



SoftScope SDK 1.1 Function Reference Guide

Tel : +82-2 470-0491 Fax: +82-2 474-0659

Address : *soft*DSP Co., Ltd.

Jungil Bldg 203, 552-1 Sungnae-dong, Kangdong-ku, Seoul, Korea

Home page: <http://www.softdsp.com>

CONTENTS

API functions reference	4
1. sdInitialize.....	4
2. sdFinalize	5
3. sdGetDataBuffer	6
4. sdStoreData	7
5. sdGetDataVal	8
6. sdRaw2Physical.....	9
7. sdGetData.....	10
8. sdGetDataBufferSize	11
9. sdSetChannelOnOff	12
10. sdGetChannelOnOff	13
11. sdSetChannelVoltage	14
12. sdGetChannelVoltage	16
13. sdSetChannelCoupling	18
14. sdGetChannelCoupling.....	19
15. sdSetChannelOffset	20
16. sdGetChannelOffset.....	21
17. sdSetChannelOffsetVoltage.....	22
18. sdGetChannelOffsetVoltage	23
19. sdSetTimeDiv	24
20. sdGetTimeDiv.....	26
21. sdSetDelayOnOff.....	27
22. sdGetDelayOnOff.....	28
23. sdSetDelayOffset	29
24. sdGetDelayOffset	30
25. sdSetDelayOffsetTime.....	31
26. sdGetDelayOffsetTime	32
27. sdSetTriggerOffset.....	33
28. sdGetTriggerOffset	34

29.	sdSetTriggerOffsetVoltage	35
30.	sdGetTriggerOffsetVoltage.....	36
31.	sdSetTriggerSource.....	37
32.	sdGetTriggerSource.....	38
33.	sdSetTriggerMode	39
34.	sdGetTriggerMode.....	40
35.	sdSetTriggerSlope.....	41
36.	sdGetTriggerSlope.....	42
37.	sdSetETSONOff	43
38.	sdGetETSONOff	44
39.	sdSetBufferSize	45
40.	sdSetPeakDetectOnOff.....	46
41.	sdGetPeakDetectOnOff	47
42.	sdSetBandwidthLimitOnOff	48
43.	sdGetBandwidthLimitOnOff.....	49
44.	sdSetTriggerHighFrequencyRejectOnOff.....	50
45.	sdGetTriggerHighFrequencyRejectOnOff.....	51
46.	sdGetErrorMessages.....	52

API functions reference

1. sdInitialize

- **Description:**

This function initializes SDS 200 / SDS 200A and SoftScope libraries.

It allocates memory buffer and begins USB communication.

- **Format :**

bool sdInitialize();

- **Returns :**

true on success/false on failure

- **Comments:**

When an error occurs, a user can identify its reason by sdGetErrorMessages() function.

2. sdFinalize

- **Description:**

This function finalizes SDS 200 / SDS 200A and SoftScope libraries.

It frees allocated memory buffers and finalizes the USB communication.

- **Format :**

void sdFinalize();

- **Returns :**

None

- **Comments:**

Using sdInitialize will cause an error if it is ended with improper SdFinalize. In this case, SDS 200 / SDS 200A should be initialized by unplugging/plugging of the USB cable.

3. sdGetDataBuffer

- **Description:**

Using this function you can get the pointer of data buffer inside SoftScope library.

As acquired data is RAW_DATA type, sdRaw2Physical function should be employed to convert it to voltage value.

