-cased (aluminium or stainless steel) variants.



For these units, the main interface to the DM24 is via the EAM's web and command line. The manual MAN-EAM-0003, available from the Güralp web site, is more appropriate for these units than is this document.

The DM24 can also be supplied in borehole sonde form, with optional strain relief accessories. These units operate identically to surface DM24s, although the connector for the borehole unit is necessarily more compact.



Regardless of your configuration, all three device types - sensor, digitiser, and data module - can be customized to meet your installation requirements.

2.1 Features

- A built-in Digital Signal Processor (DSP), which can provide simultaneous, multiple sample-rate data-streams at user-selectable rates. Multiple streams of data for each component are available at sample rates from 1 to 1000 samples per second.
- A precision microprocessor-controlled time-base synchronizes the Analogue-to-Digital Converters (ADCs) and the DSP, while also providing time-stamps for data blocks.

- Time synchronization to an attached GPS receiver.
- An ARM microprocessor, which formats and buffers data in an on-board 256 kByte RAM ring-buffer.
- Efficient data storage and transmission using the Güralp Compressed Format (GCF).
- RS232-compatible serial data output at user-selectable baud rates, with automatic modem dial-out facility.
- Support for configuration and sensor control, including mass-locking and unlocking, centring and calibration.
- Low system power consumption.
- Flash EEPROM for program code and filter coefficients.
- High-impedance, low-noise, differential-input pre-amplifier modules with (optionally) adjustable gain.
- Internal electronic thermometer provides temperature measurements at an accuracy up to ± 0.33 °C, with a linearity of ± 0.5 °C.
- Optional Flash memory storage.
- Optional IEEE 1394 FireWire interface, allowing you to download GCF data directly onto a compatible disk by simply plugging the disk into the FireWire connector inside the casing, and monitoring progress on the optional built-in LCD screen.
- Optional USB interface, enabling fast data transfer direct to a PC or other recording device.

The DM24 can be connected to two Güralp triaxial analogue sensors, or up to six conventional single-axis seismometers, through its SENSOR A and SENSOR B connectors. (Single-sensor and integrated DM24 units have only SENSOR A connectors). Signals from these sensors are digitized to 24-bit precision, initially at 2 kHz. Data are output through the DATA OUT serial port of the digitiser in the flexible GCF format, either to a PC or a Güralp data module, such as a Güralp EAM.

3 Installing the DM24

3.1 Power supplies

The DM24 accepts power from lines in its *DATA OUT* port. This uses a Güralp standard ten-pin combined serial/power interface, a cable for which can be obtained from Güralp Systems, part number CAB-BDA-0036. The serial interface is compatible with RS232 connections, PC COM ports and serial-to-USB converters. A standard 12 Volt DC power supply is recommended.

3.2 Connecting sensors

The DM24 is configured for use with Güralp Systems feedback sensors, which can be connected directly to the *SENSOR A* and *SENSOR B* ports. No further configuration of the sensor is necessary beyond unlocking the masses, where appropriate; the input ports carry sensor power, three channels of differential velocity, mass position and calibration signals. The *AUXILIARY* port allows you to measure and digitize environmental parameters, *e.g.* local temperature, pressure, or signals from other scientific instruments.

In addition, two pins on the *AUXILIARY* port allow you to input signals for a fourth, full-rate 24-bit data channel – the 'X' channel. This channel is normally digitized continuously but it is interrupted when you calibrate the instrument so that the DM24 can return the calibration signal on a stream with a different name, when it appears as the 'C' channel. When calibration is over, the fourth channel returns to digitizing the external source.

If you prefer, you can attach any seismometer with a ±10 Volt balanced differential outputs to the signal inputs of the DM24. You will need to make up a cable to connect the relevant pins of the *SENSOR A* or *B* port to your instrument(s). See Section 9.1 on page 121 for details of the connector used.

If you have only one sensor, you should attach it to the *SENSOR A* port, since only this port can be used to send control signals to it (*e.g.* mass locking and unlocking). The *SENSOR B* port can therefore only be used for sensors which do not require remote control signals, such as Güralp accelerometers or the Güralp 6T series of medium noise instruments or 5-series accelerometers. The four-channel model of the DM24 (DM24S3) does not have a *SENSOR B* port.

3.3 Connecting external hardware

On a stand-alone DM24, the digital ports are located on the opposite side to the analogue ports:



The DM24 transmits data through its DATA OUT port. This is an RS232-compatible serial link, outputting data in GCF format according to the particular configuration of the instrument. The DM24 has a default baud rate of 38,400. You can change this from the command line using the `BAUD` command or from Güralp Systems' control, acquisition and monitoring software package, Scream! (see Chapter 5 on page 35).

A *GPS* connector is also provided on the front panel for attaching to a Güralp GPS receiver.

The *DATA IN* port can be used to access the digitiser's console over a direct serial link, or for inputting additional data streams for transmission in GCF format.

Finally, the DM24 can be supplied with an optional USB client interface for connecting to a PC. When this is present, the *USB* port acts as a direct replacement for the *DATA OUT* or console port, and appears as an additional serial port to your computer. You can configure which port is replaced from the digitiser console (see Chapter 6 on page 41).

3.3.1 FireWire disks

The optional IEEE.1394 (“FireWire”) ports on the DM24 may be used to download data onto a compatible hard disk. The unit can be supplied either with standard FireWire connectors or with robust military-specification bayonet plugs (see Section 9.5 on page 125 for pin-out details).



Before you can use the disk, you will need to reset it. The DM24 saves data on the hard disk in a special format, DFD, so you *cannot* use your PC's operating system tools to reset the disk.

You can use the DM24 itself, Scream!, or GCFXtract software to reset the disk. The very first time (for each disk), you can only use the DM24 itself. For instructions on using Scream! And GCFXtract, please see the documentation for these software packages, or their on-line help.

To reset a FireWire disk with the DM24:

1. Power up the DM24, and connect it to your computer's serial port.
2. Open its terminal console. To do this using Güralp Systems' Scream! software, right-click on the digitiser's icon ( ) and select **Terminal...**. From a Güralp EAM, issue the command

   `data-terminal`

   and select the relevant data source from the menu.
3. Issue the command `DISKMENU`. You will see the message

   ```
   Plug in FireWire cable
   ```
4. Plug in your disk. The DM24 will output (on the console) information about the disk as soon as it is detected.
5. *Within the next 7 seconds*, press any key to bring up the disk menu.

6. Key R to reset the disk.
7. When the reset is complete, remove the disk.

You will now be able to download data onto the disk when required.

3.4 The DM24BH borehole digitiser

When the DM24 is ordered in borehole form, it is supplied with a surface interface unit for installation at the top of the borehole. This unit takes the single data cable from the down-hole digitiser and provides standard connections for other equipment.



The connectors provided are:

- a *SENSOR* connector - a ten-way military-specification bayonet socket carrying serial data, GPS and power to the sensor;
- a *GPS* connector for attaching to your GPS receiver; and
- a *POWER / HOLELOCK* connector which can either be connected to a source of 12 – 30 Volt DC power for supplying to the borehole instrumentation (including sensors), or to a Hole-lock Control Unit for driving the sensor's hole-lock motors. It is not usual to power the sensor and hole-lock simultaneously; once the hole-lock is engaged, you should remove the Hole-lock Control Unit and cabling so that the sensor cannot be removed.
- On the reverse side, a standard nine-pin D-subminiature serial connector is provided for connecting to a PC running Scream! using a straight-through serial cable.
- There is also a three-pin mains/line plug, which can power the down-hole instrumentation from a 110 – 250 Volt AC supply if required.

4 Using the DM24

Once the DM24 is connected to your equipment, it will start producing data immediately. You can now start configuring it for your own needs. There are two ways you can do this:

- using the graphical interface provided by Scream!, or
- over a terminal connection (see Chapter 6 on page 56).

Both methods provide full access to the major configuration options of the digitiser.

In most circumstances, you will use Scream! to operate the system. For complete information on how to use Scream! to configure and control your instrument, please refer to the Scream! user guide.

4.1 Taps

The DM24 converts analogue signals to digital data at a high sample rate, which is then reduced in steps. This process is known as *decimation*, and each output stage is called a *tap*. Taps are numbered from zero – the highest sample rate – upwards.

The highest data rate you can choose is 1000 samples/s. This rate can only be produced by Tap 0, the first tap. Tap 0 can alternatively produce data at 500, 400, 200 or 100 samples/s.

After Tap 0 there are three more taps. Each tap produces data at an integer sample rate, which must be 2, 4, 5, 8 or 10 times lower than the previous tap.

For example, the following is a possible sequence:

- Tap 0 : 500 samples/s
- Tap 1 : 50 samples/s (divide by 10)
- Tap 2 : 10 samples/s (divide by 5)
- Tap 3 : 5 samples/s (divide by 2)

The digitiser always generates all four sample rates, but it does not have to output them. You can configure any set of taps to output. You can also have different taps configured for different sensor components.

All of these configuration options are easily changed via Scream!, or you can use the terminal commands `SET-TAPS` and `CONTINUOUS`.

4.1.1 Triggering

The DM24 has a flexible triggering system. When the digitiser triggers, it can optionally output (or store – see *DUAL* mode in section 4.3 on page 22) additional data. Any combination of tap and component can be output as the result of a trigger.

In the example above, you might configure the sensor to output Tap 2 data (at 10 samples/s) continuously, but when a trigger is declared, to output Tap 0 data (at 500 samples/s) as well.

Using triggering helps you to use limited storage capacity or bandwidth more effectively.

The digitiser can look at the data from any tap to decide if a trigger has occurred, including taps which it does not output. Which tap you choose depends on the frequencies you want the trigger system to be most sensitive to.

There are four types of trigger.

- A *LEVEL trigger* occurs when the absolute sample values exceed a configured value.

  The commands `GTRIGGERS`, `MICROG`, and `HIGHPASS` control *LEVEL* triggering.

- An *STA/LTA trigger* occurs when the ratio of a short-term average to a long-term average for recent data exceeds a configured value.

  The commands `TRIGGERS`, `TRIGGERED`, `STA`, `LTA`, `RATIOS`, and `BANDPASS` control *STA/LTA* triggering.

- A *software trigger* occurs when you issue the command `S/WTRIGGER`.

- An *external trigger* occurs when external equipment signals electrically to the DM24. See section 5.1.6 on page 46 for details.

The DM24 can output data streams for a period of time before the trigger starts, and after the trigger ends. The commands `PRE-TRIG` and `POST-TRIG` set these time periods.

Triggering options can also be configured through Scream!.

4.2 Streams

The DM24 organizes the data it produces into streams. Each stream has a 6-character identifier. The first four characters are taken from the *Serial Number* of the digitiser. (Although this setting is called *Serial Number*, you can change it to a different string if you wish, using Scream! or a terminal command.)

The next character denotes the component or output type:

- `Z`, `N`, and `E` denote the vertical, north/south, and east/west components respectively.
- `X` denotes the fourth full-rate data channel, which is provided for connection to your own monitoring equipment via the *AUXILIARY* connector (if present).
- `C` denotes the calibration input channel, which replaces the `X` stream whilst calibration is in progress.
- `M` denotes one of the eight slow-rate Mux (multiplex) channels. Three of these (`M8`, `M9` and `MA`) are used to report the sensor mass positions. Channels `MC` and `MD` are connected to the X and Y axes of the downhole inclinometer, where fitted.
- For `Z`, `N`, `E`, `X`, and `C` streams, the last character represents the output tap. Taps correspond to stages in the decimation process within the digitiser, allowing the DM24 to output several different data rates simultaneously. There are four taps, numbered 0 to 3; 0 has the highest data rate and 3 the lowest. Data streams end in `0`, `2`, `4` and `6` for taps 0, 1, 2 and 3 respectively.

  If you configure the DM24 to output triggered data, these will appear in separate streams ending with the letters `G`, `I`, `K` or `M` for taps 0, 1, 2 and 3 respectively.

The DM24 also generates a stream ending `00`. This is a status stream containing useful diagnostic information in plain text form (see Section 7.1 on page 95).

4.2.1 Digitisers for multiple instruments

The standard DM24 has 4 full-rate channels: one per component of a triaxial instrument, plus the auxiliary and calibration channel.

Some DM24 modules, known as “6-channel digitisers”, are designed for use with two instruments simultaneously, connected to the ports *SENSOR A* and *SENSOR B*. These digitisers actually have 7 full rate data channels, including the auxiliary/calibration channel. When you enable calibration on these digitisers, the signal is passed to both instruments. All 7 channels share the same settings for the four output taps.

A 6-channel digitiser will output full-rate streams with the following final characters:

Z/N/E0	*SENSOR A* at tap 0
Z/N/E2	*SENSOR A* at tap 1
Z/N/E4	*SENSOR A* at tap 2
Z/N/E6	*SENSOR A* at tap 3
Z/N/EG	*SENSOR A* at tap 0, triggered
Z/N/EI	*SENSOR A* at tap 1, triggered
Z/N/EK	*SENSOR A* at tap 2, triggered
Z/N/EM	*SENSOR A* at tap 3, triggered
X/C0	Auxiliary at tap 0
X/C2	Auxiliary at tap 1
X/C4	Auxiliary at tap 2
X/C6	Auxiliary at tap 3

Z/N/E1	*SENSOR B* at tap 0
Z/N/E3	*SENSOR B* at tap 1
Z/N/E5	*SENSOR B* at tap 2
Z/N/E7	*SENSOR B* at tap 3
Z/N/EH	*SENSOR B* at tap 0, triggered
Z/N/EJ	*SENSOR B* at tap 1, triggered
Z/N/EL	*SENSOR B* at tap 2, triggered
Z/N/EN	*SENSOR B* at tap 3, triggered

Standard 6-channel digitisers have the same eight Mux channels as their 3-channel counterparts: `M8`, `M9` and `MA` for the mass positions of SENSOR A, and `MB` to `MF` for your own equipment. These digitisers are designed for use with a strong-motion instrument or geophone attached to SENSOR B, and so do not expose the mass positions for this sensor. There is no calibration facility or equivalent to the X channel for SENSOR B. In addition, SENSOR B cannot be locked, unlocked, or centred.

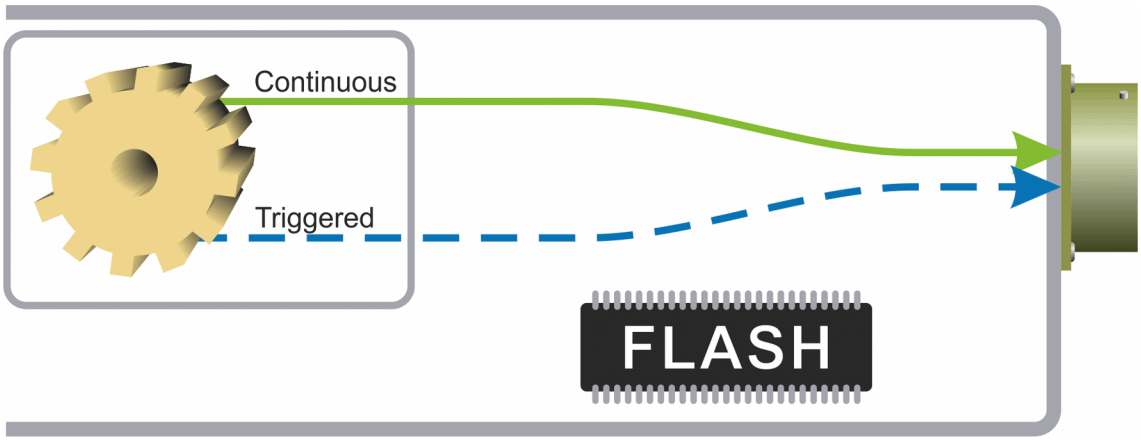
4.3 Transmission modes

Transmission modes are an important concept on the DM24. The current transmission mode determines whether the unit stores data in its on-board Flash memory, sends them over the serial link in GCF format or does some combination of these.

You can switch between transmission modes with a console command (see chapter 6 on page 56) or by using the Scream! configuration interface.

4.3.1 DIRECT

Command: `DIRECT`



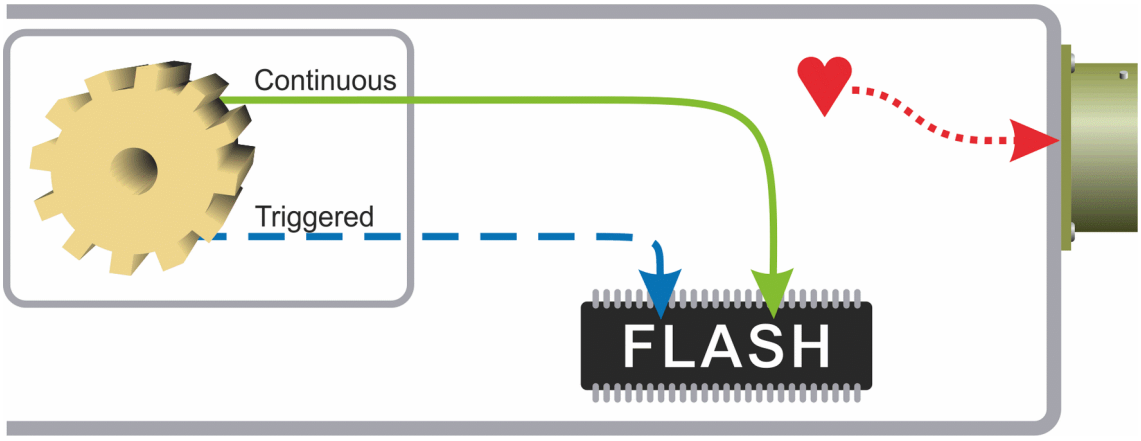
Instructs the DM24 not to use Flash memory for storage. Instead, all data are transmitted directly to clients. An instrument in *DIRECT* mode still honours the GCF Block Recovery Protocol: a temporary RAM buffer always holds the last 256 blocks generated and, if a client fails to receive a block, it can request its retransmission.

If you expect breaks in communication between the instrument and its client to last more than 256 blocks, or if you want *the instrument* to handle breaks in transmission (rather than relying on the client to request missed blocks), you should use

- *ADAPTIVE* mode, if you want data to stay as near to real time as possible (but do not mind if blocks are received out of order) or
- *FIFO* mode, if you need blocks to be received in strict order (but do not mind if the instrument takes a while to catch up to real time).

4.3.2 FILING

Command: `FILING`



Instructs the DM24 not to transmit blocks to clients automatically, but to store all digitized data in the Flash memory. If you have chosen the *RECYCLE* buffering mode (see below), the memory is used in circular fashion, *i.e.* if it becomes full, incoming blocks begin overwriting the oldest in memory. If, instead, the *WRITE-ONCE* buffering mode is active, the instrument will switch to *DIRECT* mode (see above) when the memory becomes full.

You can retrieve blocks from an instrument in *FILING* mode by connecting to its terminal interface and issuing commands such as `FLUSH`, or through Scream! (see below).

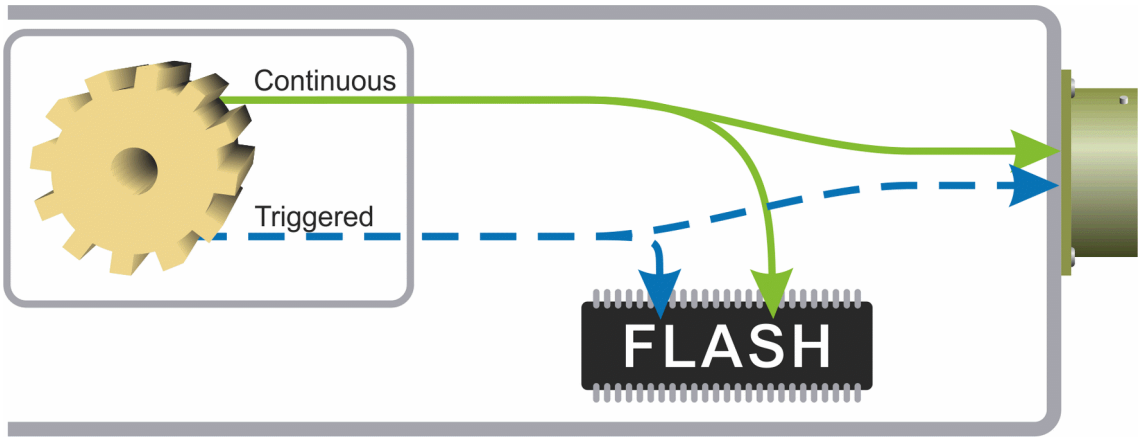
4.3.2.1 Heartbeat messages

When in *FILING* mode, an instrument transmits "heartbeat" messages over its data port. These short messages take the place of data blocks, and ensure that programs such as Scream! know that an instrument is present.

You can change the frequency of heartbeat messages from Scream!'s *Control* window, or with the command `HEARTBEAT`.

4.3.3 DUPLICATE

Command: `DUPLICATE`

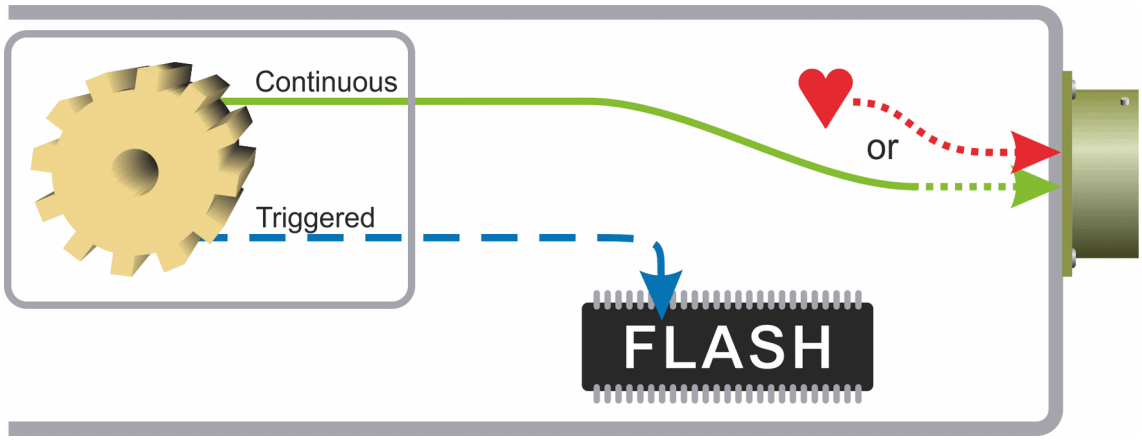


Instructs the DM24 to transmit streams directly to clients as for *DIRECT* mode, but also to store all data into Flash storage as for *FILING* mode.

If a client fails to acknowledge a block, the digitiser does not attempt to retransmit it.

4.3.4 DUAL

Command: `DUAL`

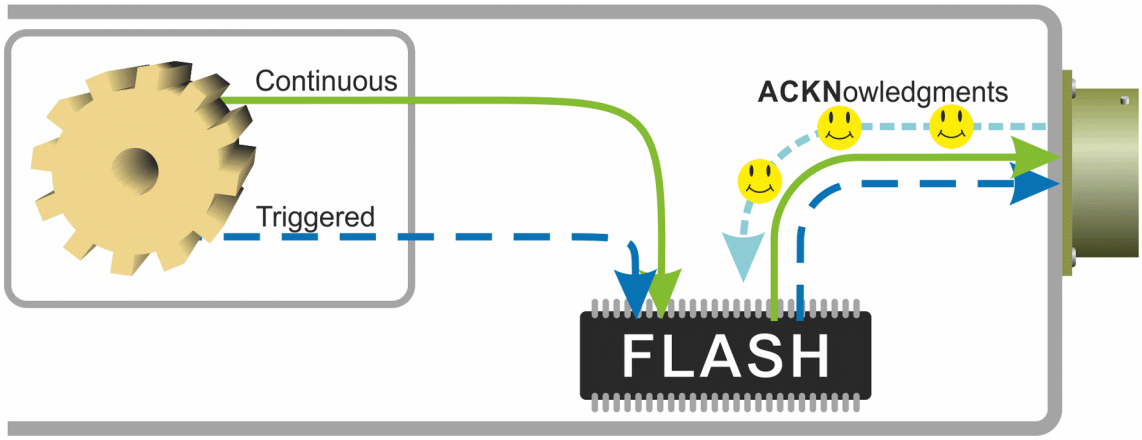


Instructs the DM24 to transmit any *continuous* streams directly to clients as for *DIRECT* mode, but to store *triggered* data into Flash storage as in *FILING* mode.

If you choose *DUAL* mode but do not select any continuous streams for output, the instrument will send heartbeat messages as in *FILING* mode. Scream! can pick these up and download new data as necessary.

4.3.5 FIFO (First In First Out)

Command: `FIFO`



Instructs the DM24 to begin writing blocks to Flash memory as for *FILING* mode, but also to transmit data to clients. Data are transmitted in strict order, oldest first; the DM24 will only transmit the next block when it receives an explicit acknowledgement of the previous block.

If the communications link is only marginally faster than the data rate, it will take some time to catch up with the real-time data after an outage. If you want data to be

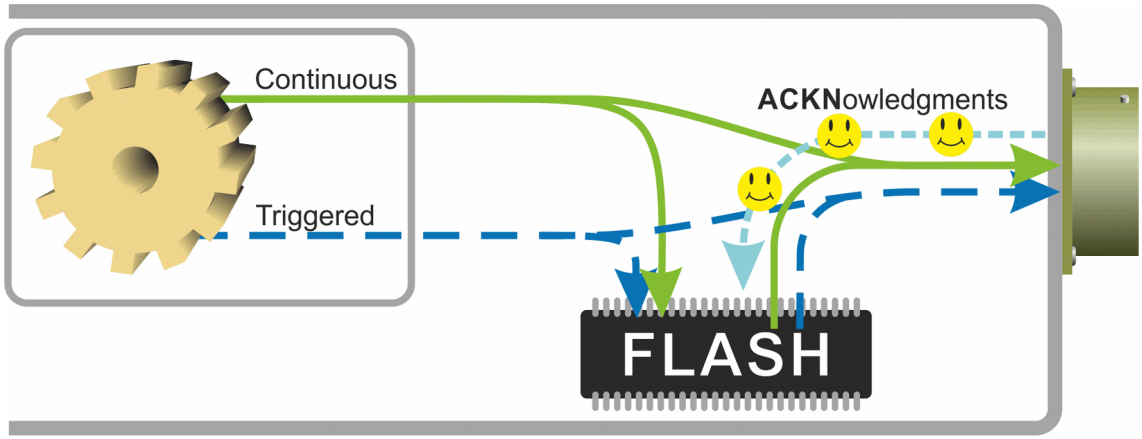
transmitted in real-time where possible, but are worried about possible breaks in communication, you should use *ADAPTIVE* mode instead.

FIFO mode will consider a data block successfully transmitted once it has received an acknowledgement from the next device in the chain. If there are several devices between you and the instrument, you will need to set up the mode for each device (if applicable) to ensure that data flow works the way you expect.

Like all the transmission modes, *FIFO* mode does not delete data once they have been transmitted. You can still request anything in the Flash memory using Scream! or over the command line. The only way data can be deleted is if they are overwritten (in the *RECYCLE* buffering mode, see below) or if you delete them manually.

4.3.5.1 ADAPTIVE

Command: `ADAPTIVE`



Instructs the DM24 to transmit current blocks to clients if possible, but to store all unacknowledged blocks in the Flash memory and re-send them, oldest first, when time allows. *ADAPTIVE* mode is best suited for "real-time" installations where the link between digitiser and client is intermittent or difficult of access.

If the communications link is only marginally faster than the data rate, it will usually be busy transmitting real-time data. Thus, it may take a while for the instrument to work through the missed blocks. In this case, and if your client supports it, you may prefer to use the Block Recovery Protocol to request missed blocks where possible (see section 8.3 on page 118).

Some software packages (most commonly Earthworm) cannot handle blocks being received out of time order. If you are using such a package, *ADAPTIVE* mode will not work and may crash the software.

4.3.6 Transmission mode summary

Transmission Mode: / Function	DIRECT	FILING	DUPLICATE	DUAL	FIFO	ADAPTIVE
Transmit continuous data:	✔	✘	✔	✔	✔	✔
Store continuous data:	✘	✔	✔	✘	✔	?[1]
Transmit trigger data:	✔	✘	✔	✘	✔	✔
Store trigger data:	✘	✔	✔	✔	✔	?[1]
Heartbeat messages:	✘	✔	✘	?[2]	✘	✘
Strict time order:	✔[3]	n/a	✔[3]	✔[3]	✔	✘[4]
Retransmit unacknowledged blocks:	✔[5]	n/a	✘	✘	✔[6]	✔[6]

Notes:

1. Acknowledged data are not stored. Unacknowledged data are stored until they are acknowledged.
2. Heartbeat messages will only be sent if no continuous streams are enabled
3. The receiver can request a "rewind" if blocks are not received correctly but retransmission resumes in strict time order
4. Real-time data are prioritised over missed blocks
5. Up to 256 blocks can be buffered to satisfy rewind requests
6. The whole of flash memory is used as a retransmission buffer

4.4 Buffering modes

There are two different buffering modes which affect how the internal flash memory is utilized. Either may be used with any of the available transmission modes.

4.4.1 RE-USE / RECYCLE

Command: `RE-USE`

Instructs the DM24 to carry on using the current transmission mode when the Flash memory becomes full, overwriting the oldest data held. This buffering mode is called *RECYCLE* in Scream! and on the EAM.

