GL220 / GL820 GBD File Specification Sheet

1. Applicable Range

Product name:	GL220 / GL820
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	16-bit binary data See "1.2 Data region" for the number of bytes.

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
AO	Alarm port output data	Alarm port output = Must be enabled, AO = 1
〔Number of bytes in the Data region〕=（N + L＋P×2＋A + ALP + AO）× 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	100mV/1V/10V	Calculate the measurement value ÷2
When the range has a ‘2’base	20mV/200mV/2V/20V	Calculate the measurement value ÷1
When the range has a ‘5’base	50mV/500mV/5V/50V/1-5V	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’
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/1-5V	Calculate the calculated results of 1. ÷ 1000	Calculate the calculated results of 1. ÷ 1
50V	Calculate the calculated results of 1. ÷ 100	Calculate the calculated results of 1. × 10

〔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	$+32764 = (+7FFC)_{16}$
Input that exceeds 110% of -FS	$-32767 = (-7FFF)_{16}$

Revision Date	February 5, 2007

1.2.3. Converting temperature data

When the measurement values are temperature data, perform the following conversion to calculate the Celsius temperature[°C/°F].

Conversion of temperature values

When the measurement values are temperature data, a value that is 10 times the temperature value 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). If the unit of the temperature is Celsius or Fahrenheit, the value has been stored in Celsius or in Fahrenheit accordingly.

Accordingly, with respect to the temperature data written in the file, the temperature can be calculated as follows.

[Temperature]=[Temperature data]÷10 [°C]
[Temperature]=[Temperature data]÷10 [°F]

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)
	8	CH9 alarm data
	9	CH10 alarm data
	10-15	Not used
+2	0	CH11 alarm data
	1	CH12 alarm data
:	:	: (omitted)
	8	CH19 alarm data
	9	CH20 alarm data
	10-15	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)

1.2.9. Alarm port output data

The alarm port output data correspondence is as follows. Alarm port output data is also written in the file in the sequence of high-order byte followed by low-order byte.

Bit	Description (When 0, alarm port is OFF; when 1, alarm port is ON)
0	Alarm port 1 output data
1	Alarm port 2 output data
2	Alarm port 3 output data
3	Alarm port 4 output data
4-15	(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
$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
		$$$Pulse	Pulse scaling settings
	$$Calculation		Inter-CH operation settings
$Alarm			Alarm settings
	$$Pulse		Alarm pulse settings
	$$Logic		Alarm logic settings
	$$Condition		Alarm edge/level settings
$Trigger	$$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            = 26111111
   Volume        = 1, 1
   HeaderSiz =   8192
   Vendor        = "GRAPHTEC Corporation"
   Model         = "GL220"
   Suffix       = "          "
   User          = "GRAPHTEC"
   Stat          = "GRAPHTEC"
   UserSel      = 1
   CH              = 10CH
   Option        = None
   Format        = "Ver1.00"
   Hardware    = "Ver1.00"
   Firmware    = "Ver1.00    "
   OS              = "Ver4.00", "Ver3.10"
   Software    = "            "
$$Data
   Format        = BinaryData
   Type           = BigEndian, Short, Setup
   Order         = CH1   , CH2   , CH3   , CH4   , CH5   , CH6   , CH7   , CH8   , CH9   , CH10 , Alarm1 ,
AlarmOut
   Sample        = 100ms
   ExtSamp      = Off
   ExtSampFilt = Off
   TempUnit       = C
   LogicCH      = 4
   Counts        =          1000
   Trigger      =                0
   Stat          = Off
$$Time
   Start         = 2010-02-17,09:55:35
   Stop           = 2010-02-17,09:55:54
   Trigger      = 2010-02-17,09:55:35
```

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 = "GL220"
Function	This is a text string that indicates the model name of the device.
Explanation	Fixed to "GL220" and "GL820" respectively for GL220 and GL820.

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 = 10CH
Function	Indicates the number of channels that are installed in the GL220 / GL820 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. Fixed to "10CH" for GL220. In the case of GL820, it will be between 20 and 200CH in accordance with the number of terminal blocks installed to GL820. 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, AlarmOut
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.
- The alarm output data is always stored at the end.

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

ExtSamp

Form	ExtSamp = <Off/On>
Sample setting	ExtSamp = Off
Function	The external sampling ON/OFF setting at the time of measurement data capture is written to this line.
Explanation	When it is ON, the data is stored at intervals that are n-times of the interval indicated by Sample.

ExtSampFilt

Form	ExtSampFilt = <Off/On>
Sample setting	ExtSampFilt = Off
Function	The AC line filter ON/OFF setting at the time of external sampling ON is written to this line.
Explanation	If ExtSamp = Off, this setting is ignored.

TempUnit

Form	TempUnit = <C/F>
Sample setting	TempUnit = C
Function	The temperature unit setting is written to this line.
Explanation	Indicates the unit of the temperature data written in the data. C: Celsius F: Fahrenheit

LogicCH

Form	LogicCH = <Number of Logic Channel>
Sample setting	LogicCH = 4
Function	The number of logic CHs is written to this line.
Explanation	Fixed to "4" for GL220 and GL820.

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. $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
```

