TASCAM Sonicview 16 / 24

Meter data protocol specifications

Ver. 1.00

Jun 11, 2025

TEAC Corporation

Contents

CAUTION

TEAC Corporation (“TEAC”) hereby grants the customer (“YOU”) the license to use of the protocol described and the specification document (collectively "The Protocol") under the following conditions that YOU consent and are compliant to the following terms ("Protocol Use Agreement").

If you do not consent to the following conditions in this Protocol Use Agreement, you may not use this protocol and should return this document to TEAC. Please kindly be advised that any violations of any of the following terms in the Protocol Use Agreement consists of infringement on the rights of TEAC, and in such a case YOUR right of use of the protocols shall be forfeited and YOU are responsible for compensating the damages or any other claims.

Terms of Protocol Use Agreement

1. This Agreement comes into effect from the moment YOU start to use of the Protocol.

2. TEAC grants YOU a non-exclusive and non-assignable right to use the Protocol only for the purpose to develop compatible devices (including software), which shall be designed for and used together with certain designated TASCAM products by TEAC.

3. YOU shall not construe that your obtaining the Protocol imply that YOU have or are granted any of TEAC’s rights, titles, licenses or interests in the Protocol unless expressly specified in this Agreement. YOU shall recognize and acknowledge that the Protocol is protected as a copyrighted work of TEAC under any applicable copyright laws such as the Universal Copyright Convention and the Berne Convention for the Protection of Literary and Artistic Works for its signatory nations as well as the Copyright Act of Japan. Without exception, the intellectual property in the Protocol belongs to TEAC or its supplier of TEAC.

4. (1) YOU shall not make copies of the Protocol except for the purpose set forth in section 2 above.

(2) YOU shall not assign or transfer the Protocol to any third party separately from the designated TASCAM product without the prior consent of TEAC.

(3) Since TEAC’s confidential information may be contained in the Protocol, YOU shall not disclose any of them to any third party without obtaining prior permission from TEAC.

(4) YOU shall neither use TEAC trademark and/or tradename nor promote and/or advertise the relationship with TEAC in relation with the Protocol, unless YOU have obtained the prior permission from TEAC.

(5) If YOU make any inventions, improvements or creations based on the Protocol ("Inventions"), YOU shall neither apply for registration or file them as YOUR intellectual property rights with any competent governmental agency, but shall report the Inventions to TEAC without delay and obtain the permission of TEAC.

5. The Protocols herein are provided on as-is basis. They are provided without any kind of warranty, express or implied, including but not limited to, of merchantability, fitness for a particular purpose, non-infringement of any third-party and/or freedom from errors or defects.

Furthermore, the Protocol may be changed or amended, in whole or in part, without any notice.

6. TEAC shall have NO obligations to answer to YOUR inquiries on this specifications document or protocols.

7. TEAC SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES (INCLUDING UNAVAILABILITY OF THE USE, BUSINESS LOSSES OR CHANCES, BUSINESS INTERRUPTION, LOSS OF BUSINESS DATA, OR OTHER FINANCIAL DAMAGES) ARISING FROM OR IN CONNECTION WITH THE USE OF THIS SPECIFICATIONS DOCUMENT OR THE PROTOCOLS. This shall be applicable, without prejudice to any rights or obligations of TEAC hereinabove, even if TEAC has any knowledge of the potential damages in advance.

8. This Agreement shall be governed by and construed under the laws of Japan. Any disputes arising out of or in connection with this Agreement shall be subject to the exclusive jurisdiction of the Tokyo District Court in the first instance.

End of Use Agreement

Overview

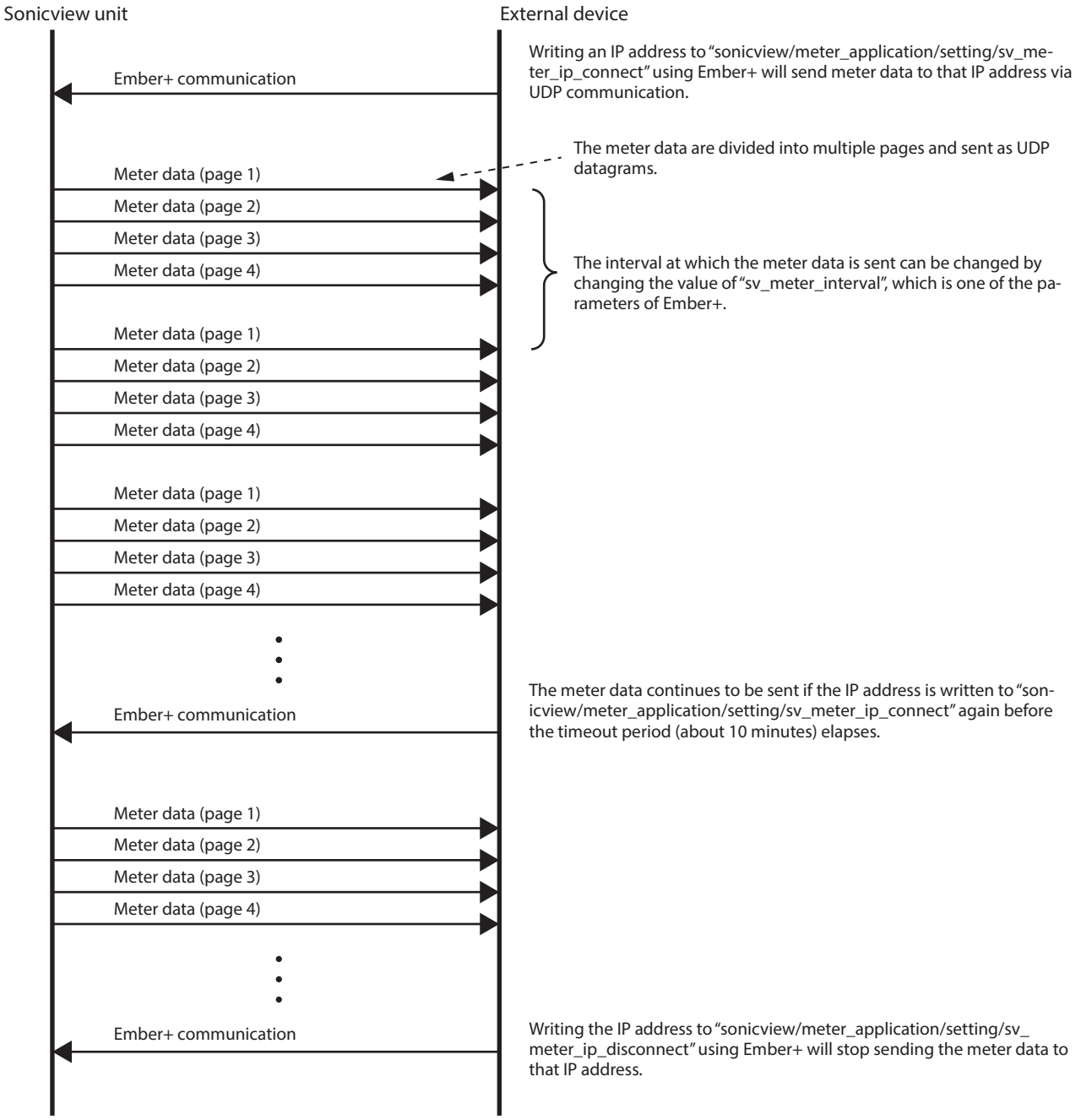
The ETHERNET port on the Sonicview 16 and Sonicview 24 ("Sonicview unit") can be used to allow external devices such as computers to retrieve audio levels from inside the Sonicview unit.