- **Format :**

```
bool sdGetDataBuffer(
    CHANNEL ch_, RAW_DATA** ch_p1, RAW_DATA** ch_p2
);
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2
_BOTH or 0	Acquire data from Channel 1 & Channel2

ch_p1/ch_p2

address of pointer variable to store the buffer address inside SoftScope library.

- **Usage :**

```
RAW_DATA *buffer1;
// Now buffer1 has the address of buffer of SoftScope library.
bool res = sdGetDataBuffer(_CH1, &buffer1, NULL);
```

- **Returns :**

true on success/false on failure

- **Comments:**

As acquired data is RAW_DATA type, sdRaw2Physical function should be employed to convert it to voltage value

4. sdStoreData

- **Description:**

Using this function you command SDS 200 / SDS 200A to acquire the data samples and fill the buffer inside SoftScope library.

- **Format :**

```
bool sdStoreData(  
    CHANNEL ch_  
);
```

- **Parameters:**

Value	Meaning
<u>_CH1</u> or 1	Acquire data from Channel 1
<u>_CH2</u> or 2	Acquire data from Channel 2
<u>_BOTH</u> or 0	Acquire data from Channel 1 & Channel2

- **Returns :**

true on success/false on failure

- **Comments:**

sdStoreData () function is to be used to fetch the stored internal data out of a buffer.

5. sdGetDataVal

- **Description:**

Using this function you can get the stored data from the buffer inside SoftScope library.

- **Format :**

```
bool sdGetDataVal(  
    CHANNEL ch_, int index  
);
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2
_BOTH or 0	Acquire data from Channel 1 & Channel2

index

index to Nth data of the buffer.

- **Returns :**

true on success/false on failure

- **Comments:**

sdGetDataVal() function is to be used to fetch the stored internal data out of a buffer.

6. sdRaw2Physical

- **Description:**

Using this function you can transform the RAW_DATA (A/D converter value) to PHYSICAL_DATA (Voltage).

- **Format :**

```
PHYSICAL_DATA sdRaw2Physical(
    CHANNEL ch_, RAW_DATA data_
);
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Transform Channel 1 data
_CH2 or 2	Transform Channel 2 data

data

RAW_DATA(2 byte unsigned short) to be transformed.

- **Returns :**

Success - Transformed PHYSICAL_DATA(8 byte double) will be returned.

Failure - SD_DATA_ERROR (defined as -10000) will be returned.

- **Comments:**

If data is successfully acquired, voltage value of index location will be returned. If an error occurs, SD_DATA_ERROR (defined as -10000) will be returned. Before using this function, a user should obtain data by sdStoreData() function.

Ch_ parameter must be _CH1 or _CH2.

Index value is supposed to be smaller than the return value of sdGetBufferSize() function.

7. sdGetData

- **Description:**

Using this function you command SDS 200 / SDS 200A to acquire data and fill the buffer you provide .

- **Format :**

```
bool sdGetData(
    CHANNEL ch_, PHYSICAL_DATA* ch_p1, PHYSICAL_DATA* ch_p2
);
```

- **Parameters:**

Ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2
_BOTH or 0	Acquire data from Channel 1 & Channel2

ch_p1/ch_p2

PHYSICAL_DATA(8 byte double) buffer to be filled.

- **Returns :**

true on success/false on failure

- **Comments:**

The desired data from each channel can be obtained on condition that channel is On. And data acquisition from both ch1 and ch2 is not allowed to obtain data from channel 1 and channel 2 simultaneously. For data acquisition, a buffer for storing data should be allocated beforehand, and data has voltage value. In other words, if it is 1, it has 1V value and if it is 0.1, it has 100mv of value.

8. sdGetDataBufferSize

- **Description:**

Using this function you can get the data size that SDS 200 / SDS 200A acquires.

- **Format :**

int sdGetDataBufferSize()

- **Returns :**

None

- **Comments:**

The buffer size per channel will be returned. The size of data buffer size in channel 1 is identical to that of channel 2.

9. sdSetChannelOnOff

- **Description:**
Using this function you can turn on/off each channel .

- **Format :**
bool sdSetChannelOnOff(
 CHANNEL ch_, bool OnOff_
)

- **Parameters:**

Ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

onOff_

Value	Meaning
True	Turn on the channel
False	Turn off the channel

- **Returns :**
true on success/false on failure
- **Comments:**
It is possible to use other functions only when the channel is On state. Accordingly, if a user wants to use the functions related to a channel, he should check in advance if the channel state is On.

10. sdGetChannelOnOff

- **Description:**

Using this function you command SDS 200 / SDS 200A to acquire data and fill the buffer you provide .

- **Format :**