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 G220 / GL820, “M” is written for all the channels. CHs with no data stored are regarded as NC.

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.

Input	
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.

2.5. $LogiPul section

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

2.6. $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.

Setting example

```
$Pulse
  Pulse1      = REVO, L, On
  Pulse2      = COUNT, H, Off
  Pulse3      = INST, H, Off
  Pulse4      = OFF, H, Off
```

Field 1: Input settings

The input settings at the time of measurement data capture are written to this field. The set values set in this field are as follows:

Pulse	
OFF	“OFF”
Revolve	“REVO”
Count	“COUNT”
Instantaneous	“INST”

Field 2: Polarity settings

The pulse polarity (slope) settings are written to this field.

L: Fall

H: Rise

Field 3: Filter settings

The filter settings at the time of measurement data capture are written to this field.

2.7. $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.

Setting example

```
$Logic
  Filter   =   On, Off, Off, Off
```

Filter

Form	Filter = <L1>,<L2>,<L3>,<L4>
Setting example	Filter = On, Off, Off, Off
Function	The logic filter ON/OFF settings are written to this field.
Explanation	The settings are stored in the order of L1, L2, L3, L4 from the left.

2.8. $Measure section

The span settings, scaling settings, and inter-CH operation settings are written to the $Measure section.

2.8.1. $$Span subsection

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.

Setting example

```
$$Span
  CH1        = -10000, +10000,   -2000, +20000
  CH2        = -10000, +10000,   -2000, +20000
  CH3        = -10000, +10000,   -2000, +20000
  CH4        = -10000, +10000,   -2000, +20000
  CH5        = -10000, +10000,   -2000, +20000
  CH6        = -10000, +10000,   -2000, +20000
  CH7        = -10000, +10000,   -2000, +20000
  CH8        = -10000, +10000,   -2000, +20000
  CH9        = -10000, +10000,   -2000, +20000
  CH10       = -10000, +10000,   -2000, +20000
  Pulse1     = 0, 500000
  Pulse2     = 0, 500000
  Pulse3     = 0, 500000
  Pulse4     = 0, 500000
```

■Analog CH fields

Field 1: CH number

The CH number is written to this field.

Field 2: Span lower limit A/D value at the voltage and humidity settings

The A/D value of the span lower limit at the time of the CH voltage setting is stored in this field. Where ±20000=±FS.

Field 3: Span upper limit A/D value at the voltage and humidity settings

The A/D value of the span upper limit at the time of the CH voltage setting is stored in this field. Where ±20000=±FS.

Field 4: Span lower limit temperature A/D value at the temperature setting

The A/D value of the span lower limit at the time of the CH temperature setting is stored in this field. Where 1°C (°F)=10.

Field 5: Span upper limit A/D value at the temperature setting

The A/D value of the span upper limit at the time of the CH temperature setting is stored in this field. Where 1°C (°F)=10.

■Pulse CH fields

Field 1: Pulse CH number

The pulse CH number is written to this field.

Field 2: Span lower limit A/D value

The A/D value of the span lower limit is stored in this field. This is a count value, and thus will not have a negative value.

Field 3: Span upper limit A/D value

The A/D value of the span upper limit is stored in this field. This is a count value, and thus will not have a negative value.

2.8.2. $$Scale subsection

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.

Setting example

```
$$Scale
  CH1     = Off,  -20000,  +20000,   -5000,   +5000,  +2, " V"
  CH2     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH3     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH4     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH5     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH6     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH7     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH8     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH9     = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
  CH10    = Off,  -20000,  +20000,   -5000,   +5000,  +1, " V"
```

Field 1: CH number

The channel numbers are written to this field.