For example, in *DUAL* mode with *RECYCLE* buffering, the latest continuous data will be transmitted to you as normal, and the latest triggered data may be retrieved from the Flash memory using Scream! or the command line. However, if you do not download data regularly from the Flash memory, you may lose older blocks. This mode thus lets you define the end point of the data held by the instrument.

4.4.2 WRITE-ONCE

Command: `WRITE-ONCE`

Instructs the DM24 to stop writing data to the Flash memory when it is full, and to switch to *DIRECT* mode automatically.

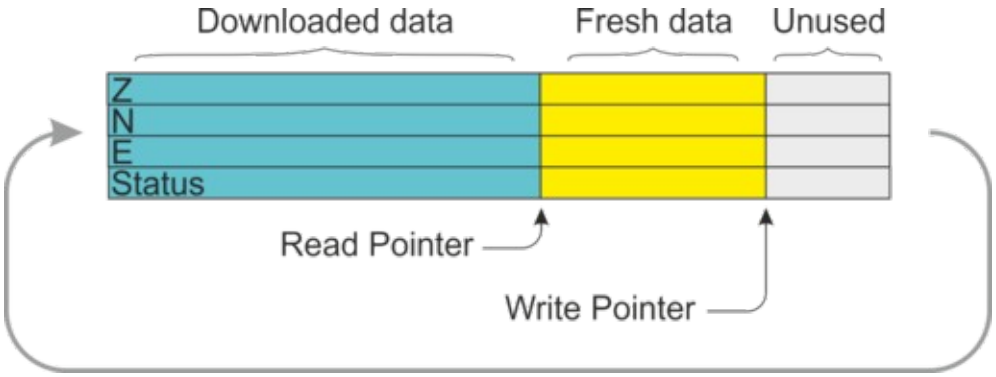
For example, in *FIFO* mode with *WRITE-ONCE* buffering, the station will transmit data to you continuously, but also save them into the Flash memory until it is full. Once full, the instrument will switch to *DIRECT* mode and continue transmitting, though no further data will be saved. This mode thus lets you define the start point of the data held by the instrument.

4.5 Downloading stored data

If you choose a transmission mode where data are stored in Flash memory, you can recover these data at a later date, using either the DM24's FireWire interface or a serial link.

Note: The FireWire interface is very much faster than the serial link: the serial link is only really useful for small, ad-hoc transfers.

The Flash memory is used as a ring buffer. Two pointers into the memory keep track of where data were last read (the "Read Pointer") and last written (the "Write pointer"). When either pointer reaches the end of physical memory, it wraps round back to the beginning. The behaviour when the write pointer reaches the read pointer (i.e. when the memory becomes full of data, none of which have been downloaded) is governed by the commands `RE-USE/RECYCLE` and `WRITE-ONCE`, described in section 4.4 on page 24.



4.5.1 Downloading over FireWire

To download data over FireWire, plug the disk in and issue the command `DISKMENU` from the digitiser's console. The digitiser will reply with a list of download options.

If you have not used the disk before, you will need to reset it. Resetting the disk prepares it for the DM24 to use and sets up the DFD filing system (see below). Press R then `ENTER` to reset the disk. When the DM24 is ready, it will return to the menu.

You can also use GCFXtract or Scream! to prepare a FireWire disk.

To download all the new data held by the DM24, press N then `ENTER`, or wait for 10 seconds. The DM24 will transfer any data it has recorded since the last download and then mark these as transferred.

You can also download selections of data (including the entire contents of the data store, if desired) with the `DISKMENU` command. See Section 6.11 on page 82 for more information.

The DM24 checks for a FireWire disk when it boots up. If there are enough new data waiting to be transferred (by default 128 Mb), the DM24 will automatically transfer them onto the disk. The digitiser then updates its internal pointers to mark these data as already downloaded.

This feature is particularly useful for autonomous installations: after the digitiser has been retrieved from a field experiment, simply powering it up with a prepared disk attached will make it transfer all the data it has recorded during the experiment.

While the FireWire interface is active, it will consume about 200 mA of current (from a 12 Volt supply). If you interrupt a transfer whilst it is in progress, the digitiser will re-boot and will not mark data as downloaded. Because of this, the next time you transfer data, some will be duplicated. You will not, however, lose any data.

4.5.2 Reading DM24 disks

When the DM24 saves data onto a FireWire disk, it uses a special format called `DFD`.

You can read these data into a PC using Scream! or the GCFXtract utility, which is freely available from the Güralp Systems Web site. Linux and Solaris command line utilities are also available for reading data from a DFD disk.

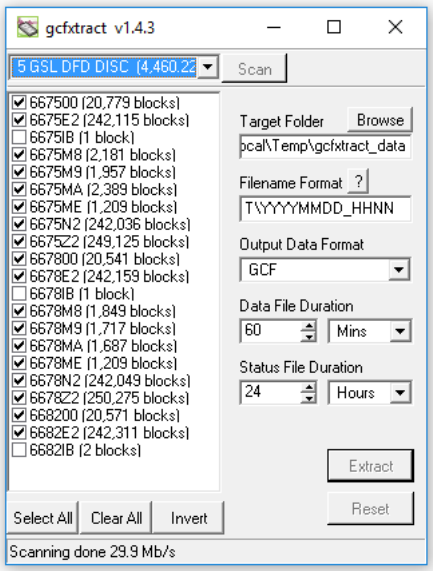
The DFD format is *not* the same as that used by the Güralp Systems EAM data acquisition module, which uses a FAT32-compatible journaling file system.

Güralp Systems can provide fully-tested, formatted disks with FireWire and USB connectors. Alternatively, a third-party FireWire disk may be used (although compatibility is not guaranteed).

To read a disk using GCFXtract:

1. Attach the disk to your computer. You can use FireWire, USB, or any other interface supported by your computer and the disk.

2. Run GCFXtract and select the required disk from the drop-down list, then click Scan.



   GCFXtract will scan the disk and display all the streams it finds in the selection area below. For each stream, the *Stream ID* and the number of blocks found are shown.

   This operation requires roughly 12 Mb of available memory for every Gb of space on the disk. If you have a very large disk, your computer may have to use its hard disk to make enough space. This will slow down scanning considerably.

3. By default, all streams containing more than 100 blocks are selected for extraction. You can change which streams to extract by ticking or clearing the check-box beside each stream.

4. You can tick or clear all of the check-boxes using the Select All and Clear All buttons. Clicking Invert Invert ticks all clear boxes, and clears all ticked boxes.

5. Enter a path name into the Target Directory field, or use the Browse button to find a directory. This will be used as the root directory for extracted data. If it does not exist, GCFXtract will create it.

6. Enter a format string into the Filename Format field. The syntax is the same as the format string in Scream! and full documentation is available by pressing the ? button beside the format entry field in interactive mode.

7. Normally, GCFXtract outputs GCF files, to ensure all the information in the original data is retained. If you want to convert to a different format, select it from the Output Data Format drop-down box. GCFXtract can output in most of the formats supported by Scream!.

8. Data are automatically placed in time order and saved in multiple files, each file containing a contiguous segment of data. By default, data streams are recorded in files 60 minutes long. To change this to some other number of minutes, alter the value in the Data File Duration (mins) box.

   If there is a gap in a data stream, GCFXtract will start a new file anyway.

   Status streams are also saved in in multiple files, but have a default length of 24 hours. To change this, alter the value under Status File Duration (hours).

9. When you are happy with the settings, click Extract to begin extracting the data.

10. Clicking Reset sets a flag on the disk which marks it as empty. Next time a digitiser wants to transfer data, it will begin at the beginning of the disk, overwriting the old data. When this happens, none of the old data can be extracted with GCFXtract or Scream!. Until then, however, you will still be able to retrieve all the data.

11. When you are finished, you should exit GCFXtract and then use your operating system's standard facility for un-mounting hardware (*e.g.* "Safely Remove Hardware" under Windows) before disconnecting the drive.

You can also read disks with Scream!. This allows you to view data in the process of being transferred, but is slightly slower, because Scream! does not read data in strict order. To read a disk with Scream!:

1. Attach the disk to your computer. You can use FireWire, USB, or any other interface supported by both your PC and the disk.

2. Run Scream!, and select File → Setup… from the main menu. Select the Files tab.

3. Set the Base Directory, Filename Format and Data Format as described above. Also, if required, set the Post-processor and Granularity options to suit your requirements. Consult the Scream! documentation for details.

4. Select the Recording tab, and check Auto Record—Enable for Data Streams and Auto Record—Enable for Status Streams. Click OK. Scream! will remember the recording options you set in steps 3 and 4 for later occasions.

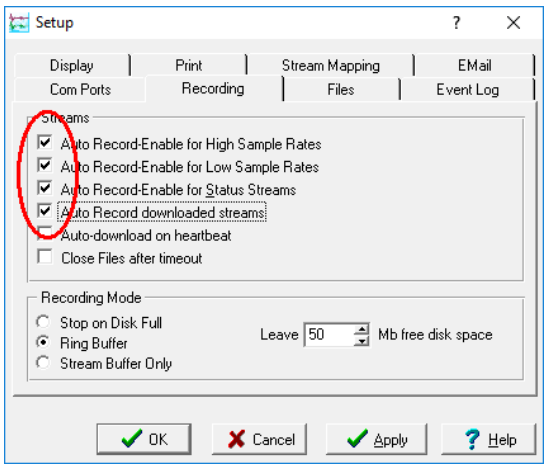
5. Select File → Read SCSI disk… from the main menu. Scream! will search for attached disks, and open a window with a list of all the streams it has found.

6. Select the streams you want to replay, and click Open. The disk will appear in the left-hand pane of Scream!'s main window, and the streams you have selected will start playing into the stream buffer, as well as being recorded.

7. When you have finished transferring the data, if you want to reset the disk, select File → Reset SCSI disk... from Scream!'s main menu. Select the disk you want to reset, and click OK.

4.5.3 Downloading over the serial link

Start by configuring Scream to record all incoming data by visiting the "Recording" tab of the File→Setup dialogue:



Ensure that the first four check-boxes are ticked. Next, switch to the "Files" tab and choose an appropriate base directory, file format and filename format.

Click OK to save your changes and close the dialogue.

Next, open a connection to the digitiser's console. To do this using Güralp Systems' Scream! software, right-click on the digitiser's icon ( ) and select **Terminal...**. From a Güralp EAM, issue the command

```
data-terminal
```

and select the appropriate data source from the menu.

To simply download all data held in the Flash memory, issue the command

```
ALL-FLASH ALL-DATA DOWNLOAD
```

followed by

```
GO
```

This will initiate a complete download. The `ALL-FLASH` and `ALL-DATA` commands act as modifiers to tell the system what to download.

If you wish to download only a subset of data, you can replace the **ALL-FLASH** and **ALL-DATA** modifiers with other modifiers to select different streams and/or time periods:

- The streams to be downloaded are specified with **ALL-DATA**, **S/S**, or **STATUS-ONLY**.
- Any desired time period can be specified with **ALL-FLASH**, **ALL-TIMES**, or **FROM-TIME** and/or **TO-TIME**.

The parameters are illustrated in the diagram below. If you miss out a parameter, **DOWNLOAD** will re-use the value you last specified.



The **DOWNLOAD** command returns immediately, so that you can issue more commands if required. To close the connection and begin downloading with the specified selectors, issue the **GO** command.

Note: You can pause a download by entering terminal mode, and either restart with another **GO** or abort with **END-DOWNLOAD**.

When you complete a DOWNLOAD without specifying a time period, the DM24 adjusts the internal read pointer to mark the latest position. This is then used as the start point for the next DOWNLOAD with the command ALL-TIMES.

The various modifiers are described in detail in section 6.12 on page 87.

Once the GO command has been issued, the window may close, depending on which emulator you are using. On an EAM, the data-terminal (minicom) session is closed by keying Ctrl + A, then Q.

4.6 DM24 USB Host Operations

DM24s can be supplied with an optional USB interface. This section describes the use of this interface.

4.6.1 DM24 Mk3 USB host option



This interface enables the DM24 to record GCF data to a standard USB disk drive or Flash memory stick using the ISO 9293 FAT filing format – readable by Linux, DOS, Windows and Mac OS.

Equipment fitted with the USB Host Port connector (labelled USB BOMS – Bulk Only Mass Storage) will automatically record the current real time data to the USB device when it is plugged in.

To do this, the DM24 should be in *DIRECT* or *DUPLICATE* transmission mode so that real time data are being fed to the standard transmit buffer used for normal serial data transmission. The USB process runs in parallel with the serial transmit thread using the data in the transmit buffer. It does not matter whether the serial port is connected or not.

Other transmission modes do not pass useful data via the serial port and so cannot be used with USB devices. In particular *FILING* mode saves all data to the internal flash memory (for later download via the serial port or FireWire interface) and only outputs "heartbeat" status to the serial port to inform systems monitoring the port that the unit is present and operating. *FIFO* and *ADAPTIVE* modes use the internal flash as an extension to the serial transmit buffer as explained in section 4.3 on

page 19. As no contiguous streams are sent via the serial port in these modes, nothing will be written to the USB device.

To record data to a USB device, the system should be configured in *DIRECT* or *DUPLICATE* mode – in *DUPLICATE+REUSE* mode the DM24 will always have the last 64Mb or more of data in its internal flash – the firmware does not currently support transferring data from the DM24 internal flash to a USB device.

4.6.2 Formatting a USB device

If you wish to use a USB device it must be formatted with 16 Kilobyte clusters. In Windows 2000, XP and Vista open a command prompt and type:

```
format e: /fs:fat32 /a:16k
```

replacing **e** with the letter of the drive which you wish to format.

4.6.3 Directory structure

The system creates the following directory (folder) structure:

The main directory is \GURALP

Beneath this is a sub-directory named after the DM24's system ID (*e.g.* `GSLA`) within which will be a further sub-directory for each stream the system is configured to produce. For example:

- `C838Z2` tap2, vertical component at selected sample rate
- `C838M8` vertical component mass position
- `C83800` the digitiser's status stream

Data for each stream are recorded as files within the corresponding sub-directory. The file name is simply the date and time of the first sample in the file, given in the format `YYMMDDHH.xxxx`

For example:

- `08102314.gcf` 2008 Oct 23 14:00 – data file
- `08102312.txt` 2008 Oct 23 12:00 – status/text file

The amount of data in each file (granularity) depends on both the stream sample-rate and type. It can be configured using the `HOURFILES` command, which takes four parameters (which *precede* the command – see chapter 6 on page 56 for details of the DM24's internal FORTH interpreter):

```
FAST SPS SLOW STATUS HOURFILES
```

FAST specifies the file size, in hours, for sample rates ≥ ***SPS***

SLOW is the size, in hours, for lower sample rates < ***SPS***

STATUS is the size, in hours, for status stream files.

The default settings are:

```
1 40 4 12 HOURFILES
```

With these settings, the system will start a new file every hour for streams with a sample rate ≥ 40 samples per second.

Lower sample rate streams will have files which last 4 hours – so the file names will have the sequence "`YYMMDD00`, `YYMMDD04`, `YYMMDD08`" etc.

Status files will have 12 hour granularity *i.e.* "`YYMMDD00` and `YYMMDD12`".

Note that status streams are recorded as plain text files with the extension '`.txt`'.

All other streams are recorded with the extension '`.gcf`' which is the same format used by 'Scream!' (*i.e.* a series of 1-kilobyte GCF data blocks), consequently the files can be viewed and processed by Scream!, simply by "drag and dropping" or right-clicking to select the option.

Due to bandwidth and processing limitations, it may not always be possible to record continuous streams with particularly high sample rates. Note that when triggers are in use, the sample rates of streams transmitted over the serial port will typically increase considerably. If your application involves sample rates higher than 250 samples per second, you should verify to your own satisfaction correct operation (under triggered conditions, if necessary) with your own USB devices before permanent installation.

4.6.4 System operation

USB memory sticks and disk drives normally have an LED indicating activity. This will flash as each block is written, so a few flashes every 10 seconds would be expected from a DM24 producing 100 sample-per-second data.

Since data blocks are recorded to their corresponding files in real time, only the current/latest block can be lost when a USB stick is detached; all other stream files will have been closed/saved.

When a USB stick is plugged in, the system interrogates it to determine the free space. This operation can take a minute or more depending on the capacity of the USB device.

When the operation has been completed, the system reports the available space using the optional LCD and, also, in the status stream, as in the example below:

```
2008 10 3 13:18:49 USB Flash stick Recording started NO NAME
D:/>
USB File granularity :
For sample rates >= 40 sps : 1 hour files otherwise 4 hour
files
  except status ; 12 hour files
2008 10 3 13:19:52  USB Stick space remaining   3.9GB
```

The system will regularly (after each 5 MB of data storage – typically an hour or two of running) report the space available in GB/MB and as a predicted duration in days/hours (based on the utilization rate for the previous 5 MB).

Recording to a USB device stops automatically when there is less than 100 kB free space. To record more data, the device must be replaced with a freshly formatted one.

It is recommended that the USB device is only removed when the display shows `USB free`, which is followed by a bar graph indicating a 5 second countdown. The system does not automatically buffer data so, if the device is removed while the display shows `USB BUSY  WAIT` or while it is scrolling the earlier status information, a small amount of data will be lost during the changeover. In *DUPLICATE* mode all data would also still be in the internal flash.

An example of a status display is shown below. This is taken from the status stream (*00) of a DM24 fitted with a USB host controller after several minutes of operation.

```
26/11/2008 13:05:58
 2008 11 26 13:00:00 GPS Date/Time 26/11/08 13:00:00
 2008 11 26 13:00:00 o/s=    -18 drift=   -76 pwm= 8399  Auto 3D
 2008 11 26 13:00:00 Auto 3D  SV#'s 3 6 16 18 21 25 7   ( 7 )
 2008 11 26 13:00:00 Lat 51'21.6572N  Long 001'09.8189W Height  112m
 2008 11 26 13:01:00 o/s=    -81 drift=   -63 pwm= 8401  Auto 3D
 2008 11 26 13:02:00 o/s=   -141 drift=   -60 pwm= 8403  Auto 3D
 2008 11 26 13:03:00 o/s=   -200 drift=   -59 pwm= 8405  Auto 3D
 2008 11 26 13:04:00 o/s=   -257 drift=   -57 pwm= 8408  Auto 3D
 2008 11 26 13:05:00 o/s=   -260 drift=    -3 pwm= 8411  Auto 3D
 2008 11 26 13:05:00 External supply : 14.1V Temperature  29.62'C
 2008 11 26 13:05:58   USB Stick space remaining   7.8GB[ >33 Days ]
26/11/2008 13:17:00
 2008 11 26 13:06:00 o/s=   -253 drift=     7 pwm= 8412  Auto 3D
 2008 11 26 13:07:00 o/s=   -201 drift=    52 pwm= 8413  Auto 3D
 2008 11 26 13:08:00 o/s=   -142 drift=    59 pwm= 8414  Auto 3D
 2008 11 26 13:09:00 o/s=    -84 drift=    58 pwm= 8415  Auto 3D
 2008 11 26 13:10:00 o/s=    -35 drift=    49 pwm= 8416  Auto 3D
 2008 11 26 13:10:00 0 MicroSeconds Fast  Freq error 0 e-9
 2008 11 26 13:10:00 Mass Positions   0%  0%  0%
```

USB STICK SPACE REMAINING (APPROX TIME)

5 Using Scream!

Scream! is a versatile seismic data visualisation program for Güralp instruments which runs on Windows and Linux computers. If a DM24's *DATA OUT* port is connected to an RS232 port on a computer running Scream!, the program can be used to analyze digitized data from the DM24's connected analogue sensors, and to control/configure both the digitiser and the sensors themselves. Scream can also work with network-attached digitisers.

Scream! is available for free download from Güralp Systems, or may have been supplied with your equipment.

5.1 Configuring digitisers

Scream! distinguishes between *configuration* and *control* of digitisers. The most important difference is that a digitiser may be *controlled* through Scream! at any time whilst it is acquiring data, whereas *configuration* options only take effect after a reboot (with consequent loss of data).

To change the configuration of any connected digitiser:

1. Locate the digitiser you want to configure. All connected digitisers have an entry in the tree on the left of Scream!'s main window. If the digitiser is transmitting data through a remote server, EAM or NAM, you may need to "unroll" the entry for that server (by clicking on the ⊞ icon) to see the digitisers connected to it.
2. Right-click on the digitiser's icon ( ).
3. Click **Configure...**. Scream! will then contact the digitiser and retrieve its current configuration, a process which will take a few seconds. This done, the *Configuration setup* window will be displayed.
4. Once you are happy with any changes you have made in the *Configuration Setup* window, click **UPLOAD** to send them to the digitiser and reboot. This will take a short while.

To control a digitiser whilst it is running, either right-click on the digitiser's entry in the list and click **Control...**, or double-click the entry. In either case Scream! will contact the digitiser to retrieve control information and display the *Control* window. The options you can control immediately are:

- the type of sensor you are using;
- GPS power cycling options;
- the short-term and long-term average values for triggering (but *not* which streams perform the trigger, or which are output by it);
- the length of pre-trigger and post-trigger periods;

- calibration signal options; and
- mass control functions.

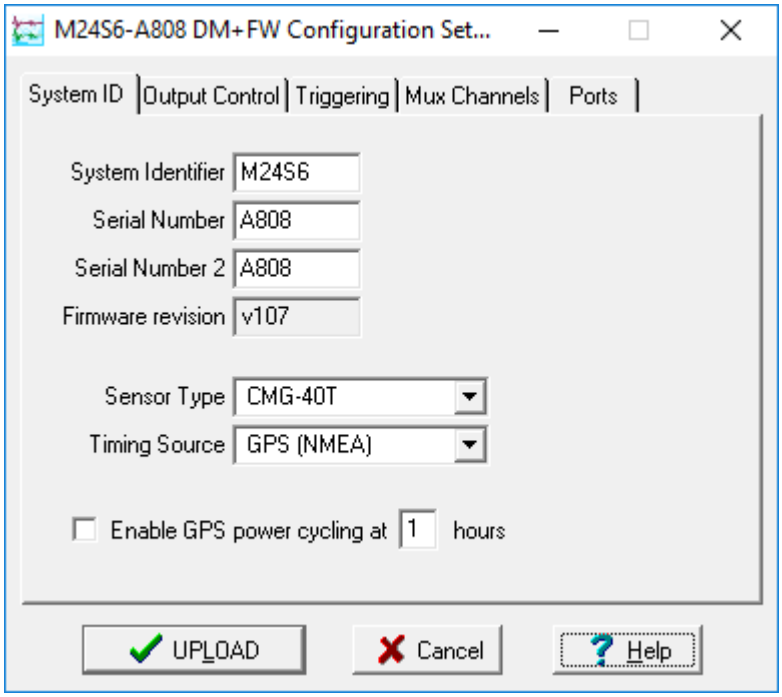
Some of these options can also be altered in the *Configuration setup* window. For more information on the *Control* window, see Section 5.2 on page 50.

If you need a more powerful interface to the DM24, you can also issue commands to it directly using Scream!'s terminal mode. A terminal window is opened by right-clicking on the digitiser's entry in the list and selecting **Terminal...**. The digitiser will stop transmitting data while you have a terminal window open. Data collected whilst the terminal is active will be stored for transmission later, memory space permitting.

The remaining sections of this chapter describe in detail the configuration options available for the DM24.

5.1.1 System ID

The *System ID* pane gives information about the digitiser and its internal software, and allows you to change GPS timing parameters.



System Identifier and *Serial Number* : The digitiser type is identified by its system identifier and serial number. Every data and status block generated by the digitiser includes these two fields at the beginning, so that the block's origin can be identified. On delivery from the factory, the system identifier and the serial number are set to the Güralp works order number and the digitiser's serial number, but any combination of letters A-Z and numbers can be used, such as an abbreviation of your

institution's name, etc. The system identifier can be up to 5 characters long, whilst the serial number cannot be longer than 4.

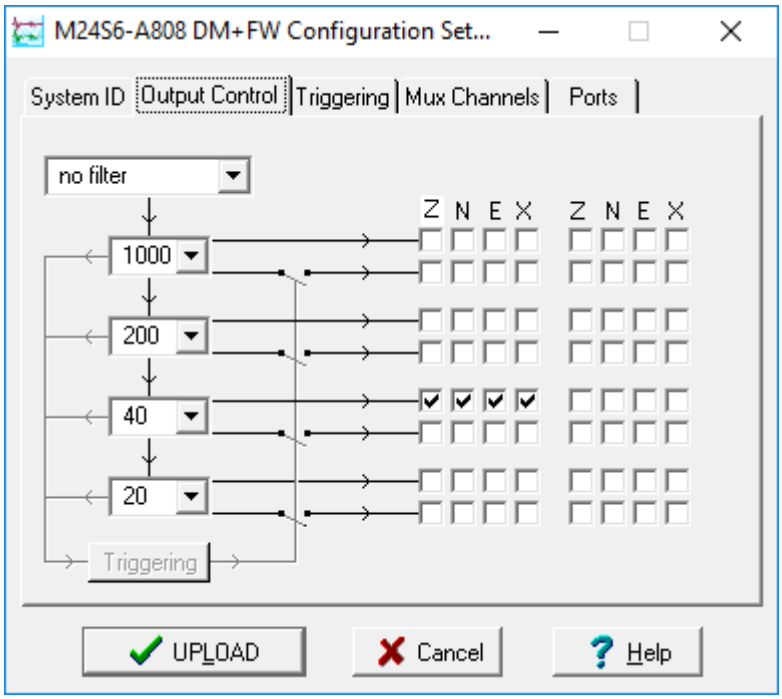
Sensor Type : If the sensor attached to the digitiser is a Güralp velocity sensor, useful seismometer functions (such as sensor locking, centring, and calibration) may be controlled through the digitiser. The *Sensor Type* you set here determines which functions will be available through the Scream! digitiser configuration set-up interface or through interactive commands.

GPS Type : The digitiser needs to be able to time-stamp accurately all data that passes through it. It can set its clock either by receiving time signals from the GPS satellite network using an attached Güralp GPS receiver unit, or by taking time information from a central site (stream sync mode). In *stream sync* mode, the digitiser expects to receive two-byte packets from the central timing source, which may have its own GPS unit, or take signals from one of the radio time standards. Choose the mode you require from the drop-down menu.

Enable GPS power cycling : If you are using a GPS unit to receive time signals, but do not experience significant drift in the system's clock (for example, in a stable-temperature environment), you can save power by selecting **Enable GPS power cycling**. With this option in use, the GPS time is only checked at intervals of a specified number of hours. Disabling this option keeps the GPS unit running constantly; if you have ample power, this will give the most accurate results. You can choose any whole number of hours for the interval.

5.1.2 Output control

The *Output control* tab allows you to configure which data streams are sent to Scream! from the digitiser.



Güralp digitisers initially sample incoming data at a high rate (*e.g.* 2000 Hz for the DM24), which are then filtered and reduced to a lower rate (*decimated*) using an on-board digital signal processing unit, or DSP. The DSP has several filtering/decimation stages, which run one after the other. Stages which can produce output are called *taps*. The Güralp DM24 can output 4 taps simultaneously.

Each configurable tap can be set to a different decimation factor by choosing values from the drop-down menus on the left. Decimation factors of 2, 4, 5, 8, and 10 are available. The numbers visible in the drop-down menu of each tap are the data rates that each of the possible decimation factors will provide, given the settings of the taps above it. Only integer (Hz) data rates are allowed: thus, for example, if one tap emits data at 25 Hz, the only possible further decimation factor is 5.

To the right of each decimation factor menu is a grid of check-boxes. These boxes mark which streams of data to generate at each sample rate. The screen-shot above shows a possible configuration for a triaxial instrument. Every channel of the digitiser may be output at any tap; In the example above, all three axes are being output at Tap 2 (20Hz). The DM24 MkIII has a fourth channel for external input and calibration, which can also be output at any tap.

If you want to change the names used for the channels, click in the white box containing a Z in the above picture, and type a letter or number. It will name the channels with a sequence of letters or numbers beginning with the one you choose *(e.g.* A, B, C; 2, 3, 4; 9, A, B), unless you type `Z` in which case they will revert to Z, N, and E.

Each combination of channel and tap has two check-boxes:

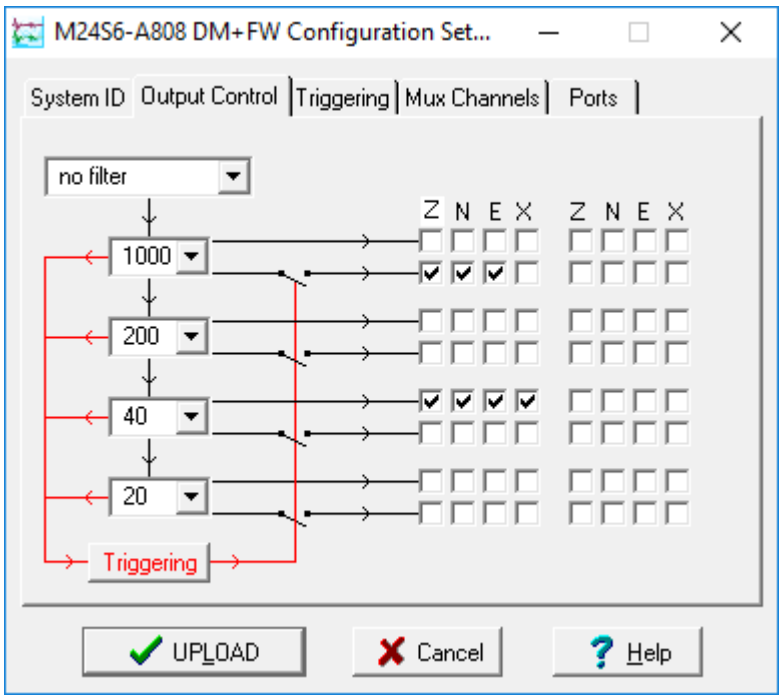
- Checking the upper check-box of each pair makes that tap output data continuously for that component;
- Checking the lower check-box makes that tap output data only when a trigger is active.

