

OR/17/058; 1.0

Last modified: 2018/10/01 10:02



**British  
Geological Survey**  
NATURAL ENVIRONMENT RESEARCH COUNCIL

# Vibration monitoring and alarm system

Hazards Programme  
Internal Report OR/17/058



BRITISH GEOLOGICAL SURVEY

HAZARDS PROGRAMME

INTERNAL REPORT XX/00/00

# Vibration monitoring and alarm system

R. Luckett and J. Laughlin

## *Keywords*

Report; keywords.

## *Bibliographical reference*

R. LUCKETT AND J. LAUGHLIN.  
2015. Vibration monitoring and alarm system. *British Geological Survey Internal Report*, OR/17/058. 24pp.

Copyright in materials derived from the British Geological Survey's work is owned by the Natural Environment Research Council (NERC) and/or the authority that commissioned the work. You may not copy or adapt this publication without first obtaining permission. Contact the BGS Intellectual Property Rights Section, British Geological Survey, Keyworth, e-mail [ipr@bgs.ac.uk](mailto:ipr@bgs.ac.uk). You may quote extracts of a reasonable length without prior permission, provided a full acknowledgement is given of the source of the extract.

Maps and diagrams in this book use topography based on Ordnance Survey mapping.

## BRITISH GEOLOGICAL SURVEY

The full range of our publications is available from BGS shops at Nottingham, Edinburgh, London and Cardiff (Welsh publications only) see contact details below or shop online at [www.geologyshop.com](http://www.geologyshop.com)

The London Information Office also maintains a reference collection of BGS publications, including maps, for consultation.

We publish an annual catalogue of our maps and other publications; this catalogue is available online or from any of the BGS shops.

*The British Geological Survey carries out the geological survey of Great Britain and Northern Ireland (the latter as an agency service for the government of Northern Ireland), and of the surrounding continental shelf, as well as basic research projects. It also undertakes programmes of technical aid in geology in developing countries.*

*The British Geological Survey is a component body of the Natural Environment Research Council.*

*British Geological Survey offices*

### **BGS Central Enquiries Desk**

Tel 0115 936 3143 Fax 0115 936 3276  
email [enquiries@bgs.ac.uk](mailto:enquiries@bgs.ac.uk)

### **Environmental Science Centre, Keyworth, Nottingham NG12 5GG**

Tel 0115 936 3241 Fax 0115 936 3488  
email [sales@bgs.ac.uk](mailto:sales@bgs.ac.uk)

### **The Lyle Centre, Research Avenue South, Edinburgh, EH14 4AP.**

Tel 0131 667 1000 Fax 0131 668 2683  
email [scotsales@bgs.ac.uk](mailto:scotsales@bgs.ac.uk)

### **Natural History Museum, Cromwell Road, London SW7 5BD**

Tel 020 7589 4090 Fax 020 7584 8270  
Tel 020 7942 5344/45 email [bgs\\_london@bgs.ac.uk](mailto:bgs_london@bgs.ac.uk)

### **Columbus House, Greenmeadow Springs, Tongwynlais, Cardiff CF15 7NE**

Tel 029 2052 1962 Fax 029 2052 1963

### **Maclean Building, Crowmarsh Gifford, Wallingford OX10 8BB**

Tel 01491 838800 Fax 01491 692345

### **Geological Survey of Northern Ireland, Colby House, Stranmillis Court, Belfast BT9 5BF**

Tel 028 9038 8462 Fax 028 9038 8461

[www.bgs.ac.uk/gsni/](http://www.bgs.ac.uk/gsni/)

*Parent Body*

### **Natural Environment Research Council, Polaris House, North Star Avenue, Swindon SN2 1EU**

Tel 01793 411500 Fax 01793 411501  
[www.nerc.ac.uk](http://www.nerc.ac.uk)

Website [www.bgs.ac.uk](http://www.bgs.ac.uk)

Shop online at [www.geologyshop.com](http://www.geologyshop.com)

# Foreword

This report describes a system designed for use in nuclear power stations, or at other infrastructure sites, where a rapid response may be required when the ground acceleration caused by an earthquake exceeds a certain threshold. The system uses free-field acceleration measurement to trigger a physical alarm. The data acquisition and alarm functions are carried out on a PC that then triggers an alarm by use of a relay. The system described here has two such PCs for redundancy, although a single PC could perform the required task.

# Acknowledgements

The authors acknowledge the helpful and constructive reviews done by Rob Clark and David Hawthorn.

# Table of Contents

<b>Foreword</b> .....	<b>i</b>
<b>Acknowledgements</b> .....	<b>i</b>
<b>Table of Contents</b> .....	<b>i</b>
<b>1 Introduction</b> .....	<b>1</b>
<b>2 Hardware</b> .....	<b>2</b>
2.1 Sensors and Digitisers .....	2
2.2 Serial splitter function .....	3
2.3 Acquisition Computers .....	3
2.4 Watchdog Relay function .....	3
<b>3 Scream!</b> .....	<b>6</b>
<b>4 Earthworm</b> .....	<b>7</b>
4.1 startstop.....	8
4.2 statmgr .....	9
4.3 scream2ew .....	9
4.4 accel_alarm.....	10
4.5 switch_relay.....	11
4.6 email_alarm .....	12
4.7 import_generic, export generic.....	13
4.8 watchdog.....	14
<b>5 Other Software</b> .....	<b>14</b>
5.1 Mailer.....	14

5.2 Email server ..... 15

5.3 Perl scripts ..... 15

**6 Testing ..... 15**

**7 Summary ..... 17**

**Appendix I..... 18**

**Table of Figures**

Figure 1. Schematic of the alarm system with two accelerometers attached. .... 2

Figure 2. Rack mounted ‘alarm control unit’ – front panel view (LEDs)..... 3

Figure 3. ‘Alarm control unit’ - rear view (serial port and alarm connections)..... 4

Figure 4. Circuit diagram for the ‘alarm control unit’. .... 5

Figure 5. Earthworm window..... 7

Figure 6. Earthworm startstop configuration. .... 8

Figure 7. Earthworm statmgr configuration..... 9

Figure 8. Earthworm scream2ew configuration..... 10

Figure 9. Earthworm accel\_alarm configuration..... 11

Figure 10. Earthworm switch\_relay configuration. .... 12

Figure 11. Earthworm email\_alarm configuration..... 12

Figure 12. Earthworm import\_generic configuration. .... 13

Figure 13. Earthworm export\_generic configuration..... 13

Figure 14. Earthworm watchdog configuration. .... 14

Figure 15. Example alarm email sent from PC1 after high acceleration (values in  $\mu\text{g}$ ). ..... 15

