Degum
LPM Data Processing Software
Installation and Operation Guide

Document Version
Version 1.0 – Original
Version 1.1 – Added DGHeader application
Version 1.2 – Added Despike option to Degum2
Version 1.3 – Added –p Output Path option and use of quotation marks to DGRange
Version 1.4 – Removed DGHeader, added DGMod

Author
David Judge, PSD BAS, 24/05/06

Abbreviations
1. LPM Low Power Magnetometer
2. STL Standard Template Library

Associated Documents
1. LPM Processing Software specification
2. Degum Technical description

Associated CD
Release 1.1 – 11/07/06

British Antarctic Survey

NATURAL ENVIRONMENT RESEARCH COUNCIL

Table of Contents

Overview

This document is the installation and operation guide for the suite of Degum LPM data processing applications. The suite is comprised of six applications: *Degum1*, *Degum2*, *Degum3*, *Degum4*, *DGRange* and *DGMod*.

The first four applications, *Degum1* to *Degum4*, are processing applications which reformat raw LPM data files into a scientifically usable form. *DGRange and DGMod* are two non-processing utility applications. *DGRange* can be used to export whole-day portions of data from a *Degum* file to a separate file and *DGMod* can be used to modify the header or data of a Degum file

The applications have been designed such that they can be compiled and run on minimal specification hardware. To this end, the source code is written in ANSI compliant C++, utilising STL support and the built applications are console based with minimal interface.

Being command line driven, the applications can be easily incorporated into scripts and may be used for batch processing.

System Requirements

The installation system can be of any architecture and operating system, such as *unix*, *Linux or MS Windows PC,* but must have, pre-loaded, an ANSI compliant C++ compiler with STL support. To assist in the compilation process, the system should ideally have *make* and *Makefile* support.

For the purposes of this document, it is assumed that the installation system is a Linux based PC with the standard gnu C++ (g++) compiler and make utility loaded.

Users wishing to use a different environment, such as the Visual C++ compiler on a MS Windows PC, will need to :

- Build a separate console based project for each Degum application
- Import the corresponding code files into the project
  (see Appendix A for a list of each application's source code file requirements)
- Compile and link the project using the standard IDE tools

Installation

File Extraction

All the Degum installation files must be extracted from the supplied media, be it CD, DVD, compressed file etc., and stored in their own folder on the installation HDD. The directory structure of the supplied files must be maintained in the installation folder. Once copied, and uncompressed if necessary, the installed files should appear as below:

```
-rw-rw-rw-  1 user  users 543 2006-04-19 12:05 degum.ini
drwxr-xr-x  3 user users 160 2006-04-03 17:41 doc
-rw-rw-rw-  1 user users 783 2006-04-05 14:58 Makefile
drwxr-xr-x  8 user users 192 2006-04-03 17:46 src
```

The *src* folder contains all the source files in 7 sub-folders:

```
drwxr-xr-x  2 user users 912 2006-04-26 14:28 common
drwxr-xr-x  2 user users 976 2006-04-19 11:57 degum1
drwxr-xr-x  2 user users  80 2006-04-19 11:58 degum2
drwxr-xr-x  2 user users  80 2006-04-19 11:58 degum3
drwxr-xr-x  2 user users  80 2006-04-19 11:59 degum4
drwxr-xr-x  2 user users  80 2006-04-19 11:59 dgmod
drwxr-xr-x  2 user users  80 2006-04-19 11:59 dgrange
```

The files contained within each sub-folder are:

common :

CCircleBuffer.cpp	CCircleBuffer.h
CDG1Header.cpp	CDG1Header.h
CDG1File.cpp	CDG1File.h
CDG2File.cpp	CDG2File.h
CDG4File.cpp	CDG4File.h
CDGMFile.cpp	CDGMFile.h
CErrOut.cpp	CErrOut.h
CFile.cpp	CFile.h
CINIFile.cpp	CINIFile.h
CSample.cpp	CSample.h
CString.cpp	CString.h
CUTTime.cpp	CUTTime.h
globals.cpp	globals.h
	LPMDefs.h

degum1 :