```
bool sdGetChannelOnOff(
    CHANNEL ch_
)
```

- **Parameters:**

Ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

- **Returns:**

Value	Meaning
True	Turn on the channel
False	Turn off the channel

- **Comments:**

It is possible to use other functions only when the channel is On state. Accordingly, if a user wants to use the functions related to a channel, he should check in advance if the channel state is On.

11. sdSetChannelVoltage

- **Description:**
Using this function you can change the voltage division .

- **Format :**
bool sdSetChannelVoltage(
 CHANNEL ch_, VOLTAGE_DIV volt_
)

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

volt_

Value	Meaning
_10MV_DIV or 1	10mV/DIV
_20MV_DIV or 2	20mV/DIV
_50MV_DIV or 3	50mV/DIV
_100MV_DIV or 4	100mV/DIV
_200MV_DIV or 5	200mV/DIV
_500MV_DIV or 6	500mV/DIV
_1V_DIV or 7	1V/DIV
_2V_DIV or 8	2V/DIV
_5V_DIV or 9	5V/DIV
_10V_DIV or 10	10V/DIV

- **Returns:**
true on success/false on failure

- **Comments:**

It is only possible to alter Volt/Div on condition that the channel state is On.

12. sdGetChannelVoltage

- **Description:**

Using this function you can get the voltage division status for each channel.

- **Format :**

```
VOLTAGE_DIV sdGetChannelVoltage(
    CHANNEL ch_
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

- **Returns:**

Value	Meaning
_10MV_DIV or 1	10mV/DIV
_20MV_DIV or 2	20mV/DIV
_50MV_DIV or 3	50mV/DIV
_100MV_DIV or 4	100mV/DIV
_200MV_DIV or 5	200mV/DIV
_500MV_DIV or 6	500mV/DIV
_1V_DIV or 7	1V/DIV
_2V_DIV or 8	2V/DIV
_5V_DIV or 9	5V/DIV
_10V_DIV or 10	10V/DIV

- **Comments:**

It is only possible to get Volt/Div value on condition that the channel state is On.

13. sdSetChannelCoupling

- **Description:**

Using this function you can change the channel coupling status .

- **Format :**

```
bool sdSetChannelCoupling(
    CHANNEL ch_, CHANNEL_COUPLING coup_
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Get status of Channel 1
_CH2 or 2	Get status of Channel 2

coup_

Value	Meaning
_DC or 1	DC Coupling
_AC or 0	AC Coupling

- **Returns:**

true on success/false on failure

- **Comments:**

It is only possible to alter coupling state on condition that the channel state is On.

14. sdGetChannelCoupling

- **Description:**

Using this function you can get the coupling status for each channel.

- **Format :**

```
CHANNEL_COUPLING sdGetChannelCoupling(
    CHANNEL ch_
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

- **Returns:**

Value	Meaning
_DC or 1	DC Coupling
_AC or 0	AC Coupling

- **Comments:**

It is only possible to get coupling value on condition that the channel state is On.

15. sdSetChannelOffset

- **Description:**

Using this function you can change the voltage offset value for each channel.

- **Format :**

```
bool sdSetChannelOffset(
    CHANNEL ch_, double offset_
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Get status of Channel 1
_CH2 or 2	Get status of Channel 2

offset_

Any value between 0 to 400.

- **Returns:**

true on success/false on failure

- **Comments:**

It is possible to adjust Offset as long as a channel is 'on' state. This function is necessary to build SoftScope and required to be adjusted to 200 / SDS 200A. As the screen size of SoftScope is 500 X 400, the middle of vertical axis, i.e., 200, is in fact to be 0.

16. sdGetChannelOffset

- **Description:**

Using this function you can get the Offset value for each channel.

- **Format :**

```
double sdGetChannelOffset(  
    CHANNEL ch_  
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

- **Returns:**

Any value between 0 to 400.