Field 2: Scaling enable/disable setting

Enable (ON) or disable (OFF) of the scaling function is written to this field.

Field 3: Input-side lower limit A/D value

The A/D value of the input-side lower limit is stored in this field. When setting the voltage, ±20000=±FS.

When setting the temperature, 1°C (°F)=10, and the upper limit is always the same as the lower limit.

Field 4: Input-side upper limit A/D value

The A/D value of the input-side upper limit is stored in this field. When setting the voltage, ±20000=±FS.

When setting the temperature, 1°C (°F)=10, and the upper limit is always the same as the lower limit.

Field 5: Output-side lower limit

The input-side lower limit is stored in this field. The output numeric value from which a decimal point is removed is stored as an integer value. This is a binary value within the range of ±100000.

Examples: +1.000 → +1000 +1.0000 → +10000 +10.00 → +1000

When setting the temperature, the upper limit is always the same as the lower limit.

Field 6: Output-side upper limit

The input-side upper limit is stored in this field. The output numeric value from which a decimal point is removed is stored as an integer value. This is a binary value within the range of ±100000.

Examples: +1.000 → +1000 +1.0000 → +10000 +10.00 → +1000

When setting the temperature, the upper limit is always the same as the lower limit.

Field 7: Output-side decimal point position

The decimal point position of the output value is stored in this field.

The final output string is created by combining with the output-side binary value.

+0 = 4 digits below the decimal point

+1 = 3 digits below the decimal point

+2 = 2 digits below the decimal point

+3 = 1 digit below the decimal point

+4 = No decimal point

Example: When the output value +1000 and the decimal point position +1 → +1.000

When the output value +1000 and the decimal point position +3 → +100.0

Field 8: Post-scaling unit

The post-scaling unit is stored as a string. Up to 8 single-byte characters.

2.8.2.1. $$$Pulse subsection

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

Setting example

```
$$$Pulse
  Pulse1   = On ,    +1,    +1,  +5000,  +5000, "C"
  Pulse2   = Off,    +1,    +1,     +1,     +1, "C"
  Pulse3   = Off,    +1,    +1,     +1,     +1, "C"
  Pulse4   = Off,    +1,    +1,     +1,     +1, "C"
```

Field 1: Pulse CH number

The pulse CH number is written to this field.

Field 2: Scaling enable/disable setting

Enable (ON) or disable (OFF) of the scaling function is written to this field.

Field 3: Input-side pulse value

The input-side pulse value is stored in this field. This value is a binary value within the range of 1-5000.

Field 4: Input-side pulse value

This field is not used. Currently the same value as that of Field 3 is written to this field.

Field 5: Output-side lower limit

The output-side pulse value is stored in this field. This value is a binary value within the range of 1-5000.

Field 6: Output-side upper limit

This field is not used. Currently the same value as that of Field 5 is written to this field.

Field 7: Post-scaling unit

The post-scaling unit is stored as a string. Up to 8 single-byte characters.

2.8.3. $$Calculation subsection

The inter-CH operation settings at the time of measurement data capture are written to this field.
Please note that only the channels for which measurement data exists are written to this section.

Setting example

```
$$Calculation
  CH1     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH2     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH3     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH4     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH5     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH6     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH7     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH8     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH9     = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
  CH10    = Off,  1, ADD ,  1, M1     ,  -10000,  +10000,  +0, "V       "
```

Field 1: CH number

The channel numbers are written to this field.

Field 2: ON/OFF setting

The inter-CH operation ON/OFF setting is written to this field.

Field 3: Left term CH setting

The left term CH of the inter-CH operation is written to this field.

Field 4: Operation item setting

The operation term of the inter-CH operation is written to this field.

Operation		
Addition	+	ADD
Subtraction	-	SUB
Multiplication	*	MULT
Division	/	DIV

Field 5: Right term CH setting

The right term CH of the inter-CH operation is written to this field.

Field 6: Scaling settings

The scaling settings for the data after the inter-CH operation are written to this field.

Scaling	
/ 1000000	D1000000
/ 1000	D1000
*1	M1
* 1000	M1000
*1000000	M1000000

Field 7: Span lower limit setting for inter-CH operation

The lower limit setting among the span settings for the inter-CH operation is written to this field.
This value is a binary value within the range of ±100000

Field 8: Span upper limit setting for inter-CH operation

The upper limit setting among the span settings for the inter-CH operation is written to this field.
This value is a binary value within the range of ±100000.

