



# **SoftScope SDK 1.0**

## **Function Reference**

### **Guide**

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

Address : *softDSP* 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 .....                 | 4  |
| 3. sdGetDataBuffer .....            | 5  |
| 4. sdGetDataBuffer .....            | 6  |
| 5. sdGetDataVal .....               | 7  |
| 6. sdRaw2Physical .....             | 8  |
| 7. sdGetData.....                   | 9  |
| 8. sdGetDataBufferSize .....        | 10 |
| 9. sdSetChannelOnOff .....          | 11 |
| 10. sdGetChannelOnOff .....         | 12 |
| 11. sdSetChannelVoltage .....       | 13 |
| 12. sdGetChannelVoltage .....       | 14 |
| 13. sdSetChannelCoupling .....      | 15 |
| 14. sdGetChannelCoupling.....       | 16 |
| 15. sdSetChannelOffset .....        | 17 |
| 16. sdGetChannelOffset.....         | 18 |
| 17. sdSetChannelOffset .....        | 19 |
| 18. sdGetChannelOffsetVoltage ..... | 20 |
| 19. sdSetTimeDiv .....              | 21 |
| 20. sdGetTimeDiv.....               | 23 |
| 21. sdSetDelayOnOff.....            | 24 |
| 22. sdGetDelayOnOff .....           | 25 |
| 23. sdSetDelayOffset .....          | 26 |
| 24. sdGetDelayOffset .....          | 27 |
| 25. sdSetDelayOffsetTime.....       | 28 |
| 26. sdGetDelayOffsetTime .....      | 29 |
| 27. sdSetTriggerOffset.....         | 30 |
| 28. sdGetTriggerOffset .....        | 31 |
| 29. sdSetTriggerOffsetVoltage ..... | 32 |

|     |                                |    |
|-----|--------------------------------|----|
| 30. | sdGetTriggerOffsetVoltage..... | 33 |
| 31. | sdSetTriggerSource.....        | 34 |
| 32. | sdGetTriggerSource .....       | 35 |
| 33. | sdSetTriggerMode .....         | 36 |
| 34. | sdGetTriggerMode.....          | 37 |
| 35. | sdSetTriggerSlope.....         | 38 |
| 36. | sdGetTriggerSlope.....         | 39 |
| 37. | sdGetErrorMessages.....        | 40 |

## API functions reference

### 1. sdInitialize

- **Description:**

This function initializes SDS 200 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 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 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                                |
|-------------------|----------------------------------------|
| <i>_CH1 or 1</i>  | Acquire data from Channel 1            |
| <i>_CH2 or 2</i>  | Acquire data from Channel 2            |
| <i>_BOTH or 0</i> | 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. sdGetDataBuffer

- **Description:**

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

- **Format :**

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

- **Parameters:**

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

- **Returns :**

true on success/false on failure

- **Comments:**

sdGetDataVal() 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 sdStoreData(  
    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 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 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 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. 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. 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.

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. This function is necessary to build SoftScope and required to be adjusted to 200. As the screen size of SoftScope is 500 X 400, the middle of vertical axis, i.e., 200, is in fact to be 0.

## 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  |

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

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