- **Comments:**

It is only possible to get coupling value on condition that the channel state is On.

17. sdSetChannelOffsetVoltage

- **Description:**

Using this function you can change the voltage offset value for each channel.

- **Format :**

```
bool sdSetChannelOffsetVoltage(
    CHANNEL ch_, double offset_
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Get status of Channel 1
_CH2 or 2	Get status of Channel 2

offset_

On zero-voltage base, there exists +/- maximum value in each Volt/Div. 1V corresponds to 1 while 100mV has the value of 0.1. Additionally, the maximum and minimum value is four times greater than adjusted Volt/Div, respectively. For instance, in the case of 1Volt/Div, the maximum/minimum value is supposed to be +4V/-4V.

- **Returns:**

true on success/false on failure

- **Comments:**

It is possible to adjust Offset as long as a channel is 'on' state.

18. sdGetChannelOffsetVoltage

- **Description:**

Using this function you can get the offset voltage value for each channel.

- **Format :**

```
double sdGetChannelOffsetVoltage(
    CHANNEL ch_
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

- **Returns:**

If the value representing voltage is 1, it means 1 Volt, and if it is 0.1, it stands for 100mV.

- **Comments:**

It is only possible to get coupling value on condition that the channel state is On.

19. sdSetTimeDiv

- **Description:**
Using this function you can change the time division .

- **Format :**
bool sdSetTimeDiv(
 CHANNEL ch_, TIME_DIV time_
)

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Acquire data from Channel 1
_CH2 or 2	Acquire data from Channel 2

time_

Value	Meaning
_2NS_DIV or 0	2ns/Div
_4NS_DIV or 1	4ns/Div
_10NS_DIV or 2	10ns/Div
_20NS_DIV or 3	20ns/Div
_40NS_DIV or 4	40ns/Div
_100NS_DIV or 5	100ns/Div
_200NS_DIV or 6	200ns/Div
_400NS_DIV or 7	400ns/Div
_1US_DIV or 8	1us/Div
_2US_DIV or 9	2us/Div
_4US_DIV or 10	4us/Div
_10US_DIV or 11	10us/Div
_20US_DIV or 12	20us/Div

_40US_DIV or 13	40us/Div
_100US_DIV or 14	100us/Div
_200US_DIV or 15	200us/Div
_400US_DIV or 16	400us/Div
_1MS_DIV or 17	1ms/Div
_2MS_DIV or 18	2ms/Div
_4MS_DIV or 19	4ms/Div
_10MS_DIV or 20	10ms/Div
_20MS_DIV or 21	20ms/Div
_40MS_DIV or 22	40ms/Div
_100MS_DIV or 23	100ms/Div
_200MS_DIV or 24	200ms/Div
_400MS_DIV or 25	400ms/Div
_1S_DIV or 26	1s/Div
_2S_DIV or 27	2s/Div
_4S_DIV or 28	4s/Div
_10S_DIV or 29	10s/Div

- **Returns:**

true on success/false on failure.

- **Comments:**

It is only possible to change time/div value on condition that the channel state is On.

20. sdGetTimeDiv

- **Description:**
Using this function you can get the time/div status.
- **Format :**
TIME_DIV sdGetTimeDiv()
- **Parameters :**
None
- **Returns:**
Current Time/div setting, refer to the second input parameter of sdSetTimeDiv() function.
- **Comments:**
It is only possible to change time/div value on condition that the channel state is On.

21. sdSetDelayOnOff

- **Description:**

Using this function you can set delay on/off .

