

Logger Dump Request Format

This format is used to request a dump of LogMany logger contents. Following reception of this request, the RUG3 will reply with the specified items up to the length of the communications buffer or 255 bytes maximum message length including header and CRC. Note that there are two dump request formats. The first initiates the dump and specifies all aspects of the dump source and format including which logger is to be dumped, time resolution of the time tags, first index to be sent, number of indices to be sent, synchronizing time stamp, and the format of all analogs to be sent. The second dump request is much shorter for efficiency and specifies whether the last dump is to be repeated (in case it was not received by the requester) or the next series of log indices is to be sent. End of log is signaled by a bit in the dump reply control field. If both the first log index to send and number of indices to send are zero, then the dump will commence with the most recent sample and continue until all samples taken since the last dump have been sent. The count of samples since the last dump is maintained by each logger and is cleared automatically at the end of each dump. Note that the format for RUG5/9 differs from that of the RUG3 in that a byte has been inserted (byte 8) for logger ID to increase the range of logger ID's within each unit.

Table 1 RUG5/9 LOGGER INITIAL DUMP REQUEST FORMAT

Byte #	0-6	7		8	9,10	11,12	13-16	17-N	N+1, N+2
Funct	HDR	Control field		Logger ID	First log index to send	No. of indices to send	Time stamp	Analog formats	CRC -16
		MS 5 bits: B7:0=Logmany 1=Spare B6:0=1sec tags 1=spare B5,4,3: 0=init request 1=send next 2=repeat last 3=restart dump with existing setup 4=clr logger's 'new sample' counter 5-7=spares	LS 3 bits: Spare	Logger ID, 0-255	MS, LS, first log index to send counting from most recent logged item	MS, LS, number of indices to send	UNIX format time stamp, 1 sec. res. to init RTC. If zero, no effect on RTC	4-bits/analog: 0=skip value 1=4-byte float 2,3=spares Below: mult analog by value then send as signed int 4=*10,000 5=*1000 6=*100 7=*10 8=*1 9=*0.1 10=*0.01 11=*0.001 12=*0.0001 13=*0.00001 14,15=spares	

Subsequent dump requests do not need full definition of all parameters, only a command to send the next set of samples, repeat the last set of samples, or restart the dump from the start. That format is presented below.

Table 2 RUG3 LOGGER SUBSEQUENT DUMP REQUEST FORMAT

Byte #	0-6	7		8	9,10
Function	HEADER	Control field			CRC-16
		MS 5 bits: B7:0=Logmany 1=Spare B6:0=1 sec tags 1=Spare B5,4,3: X=init request 1=send next 2=repeat last 3=restart dump with existing setup 4=clr logger's 'new sample' counter 5-7=spares	LS 3 bits: Spare	Logger ID, 0-255	

Logger Dump Format

This format is a response to the above dump requests and is used to dump logger contents. Contents of either the event loggers or the log many modules can be dumped using these formats. Which type of logger is to be dumped is contained in the dump request above. Following reception of the request, the RUG3 will reply with logged data or events as defined in the table below. The first byte after the standard 7-byte header is a one byte dump control byte that indicates whether the message contains the last samples from the logger and, in the case of the LogMany module, whether the analog values are followed with a 16 bit status field containing the packed states of all logged statuses for those records.

Table 3 RUG3 LOGGER DUMP FORMAT (applies to both LogMany and Event Logger)

Byte #	0-6	7	8-N	N+1,N+2
Function	HEADER	Dump control	Logged event/data item(s)	CRC-16
		For LogMany: B7: 1=End of log B6: 1=16 bit status word follows last analog B5-0: spares	One or more records from LogMany module	

Each logged item in bytes 8-N above consists of a self-contained variable length binary message consisting of a one byte preamble followed by a 2, 4, or 6 byte time tag. Formats of these items are presented below for a LogMany module. After the time tag are one or more analog values. The preamble identifies whether the time tag is absolute or relative to the last absolute tag sent; contains the event status; and specifies the event index or number of analogs following the time tag. The first item in a message will always contain an absolute time tag of 4 byte length for 1 second time resolution. Subsequent items will use relative time tags of the same resolution unless the delta time between the new item and the last absolute time tag exceeds a two byte range. In that case, a new absolute time tag will be sent with the item; and all subsequent items in that response will be relative to it. Time tags present the number of seconds since January 1, 1970. Time tags in any message are presented MS byte first followed by intermediary bytes and ending with the LS byte.