In the example above, the digitiser will output data continuously for all three channels at Tap 2, but never for any other taps. If you do not need all the streams to output at all rates, you should leave boxes unchecked to save communications capacity.

You cannot check both continuous and triggered output for the same channel and tap.

When you enable a triggered stream, the digitiser will output data in that stream only when a particular set of trigger criteria are met. To remind you that the streams are not continuous, the window shows the data streams (lines) passing through a “switch”.

In the example above, we might want the high-rate data from Tap 0 to be generated only when an event registers. To do this, the *lower* check-boxes of Tap 0 should be ticked:



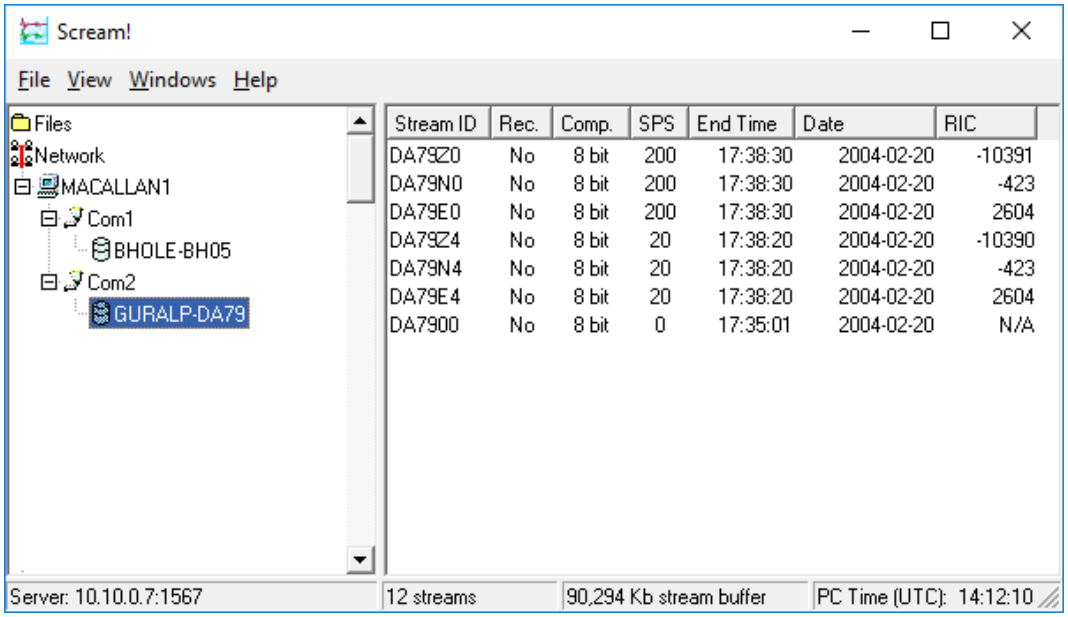
With this configuration uploaded, Tap 2 will continue to produce output at all times, but Tap 0 will also emit data whenever the trigger criteria are met. The *Triggering* button is now shown in red to remind you that the trigger is active.

When a trigger occurs, every checked box in this window will cause the digitiser to output a data stream. This stream will appear in Scream!'s main window when Scream! first receives some data from it.

Every stream is identified by a 6-character code, where the first four characters identify the digitiser, and the last two characters identify the individual stream. The first four characters are set by default to the serial number of the digitiser; you can change this on the *System ID* pane (see above) or from the digitiser's console.

Using the example above, there are three data streams, Z, N and E, which each output data at 40 samples/s (continuously) and 1000 samples/s (occasionally).

Scream's main window might then appear as below:



In this example:

- `DA79` is the serial number of the digitiser.
- `Stream Ids ending Z0`, `N0` and `E0` correspond to input channels Z, N, and E, with output derived from Tap 0.
- `Stream Ids ending Z4`, `N4` and `E4` correspond to input channels Z, N, and E, with output derived from Tap 2.

  Scream! can replace these designations with more helpful names if you wish: see the Scream! manual or online help for more details.
- `DA7900` is the digitiser's status stream (notice its zero sample rate).

5.1.3 Triggering

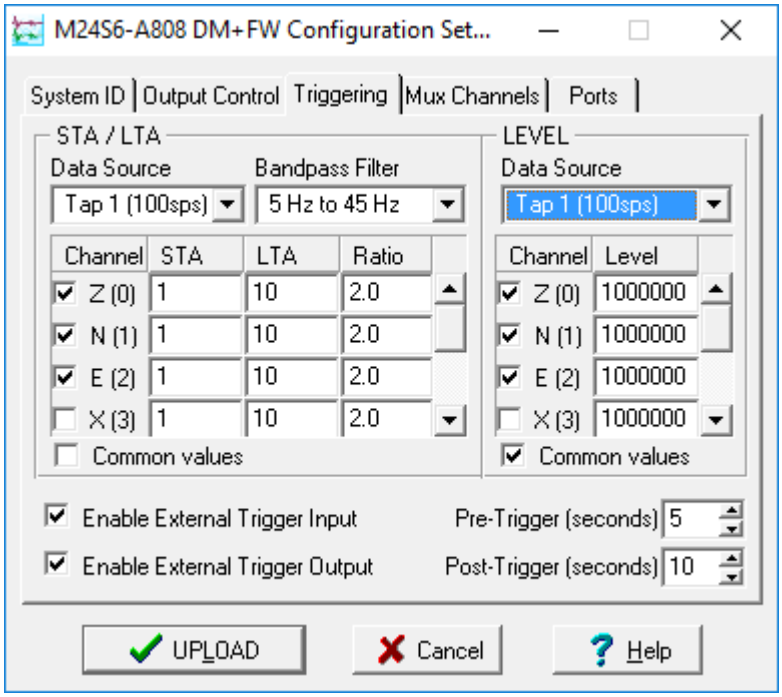
In its standard configuration, the DM24 outputs continuous data at a sample rate you specify. In addition to this, Güralp digitisers can run a triggering algorithm on the data they acquire. This allows you to record data continuously at a relatively low sample rate, but record at a much higher sample rate during short periods when the trigger is active. The parameters controlling the triggering algorithm, and controlling the data output once the system is triggered, are all selectable by the user, permitting maximum flexibility of operation and the most efficient use of available storage space.

The DM24 can be set up for triggered output, that is, to output certain data streams only when a particular trigger criterion is met. The trigger criterion can be tested with data from the same or some other stream. For example, you could use a later tap (with a lower sample rate) as a trigger for output from an earlier, more detailed

tap. Scream! 4.5 also allows you to configure each digitiser to receive triggers from other digitisers.

To create a new stream with a trigger, open Scream!'s *Digitiser configuration* window for the relevant digitiser, and click on the *Output control* tab. In the *Output control* pane, a tap which gives rise to a triggered stream has a tick in the *lower* row of its grid of check-boxes. You cannot configure the trigger criteria until you have selected at least one stream to be affected by the trigger.

Once you have decided which streams should be output when the trigger is activated, you will be able to click on the **Triggering** button to describe the trigger condition. Alternatively, click on the **Triggering** tab at the top of the window. Either action will open the *Triggering* pane:



There are two triggering algorithms which Güralp digitisers can use. However, not all models can use both methods. Scream! will interrogate the digitiser to discover which methods are supported by its on-board software.

In addition, some digitisers support external and software triggering.

5.1.4 STA/LTA – Triggering by average values

The *STA/LTA* algorithm applies a simple short-term average – long-term average calculation to the triggering stream. It works by identifying sections of an incoming data stream when the signal amplitude increases. The purpose of taking a short term average, rather than triggering on signal amplitude directly, is to make it less likely that spurious spikes will trigger the device. Averaging also introduces an element of frequency selectivity into the triggering process.

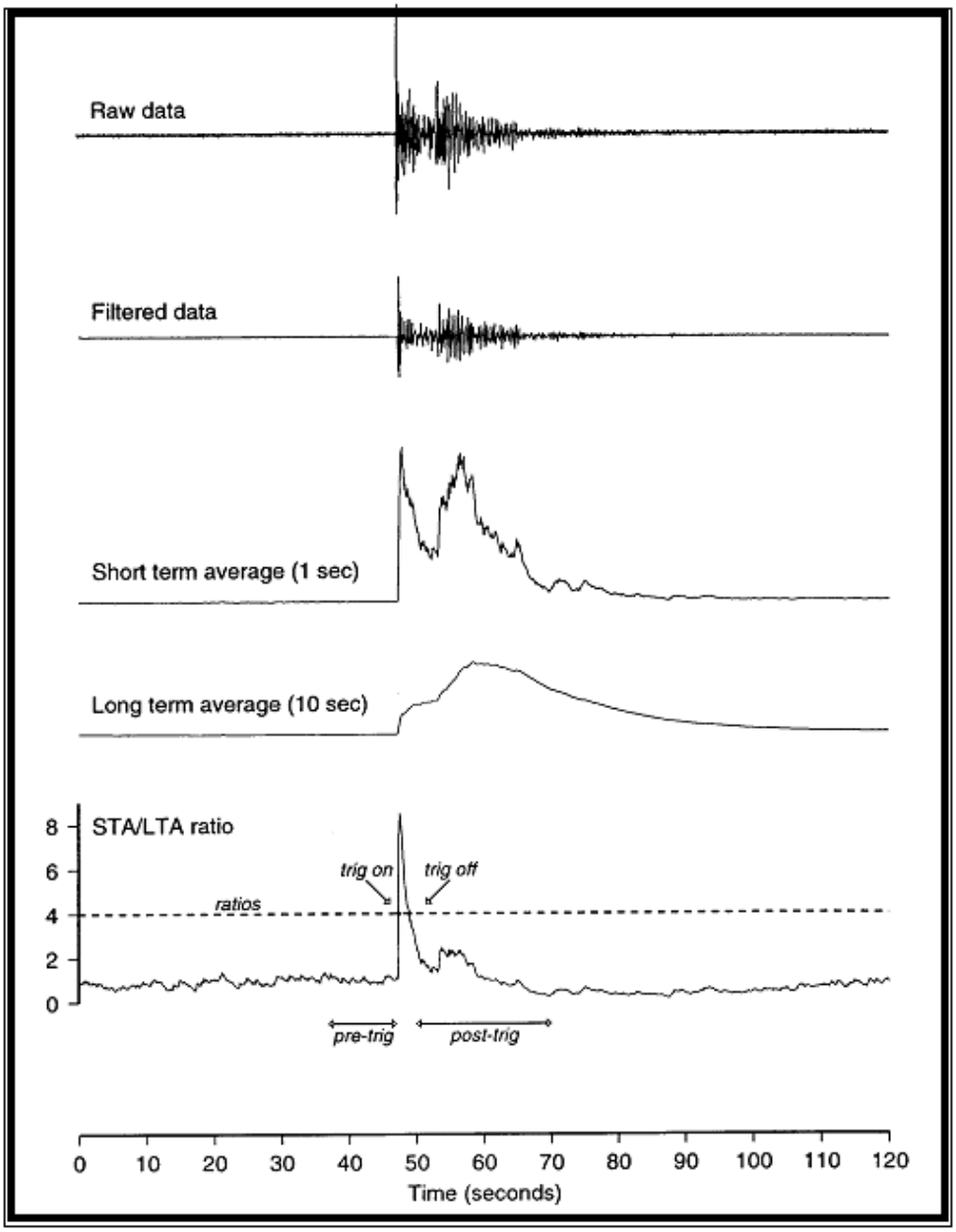
You can select which tap is tested for the trigger from the **Data source** drop-down menu. The tap does not have to output data to Scream! for you to be able to use it here.

Any or all of the channels available at that tap may be used to determine a trigger. You can select which channels are considered by checking the boxes in the **Channel** column of the table. If *any* of the checked channels passes the trigger condition, the trigger will activate, and will remain active until *all* of the checked channels have fallen below their respective ratio values.

The **STA** and **LTA** columns allow you to set the intervals over which the two averages are calculated, in seconds. Typically, the time interval for the short term average should be about as long as the signals you want to trigger on, while the long term average should be taken over a much longer interval. Both the STA and LTA values are recalculated continually, even while a trigger is active.

The **Ratio** column determines by what factor the STA and LTA must differ for the trigger to be passed. Finding the ratio most suited to your needs is best done by experiment. Too high a value will result in events being missed, while too low a value will result in spurious non-seismic noise triggering the system. Like the averages, their ratio is continuously recalculated for all components. Note that none of the boxes are allowed to be empty, and so you will need to enter the new value before removing the old one. Alternatively, you can use the up and down cursor keys to change the values.

For example, setting the **STA** to 1 second, the **LTA** to 10 seconds and the **Ratio** to 4 would give rise to the following trigger behaviour:



Usually, the values of the **STA** and **LTA** periods, and of the **Ratio**, will be the same for all checked channels. For convenience, Scream! will automatically fill in other values to match ones you enter. If you want to use different values for some channels, you should clear the Common values check-box before altering them.

Once you have enabled the *STA/LTA* triggering method on a particular channel, you can use the *Control* window to change the values of the **STA** and **LTA** periods, together with the **Ratio**, without restarting the digitiser (see Section 5.2 on page 50).

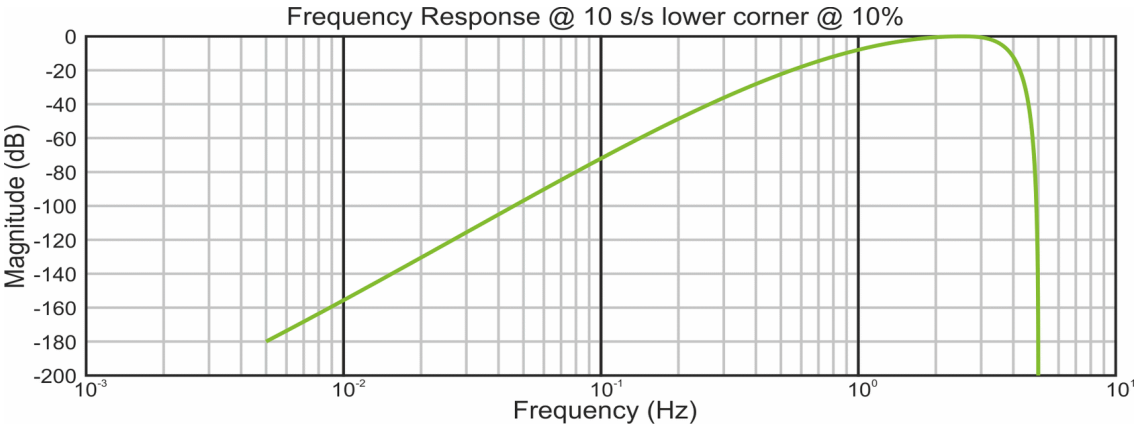
Since it is not generally advisable to trigger from broadband data, the digitiser provides a set of standard bandpass filters to apply to the data streams before they are tested for the trigger condition. This filtering serves to maximise sensitivity within the frequency band of interest, and filter out noise outside this band. You can select which bandpass filter to use from the **Bandpass filter** drop-down menu. The corner frequencies of the pass band of the filter are determined by the Nyquist frequency, which is derived from the sampling rate of the triggering data. The three filter options have pass bands between 10% and 90%, between 20% and 90% and between 50% and 90% of the data's Nyquist frequency, respectively.

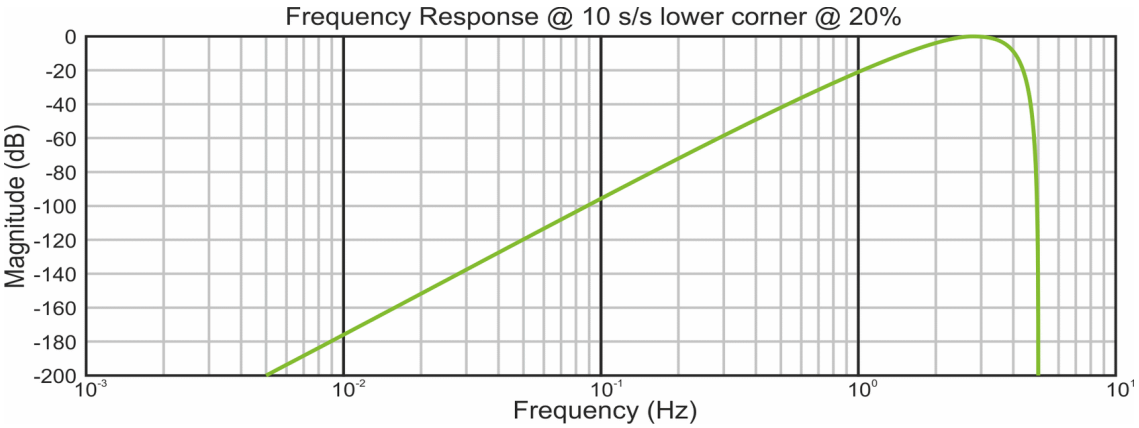
The possible filter configurations are shown in the following table:

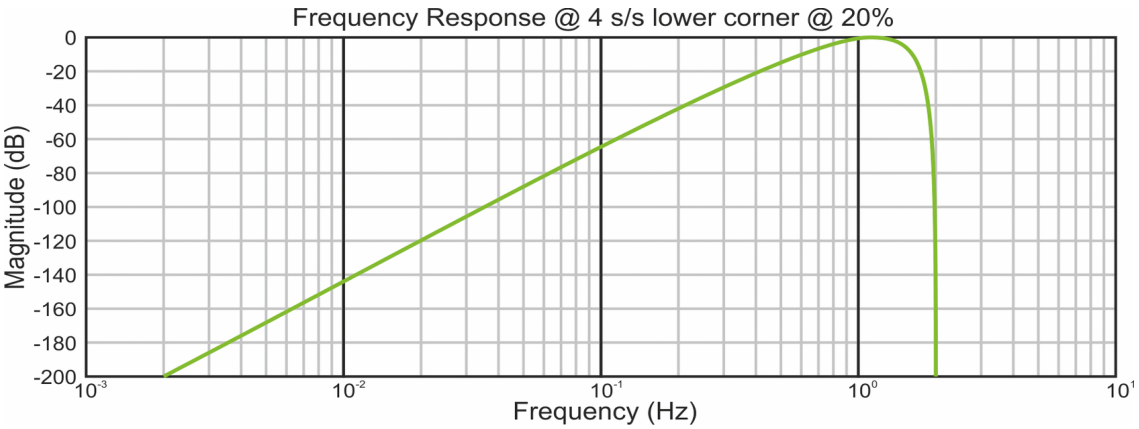
Rate (samples/s)	Bandwidth 1 (Hz)	Bandwidth 2 (Hz)	Bandwidth 5 (Hz)
1000	50 – 450	100 – 450	250 – 450
500	25 – 225	50 – 225	125 – 225
400	20 – 180	40 – 180	100 – 180
250	12.5 – 112.5	25 – 112.5	62.5 – 112.5
200	10 – 90	20 – 90	50 – 90
125	6.25 – 56.25	12.5 – 56.25	31.25 – 56.25
100	5 – 45	10 – 45	25 – 45
50	2.5 – 22.5	5 – 22.5	12.5 – 22.5
40	2 – 18	4 – 18	10 – 18
25	1.25 – 11.25	2.5 – 11.25	6.25 – 11.25
20	1 – 9	2 – 9	5 – 9
10	0.5 – 4.5	1 – 4.5	2.5 – 4.5
8	0.4 – 3.6	0.8 – 3.6	2 – 3.6
5	0.25 – 2.25	0.5 – 2.25	1.25 – 2.25
4	0.2 – 1.8	0.4 – 1.8	1 – 1.8
2	0.1 – 0.9	0.2 – 0.9	0.5 – 0.9
1	0.05 – 0.45	0.1 – 0.45	0.25 – 0.45

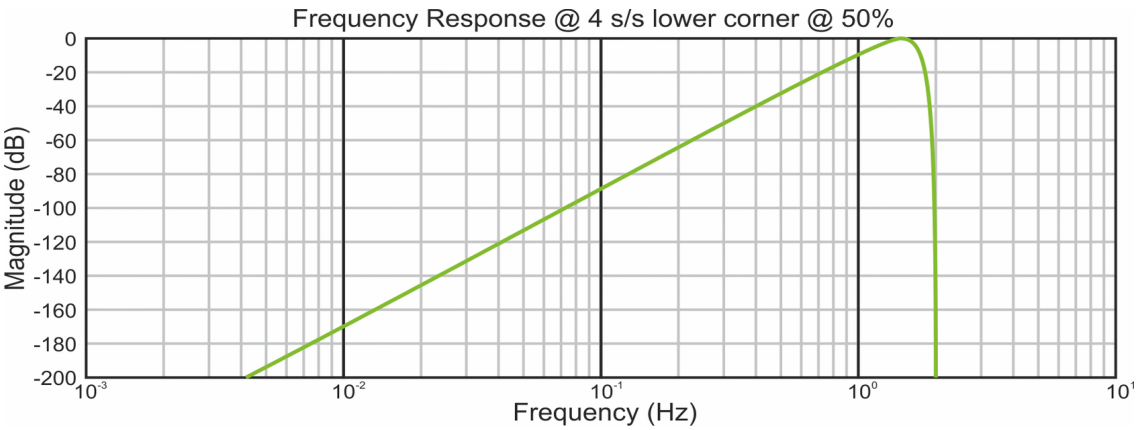
As can be seen, the filters you can choose are determined by the sample rate.

The spectral amplitudes for the various frequency responses available are shown in the following figures.


Frequency Response @ 10 s/s lower corner @ 10%
Magnitude (dB)
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
10-3
10-2
10-1
100
101
Frequency (Hz)


Frequency Response @ 10 s/s lower corner @ 20%
Magnitude (dB)
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
10-3
10-2
10-1
100
101
Frequency (Hz)


Frequency Response @ 4 s/s lower corner @ 20%
Magnitude (dB)
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
10-3
10-2
10-1
100
101
Frequency (Hz)


Frequency Response @ 4 s/s lower corner @ 50%
Magnitude (dB)
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
10-3
10-2
10-1
100
101
Frequency (Hz)

5.1.5 Level triggering

Using the *Level* triggering method, a trigger is generated whenever one of the checked components reaches a certain level above the baseline. You can select which tap is monitored from the **Data source** drop-down menu, and the channel(s) to be considered from the **Channel** column of the table. The values in the **Level** column are the number of counts above the baseline that channel must reach before a trigger is generated.

As with the *STA/LTA* method, the values of the **Level** will often be the same for all checked channels. If you want to use different values for some channels, you should clear the Common values check-box before altering them.

Once you have enabled the *Level* triggering method on a particular channel, you can use the *Control* window to change the level at which the system triggers without restarting the digitiser.

5.1.6 External triggering

When a digitiser or digital sensor triggers, it signals the fact to connected devices, as well as sending any extra data that it has been configured to transmit. You can configure other digitisers to respond to this signal by triggering in turn. This is an option which you can specify at the time of manufacture. See Section 7.4 on page 106, for details of how to set up an installation for external triggering.

As an example, to instruct a stand-alone digitiser with digital inputs to respond to triggers generated by an attached digital sensor:

1. Open the *Configuration setup* window for the digital sensor, and check **Enable External Trigger Output** - this causes it to send triggers to connected devices;
2. **UPLOAD** the new configuration to the digital sensor;
3. Open the *Configuration setup* window for the digitiser, and check **Enable External Trigger Input** - this causes it to listen for triggers coming from the digital instrument, and to record data from any attached analogue instruments when it receives one (depending on its *Output control* configuration); then
4. **UPLOAD** the new configuration to the digitiser.

If a digitiser has both **Enable External Trigger Output** and **Enable External Trigger Input** selected, it will record data when it receives an external trigger as if it had triggered itself, but it will *not* relay the trigger on to other digitisers. It will only send a trigger message if its *own* triggering criteria are satisfied. This prevents triggering loops.

5.1.7 Pre-trigger and post-trigger recording

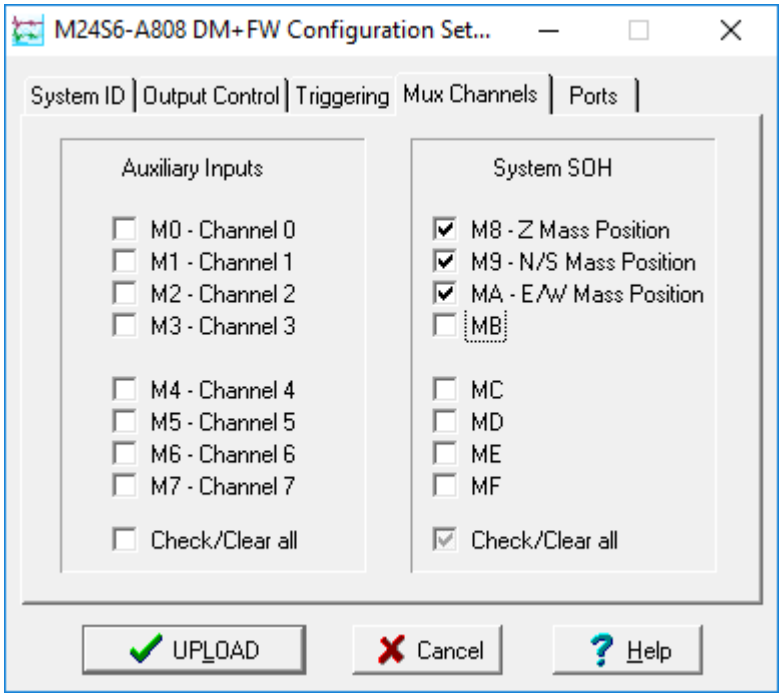
In order to capture all of a seismic event, it is often useful to be able to record data immediately preceding the trigger. Güralp digitisers have an internal buffer of some seconds which allows these data to be added to the triggered stream. Pre-trigger data are particularly useful for emergent-type signals, where the system does not trigger until one phase after the first arrival. In addition, to ensure that the coda of each event is included, some seconds of data are recorded after the system detriggers.

The two boxes at bottom right of the *Triggering* pane allow the user to set the pre-trigger and post-trigger data intervals, in seconds. These values determine the *minimum* length of time during which data will be saved before the trigger condition occurs, and after it has lapsed. Regardless of the intervals chosen, the data in the triggered streams will begin on a whole second boundary.

5.1.8 Mux Channels

Güralp digitisers provide a range of slow-rate auxiliary channels for reporting the system's state of health and other diagnostic information, known as multiplexed ("Mux") channels. The number of Mux channels depends on the model and configuration of your digitiser. Generally, three channels are used to report the sensor mass position, and another measures the internal temperature of the digitiser. In addition to these, up to 12 Mux channels may be supplied for the user's own purposes. Some digitisers have a separate *AUXILIARY* port which can be used to access these channels.

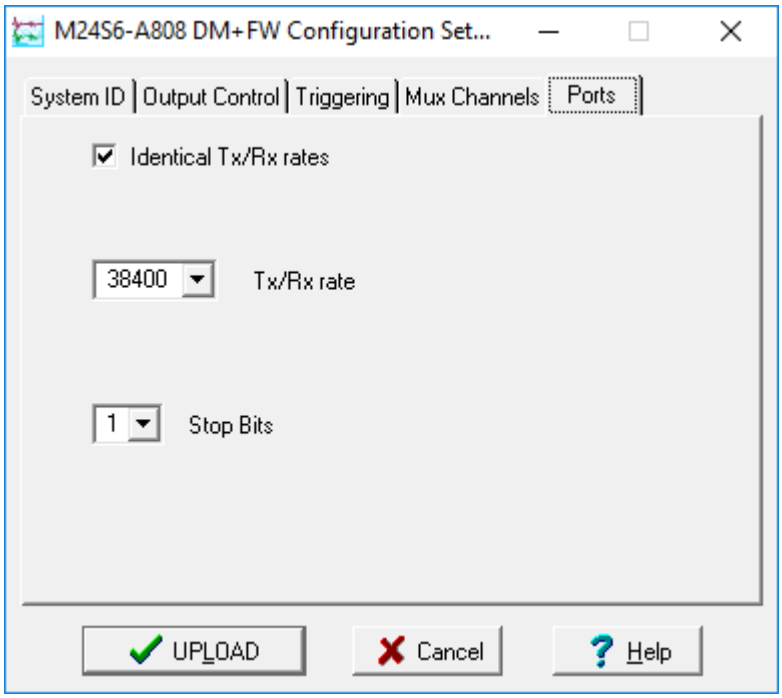
The collection and transmission of Mux channels is controlled using the *Mux Channels* pane:



If a tick is placed in the box next to a channel, its data will be collected and transmitted as a data stream in GCF format, just as with the normal data channels. To indicate that the data come from a Mux channel, the Stream ID will take the form `****MX`, where `M` stands for Mux and `X` is a hexadecimal integer (*i.e.* 0 – 9, and A – F for 10 through 15). Z**, N/S** and **E/W Mass Position** Mux channels appear as `M8`, `M9` and `MA` respectively.

