GL800 GBD File Specification Sheet

1. Applicable Range

Product name:	GL800
Firmware:	Version 1.00 to
File format	Measurement data files with the ".GBD" file extension
Compatible data	Analog measurement data, logic data, pulse data, alarm data

This specification sheet is applicable to measurement data files in binary format with the ".GBD" file extension.

Revision Date	February 5, 2007

Format Outline

A “Binary Data File (.GBD file extension)” is divided into two regions: the “Header region” where the measurement data capture conditions and so forth are written, and the “Data region” where the measurement data is recorded. To enable the measurement values to be obtained from a binary data file using a PC or other device, not only the measurement data values themselves, but also measurement conditions such as the “measurement range”, “data storage sequence”, “number of data points” and other necessary measurement conditions must be read from the Header region.

Header region	The capacity varies according to conditions such as the number of channels. 4096 to（multiples of 2048） bytes ASCII data
Unused region	Filled with blank space characters（0x20）
Data region	（Number of data channels × 2 ×the number of channels）bytes 16-bit binary data Note: Pulse data is 16-bit x 2 data

1.1. Header region

The data capture conditions and so forth for the measurement data are written as ASCII text (CR+LF for the line feed character). The size of the Header region varies according to the number of channels on which data was captured and the data capture conditions. However, adjustment is made in multiples of 2048 bytes so that the head of the Data region that follows the Header region matches the sector head position no matter what recording device was used.

The number of bytes in the Header region is written to the Header region. Moreover, the unused section from the end of the Header region (up to “$EndHeader<CR><LF>) to the start position of the Data region is filled up with blank space characters (ASCII code 0x20).

1.2. Data region

The measurement data is written using 16-bit signed binary integers. The start position is the position written into the Header region (it must be a multiple of 2048 bytes), and the end position is the end of the file. The number of bytes in the Data region varies according to the number of measurement data points (the number of data points for each channel), the number of channels used for data capture, and whether or not logic data or pulse data is present. The following method can be used for calculation.

Name	Description	Value
N	Number of measurement data channels	The number of channels on which data is captured
L	Logic data	L = 1 when Logic is enabled; L = 0 when Logic is disabled
P	Pulse data	When Pulse = enabled, P = 1 to 4 (the total number of ON channels); when Pulse = disabled or all the channels are OFF, P = 0
A	Analog alarm data	Alarm = Must be enabled A = (Actual number of installed channels +15)/16
ALP	Logic/Pulse alarm data	If Logic or Pulse is enabled, ALP = 1 If disabled, ALP = 0
〔Number of bytes in the Data region〕=(N + L＋P×2＋A + ALP)× 2 ×〔number of captured data points〕		

1.2.1. Measurement value data

Each measurement value is a 2-byte signed integer (signed short), but unlike the format used with Windows or DOS PCs, the high-order byte comes first in the sequence, followed by the low-order byte (Big Endian). Accordingly, when using a PC to load measurement values from a file, an operation to reverse the sequence of the high-order byte and the low-order byte in order to change the values for storage is required.

Even-numbered address	Odd-numbered address
High-order byte data	Low-order byte data

1.2.2. Converting the binary data in the Data region to voltage values

The measurement values written to the Data region are relative voltage values with a measurement range of +20000 of full scale. Use the following calculation methods to convert these values to actual voltage values.

1. Conversion to Voltage Values

Voltage range	Example of actual voltage ranges	Calculation used for conversion
When the range has a ‘1’base	10mV/100mV/1V/10V/100V, etc.	Calculate the measurement value ÷2
When the range has a ‘2’base	20mV/200mV/2V/20V/200V, etc.	Calculate the measurement value ÷1
When the range has a ‘4’base	40mV/400mV/4V/40V/400V, etc.	Calculate the measurement value ÷5
When the range has a ‘5’base	50mV/500mV/5V/50V/500V, etc.	Calculate the measurement value ÷4

2. Adjustment of the Decimal Point Position

Voltage range	When the voltage unit is 'V'	When the voltage unit is 'mV'
10mV/20mV	Calculate the calculated results of 1. ÷ 1000000	Calculate the calculated results of 1. ÷ 1000
50mV/100mV/200mV	Calculate the calculated results of 1. ÷ 100000	Calculate the calculated results of 1. ÷ 100
500mV/1V/2V	Calculate the calculated results of 1. ÷ 10000	Calculate the calculated results of 1. ÷ 10
5V/10V/20V	Calculate the calculated results of 1. ÷ 1000	Calculate the calculated results of 1. ÷ 1
50V/100V/200V	Calculate the calculated results of 1. ÷ 100	Calculate the calculated results of 1. × 10
500V/1000V	Calculate the calculated results of 1.÷ 10	Calculate the calculated results of 1. × 100

〔Calculation example〕

Input range: 5V

Measurement data file value: +12528

Voltage value = +12528÷4÷1000 = 3.132 [V]

Measurement data file value: –9654

Voltage value = –9654÷4÷1000 = –2.414 [V]

Handling voltage data overflow

Inputs that exceed 110% of FS of the measurement range are converted to the following data values.