The table below presents the data format for a single record from the LogMany module. The first preamble byte defines the format for that event of all data following the preamble. The most significant preamble bit is a spare. The next most significant bit defines the time tag format. Remaining bits are spares.

Table 4 RUG3 LOGMANY ITEM FORMAT

Byte #	0		1-N	N+1,...N+X
Function	PREAMBLE		Time Tag (2, 4 or 6 bytes)	Analog Value(s)
	B7: Spare B6: 0=abs time tag 1=rel time tag B5: Spare	B0-B4: Spares	Absolute 4-byte (1 sec resolution) Relative 2-byte (1 sec resolution) Byte order: MS...LS	2-byte signed integers, or 4-byte floats, (Format specified by initial request) Byte order: Integer: MS,LS Float: Exp,MS...LS

Logger Dump Request/Response Examples

Below is an example of requesting all records from a LogMany module that have been logged since the last dump. The LogMany has been recording four analog values and two statuses per record. The request specifies 1 second time tags followed by two integers, a floating point value and an integer value per record, followed by the two statuses. The logger had saved 4 records since the last dump.

INITIAL DUMP REQUEST:

\$C8 Header sync byte
\$15 Message length, 21 bytes including CRC
\$40 Message type (request logger dump using extended RTU addresses)
\$00 01 Source address \$0001
\$02 05 Destination address \$0205=517
\$00 Control field, specifies logmany with 1 sec time tags, initial request
\$04 LogMany ID=4
\$00 00 First log index to send=0
\$FF FF Number of indices to send=\$FFFF=65535 (all there are)
\$00 00 00 00 Time tag to install in RTU's RTC, 0=don't effect RTC
\$68 8=report 1st logged value as integer*1.0, 6=report 2nd value as integer value*100.0
\$81 1=report 3rd value as 4-byte float, 8=report 4th value as integer*1.0
\$1E 10 CRC

RESPONSE:

\$C9 48 40 00 01 02 05 Reply, 72 byte length, source addr=517, destination addr=1
\$C0 Dump control, end of log, 16 bit status word follows last analog of each record

\$00 Start of 1st logged record...uses absolute time tag

\$4B E6 CA D0 Absolute time tag=05/08/2010 14:46:41

\$00 28 First value=40

\$12 20 Second value=4640 (46.40 * 100.0)

\$42 F6 E6 66 Third value=123.45

\$00 2E Fourth Value=46

\$00 02 Statuses: first status=0, second status=1

\$40 Start of second logged record, uses relative time tag

\$00 01 Relative time tag, 1 second prior to first logged item

\$00 27 First value=39

\$12 1F Second value=4639 (46.39 * 100.0)

\$42 F6 E6 66 Third value=123.45

\$00 2E Fourth Value=46

\$00 02 Statuses: first status=0, second status=1

\$40 Start of third logged record, uses relative time tag

\$00 02 Relative time tag, 2 seconds prior to first logged item

\$00 26 First value=38

\$12 1E Second value=4638 (46.38 * 100.0)

\$42 F6 E6 66 Third value=123.45

\$00 2E Fourth Value=46

\$00 02 Statuses: first status=0, second status=1

\$40 Start of fourth logged record, uses relative time tag

\$00 03 Relative time tag, 3 seconds prior to first logged item

\$00 25 First value=37

\$12 1D Second value=4637 (46.37 * 100.0)

\$42 F6 E6 66 Third value=123.45

\$00 2E Fourth Value=46

\$00 02 Statures: first status=0, second status=1