5.1.9 Ports

The *Baud Rates* pane of the *Configuration setup* window allows you to program the baud rate and stop bits for the DM24's output port.



The baud rate you choose must satisfy two conditions:

- It must be high enough to allow all the transmission of all data generated by the digitiser at the sampling rates you have chosen. For three streams of data at 100 Hz, for example, 9600 baud will usually be sufficient. If you wish to transmit 200 Hz data, however, the baud rate must be at least 19200.
- It must be low enough to fit within the bandwidth of the communications infrastructure you are using. While modern modems often offer transfer rates up to 56k baud, the telephone or transmission lines may not support these rates. The same holds true for radio communication links.

Usually, the transmit and receive rates of the data port will be the same. If not, you may select different data rates by removing the check in the box marked **Identical TX/RX rates**.

The **Stop Bits** option allows you to choose whether the serial link uses 1 or 2 stop bits. In most cases this can be left at 1, although 2 may be required if you are sending data over 'difficult' transmission lines (for example, some types of radio link). Using 2 stop bits will add a 10% overhead to the data.

You will also need to set the data rate for Scream!'s local serial port, as well as for the EAM or other communications device (if you are using one). In Scream!, you can configure a serial port by right-clicking on its icon (*not* that of the digitiser) and

selecting **Configure...** from the pop-up menu: for more details, consult the online help or user guide for Scream!. If you are using an additional communications device, you should consult its documentation to learn how to set its baud rate.

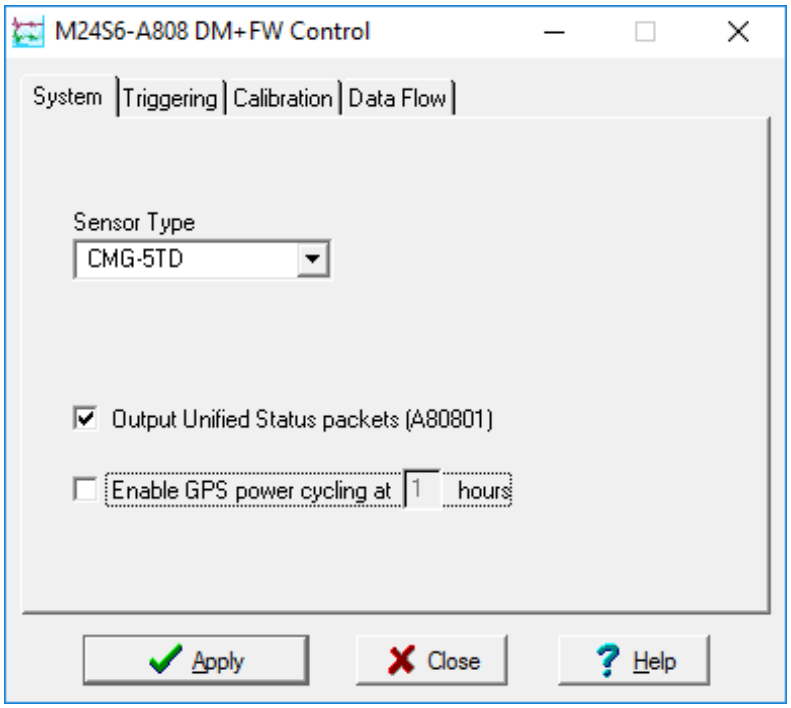
5.2 Controlling digitisers

To control a digitiser whilst it is running, either right-click on the digitiser's icon in the list to the left of Scream!'s main window ( ) and click **Control...**, or simply double-click the entry. Scream! will then contact the digitiser and retrieve its current status - a process which will take a few seconds - after which the *Control* window will be displayed. Once you are happy with any changes you have made in the *Control* window, click Apply to send them to the digitiser, where they will take effect immediately.

This chapter describes the control options available to you for the DM24. Many of these options will also be available for other Güralp digitisers. For the most accurate information, you should consult the Operator's Guide for the digitiser or sensor you connect.

5.2.1 System

When the *Control* window is first opened, it will be showing the *System* pane.

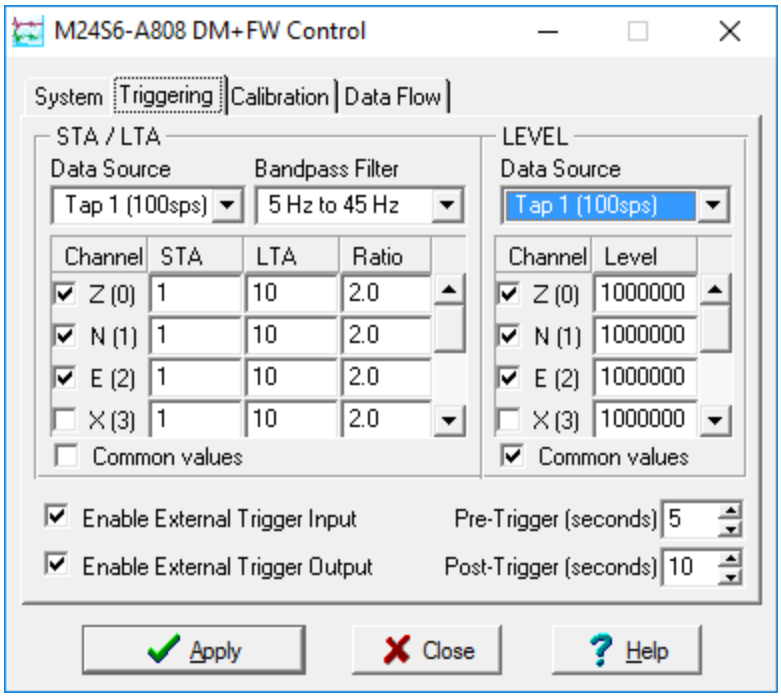


Sensor Type: If the sensor attached to the digitiser is a Güralp velocity sensor, you can send control commands to it from the *Mass Control* tab (see below). Which functions are available on this tab depends on the *Sensor Type* you have set here.

If you change the **Sensor Type**, you will have to click Apply, close the *Control* window, and open a new one before you can access the *Mass Control* options.

Enable GPS power cycling: If you are using a GPS unit to receive time signals, but do not experience significant drift in the system's clock (for example, in a stable-temperature environment), you can save power by selecting **Enable GPS power cycling.** With this option in use, the GPS time is only checked at intervals of the specified number of hours and the GPS receiver is powered down in between checks. When the time is checked, the GPS receiver is powered up for as long as it takes to obtain a satellite "fix" and both synchronize and stabilise the internal clock. Under extreme adverse conditions, ensuring the internal clock is stabilised may take longer than the power cycling interval, in which case the GPS receiver will remain powered up throughout.

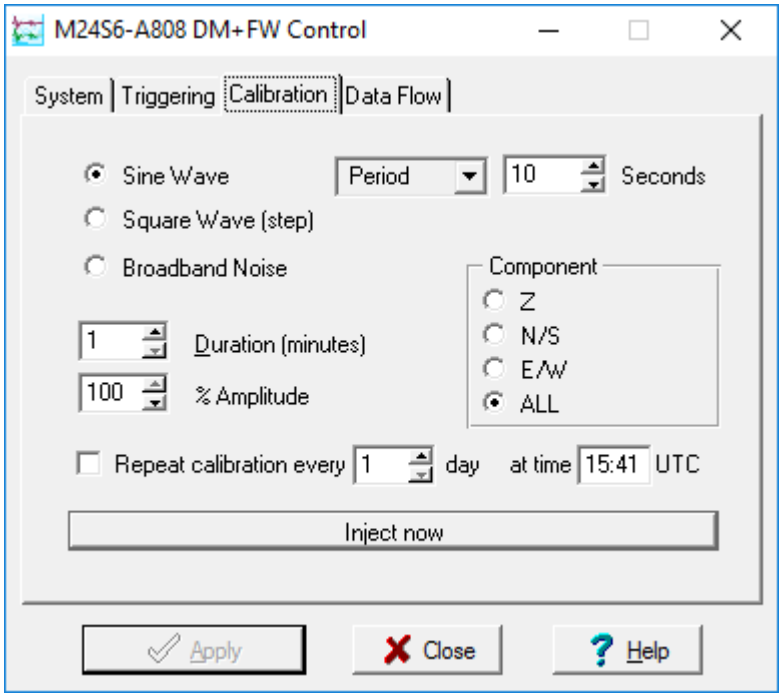
5.2.2 Triggering



The *Triggering* pane is very similar to the corresponding pane of the *Configuration setup* window (see section 5.1.3 page 40), although not all of the options described there are available here, since some require rebooting the digitiser.

5.2.3 Calibration

You can check that your instrumentation is correctly calibrated by injecting known signals into the sensor's feedback loop. The *Calibration* pane allows you to do this once the sensors are installed.



Each channel **Z, N/S** and **E/W** can be calibrated separately. For most triaxial digital instruments, each channel calibrates the corresponding axis of the instrument; simply select one of the **Z, N/S** and **E/W** check boxes to calibrate that axis. Some instruments use only one calibration loop, which is reproduced for all three components: if you have one of these instruments, you should select **Z** to calibrate the sensor.

The calibration signal is digitized at the same rate as the *highest-rate* data stream you have selected for either continuous or triggered output. It is returned on a channel ending Cn (n = 0 – 7). The auxiliary analogue channel is interrupted whilst calibration is active.

The **Duration** box tells the digitiser how long to provide the calibration signal before disconnecting. This avoids the system being inadvertently left in calibration mode. The default is 2 minutes.

All Güralp digitisers can produce either sine-wave or square-wave (step) calibration signals; newer models can also carry out broadband noise calibration. The **Sine wave** calibration signal always starts and stops on the zero crossing. The frequency or period is given by the boxes at the top, right-hand side of the dialogue. Only integers between 1 and 10 may be specified for either frequency or period, so to generate a 0.5 Hz signal you should select **Period** and set the time to 2 (seconds). Likewise, if you require a 0.25 second period you should select **Frequency** and set the

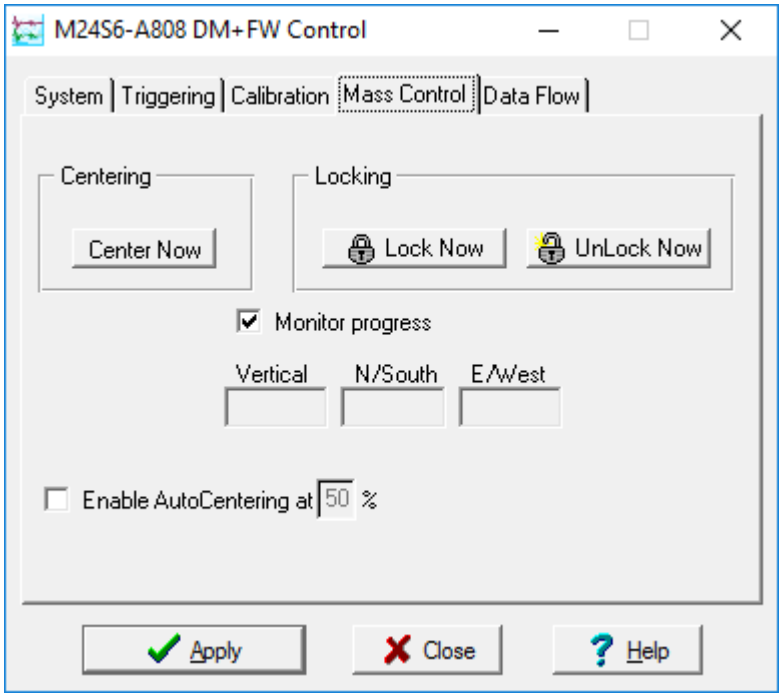
rate to 4 (Hz). In this manner, you can select frequencies ranging from 0.1 to 10 Hz (10 to 0.1 s periods).

You can specify step calibration by selecting the **Square wave** button. The square wave consists of a positive step at the start of the next minute of the digitiser's internal clock, followed by a negative step after a specified number of minutes. After a further delay of the same number of minutes, the calibration signal is disconnected. The default is 2 minutes. The **Period** and **Frequency** settings are ignored.

The **Broadband Noise** calibration signal consists of a constant stream of white noise, which lasts for the specified number of minutes. The **Period** and **Frequency** settings are ignored.

5.2.4 Mass Control

Many Güralp instruments respond to control signals to centre, unlock, and lock the sensor masses. These signals are generated by the digitiser. You can tell the digitiser to send a signal using the Center Now, Lock Now and UnLock Now buttons on the *Mass Control* tab.



If you select **Monitor progress**, the window will remain open during the process, and the mass positions will be displayed in the `Vertical`, `N/South` and `E/West` boxes. If you leave **Monitor progress** unselected, the window will close. You can still monitor mass positions by selecting the appropriate streams and opening a WaveView window for them.

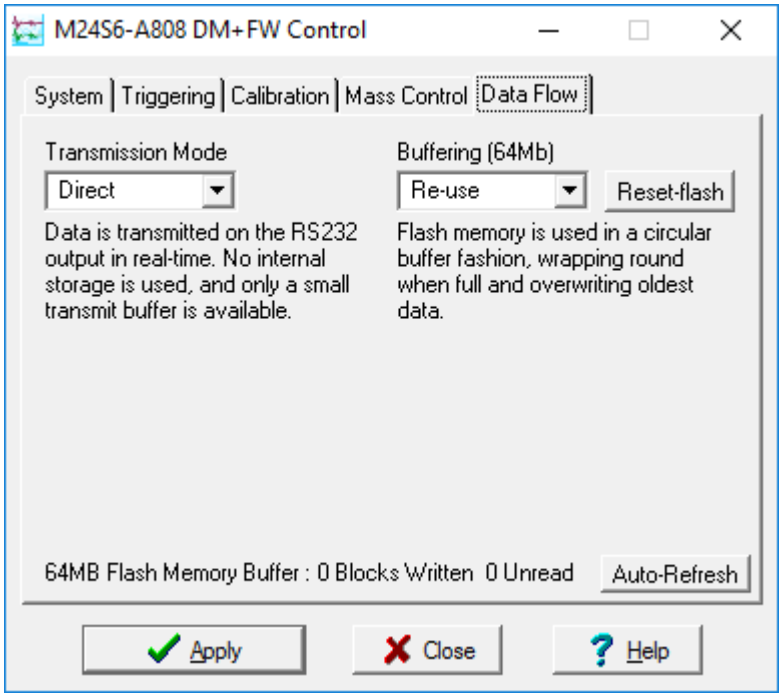
Depending on which Sensor Type you have chosen in the System tab (see above), not all the control signals will be available. For example, a Güralp 3ESP can be centred

from Scream, but must be locked and unlocked manually, whilst a 3T has remote lock and unlock commands. Conversely, Güralp 5T strong-motion instruments must be centred manually, and do not need locking or unlocking. If you have set the Sensor Type to CMG-5T or CMG-5TD, the Mass Control tab will not be available.

5.2.5 Data flow

The DM24 operates in one of several *transmission modes* and, independently, in one of several *buffering modes*. These modes relate to how the unit uses its Flash memory and are described fully in section 4.3 on page 19.

To select transmission and buffering modes in Scream!, right-click on the digitiser, then click on **Control...** and navigate to the **Data Flow** pane:



Select the desired Transmission mode to use the flash memory:

- as a simple data store, from which you can request data (*FILING* and *DUAL* modes);
- as a buffer holding unacknowledged blocks, which are transmitted in preference to real-time data (*FIFO* mode);
- as a buffer holding unacknowledged blocks, which are transmitted whenever the channel is free but no real-time data blocks are ready (*ADAPTIVE* mode);
- not at all (*DIRECT* mode).

Select the desired Buffering mode, which tells the unit what to do when its Flash memory becomes full: either

- carry on, overwriting the oldest data held, or

- stop writing and switch the DM24 into *DIRECT* mode.

Clicking  in this window immediately activates the transmission mode you have selected—there is no need to reboot.

If you prefer, you can use the DM24 command line to switch between transmission modes, as described in section 6.11 on page 82.

6 Command line interface

6.1 Introduction

You can connect to the internal software of the DM24 over its output serial port and communicate with it.

To enter command mode from Scream!, right-click on the digitiser's icon () and select **Terminal...** from the menu that pops up. A window will open, and once the digitiser and computer are communicating properly you will see the prompt

```
ok
```

If you prefer, you can use a terminal program on your computer (such as `minicom` on Linux, or PuTTY on Microsoft Windows) to connect to the DM24.

Whilst you are in terminal mode, data transfer will be interrupted; the digitiser may use any Flash memory as a temporary store, depending on how you have configured it. Some commands, such as `SET-TAPS`, require a reboot to take effect.

When you are finished with your command-line session, simply close the Scream! or emulator window. Scream! will automatically issue the `GO` command to resume data-stream transmission. The session will, in any case, time-out after a minute. If this happens while you are using `minicom` or PuTTY, and the digitiser starts transmitting stream data (which will appear as bursts of unintelligible characters), key Ctrl + S to interrupt the streams and resume your session.

Güralp EAM modules also allow you to send commands direct to the digitiser using the command-line tool `data-terminal`. Older DCMs and AMs used the command-line tool `gcli`. For more information, please see the manual for your data module.

If you have problems connecting to the digitiser's console, you should check that the serial port's options and baud rate are set correctly in Scream! or your terminal program. As supplied, the DM24 expects connections at 19200 baud, with eight data bits, no parity bit and one stop bit. No flow control is used: neither hardware nor software.

6.1.1 FORTH

The DM24 uses a FORTH interpreter to implement its features. To issue a command in FORTH, you must supply the arguments before the command, for example:

```
3 SENSOR-TYPE
```

In FORTH, anything you enter is termed a *word*. New words (case insensitive) are placed on a stack. Some words are known to the system, and may represent commands; if a command finds itself at the top of the stack (e.g. because it is the last

thing you typed), they will execute, remove themselves from the stack, and then remove further items from the stack to use as arguments. Thus, in the command

```
3 SENSOR-TYPE
```

the 3 has no immediate effect, so stays on the stack; `SENSOR-TYPE` takes itself and the previous item (here `3`) off the stack, then performs its action on the `3` (here, setting the configured sensor type to 3, representing a Güralp 3T sensor).

If a command completes with nothing remaining on the stack, the digitiser will show the prompt `ok`. Otherwise, the prompt will remind you that the digitiser is waiting for you to complete the command. Some commands, such as `SAMPLES/SEC`, clear the stack automatically after they execute.

Throughout this manual, examples are given of command-line interactions. In these examples, a fixed-width typeface will be used:

```
Example of the fixed-width typeface used.
```

Commands that you are required to type will be shown in bold:

`Example of the fixed-width, bold typeface.`

Some commands are interactive, and will ask you to provide extra information after you execute them. Where data that you type may vary depending on your individual configuration, such as parameters to commands, these are shown in italics:

`Example of the fixed-width, bold, italic typeface.`

Putting these together into a single example:

`System prompt:` ***`variable parameters`*** **`command`**

6.2 Getting help

The DM24 has an on-board help system which can be accessed with the commands `HELP` and `EXPLAIN`.

6.2.1 HELP

Syntax: `HELP`

Lists all the commands which are recognized by the digitiser.

If the command you want is not listed:

- The command may be in the extended dictionary. This dictionary contains certain advanced commands, as well as exposing internal digitiser functions and variables. To enable the extended dictionary, issue the command `ok-1` (that's a number one, not a lower-case 'L').

When you have finished using the extended dictionary, close it with the command `[seal]` (including the square brackets), so you do not inadvertently change the internal settings with subsequent commands.

- *Alternatively*, the command may not be implemented in the firmware version you have installed. Check the Güralp Systems web site for updated firmware.

6.2.2 EXPLAIN

Syntax: `EXPLAIN` *`command`*

Displays a short message describing the syntax of *`command`* and its effect.

6.3 General configuration

6.3.1 SET-ID

Syntax: `SET-ID` (interactive)

Sets the system identifier and serial number of the DM24 to values you supply.

```
SET-ID
System Identifier ? {ALPHA} DM24,
Serial # ? (C23300) C902,00
DM24 C90200 CMG-3T  ok_C902
```

The system identifier you supply may contain up to 5 alphanumeric (0-9A-Z) characters and should be followed by a comma (','), which is not treated as part of the serial number. The system identifier can not start with a zero ('0') character.

The serial number you supply must contain exactly 4 alphanumeric (0-9A-Z) characters followed by a comma and two zeroes (0) in almost all circumstances. The first four characters are copied verbatim into stream IDs and the final two characters are used as the "base" for generating stream identifiers. All stream IDs used in this manual assume that these two characters are zeros. The serial number can not start with a zero ('0') character.

The system responds by printing the new system identifier and serial number, the sensor type and the "ok" prompt (which is now derived from the new serial number). If the sensor type has not been set, the word `NOTSET` is printed in its place.

6.3.2 SERIAL2

Syntax: `SERIAL2` (interactive)

Sets the serial number for a second sensor (for seven-channel digitisers) to a value you supply.

```
SERIAL2
Secondary Serial # {0} ? 6 characters : C903,00
```

The serial number you supply must contain exactly four alphanumeric (0-9A-Z) characters followed by a comma and two zeros (which are ignored - see the discussion of serial numbers at `SET-ID`, above).

6.3.3 RESETLABELS

Syntax: `RESETLABELS`

Clears any second serial-number and resets any custom stream labels so that, for example, the first tap of the vertical component of the first sensor ends Z0 and the first tap of the vertical component of the second sensor ends Z1.

6.3.4 SENSOR-TYPE

Syntax: *`type`* `SENSOR-TYPE`

Tells the DM24 which kind of sensor is attached to it. This affects whether or not the digitiser exposes commands such as locking and centring.

`type` can be one of

- `1`, for 40T sensors,
- `2`, for 3ESP sensors,
- `3`, for 3T analogue sensors, or
- `4`, for 3TD sensors.

6.3.5 AZIMUTH

Syntax: *`instrument angle`* `AZIMUTH`

Some borehole digitisers can rotate incoming signals algorithmically as they are digitized. This feature allows you to compensate for an instrument being out of alignment with the North/South axis. To use it, first measure the orientation of the sensor, then connect to the digitiser and issue an `AZIMUTH` command, giving the angle you have measured.

`instrument` tells the digitiser which instrument to apply the rotation to; `0` for sensor A or 1 for sensor B.

angle is the measured angle of deviation from the North/South axis, in tenths of a degree. The angle of deviation is the *negative* of the angle by which you want incoming signals to be rotated. Like all other parameters, it must be an integer (*i.e.* a whole number of tenths of a degree).

You can measure the orientation of a borehole sensor by placing a reference sensor nearby and carrying out coherence calculations. An extension to Scream! is available which can do this for you. For details, see the manual for your sensor or for Scream!.

The AZIMUTH command only alters the digitiser's CMOS settings. You will need to re-boot the digitiser to transfer them to the DSP.

6.3.6 CROSS

Syntax: *channel* CROSS

Alters the DSP's transformation matrix to invert one of the digitiser channels. *channel* is one of 0 (vertical), 1 (N/S), 2 (E/W) and 3 (auxiliary).

The CROSS command only alters the digitiser's CMOS settings. You will need to re-boot the digitiser to transfer them to the DSP.



Note: If you need to set both AZIMUTH and CROSS, you must issue AZIMUTH first, since it overwrites the matrix.

6.4 GPS, GNSS and timing

6.4.1 GPS-TYPE

Syntax: *type* GPS-TYPE

Tells the DM24 which kind of GNSS receiver is attached to it.

type can be one of

- 0, if no receiver is available, or
- 2, for attached GNSS receivers using the NMEA protocol at 4800 Baud
- 4, for attached GPS equipment using the NMEA protocol at a different Baud-rate. This is required for GLONASS-capable receivers which need to operate at 38,400 Baud in order to accommodate all of the NMEA sentences which need to be transmitted within each second. The Baud rate must be set manually using the BAUD command in this case.

6.4.2 HR-CYCLE

Syntax: *interval* HR-CYCLE

Sets the time to wait between GPS synchronizations.

To save power, the DM24 can switch off the attached GPS receiver once its internal precision clock has been successfully synchronized. The GPS receiver is reactivated every *interval* hours to ensure that the clock remains correct, and shut down again once the clock error has stabilized.

Setting *interval* to zero disables this function, *i.e.* the GPS will be powered all the time. This is the best setting to use in installations with access to mains (outlet) power. The maximum allowed value for *interval* is 23.

6.4.3 SET-CLOCK

Syntax: SET-CLOCK (interactive)

The DM24 has two clocks: a hardware real-time clock (RTC), which runs continuously (even when the unit is powered down) and a software clock that is used for time-stamping digitised data. The software clock is set from the RTC at boot-up: this is the only time that the system consults the RTC. Both clocks are set whenever the DM24 synchronises to GPS time.

The SET-CLOCK command sets the RTC. It does not affect the software clock.

```
SET-CLOCK
Enter Date & Time -
YYYY MM DD HH MM SS
2006 02 01 12 53 25  Clock set to 2006  2  1 12:53:27
ok_SBHY
```

The time should be entered in the form year month day hour minute second, padding each field with zeroes so that they line up with the guide above.

If the DM24 does not recognize the time format you have used, it will output the message:

```
Invalid Time Entry.
```

The RTC time can be copied to the software clock by rebooting the system.

This setting will be overridden when the GPS system next synchronizes the clocks.

6.4.4 TIME?

Syntax: `TIME?`

Displays the current date and time from the software clock that is used to time-stamp digitised data. See `SET-CLOCK` for more details.

6.4.5 getWNROpivot

Syntax: `getWNROpivot`

Prints the configured GPS week-number roll-over pivot date. See `setWNROpivot`, below, for more details.

6.4.6 setWNROpivot

Syntax: *`YYYY MM DD`* `setWNROpivot`

This command sets the "pivot date" which will be used for any required GPS week-number roll-over (WNRO) corrections, where:

- *`YYYY`* specifies the four-digit year;
- *`MM`* specifies the month, with `01` representing January and `12` representing December; and
- *`DD`* specifies the day of the month.

If an attached GPS receiver reports that the current date is *earlier* than the pivot date specified with this command, the digitiser will assume that the receiver is affected by the WNRO problem. The digitiser will add one or more multiples of 1024 weeks to the reported date until the result is later than the specified pivot date. This modified date is then used for time-stamping data. See section 7.2.3 on page 100 for more details.

For best results, specify a date in the very recent past and repeat every nineteen years.

6.4.7 LEAPSECOND

Syntax: *`yyyy mm dd`* `LEAPSECOND`

Manually notify the digitiser of an upcoming leap second. This command serves two purposes.

When a positive leap-second occurs, the GPS receiver will signal this fact by sending the special time 23:59:60. The digitiser recognises this and all data will be time-stamped appropriately. If for any reason, the digitiser does not receive this time - the

system may, for example, be configured for GPS power-cycling and the receiver may be powered down - the digitiser will produce data with incorrect time-stamps until it can resynchronise to the GPS receiver again. If the digitiser is notified in advance about an upcoming leap-second, this situation can be avoided and the leap-second will be handled correctly even if GPS data are unavailable at the time.

In addition, some older GPS receivers - those based on Trimble Lassen SQ modules - were supplied with a firmware revision that did not properly implement leap-seconds. See `SQPATCH`, below.



Note: The leap second is taken to be at the *end* of the day ***yyyy-mm-dd***.

6.4.8 SQPATCH

Syntax: `SQPATCH ENABLE` or `SQPATCH DISABLE`

Enables or disables the internal patch for older GPS receivers based on Trimble Lassen SQ units with faulty firmware. These units misinterpreted the GPS system's advance notification of a leap second and, consequently, reported times which were one second out during the period between the leap second announcement and the actual leap second itself.

With `SQPATCH` enabled, the time reported by the digitiser is offset by 1 second to counteract this problem. If you have set `LEAPSECOND`, above, `SQPATCH` will automatically be disabled when the leap second occurs and the digitiser will then run normally. GPS receivers with the latest firmware do not suffer from this problem.

To find out whether `SQPATCH` is currently enabled, issue the command

```
.SQPATCH
```

6.5 Port configuration

6.5.1 BAUD

Syntax: `port baud-rate BAUD`

Sets the baud rate for one of the serial ports on the DM24, in bytes per second. `port` can be:

- `0`, for the DATA OUT port;
- `1`, for the GPS port; or
- `2`, for the DATA IN port.

For example,

```
0 38400 BAUD
1 4800 BAUD
2 38400 BAUD
```

This will reset a standard single-sensor DM24 to its default configuration.

The allowable values for `baud-rate` are 4800, 7200, 9600, 14400, 19200, 38400, 57600, 115200 and 230400. For compatibility reasons, 1152 is also accepted, setting the baud rate to 115200.

Note: The GPS port (port 1) should always be set to 4800 baud unless a GLONASS-compatible receiver is being used.

6.5.2 STOPBITS

Syntax: `port number-bits STOPBITS`

Sets the number of stop bits used for one of the serial ports on the DM24. `port` can be:

- `0`, for the DATA OUT port;
- `1`, for the GPS port; or
- `2`, for the DATA IN port.

By default, the DM24 uses one stop bit on all ports.

Note: This command has no effect on MkIII DM24s, which always use one stop-bit.

6.5.3 +MODEM

Syntax: `+MODEM` (interactive)

Instructs the DM24 to use a modem connected to the *DATA OUT* port to contact a remote device and transmit data stored in the Flash memory.