Input that exceeds 110% of +FS	+32765 = $(+7FFD)_{16}$
Input that exceeds 110% of -FS	–32767 = $(–7FFF)_{16}$

1.2.3. Converting temperature data

When the measurement values are temperature data, perform the following conversion to calculate the Celsius temperature[℃].

Conversion of temperature values

When the measurement values are temperature data, a value that is 10 times the temperature value (the unit is ℃) is written as a 16-bit signed integer in the same way as for voltage data (the byte sequence is the same sequence where the high-order byte is followed by the low-order byte). Accordingly, with respect to the temperature data written in the file, the temperature can be calculated as follows.

[Temperature]=[Temperature data]÷10 [℃]

In the case of temperature data, regardless of the type of thermocouple used, the relationship between the temperature data written in the file and the actual temperature is constant.

1.2.4. Converting humidity data

In the case of humidity data, the scaling function is used to perform unit conversion with respect to the voltage data in 1.2.3 above.

When the voltage range is specified as 1 V, the scale is converted as follows: 0V→0% and 1V→100%.

Therefore, when performing humidity conversion of binary data, refer to 1.2.3 above. After converting the value to a 1V value, use the scaling function to perform unit conversion.

Example: 500mV → 50%、 82mV → 8.2%

1.2.5. Logic data

The logic data correspondence is as follows. Logic data is also written to the file in the sequence of high-order byte followed by low-order byte when looking at the data corresponding to the bits for the 16-bit data.

Bit Position	Number
Bit 0	1
Bit 1	2
Bit 2	3
Bit 3	4
Bits 4 to15	Not used

1.2.6. Pulse data

Pulse data is 32-bit data comprised of 16-bit data x 2. Data is written to the file in the sequence of high-order 16-bit data followed by low-order 16-bit data.

1.2.7. Alarm data for analog channels

The alarm data correspondence is as follows. Alarm data is also written to the file in the sequence of high-order byte followed by low-order byte when looking at the data corresponding to the bits for the 16-bit data.

The alarm data is expressed as a bit corresponding to each analog channel.
Accordingly, the number of bytes also varies according to the number of input terminal units that are actually installed.
For example, if the number of channels installed is 20, since 20 bits are required, the 16-bit Big Endian format is used as the data unit and 4 bytes (= 32 bits) are transferred as alarm data.

Offset	Bit	Description (When 0, an alarm has not been generated; when 1, an alarm has been generated)
0	0	CH1 alarm data
	1	CH2 alarm data
:	:	: (omitted)
	14	CH15 alarm data
	15	CH16 alarm data
+2	0	CH17 alarm data
	1	CH18 alarm data
	2	CH19 alarm data
	3	CH20 alarm data
	4-15	Not used

1.2.8. Logic/Pulse alarm data

The alarm data correspondence is as follows. Alarm data is also written to the file in the sequence of high-order byte followed by low-order byte when looking at the data corresponding to the bits for the 16-bit data.

Bit	Description (When 0, an alarm has not been generated; when 1, an alarm has been generated)
0	Pulse 1 alarm data
1	Pulse 2 alarm data
2	Pulse 3 alarm data
3	Pulse 4 alarm data
4	Logic 1 alarm data
5	Logic 2 alarm data
6	Logic 3 alarm data
7	Logic 4 alarm data
8-15	(Not used)

2. Details on the Header Region

2.1. Characters and codes used

The header is written in ASCII text with a line feed (CR+LF) used as the delimiter. The basic configuration is as follows.

