EPICS Device Support
For SL1000 Digitizer Modules

Version 1.1

Hajime Nakamura and Takashi Asakawa
Yokogawa Electric Corporation, Japan

Joseph Ting
Yokogawa Corporation of America, USA

25 May 2009

License Agreement

This product is available via the open source license described at the end of this document.

Contents

1. SL1000 Series and Supported Modules

The *SL1000* is a high-performance data acquisition unit featuring fast data acquisition, transfer, and storage capabilities. It is a module-based instrument with a wide and varied module lineup. We have developed device support for waveform digitizer modules within the *SL1000* series. The main specifications of these devices are summarized in Tables1 and 2. For more details, please visit the Yokogawa Web site or see the product manuals.

Figure 1: The *SL1000* Data Acquisition Unit (left) and the *720210* 100MS/s Digitizer Module (right)

Main Specifications	Description
Number of Slots	8
Max. No. of Channels	16 (2 channels x 8 slots)
Max. Sampling Rate	100MHz
Acquisition Memory	128MP
Ethernet	1000BASE-T
Dimensions	319 mm(W) x 154 mm (D)x 350 mm (D)
Weight	Approx. 6 kg (SL1000 unit only)

Table 1: Main specifications of the SL1000 data acquisition unit.

Model	Type	No. of Channels	Sampling Rate	Bandwidth	Resolution	Isolation
720210	digitizer	2	100MS/s	20MHz	12 bits	isolated
701250	digitizer	2	10MS/s	3MHz	12 bits	isolated
701251	digitizer	2	1MS/s	300kHz	16 bits	isolated
701255	digitizer	2	10MS/s	3MHz	12 bits	non-isolated
701260	digitizer	2	100kS/s	40kHz	16 bits	isolated

Table 2: Supported modules of the SL1000 Series

2. Device Support Details

The SL1000 employs the VXI-11 protocol, and I/O commands for controlling the device are fully supported by asynDriver.

We used a PC with the Linux operating system (CentOS 5) and the EPICS base (version R3.14.9) with asynDriver (version 4.11, beta version) in developing our device support.

A. Key Features

The device driver supports the following key features of the SL1000 series:

- SRQ Function
- Acquisition of Compressed Data
- Data Storage of Historical Waveforms

SRQ Function

The SRQ function is supported. At present (May, 2009) the Asyn driver in the CVS repository at ANL is required. Hopefully, the next version (Asyn4.11) will officially support the function.

Acquisition of Compressed Data

The SL1000 stores both raw data and compressed data in the device (See Figure 2). The data size of the compressed data is fixed to 2000 points; it does not depend on the record length. Compressed data is made by the peak-to-peak compression technique. For example, if the record length is 1M points, only a maximum and minimum pair is stored out of every 1000 points of raw data, resulting in 2000 points (1000 pairs) of data. The device support further compresses the transferred 2k point data to a pre-determined number of points. The default size is 1000 points (500 pairs), which is defined in "`st.cmd`" using the environment variable of "`DISPWF_NELM`". A client can also change the data size by selecting a preferable value using the record "`dispWavePointsMO`".

It is very beneficial to use the compressed data in displaying waveforms in a screen. In most cases, 1k points of data would be enough for display use because of the limited resolution of displays, while we can highly reduce the network traffic load by using compressed data.

Since raw data is always available in the device memory, for example, when some anomaly in a signal is detected, we can use the raw data for precise analysis of the phenomenon. Note that we should stop the acquisition before accessing raw data; if the acquisition is resumed, the device memory might be overwritten by newly acquired data.

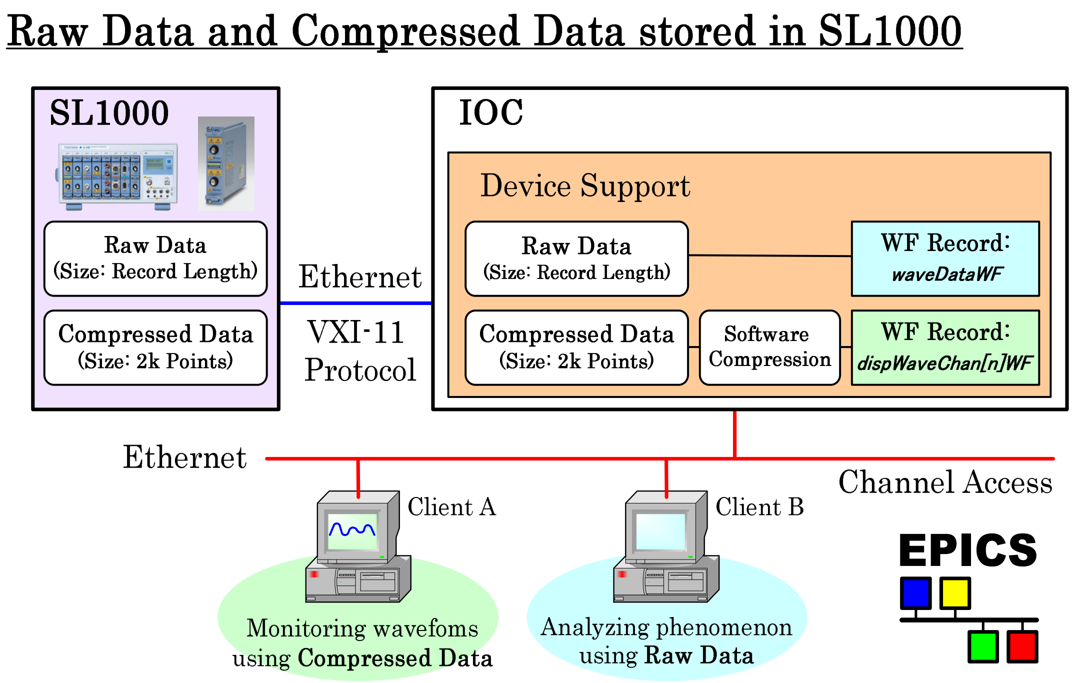


Figure 2: The SL1000 stores raw data and compressed data.
The device support is designed to access either type of data.