The DM24 can dial out every time it generates a heartbeat message, and also whenever a trigger occurs. You can change the interval between heartbeat messages with the `HEARTBEAT` command, below.

When you issue `+MODEM` for the first time, the digitiser will prompt you to enter a telephone number:

```
+MODEM
Phone# is not set Enter phone# 00441189819056
Phone# 00441189819056 Flash FIFO : Circular  Configuration
for dial out  ok_DEMO00
```

If you have already configured a telephone number, the command will print out the current setting and ask if you want to change it.

```
+MODEM
Phone# 00441189819056 New number? y/n y
Enter phone# 00441189819943
Phone# 00441189819943 Flash FIFO : Circular  Configuration
for dial out  ok_DEMO00
```

`+MODEM` also changes the transmission mode of the digitiser to `FILING`, so that all incoming data are stored in Flash memory, ready to be downloaded. When the dial-out takes place, all new data in Flash memory are transmitted. By configuring the DM24 to transmit only triggered streams, you can create a system which "wakes up" on a trigger, connects to a data centre using the modem, and delivers the triggered data in near real time.

6.5.4 –MODEM

Syntax: `–MODEM`

Disables automatic modem dial-out. Note that this command does not change the current transmission mode so, if you have previously issued `+MODEM`, the DM24 will be left in *`FILING`* mode.

If you issue `–MODEM` over a dial-out modem link, you should make sure you have arrangements already in place for contacting the DM24 by other means, since the command will cause the connection to close.

6.5.5 +GCFIN

Syntax: *`input-port`* `+GCFIN`

Instructs the digitiser to accept GCF input on either the *GPS* or the *DATA IN* port. *`input-port`* can be:

- `1`, for the *GPS* port; or
- `2`, for the *DATA IN* port.

The GCF streams received will be combined with the DM24's own streams and output with them through the *DATA OUT* port or saved to Flash memory, according to the current transmission mode.

Whilst GCF input is active, the normal function of each port (GPS or console access) is disabled.

6.5.6 –GCFIN

Syntax: *input-port* -GCFIN

Instructs the digitiser to stop accepting GCF input on *input-port*, and to return the port to normal operation.

6.5.7 OPEN

Syntax: OPEN *system-id stream-id-prefix* (n.b.)

Opens a terminal session to a second, connected digitiser.

The DM24 scans all the ports which are currently accepting GCF input (see +GCFIN, above) to find a digitiser which matches *system-id* and *stream-id-prefix*.

For *stream-id-prefix* you should use the first 4 characters of the Stream IDs produced by the required digitiser, *i.e.* without the component and tap designations.

A terminal connection is then made to the second digitiser, allowing you to issue commands.

To finish the session and return to the first digitiser's terminal mode, use the command GO.

6.5.8 USBPORT



Note: This command is only available on units with the optional USB interface.

Syntax: *port* USBPORT

Instructs the digitiser to redirect input or output from port *port* to the USB interface. You can then attach the *USB* port to your computer with a suitable cable, where it will appear as an additional serial port with a significantly faster data transfer rate. The value for *port* can be:

- 0, for the *DATA OUT* port;
- 1, for the *GPS* port; or
- 2, for the *DATA IN* port.

The USB interface is provided by an internal FTDI245 chip, which is compatible with Microsoft Windows, Linux and Apple Macintosh systems.

You can use the USB port only to connect to a computer, not directly to a peripheral (*e.g.* a hard disk). If you want to download data to a hard disk, you should specify the FireWire interface instead.

6.6 Output configuration

6.6.1 SAMPLES/SEC

Syntax: *tap-0 tap-1 tap-2 tap-3* samples/sec

The DSP software on the DM24 supports up to 7 cascaded filter/decimation stages. At each stage, the sample rate can be divided by a factor of 2, 4 or 5. The internal ADC outputs data at 2000 samples per second, so decimated data streams are available from 1,000, 500 and 400 samples/sec down to 1 sample/sec.

You can specify the sample rate to use at four of these stages, known as *taps*. You can also take output from any of these four stages. The DM24 will arrange the remaining stages according to your settings.

The arguments *tap-0* to *tap-3* are the sample rates at each tap in turn, starting with the highest. You must ensure that each rate is lower than the previous one by a factor of 2, 4, 5, 8 (= 2 then 4), 10 (= 2 then 5) or 16 (= 4 then 4). Non-integer values are not allowed.

For example:

```
1000  125   25    5   samples/sec
1000  500   100   10  samples/sec
500   100   20    4   samples/sec
400   40    10    5   samples/sec
```

As long as you specify the initial taps, you can omit later ones. The command fills in the value of the missing taps, using a decimation factor of 2 where possible. Thus, the following commands are equivalent:

```
400 40 20 10 samples/sec
400 40 samples/sec
```



Note: Older DM24 software did not support the new GCF format for higher sample rates. It also did not properly support 250 samples/s, nor rates below 1 sample/s.

6.6.2 CONTINUOUS

Syntax: *tap components* CONTINUOUS

Sets which components are output under normal conditions, and at which tap(s).

tap is the tap number at which to output the triggered stream. You can set which taps output which sample rate using the SAMPLES/SEC command, described above.

components is an integer below 16, whose binary bits represent the *Z* (1), *N* (2), *E* (4) and auxiliary (8) components respectively.

Thus, for example:

- ***0 1* CONTINUOUS** will output the *Z* component only (1) at the first tap;
- ***0 2* CONTINUOUS** will output the *N* component only (2);
- ***0 4* CONTINUOUS** will output the *E* component only (4);
- ***0 7* CONTINUOUS** will output all three components (1+2+4 = 7);
- ***0 12* CONTINUOUS** will output the *E* component and the auxiliary component (4 + 8 = 12);
- ***0 0* CONTINUOUS** will output nothing at the first tap.

6.6.3 SET-TAPS

Syntax: *tap-0 tap-1 tap-2 tap-3* SET-TAPS

This is an alternative to the CONTINUOUS command, which allows you to set the outputs for all taps simultaneously.

tap-0 to *tap-3* are integers below 16, whose binary bits represent the *Z* (1), *N* (2), *E* (4) and auxiliary (8) components respectively. Each one sets which components are output at that tap under normal conditions.

For example, if you issue

***9 7 0 15* SET-TAPS**

then

- tap 1 will output *Z* and the auxiliary component (1 + 8 = 9);
- tap 2 will output all three directional components (1 + 2 + 4 = 7);
- tap 3 will not output anything; and
- tap 4 will output all four components (1 + 2 + 4 + 8 = 15).

To set triggered output streams, you should use the TRIGGERED command detailed below.

6.6.4 COMPRESSION

Syntax: *bits size* COMPRESSION or NORMAL COMPRESSION

Sets the maximum amount of compression to use. Greater compression means the digitiser packages data more efficiently, so more can be transmitted over a link with a bandwidth restriction. Compressing streams increases data latency because more samples are fitted into each packet.

The digitiser compresses data without loss, so compression is most effective when the data contain relatively little information. During large or local seismic events, the digitiser normally needs to decrease the compression level.

bits can be one of 8BIT, 16BIT and 32BIT. 8BIT (the default) is the maximum amount of compression; 32BIT denotes no compression.

size determines the maximum number of four-byte records in a GCF block. This must be between 20 and 250; the default is 250. Each record can represent 1, 2 or 4 samples, depending on the compression level.

GCF blocks must be a whole number of seconds long. If you set *size* to a value so small that packets would contain data for less than one second, the digitiser will ignore the exact value of *size* and simply output one block every second (per stream).

Thus, if you issue 32BIT 20 COMPRESSION, streams with a sample rate of 20 samples/s and higher will output one block per second, whilst lower rate streams will output 20-sample blocks: every 5 seconds for 4 samples/s data etc. The special command MINIMUM COMPRESSION is a shorthand for this.

Another special command, NORMAL COMPRESSION, returns these settings to their default values, equivalent to 8BIT 250 COMPRESSION.

6.6.5 LOWLATENCY

Syntax: LOWLATENCY

This is a special mode for applications requiring lower sample rate data in as close to real time as possible. In this mode, the digitiser takes a 200 samples/s stream and passes it through low-pass and decimation filters internally, before exposing it as a 20 samples/s stream at an increased rate of 1 block per second.

To enable low latency mode on all three components, first use SAMPLES/SEC to set Tap 0 to 200 samples/s, then issue the following commands:

```
NORMAL COMPRESSION
0 7 CONTINUOUS
0 1 BANDPASS
LOWLATENCY
7 TRIGGERS
```

If you want to enable low latency mode on some components only, replace the 7s above with another value (*e.g.* 1 = *Z*, 2 = *N*, 4 = *E*).

To disable low latency mode, issue

```
0 1 BANDPASS
0 TRIGGERS
```



Note: When `LOWLATENCY` is active, the `TRIGGERS` and `BANDPASS` commands have special meanings. Because of this, the STA/LTA triggering function is disabled. You can still use `LEVEL` triggering if you require (see below).

6.6.6 SETDSS

Syntax: `output-spec SETDSS`

This command alters the way the DM24 outputs its status streams.

In normal operation, the DM24 stores lines of status information until it has filled a 1K GCF block. For low latency applications, however, status information is required more frequently. In this mode, the DM24 will output a status block every minute. This mode was required by the DSS server distributed with Güralp Systems' DCM, which is now obsolete.

`output-spec` is an integer below 32, whose bits tell the digitiser which status lines to output.

- Setting bit 0 (1) produces lines showing the current internal clock offset, frequency error, and GPS fix type. For example:

  ```
  2006  9  4 13:13:00 2 MicroSeconds Fast  Freq error 0 e-9
  Auto 3D [-4]
  ```

- Setting bit 1 (2) produces lines showing the current GPS offset, drift, PWM measure, and fix type. For example:

  ```
  2006  9  4 13:13:00 o/s= 340 drift=  0 pwm= 8382  Fix=
  "3D"
  ```

- Setting bit 2 (4) produces lines showing the GPS latitude and longitude. For example:

  ```
  2006  9  4 13:13:00 Lat= "51'21.6526N" Long="001'09.8148W"
  ```

- Setting bit 3 (8) produces lines like

  ```
  Boot Log : Power_cycles=    11 Watchdog_resets=    51
  Last boot  2006 8 23 08:49:22    2006  8 23 08:50:40 Run
  12 Days 04:22:20
  ```

  showing the number of times the digitiser has been reset, with the time of the last reset and the current up-time.

- Setting bit 4 (16) produces lines showing the current power supply voltage and temperature. For example:

  ```
  2006  9  4 13:13:00 External supply : 14.3V Temperature
  29.75'C
  ```

The bits should be logically ORed (*i.e.* the decimal numbers added) to produce multiple status lines. For example, the command

***6* SETDSS**

(6 = bit 2 OR bit 4) will cause the digitiser to begin outputting status blocks containing GPS timing and position information only:

```
2006  9  4 13:13:00 o/s=    340 drift=     0 pwm= 8382  Fix=
"3D"
2006  9  4 13:13:00 Lat= "51'21.6526N"  Long="001'09.8148W"
```

To use a DM24 with a DCM running a DSS server, you should issue

***1* SETDSS**

To return to normal operation, issue the command

***0* SETDSS**

6.7 Variable Gain

Some DM24s have variable-gain inputs, allowing gains from unity to ×64 to selected individually for each of the main inputs. These digitisers will report the gain settings to the status stream, shortly after boot-up, in a message like

```
Gain Control : ADC#1 : ADC#2
Gain settings :  Ch#0 *16  Ch#1 *16  Ch#2 *16  Ch#3 *2  Ch#4
*1  Ch#5 *1  Ch#6 *1
```

In this example, the Z, N and E channels of SENSOR A each have ×16 gain, the X channel has ×2 gain selected while all channels of SENSOR B have unity gain.

6.7.1 *GAINS

Syntax: *gain* *GAINS

Simultaneously sets all channels to the specified gain.

gain specifies the gain and is one of 1, 2, 4, 8, 16, 32 or 64.

The digitiser must be rebooted before this change will take effect.

6.7.2 *GAIN

Syntax: *channel gain* *GAIN

Set one channel to the specified gain.

channel specifies the input to be adjusted and is one of 0 (vertical), 1 (North/South), 2 (East/West) or 3 (auxiliary/calibration). On seven channel digitisers, this

parameter can also be one of 4 (vertical), 5 (North/South) or 6 (East/West), referring to the components from the second instrument.

gain specifies the gain and is one of 1, 2, 4, 8, 16, 32 or 64.

The digitiser must be rebooted before this change will take effect.

6.8 Triggering

6.8.1 TRIGGERS

Syntax: *components* TRIGGERS

Selects which component or components can generate an STA/LTA trigger. Only these components will be examined by the triggering algorithm.

components is an integer below 16, whose binary bits represent the *Z* (1), *N* (2), *E* (4) and auxiliary (8) components respectively as described above. Issuing ***0 TRIGGERS*** will disable the triggering system.



Note: When LOWLATENCY is active, this command is used to set the components which are output as low-latency streams. Issuing **0 TRIGGERS** will disable the low latency mode.

6.8.2 GTRIGGERS

Syntax: *components* GTRIGGERS

Selects which component or components can generate a level trigger. Only these components will be examined by the triggering algorithm.

components is an integer below 16 with the same meaning as in TRIGGERS.

6.8.3 S/WTRIGGER

Syntax: S/WTRIGGER

Issues a software trigger, causing the digitiser to trigger immediately, and begin outputting triggered streams according to its configuration.

6.8.4 TRIGGERED

Syntax: *tap components* TRIGGERED

Selects which component or components will be *output* when a trigger is generated, and at which tap (sample rate).

tap is the tap number at which to output the triggered stream. You can set which taps output which sample rate using the SAMPLES/SEC command, described above.

components is an integer below 16, which represents which components to output in the same fashion as in the CONTINUOUS command, above.



Note: This command and the previous one have similar names; remember that a component TRIGGERS the system, whilst taps and components are TRIGGERED.

6.8.5 STA

Syntax: Z-secs N-secs E-secs X-secs STA

Sets the length of the short-term averaging period in the STA/LTA triggering algorithm.

Z-secs, N-secs, E-secs and X-secs define the time periods over which to calculate the average for the Z, N, E, and auxiliary components, respectively. If a component is not considered by the triggering algorithm (see TRIGGERS, above), any value you specify here for that component will be ignored.

For example, 1 1 2 2 STA will calculate short-term averages over 1 second for the Z and N components, and over 2 seconds for the E and auxiliary components.



Note: If you issue STA with only one argument, it is used for all four components. For example, ***5 STA*** will calculate averages over 5 seconds for all components.

6.8.6 LTA

Syntax: *Z-secs N-secs E-secs X-secs* LTA

Sets the length of the long-term averaging period in the *STA/LTA* triggering algorithm.

Z-secs, *N-secs*, *E-secs* and *X-secs* define the time periods over which to calculate the average for the *Z*, *N*, *E*, and auxiliary components, respectively. If a component is not considered by the triggering algorithm (see TRIGGERS, above), any value you specify here for that component will be ignored.

For example, 15 20 20 20 STA will calculate long-term averages over 15 seconds for the *Z* component, and over 20 seconds for the *N*, *E* and auxiliary components.

If you issue STA with only one argument, it is used for all four components. For example, 5 STA will calculate averages over 5 seconds for all components.

6.8.7 RATIOS

Syntax: *Z-ratio N-ratio E-ratio X-ratio* RATIOS

Sets the ratio of STA to LTA above which a trigger will be declared in the *STA/LTA* triggering algorithm.

Z-ratio, *N-ratio*, *E-ratio* and *X-ratio* are the threshold ratios for the *Z*, *N*, *E*, and auxiliary components respectively. If a component is not considered by the triggering algorithm (see TRIGGERS, above), any value you specify here for that component will be ignored.

For example, 4 10 10 10 RATIOS will cause the DM24 to trigger if the STA/LTA ratio is above 4 for the *Z* component, *or* above 10 for any of the remaining components.

The arguments to this command must be integers. If you want to specify a fractional ratio, use the command FRATIOS instead.

6.8.8 FRATIOS

Syntax: *Z-ratio N-ratio E-ratio X-ratio* FRATIOS

Sets the ratio of STA to LTA above which a trigger will be declared in the *STA/LTA* triggering algorithm.

This command is exactly the same as RATIOS, except that its arguments are divided by 10 before being entered into the system. For example, the command

***25 100 100 100* FRATIOS**

will cause the DM24 to trigger if the STA/LTA ratio is above 2.5 for the *Z* component, *or* above 10 for any of the remaining components.

6.8.9 BANDPASS

Syntax: *tap filter* BANDPASS

Tells the DM24 which tap to use for *STA/LTA* triggering calculations.

Data are taken from this tap and passed through a band-pass filter before it is examined. This filtering maximises sensitivity within the frequency band of interest whilst filtering out noise outside this band.

The corner frequency of the band-pass filter depends on the values of *tap* and *filter*:

- *filter* = 1 creates a filter with a corner at 10 % of the Nyquist frequency for tap *tap* (*i.e.* 5 % of its sample rate);

- *filter* = 2 creates a filter with a corner at 20 % of the Nyquist frequency for tap *tap* (*i.e.* 15 % of its sample rate); and
- *filter* = 5 creates a filter with a corner at 50 % of the Nyquist frequency for tap *tap* (*i.e.* 25 % of its sample rate).

When LOWLATENCY is active, this command is used to set the tap used as the source of low-latency data. Data from this tap are decimated inside the DSP and output on a separate stream at a rate ten times slower than the configured rate.

6.8.10 MICROG

Syntax: *level* MICROG

Sets the output level above which a trigger will be declared when using the *LEVEL* triggering algorithm. *level* is measured in counts.

6.8.11 HIGHPASS

Syntax: *filter* HIGHPASS

Instructs the DM24 to pass the stream(s) which generate *LEVEL* triggers through a high-pass filter before examining them. *filter* can be:

- 1, for a 100 s high pass filter;
- 2, for a 300 s filter;
- 3, for a 1000 s filter; or
- 0, to disable the high pass filter.



Note: the main continuous outputs, both transmitted and recorded are also affected by any filter specified using this command.

6.8.12 PRE-TRIG

Syntax: *time* PRE-TRIG

Sets the pre-trigger recording time, where *time* is the number of seconds of buffered data to output from *before* a trigger is declared.

6.8.13 POST-TRIG

Syntax: *time* POST-TRIG

Sets the post-trigger recording time, where *time* is the number of seconds of data to output after a trigger condition lapses. If an event persists for some time, *all* triggering components must fall below the threshold before the trigger condition will lapse; only then will the post-trigger period begin.

6.8.14 TRIGGERIN

Syntax: `TRIGGERIN ENABLE | TRIGGERIN DISABLE`

Enables or disables external trigger input, in instruments equipped with this option.

Enabling external trigger input allows you to trigger the DM24 from an external logic level supplied through its digital output port. This voltage can be between 5 and 10 V supplied between the *Trigger In* pin and signal ground. If the DM24 is triggered externally, it will behave exactly as if it had generated the trigger itself, other than relaying the trigger (see `TRIGGEROUT`, below).

6.8.15 TRIGGEROUT

Syntax: `TRIGGEROUT ENABLE | TRIGGEROUT DISABLE`

Enables or disables external trigger output, in instruments equipped with this option.

Enabling external trigger output allows you to trigger other equipment through a relay contained within the DM24 whenever it triggers. The DM24's digital output port contains two pins (*Trigger out, common* and *Trigger out, normally-open*) which are connected when it triggers. In particular, you can connect a second digitiser with `TRIGGERIN ENABLE` in effect, in which case triggered data from both instruments will be transmitted whenever the DM24 triggers.

If a DM24 has both `TRIGGERIN ENABLE` and `TRIGGEROUT ENABLE` in effect, *only* triggers which the DM24 itself has generated will be output. Triggers received through the *Trigger in* port will cause the DM24 to output triggered streams, but will *not* be passed on to other digitisers, in order to prevent trigger loops.

6.8.16 POLARITY

Syntax: `0 POLARITY` or `1 POLARITY`

Sets the sense of the *Trigger In* pin on the *DATA OUT* port.

With `0 POLARITY` set, the *Trigger In* pin is active low, *i.e.* a trigger will be generated if this pin is connected to ground. `1 POLARITY` switches the pin to be active high.

You must issue `TRIGGERIN ENABLE` to enable external trigger input.

6.9 Calibration

6.9.1 SINEWAVE

Syntax: *component freq-or-period unit* SINEWAVE

Instructs the DM24 to inject a sine-wave calibration signal, starting on the zero crossing.

component specifies which component is to be calibrated, expressed as one of Z, N/S, E/W, or ALL. Some sensors use only the Z calibration loop for all three components.

freq-or-period and *unit* together determine the frequency of the calibration signal. If *unit* is HZ, then *freq-or-period* is taken as a frequency, in Hertz; if SECOND, then it is interpreted as a period, in seconds.

For example:

```
N/S 4 HZ SINEWAVE
```

freq-or-period must be an integer; if you want to specify a period of, for example, 0.5 SECOND, you must specify it as 2 HZ instead.

The calibration signal will be automatically disconnected after 2 minutes (unless you have altered the setting using the MINUTE command, described below).

Whilst calibration is in progress, the fourth (auxiliary, or *X*) data channel is switched to monitor the returning calibration signal.

6.9.2 SQUAREWAVE

Syntax: *component* SQUAREWAVE

Instructs the DM24 to inject a square-wave (step function) calibration signal, consisting of a positive step on the start of the next clock minute, followed by a negative step some minutes later (by default, 2). The calibration is disconnected the same number of minutes after the negative edge.

component specifies which component is to be calibrated: one of Z, N/S, E/W, or ALL. Some sensors use only the Z calibration loop for all three components.

You can alter the duration of each step using the MINUTE command, described below.

Whilst calibration is in progress, the fourth (auxiliary, or *X*) data channel is switched to monitor the returning calibration signal.

6.9.3 RANDOMCAL

Syntax: *component* RANDOMCAL

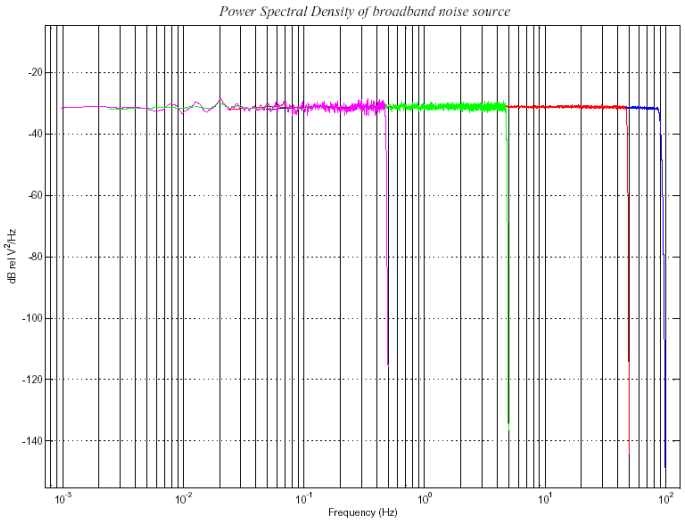
Instructs the DM24 to inject a white-noise calibration signal generated by an onboard pseudo-random number generator.

component specifies which component is to be calibrated: one of Z, N/S, E/W, or ALL. Some sensors use only the Z calibration loop for all three components.

The calibration signal will be automatically disconnected after 2 minutes if you have not altered the setting using the MINUTE command, described below.

Whilst calibration is in progress, the fourth (auxiliary, or *X*) data channel is switched to monitor the returning calibration signal.

The pseudo-random number generator produces true white noise over the entire passband, as shown in the power spectral density plot below:



The different coloured lines show the results of experiments which measure the power spectral density over several frequency bands. Together, the results show a flat noise level over the entire passband to within experimental error.

6.9.4 MINUTE

Syntax: *duration* MINUTE

Sets for how long the next SINEWAVE or RANDOMCAL calibration signal will be injected, or the period of the next SQUAREWAVE calibration signal.

duration is the desired interval, expressed in minutes. If you now issue a SINEWAVE or RANDOMCAL command, the calibration will last *duration* minutes; if

the next calibration command is SQUAREWAVE, a positive step of *duration* minutes will be generated, followed by a negative step of a further *duration* minutes.

If you do not issue MINUTE, calibration signals will default to 2 minutes.

Because of the way FORTH works, you can insert MINUTE commands into SQUAREWAVE, SINEWAVE or RANDOMCAL commands, for example:

```
N/S 4 HZ 5 MINUTE SINEWAVE
E/W 10 MINUTE SQUAREWAVE
```

6.9.5 %AMPLITUDE

Syntax: value %AMPLITUDE

Sets the relative amplitude of the signal, as a percentage. For SQUAREWAVE and RANDOMCAL calibrations, the default of 100% is normally suitable. If you have a high-gain sensor, you may have to use a lower value. For SINEWAVE calibrations, you may have to adjust the %AMPLITUDE setting to suit the frequency of the calibration signal.

You can insert %AMPLITUDE commands into calibration commands as described above, e.g.:

```
Z 3 MINUTE 50 %AMPLITUDE RANDOMCAL
```

6.9.6 AUTOCAL

Syntax: *days hh mm* AUTOCAL

Instructs the system to carry out an automatic calibration at time *hh* hours and *mm* minutes every *days* days. For example, the command

```
7 12 00 AUTCAL
```

will cause a calibration to be carried out once each week at noon.

The type of calibration and the associated parameters will be exactly as in the previous calibration, whether manual or automatic.

6.9.7 AUTOCAL?

Syntax: AUTOCAL?

Reports the current automatic calibration settings.

6.9.8 -AUTOCAL

Syntax: `-AUTOCAL`

Cancels any automatic calibration setting in force. If automatic calibration is not enabled, this command has no effect.

6.10 Actions

6.10.1 LOCK

Syntax: `LOCK` or `LOCK MONITOR`

Locks the sensor mass(es) of the instrument connected to *SENSOR A*.

If you issue LOCK, the command returns immediately, allowing you to close the terminal connection and follow the progress of the lock from the data streams.

If you issue `LOCK MONITOR`, the system displays the mass position for each component, measuring one position each second (*i.e.* one reading every 4 seconds for each component of a 4-component instrument). The process returns when locking is complete.

The mass positions are measured to 16-bit accuracy, with full deflection corresponding to values around ±32000 counts.

6.10.2 UNLOCK

Syntax: `UNLOCK`

Unlocks the sensor mass(es) of the instrument connected to *SENSOR A*.

`UNLOCK` returns immediately, whilst `UNLOCK MONITOR` monitors the progress of the unlock by displaying mass positions, as for `LOCK MONITOR` (above).

6.10.3 CENTRE

Syntax: `CENTRE` or `CENTER MONITOR`

Centres the sensor mass(es) of the instrument connected to *SENSOR A*.

`CENTRE` returns immediately, whilst `CENTRE MONITOR` monitors the progress of the unlock by displaying mass positions, as for `LOCK MONITOR` (above).

When the masses are correctly centred, the mass positions should read less than ±1000 counts.

6.10.4 MASSES?

Syntax: `MASSES?`

Begins reporting the current mass positions for all three components, once per second, until you press a key. Mass position information is also output on Mux channels M8, M9 and MA, if you have enabled them.

Over time, and especially if the temperature changes significantly, the sensor masses may drift off centre. If this is not corrected, velocity output will eventually be lost.

6.10.5 %AUTO-CENTRE

Syntax: *`distance`* `%AUTO-CENTRE`

Instructs the DM24 to perform a round of centring whenever a mass position drifts further than *`distance`* from zero. *`distance`* is measured as a percentage of full scale.

Setting *`distance`* to zero disables automatic centring.

6.10.6 O/SNULL

Syntax: *`component value`* `O/SNULL`



Note: This command is in the extended dictionary: to use it, first type `ok-1` and finish with `[seal]`.

Remove any DC offsets in a stream by applying a counter-offset in software. As manufactured, for technical reasons, the ADC offset is set to around 75000 counts. You should only need to change this if the digitiser's CMOS RAM becomes corrupted; it is always better to zero the instrument by levelling and centring as far as is possible, or by adjusting potentiometers in sensors (where these are present).

`component` specifies which component to adjust: 0, 1, 2, and 3 denote the Z, N, E and X components of *SENSOR A*, and 4, 5, 6 and 7 are the corresponding components of *SENSOR B*.

`value` specifies the adjustment to apply to the component.

6.10.7 RE-SYNC

This command is only available in units with a precision real-time clock module fitted.

Syntax: `RE-SYNC` *`ON|OFF`*

`RE-SYNC ON` instructs the DM24 to resynchronize the real-time clock module to GPS time when a GPS fix is next obtained.

`RE-SYNC OFF` cancels a previous `RE-SYNC ON` instruction, so that attaching a GPS module will no longer cause the DM24 to resynchronize.

6.10.8 ?LITHIUM

Note: This command is only available in units with a precision real-time clock module fitted.

Note: This command is in the extended dictionary: to use it, first type `ok-1` and finish with `[seal]`.

Syntax: `?LITHIUM`

Query the current status of the RTC's internal battery.

6.10.9 RE-BOOT

Syntax: `RE-BOOT` (interactive)

Causes the DM24 to reboot after a delay of 2 s.

```
RE-BOOT
Confirm with 'y' ? y
```

Responding to the confirmation message with anything other than `y` will abort the reboot.