$<section name><CR><LF>
$$<section name><CR><LF>
$$$<section name><CR><LF>
<Setting name> = <Setting value>,<Setting value>,…<CR><LF>

2.1.1. Line configuration

The end of the line is a line feed (CR LF). Blank space characters (0x20) and tab characters (0x09) are ignored unless they are enclosed in double quotation marks within the “text string”. Only alphanumeric characters (a differentiation is made between uppercase and lowercase text) and symbols (+, -, _, %, $, :) can be used for keywords such as the setting name and setting value. Lines starting with the “#” symbol are considered to be comment lines and are ignored. Moreover, blank lines are also ignored. Settings that are saved as text strings are enclosed in double quotation marks.

2.2. Sections

The Header region is divided into “sections” according to the type of function. A section may be further divided into subsections, and again into sub-subsections. Sections must start with the “$” symbol. Subsections start with the “$$” symbols, and sub-subsections with the “$$$” symbols. The Header region sections are shown in the following table. The setting values in the $Common section and in the $Amp section are used for reference when measurement data is converted to actual voltage values.

Section	2nd level	3rd level	Description
$Common			General information such as the model name and version
	$$Data		Setting values for the captured data, etc.
	$$Time		Date and time information for the captured data
$Comment			Comments added when the data was saved
$Amp			Range settings, etc. for the captured data
$LogiPul			Selection of Logic or Pulse
$Pulse			Pulse settings
$Logic			Logic settings
$Measure	$$Span		Span settings
	$$Scale		Scaling（EU）settings
		$$$Pulse	Pulse scaling settings
$Alarm			Alarm settings
	$$Pulse		Alarm pulse settings
	$$Logic		Alarm logic settings
$Trigger	$$Start		General trigger start condition settings
	$$Stop		General trigger stop condition settings
$Annotation			Annotation settings
$EndHeader			End of Header

2.3. $Common section

Setting example

```
$Common
   ID            = 3E0512D6
   Volume        = 1, 1
   HeaderSiz = 10240
   Vendor        = "GRAPHTEC Corporation"
   Model         = "GL800"
   Suffix        = "           "
   User          = "GRAPHTEC"
   Stat          = "GRAPHTEC"
   UserSel       = 1
   CH             = 20CH
   Option        = None
   Format        = "Ver1.00"
   Hardware      = "Ver1.00"
   Firmware      = "Ver0.81    "
   OS             = "Ver3.02", "Ver1.00"
$$Data
   Format        = BinaryData
   Type          = BigEndian, Short, Setup
   Order         = CH1    , CH2    , CH3    , CH4    , CH5    , CH6    , CH7    , CH8    , CH9    , CH10 ,
CH11 , CH12 , CH13 , CH14 , CH15 , CH16 , CH17 , CH18 , CH19 , CH20 , Pulse1, Pulse2,
Pulse3, Pulse4, Alarm1 , Alarm2 , AlarmLP
   Sample        = 200ms
   LogicCH       = 4
   Counts        =              30
   Trigger       =              0
   Stat          = Off
$$Time
   Start         = 2006-11-28,10:38:22
   Stop          = 2006-11-28,10:38:29
   Trigger       = 2006-11-28,10:38:23
```

2.3.1. General settings

ID

Form	ID = <8-digit hexadecimal number>
Setting example	ID = 3E0512D6
Function	ID number generated from the date and time the file was created
Explanation	In the current version, the ID is only written to this line. It is not used.

Volume

Form	Volume = <Volume number>, <General number of Volumes>
Setting example	Volume = 1, 1
Function	Consecutive numbers for the devices used to configure the data files and the number of devices
Explanation	Since the capacity of the data file is smaller than the capacity of the device, “Volume = 1, 1 is always written to this line.

HeaderSiz

Form	HeaderSiz = <Number of byes in the Header>
Setting example	HeaderSiz = 10240
Function	Number of bytes in the Header region（the start point of the measurement data）
Explanation	The number of bytes in the Header region is written as a decimal number. So that the start point of the measurement data is the same as the head of the sector regardless of the recording device, the header size is rounded up to a multiple of 2048. The number of bytes in the Header region varies according to the number of channels used to capture data and the settings that were made. When actually browsing the measurement data, read the data in this field to detect the start point of the measurement data.

Vendor

Form	Vendor = "identifying text string in the Header"
Setting example	Vendor = "GRAPHTEC Corporation"
Function	This is a text string that indicates the manufacturer of the device.
Explanation	Fixed to “GRAPHTEC Corporation”

Model

Form	Model = "product model"
Setting example	Model = "GL800"
Function	This is a text string that indicates the model name of the device.
Explanation	Fixed to “GL800”

Suffix

Form	Suffix = "text string"
Setting example	Suffix = " "
Function	When the device is not the standard model, this text string may be specified to identify the device.
Explanation	Six blank space characters are always written to this line for the standard model.