- **Format :**

```
bool sdSetDelayOnOff(
    bool onOff_
)
```

- **Parameters:**

onOff_

Value	Meaning
True	Delay On
False	Delay Off

- **Returns:**

true on success/false on failure.

- **Comments:**

If Time/div is adjusted when Delay status is 'On' state, the time for trigger will be fixed on horizontal axis, and the location of Delay will be changed.

22. sdGetDelayOnOff

- **Description:**
Using this function you can get the delay on/off status.

- **Format :**
bool sdGetDelayOnOff()

- **Parameters :**
None

- **Returns:**

Value	Meaning
True	Delay On
False	Delay Off

- **Comments:**
None

23. sdSetDelayOffset

- **Description:**

Using this function you can change the delay offset value.

- **Format :**

```
bool sdSetDelayOffset(  
    double offset_  
)
```

- **Parameters:**

offset_

To set the location for trigger on the screen If Delay is On: To set between -4500 and 500 If Delay is Off: To set between 0 to 500 250 is the center of screen with Delay 0

- **Returns:**

true on success/false on failure

- **Comments:**

250 is the value with delay 0 based on SoftScope screen. As the screen size of horizontal axis is 500, the starting location of horizontal axis is 0 and ending location is 500 when Delay is Off state. For each 50, it will be pre or post as much as the time of current Time/Div.

24. sdGetDelayOffset

- **Description:**

Using this function you can get the offset value in screen scale.

- **Format :**

double sdGetDelayVoltage()

- **Parameters:**

None

- **Returns:**

If Delay is On: To return the value between -4500 and 500

If Delay is Off: To return the value between 0 and 500

250 is the middle value with no Delay.

- **Comments:**

250 is the value with delay 0 based on SoftScope screen. As the screen size of horizontal axis is 500, the starting location of horizontal axis is 0 and ending location is 500 when Delay is Off state. For each 50, it will be pre or post as much as the time of current Time/Div.

25. sdSetDelayOffsetTime

- **Description:**

Using this function you can change the delay offset value in time scale.

- **Format :**

```
bool sdSetDelayOffsetTime(  
    double offset_  
)
```

- **Parameters:**

offset_

If Delay is On: To set the value between current Time/Div value * (-95) and current Time/Div value * 5

If Delay is Off: To set the value between current Time/Div value * (-5) and current Time/Div value * 5

1 Sec. has the value of 1 and 100mSec. corresponds to 0.1. Zero is the middle value with Delay 0.

- **Returns:**

true on success/false on failure

- **Comments:**

None

26. sdGetDelayOffsetTime

- **Description:**

Using this function you can get the offset value in time scale.

- **Format :**

double sdGetDelayOffset()

- **Parameters:**

None

- **Returns:**

If Delay is On: To return the value between current Time/Div value * (-95) and current Time/Div value * 5

If Delay is Off: To return the value between current Time/Div value * (-5) and current Time/Div value * 5

1 corresponds to 1 Sec. while 100mSec has the value of 0.1. 0 means the middle value with Delay 0.

- **Comments:**

None

27. sdSetTriggerOffset

- **Description:**

Using this function you can change the trigger offset value.

- **Format :**

```
bool sdSetTriggerOffset(  
    double offset_  
)
```

- **Parameters:**

offset_

The measured difference will be true offset on the basis of channel offset. Therefore, the differential value with channel offset is to be put in as a desired trigger offset value. The offset value of trigger is linked to current Volt/Div value and the difference of offset 50 is equal to Volt value of Volt/Div.

- **Returns:**

true on success/false on failure

- **Comments:**

As this function is based on 500 x 400 screen size, the middle of vertical axis is 200 with offset 0. This trigger offset is relative on the basis of channel offset. Consequently, the actual trigger offset will be 0 on condition that channel offset is equal to trigger offset.