6.11 Data storage and transmission

6.11.1 SHOW-FLASH

Syntax: `SHOW-FLASH`

Reports status information about the Flash memory in the DM24, if fitted. For example, the output from `SHOW-FLASH` for a new system with 8 × 64Mb chips fitted would look like this:

```
512MB Flash File buffer : 16 Blocks Written  0 Unread
524,288 Free
Oldest data  [16] Blank
 Read point  [16] Blank
Latest data  [16] Blank
File Replay  [16] Blank
```

For a system in use with 1 × 64Mb chip fitted, it might look like this:

```
64MB Flash File buffer : 65,536 Blocks Written  65,532
Unread  4 Free
Oldest data  [36,272]  GURALP TESTZ1 2003 12 3 20:20:41
 Read point  [36,272]  GURALP TESTZ1 2003 12 3 20:20:41
Latest data  [36,268]  GURALP TESTX2 2003 12 4 10:09:12
File Replay  [36,272]  GURALP TESTZ1 2003 12 3 20:20:41
```

The first line displays general status information, namely the quantity of Flash memory available (64MB Flash File buffer), how many 1k GCF blocks have been written (65,536 Blocks Written), how many remain to be transmitted (65,532 Unread), and how much space (in 1k blocks) remains (4 Free).

The next lines show, in turn:

- the stream and timestamp of the oldest data in memory;
- the current position of the read pointer;
- the stream and timestamp of the latest data in memory; and
- the current position of the replay pointer.

The read pointer normally points to the oldest data in the file; however, if all the data up to the replay pointer have already been downloaded, the read pointer is set to the replay pointer and the number of blocks remaining `Unread` is updated.



Note: Downloads where a time-period is not specified *always* start from the read pointer.

6.11.2 RESET-FLASH

Syntax: `RESET-FLASH`

Resets the Flash memory pointers. The DM24 will start overwriting old data from the beginning of memory. You can still access these data, however, up until the point they gets overwritten (unlike `ERASEFILE`).

6.11.3 ERASEFILE

Syntax: `ERASEFILE` (interactive)

Removes all data from the Flash memory and resets all pointers to the beginning. You will be asked for confirmation before you do this, as all data will be destroyed.

If you have been running the DM24 with a FireWire disk attached, remember that there will be some data still in Flash memory awaiting transfer (up to the amount you last set with `MBTRANSFER)`. You should make sure these data are transferred with the `DISKMENU` command before you issue the `ERASEFILE` command.

Depending on the amount of Flash memory installed, `ERASEFILE` may take several minutes to complete.

6.11.4 EVENTSLIST

Syntax: *event-specification* EVENTSLIST

Displays details of triggered events. The parameter *event-specification* can have several forms:

- RECENT, to display the most recent events. By default, 16 events are displayed. You can double or halve this number by prefixing RECENT with either SHORTER or LONGER. You can use several of these commands together. For example, the command

  LONGER LONGER RECENT EVENTSLIST

  will display 64 events. The DM24 remembers this setting, so a further LONGER RECENT EVENTSLIST would display 128 events.

- EARLIER, to display the previous set of events of the currently-configured length, *e.g.* if RECENT EVENTSLIST displays the 16 events #30 to #45, EARLIER EVENTSLIST will display the 16 events #14 to #29.

- COMPLETE, to display all events currently held in Flash memory.

- *yyyy mm dd hh mm* SEARCH, to look for events in the 10-minute period defined by *yyyy mm dd hh mm*, For example,

  ***2005 12 26 04 13* SEARCH EVENTSLIST**

  will display events between 04:10 and 04:20 on the 26th of December, 2005. The time format is the same as for FROM-TIME and TO-TIME, as documented in section 6.12.8 on page 89.

  Note: If an event has occurred, but nothing was filed (for example, because the digitiser is in *DIRECT* mode), it will be listed with the message No File.

- two numbers *event-id number-of-events*, to display *number-of-events* starting at event number *event-id*.

6.11.5 RESET-LIST

Syntax: RESET-LIST

Empties the events list. You will no longer be able to see event information with the EVENTSLIST command, or download events with EVENTS (see below).

This command does not delete any data. If you empty the events list, and the digitiser is in a suitable transmission mode, you can still download event data using the commands in Section 6.12 on page 87, 'Downloading over the serial port'.

6.11.6 DIRECT

Syntax: `DIRECT`

Instructs the DM24 not to use Flash memory for storage. Instead, all data are transmitted directly to clients. A temporary RAM buffer allows clients to request blocks they fail to receive using the Block Recovery Protocol.

6.11.7 FILING

Syntax: `FILING`

Instructs the DM24 not to transmit blocks to clients, but to store all digitized data in the Flash memory. The memory is used in circular fashion, *i.e.* if it becomes full, incoming blocks begin overwriting the oldest in memory.

6.11.8 DUPLICATE

Syntax: `DUPLICATE`

Instructs the DM24 to transmit all new data directly to clients (as in *`DIRECT`* mode) as well as storing them in Flash memory (as in *`FILING`* mode). If a client fails to acknowledge a block, the digitiser does not attempt to retransmit it.

6.11.9 DUAL

Syntax: `DUAL`

Instructs the DM24 to transmit any *continuous* streams directly to clients (as for *`DIRECT`* mode) but to store *triggered* data into Flash storage (as for *`FILING`* mode).

If you choose this mode, the instrument will send heartbeat messages in addition to any continuous streams you have configured. Scream! can pick these up and download new data as necessary.

6.11.10 ADAPTIVE

Syntax: `ADAPTIVE`

Instructs the DM24 to transmit current blocks to clients if possible, but to store all unacknowledged blocks in the Flash memory and re-send them when time allows. This differs from *DIRECT* in the following ways:

- the client does not have to keep track of which blocks it has not received, and
- the entire Flash memory can be used for blocks awaiting retransmission, rather than a temporary RAM buffer.

ADAPTIVE mode is best suited for installations where the link between digitiser and client is intermittent.

6.11.11 FIFO

Syntax: `FIFO`

Instructs the DM24 to begin writing new blocks to Flash memory as for *`FILING`*, but also to transmit data to clients. Data are transmitted in strict order, oldest first; the DM24 will only transmit the next block when it receives an acknowledgement of the previous block. Once a block is transmitted, it is marked as such in the Flash memory, and may be overwritten later.

6.11.12 HEARTBEAT

Syntax: *`interval`* `HEARTBEAT`

When the DM24 is in *FILING* mode, it outputs status messages periodically over the RS232 port (see Section 7.1 on page 95). This command changes how often these messages are produced.

If the digitiser has been configured to dial out over a modem link (see `+MODEM`, above), it will dial out whenever it generates a heartbeat message.

`interval` is normally measured in whole units of 30 milliseconds. For example, issuing

```
600 HEARTBEAT
```

will cause the DM24 to output heartbeat messages every (600 × 30ms) = 18,000 milliseconds = 18 seconds.

Alternatively, you can insert the word `HOUR` and specify `interval` in hours:

```
5 HOUR HEARTBEAT
```

The maximum value allowed for `HEARTBEAT` is 300,000,000 milliseconds, which is 2,500 hours (approximately 100 days).

If you connect to a digitiser in *`FILING`* mode using Scream!, Scream! will not detect the digitiser until it has sent a heartbeat message. Therefore, you should use a relatively short heartbeat interval if you have a continuously-connected instrument.

6.11.13 MS-GAP

Syntax: *`interval`* `MS-GAP`

The GCF protocol allows for transmitted data blocks to be acknowledged by the receiver in order to improve the resilience of links. Unacknowledged blocks can be re-transmitted immediately or stored for later transmission. MS-GAP sets the

interval the digitiser should wait for an acknowledgement message, before assuming that the block could not be transmitted. If a period of *interval* passes without an acknowledgement, the digitiser's behaviour then depends on the current transmission mode (see above).

interval is measured in milliseconds. The default is 150. If the value is greater than the average time between blocks being generated, and an outage occurs in the return communications link, the digitiser will be producing data faster than it can transmit them, and gaps will start to be observed. However, systems using slower communications links (e.g. radio links) may be unable to acknowledge blocks in under 150 ms. You should choose a value for *interval* which is suitable for your particular installation.

6.11.14 RE-USE

Syntax: RE-USE

Instructs the DM24 to use its Flash memory as a circular buffer, overwriting the oldest data when it becomes full.

In this mode, the Flash memory will always contain the latest data.

6.11.15 WRITE-ONCE

Syntax: WRITE-ONCE

Instructs the DM24 not to overwrite data in the Flash memory when it becomes full Instead, when the digitiser runs out of space, it switches to *DIRECT* transmission mode and will leave the Flash memory untouched.

In this mode, the Flash memory will always contain the earliest data.

6.11.16 MODE?

Syntax: MODE?

Queries and displays the current Flash memory usage mode: RE-USE or WRITE-ONCE, as described above.

6.12 Downloading over the serial port

6.12.1 DOWNLOAD

Syntax: DOWNLOAD

Initiate a data transfer from the Flash memory over the serial connection. Scream! Can be configured to auto-record these data. Refer to the Scream manual for a

detailed description of data recording; a summary is given in section 4.5.3 on page 29.

The flash memory is used as a ring buffer. Two pointers into the memory keep track of where data were last read (the "Read Pointer") and last written (the "Write pointer"). When either pointer reaches the end of physical memory, it wraps round back to the beginning. The behaviour when the write pointer reaches the read pointer (i.e. when the memory becomes full of data, none of which have been downloaded) is governed by the commands RE-USE/RECYCLE and WRITE-ONCE, described above.

Which data are downloaded depends on the current *read pointer*, as well as various user-configurable parameters, which allow you to select a particular stream, streams of a specified sample rate or streams within a certain time window. You can set parameters separately or string the definitions before the DOWNLOAD command. For example,

```
ALL-FLASH STREAM HPA0N1 DOWNLOAD
2004 12 01 00 00 FROM-TIME ALL-DATA DOWNLOAD
100 S/S ALL-TIMES DOWNLOAD
```

See below for full details of the parameter-setting commands.

After each download completes, the read point is normally set to indicate the start of the first unread block – *i.e.* where downloading would normally resume. The exception is a download issued after an ALL-DATA command, which does not affect the read point. The read point can be reset with the ALL-FLASH command, described below.

The DOWNLOAD command, when invoked with no parameters, starts another download with the same parameters as last time.

The DOWNLOAD command returns immediately, so that you can issue more commands if required. To close the connection and begin downloading, issue the GO command. You can pause the download by entering terminal mode, and then either restart it (with another GO) or abort it with the command END-DOWNLOAD.

6.12.2 ALL-FLASH

Syntax: ALL-FLASH

Sets the read point to the oldest data held by the DM24. This command does not alter which streams are to be transmitted; you should specify streams or use the ALL-DATA command in addition to this one.

6.12.3 ALL-DATA

Syntax: ALL-DATA

Instructs the DM24 to transmit all the data streams it holds next time a `DOWNLOAD` is issued. This command does not alter the read point or time period; you should either specify a time period or use the `ALL-FLASH` command in addition to this one.

6.12.4 STREAM

Syntax: `STREAM` *`stream-id`* (*n.b. - trailing parameter*)

Instructs the DM24 to transmit only the stream with ID *`stream-id`*. Stream IDs are normally a 4-character Serial n° followed by a component letter and a tap number.



Note: Unlike most FORTH commands, the stream-id parameter goes after the command.

6.12.5 STATUS-ONLY

Syntax: `STATUS-ONLY`

Instructs the DM24 to transmit only status streams (text streams, normally with stream IDs ending in `00`).

6.12.6 S/S

Syntax: *`rate`* `S/S`

Instructs the DM24 to transmit only streams with sample rates equal to *`rate`*. If *`rate`* is zero, only status streams are transmitted.



Note: This command should not be confused with the SAMPLES/SEC command.

6.12.7 ALL-TIMES

Syntax: `ALL-TIMES`

Clears any time selection in force, allowing you to download all data held by the DM24. This command does not alter which streams are to be transmitted; you should specify streams or use the `ALL-DATA` command in addition to this one.

6.12.8 FROM-TIME

Syntax: *`yyyy mm dd hh mm`* `FROM-TIME`

Instructs the DM24 to transmit only data more recent than the time specified by *`yyyy-mm-dd hh:mm`*, where:

- *`yyyy`* is a four-digit year (1989 – 2069);

- *mm* is the month number (1 – 12);
- *dd* is the day of the month (1 – 31);
- *hh* is the hour of the day (0 – 23); and
- *mm* is the minute of the hour (0 – 59).

6.12.9 TO-TIME

Syntax: *yyyy mm dd hh mm* TO-TIME

Instructs the DM24 to transmit only data older than the time specified by *yyyy-mm-dd hh:mm*, where *yyyy*, *mm*, *dd*, *hh* and *mm* have the same meanings as in FROM-TIME, above.

6.12.10 EVENTS

Syntax: *event-specification* EVENTS

Instructs the DM24 to download only data recorded during triggered events. *event-specification* can have several forms:

- RECENT, to download the most recent events. By default, 16 events are displayed. You can double or halve this number by prefixing RECENT with either SHORTER or LONGER. You can use several of these commands, *e.g.*

  LONGER LONGER RECENT EVENTS

  will download 64 events. The DM24 remembers this setting, so a further LONGER RECENT EVENTS would download 128 events.

- EARLIER, to download the previous set of events, using the currently-configured length, *e.g.* if RECENT EVENTS downloads the 16 events #30 to #45, EARLIER EVENTS will download the 16 events #14 to #29.

- COMPLETE, to download all events currently held in Flash memory.

- *yyyy mm dd hh mm* SEARCH, to look for events in the 10-minute period containing *yyyy mm dd hh mm*, *e.g.*

  ***2005 12 26 04 13 SEARCH* EVENTS**

  will download events between 04:10 and 04:20 on the 26th of December, 2005. The time format is the same as for FROM-TIME and TO-TIME, as documented in section 6.12.8 on page 89.

  If an event occurs, but nothing is filed (*e.g.* because the digitiser is in *DIRECT* mode), it will be listed with the message No File.

- two numbers *event-id number-of-events*, to download *number-of-events* starting at event number *event-id*. You can find out *event-id* using the command EVENTSLIST (described in section 6.11.4 on page 84).

Thus, if EVENTSLIST returns

```
#    7  Filed @ 63,772   PLPGG SBHYZG 2006  1  4 16:07:50
#    8  Filed @ 63,812   PLPGG SBHYZG 2006  1  4 16:22:24
#    9  Filed @ 63,952   PLPGG SBHYZG 2006  1  5 10:14:59
#   10  Filed @ 64,005   PLPGG SBHYZG 2006  1  5 10:16:49
#   11  Filed @ 64,039   PLPGG SBHYEG 2006  1  5 10:17:45
```

you can instruct the digitiser to download these 5 events with the command:

```
7 5 EVENTS
```

Once you have specified the events to be downloaded, issuing DOWNLOAD will make the DM24 begin downloading data as soon as your terminal session is ended.

6.12.11 GO

Syntax: GO

Closes terminal mode, and begins the download you have set up. You can pause the download by re-entering terminal mode, and restart with another GO or abort with END-DOWNLOAD.

Once the download has completed, the DM24 will carry on transmitting real-time data, if you have so configured it.

6.12.12 END-DOWNLOAD

Syntax: END-DOWNLOAD

Aborts any downloads which were in progress at the time you entered terminal mode.

6.13 Downloading over FireWire

6.13.1 DISKMENU

Syntax: DISKMENU (interactive)

This command displays a menu which allows you to select various options for transferring data over the optional FireWire interface to a connected device. When you issue this command you will see the message

```
Plug in FireWire cable
```

and the DM24 will wait for you to plug in a compatible disk. When it detects one, it will print out information about the disk:

```
FireWire Connection *BR
Node: FFC1 #Nodes 2
```

```
Reading Node FFC0 .
LaCie Group SA
1394-IDE Rev B2
LaCie DATA BANK drive LUN 0
Logging on @AgentCSR= 0010:00000
Capacity 39.0GB

Press a key for options else
7 seconds to automatic disk backup
```

At this point, pressing a key will cause the DM24 to display an options menu and then pause. If you do not press a key within seven seconds, the disk will be automatically backed up.

The menu looks like this:

```
Disk Options :-
 (A) save All data
 (N) save New data
 (S) save Selected time window
 (D) Directory - contents
 (R) Reset disk - overwrite
 (X) eXit
A/N/S/D/R/X ?
```

During the operation of any of these options, progress reports will occasionally be displayed on the terminal.

6.13.2 MBTRANSFER

Syntax: *size* MBTRANSFER

If a FireWire disk is plugged in to the DM24 when you are *not* running DISKMENU, and there are enough new data in Flash memory, the DM24 will automatically transfer the new data to the disk.

The MBTRANSFER command sets how much new data there needs to be before the DM24 will power up the disk and transfer it. If the disk is left attached, the DM24 will then wait until it has collected the same amount of data again before starting another transfer.

size is the amount of data to transfer, in megabytes; it must be greater than 10 and less than the total quantity of Flash memory in the device. Larger values will save power, because the disk needs to be accessed less often.

If you plug a FireWire disk into the DM24 when there is less than *size* Mb of data ready, no data will be transferred. If you need to transfer smaller amounts of data, you should use the DISKMENU system.



Note: The number of files stored on a disk is limited to 510. This may prevent you from using the full capacity of the drive if the value of `MBTRANSFER` is chosen to be particularly low. For example, a setting of 10Mb for `MBTRANSFER` will limit the total quantity of data stored to 5.1Gbytes, regardless of the drive capacity.

6.13.3 DIR

Syntax: `DIR` (interactive)

Lists the files already present on a FireWire disk. When you issue this command you will see the message

```
Plug in FireWire cable
```

and the DM24 will wait for you to plug in a compatible disk. When it detects one, it will display information about the disk and list all the files that it contains. Each time you use `DISKMENU` to download data, or power on the digitiser with a FireWire disk connected (and enough new data to trigger a download), a new file is created.

7 Inside the DM24

7.1 State of health information

The DM24 outputs diagnostic information in a *status stream*. This is a special GCF stream whose name ends in `00`.

Status streams can take two different forms, depending on the current transmission mode:

- In *`FILING`* mode, a short *heartbeat message* summarizing the current status is transmitted regularly at a configurable interval of between 10 seconds and a few minutes. The heartbeat message lets clients know that the digitiser is still operational, even though it is not sending data. For example, if you connect a PC running Scream! to an autonomous station over a dial-up link, Scream! can automatically download recorded data: this is made possible by the heartbeat message, which tells Scream! that the data exist.
- In all other modes, status messages are output continuously, in GCF blocks. When each block is full, it is transmitted from the *DATA OUT* port with the rest of the data.

7.1.1 Heartbeat messages

These are output at regular intervals, and always have the same format. Older digitisers may not include all the lines described here.

```
  64MB Flash File buffer : 65,520 Blocks Written  65,520 Unread
16 Free
  Latest data  [392]      PLPGG SBHYX2 2006  1 18 14:55:57
  Oldest data  [400]      PLPGG SBHYN4 2005 11 30 06:47:38
  #  22 2006  1 18 14:48:36  No File Last Event
  2006  1 18 14:56:15 1 MicroSeconds Fast  Freq error 0 e-9  Auto
3D [-4]
  PLPGG SBHY00 CMG-3T
  Boot Log :   143 Power cycles    294 Watchdog resets
  Last boot  2006  1 12 16:18:57    2006  1 17 17:15:14
  2006  1 18 14:56:15 External supply : 13.0V Temperature  24.68'C
  2006  1 18 14:56:15 Mass positions   -486    -300    -424
```

7.1.1.1 Flash memory usage

The first three lines describe the internal Flash memory. First, the overall usage is described:

```
64MB Flash File buffer : 65,520 Blocks Written  65,520 Unread   16
Free
```

Here, a 64 Mb Flash memory store contains 65,520 data blocks, of which none have already been downloaded. This leaves 16 blocks of free space, since each block is exactly 1Kb long, and (65,520 + 16) × 1 Kb = 65,536 Kb = 64 Mb.

```
Latest data  [392]      PLPGG SBHYX2 2006  1 18 14:55:57
Oldest data  [400]      PLPGG SBHYN4 2005 11 30 06:47:38
```

These two lines show the date and time of the oldest and newest data in the buffer, together with the *System ID* (`PLPGG`) and the *Stream ID*s (`SBHYX2` and `SBHYN4`) of the corresponding data stream.

7.1.1.2 Triggered events

```
#  15  Filed @ 64,167   PLPGG SBHYZG 2006  1  5 10:31:30  Last
Event
```

This line gives the date and time of the last trigger, together with its *System ID* and *Stream ID* as above. Also given is the event sequence number (15, here) as used by the `EVENTSLIST` command, and the position in the buffer where data from the event begin.

If a trigger has occurred, but no data were output (*e.g.* because the digitiser was in *DIRECT* transmission mode), the digitiser will report

```
#  22 2006  1 18 14:48:36  No File Last Event
```

7.1.1.3 GPS

The next line describes the current status of the GPS system. When the digitiser is first powered up, the message will be something like

```
2005 8 10 12:35:00 GPS control OFF  Auto 3-D [-5 Not sync'd]
```

The message GPS control OFF denotes that the clock is not yet being controlled by the GPS. Owing to noise purposely injected into the GPS timing stream, the GPS time cannot be directly used to synchronize the clock. Instead, the digitiser maintains a rolling average of time measurements, and uses this to apply corrections to the clock so that it remains accurate over a long period.

The `Auto 3-D` message denotes that the signal is strong enough to obtain a full 3-dimensional GPS fix. If the signal is weak, or there are too few satellites visible to the receiver, you may see `Auto 2-D` here. If the signal is too weak to obtain a fix, `No FIX` will be displayed.

After a short time, the message will change to

```
2005 8 10 12:36:00 GPS Control settling  Auto 3-D [-5 Not sync'd]
```

Eventually, the GPS control system will be able to synchronize the internal clock, and the message will change to

```
2005 8 10 12:38:01 GPS Control settling  Auto 3-D [-4]
2005 8 10 12:41:01 326 MicroSeconds Slow  Freq error -7 e-9  Auto
3-D [-4]
```

The system is now reporting the current offset of the internal clock from GPS time (whether slow or fast), and the current frequency error. It will now continuously adjust the internal clock for as long as the GPS is powered up.

If you have chosen to save energy by enabling *GPS power cycling*, the DM24 will switch off the GPS once a satisfactory fix is obtained, and begin free-running on the internal clock. Whilst this is happening, the line will report

```
2005 8 10 17:06:27 GPS switched Off
```

7.1.1.4 Boot status

```
PLPGG SBHY00 CMG-3T
Boot Log :   143 Power cycles    294 Watchdog resets
Last boot  2006  1 12 16:18:57    2006  1 17 17:15:14
```

These three lines repeat information from the last re-boot:

- in the first, the *System ID*, the *Stream ID* of the status stream, and the current *Sensor Type*;
- in the second, the number of times the digitiser has been power cycled, and the number of times it has been reset without power cycling (*e.g.* by a `re-boot` after reconfiguring the module: a "soft" reset); and
- in the third, the date and time of the last power cycle and of the last soft reset.

7.1.1.5 Environmental measurements

```
2006  1 18 14:56:15 External supply : 13.0V Temperature  24.68'C
```

The next line displays the current state of the DM24's power supply, and its internal temperature.

7.1.1.6 Mass positions

```
2006  1 18 14:56:15 Mass positions   -486    -300    -424
```

The final line provides the instantaneous mass positions reported by the sensor at the time given.

7.1.2 Status blocks in continuous mode

In continuous mode, each status block contains a number of single-line messages:

```
2006  1 18 14:40:00 o/s=     90 drift=     0 pwm= 8187  Auto 3D
2006  1 18 14:41:00 o/s=     90 drift=     0 pwm= 8187  Auto 3D
2006  1 18 14:42:00 o/s=     90 drift=     0 pwm= 8187  Auto 3D
2006  1 18 14:43:00 o/s=     78 drift=   -12 pwm= 8188  Auto 2D
2006  1 18 14:44:00 o/s=     89 drift=    11 pwm= 8188  Auto 3D
2006  1 18 14:45:00 o/s=     94 drift=     5 pwm= 8188  Auto 3D
2006  1 18 14:45:00 External supply : 13.0V Temperature  24.62'C
2006  1 18 14:46:00 o/s=    148 drift=    54 pwm= 8188  Auto 3D
2006  1 18 14:47:00 o/s=    174 drift=    26 pwm= 8188  Auto 3D
```

```
2006  1 18 14:48:00 o/s=    211 drift=    37 pwm= 8188  Auto 3D
2006  1 18 14:48:36 SOFTWARE Trigger  : Trigger# 22
2006  1 18 14:49:00 o/s=    263 drift=    52 pwm= 8187  Auto 3D
2006  1 18 14:49:10 End of Trigger
```

The messages appear in the order they are generated by the different software modules inside the digitiser.

7.1.2.1 GPS status messages

The status of the attached GPS receiver is reported every minute:

```
2006  1 18 14:47:00 o/s=    174 drift=    26 pwm= 8188  Auto 3-D
```

In this line:

- `o/s` gives the current offset of the internal clock from GPS time, in arbitrary internal units;
- `drift` is the difference between the current `o/s` and the last, which gives an indication of the level of drift;
- `pwm`, or *Pulse Width Modulation*, is the feedback control parameter;
- `Auto 3-D` denotes that enough GPS satellites are currently visible to obtain a full 3D fix. If, instead, `No Fix` is displayed, the GPS received has not (yet) been able to obtain a satisfactory fix. If `Missing PPS` is displayed in this field, the digitiser has received fewer than 60 pulses on the 1pps (pulse per second) line in the last minute. You will see at least one `Missing PPS` message in the first minute of operation.

Every 20 minutes, at 10, 30 and 50 minutes past the hour, the GPS synchronization status is displayed. This line has the same format as the GPS line in the heartbeat message, *e.g.*

```
2005 8 10 12:41:01 326 MicroSeconds Slow  Freq error -7 e-9  Auto
3-D [-4]
```

Finally, every hour, the DM24 displays general timing and position information, including details of the visibility of GPS satellites:

```
2005 6 8 11:00:00 GPS Date/Time 08/06/05 11:00:00
2005 6 8 11:00:00 Auto 3-D  SV#'s 4 7 13 20 23 24 25   ( 7 )
2005 6 8 11:00:00 Lat 51'21.6591N  Long 001'09.8218W
```

SV# stands, officially, for **S**pace **V**ehicle number. Each GPS satellite has it's own unique identification number, or SV#, and the list shows which are currently usable (satellite numbers 4, 7, 13, 20, 23, 24 and 25 in this example). A minimum of four is required to obtain a full 3-dimensional GPS fix.

7.1.2.2 Trigger notifications

When a trigger occurs, a message is immediately saved to the current status block. You may not see this message straight away, however, since the status block must be full before the digitiser will send it.

```
2006  1 18 14:48:36 SOFTWARE Trigger  : Trigger# 22
```

The trigger type is mentioned, as well as the sequence number (for later retrieval, if necessary, using the `EVENTSLIST` command). When the trigger condition ends, the system reports

```
2006  1 18 14:49:10 End of Trigger
```

7.1.2.3 Environmental measurements

These are measured every 10 minutes (at 5, 15, 25, 35, 45 and 55 minutes past the hour), and are given in the same form as for heartbeat messages:

```
2006  1 18 14:45:00 External supply : 13.0V Temperature  24.62'C
```

7.2 GPS and Synchronisation

7.2.1 Overview

The DM24 has two internal clocks: a hardware real-time clock (RTC), which runs continuously (even when the unit is powered down) and a software clock that is used for time-stamping digitised data. The software clock uses a voltage-controlled crystal oscillator (VCXO) to provide a time reference which is stable to below 0.1 ppm.

The software clock is set from the RTC at boot-up: this is the only time that the system consults the RTC.

The software clock is normally synchronised to GPS time, as reported by an attached Güralp GPS receiver. After power-up, the output from the GPS receiver is monitored until a stable fix is obtained, at which point both clocks are synchronised to GPS time. A sophisticated algorithm then monitors the frequency and phase difference between the GPS's PPS signal and the VCXO's output and adjusts the voltage input to the VCXO so that the software clock precisely tracks the average GPS time. (Recall that GPS time includes deliberate fluctuations to reduce the instantaneous accuracy for non-military users.)

If the GPS signal becomes too weak to use or if the receiver is deliberately powered down to reduce the power consumption, the software clock will continue to run with the same VCXO setting until a stable GPS signal is received again. At this point, the software clock will normally be slewed to bring it back into precise alignment with GPS time. If the difference between GPS time and the software clock after a period of free-running is too large, a step-change is performed and both clocks are re-set to GPS time. This is reported in the status stream as a resynchronisation.

7.2.2 Leap-seconds

GPS time does not include leap-seconds so the firmware in the GPS receiver performs corrections to account for the difference between GPS time and UTC. This difference is the sum of all of the leap-seconds which have occurred since the GPS epoch (January 6th, 1980) and, at the time of writing, UTC leads GPS time by 17 seconds.

The current value of the offset and the date of the next leap-second (if known) are regularly transmitted by the GPS space vehicles to the receivers. When a leap-second occurs, the receiver announces the time 23:59:60 as the extra second between 23:59:59 and 00:00:00.

If the DM24 sees this time, it recognises the leap-second and all data will be time-stamped appropriately. If for any reason, the digitiser does not receive this time - the system may, for example, be configured for GPS power-cycling and the receiver may be powered down - the digitiser will produce data with incorrect time-stamps until it can resynchronise to the GPS receiver again. If the digitiser is notified in advance about an upcoming leap-second, this situation can be avoided and the leap-second will be handled correctly even if GPS data are unavailable at the time. Güralp's newest GPS receivers - those based on the PAM7Q chip-set - will automatically notify the DM24 of upcoming leap-seconds but older receivers cannot do this.