Specifications

Transmission protocol:	UDP
Destination port number:	41624 (can be changed via Ember+)
Transmission starts when:	IP address is written to "sonicview/meter_application/setting/sv_meter_ip_connect" using Ember+ or, during Ember+ connection, "sv_meter_interval", which is one of the Ember+ parameters, is set to a value other than 0.
Transmission ends when:	IP address is written to "sonicview/meter_application/setting/sv_meter_ip_disconnect" using Ember+ or, during Ember+ connection, "sv_meter_interval", which is one of the Ember+ parameters, is set to 0. The timeout period (about 10 minutes) elapses after the IP address is written to "sonicview/meter_application/setting/sv_meter_ip_connect" using Ember+.

For more information about Ember+, refer to "Sonicview Ember+ Tree Structure Specification".

Sequence example



Meter data

Meter data packets are divided into a header section and a data section.
All data other than character strings are 2-byte little-endian integers.

		size (bytes)
Header section	ASCII character string "SV_METER" at the beginning of packet	**8**
	Packet version	**2**
	Packet type character string "DATA"	**4**
	reserved	**10**
	Packet counter	**2**
	Page number	**2**
	CRC16 for data section	**2**
	Data section size	**2**
Data section	Refer to "Data section".	

Header section

ASCII character string "SV_METER" at the beginning of packet

Fixed at the 8-byte character string "SV_METER".

Packet version

The current packet version is "0x0000".

The data section specifications are determined by the packet version. Specifically, the meaning of the tag and the total number of pages are determined.

In other words, the contents of the data section change depending on the packet version.

Packet type character string

Fixed at the 4-byte character string "DATA".

Packet counter

This is a counter that increases by 1 each time a meter is sent (for each transmission interval). After 65535 (0xffff), it returns to 0.

If the following condition is not met, the meter data has been lost or the order of arrival has been swapped.

(Due to UDP communication, there may be cases where data is lost or the order of arrival is swapped.)

<previous meter data counter value> + 1 = <current meter data counter value>

Page number

The contents of the data section are determined by the packet version and the page number. Details will be explained below.

CRC16 for data section

The "CRC" used in this protocol is referred to as CRC-16/XMODEM, CRC-16/ACORN, CRC-16/LTE, CRC-16/V-41-MSB, etc. The generating polynomial is "x^16 + x^12 + x^5 + 1".

Using the program explained below, the CRC value of the data section is calculated and compared with the CRC value of the header.

If they do not match, the data contains an error area and is thus invalid.

Data section size

This is the size of the data section that follows.

Data section

The data section consists of three items, "tag", "size" and "payload", and is repeated multiple times.

The contents of the data section change depending on the page number. See "Actual data section" on page 13 for details.

"tag" indicates the type of payload data. See "Data section tag" on page 6 for details.

"size" indicates the size of payload.

"payload" consists of one or more 2-byte integers. In addition, payload is a meter value or RTA Meter index.

Each data is a 2-byte little-endian integer.

For example, the meter data for the built-in player and built-in recorder are as follows.

			bytes	Contents
tag	Built-in player		2	0x001F
size			2	0x0004
payload	2 pieces of meter data (L, R)	Left-channel meter data	2	0x____
		Right-channel meter data	2	0x____
tag	Built-in recorder		2	0x0020
size			2	0x0004
payload	2 pieces of meter data (L, R)	Left-channel meter data	2	0x____
		Right-channel meter data	2	0x____

In addition to the above, all meter data, RTA data and peak hold values are between 0x0000 and 0x7fff and are converted to dB values as shown in "Converting to dB" on page 11.
Below is an example of RTA (real-time analyzer) data.

			bytes	Contents
tag	RTA 1L Meter index		2	0x0050
size			2	0x0002
payload	Meter index		2	0x____
tag	RTA 1L data		2	0x0051
size			2	0x003e
payload	31 pieces of RTA data	band-1 RTA data	2	0x____
		band-2 RTA data	2	0x____
		band-3 RTA data	2	0x____
		band-30 RTA data	2	0x____
		band-31 RTA data	2	0x____

For specific values of the Meter index, refer to "Tags and metering points" on page 8.

Requesting RTAs

To request an RTA, use Ember+ to write the Channel ID you want the RTA for to "sonicview/meter_application/setting/sv_meter_rta_ch_id".
See "Meter Indexes and Channel IDs" on page 11 for Channel IDs.
For stereo-linked CH modules or MIX modules, use the channel ID of the left channel.
Sonicview 16/Sonicview 24 has 3 RTA processors, where RTA requests are overwritten in order from oldest to newest.
The 3 RTA processors are also shared when displaying RTA on the Sonicview unit's screen or in Sonicview Control.
If an RTA request is made from another Ember+ app, the RTA data for a channel you do not want might be received.
Always check the Meter index in the data section to confirm that the RTA data received is the one you want.
A Channel ID is used for an RTA request, but the data sent is indicated by the Meter index.

