

## Declaration

DEVICE\_NAME

DEVICE\_INI : the \*.ini file, which is match for device.

DLL\_API : extern "c" \_declspec(dllimport)

WINAPI : \_stdcall

## General

The following is the user guide of command field.

1. Horizontal
2. Vertical
3. Trigger Main
4. Edge Trigger
5. Video Trigger
6. Pulse Trigger
7. Slope Trigger
8. Some Functions

## Detail Instruction:

### 1. Horizontal

Notes: This is about horizontal part of the oscilloscope.

#### a. Main Scale:

Keywords:

HORIZ-TB

Value:

According to the field "TB", "itemXX" is time base and "OppoXX" is index of device time base.

Example:

If you want to set horizontal scale is 200us, please do it as the following.

Check the DEVICE\_INI, find the filed "TB" and its items and oppos.

For example, "Item10=200us, Oppo10=15", you must Command value is 0x0F.

#### b. Window Scale:

Keywords:

HORIZ-WIN-TB

Value:

According to the field "WINTB", "itemXX" is window show time base and "OppoXX" is index of window show time base.

Example:

If you want to set horizontal scale is 200us, please do it as the following.

Check the DEVICE\_INI, find the filed "WINTB" and its items and oppos.

For example, “**Item16=200us, Oppo16=15**”, you must Command value 0x0F.

**c. Horizontal Trigger Time:**

Keywords:

**HORIZ-TRIGTIME**

Value:

When horizontal trigger position is in the middle of the device screen, this value is 0.

Move horizontal trigger position to the left or the right, this value (horizontal trigger time) is increasing or decreasing.

Use picosecond as a unit of time and is rounded to the nearest integer.

Example:

If you want to set horizontal trigger time is -505.5us, set command value is -505500000000.

## **2. Vertical**

Notes: This is about vertical part of the oscilloscope.

**a. Channel Setting**

Take the case of CH1. (The same as others channels)

**a. Display**

Keywords:

**VERT-CH1-DISP**

Value:

Channel On: 0x01;

Channel Off: 0x00;

**b. Bandwidth**

Keywords:

**VERT-CH1-20MHZ**

Value:

Bandwidth On: 0x01;

Bandwidth Off: 0x00;

**c. Coupling**

Keywords:

**VERT-CH1-COUP**

Value:

According to the field “**VCOUP**”, “**itemXX**” is channel coupling and “**OppoXX**” is index of channel coupling.

Example:

If you want to set channel coupling is “**DC**”, please do it as the following.

Check the DEVICE\_INI, find the filed “**VCOUP**” and its items and oppos.

For example, “**Item1=DC, Oppo1=0**”, you must Command value 0x00.

**d. Position**

Keywords:

**VERT-CH1-POS**

Value:

When CH1 zero level is in the middle of Screen, the value is 0x0000;

When CH1 zero level is in the top of Screen, the value is 0x0064;  
When CH1 zero level is in the bottom of Screen, the value is 0xFF9B;  
So every vertical div is 25, and the total is range from -100 to 100.

Example:

If you want to set channel position is -1.20div, the control value is  $-1.20 \times 25$ .  
So you must set value is -30.

#### e. Scale

##### ①. Vertical Scale Setting

Keywords:

VERT-CH1-VB

Value:

According to the field “VBASE”, “itemXX” is channel coupling and  
“OppoXX” is index of channel coupling.

Example:

If you want to set channel coupling is “500mv”, please do it as the following.

Check the DEVICE\_INI, find the filed “VBASE” and its items and oppos.

For example, “Item8=500mv, Oppo8=7”, you must Command value 0x07.

##### ②. Coarse/Fine Adjustment

Keywords:

VERT-CH1-RPHASE

Value:

Coarse Adjustment: 0x00;

Fine Adjustment: 0x01;

##### ③. Fine Value Setting

Notes:

No Support .

#### f. Prob

Keywords:

VERT-CH1-PROBE

Value:

According to the field “PROBE”, “itemXX” is channel probe and  
“OppoXX” is index of channel probe.

Example:

If you want to set channel probe is “100X”, please do it as the following.

Check the DEVICE\_INI, find the filed “PROBE” and its items and oppos.

For example, “Item3=100X, Oppo3=2”, you must Command value 0x02.

#### g. Invert

Keywords:

VERT-CH1-RPHASE

Value:

Invert On: 0x01;

Invert Off: 0x00;

### ***β. Math Setting***

Notes:

No Support .