User

Form	User = <text string>
Setting example	User = “GRAPHTEC”
Function	This text string is the name of the user who stored the data.

Stat

Form	Stat = <text string>
Setting example	Stat = “GRAPHTEC”
Function	This text string is the department where the data was stored.

UserSel

Form	UserSel = <0／1／2>
Setting example	UserSel = 1
Function	Numbers assigned to the users who stored the data.

CH

Form	CH = <number of channels installed in the device>
Setting example	CH = 20CH
Function	Indicates the number of channels that are installed in the GL800 device
Explanation	The number of channels that are actually installed in the device used to create the measurement data file is written to this line. Depending on the number of input terminal units mounted in the GL800, the number of channels ranges from 20 to 200. Since the amp settings up to the maximum number of channels that are written to this line are written to the $Amp section, the setting value in this line will always match the number of lines in the $Amp section.

Option

Form	Option = <option>,…,<option>
Setting example	Option = None
Function	No options are currently installed
Explanation	

2.3.2. $$Data subsection

The settings that are related to the captured data as a whole are written to the $$Data subsection.

Format

Form	Format = BinaryData
Setting example	Format = BinaryData
Function	Indicates the format used for the measurement data
Explanation	The format of measurement data files with the ".GBD" file extension is always "BinaryData".

Type

Form	Type = BigEndian, Short, Setup
Setting example	Type = BigEndian, Short, Setup
Function	Indicates the data type used for the measurement data
Explanation	The setting given above is always written for measurement data files with the ".GBD" file extension.

Order ※

Form	Order = <1st Data>,<2nd Data>,…,<Last Data>
Setting example	Order = CH1, CH2, CH8, CH9, Logic, Pulse1, Alarm1
Function	Indicates the order of the measurement data
Explanation	The order of the binary measurement data is written to this line. Since the order of the measurement data actually written to the data file varies according to the capture time settings and the input terminal unit configuration, refer to this line and check the measurement data storage sequence

Notes on the storage sequence of measurement data

- The analog measurement values are stored to memory in order starting from CH1. However, the data of channels for which Input = Off when data is captured is not written to a file. Only the measurement data of channels for which input has been enabled is written to a file.
- If there is any logic data, it will be stored to memory after the analog channel data.
- If there is any pulse data, it will be stored to memory after the analog channel data.
- Logic and pulse data cannot be stored to memory at the same time.
- The alarm data in the analog channels is always included. This data is divided into16-channel units.
- If Logic or Pulse has been enabled, the logic/pulse alarm data is stored to memory. This data is stored last.

Sample

Form	Sample = <sampling interval>
Sample setting	Sample = 100ms
Function	The sampling interval used when measurement data was captured is written to this line.
Explanation	The sampling interval used when measurement data was captured is written to this line. Refer to this line if you need to check the relationship between the time and the data position. The time units used are as follows: ms→ ms s → s min → min hour → h

Counts ※

Form	Counts = <number of data points>
Setting example	Counts = 1000
Function	The number of measurement data points is written to this line.
Explanation	The number of measurement data points is written to this line.

Trigger

Form	Trigger = <trigger point>
Setting example	Trigger = 0
Function	The position of the trigger point is written to this line.
Explanation	The position of the trigger point, with the first data position in the file as 0, is written to this line.

2.3.3. $$Time subsection

The measurement start time, measurement stop time and the time that the trigger was activated are written to this subsection.

Start

Form	Start = <date>,<time>
Setting example	Start = 1999-11-10,14:15:12
Function	The start time of measurement is written to this line.
Explanation	The start time of measurement is written to this line. Normally, the time that is written is the time that the [Start] key was pressed.

Stop

Form	Stop = <date>,<time>
Setting example	Stop = 1999-11-10,14:15:18
Function	The stop time of measurement is written to this line.
Explanation	The stop time of measurement is written to this line. Normally, the time that is written is the time that the [Stop] key was pressed, or else the time when memory recording ended.

Trigger

Form	Trigger = <date>,<time>
Setting example	Trigger = 1999-11-10,14:15:12
Function	The time that the specified trigger was activated is written to this line.
Explanation	The trigger activation time will be a time between the measurement start time and the measurement stop time.

2.4. $Comment section

This section is not used with the GL800.

2.5. $Amp section

The amp type, and input, range and filter settings for each channel are written to this section.