Data section tag

The tag for a packet version of 0x0000 is shown below.

Value	Explanation
0x0001	CH 1–40: INPUT
0x0002	CH1-40: POST D.TRIM (PRE HPF)
0x0003	CH1-40: POST GATE (PRE EQ)
0x0004	CH1-40: POST EQ (PRE COMP)
0x0005	CH1-40: POST COMP (PRE FADER)
0x0006	CH1-40: POST FADER
0x0007	ST-IN 1-2: INPUT
0x0008	ST-IN 1-2: POST D.TRIM (PRE EQ)
0x0009	ST-IN 1-2: POST EQ (PRE FADER)
0x000A	ST-IN 1-2: POST FADER
0x000B	FX 1-4: INPUT
0x000C	FX 1-4: POST FX (PRE EQ)
0x000D	FX 1-4: POST EQ (PRE FADER)
0x000E	FX 1-4: POST FADER
0x000F	MIX1-22: INPUT (PRE EQ)
0x0010	MIX1-22: POST EQ (PRE GEQ)
0x0011	MIX1-22: POST GEQ (PRE COMP)
0x0012	MIX1-22: POST DELAY (PRE FADER)
0x0013	MIX1-22: POST FADER
0x0014	MIX1-22: OUTPUT
0x0015	Main: INPUT (PRE EQ)
0x0016	Main: POST EQ (PRE GEQ)
0x0017	Main: POST GEQ (PRE COMP)

Value	Explanation
0x0018	Main: POST DELAY (PRE FADER)
0x0019	Main: POST FADER
0x001A	Main: OUTPUT
0x001B	OSC post level
0x001C	TB POST D.TRIM
0x001D	RTB POST D.TRIM
0x001E	MONITOR 1/2 stereo
0x001F	Built-in player
0x0020	Built-in recorder
0x0030	CH1-40: GATE KEY-IN
0x0031	CH1-40: GATE GR
0x0032	CH1-40: COMP KEY-IN
0x0033	CH1-40: COMP GR
0x0034	MIX1-22: COMP KEY-IN
0x0035	MIX1-22: COMP GR
0x0036	Main: COMP KEY-IN
0x0037	Main: COMP GR
0x0040	Auto Mixer: input/output
0x0041	Auto Mixer: GAIN
0x0042	MTR: input monitor/player
0x0050	RTA 1L Meter index
0x0051	RTA 1L data
0x0052	RTA 1R Meter index
0x0053	RTA 1R data
0x0054	RTA 2L Meter index
0x0055	RTA 2L data
0x0056	RTA 2R Meter index
0x0057	RTA 2R data
0x0058	RTA 3L Meter index
0x0059	RTA 3L data
0x005A	RTA 3R Meter index
0x005B	RTA 3R data
0x0101	CH1-40: INPUT peak hold value
0x0102	CH1-40: POST D.TRIM (PRE HPF) peak hold value
0x0103	CH1-40: POST GATE (PRE EQ) peak hold value
0x0104	CH1-40: POST EQ (PRE COMP) peak hold value
0x0105	CH1-40: POST COMP (PRE FADER) peak hold value
0x0106	CH1-40: POST FADER peak hold value
0x0107	ST-IN 1-2: INPUT peak hold value
0x0108	ST-IN 1-2: POST D.TRIM (PRE EQ) peak hold value
0x0109	ST-IN 1-2: POST EQ (PRE FADER) peak hold value
0x010A	ST-IN 1-2: POST FADER peak hold value
0x010B	FX 1-4: INPUT peak hold value
0x010C	FX 1-4: POST FX (PRE EQ) peak hold value
0x010D	FX 1-4: POST EQ (PRE FADER) peak hold value
0x010E	FX 1-4: POST FADER peak hold value
0x010F	MIX1-22: INPUT (PRE EQ) peak hold value
0x0110	MIX1-22: POST EQ (PRE GEQ) peak hold value
0x0111	MIX1-22: POST GEQ (PRE COMP) peak hold value
0x0112	MIX1-22: POST DELAY (PRE FADER) peak hold value
0x0113	MIX1-22: POST FADER peak hold value