## **3. Trigger Main**

Notes: This is about common part of trigger settings.

### **a. Trigger State**

Keywords:

**TRIG-STATE**

Notes:

Only for getting device trigger state. There is no use for Setting it.

Example:

Use “**TRIG-STATE**” to gain the device trigger state. The gained values is as the following.

For example, the value is 2.

Check the DEVICE\_INI, find the filed “**TGSTATE**” and its items and oppos “**Oppo3=2,Item3=Auto**”, which shows that device is auto trigger acquisition.

### **b. Trigger Type**

Keywords:

**TRIG-TYPE**

Value:

According to the field “**TGTYPE**”, “**itemXX**” is channel probe and “**OppoXX**” is index of channel probe.

Example:

If you want to set device trigger type is “**OT**”, please do it as the following.

Check the DEVICE\_INI, find the filed “**TGTYPE**” and its items and oppos.

For example, “**Item5=OT, Oppo5=4**”, you must Command value 0x04.

### **c. Trigger Mode**

Keywords:

**TRIG-MODE**

Value:

0: Auto

1: Normal

2: Single

### **d. Trigger Coupling**

Keywords:

**TRIG-COUP**

Value:

According to the field “**TGCOUP**”, “**itemXX**” is channel probe and “**OppoXX**” is index of channel probe.

Example:

If you want to set device trigger type is “Noise reject”, please do it as the following.  
Check the DEVICE\_INI, find the filed “TGCOUP” and its items and oppos.  
For example, “Item3=Noise reject, Oppo3=2”, you must Command value 0x02.

**e. Trigger Level**

Keywords:

TRIG-VPOS

Value:

When CH1 zero level is in the middle of Screen, the value is 0x0000;  
When CH1 zero level is in the top of Screen, the value is 0x0064;  
When CH1 zero level is in the bottom of Screen, the value is 0xFF9B;  
So every vertical div is 25, and the total is range from -100 to 100.

Example:

If you want to set channel position is -1.00div, the control value is -1.00\*25.  
So you must set value is -25.

**f. Trigger Holdoff Time**

Keywords:

TRIG-HOLDTIME

Value:

When the default value of trigger holdoff time is 100.0ns.  
Use picosecond as a unit of time and is rounded to the nearest integer.

Example:

If you want to set trigger holdoff time is 88.6us, set command value is 88600000.

**4. Edge Trigger**

**a. Trigger Slope**

Keywords:

TRIG-EDGE-SLOPE

Value:

0: Rising;  
1: Falling.

**b. Trigger Source**

Keywords:

TRIG-SRC

Value:

According to the field “TGSRC”, “itemXX” is channel probe and “OppoXX” is index of channel probe.

Example:

If you want to set channel probe is “CH2”, please do it as the following.  
Check the DEVICE\_INI, find the filed “TGSRC” and its items and oppos.  
For example, “Item2=CH2, Oppo2=1”, you must Command value 0x01.

**5. Video Trigger**

**a. Trigger Polarity**

Keywords:

TRIG-VIDEO-NEG

Value:

- 0: Positive;
- 1: Negative.

#### **b. Trigger Source**

Keywords:

TRIG-SRC

Value:

According to the field “TGVIDEOSRC”, “itemXX” is channel probe and “OppoXX” is index of channel probe.

Example:

If you want to set channel probe is “CH2”, please do it as the following.

Check the DEVICE\_INI, find the filed “TGVIDEOSRC” and its items and oppos.

For example, “Item2=CH2, Oppo2=1”, you must Command value 0x01.

#### **c. Trigger Standard**

Keywords:

TRIG-VIDEO-PAL

Value:

- 0: NTSC;
- 1: PAL/SECAM.

#### **d. Trigger Synchronization**

Keywords:

TRIG-VIDEO-SYN

Value:

- 0: ALL Lines;
- 1: Line Number.
- 2: Odd Field
- 3: Even Field
- 4: ALL Field

#### **e. Trigger Line Number**

Notes:

This is only for Trigger Synchronization “ALL Filed”. That is , this is no use in other Trigger Synchronization. Besides, when change the trigger standard, the maximum is different.

Keywords:

TRIG-VIDEO-LINE

Value:

- When use NTSC, range from 1 to 525;
- When use PAL/SECAM, range from 1 to 625;

## **6. Pulse Trigger**

#### **a. Trigger Polarity**

Keywords:

#### TRIG-PULSE-NEG

Value:

- 0: Positive;
- 1: Negative.