When a user is to set the actual Trigger Offset value, he can use sdSetTriggerOffsetVoltage

28. sdGetTriggerOffset

- **Description:**
Using this function you can get the trigger offset in screen scale.
- **Format :**
`double sdGetTriggerOffset()`
- **Parameters:**
None
- **Returns:**
The differential value based on channel offset will be returned.
- **Comments:**
None

29. sdSetTriggerOffsetVoltage

- **Description:**

Using this function you can change the trigger offset value.

- **Format :**

```
bool sdSetTriggerOffsetVoltage(  
    double offset_  
)
```

- **Parameters:**

offset_

As a trigger offset voltage value to be set, 1 is set for 1V and 0.1 for 100mV.

- **Returns:**

true on success/false on failure

- **Comments:**

None

30. sdGetTriggerOffsetVoltage

- **Description:**

Using this function you can get the trigger offset in voltage scale.

- **Format :**

double sdGetTriggerOffsetVoltage()

- **Parameters:**

None

- **Returns:**

As a trigger offset voltage value to be set, 1 is set for 1V and 0.1 for 100mV.

- **Comments:**

None

31. sdSetTriggerSource

- **Description:**
Using this function you can set the trigger source .

- **Format :**
bool sdSetTriggerSource(
 TRIGGER_SOURCE ch_
)

- **Parameters:**
onOff_

Value	Meaning
_TRIGGER_SOURCE_CH1 or 1	CH1 is trigger source
_TRIGGER_SOURCE_CH2 or 2	CH2 is trigger source

- **Returns:**
true on success/false on failure.
- **Comments:**
None

32. sdGetTriggerSource

- **Description:**
Using this function you can get the trigger source.

- **Format :**
bool sdGetTriggerSource()

- **Parameters :**
None

- **Returns:**

Value	Meaning
_TRIGGER_SOURCE_CH1 or 1	CH1 is trigger source
_TRIGGER_SOURCE_CH2 or 2	CH2 is trigger source

- **Comments:**
None

33. sdSetTriggerMode

- **Description:**

Using this function you can set the trigger mode .

- **Format :**

```
bool sdSetTriggerMode(
    TRIGGER_MODE mode_
)
```

- **Parameters:**

mode_

Value	Meaning
_AUTO or 0	AUTO trigger mode
_NORMAL or 1	NORMAL trigger mode

- **Returns:**

true on success/false on failure.

- **Comments:**

Arbitrary trigger signal will be generated in Auto Trigger Mode if trigger is not activated. On the other hand, there will be no input waveform in Normal Trigger Mode, if trigger is not activated.

34. sdGetTriggerMode

- **Description:**
Using this function you can get the trigger mode.

- **Format :**
bool sdGetTriggerMode()

- **Parameters :**
None

- **Returns:**

Value	Meaning
_AUTO or 0	AUTO trigger mode
_NORMAL or 1	NORMAL trigger mode

- **Comments:**
None

35. sdSetTriggerSlope

- **Description:**

Using this function you can set the trigger slope .

- **Format :**

```
bool sdSetTriggerSlope(
    TRIGGER_SLOPE slope_
)
```

- **Parameters:**

slope_

Value	Meaning
_TRIGGER_SLOPE_DOWN or 0	Down slope
_TRIGGER_SLOPE_U or 1	Up slope

- **Returns:**

true on success/false on failure.

- **Comments:**

None

36. sdGetTriggerSlope

- **Description:**
Using this function you can get the trigger slope.
- **Format :**
bool sdGetTriggerSlope()
- **Parameters :**
None
- **Returns:**

Value	Meaning
_TRIGGER_SLOPE_DOWN or 0	Down slope
_TRIGGER_SLOPE_U or 1	Up slope

- **Comments:**
None

37. sdSetETSONOff

- **Description:**

Using this function you can turn on/off ETS.

- **Format :**

```
bool sdSetETSONOff(
    bool onOff_
)
```

- **Parameters:**

onOff_

Value	Meaning
True	ETS Mode
False	RIS RT Mode

- **Returns:**

true on success/false on failure.