Setting example

```
$Amp
  CH1      = M    , DC   ,     2V, Off    ,   TC_K , +0
  CH2      = M    , DC   ,     2V, Off    ,   TC_K , +0
  CH3      = M    , DC   ,     2V, Off    ,   TC_K , +0
  CH4      = M    , DC   ,     2V, Off    ,   TC_K , +0
  CH5      = M    , DC   ,   10V, Off     ,   TC_K , +0
  CH6      = M    , DC   ,   10V, Off     ,   TC_K , +0
  CH7      = M    , DC   ,   10V, Off     ,   TC_K , +0
  CH8      = M    , DC   ,   50mV, Off    ,   TC_K , +0
  CH9      = M    , DC   ,   50mV, Off    ,   TC_K , +0
  CH10     = M    , DC   ,   100mV, Off   ,   TC_K , +0
```

2.5.1. Explanation of the fields

Field 1: CH number

The channel numbers are written to this field. The settings for all the channels up to the maximum number of channels for the device are written to the $Amp section.

Field 2: Amplifier type

The amplifier type is written to this field. For the GL800, “M” is written for all the channels.

Field 3: Input settings

The input settings that were specified at the time of measurement data capture are written to this field. Data is not captured to channels for which Input = Off was selected, and so “Off” is never displayed in this field. The setting values that are displayed in this field are shown in the following table.

M-AMP		Remarks
DC	“DC”	
TEMP	“TEMP”	
RH	“RH”	

Field 4: Range settings*

The voltage range settings that were specified at the time of measurement data capture are written to this field. If “TEMP” was specified for the input setting, ignore the setting value for this field.

To convert the measurement data to actual voltage data, refer to the voltage range setting written to this field and then make the conversion.

Field 5: Filter setting

The filter setting that was specified at the time of measurement data capture is written to this field.

Field 6: Thermocouple/Resistance temperature detector type setting

This field is only added if the amplifier is an M-type amplifier. Refer to this field if the captured data is temperature data, and you need to know the type of thermocouple or resistance temperature detector used. The setting in this field is ignored for all other input types.

2.6. $LogiPul section

The Logic and Pulse enabled/disabled settings are written to the $LogiPul section.

2.7. $Pulse section

The Pulse channel settings are written to the $Pulse section.

Even if Pulse has been disabled, the settings are always written to this section.

2.8. $Logic section

The Logic channel settings are written to the $Logic section.

Even if Logic has been disabled, the settings are always written to this section.

2.9. $Measure section

The span settings and the scaling settings are written to the $Measure section.

2.9.1. $$Span section

The span settings that were specified at the time of measurement data capture are written to this section. Please note that only the channels for which measurement data exists are written to this section.

2.9.2. $$Scale section

The scaling settings that were specified at the time of measurement data capture are written to this section. Please note that only the channels for which measurement data exists are written to this section.

2.9.2.1. $$$Pulse section

The pulse channel scaling settings that were specified at the time of measurement data capture are written to this section.

2.10. $Alarm section

The alarm setting status is written to this section.

2.10.1. $$Pulse section

The pulse alarm settings are written to this section.

2.10.2. $$Logic section

The logic alarm settings are written to this section.

2.11. $Trigger section

The trigger settings (both the Start and Stop conditions) that were specified at the time of measurement data capture are written to this section.

2.12. $Annotation section

The annotation settings are written to this section.

2.13. $EndHeader

The “$EndHeader” “line” ends the Header region (the line is actually “$EndHeader r<CR><LF>”). Blank character spaces (ASCII code 0x20) are inserted in the region that is not used between the end of the $EndHeader line and the start position (a multiple of 2048) of the Data region.

3. Points to Note When Browsing the Binary Data File

When creating a data browsing program for a binary file, please note the following points.

Variable length text data

The text data used in the Header region is basically variable length data. Even within one line, there is no set rule for the position of the start of the text string in the setting field with respect to the head. To obtain the setting value, take into account the delimiter characters and the blank space characters to enable you to detect the text string.

Variable line positions

In the same way, the order of the settings in the Header region can vary depending on the setting conditions and function extensions. As a rule, write the program so that the section name and the setting name can be detected for each setting.

Changing the Header

Do not change the contents of the header. Since the position of the measurement data is determined by the size of the header, you will not be able to access the measurement data correctly if you change its position in the file. Moreover, if you change the setting details, problems may occur when replaying data on the GL800.