#### b. Trigger Source

Keywords:

#### TRIG-SRC

Value:

According to the field “TGPULSESRC”, “itemXX” is channel probe and “OppoXX” is index of channel probe.

Example:

If you want to set channel probe is “CH2”, please do it as the following.

Check the DEVICE\_INI, find the filed “TGPULSESRC” and its items and oppos.

For example, “Item2=CH2, Oppo2=1”, you must Command value 0x01.

#### c. Trigger When

Keywords:

#### TRIG-PULSE-WHEN

Value:

- 0: equal to;
- 1: not equal to;
- 2: more than;
- 3: less than.

#### d. Trigger Pulse Width

Notes:

This is the pulse width time.

Keywords:

#### TRIG-PULSE-TIME

Value:

When the default value of pulse width time is 500.0ns.

It ranges from 20.0ns to 10.0s.

Use picosecond as a unit of time and is rounded to the nearest integer.

Example:

If you want to set pulse width is 10.0us, set command value is 10000000.

## 7. Slope Trigger

#### a. Trigger Slope Set

Keywords:

#### TRIG-SLOPE-SET

Value:

- 0: Rising Set;
- 1: Falling Set.

#### b. Trigger Source

Keywords:

#### TRIG-SRC

Value:

According to the field “TGSLOPESRC”, “itemXX” is channel probe and “OppoXX” is index of channel probe.

Example:

If you want to set channel probe is “CH2”, please do it as the following.

Check the DEVICE\_INI, find the filed “TGSLOPESRC” and its items and oppos.

For example, “Item2=CH2, Oppo2=1”, you must Command value 0x01.

### c. Slope Win

Keywords:

TRIG-SLOPE-WIN

Value:

0: Only select V1;

1: Only select V2;

2: Select both.

### d. Slope Time

Keywords:

TRIG-SLOP-TIME

Value:

When the default value of slope time is 500.0ns.

It ranges from 20.0ns to 10.0s.

Use picosecond as a unit of time and is rounded to the nearest integer.

Example:

If you want to set slope time is 10.0us, set command value is 10000000.

### e. Slope V1 and Slope V2

Notes:

This is the pulse width time.

Keywords:

TRIG-SLOPE-V1 (or TRIG-SLOPE-V2)

Value:

When slope trigger level is in the middle of Screen, the value is 0x0000;

When slope trigger level is in the top of Screen, the value is 0x0064;

When slope trigger level is in the bottom of Screen, the value is 0xFF9B;

So every vertical div is 25, and the total is range from -100 to 100.

Example:

If you want to set channel position is -1.00div, the control value is -1.00\*25.

So you must set value is -25.

### f. Trigger When

Keywords:

TRIG-SLOPE-WHEN

Value:

0: equal to;

1: not equal to;

2: more than;



3: less than.

## 8. Some Functions

**a. DLL\_API\_WORD WINAPI **dsoResetAllBuffer()****

Variable introduction:

NULL.

Notes:

It's used for resetting all buffer in DSTHardDll. When you first connect device or reconnect with power failure, you must do this at the first step.

If successful return 1, otherwise return 0.

**b. DLL\_API\_WORD WINAPI **dsoReadInfProtocol()****

Variable introduction:

NULL.

Notes:

It's used for reading protocol.inf, which is about device control Transmission protocol. When you first connect device or reconnect with power failure, you must do this at the second step.

If successful return 1, otherwise return 0.

**c. DLL\_API\_WORD WINAPI **dsoReadInfKeyProtocol()****

Variable introduction:

NULL.

Notes:

It's used for reading keyprotocol.inf, which is about keyboard control Transmission protocol. When you first connect device or reconnect with power failure, you must do this at the third step.

If successful return 1, otherwise return 0.

**d. DLL\_API\_WORD WINAPI **dsoReadAllSysParametereter()****

Variable introduction:

NULL.

Notes:

It's used for capturing all device states in oscilloscope and saving these state to buffer in DSTHardDll.

If successful return 1, otherwise return 0.

**e. DLL\_API\_WORD WINAPI **dsoWriteAllSysParametereter()****

Variable introduction:

NULL.

Notes:

It's used for setting all device states in oscilloscope, and send buffer in DSTHardDll to device to control the oscilloscope.

If successful return 1, otherwise return 0.

**f. DLL\_API\_WORD WINAPI dsoSetKeyboardCtrlLockState(WORD nState)**

Variable introduction:

**nState:**

WORD Type.