Data Storage of Historical Waveforms

Since the SL1000 is equipped with a large memory of 128MP, multiple waveforms can be stored. The maximum number of waveforms that can be stored depends on the number of channels employed and the record length as summarized in Table 3.

In our device support, two types of trigger numbers are handled. One is a trigger number, which is the number incremented by providing a trigger signal. This number is reset when the acquisition conditions (e.g. the record length, the number of channels, the sampling rate, and the trigger parameters) are changed or by providing a reset command. The other is a relative trigger number, called "History Number" hereafter. The history number for the most recent waveform acquired is treated as the starting point (zero), and the number is defined to be zero or negative. For example, the value "-1" corresponds to the waveform that is previous to the most recent waveform stored in the device.

When we have access to a certain waveform in the device, either of the trigger number or the history number must be specified (See Figure 3).

Record Length	Number of Channels				
	1	2	3, 4	5 to 8	9 to 16
2k	5000	5000	5000	5000	3275
5k	5000	5000	5000	3275	1637
10k	5000	5000	3275	1637	818
20k	5000	2620	1309	654	326
50k	2620	1309	654	326	162
100k	1309	654	326	162	80
200k	523	261	130	64	31
500k	261	130	64	31	15
1M	127	63	31	15	7

Table 3: The maximum number of waveforms that can be stored in the device.

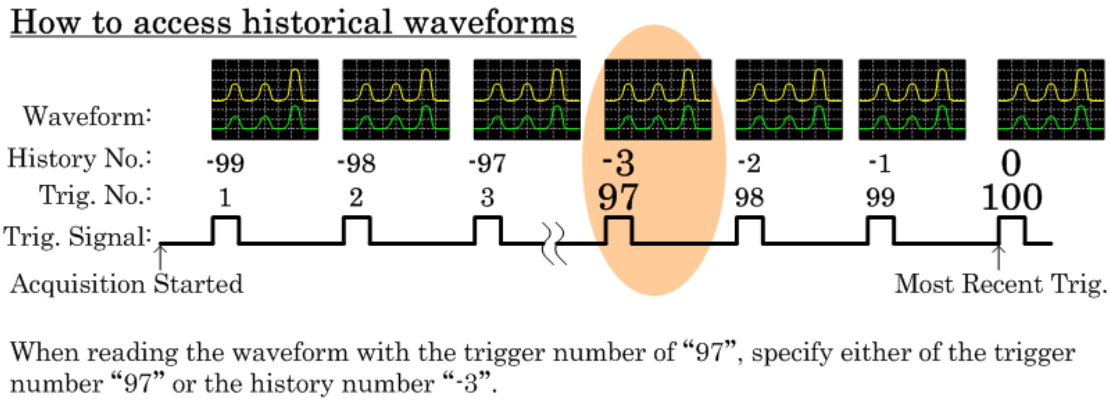
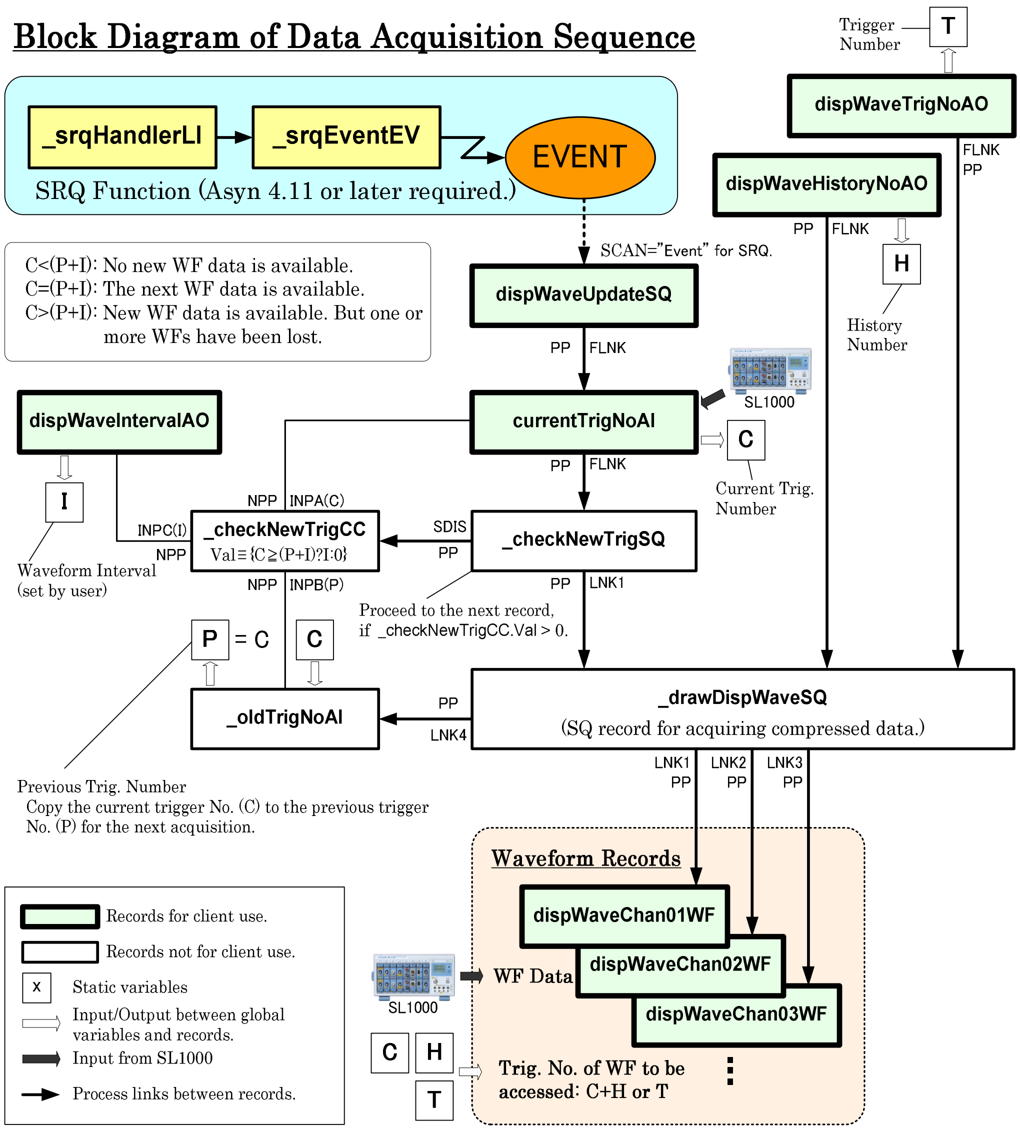