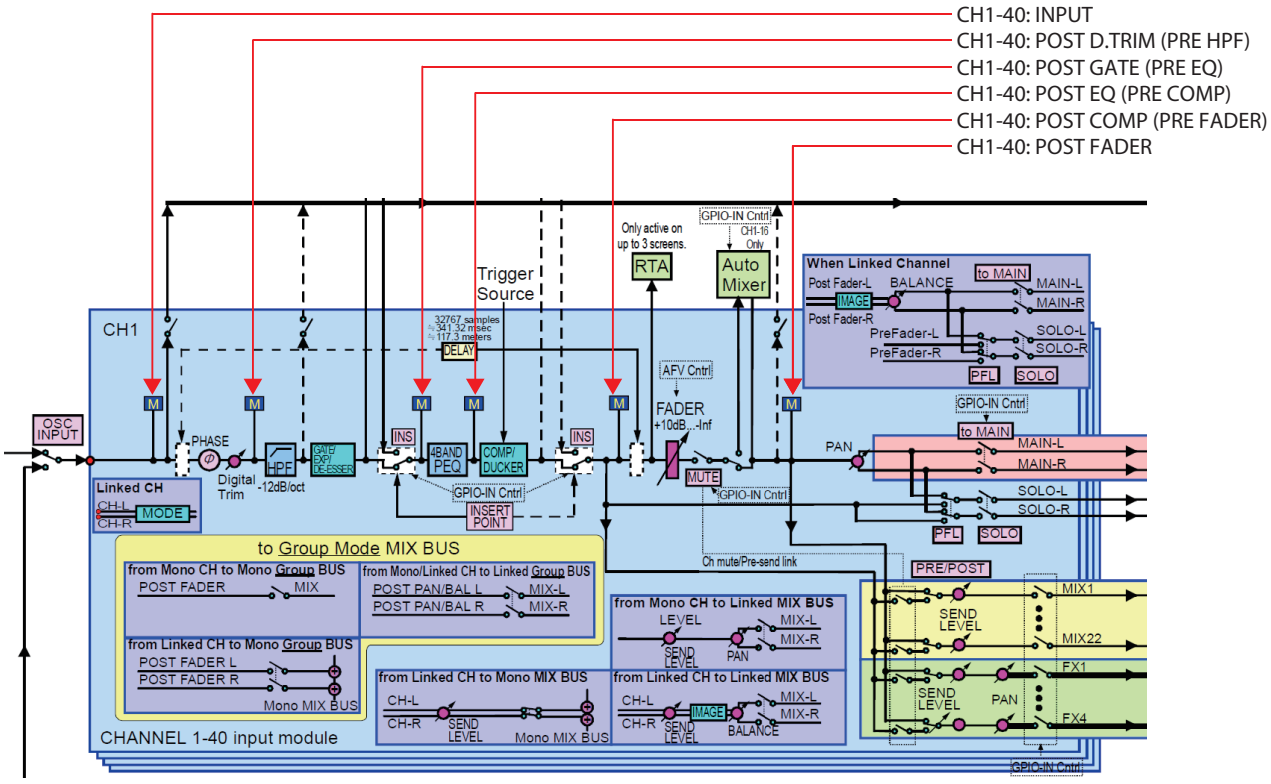
Value	Explanation
0x0114	MIX1-22: OUTPUT peak hold value
0x0115	Main: INPUT (PRE EQ) peak hold value
0x0116	Main: POST EQ (PRE GEQ) peak hold value
0x0117	Main: POST GEQ (PRE COMP) peak hold value
0x0118	Main: POST DELAY (PRE FADER) peak hold value
0x0119	Main: POST FADER peak hold value
0x011A	Main: OUTPUT peak hold value
0x011B	OSC post level peak hold value
0x011C	TB POST D.TRIM peak hold value
0x011D	RTB POST D.TRIM peak hold value
0x011E	MONITOR 1/2 stereo peak hold value
0x011F	Build-in Player peak hold value
0x0120	Build-in Recorder peak hold value
0x0130	CH1-40: GATE KEY-IN peak hold value
0x0132	CH1-40: COMP KEY-IN peak hold value
0x0134	MIX1-22: COMP KEY-IN peak hold value
0x0136	Main: COMP KEY-IN peak hold value
0x0140	Auto Mixer: input/output peak hold value
0x0142	MTR: input monitor/player peak hold value

To change the peak hold time, change the value of "sonicview/meter_setup/peak_hold_time" using Ember+.
Note that some items have no peak hold value.

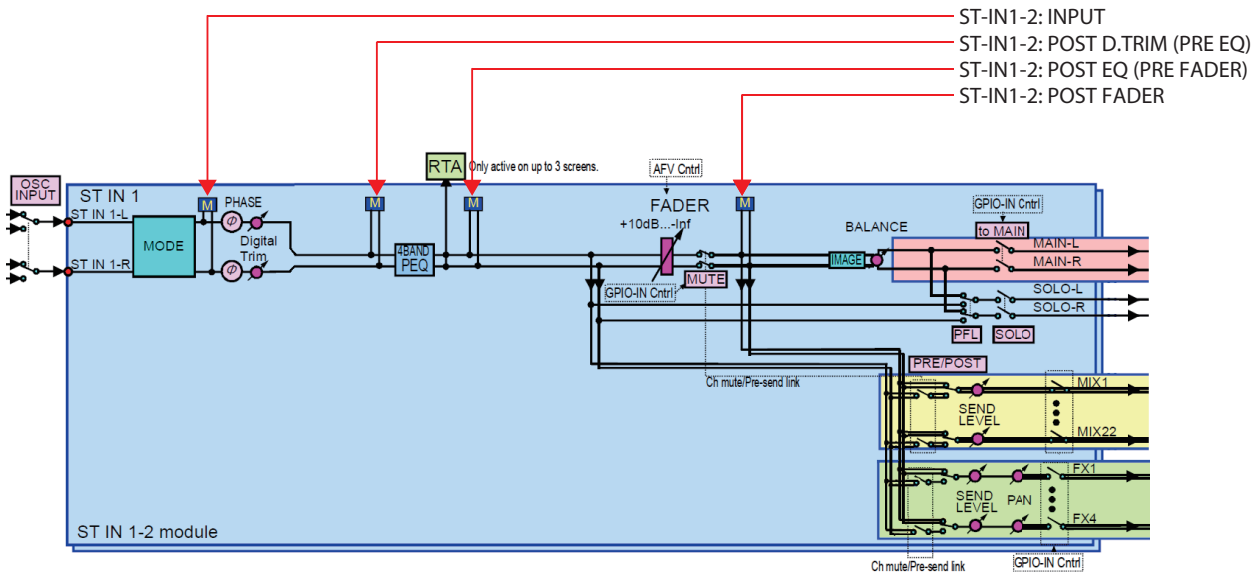
Tags and metering points

A module has multiple metering points, making it possible to retrieve each meter value.
Below is a diagram taken from the block diagram, showing the relationships between tags and metering points.

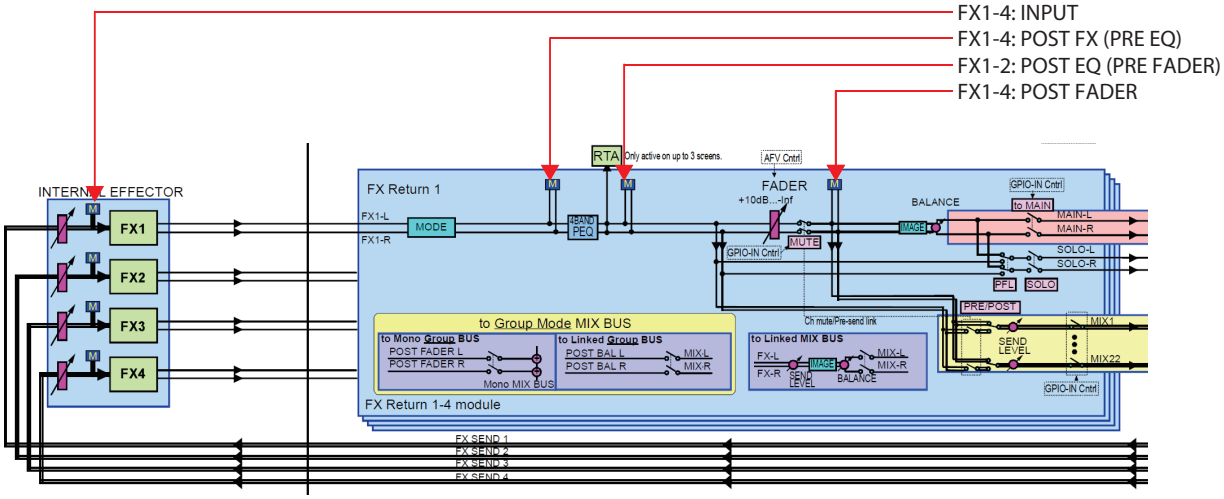
CH module



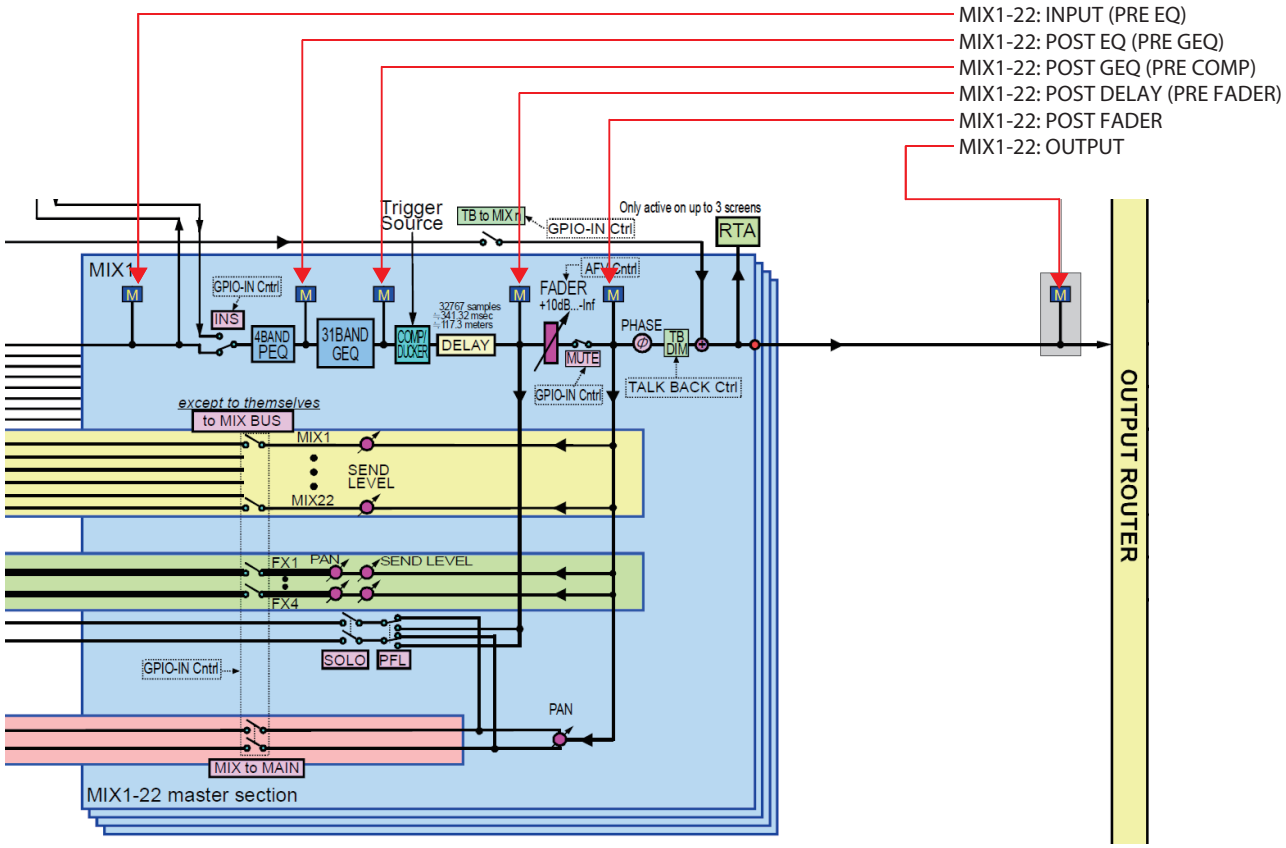
ST-IN module



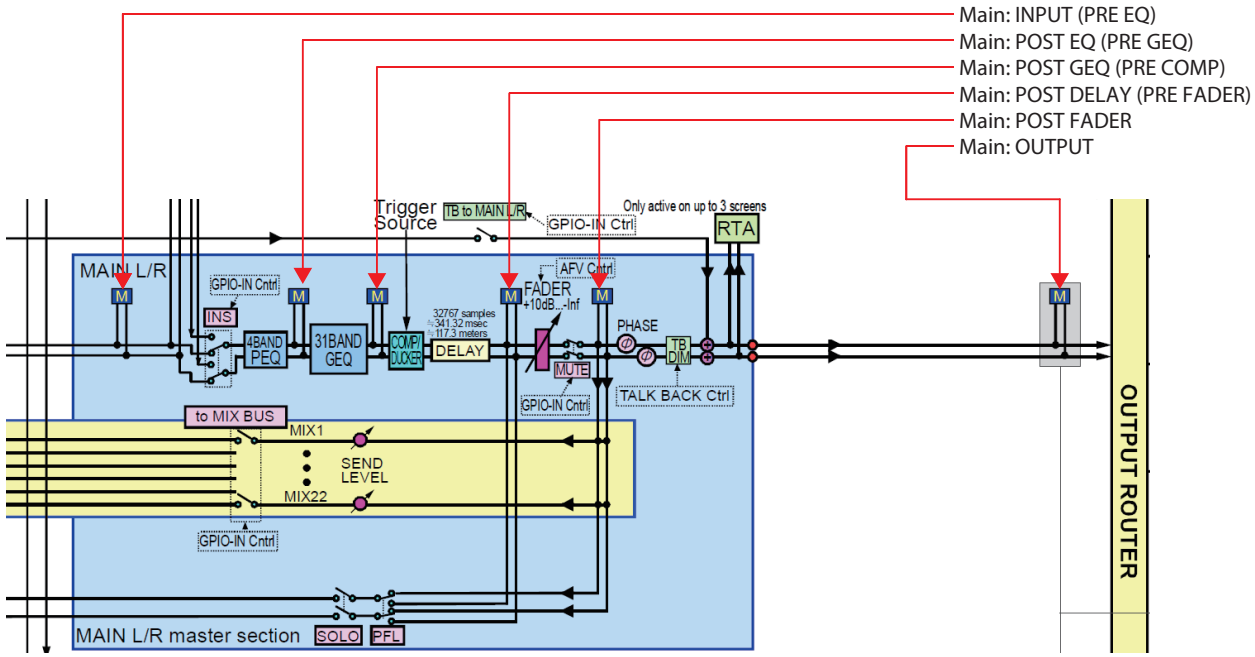
FX module



MIX module



Main L/R module



Converting to dB

The meter data has a 2-byte value ranging from 0 to 0x7fff. Use the application to convert it to dB.
Below is an example in C language.

```
#include <stdint.h>
#define METER_LVL_MIN               (-100.0)
// By setting the argument to int16_t type, limit the maximum value to 0x7fff.
double to_dB(int16_t value_hex)
{
  static const double log10_0 = log(10.0);
  double dB = METER_LVL_MIN;
  if (value_hex > 0)
  {
    dB = 20.0 * (log((double)value_hex / (double)0x7fff)) / log10_0;
  }
  return dB;
}
```

Meter Indexes and Channel IDs

Channel ID

CH1	0x0000
CH2	0x0001
CH3	0x0002
CH4	0x0003
CH5	0x0004
CH6	0x0005
CH7	0x0006
CH8	0x0007
CH9	0x0008
CH10	0x0009
CH11	0x000A
CH12	0x000B
CH13	0x000C
CH14	0x000D
CH15	0x000E
CH16	0x000F
CH17	0x0010
CH18	0x0011
CH19	0x0012
CH20	0x0013
CH21	0x0014
CH22	0x0015
CH23	0x0016
CH24	0x0017
CH25	0x0018
CH26	0x0019
CH27	0x001A
CH28	0x001B
CH29	0x001C
CH30	0x001D
CH31	0x001E
CH32	0x001F
CH33	0x0020

Meter Index

CH1	0x0000
CH2	0x0001
CH3	0x0002
CH4	0x0003
CH5	0x0004
CH6	0x0005
CH7	0x0006
CH8	0x0007
CH9	0x0008
CH10	0x0009
CH11	0x000A
CH12	0x000B
CH13	0x000C
CH14	0x000D
CH15	0x000E
CH16	0x000F
CH17	0x0010
CH18	0x0011
CH19	0x0012
CH20	0x0013
CH21	0x0014
CH22	0x0015
CH23	0x0016
CH24	0x0017
CH25	0x0018
CH26	0x0019
CH27	0x001A
CH28	0x001B
CH29	0x001C
CH30	0x001D
CH31	0x001E
CH32	0x001F
CH33	0x0020

Channel ID

CH34	0x0021
CH35	0x0022
CH36	0x0023
CH37	0x0024
CH38	0x0025
CH39	0x0026
CH40	0x0027
STIN1	0x0100
STIN2	0x0101
FXRTN1	0x0200
FXRTN2	0x0201
FXRTN3	0x0202
FXRTN4	0x0203
MIX1	0x0300
MIX2	0x0301
MIX3	0x0302
MIX4	0x0303
MIX5	0x0304
MIX6	0x0305
MIX7	0x0306
MIX8	0x0307
MIX9	0x0308
MIX10	0x0309
MIX11	0x030A
MIX12	0x030B
MIX13	0x030C
MIX14	0x030D
MIX15	0x030E
MIX16	0x030F
MIX17	0x0310
MIX18	0x0311
MIX19	0x0312
MIX20	0x0313
MIX21	0x0314
MIX22	0x0315
Main	0x0400

Meter Index

CH34	0x0021
CH35	0x0022
CH36	0x0023
CH37	0x0024
CH38	0x0025
CH39	0x0026
CH40	0x0027
STIN1 L	0x0100
R	0x0101
STIN2 L	0x0102
R	0x0103
FXRTN1 L	0x0200
R	0x0201
FXRTN2 L	0x0202
R	0x0203
FXRTN3 L	0x0204
R	0x0205
FXRTN4 L	0x0206
R	0x0207
MIX1	0x0300
MIX2	0x0301
MIX3	0x0302
MIX4	0x0303
MIX5	0x0304
MIX6	0x0305
MIX7	0x0306
MIX8	0x0307
MIX9	0x0308
MIX10	0x0309
MIX11	0x030A
MIX12	0x030B
MIX13	0x030C
MIX14	0x030D
MIX15	0x030E
MIX16	0x030F
MIX17	0x0310
MIX18	0x0311
MIX19	0x0312
MIX20	0x0313
MIX21	0x0314
MIX22	0x0315
Main L	0x0400
R	0x0401

Note that the Meter indexes and Channel IDs for STIN, FXRTN, and Main are different.

Actual data section

The data contents for pages 1 to 4 when the packet version is 0x0000 are shown below.

Page: 0x0001, Data section size: 1014 bytes

Item		bytes	Contents
tag	CH1-40: INPUT	2	0x0001
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: POST D.TRIM (PRE HPF)	2	0x0002
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: POST GATE (PRE EQ)	2	0x0003
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: POST EQ (PRE COMP)	2	0x0004
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: POST COMP (PRE FADER)	2	0x0005
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: POST FADER	2	0x0006
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	ST-IN 1-2: INPUT	2	0x0007
size		2	0x0004
payload	4 pieces of meter data (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	ST-IN 1-2: POST D.TRIM (PRE EQ)	2	0x0008
size		2	0x0004
payload	4 pieces of meter data (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	ST-IN 1-2: POST EQ (PRE FADER)	2	0x0009
size		2	0x0004
payload	4 pieces of meter data (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	ST-IN 1-2: POST FADER	2	0x000A
size		2	0x0004
payload	4 pieces of meter data (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	FX 1-4: INPUT	2	0x000B
size		2	0x0010
payload	8 pieces of meter data (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	FX 1-4: POST FX (PRE EQ)	2	0x000C
size		2	0x0010
payload	8 pieces of meter data (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	FX 1-4: POST EQ (PRE FADER)	2	0x000D
size		2	0x0010
payload	8 pieces of meter data (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	FX 1-4: POST FADER	2	0x000E
size		2	0x0010
payload	8 pieces of meter data (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	MIX1-22: INPUT (PRE EQ)	2	0x000F
size		2	0x002c

Item		bytes	Contents
payload	22 pieces of meter data	44	<44-byte long data>
tag	MIX1-22: POST EQ (PRE GEQ)	2	0x0010
size		2	0x002c
payload	22 pieces of meter data	44	<44-byte long data>
tag	MIX1-22: POST GEQ (PRE COMP)	2	0x0011
size		2	0x002c
payload	22 pieces of meter data	44	<44-byte long data>
tag	MIX1-22: POST DELAY (PRE FADER)	2	0x0012
size		2	0x002c
payload	22 pieces of meter data	44	<44-byte long data>
tag	MIX1-22: POST FADER	2	0x0013
size		2	0x002c
payload	22 pieces of meter data	44	<44-byte long data>
tag	MIX1-22: OUTPUT	2	0x0014
size		2	0x002c
payload	22 pieces of meter data	44	<44-byte long data>
tag	Main: INPUT (PRE EQ)	2	0x0015
size		2	0x0004
payload	2 pieces of meter data (Main L, Main R)	4	<4-byte long data>
tag	Main: POST EQ (PRE GEQ)	2	0x0016
size		2	0x0004
payload	2 pieces of meter data (Main L, Main R)	4	<4-byte long data>
tag	Main: POST GEQ (PRE COMP)	2	0x0017
size		2	0x0004
payload	2 pieces of meter data (Main L, Main R)	4	<4-byte long data>
tag	Main : POST DELAY (PRE FADER)	2	0x0018
size		2	0x0004
payload	2 pieces of meter data (Main L, Main R)	4	<4-byte long data>
tag	Main: POST FADER peak	2	0x0019
size		2	0x0004
payload	2 pieces of meter data (Main L, Main R)	4	<4-byte long data>
tag	Main: OUTPUT peak	2	0x001A
size		2	0x0004
payload	2 pieces of meter data (Main L, Main R)	4	<4-byte long data>
tag	OSC post level	2	0x001B
size		2	0x0002
payload	1 piece of meter data	2	<2-byte long data>
tag	TB POST D.TRIM	2	0x001C
size		2	0x0002
payload	1 piece of meter data	2	<2-byte long data>
tag	RTB POST D.TRIM	2	0x001D
size		2	0x0008
payload	1 piece of meter data	2	<2-byte long data>
tag	MONITOR 1/2 stereo	2	0x001E
size		2	0x0004
payload	4 pieces of meter data (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	Built-in player	2	0x001F
size		2	0x0004

Item		bytes	Contents
payload	2 pieces of meter data (L, R)	4	<4-byte long data>
tag	Built-in recorder	2	0x0020
size		2	0x0004
payload	2 pieces of meter data (L, R)	4	<4-byte long data>

Page: 0x0002, Data section size: 1020 bytes

Item		bytes	Contents
tag	CH1-40: GATE KEY-IN	2	0x0030
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: GATE GR	2	0x0031
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: COMP KEY-IN	2	0x0032
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	CH1-40: COMP GR	2	0x0033
size		2	0x0050
payload	40 pieces of meter data	80	<80-byte long data>
tag	MIX1-22: COMP KEY-IN	2	0x0034
size		2	0x002c
payload	22 pieces of meter data	44	<44-byte long data>
tag	MIX1-22: COMP GR	2	0x0035
size		2	0x002c
payload	22 pieces of meter data	44	<44-byte long data>
tag	Main: COMP KEY-IN	2	0x0036
size		2	0x0004
payload	2 pieces of meter data (Main L, Main R)	4	<2-byte long data>
tag	Main: COMP GR	2	0x0037
size		2	0x0002
payload	2 pieces of meter data (Main L, Main R)	4	<4-byte long data>
tag	Auto Mixer: input/output	2	0x0040
size		2	0x0020
payload	16 pieces of meter data	32	<32-byte long data>
tag	Auto Mixer: GAIN	2	0x0041
size		2	0x0020
payload	16 pieces of meter data	32	<32-byte long data>
tag	MTR: input monitor/player	2	0x0042
size		2	0x0040
payload	32 pieces of meter data	64	<64-byte long data>
tag	RTA 1L Meter index	2	0x0050
size		2	0x0002
payload	Meter index	2	<2-byte long data>
tag	RTA 1L data	2	0x0051
size		2	0x003e
payload	31 pieces of RTA data	62	<62-byte long data>
tag	RTA 1R Meter index	2	0x0052

Item		bytes	Contents
size		2	0x0002
payload	Meter index	2	<2-byte long data>
tag	RTA 1R data	2	0x0053
size		2	0x003e
payload	31 pieces of RTA data	62	<62-byte long data>
tag	RTA 2L Meter index	2	0x0054
size		2	0x0002
payload	Meter index	2	<2-byte long data>
tag	RTA 2L data	2	0x0055
size		2	0x003e
payload	31 pieces of RTA data	62	<62-byte long data>
tag	RTA 2R Meter index	2	0x0056
size		2	0x0002
payload	Meter index	2	<2-byte long data>
tag	RTA 2R data	2	0x0057
size		2	0x003e
payload	31 pieces of RTA data	62	<62-byte long data>
tag	RTA 3L Meter index	2	0x0058
size		2	0x0002
payload	Meter index	2	<2-byte long data>
tag	RTA 3L data	2	0x0059
size		2	0x003e
payload	31 pieces of RTA data	62	<62-byte long data>
tag	RTA 3R Meter index	2	0x005A
size		2	0x0002
payload	Meter index	2	<2-byte long data>
tag	RTA 3R data	2	0x005B
size		2	0x003e
payload	31 pieces of RTA data	62	<62-byte long data>

page: 0x0003, Data section size: 1014 bytes

Item		bytes	Contents
tag	CH1-40: INPUT peak	2	0x0101
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>
tag	CH1-40: POST D.TRIM (PRE HPF) peak	2	0x0102
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>
tag	CH1-40: POST GATE (PRE EQ) peak	2	0x0103
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>
tag	CH1-40: POST EQ (PRE COMP) peak	2	0x0104
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>
tag	CH1-40:POST COMP (PRE FADER) peak	2	0x0105
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>

Item		bytes	Contents
tag	CH1-40: POST FADER peak	2	0x0106
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>
tag	ST-IN 1-2: INPUT peak	2	0x0107
size		2	0x0004
payload	4 peak hold values (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	ST-IN 1-2: POST D.TRIM (PRE EQ) peak	2	0x0108
size		2	0x0004
payload	4 peak hold values (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	ST-IN 1-2: POST EQ (PRE FADER) peak	2	0x0109
size		2	0x0004
payload	4 peak hold values (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	ST-IN 1-2: POST FADER peak	2	0x010A
size		2	0x0004
payload	4 peak hold values (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	FX 1-4: INPUT peak	2	0x010B
size		2	0x0010
payload	8 peak hold values (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	FX 1-4: POST FX (PRE EQ) peak	2	0x010C
size		2	0x0010
payload	8 peak hold values (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	FX 1-4: POST EQ (PRE FADER) peak	2	0x010D
size		2	0x0010
payload	8 peak hold values (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	FX 1-4: POST FADER peak	2	0x010E
size		2	0x0010
payload	8 peak hold values (1L, 1R, 2L, 2R, 3L, 3R, 4L, 4R)	16	<16-byte long data>
tag	MIX1-22: INPUT (PRE EQ) peak	2	0x010F
size		2	0x002c
payload	22 peak hold values	44	<44-byte long data>
tag	MIX1-22: POST EQ (PRE GEQ) peak	2	0x0110
size		2	0x002c
payload	22 peak hold values	44	<44-byte long data>
tag	MIX1-22: POST GEQ (PRE COMP) peak	2	0x0111
size		2	0x002c
payload	22 peak hold values	44	<44-byte long data>
tag	MIX1-22: POST DELAY (PRE FADER) peak	2	0x0112
size		2	0x002c
payload	22 peak hold values	44	<44-byte long data>
tag	MIX1-22: POST FADER peak	2	0x0113
size		2	0x002c
payload	22 peak hold values	44	<44-byte long data>
tag	MIX1-22: OUTPUT peak	2	0x0114
size		2	0x002c
payload	22 peak hold values	44	<44-byte long data>
tag	Main: INPUT (PRE EQ) peak	2	0x0115
size		2	0x0004
payload	2 peak hold values (Main L, Main R)	4	<4-byte long data>

Item		bytes	Contents
tag	Main: POST EQ (PRE GEQ) peak	2	0x0116
size		2	0x0004
payload	2 peak hold values (Main L, Main R)	4	<4-byte long data>
tag	Main: POST GEQ (PRE COMP) peak	2	0x0117
size		2	0x0004
payload	2 peak hold values (Main L, Main R)	4	<4-byte long data>
tag	Main: POST DELAY (PRE FADER) peak	2	0x0118
size		2	0x0004
payload	2 peak hold values (Main L, Main R)	4	<4-byte long data>
tag	Main: POST FADER peak	2	0x0119
size		2	0x0004
payload	2 peak hold values (Main L, Main R)	4	<4-byte long data>
tag	Main: OUTPUT peak	2	0x011A
size		2	0x0004
payload	2 peak hold values (Main L, Main R)	4	<4-byte long data>
tag	OSC post level peak	2	0x011B
size		2	0x0002
payload	1 peak hold value	2	<2-byte long data>
tag	TB POST D.TRIM peak	2	0x011C
size		2	0x0002
payload	1 peak hold value	2	<2-byte long data>
tag	RTB POST D.TRIM peak	2	0x011D
size		2	0x0002
payload	1 peak hold value	2	<2-byte long data>
tag	MONITOR 1/2 stereo peak	2	0x011E
size		2	0x0008
payload	4 peak hold values (1L, 1R, 2L, 2R)	8	<8-byte long data>
tag	Build-in Player peak	2	0x011F
size		2	0x0004
payload	2 peak hold values (L, R)	4	<4-byte long data>
tag	Build-in Recorder peak	2	0x0120
size		2	0x0004
payload	2 peak hold values (L, R)	4	<4-byte long data>

Page: 0x0004, Data section size 328 bytes

Item		bytes	Contents
tag	CH1-40: GATE KEY-IN peak	2	0x0130
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>
tag	CH1-40: COMP KEY-IN peak	2	0x0132
size		2	0x0050
payload	40 peak hold values	80	<80-byte long data>
tag	MIX1-22: COMP KEY-IN peak	2	0x0134
size		2	0x002c
payload	22 peak hold values	44	<44-byte long data>
tag	Main: COMP KEY-IN peak	2	0x0136
size		4	0x0004

payload	2 peak hold values (Main L, Main R)	2	<2-byte long data>
tag	Auto Mixer: input/output peak	2	0x0140
size		2	0x0020
payload	16 peak hold values	32	<32-byte long data>
tag	MTR: input monitor/player peak	2	0x0142
size		2	0x0040
payload	32 peak hold values	64	<64-byte long data>

CRC16 sample program

```
static const uint16_t crc16_table[256] = {
  0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
  0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,
  0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,
  0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,
  0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
  0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,
  0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,
  0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,
  0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,
  0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
  0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,
  0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,
  0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,
  0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,
  0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
  0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,
  0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,
  0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,
  0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,
  0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
  0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,
  0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
  0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,
  0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,
  0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
  0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,
  0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,
  0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,
  0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,
  0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
  0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,
  0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};

uint16_t crc16( const uint8_t *buf, uint32_t len )
{
  uint16_t crc = 0;
  uint32_t n;
  for (n = 0; n < len; n++){
    crc = crc16_table[(crc >> 8) ^ buf[n]] ^ (crc << 8);
  }
  return crc;
}
```

DATE	Version	CONTENTS
2025/6/11	1.0	First edition published