You can manually notify a DM24 of an upcoming leap-second from the command line using the `LEAPSECOND` command - see section 6.4.7 on page 62 for full details. This can be done at any time between the official announcement and the actual leap-second.

Upcoming leap-seconds are announced by the International Earth Rotation Service (https://www.iers.org) in their Bulletin C, to which anyone can subscribe, and also on the Güralp web-site in the Frequently-Asked Questions (Support→FAQs) section.

7.2.3 Week-number roll-over

Although the GPS system can be used to determine the date and time with extreme accuracy, the GPS satellite constellation does not actually transmit the full date to GPS receivers. Instead, a ten-bit value called "Week Number" is transmitted every thirty seconds, as part of each subframe of the "Navigation Message". It is the responsibility of the receiver to calculate the date from this value. (The time within the week is transmitted as the number of seconds since midnight on Saturday/Sunday.)

GPS week zero started at the beginning of 00:00:00 UTC on January the 6th, 1980. A ten-bit field can only hold 1024 different values so this system was never going to last forever. Indeed, week 1023 was first reached on August the 15th, 1999. The following week, the GPS satellites populated the Week Number field with a value of

zero. (Because GPS time does not recognise leap-seconds, the "roll-over" from week 1023 to week zero actually took place at the end of 23:59:47 UTC on August the 21st.)

Manufacturers of GPS receivers must each choose a way to determine the correct date from the GPS Week Number. If the chosen method fails, the announced date will be 1024 weeks - about 19.7 years - in the past or, possibly, the future. One common method uses the date of the version of the firmware as a hint, which works well if the receiver is new or regularly updated. A significant problem with this method arises when the firmware is not updated: the receiver can start producing incorrect dates at the 1024-week anniversary of the firmware date. This means that problems can actually appear at any time, irrespective of the actual roll-over date.

Many significant problems were reported after the 1999 roll-over and this became known as the GPS "week-number roll-over" or "WNRO" problem. Many manufacturers had to update their receiver firmware as a result but some of these solutions were time-limited, leading to continuing problems around the April, 2019 roll-over.

The DM24 can automatically correct for receivers which report historical times as a result of this problem. If the `setWNROpivot` command (see section 6.4.6 on page 62) is used to set a "pivot date", any dates reported by a GPS receiver which are *earlier* than the pivot date will be modified by repeatedly adding 1024 weeks to the reported date until a date *later* than the pivot date is reached. This will then be the date used for time-stamping data. Because you can set the pivot date yourself, this technique should be able to correct for any problems encountered both before and after the next roll-over, in 2038.

The following extract of a status stream shows the technique in action:

```
2019  7 27 23:59:00 o/s= -10263 drift= -1636 pwm= 8315  Auto 3D
2019  7 28 00:00:00 GPS Date/Time 12/12/99 00:00:00
2019  7 28 00:00:00 o/s=  -9497 drift=   766 pwm= 8342  Auto 3D
2019  7 28 00:00:00 Auto 3D  SV#'s 28 2 18 21 23 25 26   ( 7 )
2019  7 28 00:00:00 Lat 51'21.8502N  Long 001'09.9228W Height   6m
```

Note how the third line starts by reporting that the system data is July 28th, 2019 but then reports that the GPS Date is December 12th, 1999. These two dates are exactly 1024 weeks apart because of the WRNO correction.

7.2.4 Multiple re-synchronisation events

In an ideal world, the digitiser would synchronise to GPS shortly after power-up and then sit happily, training its clock to GPS time with a few tiny adjustments to the VCXO control voltage to compensate for, say, thermal variations.

It would be very unusual for the GPS receiver to suddenly report a radically different time. If it does happen, the digitiser becomes sceptical and waits a whole minute, monitoring the situation without changing the internal clock. It increments an internal counter - the resync counter - to record the event and prints a message like

```
Clock check 1 0 Not sync'd
```

followed by the difference between the software clock's time and GPS time. A minute later, if the new GPS time has been stable and consistent over that minute, the digitiser will step-change its clock - a resynchronisation.

If the same thing happens again, the digitiser is even more sceptical and waits two minutes before reacting, again incrementing the resync counter. The messages are

```
Clock check 1 0 Not sync'd  -1secs error
…
Clock check 2 0 Not sync'd  -1secs error
```

appearing one minute apart. Again, the digitiser will resynchronise after the second message.

If the same thing keeps happening, as the resync count rises, the degree of scepticism increases to match and the digitiser waits longer and longer before resynchronising. In the example status stream below

```
2016  6 14 01:52:00 Clock check 8 0 Not sync'd  -1secs error
2016  6 14 01:52:00 Auto 2D
2016  6 14 01:53:00 Clock check 9 0 Not sync'd  -1secs error
2016  6 14 01:53:00 Auto 2D
2016  6 14 01:54:00 Clock check 10 0 Not sync'd  -1secs error
2016  6 14 01:53:59 Clock sync'd to Reference =>> 2016  6 14
01:54:01 .
```

the process is at an advanced state. At this point, the GPS time has jumped unaccountably (compared to digitiser time) ten times since the last boot-up and the digitiser has had to resynchronise ten times to match. It has just waited ten minutes before synchronising and, if the situation were to recur, it would then wait eleven minutes.

Eventually, if the counter reaches sixteen, the digitiser stops trusting GPS time and disables resynchronisation on the grounds that something is very wrong and the user would probably prefer contiguous data with incorrect time-stamps to salami-sliced data with random time-stamps, which would be very difficult to reassemble. This condition is signified in the status stream by a message like

```
2015 10 18 22:56:58 Not sync'd and Re-sync Disabled!
```

You can reset the counter by rebooting the digitiser but it is important to diagnose and correct the underlying problem.

- If the GPS receiver cannot output the correct time for a very lengthy period, the digitiser's clock may drift so far that, when a GPS lock is re-established, a synchronisation (jump) rather than a correction (slew) is appropriate. This may happen if the receiver's view of the sky is significantly obscured by, for example, vegetation.
- If the crystal in the digitiser's clock has aged and deteriorated to the point where the control voltage cannot bring it into lock-step with GPS time, frequent resynchronisations may be observed. This condition can also be

recognised by constant extreme values of the pwm parameter as reported in the status stream. A physical repair of the DM24 is required in this case.

- Some Trimble Lassen GPS receivers would announce a time which was exactly one second out for a minute or so after power up. This is because they used a hard-coded value for the leap-second offset (see section 7.2.2 on page 100) until the actual offset value was received from the satellites. When used in GPS power-cycling applications, this would cause the symptoms described (with a particularly fast escalation) because the DM24's software clock would need to jump one second backwards every receiver power-cycle and then one second forwards again each time once the offset was corrected. The solution in this case is to disable GPS power-cycling or replace the GPS receiver with a modern version.

7.3 Updating firmware

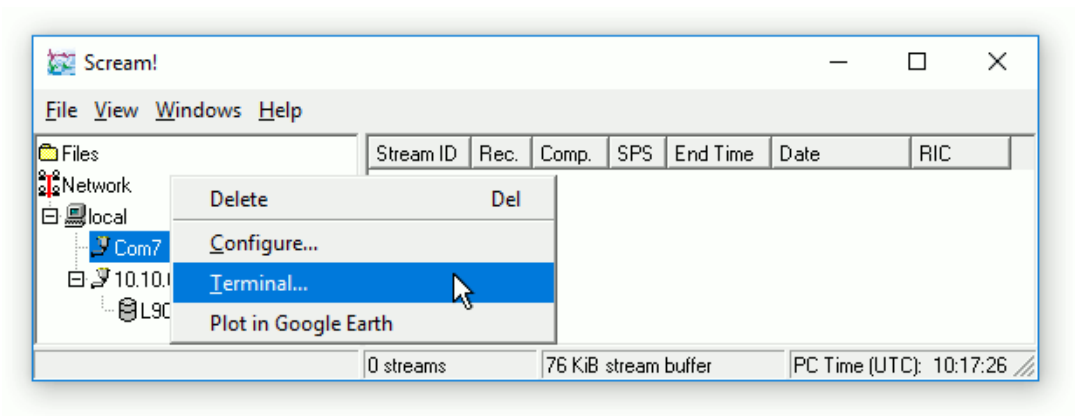
7.3.1 Updating firmware using an EAM, Affinity or NAM

The easiest way to update the digitiser firmware is by connecting it to a Platinum device, such as an EAM, Affinity or NAM. If the digitiser is part of a DM24-S*x*EAM or integrated *TDE instrument, this connection is already made.

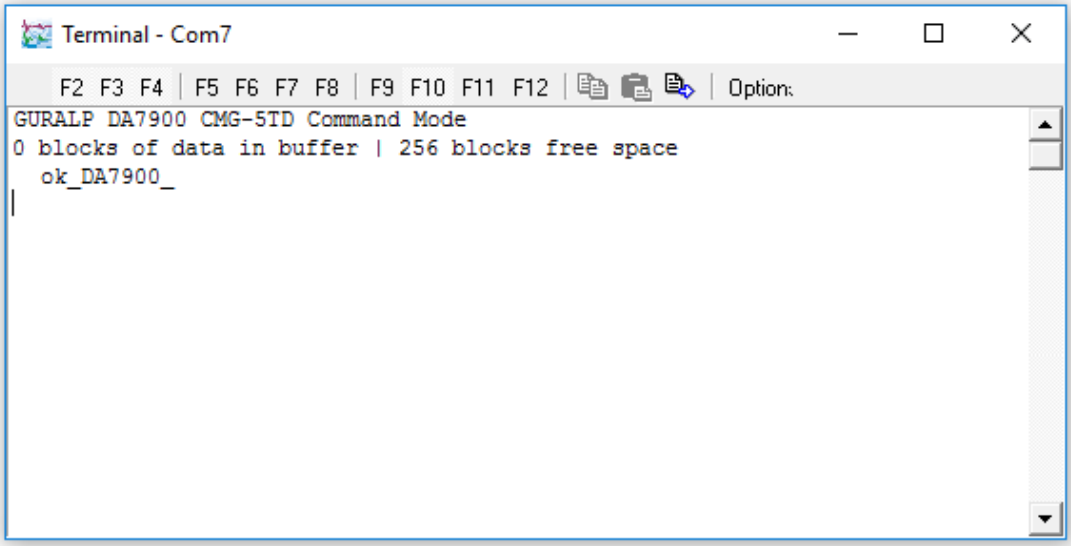
The DM24 firmware can be upgraded through the web interface, using a check-box at the bottom of the data acquisition configuration page, or from the command line, with the `dm24-upgrade` command. Please see the platinum manual, MAN-EAM-0003 for more details.

7.3.2 Updating firmware with Scream!

You can update the digitiser firmware using any terminal program which supports the Xmodem protocol, such as `minicom`, `ExtraPuTTY` or Scream!. If you are not using Scram, open a terminal to the output of the DM24 and key Ctrl + S a few times until you see the `ok` prompt. If you are using Scream!, simply right-click on the digitiser's icon in the main window and select **Terminal** from the pop-up menu:



A terminal window opens, presenting you with the command line of the digitiser:



If you are using a different emulator, open a window as normal with the appropriate Baud rate. The communications settings should be 8-N-1 with no flow control.

Check that there is two-way communication with the digitiser by pressing Enter. The digitiser should reply with `ok` on a new line.

Type **`re-boot`** to reinitialize the digitiser, and confirm with **`y`**. As it is restarting, the digitiser will report its status over the terminal connection, followed by a maintenance menu:

```
MPE ARM ANS ROM PowerForth v6.30
ARM Serial BootStrap v1.100, 11 August 2003
Copyright (c) 2002-3  GSL, EDSL & MicroProcessor Engineering Ltd.
Port 0 38400 baud  Port 1 4800 baud  Port 2 38400 baud

Guralp Systems Ltd - ARM-BOOT v3.0 mgs  13/12/12 (Build _02)
System Code versions loaded :-
Current  0105:0000 Guralp Systems Ltd - DM+FW  v.107  mgs  13/09/16 (Build _26)
Backup   010E:0000 Guralp Systems Ltd - DM+FW  v.107  mgs  13/09/16 (Build _26)
Previous 0117:0000 Guralp Systems Ltd - DM+FW  v.106  mgs  11/11/14 (Build 57w)
DSP Code :
0103:0000  dsp1091.bin loaded 2016-09-29   Default
0104:0000  dsp1091.bin loaded 2016-09-29
Command keys:
C - set real time Clock (2016 9 29 14:37:06 )
I - view/upload InfoBlock
F - run the Forth monitor
S - update System program
O - select Other system program
B - update Boot program
D - update DSP code
T - Toggle default DSP code
Q - Quit maintenance system
5  seconds to auto-start:
```

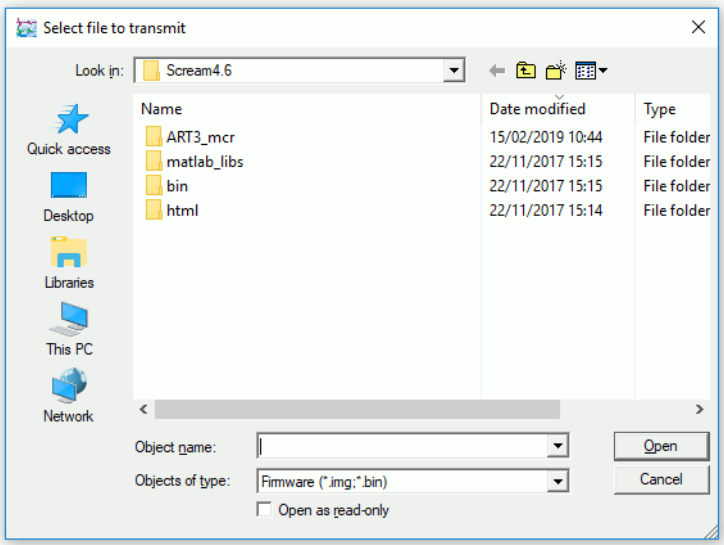
If you do not press a key in the next five seconds, the DM24 will start up normally. If you need more time to read the menu, press the space-bar.

The DM24 has three firmware components, which can be updated separately: the system program, the DSP code and the boot loader. Firmware releases are available on the Güralp Systems website (Select "Firmware" from the "Support" menu). If a release updates more than one component, you must update the boot loader first, followed by the system program and, finally, the DSP code.

7.3.2.1 Loading a new boot loader

Press B at the maintenance menu to update the boot-loader.

The digitiser will then request a transfer using the Xmodem protocol. If you are using Scream!, a file browser window will appear automatically.



Navigate through the directories on your computer and select the file to be uploaded, or type in its full path and file name. Click Open.

Caution: Be sure to check beforehand that the file is a valid DM24 boot loader! Replacing the boot loader with invalid code will make it impossible to boot up the digitiser. If in any doubt, please contact Güralp Systems technical support for advice before proceeding.

Whilst the file is loading, a progress window will be displayed.

Once the file is fully transferred, the DM24 will need to reboot. It will ask you for confirmation: Press ENTER, and wait whilst the system boots.

When you are returned to the maintenance menu, you can proceed to install other firmware modules or key ENTER, which will cause the system to continue to boot.

7.3.2.2 Loading a new system program

To update the system firmware, press S at the maintenance menu. The digitiser will display

```
Update System program
```

The digitiser will then request a transfer using the Xmodem protocol, as described above.

Whilst the file is loading, a progress window will be displayed. Depending on the speed of the link, it may take up to 20 minutes to transfer the firmware.

Once the file is fully transferred, the DM24 will return to the maintenance menu. Press ENTER, and wait whilst the system boots.

7.3.2.3 Loading new DSP code

Press D at the maintenance menu. The digitiser will reply with

```
Enter 0/1 to select DSP code to update
```

Select which of the two DSP code slots you want to overwrite, and press ENTER. The default is always 0.

```
Enter Filename/date - upto      31 characters
```

You can enter a descriptive string for the DSP code here. The DM24 will print this string at every boot-up, to remind you which version of the DSP code you are using.



Note: The EAM and other automated systems populate this with a string like:

```
dsp1091.bin  loaded 2016-09-23
```

When you press ENTER, the digitiser will then request a transfer using the Xmodem protocol, as described above.

7.4 Setting up external triggering

Güralp digitisers and digital instruments can be supplied with external triggering capabilities installed. The external trigger system is designed for maximum flexibility: you can trigger other digitisers or your own equipment using built-in relays, or any equipment may be used as a trigger source. In addition, a trigger circuit can link together any number of digitisers so that they trigger simultaneously when they receive a signal.

7.4.1 Overview

The digitiser or digital instrument has two internal components related to triggering.

- The trigger generator runs the triggering algorithm and determines whether a trigger has occurred. If external trigger *output* has been enabled, the trigger generator also operates a relay which connects the two *Trigger Out* pins of the *DATA OUT* port. These are normally pins E and F.

  The *Trigger Out* pins are only connected whilst the *generator* is active. This does not include any pre-trigger or post-trigger period, which is dealt with by the trigger receiver. A trigger signal may be as short as one second.

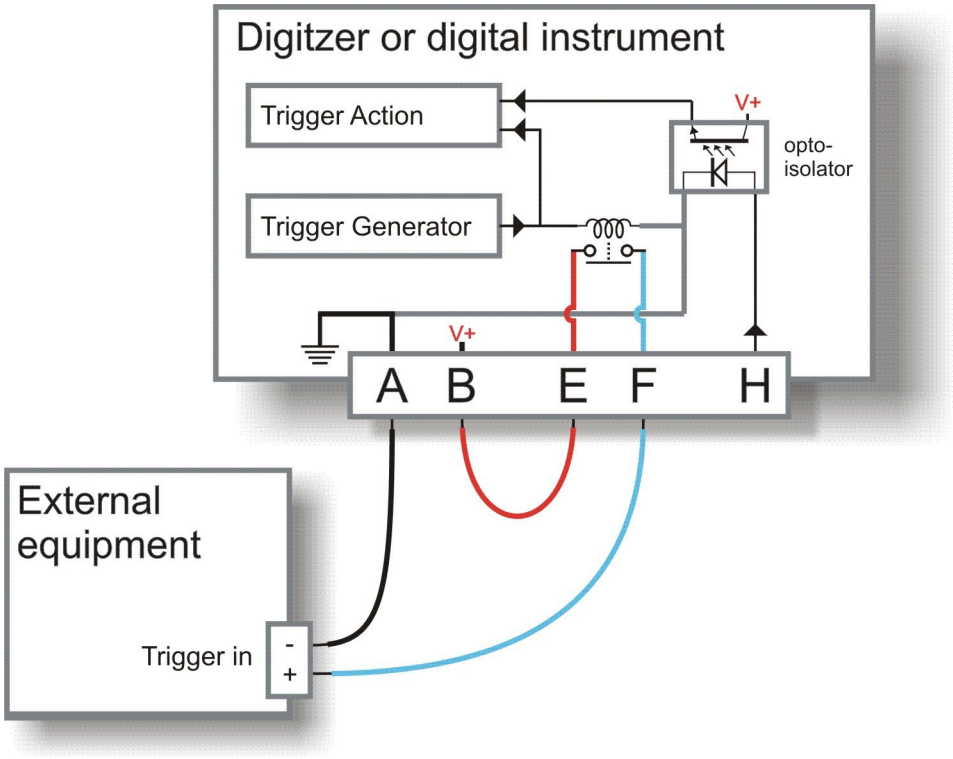
- The trigger receiver acts upon trigger signals: normally, by recording additional data streams. The receiver will always start recording if the trigger generator produces a signal. If external trigger *input* has been enabled, the receiver will also act on a logic signal received on the *Trigger In* pin of the *DATA OUT* port.

  Any signal between +3 and +30 V referenced to power ground (pin A) can be used. The instrument's own power supply (pin B) could be connected via a relay to supply this signal, for example.

The *Trigger Out* pins are *not* connected if you provide a *Trigger In* voltage. The instrument must trigger *itself* for the relay to switch. This arrangement prevents any trigger loops from occurring.

The diagrams on this page show the additional connections you will need to make. Pins A and B, and the data pins, should be connected to your power and data systems as normal.

7.4.2 Triggering your own equipment



If your equipment can trigger from 24 V DC, the simplest arrangement is to use the voltage across the sensor's power supply as a trigger.

- Connect the ground pin A to your equipment's trigger return line. If your equipment does not have a separate trigger return line, consult its documentation for how to apply trigger voltages. Güralp digitisers use the power ground line as a trigger return. (The diagram above does not show the power supply to either the sensor or the triggered equipment: only the triggering system is shown).

  Because a trigger signal may last for only a short time, inserting pins E and F directly into your equipment's power circuit is not normally advisable. Ideally, the equipment should be continuously powered and listening on dedicated trigger input lines. If this is not possible, it may be enough to build a control circuit with a time-out period, which supplies your equipment with power for a suitable minimum length of time whenever pins E and F are connected.

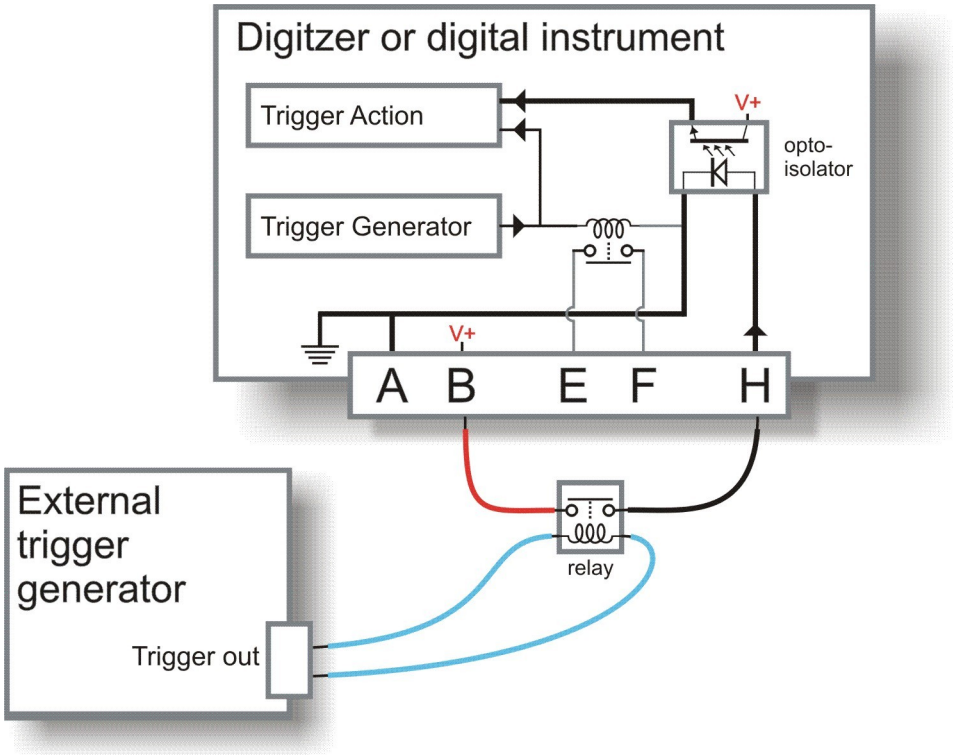
  The digitiser cannot anticipate a trigger. Pre-trigger recording is achieved using a continuously-updated buffer of the most recent data. Any external equipment must have its own buffering capabilities if you need pre-trigger data.

- Connect pin B to pin E, and pin F to your equipment's trigger signal input.

- Connect the remaining pins as normal.

If your equipment needs a different voltage, you will need to provide a separate voltage source for the triggering system. **Be careful not to connect two power supplies together.** The instrument's power supply can be protected by inserting a diode from pin B to pin E (see below).

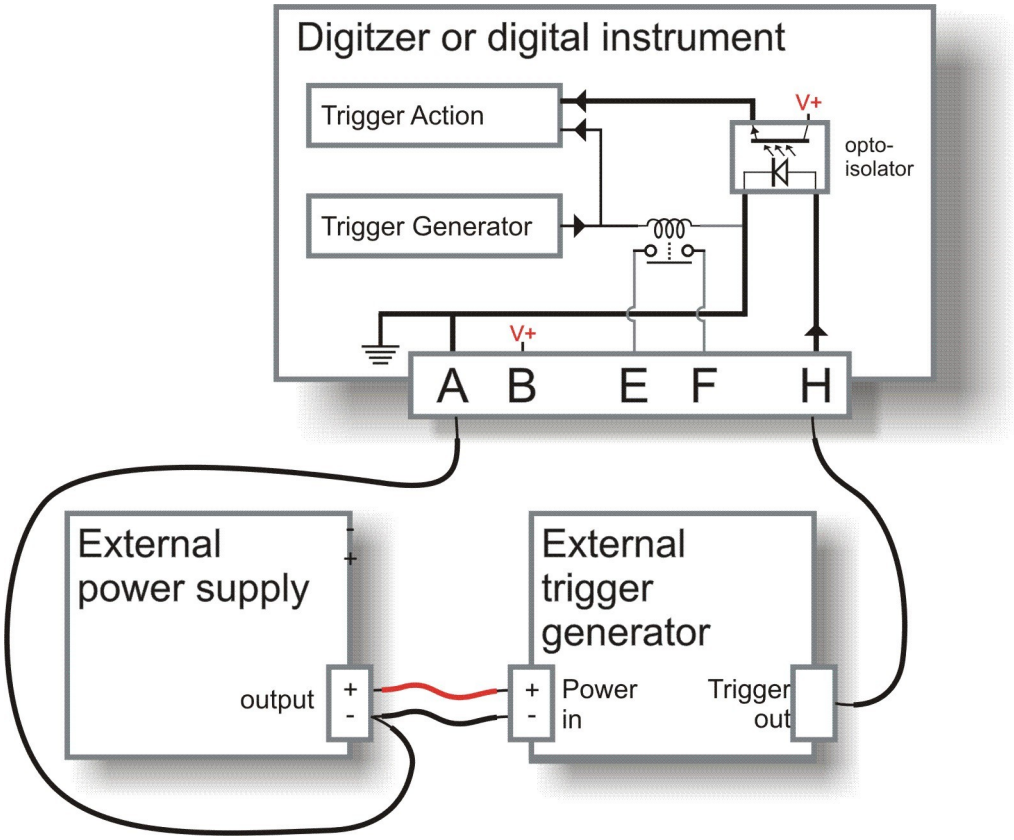
7.4.3 Using an external trigger source



To trigger an instrument from an external source through a relay:

- Connect a the normally-open switch contacts of a suitable relay between pins B and H, with the coil connected to your generator so that pins B and H are connected when a trigger occurs.
- The relay allows you to power the external generator from any source. This includes the instrument's own power supply at pins A and B, if it is suitable.
- Connect the other pins as normal.

If you prefer, you can use a separate power source for your trigger generator. However, because pin H is referenced to power ground, the external power supply must share a common ground with that of the digitiser, so you should connect the power supply ground to pin A.



7.4.4 Triggering several Güralp instruments simultaneously

A common use of the external triggering feature is to ensure that all instruments in an array trigger at the same time. This can be achieved with the wiring layout below. With this arrangement, any instrument can generate a trigger, which will be passed on to all the instruments in the array.

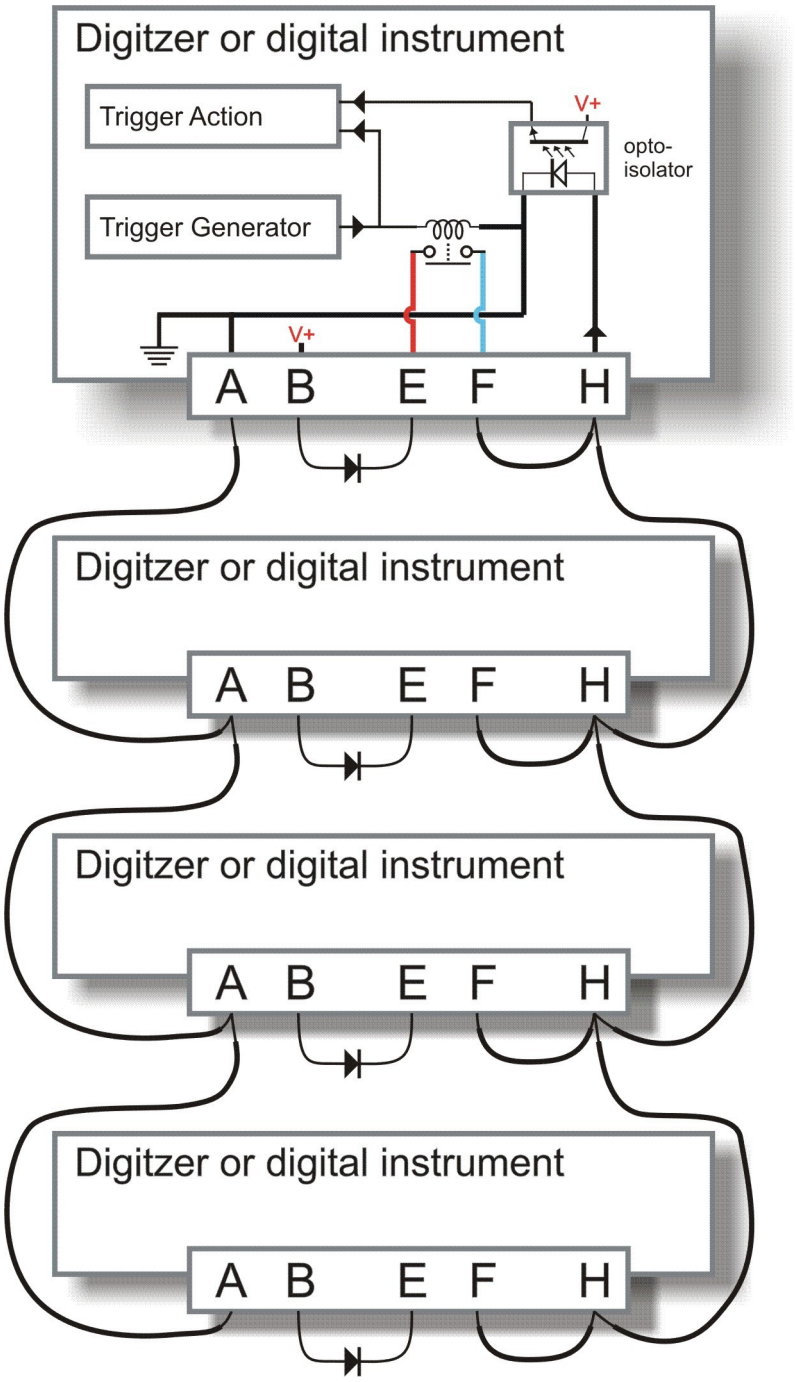
For each instrument:

- pin A is connected to a common ground;
- pin B is connected to the instrument's power supply as normal and also to pin E through a diode (an IN4001 is suitable);
- pins F and H are connected to each other and to a common trigger line; and
- the remaining data pins are connected to your recording equipment as normal.