When nstate is 1, lock the device. So keyboard is invalid.

When nstate is 0, unlock the device. So keyboard is invalid.

Notes:

It's used for setting enable control of device.

If successful return 1, otherwise return 0.

**g. DLL\_API\_WORD WINAPI dsoSetDeviceSystemRunStop(WORD nRun)**

Variable introduction:

**nRun:**

WORD Type.

When nRun is 1, suspend the device . So keyboard is invalid.

When nRun is 0, run the device. So keyboard is invalid.

Notes:

It's used for setting enable control of device.

If successful return 1, otherwise return 0.

**h. DLL\_API\_WORD WINAPI dsoGetCollectData(short\* chData, WORD \*nNum, WORD nCH)**

Variable introduction:

**chData:**

point to short array. Use for gaining collected data.

**nNum:**

point to short variable. Use for gaining collect data length.

Notes:

It's used for Reading collected data.

If successful return 1, otherwise return 0.

**i. DLL\_API\_WORD WINAPI dsoGetBytesNumberFromkeyString(  
CHAR\* InBuf,  
int nBufNum  
)**

Variable introduction:

**InBuf:**

point to char array. Use for storing keywords.

**nBufNum:**

int variable. Its value is the string length of keywords

Notes:

It's used for getting command bytes size of keywords.  
The return value of this function is command bytes size.  
For example,

```
//cpp
Char InBuf[65535] = "HORIZ-WIN-TB"; //keywords
int nBufNum= 12; //keywords length
WORD nLength;
nLength = dsoGetBytesNumberFromkeyString(InBuf,nBufNum);
```

At the moment,you will find that nLength is 1.

**j. DLL\_API\_WORD WINAPI dsoGetControlResultFromkeyString(**  
CHAR\* InBuf,  
CHAR\* OutValue,  
int nBufNum,  
int \*nValueNum,  
)

Variable introduction:

**InBuf:**

point to char array. Use for storing keywords.

**OutValue:**

point to char array. Use for storing command value.

**nBufNum:**

int variable. Its value is the string length of keywords.

**nValueNum:**

point to int variable. It is used for getting bytes size.

Notes:

It's used for getting command bytes size of keywords and command value.  
If successful return 1, otherwise return 0.  
For example,horizontal trigger time is 88.6us. Please get the value as the following.

```
//cpp
Char InBuf[65535] = "HORIZ-TIGTIME"; //keywords
Char OutValue[65535];
memset(OutValue,0,sizeof(OutValue)); //empty
int nBufNum= 13; //keywords length
int nValueNum;
If(dsoGetControlResultFromkeyString(InBuf,OutValue,nBufNum,
&nValueNum))
{
}
```

At the moment,you will find that nValueNum equals to 8  
and OutValue[0] = 0xc0

```

OutValue[1] = 0xed
OutValue[2] = 0x47
OutValue[3] = 0x05
OutValue[4] = 0x00
OutValue[5] = 0x00
OutValue[6] = 0x00
OutValue[7] = 0x00

```

However,  $88.6\mu s = 88600000ps$ , that is  $0x00000000547edc0$ . So you need to reverse the sort of OutValue bytes.

**k. DLL\_API\_WORD WINAPI dsoSetkeyStringToControlValue(**  
CHAR\* InBuf,  
CHAR\* OutValue,  
int nBufNum,  
int nValueNum,  
**)**

Variable introduction:

**InBuf:**

point to char array. Use for storing keywords.

**OutValue:**

point to char array. Use for storing command value.

**nBufNum:**

int variable. Its value is the string length of keywords.

**nValueNum:**

int variable. It is used for showing bytes size.

Notes:

It's used for setting command value.

If successful return 1, otherwise return 0.

For example, vertical trigger level is -2.0 div. Please set the value as the following.

//cpp

Char InBuf[65535] = "TRIG-VPOS"; //keywords

Char OutValue[65535];

memset(OutValue,0,sizeof(OutValue)); //empty

int nBufNum= 9; //keywords length

int nValueNum = 2;

//Here is computer control value.

//-2.0 \* 25 = -50. So the value is -50,that is 0xFFCD

OutValue[0] = 0xCD;

OutValue[1] = 0xFF;

If(**dsoSetkeyStringToControlValue**(InBuf,OutValue,nBufNum, nValueNum))

{

}

At the moment,you will find trigger level is 2 div under the middle position.