CDataEvent.cpp	CDataEvent.h
CDataEventV2.cpp	CDataEventV2.h
CDataEventV3.cpp	CDataEventV3.h
CDBData.cpp	CDBData.h
CEvent.cpp	CEvent.h
CEventManager.cpp	CEventManager.h
CGPSEvent.cpp	CGPSEvent.h
CGPSInterval.cpp	CGPSInterval.h
CHKEvent.cpp	CHKEvent.h
CLineBuffer.cpp	CLineBuffer.h
CLPMFile.cpp	CLPMFile.h
CLPMFileV2.cpp	CLPMFileV2.h
CLPMFileV3.cpp	CLPMFileV3.h
CQCEvent.cpp	CQCEvent.h
degum1.cpp	

degum2 : degum2.cpp

degum3 : degum3.cpp

degum4 : degum4.cpp

dgrange : dgrange.cpp

dgmod : dgmod.cpp

Application Build

To build the applications, open a terminal window and navigate to the installation folder then type *make* at the command prompt to invoke the make utility. *Make* will automatically read the *Makefile* and execute the commands required to build the Degum applications.

A typical example of the build output would be:

```
g++    -c -o src/common/CCircleBuffer.o src/common/CCircleBuffer.cpp
g++    -c -o src/common/CDG1Header.o src/common/CDG1Header.cpp
g++    -c -o src/common/CDG4File.o src/common/CDG4File.cpp
g++    -c -o src/common/CErrOut.o src/common/CErrOut.cpp
g++    -c -o src/common/CINIFile.o src/common/CINIFile.cpp
g++    -c -o src/common/CString.o src/common/CString.cpp
g++    -c -o src/common/globals.o src/common/globals.cpp
g++    -c -o src/common/CDG1File.o src/common/CDG1File.cpp
g++    -c -o src/common/CDG2File.o src/common/CDG2File.cpp
g++    -c -o src/common/CDGMFile.o src/common/CDGMFile.cpp
g++    -c -o src/common/CFile.o src/common/CFile.cpp
g++    -c -o src/common/CSample.o src/common/CSample.cpp
g++    -c -o src/common/CUTTime.o src/common/CUTTime.cpp
g++ -o Degum1 src/common/CCircleBuffer.o src/common/CDG1Header.o
src/common/CDG4File.o src/common/CErrOut.o src/common/CINIFile.o
src/common/CString.o src/common/globals.o src/common/CDG1File.o
src/common/CDG2File.o src/common/CDGMFile.o src/common/CFile.o
src/common/CSample.o src/common/CUTTime.o src/degum1/*.cpp -Isrc/common
g++ -o Degum2 src/common/CCircleBuffer.o src/common/CDG1Header.o
src/common/CDG4File.o src/common/CErrOut.o src/common/CINIFile.o
src/common/CString.o src/common/globals.o src/common/CDG1File.o
src/common/CDG2File.o src/common/CDGMFile.o src/common/CFile.o
src/common/CSample.o src/common/CUTTime.o src/degum2/*.cpp -Isrc/common
g++ -o Degum3 src/common/CCircleBuffer.o src/common/CDG1Header.o
src/common/CDG4File.o src/common/CErrOut.o src/common/CINIFile.o
src/common/CString.o src/common/globals.o src/common/CDG1File.o
src/common/CDG2File.o src/common/CDGMFile.o src/common/CFile.o
src/common/CSample.o src/common/CUTTime.o src/degum3/*.cpp -Isrc/common
g++ -o Degum4 src/common/CCircleBuffer.o src/common/CDG1Header.o
src/common/CDG4File.o src/common/CErrOut.o src/common/CINIFile.o
src/common/CString.o src/common/globals.o src/common/CDG1File.o
src/common/CDG2File.o src/common/CDGMFile.o src/common/CFile.o
src/common/CSample.o src/common/CUTTime.o src/degum4/*.cpp -Isrc/common
g++ -o DGRange src/common/CCircleBuffer.o src/common/CDG1Header.o
src/common/CDG4File.o src/common/CErrOut.o src/common/CINIFile.o
src/common/CString.o src/common/globals.o src/common/CDG1File.o
src/common/CDG2File.o src/common/CDGMFile.o src/common/CFile.o
src/common/CSample.o src/common/CUTTime.o src/dgrange/*.cpp -Isrc/common
g++ -o DGMod src/common/CCircleBuffer.o src/common/CDG1Header.o
src/common/CDG4File.o src/common/CErrOut.o src/common/CINIFile.o
src/common/CString.o src/common/globals.o src/common/CDG1File.o
src/common/CDG2File.o src/common/CDGMFile.o src/common/CFile.o
src/common/CSample.o src/common/CUTTime.o src/dgmod/*.cpp -Isrc/common
```