Figure 3: How to have access to a certain waveform.

B. Data Acquisition Sequence

Figure 4 shows the data acquisition sequence with this device support. When the SRQ function is used, the "SCAN" field of the record "dispWaveUpdateSQ" is set to be "Event". The record "currentTrigNoAI" gets the current trigger number from the device and stores the value in a static variable ("C"). The record "_checkNewTrigSQ" compares the current trigger number "C" and the previous trigger number (another static variable "P"), and the waveform records are processed only when the difference between the two values ("C−P") is greater than a pre-determined value "I" (=waveform interval, set with "dispWaveIntervalAI"). The value "C" is then copied into "P" for the next acquisition. In this way, the waveform records are processed only when waveform data to be transferred are available.



I

Figure 4: The data acquisition sequence.

C. Performance

We have evaluated the performance of the SL1000 and our device support using our MEDM viewer tool. We have used an SL1000 unit with 8 digitizer modules (=16 channels) and a Linux PC. The SL1000 unit and the PC are connected through a G-bit hub (see Figure 5). Both the IOC and the sample MEDM viewer run on the same machine. The transferred data are of the compressed waveforms only. With this test we have changed (1) the sampling rate, (2) the record length, and (3) the trigger rate as evaluation parameters, and have recorded the maximum number of channels with which the system works properly. The judgment whether the system works properly or not is made by:

i) checking the value "C−P", where "C" and "P" are defined in Figure 4. When the value "C−P" is equal to the preset value of "I", the device support works properly, and when the value is greater than "I", it misses some WF data to be transferred (some WF could not be transferred to IOC).

ii) checking if the trigger number is incremented as provided. (E.g. we send 500 pulses to the SL1000 from a signal generator and see if the SL1000 can acquire 500 waveforms.)

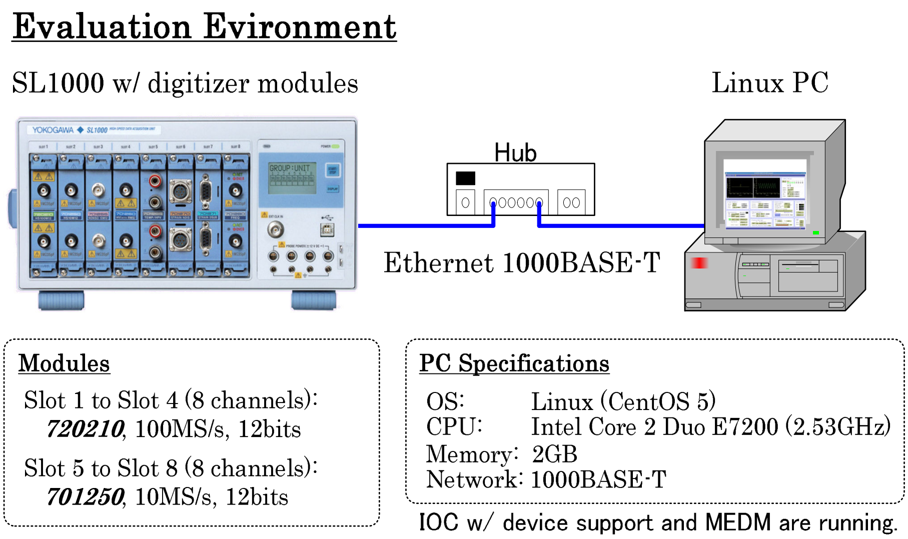


Figure 5: The evaluation environment. The SL1000 and the PC are connected using a G-bit hub. The IOC with the device support and the MEDM viewer are running on the same PC.