The diode at pin E is important, because if two instruments generate a trigger simultaneously, their power supplies will both be connected via the internal relay contacts to the common trigger line on pins H, and hence connected together. This can cause considerable current flow with consequent risk of fire, as well as severe damage to equipment and cables.

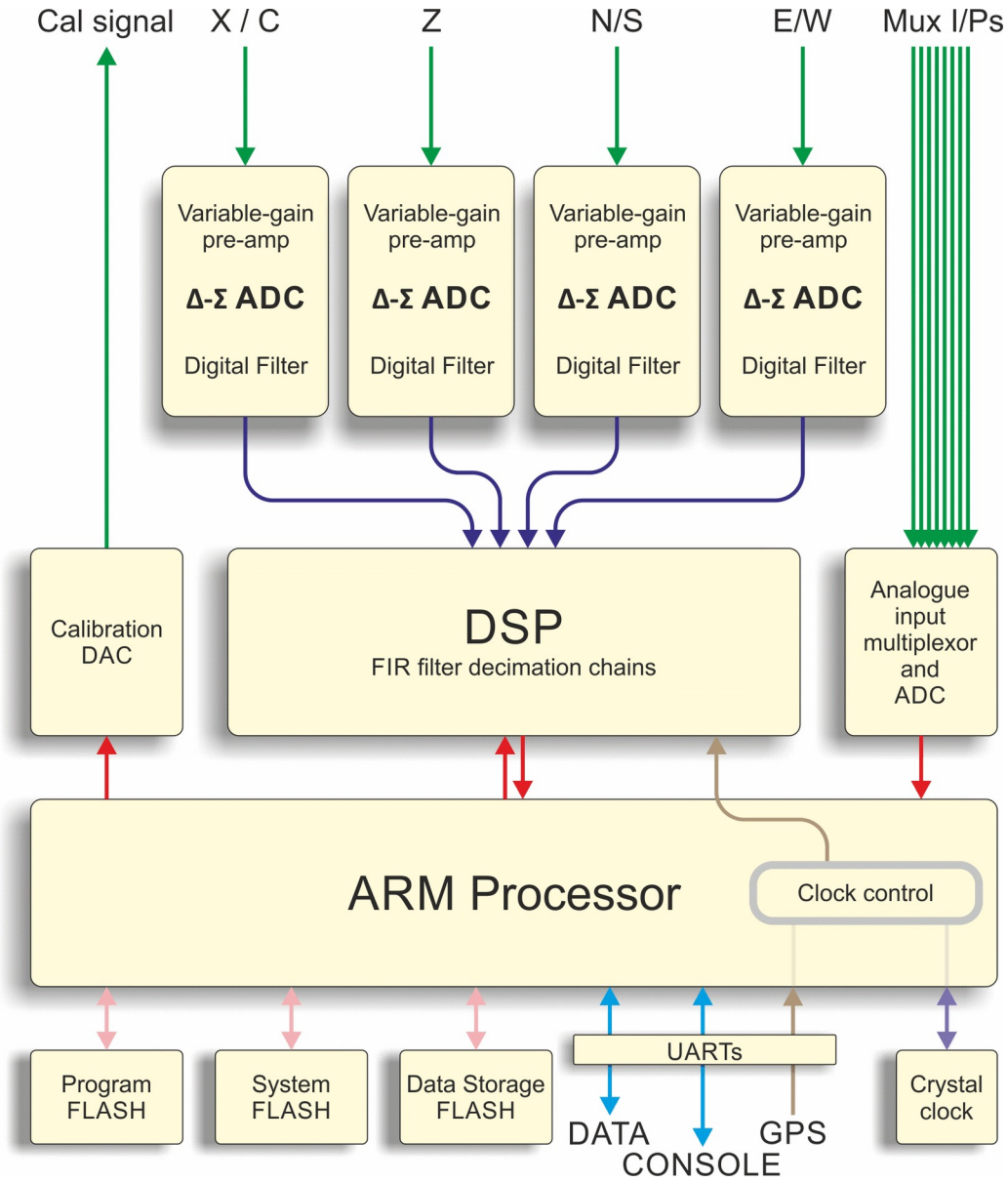
In the case where a single power supply is used to power all of the equipment, this is not an issue and pin E should be connected directly to pin B on all units. Pin B should then also be connected to the power supply's positive terminal.

The more common case, where each unit has it's own power supply, is shown below. Power supply wiring is omitted for the sake of clarity but note that all the power supply's ground terminals will be connected together via the links between pin A on each unit. The power supplies must, therefore, have floating outputs or grounded negative terminals.



7.5 Technical details

Internally, DM24 digitisers are structured as shown in the following diagram, where each box represents a separate internal subsystem.



The system is designed around a low power, high performance ARM microprocessor. This is a 32-bit processor with a large address space for data storage and manipulation. It also includes many integrated functions such as multiple timers and serial I/O ports. In addition, the system contains a Cirrus Logic CS5376 digital filtering chip-set and TMS320VC33 digital signal processor (DSP). The CS5376 provides data to the DSP at 2,000 samples per second, triggered by timing signals

from the ARM processor. The DSP can control all 5 ADCs and process the data in real time.

An important feature of the system design is its ability to synchronise the sampling of the analogue to digital converter to an external time reference. This way, data samples are accurately time stamped at source. To keep sampling accurately in step with UTC, the microprocessor's time-base is synchronized to an external reference, derived from GPS or, in larger arrays, to a centrally-transmitted time reference. Sharing a time reference avoids the cost and power consumption of multiple GPS receivers and, since it only involves sending 2 characters per second, it can utilise a low bandwidth link.

To achieve the high degree of timing precision required for a 24-bit digitiser system, the microprocessor time-base is run from a precision voltage controlled oscillator. On-board software keeps this oscillator tuned to the external reference so that its frequency is accurately set and maintained through changes in temperature or ageing. Once the system has stabilised, the control is sufficiently accurate to maintain precision sampling for several days without an external reference. The system also automatically compensates for the pure time delay introduced by the digital filtering/decimation processes in the DSP.

The DSP software consists of 6 cascaded programmable filter/decimation stages, which allow you to select multiple data output rates simultaneously. Each stage can be set individually for decimation factors of 2, 4, or 5. Data can be output at up to four different rates. For example, a system can be configured to provide data at 200, 50, and 10 samples per second, covering the whole of the seismological broad band range. The configuration of the DSP is programmable in the field *via* the host ARM microprocessor.

The DM24's interface to the world is via its serial ports. Port 0 is used for GCF data output. Port 1 is the NMEA input from the GPS receiver and Port 2 is the system's console. The console can be used as a command line interface without interrupting data flow via Port 0.

8 Güralp Compressed Format

Güralp Systems sensors and data modules use Güralp Compressed Format (GCF) to share seismic data. The format can be used for data storage or transmission over a serial link or TCP/UDP network.

This section describes the specification of GCF format.

8.1 GCF blocks

A GCF file or stream consists of a sequence of *blocks*, which can be up to 1024 bytes long. Each block consists of a 16-byte header followed by either

- a series of data records, containing initial and final sample values and a sequence of first differences between intervening sample values, or
- status information in ASCII text format.

The format of the block's body is determined by information in the header.

8.1.1 The GCF header

The header is 16 bytes long, split into four 4-byte fields:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
System ID				Stream ID				Date code				Data format			

System ID : If the top bit of this field is unset, the bottom 31 bits of this field specify a label of up to 6 characters identifying the originating system, encoded as a base 36 number. The base 36 digit in each position corresponds to a single character (0 – "`0`" … 9 – "`9`", 10 – "`A`" … 35 – "`Z`"), with the least significant digit placed at the right-hand end of the string.

If the most significant base 36 digits are zero, they are omitted from the string; hence the encoding 1 corresponds to the string `1`, not `000001`.

For example, the string `HPA1` would be encoded as the number (17 × 36 × 36 × 36) + (25 × 36 × 36) + (10 × 36) + 1 = 825913.

The field is a signed 31-bit positive integer, allowing System IDs up to ZIK0ZJ.

If the top bit of this field is set, the bottom 26 bits of this field specify a label of up to 5 characters using the same encoding. In this case, the field is a signed 26-bit positive integer, allowing System IDs up to ZZZZZ. Bits 26 – 30 of the System ID field are reserved in this case.

Stream ID : a unique 6-character label identifying the device, component and sample rate:

1	2	3	4	5	6
Device serial number				Component	O/P tap

The 6-character label is encoded in 4 bytes as a base 36 number, in the same manner as the *System ID*, above.

Date code : The date and time when the data in the block begin, expressed as a 32-bit number where

- the bottom 17 bits are the time in seconds since midnight, normally between 0 and 86399 but possibly 86400 or 86401 in years with 'leap seconds'; and
- the top 15 bits are the Güralp day number, with day zero on 17 November 1989. The day number increments when the seconds number rolls over at midnight.

Data format : The final header element contains 4 bytes defining the format of the data in the block:

1	2	3	4
reserved	Sample rate	Compression code	Number of data records

Sample rate : The sample rate of the data records, an integer number of samples per second. If this field is zero, the block body contains status information in ASCII text format.

Compression code : A code for the compression format used for all the data in the block. Currently accepted values are:

1	Data records contain one 32-bit difference
2	Data records contain two 16-bit differences
4	Data records contain four 8-bit difference

Other values are reserved for future expansion.

Number of data records : The number of 32-bit data records in the block. Combining this with the compression code enables you to find out the total number of sample points in the block; dividing by the sample rate allows the time duration of the block to be determined. You should check for status blocks (which have *Sample rate* = 0) before performing this calculation. For status blocks, the number of characters is *Number of data records* × 4.

The block duration is always a whole number of seconds, and always starts on a whole second boundary. A data block has a maximum size of 1024 bytes. If the

compression algorithm changes the compression format, blocks may appear which are not filled to the maximum capacity.

The rest of the block contains the data fields.

8.1.2 The GCF body

The body of a GCF block contains

- the absolute value of the first sample, a 32-bit signed integer referred to as the Forward Integrating Constant or FIC (this is not a data record, and is not counted in the Number of data records);
- a series of differences between following samples—32-bit, 16-bit or 8-bit signed integers according to the Compression code;
- the absolute value of the last sample, a 32-bit signed integer, referred to as the Reverse Integrating Constant or RIC. The RIC can be used as a checksum, as it should match the last decompressed sample value. It is not a sample in itself.

Because the value of the first sample is used as the FIC, the data record immediately after it is always zero.



Note: When data from a 24-bit digitiser is compressed into GCF blocks and the digitiser selects compression code 1 (32-bit data records) and the block is being transmitted over a serial link, the digitiser may truncate the 32 bit differences to 24 bits by omitting the top byte of each difference. This does not apply to true 32-bit data, even for blocks where the differences would fit into 24 bits.

This optimization is made purely to use the serial link more efficiently, and is not part of the GCF file format. When such data are stored on a disk or transmitted over a network using TCP or UDP, they must be expanded back to the full 32 bits.

Because 24-bit values have a range of 16 million (±8 million), differences between them must be able to indicate ±16 million, which is 25 bits. However, the serial compression code discards the 25th bit. When decompressing the data, you will need to reconstitute this bit.

You can check that you have correctly reconstituted the sign bits by comparing the value you obtain at the end of the GCF block with the RIC.

8.2 Sending GCF streams over a network

GCF blocks can be embedded in TCP or UDP packets for transmission over a network to other Güralp instruments or computers running GCF-compatible software such as Scream!.

A GCF server is expected to understand control messages, and to respond with data as appropriate. This allows clients to start and stop data transfer and to request that any missing blocks be re-transmitted.

8.2.1 Initiating a connection

The default behaviour of a GCF server is to listen for UDP packets containing GCF commands. A client sends a command as a null-terminated string in a single UDP packet. The commands a client can issue and the expected responses are:

- *GCFPING :* acknowledge this packet;
- *GCFSEND:B :* start sending data in big-endian ("network") byte order;
- *GCFSEND:L :* start sending data in little-endian ("Intel") byte order;
- *GCFSEND :* start sending data, in default byte order (Scream! interprets this as *GCFSEND:B*).

The server responds to any of these with a packet containing the string *GCFACKN*, again null-terminated.

A client should continue issuing *GCFSEND* packets periodically while it still wants to be sent data. If a client does not issue a *GCFSEND* packet for a long period, the server will stop sending data to it.

If the server has active clients when it shuts down, it sends a packet containing the string *GCFNOSV* to all of the clients.

8.2.2 Data packets

Once the client has sent a *GCFSEND* packet, the server will start sending it GCF blocks, each in a single UDP packet. There are two formats used, “Version 31” and “Version 40”.

8.2.2.1 Version 31 data packets

1024 octets – The GCF block as described above.

1 octet – The version number (decimal 31)

1 octet – The length of the following string, in octets

32 octets – A string describing the source of the data, padded to length 32 with zeros. The string takes the form STREAM-ID/COMxx/HOSTNAME where COMxx is the name of the serial port (COM1, etc.) and STREAM-ID is as in the GCF block (see above)

2 octets – The sequence number, in the byte order specified below (same as for data)

1 octet – A code for the byte order, 1 = big-endian (default), 2 = little-endian

8.2.2.2 Version 40 data packets

1024 octets – The GCF block.

1 octet – The version number (decimal 40)

1 octet – A code for the byte order, as above

2 octets – The sequence number, in the byte order specified

1 octet – The length of the following string, in octets

48 octets – A string describing the data source, as above but padded to length 48 with zeros.

8.2.3 Recovering missing data

Because the UDP packets have sequence numbers, the client can detect when a packet has been lost. To recover this data, it initiates a TCP connection to the server, on the same port number. If the server accepts the connection, the client sends a single-byte command to the server. The commands available are:

`0xf9` *(249) :* Requests the server to use TCP only when communicating with this client. Subsequent GCF data packets will be sent over the active TCP connection, with the same format as the normal UDP packets.

This feature is supported by the server in Scream! 4.0 and above.

`0xfc` *(252) :* Requests a version string, returned in PASCAL format (*i.e.* first octet = length, remainder = null-terminated string)

`0xfe` *(254) :* Requests the oldest sequence number held by the server. The sequence number is returned as 2 octets in big-endian byte order

`0xff` *(255)* `sequence-number` *:* Requests a block by sequence number. The block is returned on the TCP connection in one of the above formats. If the block is not available, the server returns `0xff 0xff 0xff 0xff`. The sequence number is 2 octets long, with big-endian byte order.

When the client has all the blocks it needs, it is free to close the connection.

8.3 Sending GCF streams over a serial link

Before it can be sent over a serial link, each GCF data block must be packaged in a 'transport layer', which consists of a 4-byte transmission header and a 2-byte checksum tail.

8.3.1 The transport layer

The transmission header consists of 4 bytes:

1	2	3	4
ASCII G (0x47)	Block sequence number	Block size (MSB)	Block size (LSB)

Block sequence number : An unsigned integer, which increments by 1 after each block, wrapping round to 0 from 255.

Block size : The size of the block in bytes, excluding the transmission header and checksum tail, but including the GCF block header.

After the transmission header, the GCF block is sent (with a length equal to the number of bytes specified in *Block Size*), followed by a two-byte checksum. This value is the sum of all bytes in the block header and body, modulo 65536, and is presented in big-endian byte order.

To optimise the use of available transmitter bandwidth the transmitted data block is truncated to the actual data length. This reduction is only applied to the difference records - the first and last absolute values are still transmitted as 32 bit values.

8.3.2 ACK and NACK signals

After transmission of the block is complete, the transmitter waits a nominal 150 ms for either a *NACK* or *ACK* signal before starting to transmit the next block.

If it receives an *ACK*, transmission of the next block commences immediately. If *NACK* is received, the block is re-transmitted.

The receiver should read the block, calculate the data checksum, and compare that with the value sent in the transmission header. If the two values match, the receiver should send an *ACK* signal, otherwise a *NACK*.

ACK and *NACK* signals can be in two formats: an older 2-byte format and a newer 6-byte format. The 2-byte signal is simply the first 2 bytes of the 6 byte signal. When both the transmitter and receiver use the 6 byte version, the link is said to be operating BRP (Block Recovery Protocol), which allows recovery of lost data (from up to 255 blocks ago) as well as error correction on a per-block basis.

The format of an *ACK* or *NACK* message is

Byte	1	2	3	4	5	6
ACK	0x01	Stream ID LSB	^S (0x13) or 0	Stream ID	Stream ID	Stream ID MSB
NACK	0x02	Stream ID LSB	Block number	Stream ID	Stream ID	Stream ID MSB

Stream ID should match the 4-byte stream ID of the last received block. Each system is able to identify its own *ACK* or *NACK* by matching this ID. 2-byte systems only match the least significant byte and ignore the rest.

For a *NACK* message, *Block number* specifies which block number to rewind to. For an *ACK* message, the third byte is either unused (0), or `0x13` (ASCII DC3, XOFF or Ctrl + S) to switch to command mode.

DM24 digitiser units transmit data as complete blocks become available, without any flow control apart from *ACK* and *NACK*.

9 Connector pin-outs

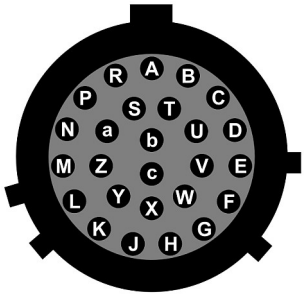
9.1 SENSOR A and B ports

These are standard 26-pin military-specification plugs, conforming to MIL-DTL-26482 (formerly MIL-C-26482). A typical part-number is 02E-16-26P although the initial "02E" varies with manufacturer.

Suitable mating connectors have part-numbers like ***-16-26S and are available from Amphenol, ITT Cannon and other manufacturers.



Pin	Function	Pin	Function
A	Vertical velocity +ve	**P**	Calibration signal
B	Vertical velocity –ve	**R**	Vertical calibration enable
C	N/S velocity +ve	**S**	N/S calibration enable
D	N/S velocity –ve	**T**	E/W calibration enable
E	E/W velocity +ve	**U**	Centre
F	E/W velocity –ve	**V**	*not connected*
G	Vertical mass position	**W**	Unlock
H	*not connected*	**X**	Lock
J	N/S mass position	**Y**	Logic signal ground
K	Busy indicator LED	**Z**	*not connected*
L	E/W mass position	**a**	*not connected*
M	*not connected*	**b**	Power 0 V
N	Signal ground	**c**	Power +10 to +24 V



Wiring details for the compatible socket, ***-16-26S, as seen from the cable end.

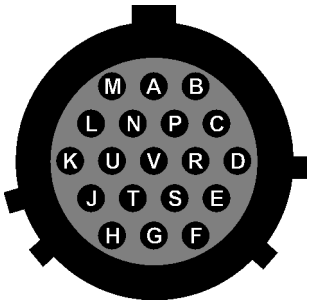
9.2 AUXILIARY port

This is a standard 19-pin military-specification plug, conforming to MIL-DTL-26482 (formerly MIL-C-26482). A typical part-number is 02E-14-19P although the initial "02E" varies with manufacturer.

Suitable mating connectors have part-numbers like ***-14-19S and are available from Amphenol, ITT Cannon and other manufacturers.



Pin	Function	Pin	Function
A	Optional *SENSOR B* auxiliary / calibration channel +ve	L	*SENSOR A* signal ground
B	Optional SENSOR B auxiliary / calibration channel −ve	M	Optional *SENSOR B* Mux channel M6
C	Optional *SENSOR B* signal ground	N	Digital ground
D	Optional *SENSOR B* Mux channel M3	P	*not connected*
E	Optional *SENSOR B* Mux channel M4	R	*SENSOR A* Mux channel MB
F	Optional *SENSOR B* Mux channel M7	S	*SENSOR A* Mux channel MC
G	*not connected*	T	*SENSOR A* Mux channel MF
H	Optional *SENSOR B* Mux channel M5	U	*SENSOR A* Mux channel MD
J	Auxiliary / *SENSOR A* calibration channel +ve	V	*SENSOR A* Mux channel ME
K	Auxiliary / *SENSOR A* calibration channel −ve		



Wiring details for the compatible socket, ***-14-19S, as seen from the cable end.

9.3 GPS Port

This is a standard 10-pin military-specification plug, conforming to MIL-DTL-26482 (formerly MIL-C-26482). A typical part-number is 02E-12-10P although the initial "02E" varies with manufacturer.

Suitable mating connectors have part-numbers like ***-12-10S and are available from Amphenol, ITT Cannon and other manufacturers.



Pin	Function
A	Power 0 V
B	Power +12 V
C	1pps signal
D	*not connected*
E	Digitiser console transmit
F	Digitiser console receive
G	RS232 ground
H	Digitiser console ground
J	RS232 transmit to GPS
K	RS232 receive from GPS



Wiring details for the compatible socket, ***-12-10S, as seen from the cable end.

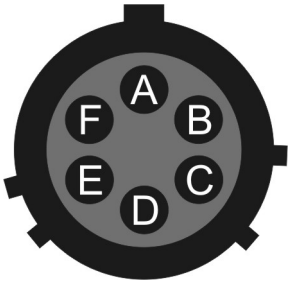
9.4 DATA IN port

This is a standard 6-pin military-specification plug, conforming to MIL-DTL-26482 (formerly MIL-C-26482). A typical part-number is 02E-10-06P although the initial "02E" varies with manufacturer.

Suitable mating connectors have part-numbers like ***-10-06S and are available from Amphenol, ITT Cannon and other manufacturers.



Pin	Function
A	RS232 transmit
B	RS232 receive
C	*not connected*
D	*not connected*
E	*not connected*
F	RS232 ground



Wiring details for the compatible socket, ***-10-06S, as seen from the cable end.

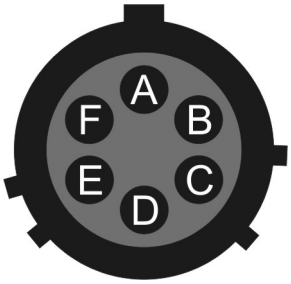
9.5 FIREWIRE port

When fitted, this is a standard 6-pin military-specification plug, conforming to MIL-DTL-26482 (formerly MIL-C-26482). A typical part-number is 02E-10-06P although the initial "02E" varies with manufacturer.

Suitable mating connectors have part-numbers like ***-10-06S and are available from Amphenol, ITT Cannon and other manufacturers.



Pin	Function
A	Power 0 V
B	TPA +ve
C	TPA −ve
D	TPB −ve
E	TPB +ve
F	Power + V



Wiring details for the compatible socket, ***-10-06S, as seen from the cable end.

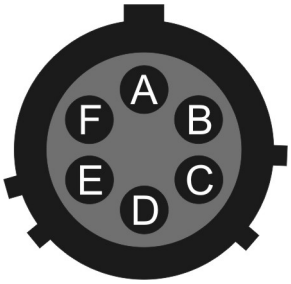
9.6 USB port

When fitted, this is a standard 6-pin military-specification plug, conforming to MIL-DTL-26482 (formerly MIL-C-26482). A typical part-number is 02E-10-06P although the initial "02E" varies with manufacturer.

Suitable mating connectors have part-numbers like ***-10-06S and are available from Amphenol, ITT Cannon and other manufacturers.



Pin	Function
A	+5 V DC (USB Type A pin 1)
B	Data −ve (USB Type A pin 2)
C	Data +ve (USB Type A pin 3)
D	0 V (USB Type A pin 4)
E	Shielding
F	Switched power +10 to +35 V



Wiring details for the compatible socket, ***-10-06S, as seen from the cable end.

9.7 DATA OUT port

This is a standard 10-pin military-specification plug, conforming to MIL-DTL-26482 (formerly MIL-C-26482). A typical part-number is 02E-12-10P although the initial "02E" varies with manufacturer.

Suitable mating connectors have part-numbers like ***-12-10S and are available from Amphenol, ITT Cannon and other manufacturers.



Note: The pin-out is the same as the serial output of a Güralp EAM, allowing you to insert a EAM into a pre-existing installation and maintain connectivity.

Pin	Function
A	Power 0 V
B	Power +10 to +36 V
C	RS232 CTS
D	RS232 RTS
E	External trigger output
F	External trigger output
G	RS232 ground
H	External trigger input
J	RS232 receive
K	RS232 transmit



Wiring details for the compatible socket, ***-12-10S, as seen from the cable end.

10 Digitiser specifications

Seismic channels	Number	4 – 8
	Format	32 bit
	Inputs	Differential with transient protection; ±10 V input range
	Input impedance	1 MΩ, 10 nF
	Common mode rejection ratio	110 dB at 10 Hz
Mux channels	Number and Format	Eight 24-bit channels
	Inputs	Single ended; ±10 V input range
GPS	Unit	External GPS (CMGGPS2)
	Power	supplied through *GPS* connector
	Time format	NMEA
	Max. cable length	50 m
Sensor control	Functions available	Lock, unlock and centre sensor (active low CMOS); enable calibration
Sensor calibration	Types	Sine and square wave; adjustable amplitude and frequency
Digital signal processing	Type	Texas TMS320CV33
	Hardware sampling rate	2 KHz
	Sample output rates	6 cascaded rates, each 2, 4, or 5 times lower than the previous rate
	Anti-aliasing filters	2 pole
	Low-pass filters	FIR
	Out-of-band rejection	140 dB
	In-band ripple	−140 dB
	Trigger modes	STA/LTA, threshold, external trigger
Digitiser performance	Output format	32 bit
	Noise-free resolution (NPR)	22.5 bits at 20 samples/s; 21.5 bits at 100 samples/s

	Absolute accuracy	0.5% standard (0.1% optional)
	Type	5^{th}-order, single-bit, low-pass noise shaper
Clock	Oscillator	8×10^{-7} standard (5×10^{-8} oven-controlled optional)
	External receiver interface	GPS or GLONASS: RS232 and PPS
	External receiver synchronisation	<50 µs
Internal thermometer	Accuracy	±0.33 °C (30 °C) ±0.5 °C (10..50 °C) ±1.0 °C (−10..75 °C)
	Linearity	±0.5 °C
	Resolution	0.0625 °C
Power	Supply	+10 to +24 V DC
	Current at 12 V DC	~0.12 A

11 Revision history

16/12/21	U	Changed description of `SET-ID` to require adding a trailing comma to work-around RT28026.
1/05/19	T	Separated timing-related commands from general configuration commands and added a discussion of GPS issues. Added `getWNROpivot`, `setWNROpivot` and `GPS-TYPE 4` commands. Changed transmission mode graphics and added a summary. Replaced most of "GCF Format" chapter with reference to the on-line protocol specification.
13/03/17	S	Added `AUTOCAL` and related commands.
22/09/16	R	Rebranded. Added new-style warnings, cautions and notes. Many minor alterations, clarifications and corrections. Updated many Scream! Screen-shots. Improved description of serial downloads. Amended `COMPRESSION`.
15/01/14	P	Amended `STOPBITS`, added `RESETLABELS`
18/11/09	O	Corrected USB information.
05/10/09	N	Corrected significant typographical error in format command for USB storage device
14/09/09	M	Replaced `ReadSCSI` instructions with those for `GCFXtract`. Enhanced connector pin-out documentation.
07/09/09	L	Added reference to EAMs and data-terminal command
14/05/09	K	Added `SERIAL2` and amended `SET-ID`
20/02/09	J	Improved style, corrected stream IDs, new graphics, etc
27/01/09	I	Minor Update
26/11/08	H	USB port use information added
23/11/06	G	Revised and reorganized
04/09/06	F	Firmware v103b12 (`SETDSS`)
19/05/06	E	Firmware v102b73 and DSP v1055 (`LOWLATENCY`)
21/04/06	D	Firmware v102b68
07/02/06	C	Firmware v102b58 (`GCFIP`, `EVENTSLIST`, *etc.*)
23/12/05	B	New functionality and corrected pinouts
16/08/05	A	New document