Figure 16. Example alarm email sent from PC2 after high acceleration (values in  $\mu\text{g}$ ). ..... 16

Figure 17. Example diagnostic email sent on failure to connect with the other PC. .... 16

Figure 18. Example alarm email sent on failure to connect with the other PC..... 16

Figure 19. Example alarm email sent following instrument failure..... 16

# 1 Introduction

The system is designed to be used in conjunction with a number of free field accelerometers (that is instruments buried in pits, rather than connected to buildings) to monitor ground acceleration near sensitive infrastructure. If ground acceleration exceeds a threshold at a given number of sites, within a short time window, an alarm will be sounded and automated emails will be sent. The time window, number of stations and thresholds are all fully configurable. In the example given in this report two accelerometers are used. The vibration monitoring and alarm system software is configured so that the alarm will also sound if monitoring is interrupted. Such loss of monitoring includes data from too many accelerometers stopping, as well as a failure of the software. The software has been written in such a way that redundancy can easily be built in. Two computers each running an instance of the software can be run in parallel with each monitoring the status of the other. Failure of one computer or software instance will result in automated emails being sent but the alarm will only be activated if monitoring functionality is lost.

Each dedicated computer runs an instance of the system, which consists of two separate programs. The two programs are Scream! and Earthworm. Scream! is a data acquisition application provided by Guralp, the manufacturer of the accelerometers. Earthworm is a modular, open source data processing package. Earthworm modules have been developed by the BGS to achieve the required functionality. One of these carries out the alarm function. Another is a watchdog module that continually checks the status of Earthworm instances on other computers.

The dedicated acquisition computers have an identical software and hardware set up and receive data simultaneously from both accelerometers. Each computer is also connected to an alarm control unit, which is the hardware part of the alarm system. It monitors a pulse output on a serial port from each computer. The combined pulses are used to control a relay that stops the alarm sounding. If pulses from all the computers stop or power to the relay fails, an alarm is triggered. The Earthworm module that generates the pulse to control the relay was developed by the BGS.

In the following description, it is assumed that two computers are being used. Each reads in data from the same two accelerometers and has identical software set up in the same way. If either of the acquisition computers, or the software on it, fails then the pulse from that computer will stop. However, the alarm will only sound if both pulses stop – meaning that a single computer failure will not result in an alarm. It will, however, result in automated emails being sent by the other computer, which monitors a heartbeat generated by one of the Earthworm modules. If ground motion surpasses the acceleration threshold both pulses will stop and the alarm will sound. This will also happen if data stops from one or more accelerometer as this will render the system unfunctional.

In the event of an alarm, the trigger will need to be reset on each PC. This is done by clicking on a shortcut located on the PC desktop.

This report describes the alarm system in general terms. The system is fully configurable and installation specific configuration values are given in Appendix I.

## 2 Hardware

The hardware for the system described includes two free field accelerometers (that is instruments buried in pits) and two computers running the acquisition and alarm software. It is assumed that the data is transmitted from the pits to the acquisition system via serial communication lines. This requires a serial splitter to be used if the acquisition is to be carried out independently on two computers for redundancy. This report also assumes the need to interrupt an external circuit to activate the alarm. This requires relays that can be switched by a serial output from the acquisition computers. This functionality has been called the Watchdog relay functionality. The serial splitter and Watchdog relays are housed in the same cabinet, called the 'alarm control unit'. Finally, to allow BGS to monitor the alarm system and provide rapid advice to power station staff in event of an alarm or other relevant seismicity, an Ethernet router is included in the installation. Fig 1 shows the relationship between the hardware elements in the vibration monitoring and alarm system.

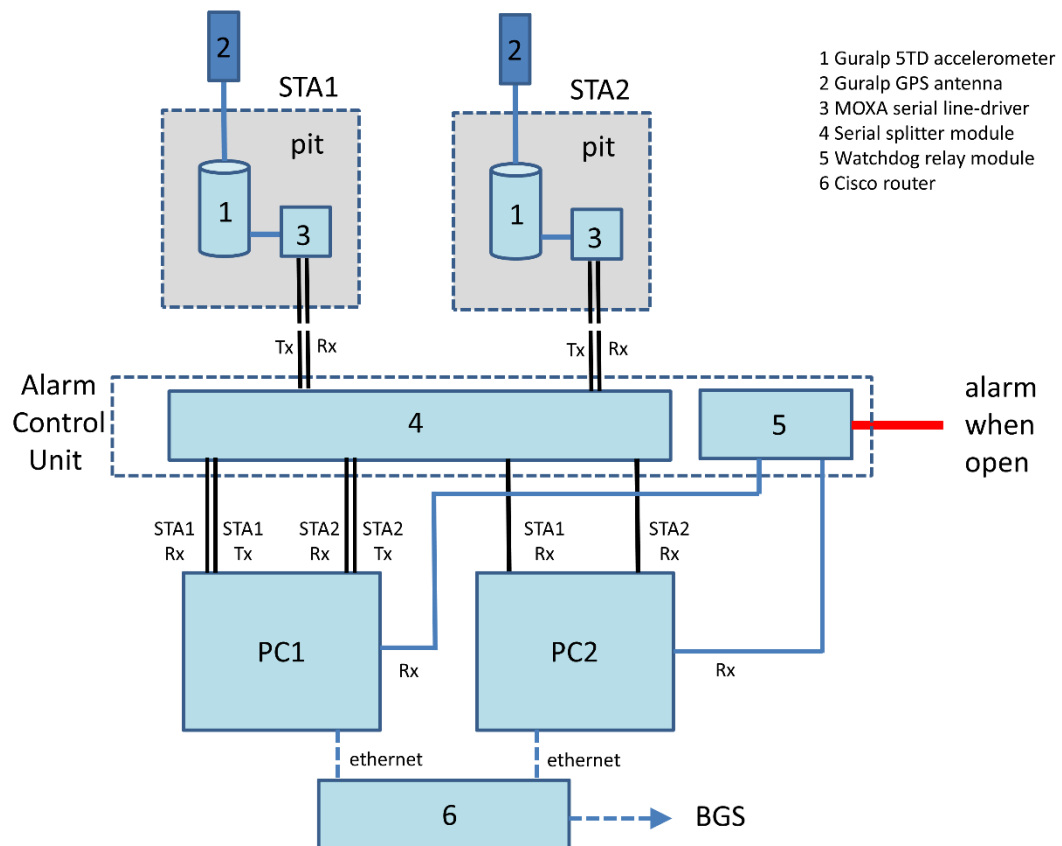


Figure 1. Schematic of the alarm system with two accelerometers attached.

### 2.1 Sensors and Digitisers

In the setup described, two CMG-5TD accelerometers, manufactured by Guralp Systems Ltd with built-in 24-bit digitisers (DM24) are used. This is typical of the instrumentation installed at power stations and other infrastructure by the BGS. The instruments are low-noise with a high dynamic range and would normally have full-scale sensitivity set to 1g. Timing is supplied by GPS. For this system to work as described Guralp digitisers must be used, as they are compatible with Scream!, but adapting it to use many alternative digitisers would be trivial.



## 2.2 Serial splitter function

Two accelerometers are used – the minimum number for the system to work. As two PCs are used, both must receive (Rx) input data from the two accelerometers via RS232 serial cables. Serial splitters are used to separate the transmit-receive (Tx Rx in Fig. 1) serial signals from each accelerometer into two separate data links. Only one PC (PC1 as shown in Fig. 1) is used to control the Tx line to the accelerometers in order to avoid conflict with the data transfer hand-shaking protocol used. This means that the two PCs behave identically when monitoring the incoming data but only the PC connected to the “PC1” output, as shown in Fig. 1, is used to control and send calibration pulses to the instruments. The serial splitter functionality is installed in the same ‘alarm control unit’ as the watchdog relay module.

## 2.3 Acquisition Computers

Two Amplicon rack mounted industrial PCs installed with a 32-bit embedded Windows operating system are used in the vibration monitoring and alarm system described here. The two computers are named BGS\_MONITOR\_1 and BGS\_MONITOR\_2. They are logged onto as *administrator* with the password BGSSe1smo. They are connected to a CISCO 800 series router that creates a local area network. This router can, in turn, be connected to an ADSL, satellite or 3G modem to connect it to the wider internet. The CISCO firewall is configured in such a way that the two PCs can be accessed by remote desktop using the public IP address provided by the ISP on ports 10011 and 10012 respectively.

## 2.4 Watchdog Relay function



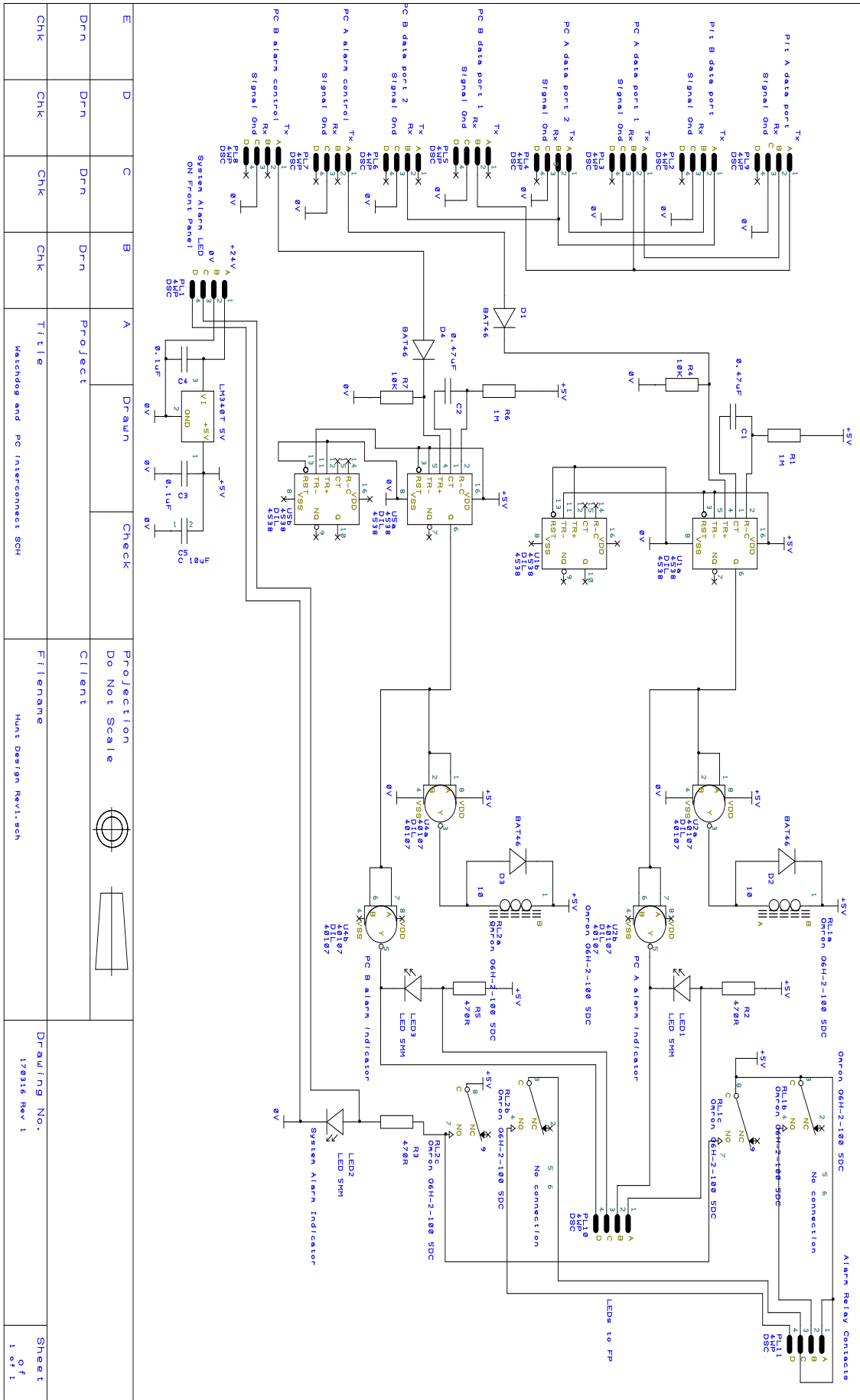
**Figure 2. Rack mounted ‘alarm control unit’ – front panel view (LEDs).**



**Figure 3. 'Alarm control unit' - rear view (serial port and alarm connections).**

Because there are two PCs, the configuration considered here has two Watchdog circuits, each one being triggered by pulses from one PC. Each Watchdog circuit controls a relay, which is kept energised in normal conditions. If both relays are de-energised then a third, higher current, relay is also de-energised and the alarm sounds. This provides the required behaviour; that both Watchdog circuits must trip to de-energise the final relay and provide an alarm. This has an added fail-safe, in that a failure of the relay hardware will also cause an alarm condition.