Table 4 summarizes some acquisition conditions in which the system works properly without missing any data. Even though the size of data transferred per channel does not depend on the record length (it is fixed by the data compression), the performance limit is affected by the record length and the number of channels, and the dependency on the record length is not simple but somewhat complicated (see Table 4). This is because the SL1000 is optimized for transferring long data (e.g. 500k and 1M).

We should note that this test has been performed with an ideal condition, while the total performance depends on the device, the device support, the PC, and the network condition. These results do not warrant the performance.

Sampling Rate	Record Length	Trigger Rate (Waveform Interval)			
		50Hz (1)	50Hz (5)	25Hz (1)	12.5Hz (1)
100MS/s	1M	16	16	16	16
	500k	16	16	16	16
	200k	2	16	6	12
	100k	4	16	10	16
	50k	7	16	16	16
	20k	12	16	16	16
	10k	14	16	16	16
10MS/s	500k				16
	200k			16	16
	100k	16	16	16	16
	50k	16	16	16	16
	20k	12	16	16	16
	10k	14	16	16	16
	5k	16	16	16	16
	2k	16	16	16	16
1MS/s	50k				16
	20k			16	16
	10k	14	16	16	16
	5k	16	16	16	16
	2k	16	16	16	16
100kS/s	5k				16
	2k			16	16

Note 1: The SRQ function is enabled (SCAN is set to "Event".).

Note 2: This test is for transferring compressed data (dispWaveChan[n]WF).

Note 3: When "Waveform Interval=N", a waveform data is transferred for every N-waveforms.

Table 4: The performance test results. The maximum numbers of channels with which the system works properly are summarized. The results do not warrant the performance.

3. Record List

A. Acquisition

Record	Description	Value
startBO	Start acquisition.	
stopBO	Stop acquisition.	
manualTrigBO	Manually execute trigger action.	

B. Trigger Number

Comments:		
When specifying a certain waveform, use "trigger number" or "history number". The maximum number of waveforms which can be stored in the device depends on the number of enabled channels and the record length. The number is automatically set to the record "***maxHistorySizeAI***" according to a given condition.		
Record	**Description**	**Value**
currentTrigNoAI	Read current trigger number.	{1 to 2^52}
maxHistorySizeAI	Read maximum number of wave-forms secured for historical data.	{1 to 5000}
resetTrigNoBO	Reset trigger number. All data stored in the device memory is deleted with this action.	

C. Acquisition Condition

Comments:		
These are records for the acquisition condition and are common to all channels. The acceptable combinations of record lengths and sampling rates are shown in the following table(right):	*Sampling Rate*: 1kHz to 20MHz; 50MHz; 100MHz	*Record Length (Points)*: 2k to 1M; 5k to 1M; 10k to 1M
Record	**Description**	**Value**
acqModeBO	Select acquisition mode.	{"Repeat" | "Single"}
acqModeBI	Read acquisition mode.	
recLenMO	Select record length of waveform data.	{"2k"|"5k"|"10k"|"20k"|…|"500k"|"1M"}
recLenMI	Read record length of waveform data.	e.g. "10k"
recLenAI	Read record length of waveform data.	e.g. 10,000 for "10k"

Sampling Rate	Record Length (Points)
1kHz to 20MHz	2k to 1M
50MHz	5k to 1M
100MHz	10k to 1M

smplRateAMO	Select sampling rate (value part). The unit is set by “smplRateBMO”.	{“1”|”2”|”5”| … |”200”| “500”}
smplRateBMO	Select sampling rate (unit part). The value is set by ”smplRateAMO”.	{ “Hz” | “kHz” | “MHz” }
smplRateAMI	Read sampling rate (value part).	e.g. “100” for 100kHz
smplRateBMI	Read sampling rate (unit part).	e.g. “kHz” for 100kHz
smplRateAI	Read sampling rate in Hz.	e.g. 100,000 for “100”+”kHz”
trigDelayAO	Set trigger delay time in seconds.	{0 to 10, 10 ns step}
trigDelayAI	Read trigger delay time in seconds.	
trigHoldoffAO	Set trigger holdoff time in seconds.	{0 to 10, 10 ns step}
trigHoldoffAI	Read trigger holdoff time in seconds.	
trigLevAO	Set trigger level in volts.	{ (-chanVdivMO)*10 to (chanVdivMO)*10 }
trigLevAI	Read trigger level in volts.	
trigPosMO	Select trigger position.	{“0%” | “10%” | “20%” | “30%”| … | “90%” | “100%” }
trigPosMI	Read trigger position.	e.g. “30%”
trigPosAI	Read trigger position.	e.g. 30 for “30%”
trigSlopeMO	Select trigger slope.	{ “RISE” | “FALL” }
trigSlopeMI	Read trigger slope.	
trigSourceMO	Select trigger source.	{ “EXT” | “LINE” | “CH1” | “CH2” | … | “CH16” }
trigSourceMI	Read trigger source.	