Once complete, the installation folder should contain the six Degum applications:

```
-rwxr-xr-x  1 user users 291314 2006-04-28 14:23 Degum1
-rwxr-xr-x  1 user users 196713 2006-04-28 14:23 Degum2
-rwxr-xr-x  1 user users 196713 2006-04-28 14:23 Degum3
-rwxr-xr-x  1 user users 192625 2006-04-28 14:23 Degum4
-rw-rw-rw-  1 user users    547 2006-04-19 12:05 degum.ini
-rwxr-xr-x  1 user users 192619 2006-04-28 14:23 DGMod
-rwxr-xr-x  1 user users 196757 2006-04-28 14:23 DGRange
drwxr-xr-x  3 user users    160 2006-04-03 17:41 doc
-rw-rw-rw-  1 user users    871 2006-04-05 14:58 Makefile
drwxr-xr-x  8 user users    192 2006-04-03 17:46 src
```

Post Build Location

The applications are independent executables and can now be moved, if required, to a more convenient location for use. The source files do not need to be moved.

Operation

The applications are designed to be run from the command line or from within a batch script. Run-time parameters can be passed on the command line or, more commonly, via an INI file. A complete description of the Degum application's parameter options is provided in the associated documents.

INI File

An example INI file, degum.ini, is included with the installation files, it contains parameters for the four processing applications, Degum1 – Degum4:

```
;INI file for Degum, LPM processing software

[General]
ShowHistory = True
TXTDelimiter = ,
CommentDelimiter = ;
DAC2nT1 = 128.0
DAC2nT2 = 70000.0
ADC2nT = 0.77213

[Degum1]
MaxLineLength = 256
NumSatMin = 1
Intervals = 1, 3, 5, 9, 10, 15, 60
FixingLoopsMin = 10
FixingLoopsMax = 600
NormalLoopsMin = 240
NormalLoopsMax = 21300
SwapBytes = False
RateAverage = 10
GPSQC = False
BlockQC = False
TimingQC = False

[Degum2]
ADCZero = 18759
;Despike =
ADCQC = False
ADCQCMin = -1
ADCQCMax = 1
ADCQCRange = 5

[Degum3]
WindowLength = 86400
Moveup = 43200
```

A single INI file can be used for all Degum applications. The provided example should suffice for most situations but It may be necessary to modify parameters such as *Degum1 Intervals*, *Degum3 WindowLength* and *Moveup* and to enable the QC options in *Degum2*, in some situations.

Processing

To process a raw LPM data file it is normal to run the first three processing applications in turn from Degum1 to Degum3 - the output file of one application being the input file of the next. The output files will be named by the application using following the convention:

> Output File Name = Input File Name + ".dg#"
> where # is the number of the Degum application

For example, if a Degum1 input file was called M78-337-2004 then the output would be called M78-337-2004.dg1

There is no data file output from Degum3, only a rotation angle that must be supplied as a parameter to Degum4 hence Degum4 must always run separately from the first three applications.

The processing applications, especially Degum1, can write a considerable amount of processing information to the console during run time. This information may be required at a later date and should be retained so it is advisable to redirect each application's output to a separate file.

The usage for Degum1 – 4, ignoring the processing parameter options (which will be addressed later), are:

```
Degum1 [INIfile] Datafile

Degum2 [INIfile] DG1File

Degum3 [INIfile] DG2File

Degum4 [INIfile] Rotation(Deg) DG2File
```

Thus, with a raw LPM file, M84-336-2002, using an INI file and redirecting the run time output to a separate text file, an example processing run would be:

```
Degum1 degum.ini M84-336-2002 > M84-336-2002.dg1.txt
Degum2 degum.ini M84-336-2002.dg1 > M84-336-2002.dg2.txt
Degum3 degum.ini M84-336-2002.dg2 > M84-336-2002.dg3.txt

Degum4 degum.ini 22.34 M84-336-2002.dg2 > M84-336-2002.dg4.txt
```

It is common to run the applications for several raw data files from within a batch script:

```
Degum1 degum.ini M84-336-2002 > M84-336-2002.dg1.txt
Degum1 degum.ini M70-039-2004 > M70-039-2004.dg1.txt
Degum2 degum.ini M84-336-2002.dg1 > M84-336-2002.dg2.txt
Degum2 degum.ini M70-039-2004.dg1 > M70-039-2004.dg2.txt
Degum3 degum.ini M84-336-2002.dg2 > M84-336-2002.dg3.txt
Degum3 degum.ini M70-039-2004.dg2 > M70-039-2004.dg3.txt
```

To be followed by the Degum4 script once the required rotation angles have been recovered from the Degum3 run time output files:

```
Degum4 degum.ini  22.34 M84-336-2002.dg2 > M84-336-2002.dg4.txt
Degum4 degum.ini -16.88 M70-039-2004.dg2 > M70-039-2004.dg4.txt
```

Intermediate Files

To save file storage space, the intermediate data files, those ending .dg1 or .dg2, may be deleted once the Degum4 file (.dg4) has been created.

Command Line Processing Parameters

Additional command line parameters may be used when running the Degum1-3 applications to override a limited subset of the INI file supplied processing parameters. This is usually done to test the effects of various parameter changes for individual files. A brief description is given here but for a full explanation please refer to the associated documents.

The following section gives the full usage for each application showing the available processing parameter options, a table explaining those options and an example:

Degum1

Usage: `Degum1 [INIfile] [-s#][-a#][-g][-b][-t] Datafile`

Parameter	Equivalent INI Parameter	Description
-s#	NumSatMin	Sets the minimum number of tracked satellites required for a GPS fix to be considered good
-a#	RateAverage	Sets the number of previous clock rates to average to determine the current local clock rate
-g	GPSQC	Use to enable the GPS QC option
-b	BlockQC	Use to enable the Block QC option
-t	TimingQC	Use to enable the Timing QC option

Example: `Degum1 degum.ini -s5 -a20 -g -b -t M84-336-2002`

Degum2

Usage: `Degum2 [INIfile] [-z # #...][-d # #...][-q min max range] DG1File`

Parameter	Equivalent INI Parameter	Description
-z # #...	ADCZero	Sets the sample indices to be tested for ADC Zero fix
-d # #...	Despike	Sets the sample indices to be interpolated for despike fix
-q min max range	ADCQC ADCQCMin ADCQCMax ADCQCRange	Use to enable the ADCQC option using the supplied *min*, *max* and *range* parameters

Example: `Degum2 degum.ini -z 18759 20120 -q -5 5 10 M84-336-2002.dg1`

Degum3

Usage: `Degum3 [INIfile] [-w length moveup] DG2File`

Parameter	Equivalent INI Parameter	Description
-w length moveup	WindowLength Moveup	Sets, in seconds, the *length* and *moveup* of the quiet time scan window

Example: `Degum3 degum.ini -w 3600 3600 M84-336-2002.dg2`

DGRange

DGRange is a non-processing utility application that allows the user to extract whole-day portions of data from a Degum file and export it to a new file. Degum1, Degum2 or Degum4 data may be used and the output can be in binary or text format. It is purely command line driven – no INI file is used.

For text output files, the field delimiting character can be specified and the export of the file processing history is optional.

The utility can also be used to just report the time range and/or processing history of a supplied Degum file.

Please refer to the associated documents, especially the Degum Technical Description, for a full description of the DGRange application.

Command Line Parameters

DGRange usage is:

```
DGRange [-t['ch']][-h['ch']][-s][-pOutput Path][-rDay Month Year Day Month Year]] DGFile
```

where *DGFile* is the name of the input file.

The options are:

`-t['ch']` Export in text format.
The number and content of fields will vary with the type of input file (see below)
The optional *ch* character will be used as the field delimiter. Default is ' ' (space)

`-h['ch']` Export processing history information.
This option only applies if exporting in text format or reporting the input file range
The optional *ch* character will be used as the comment character. Default is ' ' (space)

`-s` Split a multi-day export range into separate output files.

`-pOutput Path`
Supply alternative path for the output file. Output files will be placed in the input directory if this option is not supplied. Do not use the '~' character for /home paths.

`-rDay Month Year Day Month Year`
The range of data to export. If not specified, the whole input file will be exported.
First *Day*, *Month* and *Year* (DMY) specify the start date, the second triplet specify the end date, both inclusive
Only whole days can be exported. Duplicate DMY triplets will export a single day's data
DMY parameters are numerical and integers. Years are 4 digit.
All 6 DMY values must be specified

A command with no parameters, except the input file name, will report the range of the input file only.

If the `-t` option is used without the `-r` option then the whole input file will be exported as a text file.

For some parameters, such as single characters or path names containing spaces, it may be necessary to wrap the string or character in quotation characters. Single or double quotes should be used – depending on operating system – but in general, use single quotes for linux/unix systems and double quotes for Windows/DOS systems.