The Watchdog circuits and relays are housed in a rack mounted 'alarm control unit'. This has three LEDs on the front panel to indicate that the Control PC PC1 and the Monitor PC PC2 are alive and sending pulses. The central LED, when lit, indicates that at least one PC is alive and sending pulses. All three LEDs extinguished would correspond to an alarm condition. This box also contains the serial splitters described in section 2.2. The circuit diagram for the 'alarm control unit' is shown in Fig. 4. This shows both the serial signal from two accelerometers being split for the two acquisition PCs and the relays that control the external alarm.



E	D	C	B	A	Drawn	Check	Projection	Client	Do Not Scale		Drawing No.	Sheet
Drn	Drn	Drn	Drn	Project							176316 Rev 1	1 of 1
Chk	Chk	Chk	Chk	Title	Matches and PC Interconnect SCH		Filename	Hunt Design Rev1.sch				

Figure 4. Circuit diagram for the 'alarm control unit'.

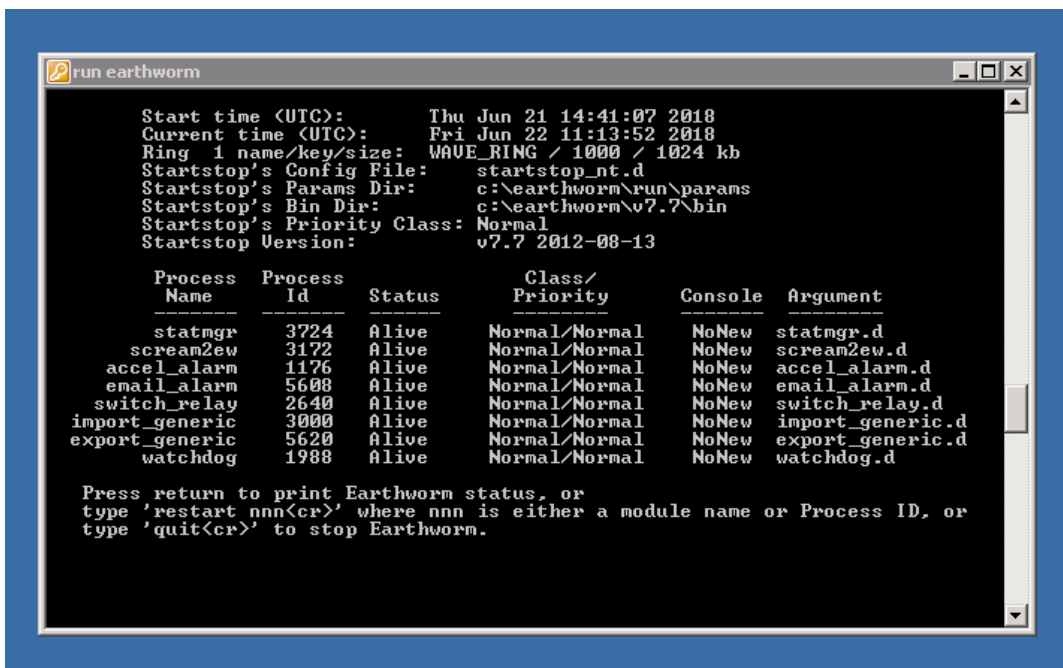
### 3 Scream!

Scream! ([www.guralp.com/documents/html/MAN-SWA-0001](http://www.guralp.com/documents/html/MAN-SWA-0001) - Scream! users guide) is a real-time data acquisition application developed by Guralp Systems Ltd to interact with their digitisers. The version provided with the alarm system is Scream! 4.5. The Scream software acquires data from the accelerometers connected to the acquisition PCs via serial ports. It acts as a network server to serve time series data both to the local Earthworm software (see section 4) and to clients situated at the BGS in Edinburgh. It records the data in one-hour miniseed files on the local disk (C:\data). This provides a 35GB (one year) buffer of data that is available in the event of a failure of the external communication link. In addition, Scream! provides an interface to the instruments that allows detailed monitoring of their state of health, such as the status of the GPS clock. It also allows calibration pulses to be administered. This interface can be accessed from clients of the application, such as those in Edinburgh. To start Scream! on the computers forming the alarm system, a shortcut named Scream! is clicked on the desktop. The configuration file is in the default location, which is c:\scream\scream.ini. One of the parameters in this file controls the size of the data buffer – that is how much disk space will be used by data files before Scream! deletes older files to make room for new ones.

## 4 Earthworm

Earthworm ([www.earthwormcentral.org](http://www.earthwormcentral.org) - central website for the Earthworm software community) is the most widely used real-time seismic data processing software for regional seismic networks. It was originally developed by the United States Geological Survey in 1993 and can now be considered both mature and robust. The version used in this installation is 7.7. Earthworm is open source software with a modular design, making it easy to add functionality without changing existing modules. In this system, it provides the alarm functionality. It sends a continuous 'keep alive' pulse to a serial port, which has a hardware Watchdog connected to it. This pulse is interrupted if the recorded ground acceleration exceeds configured limits or a problem occurs meaning that the acceleration is not being checked. It performs continuous state of health checks on both data and software and will send automated emails, both if an alarm condition is met and in cases where the system is working but redundancy has been lost. Each instance of Earthworm described here consists of nine modules that communicate with each other via a shared memory area known as a ring. A description of each module with its purpose and typical configuration is given in sections 4.1 to 4.8. The parameter files for all modules are in the directory `c:\earthworm\run\params`. These files have been opened in a text editor to give the figures below.

To start Earthworm on each of the computers forming the alarm system the `run_ew.bat` batch file is clicked on the desktop. This batch file changes to the correct directory, sets up environment variables (by running `ew_nt.cmd`) and then runs `startstop`, which opens a command window (Fig 5) and starts all the other modules. Once Earthworm is running, then individual modules can be restarted by noting their process ID from the list in the Earthworm window and typing, "restart `pid`". This is how, for example, the module `switch_relay` could be restarted to clear an alarm condition by reinitiating the pulse train to the serial port. There is also a script that allows a user to do exactly this, without any typing, by clicking on the `restart_switch_relay` icon on the desktop.



```

run earthworm

Start time (UTC):      Thu Jun 21 14:41:07 2018
Current time (UTC):   Fri Jun 22 11:13:52 2018
Ring 1 name/key/size: WAUE_RING / 1000 / 1024 kb
Startstop's Config File: startstop_nt.d
Startstop's Params Dir: c:\earthworm\run\params
Startstop's Bin Dir:   c:\earthworm\v7.7\bin
Startstop's Priority Class: Normal
Startstop Version:    v7.7 2012-08-13

  Process  Process  Status  Class/  Console  Argument
  Name     Id         -----  Priority  -----  -----
  statmgr  3724      Alive   Normal/Normal  NoNew  statmgr.d
  scream2ew 3172      Alive   Normal/Normal  NoNew  scream2ew.d
  accel_alarm 1176      Alive   Normal/Normal  NoNew  accel_alarm.d
  email_alarm 5608      Alive   Normal/Normal  NoNew  email_alarm.d
  switch_relay 2640      Alive   Normal/Normal  NoNew  switch_relay.d
  import_generic 3000      Alive   Normal/Normal  NoNew  import_generic.d
  export_generic 5620      Alive   Normal/Normal  NoNew  export_generic.d
  watchdog  1988      Alive   Normal/Normal  NoNew  watchdog.d

Press return to print Earthworm status, or
type 'restart nnn<cr>' where nnn is either a module name or Process ID, or
type 'quit<cr>' to stop Earthworm.

```

Figure 5. Earthworm window

## 4.1 startstop

This is a standard Earthworm module for starting the other modules and administering the shared memory. It will also give a status message for the whole Earthworm installation and allow individual modules to be restarted if necessary. Pressing return in the Earthworm window (Fig. 6) will cause startstop to write a status table showing a list of active modules and their process IDs. A module can be restarted by typing restart and the process ID for that module.

```
#
#           startstop Configuration File for Windows
#
nRing      1           # Number of transport rings to create.
Ring  WAVE_RING 1024   # Name of a ring followed by its size (kb)

MyModuleId MOD_STARTSTOP # Module Id for this program
HeartbeatInt 15         # Heartbeat interval in seconds
MyPriorityClass Normal   # For startstop
LogFile       2         # 0 STDOUT; 1 log and STDOUT; 2 log only
KillDelay     30        # Wait (sec) on shutdown for a child to die before killing it

Process      "statmgr statmgr.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
#
Process      "scream2ew scream2ew.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
#
Process      "accel_alarm accel_alarm.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
#
Process      "email_alarm email_alarm.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
#
Process      "switch_relay switch_relay.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
#
Process      "import_generic import_generic.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
#
Process      "export_generic export_generic.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
#
Process      "watchdog watchdog.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
```

**Figure 6. Earthworm startstop configuration.**

## 4.2 statmgr

This is a standard Earthworm module for overseeing the state of health of other modules. It monitors heartbeat messages from all modules and will restart a module if its heartbeat stops. It also reads error messages from other modules and can pass these on as emails. It is highly configurable, with a separate ‘descriptor’ file for every module giving instructions on how *statmgr* should treat each possible error from that module. In this case (Fig. 7), emails from *statmgr* are only sent to administrators within BGS, as, while possibly indicative of potential problems, they do not indicate a loss of functionality.

```
#
#           statmgr Configuration File
#

MyModuleId      MOD_STATMGR      # Module id for this instance of statmgr
RingName        WAVE_RING        # Ring to get RSAM samples and write messages
LogFile         1                 # 0 STDOUT; 1 log and STDOUT; 2 log only
CheckAllRings   1                 # Check all rings for status messages
MailServer      localhost
From            alarm@bgs.ac.uk
Subject         "BGS MONITOR 1"  # Subject line for the email messages

# Between 0 and 10 email recipients may be specified below.
mail rrl@bgs.ac.uk

# The modules whose heartbeats and error messages statmgr should monitor
#           Installation      Module      Message Types
GetStatusFrom  INST_BGS_MON_1  MOD_WILDCARD  # heartbeats & errors

# Pager functionality not used at BGS
heartbeatPageit 60                # pageit system heartbeat interval (sec)
pagegroup       ew-ops            # The pager group name.

# List of descriptor files which control error reporting.
# One descriptor file is needed for each earthworm module.
Descriptor statmgr.desc
Descriptor startstop.desc
Descriptor export_generic.desc
Descriptor import_generic.desc
Descriptor scream2ew.desc
Descriptor accel_alarm.desc
Descriptor email_alarm.desc
Descriptor switch_relay.desc
Descriptor watchdog.desc
```

Figure 7. Earthworm *statmgr* configuration.

## 4.3 scream2ew

This module was written by Guralp and was added to the main Earthworm distribution many versions ago. It receives real-time data from Scream! over a TCP connection. It converts from Guralp’s own format (GCF) to the native Earthworm format and makes the data available to other modules by placing it on a shared memory ring. This module needs to know the identities of the channels being used, which depend on the instruments deployed. A channel identity consists of a station name and a GCF ‘stream ID’. The station name is defined by the user when they configure a digitiser – this will not change if an instrument is replaced. The GCF ‘stream ID’ is a unique string identifying data from a particular channel (in Fig.8, **5675Z2** on the top *ChanInfo* line is a stream ID). This string is assigned by Scream! and depends on the serial number of the digitiser. If an instrument is replaced, the corresponding stream IDs for all three channels will need to be edited in *scream2ew.d*.

```

#
#          scream2ew Configuration File
#
MyModuleId      MOD_SCREAM2EW # module id for this instance of scream2ew
RingName        WAVE_RING     # shared memory ring for input/output
LogFile         2              # 0, STDOUT 1, log and STDOUT 2, log only
Verbose         1              # 1=> log every packet. 0=> don't
HeartBeatInterval 30          # Time in seconds between heartbeats
Server          127.0.0.1
PortNumber      1567

# These lines specify the mapping between the GCF system ID and stream ID and the earthworm
# network, station, location, channel and pin number (pin number is redundant).

ChanInfo "STA1 5675Z2 GB STA1 00 ENZ 1"
ChanInfo "STA1 5675N2 GB STA1 00 ENN 1"
ChanInfo "STA1 5675E2 GB STA1 00 ENE 1"

ChanInfo "STA2 C258Z2 GB STA2 00 ENZ 1"
ChanInfo "STA2 C258N2 GB STA2 00 ENN 1"
ChanInfo "STA2 C258E2 GB STA2 00 ENE 1"

```

**Figure 8. Earthworm scream2ew configuration.**

#### 4.4 accel\_alarm

This module was written at the BGS. For channels listed in its configuration file, it checks the value of each sample against an acceleration threshold. A moving average is kept for each channel and this average is subtracted from each value to account for the offset, which will change with time. If a given number of channels exceed their threshold within a configurable time window then an **ALARM\_ACCEL** alarm message is placed on the memory ring.

In this case (Fig. 9), such a message would be generated if any channel from one station exceeded 5000 $\mu$ g (threshold) within 20 seconds (AlarmWindow) of a channel from the second station doing the same. This module also checks for continuity of data. If no data is received from one or more stations within a configurable time window, or if data from any station exceeds an allowed latency, then an **ALARM\_NO\_DATA** alarm message is placed on the memory ring. In this example, such a message will be sent if data from either station stops for 100 seconds (NoDataAlarmTime) or if the data arriving from either station has a latency of more than 200 seconds (MaxChanLatency).



```

#
#           accel_alarm Configuration File
#
MyModId      MOD_ACCEL_ALARM # Module id for this instance of accel_alarm
RingName     WAVE_RING      # Shared memory ring for input/output
HeartBeatInt 45             # Heartbeat interval, in seconds,
LogSwitch    2              # 0, STDOUT 1, log and STDOUT 2, log only
Debug        1              # Unless 0 will get extra diagnostics
MaxGap       1              # Restart offset buffer after gap bigger than this (seconds)
NoDataAlarmTime 100        # If no new data for this long sound alarm (seconds)
MaxChanLatency 200         # If any channel has greater latency (sec) then sound alarm
ReadModID    MOD_SCREAM2EW  # Module to accept data from for monitoring
OffWin       100           # Window (sec) for moving average removed from each sample
NumTrig      2             # Number of stations required to trigger before alarm sounds
AlarmWindow  20            # Window (sec) for triggers at different stations
AlarmInterval 600          # Minimum time between alarms (seconds)

# SCN Selection - no wildcards allowed
# STA1: 0.267e6 counts/m/s2 = 2.61e6 counts/g
# STA2: 1.270e6 counts/m/s2 = 1.24e7 counts/g
#
#   sta  chan net loc gain(counts/ug) threshold(ug) sample rate
AlarmSCNL STA2 ENZ GB 00 12.4          5000      100
AlarmSCNL STA2 ENE GB 00 12.4          5000      100
AlarmSCNL STA2 ENN GB 00 12.4          5000      100
AlarmSCNL STA1 ENZ GB 00 2.61          5000      100
AlarmSCNL STA1 ENE GB 00 2.61          5000      100
AlarmSCNL STA1 ENN GB 00 2.61          5000      100

```

**Figure 9. Earthworm accel\_alarm configuration.**

## 4.5 switch\_relay

This module was written at the BGS. It is responsible for a ‘keep-alive’ pulse, which is monitored by a hardware watchdog, as explained in section 2.4. If an appropriate alarm message is detected, the pulse is stopped but the alarm relay will only change state if the pulse has stopped from both PCs. In the configuration shown (Fig. 10) the pulse will be stopped on receipt of an **ALARM\_ACCEL** message or an **ALARM\_NO\_DATA** AlarmType message.

An **ALARM\_ACCEL** message is sent by the *accel\_alarm* module if the acceleration threshold is exceeded.

An **ALARM\_NO\_DATA** message is sent if data from the one or more accelerometers stops. An **ALARM\_NO\_DATA** message is also sent by the software *watchdog* module if it detects that the *accel\_alarm* module has stopped.

The pulse will restart after an interval set by the `ResetInterval` configuration variable. In normal use, it is anticipated that `ResetInterval` will be set to zero. In this case, the pulse will not be restarted until the module is manually restarted. This can be done by typing “restart *nnnn*” in the Earthworm window where *nnnn* is the process ID given in the status table. As an alternative, an icon is provided on the desktop that runs a script to perform such a restart.

The way that this module works means that, it was necessary to test the whole system for an extended period. On each cycle a pulse a few ms long (10 in the example here) is sent to the hardware Watchdog and prevents it from changing state. However, the size of the variable holding the time since the module started means that a cycle will be skipped every 49 days. The code is designed to take this into account but needed to be run continuously for at least 49 days to test this properly.

```

#
#           switch_relay Configuration File
#
MyModName      MOD_SWITCH_RELAY  # Module id for this instance of scream2ew
RingName       WAVE_RING         # Shared memory ring for input/output
LogSwitch      2                 # 0 STDOUT; 1 log and STDOUT; 2 log only
Debug          0                 # If not 0 get dignostic messages
HeartBeatInt   60                # Time (sec) between heartbeats put onto ring

# RST on ComPort set to 1 for PulseLength ms every CycleLength ms
ComPort        COM1
PulseLength    10
CycleLength    200

# Start pulsing again this many seconds after an alarm
ResetInterval  0

# Alarm types to look out for and stop pulse if received
AlarmType      ALARM_ACCEL
AlarmType      ALARM_NO_DATA

```

**Figure 10. Earthworm switch\_relay configuration.**

## 4.6 email\_alarm

This module was written at the BGS. It listens for alarm type messages on the memory ring and when it receives one it sends emails to a list of recipients. With the configuration file shown (Fig. 11) it will send emails upon receipt of **ALARM\_ACCEL** or **ALARM\_NO\_DATA** AlarmType messages as explained above for *accel\_alarm*. It will also send email on receipt of an **ALARM\_MOD\_DEAD** message. This is sent by the *watchdog* module if it cannot confirm that the other Earthworm is alive. In this case, it must be assumed that redundancy has been lost, but an alarm condition has not been met on this computer.

```

#
#           email_alarm Configuration File
#
MyModName      MOD_EMAIL_ALARM  # Module id for this instance of email_alarm
RingName       WAVE_RING         # Shared memory ring for input/output
LogSwitch      1                 # 0 STDOUT; 1 log and STDOUT; 2 log only
Debug          1                 # Debug - 0 to turn off
HeartBeatInt   60                # Time in seconds between heartbeats
MailServer     localhost
Subject        "SEISMIC ALARM BGS MONITOR 1"      # Subject for email messages.
From           alarm1@bgs.ac.uk      # From line for email messages

# 0 to 8 email recipients may be specified below.
mail rrl@bgs.ac.uk
mail alarm_test@outlook.com

# Alarm types to look out for and time interval (minutes) before repeat message
AlarmType      ALARM_ACCEL        60
AlarmType      ALARM_MOD_DEAD     1440
AlarmType      ALARM_NO_DATA      1440

```

**Figure 11. Earthworm email\_alarm configuration.**

It is anticipated that the list of recipients for this module includes staff at the alarm's location, as well as the administrators at BGS. Emails will be sent from this module either if there is an alarm condition or if redundancy has been lost and immediate maintenance is required. Less urgent diagnostic emails will be sent to BGS administrators by *statmgr* as described in 4.2.

*email\_alarm* and *statmgr* are the only modules that write messages to the Earthworm window (because *LogSwitch* is set to 1 rather than 2 in their configuration files). This is done so that any message being emailed will also be shown in the Earthworm window. Once an email has been sent, it will not be repeated until the number of minutes specified in the configuration file has been reached. This is to prevent excessive emails being sent. In the example here, messages warning of **ALARM\_NO\_DATA** will be sent once a day.

## 4.7 import\_generic, export\_generic

These are standard Earthworm modules for passing messages between two Earthworm instances. In this case, they are used to transmit heartbeat messages from the shared memory on one computer to the shared memory on the other computer. These foreign heartbeats are then monitored by the *watchdog* module so that a warning email can be sent if the other instance of Earthworm fails.

Unlike those for other modules, the configuration files (Fig. 12 and Fig. 13) for these two modules are different on the two acquisition computers. *SenderIpAdr* in the *import\_generic* configuration file must be the IP address of the other computer, while *ServerIPAdr* in the *export\_generic* configuration file must be the address of this computer.

```
#
# import_generic Configuration File
#
MyModuleId      MOD_IMPORT_GENERIC  # Module id for this import,
RingName        WAVE_RING        # Shared memory ring to use for input/output,
HeartBeatInt    120                # Heartbeat in sec (>= SenderHeartRate)
LogFile         2                  # 0, STDOUT 1, log and STDOUT 2, log only
MaxMsgSize      65000             # Maximum size (bytes) for input/output msgs
MyAliveString   "alive"          # Heartbeat text to foreign server machine
MyAliveInt      100               # Seconds between alive msgs sent to server
SenderIpAdr     10.10.10.12       # IP address of the machine we're to listen to
SenderPort      16006             # IP port number used by sender
SenderHeartRate 500               # Sender's heart beat interval in seconds
SenderHeartText "alive"          # Sender's heartbeat text
```

**Figure 12. Earthworm import\_generic configuration.**

```
#
# export_generic configuration file
#
MyModuleId      MOD_EXPORT_GENERIC # module id for this program
RingName        WAVE_RING        # Shared memory ring to use for input/output
HeartBeatInt    120                # Heartbeat interval (sec) (>= RcvAliveInt)
LogFile         2                  # 0, STDOUT 1, log and STDOUT 2, log only
#Verbose        # If uncommented get VERY LARGE logfiles
ServerIPAdr     10.10.10.11       # Export messages via this ethernet card
ServerPort      16007             # Well-known port number to export msgs on
MaxMsgSize      4096             # Maximum size (bytes) for input/output msgs
RingSize        1000             # Number of msgs to buffer for export
SendAliveText   "alive"          # String sent to client as heartbeat
SendAliveInt    100               # Seconds between alive msgs sent to client.
RcvAliveText    "alive"          # Text of client's heartbeat (we get this)
RcvAliveInt     500               # Seconds between client's heartbeats to us.

# Logos of messages to export to client systems
#
GetMsgLogo      Installation      Module      Message Type
INST_BGS_MON_1  MOD_WILDCARD  TYPE_HEARTBEAT
```

**Figure 13. Earthworm export\_generic configuration.**

## 4.8 watchdog

This module was written at the BGS. It monitors heartbeat messages from other modules in a similar way to *statmgr*. Instead of directly sending an email however, it puts an alarm message onto the memory ring for other modules. This means that if a listed module dies the *switch\_relay* module can be messaged and the hardware Watchdog triggered. The interval between heartbeats is long at 600 seconds so that *statmgr* has ample opportunity to restart the module.

In the configuration shown (Fig. 14) two modules are monitored and a different response is required for the failure of each. If the *accel\_alarm* module on this computer (BGS\_MON\_1) fails then functionality has been lost and an **ALARM\_NO\_DATA** message is sent. This is listened for by *switch\_relay* and *email\_alarm* and will result in the pulse for this machine stopping and an email. If the *email\_alarm* module on the other Earthworm (BGS\_MON\_2) fails an **ALARM\_MOD\_DEAD** message is sent. A message of this type is the only listened for by *email\_alarm* and will result in an email being sent. The most likely reason for this situation is that the other PC or Earthworm has died. Redundancy has been lost and notification is required, but because the pulse from the other computer will have stopped, stopping the pulse from this computer would result in an audible alarm. As this computer is still working fine this is not the desired outcome.

```
#
#           watchdog Configuration File
#
MyModId      MOD_WATCHDOG      # Module id for this instance of watchdog
RingName     WAVE_RING         # Shared memory ring for input/output
HeartBeatInt 45                # Heartbeat interval, in seconds,
LogSwitch    2                 # 0 STDOUT; 1 log and STDOUT; 2 log only
Debug        1                 # Unless 0 will get extra diagnostics
AlarmInterval 800              # Minimum time between alarms (seconds)

# Which modules to watch and how long to give them before sending which type of alarm
# Installation name
# Module name form that installation
# Time in sec to wait for statmgr to restart module
# Type of alarm message to send
# Number of times will send the message before giving up
#
WatchMod INST_BGS_MON_1 MOD_ACCEL_ALARM 600 ALARM_NO_DATA 5 # Sound alarm if local accel_alarm dead
WatchMod INST_BGS_MON_2 MOD_EMAIL_ALARM 600 ALARM_MOD_DEAD 5 # Email if other earthworm dead
```

Figure 14. Earthworm watchdog configuration.

## 5 Other Software

### 5.1 Mailer

A command line SMPT mailer utility called BLAT is installed as part of Earthworm. When an Earthworm module needs to send an email it does so by invoking BLAT. BLAT is initialised to send mail from a localhost email server.

## 5.2 Email server

The email server hMailServer is installed on the computers to handle the emails sent via BLAT. The software is free, distributed under the GNU General Public License, and was downloaded from <https://www.hmailserver.com/>, the version installed is 5.6.3.

These computers are not part of a domain registered with the DNS (Domain Name System) and so will not be able to receive email. It is possible to send email, however, as long as the sender address for the email contains a valid domain. hMailServer is set up to act as if it is the domain bgs.ac.uk and so emails will come from **alarm1@bgs.ac.uk** and **alarm2@bgs.ac.uk**. The slight complication is that hMailServer thinks that emails addressed to bgs.ac.uk are local and so will not send emails to real BGS addresses by default. This is worked around by setting up a route for the domain bgs.ac.uk. This means that emails to this domain are sent straight to the NERC mail server. The disadvantage of this is that the normal MX server is by-passed, meaning that if, in the future, BGS were to change their email provider the server address would change and the corresponding mapping would be ignored. To remedy this emails are sent to two addresses for some BGS staff, their BGS address and an alternative address. This insures that any change in the actual address for bgs.ac.uk would be noticed and the corresponding change made in the email servers on the alarm computers. The setup of the mail server with the domain and user details is saved in c:\setup in a file with format “HMBackup yyyy-mm-dd hhmmss.7z”.

## 5.3 Perl scripts

There is a Perl script called *delete\_old.pl* that is run every day by the task scheduler. This cleans up log directories so that the acquisition computers disks do not fill up. Data files are written by Scream! and Scream! deletes older data files to stop these filling up the disk. There is also a Perl script called *restart\_switch\_relay* that allows a user to restart the *switch\_relay* module using a desktop icon rather than by typing in the Earthworm window.

## 6 Testing

These tests and their consequences are given as examples to show how the system reacts to different circumstances. The emails generated by these tests show what the emails generated in real situations will look like. In all cases, when an LED goes off it is necessary to restart the *switch\_relay* module on the corresponding PC for it to go back on again.

1) A heavy stamp near the two accelerometers (which are side by side) causes both LEDs to go off and the “alarm” LED to go off – meaning that an audible alarm would sound. Here, a stamp at 11:35 results in two emails, one from the *email\_alarm* module on each computer. The acceleration values that exceed the threshold for that channel are listed (Fig. 15 and Fig. 16).

```
From: alarm1@bgs.ac.uk
Sent: 10 December 2015 11:35
To: Lockett, Richard R.
Subject: SEISMIC ALARM BGS MONITOR 1
```

```
ALARM_ACCEL: Values: STA1.ENZ.GB.00=-5250 STA2.ENZ.GB.00=-5737 STA2.ENN.GB.00=-5224 at UTC Thu Dec 10 11:35:16 2015
```

**Figure 15. Example alarm email sent from PC1 after high acceleration (values in  $\mu\text{g}$ ).**

```
From: alarm2@bgs.ac.uk
Sent: 10 December 2015 11:35
To: Lockett, Richard R.
Subject: SEISMIC ALARM BGS MONITOR 2
```

```
ALARM_ACCEL: Values: STA1.ENZ.GB.00=-5250 STA2.ENZ.GB.00=-5737 STA2.ENN.GB.00=-5224 at UTC Thu Dec 10 11:35:16 2015
```

**Figure 16. Example alarm email sent from PC2 after high acceleration (values in  $\mu\text{g}$ ).**

2) Turning PC1 off causes LED 1 to go off but the “alarm” LED stays on. Here, power was cut to PC1 at 12:46 resulting in two emails from PC2 at 12:49 and 12:56. The first message (Fig. 17) is sent by *statmgr* to the system admin email list (probably BGS addresses). The second message (Fig. 18) is sent by *email\_alarm* to the alarm email list (everyone) as redundancy has been lost.

```
From: alarm2@bgs.ac.uk
Sent: 19 January 2016 12:49
To: Lockett, Richard R.
Subject: BGS MONITOR 2
```

```
UTC_Tue Jan 19 12:48:52 2016 BGS-MONITOR-2/export_generic No connection after 120 seconds
```

**Figure 17. Example diagnostic email sent on failure to connect with the other PC.**

```
From: alarm2@bgs.ac.uk
Sent: 19 January 2016 12:57
To: Lockett, Richard R.
Subject: SEISMIC ALARM BGS MONITOR 2
```

```
watchdog: ALARM_MOD_DEAD: Module INST_BGS_MON_1.MOD_EMAIL_ALARM dead at UTC Tue Jan 19 12:56:33 2016
```

**Figure 18. Example alarm email sent on failure to connect with the other PC.**

3) Disconnecting one accelerometer results in all three LEDs going off 3 minutes later and an audible alarm would sound. Here, the accelerometer identified as STA2 was unplugged at 13:40 and an email (Fig. 19) is sent to the alarm list.

```
From: alarm1@bgs.ac.uk
Sent: 21 January 2016 13:43
To: Lockett, Richard R.
Subject: SEISMIC ALARM BGS MONITOR 1
```

```
accel_alarm: LATENCY > 200 for: STA2.ENZ.GB.00 STA2.ENE.GB.00 STA2.ENN.GB.00 at UTC Thu Jan 21 13:42:59 2016
```

**Figure 19. Example alarm email sent following instrument failure.**

## 7 Summary

This report describes a system designed for use in nuclear power stations, or at other infrastructure where a rapid response may be required when the vibration caused by an earthquake exceeds a certain threshold. The system uses free-field acceleration measurement to trigger a physical alarm.

The monitoring system comprises of one or more windows PCs running two pieces of readily available software, Scream! and Earthworm. The BGS have developed modules for Earthworm to trigger on peaks in acceleration and to interact with a relay that can control an existing hardware alarm. An alarm is sounded if acceleration thresholds (chosen to be specific to the installation) are exceeded at more than one station or if data from one of the stations stops for any reason. The hardware watchdog switches if a regular pulse stops, meaning that the system will also give a visible indication (one LED will go out) in the case of hardware or software failure on one PC. By having two PCs, redundancy is added and the failure of a single PC will not trigger the alarm, rather the second PC will detect the failure and send emails. This report assumes that two PCs are used and describes the behaviour of the system with this configuration.

There are two Earthworm modules that send emails and they can do so to different sets of recipients. *statmgr* sends emails when any module produces an error. These are normally minor and will be sent only to administrators at BGS. *email\_alarm* sends emails when acceleration thresholds are passed or data ceases from enough accelerometers to make an acceleration trigger impossible (one in the case of only two accelerometers being used). In these cases the audible alarm will be sounding. The *email\_alarm* module will also send emails if there is a software or hardware failure on one PC only. In this case, the alarm will not be sounding but redundancy will have been lost. It is anticipated that *email\_alarm* will send emails to a list of addresses supplied by the client as well as to BGS addresses.

# Appendix I