D. Channel Setting

Comments:		
These records are for channel settings. Select a target channel with “***chanNoSelectMO***” in advance.		
Record	**Description**	**Value**
maxChanNumAI	Read maximum channel number. The number of channels available in the device is automatically detected when IOC starts up.	{ 0 to 16 }
chanNoSelectMO	Select target channel.	{ “CH1” | “CH2” | … | “CH16” }
chanNoSelectMI	Read target channel number.	e.g. “CH1”
chanEnableBO	Set On/Off status of selected channel.	{ “Off” | “On” }

	The default value is “Off”.	
chanEnableBI	Read On/Off status of selected channel.	
Chan[n]EnableBI	Read On/Off status of Channel [n]. ([n]: “01”, “02”, …, “16”)	{ “Off” | “On” }
chanCoupleMO	Select coupling type of selected channel.	{ “AC” | “DC” | “GND” }
chanCoupleMI	Read coupling type of selected channel.	
chanProbeMO	Select probe condition of selected channel.	{ “1:1” | “10:1” | “100:1” | “1000:1” }
chanProbeMI	Read probe condition of selected channel.	
chanVdivMO	Select voltage per division of selected channel. Note1: The setting range depends on the probe setting: e.g. Model 720210 i) 10mv to 50V for probe=1:1 ii) 100mV to 100V for probe=10:1 iii) 1V to 100V for probe=100:1 iv) 10V to 100V for probe=1000:1 Note2: Actual measurement range is given by: -10×(Vdiv) to +10×(Vdiv).	{ “10mV” | “20mV” | “50mV” | … | “1000V” }
chanVdivMI	Read voltage per division of selected channel.	e.g. “20mV”
chanVdivAI	Read voltage per division of selected channel.	e.g. 0.02 for “20mV”

E. Current Value Acquisition

Comments:		
The device has a function measuring current voltage values.		
Record	**Description**	**Value**
currValUpdateSQ	Process sequence of current value measurement.	
currValChan[n]AI	Read current voltage of channel [n] in	e.g. 1.5 for 1.5 volts

	volts. ([n]: “01”, “02”, …, “16”)	

F. Compressed Waveform Data Acquisition

Comments:
The records whose names begin with “***dispWave***” are for transferring compressed waveform data. The device support further compresses the transferred data so that the data length matches with “***dispWavePointsMO***”. This record relates only on the software compression rate and does not change the time length of the waveform. Set this value by taking into account the display resolution and the network traffic condition. The data transfer is performed by processing the record “***dispwaveUpdateSQ***”. The waveform data will be stored in records “***dispWaveChan[n]WF***” secured for each channel. Specify a trigger number (“***dispWaveTrigNoAO***”) or a history number (“***dispWaveHistoryNoAO***”) in advance.

Record	**Description**	**Value**
dispWaveTrigNoAO	Set trigger number of waveform to be displayed.	{ 1 to -10^11-1 }
dispWaveTrigNoAI	Read trigger number of waveform to be displayed.	
dispWaveHistoryNoAO	Set history number of waveform to be displayed. This is a relative number; the history number for the most recent trigger is treated as a starting point (zero), and the history number is defined as zero or a negative value. The absolute value of this number should be less than the value of “maxHistorySize”.	{ 0 ~ -5000 } e.g. The value -1 corresponds to the waveform previous to the most recent waveform.
dispWaveHistoryNoAI	Read history number of waveform to be displayed.	{ 0 ~ -5000 }
dispWavePointsMO	Set data length of displayed waveform. This is the actual size of the waveform record (“dispWaveChan[n]WF”). If this value is less than 2000 (= the data length of the waveform transferred from the device to IOC), the software data compression is performed. Set this value by taking into	{ “NELM” | “200” | “500” | “1000” | “2000” | “5000” } Note “NELM” is defined by the environment variable of “DISPWF_NELM”. The default value is 1000.

	account the display resolution and the network traffic condition.	
dispWavePointsMI	Read data length of waveform to be displayed.	e.g. "500" e.g. "NELM"
dispWavePointsAI	Read data length of waveform to be displayed.	e.g. 500 for "500" e.g. 1000 for "NELM"
dispWaveTrigNoDiffAI	Read trigger number difference between currently and previously acquired waveforms. =0: No new data available. =1: The next data are ready. >1: New data are available. But some waveforms have been lost.	
dispWaveUpdateSQ	Update displayed waveforms. When SCAN="Event" is selected, waveforms will be updated at the timing of receiving an interrupt signal meaning the end of trigger acquisition.	
dispWaveTimeAxisWF	Time data of waveform to be displayed (in seconds).	
dispWaveChan[n]WF	Waveform data of channel [n]. ([n]: "01", "02", …, "16")	
dispWaveIntervalAO	Interval of waveforms to be transferred to IOC. When this value is *"n"*, waveform data is transferred once in *n*-waveform(s) acquired by the device. When the value is *"1"*, every data is transferred. When the value is *"0"*, no data is transferred.	{ 0 to 10000 }
dispWaveIntervalAI	Read interval of waveforms to be transferred.	

G. Raw Waveform Data Acquisition

Comments:
The records of which names begin with "***wave***" are for the raw data transfer function. The data transfer is performed by processing the record "***waveUpdateSQ***".