Examples

`DGRange -h M66-294.dg1`	Reports the range of file *M66-294.dg1* and displays its processing history
`DGRange -t -r14 3 2003 15 3 2003 M84-336.dg4`	Exports data from 14/3/03 to 15/3/03 of *M84-336.dg4* file in text format with space delimiter and no processing history
`DGRange -r2 8 2005 11 8 2005 M83-348.dg3`	Exports data from 2/8/05 to 11/8/05 of *M83-348.dg3* file in binary format
`DGRange -t, -h; -r1 6 2002 1 6 2002 M78-337.dg4`	Exports a comma delimited text file with colon commented processing history for data from the single day 1/6/02 of file *M78-337.dg4*
`DGRange -t -s -r1 6 2002 3 6 2002 M78-337.dg4`	Exports, to separate text files, data from each day between 1/6/02 and 3/6/02 of file *M78-337.dg4*
`DGRange -t –p/home/lpm -r1 6 2002 3 6 2002 M78-337.dg4`	Exports data from 1/6/2002 to 3/6/2002 of *M78-337.dg4* in text format. The output is directed to the folder: */home/lpm*
`DGRange -t –p”C:\lpm data” -r1 6 2002 3 6 2002 M78-337.dg4`	Exports data from 1/6/2002 to 3/6/2002 of *M78-337.dg4* in text format. The output is directed to the folder: *C:\lpm data*

DGMod

DGMod is a non-processing utility application that allows the user to view or modify any Degum file's header (processing history) or replace the data of a Degum4 file. If the input file is modified, a new binary file will be created. The status byte, bit 5, is set for all replaced data samples.

DGMod is purely command line driven – no INI file is used.

Command Line Parameters

DGMod usage is:

```
DGMod [-h TXTFile] [-d Component D M YYYY D M YYYY Value] DGFile
```

where *DGFile* is the name of the input degum file.

The options are:

`-h TXTFile` Append the text information in file *TXTFile* to the input file's header.

-d Component D M YYYY D M YYYY Value
Replaces the data values of a Degum4 file.
Component specifies which components to replace where 1 = X, 2 = Y and 4 = Z. These can be combined to replace multiple components simultaneously.
D M YYYY D M YYYY are the start and end dates to apply the modification.
Value is the replacement value to use

A command with no parameters, except the input file name, will report the header of the input file only.

Examples

`DGMod M66-294.dg1`
Displays the processing history of file *M66-294.dg1*

`DGMod –h info.txt M84-336.dg4`
Appends the text contained within the file *info.txt* to the header of file *M84-336.dg4*. A separate output file is created

`DGMod –d 5 28 12 2004 31 1 2005 99999.9 M84-336.dg4`
Replaces the X and Z components of the data within file *M84-336.dg4* with the value 99999.9 over the date range 28/12/2004 to 31/01/2005.
A separate output file is created

Appendix A

Source file list by application:

Application	Source Sub-folder	Files Used	
All	common	CDGMFile.cpp CErrOut.cpp CFile.cpp CINIFile.cpp CSample.cpp CString.cpp CUTTime.cpp globals.cpp	CDGMFile.h CErrOut.h CFile.h CINIFile.h CSample.h CString.h CUTTime.h globals.h LPMDefs.h
Degum1	common	CCircleBuffer.cpp CDG1Header.cpp	CCircleBuffer.h CDG1Header.h
	degum1	CDataEvent.cpp CDataEventV2.cpp CDataEventV3.cpp CDBData.cpp CEvent.cpp CEventManager.cpp CGPSEvent.cpp CGPSInterval.cpp CHKEvent.cpp CLineBuffer.cpp CLPMFile.cpp CLPMFileV2.cpp CLPMFileV3.cpp CQCEvent.cpp degum1.cpp	CDataEvent.h CDataEventV2.h CDataEventV3.h CDBData.h CEvent.h CEventManager.h CGPSEvent.h CGPSInterval.h CHKEvent.h CLineBuffer.h CLPMFile.h CLPMFileV2.h CLPMFileV3.h CQCEvent.h
Degum2	common	CDG1File.cpp CDG1Header.cpp	CDG1File.h CDG1Header.h
	degum2	degum2.cpp	
Degum3	common	CDG2File.cpp	CDG2File.h
	degum3	degum3.cpp	
Degum4	common	CDG2File.cpp	CDG2File.h
	degum4	degum4.cpp	
DGRange	common	CDG1File.cpp CDG1Header.cpp CDG2File.cpp CDG4File.cpp	CDG1File.h CDG1Header.h CDG2File.h CDG4File.h
	dgrange	dgrange.cpp	
DGMod	dgmod	dgmod.cpp	