- **Comments:**

None

38. sdGetETSONOff

- **Description:**
Using this function you can get the state of ETS on/off.

- **Format :**
bool sdGetETSONOff()

- **Parameters :**
None

- **Returns:**

Returns: Value	Meaning
True	ETS Mode
False	RIS RT Mode

- **Comments:**
None

39. sdSetBufferSize

- **Description:**

Using this function you can select buffer size 10K or 500K .

- **Format :**

```
bool sdSetBufferSize(
    BUFFER_SIZE_MODE bufferSize_
)
```

- **Parameters:**

bufferSize_

Value	Meaning
_10K or 3	Buffer size is 10K
_500K or 7	Buffer size is 500K

- **Returns:**

true on success/false on failure.

- **Comments:**

This function is available on SDS 200A.

40. sdSetPeakDetectOnOff

- **Description:**

Using this function you can turn on/off the peak detection.

- **Format :**

```
bool sdSetPeakDetectOnOff(  
    bool onOff_  
)
```

- **Parameters :**

onOff_

Value	Meaning
True	Peak Detection ON
false	Peak Detection OFF

- **Returns:**

true on success/false on failure.

- **Comments:**

This function is available on SDS 200A.

41. sdGetPeakDetectOnOff

- **Description:**
Using this function you can get the state of Peak Detection on/off.

- **Format :**
bool sdGetPeakDetectOnOff ()

- **Parameters :**
None

- **Returns:**

Value	Meaning
True	Peak Detection ON
false	Peak Detection OFF

- **Comments:**
This function is available on SDS 200A.

42. sdSetBandwidthLimitOnOff

- **Description:**

Using this function you can turn on/off Bandwidth Limit of selected channel.

- **Format :**

```
bool sdSetBandwidthLimitOnOff (
    CHANNEL ch_, bool onOff_
)
```

- **Parameters:**

ch_

Value	Meaning
_CH1 or 1	Channel 1
_CH2 or 2	Channel 2

onOff

Value	Meaning
True	Bandwidth Limit ON
false	Bandwidth Limit OFF

- **Returns:**

true on success/false on failure.

- **Comments:**

This function is available on SDS 200A.

43. sdGetBandwidthLimitOnOff

- **Description:**

Using this function you can get the state of Bandwidth Limit of selected channel On/Off.

- **Format :**

```
bool sdGetBandwidthLimitOnOff (
    CHANNEL ch_
)
```

- **Parameters :**

ch_

Value	Meaning
_CH1 or 1	Channel 1
_CH2 or 2	Channel 2

- **Returns:**

Value	Meaning
True	Bandwidth Limit ON
false	Bandwidth Limit OFF

- **Comments:**

This function is available on SDS 200A.

44. sdSetTriggerHighFrequencyRejectOnOff

- **Description:**

Using this function you can turn on/off the High Frequency Rejection..

- **Format :**

```
bool sdSetTriggerHighFrequencyRejectOnOff (  
    bool onOff_  
)
```

- **Parameters :**

onOff_

Value	Meaning
True	High Frequency Rejection ON
false	High Frequency Rejection OFF

- **Returns:**

true on success/false on failure.

- **Comments:**

This function is available on SDS 200A.

45. `sdGetTriggerHighFrequencyRejectOnOff`

- **Description:**

Using this function you can get the state of High Frequency Rejection on/off.

- **Format :**

`bool sdGetTriggerHighFrequencyRejectOnOff ()`

- **Parameters :**

None

- **Returns:**

Value	Meaning
True	High Frequency Rejection ON
false	High Frequency Rejection OFF

- **Comments:**

This function is available on SDS 200A.

46. sdGetErrorMessages

- **Description:**
When an error occurs, an error message will be given.
- **Format :**
bool sdGetErrorMessages()
- **Parameters :**
None
- **Returns:**
Error message string
- **Comments:**
None