When transferring data, select the channel number of interest using "*waveChanNoMO*", specify a trigger number ("*waveTrigNoAO*") or a history number ("*waveHistoryNoAO*"), the data length ("*wavePosAO*"), and the start position ("*wavePosAO*") in the specified waveform. The selected waveform will be stored in the record of "*waveDataWF*".		
Record	**Description**	**Value**
waveChanNoMO	Select channel number of raw waveform data to be transferred.	{ "CH1" | "CH2" | … | "CH16" }
waveChanNoMI	Read channel number of raw waveform data to be transferred.	
waveTrigNoAO	Set trigger number of raw waveform data to be transferred. The record for the history number ("waveHistoryNoAO") is valid when this record is set to "zero". The channel number is defined by "waveChanNoMO".	{ 1 to -10^11-1 }
waveTrigNoAI	Read latest trigger number of raw waveform data. The channel number is defined by "waveChanNoMO".	
waveHistoryNoAO	Set waveform history number of raw waveform data to be transferred. This function is valid when the trigger number is set to "zero". Note: This is a relative number; the most recent trigger number is treated as a starting point (zero), and so the number is defined as zero or a negative value. The absolute value of this number should be less than the value of "maxHistorySize".	{ 0 to -5000 } e.g. The value "-1" corresponds to the waveform previous to the most recent waveform.
waveHistoryNoAI	Read history number of raw waveform data to be transferred. The channel number is defined by "waveChanNoMO".	
waveLenAO	Set data length of raw waveform data	{ 0 to WF_NELM }

	to be transferred. The channel number is defined by "waveChanNoMO".	
waveLenAI	Read data length of raw waveform data to be transferred. The channel number is defined by "waveChanNoMO".	
wavePosAO	Set start position of raw waveform data to be transferred. The channel number is defined by "waveChanNoMO".	{ 0 to "Record Length"-1 }
wavePosAI	Read start position of raw waveform data to be transferred. The channel number is defined by "waveChanNoMO".	
waveUpdateSQ	Update raw waveform data. The channel number is defined by "waveChanNoMO". Note: When SCAN="Event" is selected, the waveform will be updated at the timing of receiving interrupt signal meaning the end of trigger acquisition.	
waveAutoUpdateEnableBO	Set Enable/Disable of auto-update function of raw waveform data.	{ "Disable" | "Enable" }
waveAutoUpdateEnableBI	Read Enable/Disable status of auto-update function of raw waveform data.	
waveTimeAxisWF	Time data (in micro seconds) of raw waveform to be transferred. The channel number is defined by "waveChanNoMO"..	
waveDataWF	Raw waveform data (in volts) to be transferred. The channel number is defined by "waveChanNoMO".	

H. Acquisition Status

Record	Description	Value
statusUpdateSQ	Update statuses. This processes “acqStatusBI” and “trigStatusMI”.	
acqStatusBI	Read data acquisition status.	{ "Stop" | "Run" }
trigStatusMI	Read trigger status.	{ "Stop" | "Wait" | "Capture" }

I. Setting Condition

Comments:		
Preferred setting condition can be stored in the device with a specified data name.		
Record	**Description**	**Value**
resetBO	Reset values to default.	
setupDataNameSO	Data name of setting condition.	
saveSetupDataBO	Save setting condition with specified data name into the device.	
recallSetupDataBO	Recall setting condition of specified data name from the device.	

J. SRQ

Comments:		
SQR interrupt function is supported by asyn4.11 or later.		
Record	**Description**	**Value**
srqEnableBO	Set Enable/Disable status for receive-ing SRQ interrupt.	{ "Disable" | "Enable" } “Enable” is default.
srqEnableBI	Read Enable/Disable status for SRQ interrupt.	
srqEventNoAO	Set SRQ event interrupt number. The default value is 1. The default number is set by the environmental variable of “SRQ_EVNT”.	{ 1 to 255 }

K. MISC

Record	Description	Value
frontPanelLockBO	Lock/unlock front panel.	{ "Unlock" | "Lock" }
frontPanelLockBI	Read lock/unlock status of front panel.	
Name	Device's ID label.	

4. Sample MEDM Waveform Viewer

Figures 6 and 7 are images of our sample viewer developed with MEDM. Figure 6 is the image of a main display, which is used to control the device and to display compressed waveforms (dispWaveChan[n]WF). Figure 7 shows a sub-waveform viewer, which displays a raw waveform (waveDataWF).