Field 9: Span decimal point position setting for operation

The decimal position among the span settings for the inter-CH operation is written to this field.
The final value is created by combining with the span binary value.

+0 = 4 digits below the decimal point
+1 = 3 digits below the decimal point
+2 = 2 digits below the decimal point
+3 = 1 digit below the decimal point
+4 = No decimal point

Example: When the output value +1000 and the decimal point position +1 → +1.000
When the output value +1000 and the decimal point position +3 → +100.0

Field 10: Operation unit setting

The unit setting for the inter-CH operation is written to this field.

2.9. $Alarm section

The alarm setting status is written to this section. It is always written for all CHs.

Setting example

```
$Alarm
  CH1    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH2    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH3    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH4    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH5    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH6    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH7    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH8    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH9    = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
  CH10   = Off  , +0,  +0, +5000,  +0,  +0, +1000, 1
```

Field 1: CH number

The channel numbers are written to this field.

Field 2: Alarm function enable/disable setting

Enable (ON) or disable (OFF) of the alarm function is written to this field.

Field 3: Level value for voltage Hi/Lo

The level value in the case that the alarm is set to Hi/Lo for the voltage and humidity settings is written to this field. This value is a binary value within the range of ±20000.

Field 4: Lower limit level value for voltage WinIn/WinOut

The lower limit level value in the case that the alarm is set to WinIn/WinOut for the voltage and humidity settings is written to this field. This value is a binary value within the range of ±20000.

Field 5: Upper limit level value for voltage WinIn/WinOut

The upper limit level value in the case that the alarm is set to WinIn/WinOut for the voltage and humidity settings is written to this field. This value is a binary value within the range of ±20000.

Field 6: Level value for temperature Hi/Lo

The level value in the case that the alarm is set to Hi/Lo for the temperature setting is written to this field. This value is a binary value where 1°C (°F)=10.

Field 7: Lower limit level value for temperature WinIn/WinOut

The lower limit level value in the case that the alarm is set to WinIn/WinOut for the temperature setting is written to this field. This value is a binary value where 1°C (°F)=10.

Field 8: Upper limit level value for temperature WinIn/WinOut

The upper limit level value in the case that the alarm is set to WinIn/WinOut for the temperature setting is written to this field. This value is a binary value where 1°C (°F)=10.

Field 9: Alarm output port

The alarm output port at the time of an alarm generation is written to this field. The range is 1-4.

2.9.1. $$Pulse subsection

The pulse alarm settings are written to this section.

Setting example

```
$$Pulse
  Pulse1 = Off   ,   0,   0,   100, 1
  Pulse2 = Off   ,   0,   0,   100, 1
  Pulse3 = Off   ,   0,   0,   100, 1
  Pulse4 = Off   ,   0,   0,   100, 1
```

Field 1: Pulse CH number

The pulse CH number is written to this field.

Field 2: Alarm function enable/disable setting

Enable (ON) or disable (OFF) of the alarm function is written to this field.

Field 3: Level value for pulse Hi/Lo

The level value in the case that the pulse CH alarm is set to Hi/Lo is written to this field. This value is a binary value within the range of 0-500000000.

Field 4: Lower limit level value for pulse WinIn/WinOut

The lower limit level value in the case that the pulse CH alarm is set to WinIn/WinOut is written to this field. This value is a binary value within the range of 0-500000000.

Field 5: Upper limit level value for pulse WinIn/WinOut

The upper limit level value in the case that the pulse CH alarm is set to WinIn/WinOut is written to this field. This value is a binary value within the range of 0-500000000.

Field 6: Alarm output port

The alarm output port at the time of an alarm generation is written to this field. The range is 1-4.

2.9.2. $$Logic subsection

The logic alarm settings are written to this section.

Setting example

```
$$Logic
```

```
Pattern    = HLXX
Port       = 1, 1, 1, 1
```

Pattern

Form	Pattern = <L1><L2><L3><L4>
Setting example	Pattern = HLXX
Function	The logic alarm mode settings are written to this line.
Explanation	The settings are stored in the order of L1,L2,L3,L4 from the left. H: Hi L: Lo X: Don't Care

Port

Form	Port = <L1>,<L2>,<L3>,<L4>
Setting example	Port = 1, 2, 3, 4
Function	The output port settings at the time of a logic alarm generation are written to this line.
Explanation	The settings are stored in the order of L1,L2,L3,L4 from the left. The range is 1-4.

2.9.3. $$Condition subsection

The alarm detection method is written to this subsection.

Setting example

```
$$Condition
  Combination = Level
```

Combination

Form	Combination = <Level ／ Edge>
Setting example	Combination = Level
Function	The alarm detection method is written to this line.
Explanation	Sets the alarm detection method for analog CH and pulse CH. Level: Put in an alarm status whenever equal to or higher (lower) than the set level. Edge: Alarms when the set level is crossed.

2.10. $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.

Setting example

```
$Trigger
```

```
Repeat        = Off
```

Repeat

Form	Repeat = <On ／ Off>
Setting example	Repeat = Off
Function	The repeat capture setting is written to this line.
Explanation	Sets whether the repeat capture is performed or not when capturing is stopped by a stop trigger.

2.10.1. $$Start subsection／$$Stop subsection

The trigger operation settings are written to this subsection.

Setting example

```
$$Start / $$Stop
  Source        = Off
  Combination = LevelOR
  CH1           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH2           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH3           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH4           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH5           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH6           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH7           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH8           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH9           = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  CH10          = Off   ,  +0,   +0, +5000,    +0,    +0, +1000
  Pulse1        = Off   , 0       , 0       , 100
  Pulse2        = Off   , 0       , 0       , 100
  Pulse3        = Off   , 0       , 0       , 100
  Pulse4        = Off   , 0       , 0       , 100
  Logic         = XXXX
  AbsTime       = "1980-01-01 00:00:01"
  RelTime       = "00:00:01"
  RecTime       =     0hr00min01sec   (*Does not exist in the $$Start subsection)
  WeeklyFlag   =   85
  WeeklyTime   = "01:02:03"
  Alarm         = 1
```

Source

Form	Source = < Off／Level／Alarm／External／Date／Weekly／Time>
Setting example	Source = Off
Function	The trigger source settings are written to this line.
Explanation	Off: No triggers are used.
	Level: Triggered by level comparison.

Alarm: Triggered by an alarm generation.

External: Triggered by an external input.

Date: Triggered by a specified date and time. (*Only time of day can be specified for repeat)

Weekly: Triggered by a specified day of the week and a specified time.

Time: Triggered by a time interval. (*Only for $$Stop)

Combination

Form Combination= <LevelOR／LevelAND／EdgeOR／EdgeAND>

Setting example Combination= LevelOR

Function The operation mode settings for the level comparison trigger are written to this line.

Explanation

LevelOR: A level comparison mode. Trigger is enabled when one of the trigger enable CHs meets the conditions.

LevelAND: A level comparison mode. Trigger is enabled only when all of the trigger enable CHs meet the conditions.

EdgeOR: An edge comparison mode. Trigger is enabled when one of the trigger enable CHs meets the conditions.

EdgeAND: An edge comparison mode. Trigger is enabled only when all of the trigger enable CHs meet the conditions.

Level comparison mode: Trigger conditions are met whenever equal to or higher (lower) than the set level.

Edge comparison mode: Trigger is enabled when the set level is crossed.

■Analog CH fields

The settings when the trigger source settings are in the level comparison mode are written to these fields.

Field 1: CH number

The channel numbers are written to this field.

Field 2: Trigger function enable/disable setting

Enable (ON) or disable (OFF) of the trigger function is written to this field.

Field 3: Level value for voltage Hi/Lo

The level value in the case that the trigger is set to Hi/Lo for the voltage and humidity settings is written to this field. This value is a binary value within the range of ±20000.

Field 4: Lower limit level value for voltage WinIn/WinOut

The lower limit level value in the case that the trigger is set to WinIn/WinOut for the voltage and humidity settings is written to this field. This value is a binary value within the range of ±20000.

Field 5: Upper limit level value for voltage WinIn/WinOut

The upper limit level value in the case that the trigger is set to WinIn/WinOut for the voltage and humidity settings is written to this field. This value is a binary value within the range of ±20000.

Field 6: Level value for temperature Hi/Lo

The level value in the case that the trigger is set to Hi/Lo for the temperature setting is written to this field. This value is a binary value where 1°C (°F)=10.

Field 7: Lower limit level value for temperature WinIn/WinOut

The lower limit level value in the case that the trigger is set to WinIn/WinOut for the temperature setting is written to this field. This value is a binary value where 1°C (°F)=10.

Field 8: Upper limit level value for temperature WinIn/WinOut

The upper limit level value in the case that the trigger is set to WinIn/WinOut for the temperature setting is written to this field. This value is a binary value where 1°C (°F)=10.

■Pulse CH fields

The settings when the trigger source settings are in the level comparison mode are written to these fields.

Field 1: CH number

The channel numbers are written to this field.

Field 2: Trigger function enable/disable setting

Enable (ON) or disable (OFF) of the trigger function is written to this field.

Field 3: Level value for pulse Hi/Lo

The level value in the case that the pulse CH trigger is set to Hi/Lo is written to this field. This value is a binary value within the range of 0-500000000.

Field 4: Lower limit level value for pulse WinIn/WinOut

The lower limit level value in the case that the pulse CH trigger is set to WinIn/WinOut is written to this field. This value is a binary value within the range of 0-500000000.

Field 5: Upper limit level value for pulse WinIn/WinOut

The upper limit level value in the case that the pulse CH trigger is set to WinIn/WinOut is written to this field. This value is a binary value within the range of 0-500000000.

Logic

Form	Logic = <L1><L2><L3><L4>
Setting example	Logic = HLXX
Function	The logic trigger settings are written to this line.
Explanation	The settings when the trigger source settings are in the level comparison mode are written to this line. The settings are stored in the order of L1, L2, L3, L4 from the left. H: Hi L: Lo X: Don't Care

AbsTime

Form	AbsTime = "YYYY-MM-DD hh:mm:ss"
Setting example	AbsTime = "1980-01-01 00:00:01"
Function	The trigger date and time for triggering at a specified date and time is written to this line.
Explanation	The trigger date and time setting for the trigger source settings triggering at a specified date and time is written to this line. Used when the repeat setting is OFF.

RelTime

Form	RelTime = "hh:mm:ss"
Setting example	RelTime = "00:00:01"
Function	The trigger time for triggering at a specified time of day is written to this line.
Explanation	The trigger time setting for the trigger source settings triggering at a specified time of day is written to this line. Used when the repeat setting is ON. This is the triggering time for every day.

RecTime

Form	RecTime = <hhhh>hr<mm>min<ss>sec
Setting example	RecTime = 0hr00min01sec
Function	The time interval for triggering at a time interval is written to this line.
Explanation	The time interval setting for the trigger source settings triggering at a time interval is written to this line. Exists only for $$Stop. The range is 0hr00min00sec - 9999hr59min59sec.

WeeklyFlag

Form WeeklyFlag = <Flag value>

Setting example WeeklyFlag = 0

Function The trigger day of the week for triggering at a specified day of the week and time is written to this line.

Explanation The trigger day of the week setting for the trigger source settings triggering at a specified day of the week and time is written to this line. The trigger day of the week corresponds to respective bits, which are set to 0 for OFF and 1 for ON. The binary value corresponding to the bits is converted into a decimal value and is then written as a string.

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
Sun	Mon	Tue	Wed	Thu	Fri	Sat

Example: When Sunday, Thursday and Saturday are ON,

1010001（binary） = 51(hexadecimal) = 81(decimal), and thus

WeeklyFlag = 81

A trigger goes off at the time specified by WeeklyTime on the days specified as "ON" here.

WeeklyTime

Form WeeklyTime = "hh:mm:ss"

Setting example WeeklyTime = "00:00:00"

Function The trigger time of day for triggering at a specified day of the week and time is written to this line.

Explanation A trigger goes off at the time specified here on the days specified as "ON" by WeeklyFlag.

Alarm

Form Alarm = <1／2／3／4>

Setting example Alarm = 1

Function The alarm generation port at the time of an alarm generation trigger is written to this line.

Explanation The alarm generation port at the time of the trigger source triggered by an alarm generation is written to this line. When the alarm port specified here is put in an alarm status, it is triggered. The range is 1-4.

2.11. $Annotation section

The annotation settings are written to this section. Please note that only the channels for which measurement data exists are written to this section.

Setting example

```
$Annotation
  CH1       = " CH 1"
  CH2       = " CH 2"
  CH3       = " CH 3"
  CH4       = " CH 4"
  CH5       = " CH 5"
  CH6       = " CH 6"
  CH7       = " CH 7"
  CH8       = " CH 8"
  CH9       = " CH 9"
  CH10      = " CH10"
```

Field 1: CH number

The channel numbers are written to this field.

Field 2: Annotation string settings

An annotation string is written to this field. Up to 31 single-byte ASCII characters may be used.

2.12. $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 GL220 / GL820.