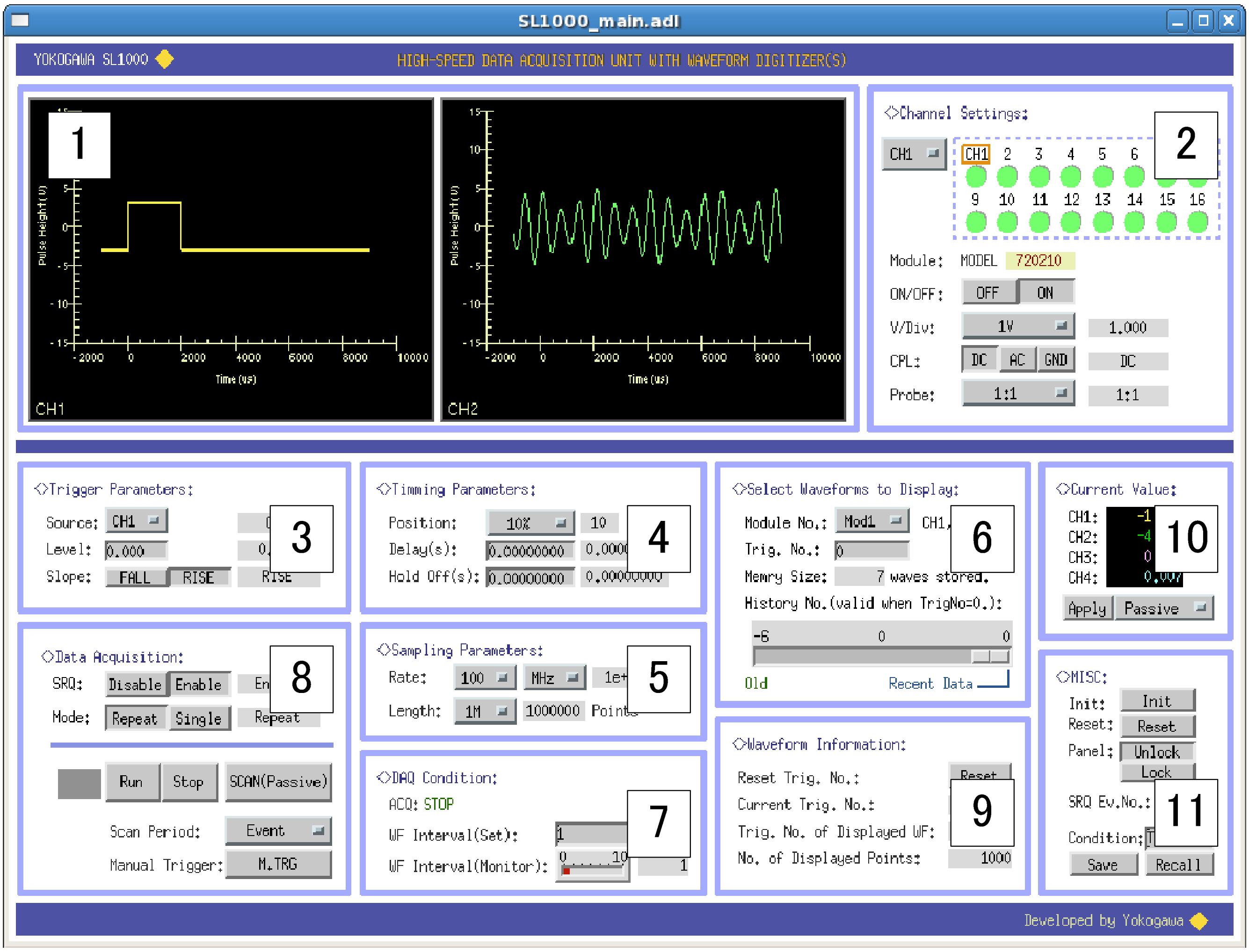


Figure 6: The main viewer display developed with MEDM.

[1]: The waveform viewer. Two waveforms of the selected module are displayed.

[2]: Used for channel settings: “On/Off”, “V/Div”, “AC/DC/GND”, and “Probe Setting”. The records for these parameters are common to all channels. When setting these parameters, select a target channel in advance.

[3]: Used to set trigger parameters: “Source”, “Trigger Level”, and “Trigger Slope”.

[4]: Used to set timing parameters: “Trigger Position”, “Trigger Delay”, and “Holdoff Time”.

[5]: Used to set sampling parameters: “Sampling Rate” and “Record Length”.

[6]: The menu “Module Number” is used to select a target module of which waveforms are displayed. The slider titled “Waveform to be displayed” is used to set a history number of waveforms. The shell command button is used to execute the sub-waveform viewer.

[7]: Indicators of the acquisition statuses. The bar is a monitor of the value of “C-V” (see Figure 4). If the device support does not lose any waveforms, the value is 1 or zero. The value of greater than 1 means that some waveforms could not be transferred to the IOC.

[8]: The data acquisition menus. When the SRQ function is used, enable “SRQ” and set “SCAN” to “Event”. The buttons “Run” and “Stop” starts and stops the data acquisition.

[9]: The information of the displayed waveforms is shown. The button “Reset” resets the trigger number of the device.

[10]: This is for the “Current Value Measurement” function. When the record “currValUpdateSQ“ is processed, current values at the timing is transferred.

[11]: The button “Init” initializes the device. The button “Lock(Unlock)” locks(unlocks) the front panel of the device. The button “Save(Recall)” saves (recalls) the setting condition in (from) the device. Specify the condition data name.

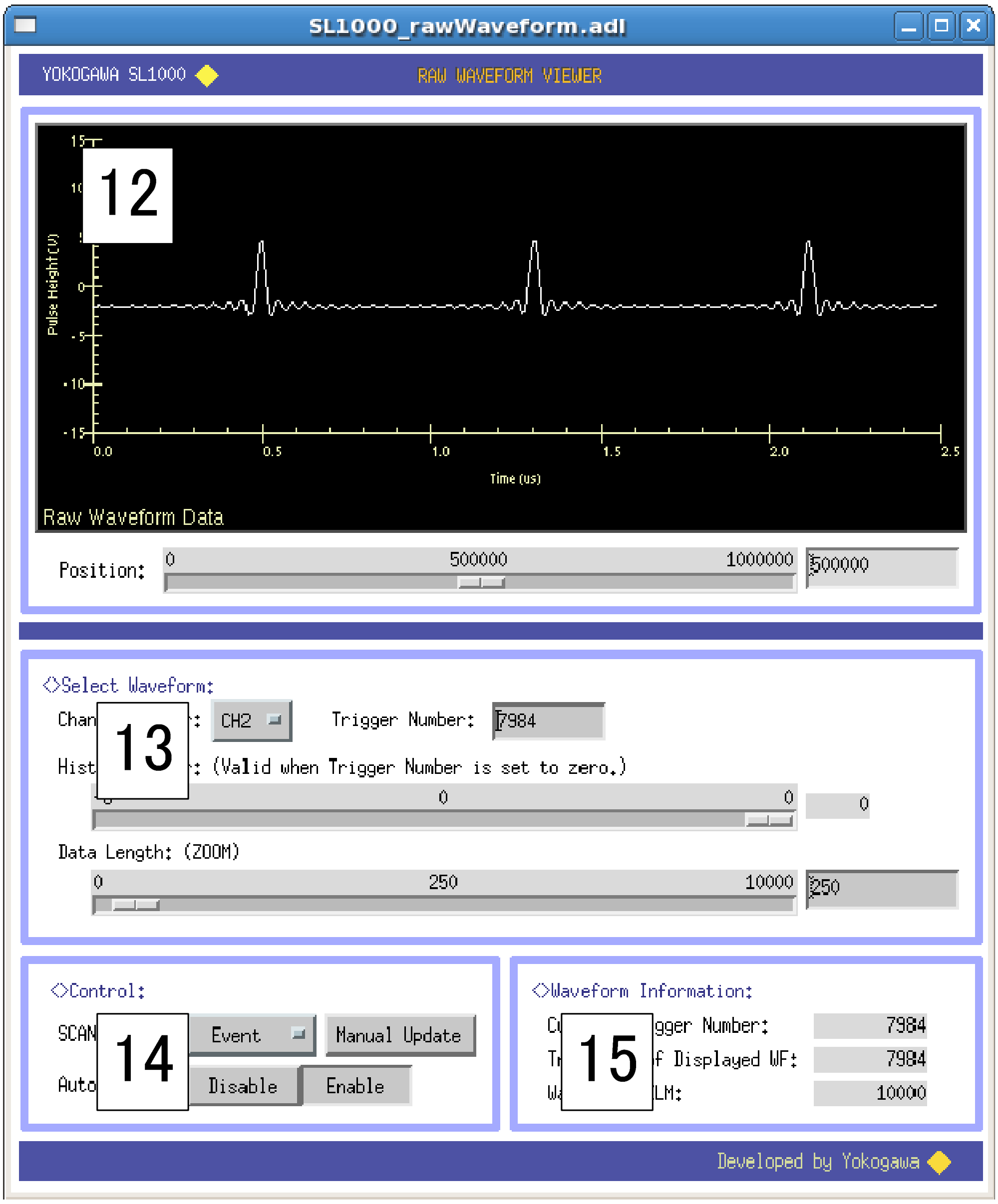


Figure 7: The sub viewer display developed with MEDM.

[12]: The raw waveform viewer. Since the record for the raw waveform data (waveDataWF) are common to all the channels, select a target channel to be displayed in advance (see [12]). Then the waveform of the selected channel is displayed. The slider is used to set the start position (point number) of the selected waveform.

[13]: Used to select a waveform. The menu "Channel Number" is used to select a target channel. Specify a trigger number or set a history number (slider) to select a waveform. The slider "Data Length" sets the number of data points to be displayed. The waveform can be zoomed up using the "Data Length" slider and "Position" slider. If "Auto Update" is enabled, the displayed waveform is updated automatically when any of the WF selection

parameters is changed.

[14]: When “SCAN”=”Event” and the SRQ function is enabled, raw waveform data can be transferred and updated automatically during the acquisition.

[15]: The information of the displayed waveform is shown.

5. Acknowledgements

We would like to thank Prof. Kazuro Furukawa, High Energy Accelerator Research Organization (Japan) for his valuable suggestions in developing this device support. We also thank the authors of the device support for the TDS3000 oscilloscope. We have learned much from their work.

6. License Agreement

SL1000 is distributed subject to the following license conditions:

SOFTWARE LICENSE AGREEMENT
Software: SL1000

1. The "Software", below, refers to SL1000 (in either source code, or binary form and accompanying documentation). Each licensee is addressed as "you" or "Licensee."

2. The copyright holders shown above and their third-party licensors hereby grant Licensee a royalty-free nonexclusive license, subject to the limitations stated herein and U.S. Government license rights.

3. You may modify and make a copy or copies of the Software for use within your organization, if you meet the following conditions:
   a. Copies in source code must include the copyright notice and this Software License Agreement.
   b. Copies in binary form must include the copyright notice and this Software License Agreement in the documentation and/or other materials provided with the copy.

4. You may modify a copy or copies of the Software or any portion of it, thus forming a work based on the Software, and distribute copies of such work outside your organization, if you meet all of the following conditions:
   a. Copies in source code must include the copyright notice and this Software License Agreement;
   b. Copies in binary form must include the copyright notice and this Software License Agreement in the documentation and/or other materials provided with the copy;
   c. Modified copies and works based on the Software must carry

prominent notices stating that you changed specified portions of the Software.

5. Portions of the Software resulted from work developed under a U.S. Government contract and are subject to the following license: the Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this computer software to reproduce, prepare derivative works, and perform publicly and display publicly.

6. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDERS, THEIR THIRD PARTY LICENSORS, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

7. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDERS, THEIR THIRD PARTY LICENSORS, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE, EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGES.