

SDK - HTHardD11.dll 说明文档

目录

一、说明	3
二、结构体介绍	3
三、函数介绍	4
1. dsoHTDeviceConnect	4
2. dsoHTSearchDevice	4
3. dsoHTReadCalibrationData	4
4. dsoHTWriteCalibrationData	4
5. dsoHTRDAmpCali	5
6. dsoHTWRampCali	5
7. dsoSetUSBBus	6
8. dsoHTSetCHPos	6
9. dsoHTSetVTriggerLevel	7
10. dsoHTSetHTriggerLength	7
11. dsoHTSetCHAndTrigger	8
12. dsoHTSetCHAndTriggerVB	8
13. dsoHTSetSampleRate	9
14. dsoHTSetSampleRateVi	9
15. dsoHTStartCollectData	10
16. dsoHTGetState	11
17. dsoHTGetData	11
18. dsoHTGetScanData	11
19. dsoHTGetRollData	12
20. dsoHTSetHardFC	12
21. dsoHTGetHardFC	13
22. dsoGetFPGAVersion	13
23. ddsSetOnOff	13

24.	ddsSetCmd.....	13
25.	ddsEmitSingle	14
26.	ddsSDKSetFre	14
27.	ddsSDKSetAmp	14
28.	ddsSDKSetOffset	14
29.	ddsSDKSetBurstNum.....	15
30.	ddsSDKSetWaveType	15
31.	ddsSDKSetWavePhase	15
32.	ddsSDKSetWaveDuty	15
33.	dsoInitHard	16
34.	dsoHTADCCHModGain	16
35.	dsoHTSetAmpCalibrate.....	16
36.	dsoHTSetRamAndTrigerControl	17
37.	dsoHTSetTrigerMode.....	17
38.	dsoHTSetVideoTriger.....	18
39.	dsoHTSetPulseTriger.....	19
四、控制流程图如下:		20
表 1: 时基与索引值的对应关系		21
表 2: 时基与通道模式		21
表 3: 电压档位与索引值的对应关系		22
表 4: 常用示波器设置		22
采集数据电压值计算		23
DDS 输出示例		23
1.	输出正弦波	23
2.	输出非连续正弦波	23
3.	输出方波	24

一、说明

所有 DLL 在 VC++ 6.0 环境下编译生成。所以数据类型符合 VC++ 6.0 标准。

WORD : unsigned short, 无符号 16bit 整型, 两个字节

BOOL : 布尔类型, 32bit 四个字节。

ULONG: unsigned long, 无符号 32bit 整型, 四个字节。

MAX_CH_NUM: 4, 详见 DefMacro.h 中定义

此 DLL 中的所有文件都是用命令行上定义的 DLL_API 符号编译的。在使用此 DLL 的任何其他项目上都不应定义此符号。这样, 源文件中包含此文件的任何其他项目都会将 DLL_API 函数视为是从 DLL 导入的。

```
#ifndef DLL_API
#define DLL_API extern "C" __declspec(dllimport)
#endif
```

定义标准调用:

```
#define WIN_API __stdcall
```

二、结构体介绍

结构体 _HT_RELAY_CONTROL 包含了所有控制继电器状态所需要的信息。

```
typedef struct _HT_RELAY_CONTROL
{
    BOOL bCHEnable[MAX_CH_NUM];
    WORD nCHVltDIV[MAX_CH_NUM];
    WORD nCHCoupling[MAX_CH_NUM];
    BOOL bCHBWLimit[MAX_CH_NUM];
    WORD nTrigSource;
    BOOL bTrigFilt;
    WORD nALT;
} RELAYCONTROL, *PRELAYCONTROL;
```

说明:

bCHEnable[MAX_CH_NUM]: 大小为 MAX_CH_NUM(CH 的总数)的数组, 表示 CH 的开/关。取值:
1 为开; 0 为关。

nCHVltDIV[MAX_CH_NUM]: 大小为 MAX_CH_NUM(CH 的总数)的数组, 表示 CH 的电压档位。电压档位以索引值形式表示。以最小电压档位为 0 开始依次递加 1 计算。

nCHCoupling[MAX_CH_NUM]: 大小为 MAX_CH_NUM(CH 的总数)的数组, 表示 CH 的耦合。耦合以索引值形式表示。取值: DC 为 0; AC 为 1;

bCHBWLimit[MAX_CH_NUM]: 大小为 MAX_CH_NUM(CH 的总数)的数组, 表示 CH 的带宽限制。取值: 1 为打开带宽限制; 0 为关闭带宽限制。

nTrigSource: 表示触发源, 以索引值形式取值。假设现在为 4CH 示波器, 则内部触发取值为: CH1 为 0; CH2 为 1; CH3 为 2; CH4 为 3; 如果有外部触发, 则 EXT 为 5; 如果有 EXT/10 触发, 则取值为 6。

bTrigFilt: 表示高频抑制。取值: 1 表示打开高频抑制, 0 表示关闭高频抑制。

nALT: 表示是否交替。取值: 1 为交替, 0 为非交替。

举例：

```
声明一个变量：RELAYCONTROL myRelayControl;  
声明一个指针：PRELAYCONTROL pRelayControl;
```

三、函数介绍

1. dsoHTDeviceConnect

函数声明： DLL_API WORD WINAPI dsoHTDeviceConnect (WORD DeviceIndex)

返回值：

返回仪器连接的情况。0：不连接；非 0：连接。

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

备注：

获取 PC 是否与仪器相连。

程序举例：

```
WORD DeviceIndex = 0;  
//调用函数  
if (0 = dsoHTSearchDevice(DeviceIndex))  
{  
    //不连接  
}  
else  
{  
    //连接中  
}
```

2. dsoHTSearchDevice

函数声明： DLL_API WORD WINAPI dsoHTSearchDevice(short* pDevInfo);

返回值：

与 PC 机连接成功的设备数。

参数：

pDevInfo

short 指针长度为 32，用来遍历 0-31 端口是否有设备，如果第 i 端口有设备返回 pDevInfo[i]==1，无设备 pDevInfo[i]==0。

备注：

遍历端口。

3. dsoHTReadCalibrationData

函数声明： DLL_API WORD WINAPI dsoHTReadCalibrationData (WORD DeviceIndex,
WORD* pLevel,
WORD nLen)

4. dsoHTWriteCalibrationData

函数声明: DLL_API WORD WINAPI dsoHTWriteCalibrationData(WORD DeviceIndex,
WORD* pLevel,
WORD nLen)

返回值:

1: 成功; 2: 失败

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pLevel:

指向一个 WORD 型数组, 用来存放校准数据。

nLen:

WORD 型变量, 表示数组的长度值。大小为(12*12)*4+1, 第一个“12”表示单个通道的单个电压档位长度, 第二个“12”表示总的电压档位数, “4”表示通道数目, “1”是一个标识如果 pLevel[nLen-1]= 0xFBCF 表示设备已经存储了零电平校准数据。

备注:

dsoHTReadCalibrationData 读取, dsoHTWriteCalibrationData 写入零电平校准 获取仪器中存放的校准参数值。

请谨慎调用 **dsoHTWriteCalibrationData** 一般不需调用, 如果在二次开发中零电平不准, 请用光盘或者网站得到的上位机程序进行零电平校正, 二次开发仅需调用读函数即可。

零点平是受温度影响的

程序举例:

```
WORD DeviceIndex = 0;  
WORD pLevel[577];  
WORD nLen = 577;  
if (0 == dsoHTReadCalibrationData (DeviceIndex, pLevel, nLen))  
{  
    //失败  
}  
else  
{  
    //成功  
}
```

5. dsoHTRDAmpCali

函数声明: DLL_API WORD WINAPI dsoHTRDAmpCali(WORD nDeviceIndex,
WORD * pLevel,
WORD nLen)

6. dsoHTWRAmpCali

函数声明: DLL_API WORD WINAPI dsoHTWRAmpCali(WORD nDeviceIndex,
WORD * pLevel,
WORD nLen)

返回值:

1: 成功; 0: 失败

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pLevel:

指向一个 WORD 型的数组, 用于存放触发幅度修正系数。注意实际幅度修正 = pLevel[i]/1024, 是一个 Float 数据。

nLen:

WORD 型变量, 表示数组的大小。大小为(6*12)*4+1, 第一个“6”表示单个通道的单个电压档位长度, 第二个“12”表示总的电压档位数目, “4”表示通道数目, “1”是一个标识。如果 pLevel[nLen-1] = 0xFACF 表示设备已经存储了幅度校准数据。

备注:

dsoHTRDAmpCali 读取, dsoHTWRAmpCali 写入 获取设备中的幅度修正参数。

由于硬件器件精度等原因, 示波器的幅度都要经过幅度修正, 否则幅度可能不准确。

请谨慎调用 dsoHTWRAmpCal 一般不需调用, 二次开发仅需调用读函数即可。

程序举例:

```
WORD DeviceIndex = 0;
WORD pAmpCoe [289];
WORD nLen = 289;
//...
//调用函数
if (0 == dsoHTRDAmpCali (DeviceIndex, pAmpCoe, nLen))
{
    //失败
}
else
{
    //成功
}
```

7. dsoSetUSBBus

函数声明: DLL_API WORD WINAPI dsoSetUSBBus(WORD DeviceIndex)

返回值:

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

备注:

设定总线触发模式, 该项设置为预留使用的, 暂未起到实际控制。

8. dsoHTSetCHPos

函数声明: DLL_API WORD dsoHTSetCHPos(WORD nDeviceIndex,
WORD* pLevel,
WORD nVoltDIV,

WORD nPos,
WORD nCH,
WORD nCHMode)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pLevel

WORD 指针型, 储存零电平校准数据要求长度为 577。从 dsoHTWriteCalibrationData 获得

nVoltDIV

WORD 型变量, 表示当前通道 nCH 电压的索引值。

nPos

WORD 型变量, 表示当前通道 nCH 垂直位置范围 0-255。

nCH

WORD 型变量, 表示当前设置的通道。范围 0~3

nCHMode

WORD 型变量, 表示当前的通道模式 (1、2、4)。通道模式请详见表 2

备注:

设定通道的垂直位置。通道垂直位置的范围为 0~255, “0”表示将通道位置设置到屏幕最下端; “128”将通道设置到屏幕最中间; “255”表示设置到屏幕最上端

9. dsoHTSetVTriggerLevel

函数声明: DLL_API WORD dsoHTSetVTriggerLevel(WORD nDeviceIndex,
WORD nPos,
WORD nSensitivity)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nPos

WORD 型变量, 表示触发的垂直位置, 范围 0-255。

nSensitivity

WORD 型变量, 表示触发灵敏度。触发灵明度越大越不会误触发, 但设置的灵敏度过大, 会导致较小幅度的信号无法触发。

备注:

设定触发的垂直位置。

10. dsoHTSetHTriggerLength

函数声明: DLL_API WORD dsoHTSetHTriggerLength(
WORD nDeviceIndex,
PCONTROLDATA pControl,

WORD nCHMod)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pControl

指向 CONTROLDATA 型变量, 详见 HTSoftD11 中 CONTROLDATA 的定义。

nCHMode

WORD 型变量, 表示当前的通道模式 (1、2、4)。通道模式请详见表 2

备注:

设定水平触发位置, 以及采集深度。

注意此函数设置的是单个通道的采集深度, 范围是 4096–64KB (64*1024), 因为设备的存储数据的内存仅有 64KB, 故仅有一个通道开启才可设置为 64KB, 更具体的说, 假如所开通道数目为 nCHEnable; 设置的采集深度为 nLength, nCHEnable*nLength<=64KB.

11. dsoHTSetCHAndTrigger

函数声明: DLL_API WORD dsoHTSetCHAndTrigger(WORD nDeviceIndex,
PRELAYCONTROL pRelayControl,
WORD nTimeDIV);

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pRelayControl

PRELAYCONTROL 指针详见文件头部。

nTimeDIV

WORD 型变量, 表示当前时基的索引值。

备注:

设定继电器状态。电压档位、通道开关、20M 带宽限制、耦合方式等改变需要调用

12. dsoHTSetCHAndTriggerVB

函数声明: DLL_API WORD dsoHTSetCHAndTriggerVB(WORD nDeviceIndex,
WORD* pCHEnable,
WORD* pCHVltDIV,
WORD* pCHCoupling,
WORD* pCHBWLlimit,
WORD nTriggerSource,
WORD nTriggerFilt,
WORD nALT,
WORD nTimeDIV);

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pCHVltDIV

WORD 型变量指针, 长度为 4, 存放 4 个通道的电压档位索引值。

pCHEnable

WORD 型变量指针, 长度为 4, 存放 4 个通道的开关状况 1 表示开, 0 表示关。

pCHCoupling

WORD 型变量指针, 长度为 4, 存放 4 个通道的输入耦合状况 0: DC, 1: AC。

pCHBWLimit

WORD 型变量指针, 长度为 4, 存放 4 个通道的带宽限制。

nTriggerSource, nTriggerFilt, nALT

WORD 型变量, 与结构体 RELAYCONTROL 同名变量相同。

nTimeDIV

WORD 型变量, 表示当前时基的索引值。

备注:

此接口为 c# 提供的, dsoHTSetCHAndTrigger 的变形, 在 DLL 中还是调用 dsoHTSetCHAndTrigger, 用来设定继电器状态。

13. dsoHTSetSampleRate

函数声明: DLL_API WORD WINAPI dsoHTSetSampleRate(WORD nDeviceIndex,
WORD *pAmpLevel,
WORD nYTFormat,
PRELAYCONTROL pRelayControl,
PCONTROLDATA pControl)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pAmpLevel,

WORD 型变量指针, 长度为 289, 存放幅度校准数据存放 289 个数度数据。

nYTFormat

WORD 型变量, 表示当前的 YT 模式 0: NORMAL 1: SCAN 2: ROLL。

pControl

指向 CONTROLDATA 型变量, 详见 HTSoftD11 中 CONTROLDATA 的定义。

pRelayControl

指向 RELAYCONTROL 型变量, 详见 RELAYCONTROL 的定义。

备注:

设定 FGPA 的采样率。

14. dsoHTSetSampleRateVi

函数声明: DLL_API WORD dsoHTSetSampleRateVi(WORD nDeviceIndex,
WORD *pAmpLevel,

```
WORD* pCHEnable,
WORD* pCHVltDIV,
WORD* pCHCoupling,
WORD* pCHBWLimit,
WORD nTriggerSource,
WORD nTriggerFilt,
WORD nALT,
PCONTROLDATA pControl);
```

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pAmpLevel,

WORD 型变量指针, 长度为 289, 存放幅度校准数据存放 289 个数度数据。

pCHVltDIV

WORD 型变量指针, 长度为 4, 存放 4 个通道的电压档位索引值。

pCHEnable

WORD 型变量指针, 长度为 4, 存放 4 个通道的开关状况 1 表示开, 0 表示关。

pCHCoupling

WORD 型变量指针, 长度为 4, 存放 4 个通道的输入耦合状况 0DC, 1 AC。

pCHBWLimit

WORD 型变量指针, 长度为 4, 存放 4 个通道的带宽限制。

nTriggerSource, nTriggerFilt, nALT

WORD 型变量, 与结构体 RELAYCONTROL 同名变量相同。

pControl

指向 CONTROLDATA 型变量, 详见 HTSoftD11 中 CONTROLDATA 的定义。

备注:

dsoHTSetSampleRate 的变形, 设定 FGPA 的采样率。

15. dsoHTStartCollectData

函数声明: DLL_API WORD WINAPI dsoHTStartCollectData(WORD nDeviceIndex,
WORD nStartControl)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nStartControl

WORD 型变量, 表示开始采集的模式。WORD 数据一共 8bit, 0:0 为 AUTO 1 为 其他为 0;

1: 1 为滚动模式 0 为正常模式

2: 1 为 采集完一次停止, 0 为正常模式

备注:

开始数据采集。

16. dsoHTGetState

函数声明: DLL_API WORD WINAPI dsoHTGetState(WORD nDeviceIndex);

返回值:

WORD 类型 8bit

0: 1 有触发, 0 无触发

1: 1 采集结束, 0 采集未结束

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

备注:

获取采集状态。仅当采集结束才可读取采集数据

17. dsoHTGetData

函数声明: DLL_API WORD WINAPI dsoHTGetData(WORD nDeviceIndex,
WORD* pCH1Data,
WORD* pCH2Data,
WORD* pCH3Data,
WORD* pCH4Data,
PCONTROLDATA pControl)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pCH%n%Data

WORD 型变量指针, 长度为 pControl-> nReadDataLen, 用来存放通道 n 的采集数据数据范围 0-255

pControl

指向 CONTROLDATA 型变量, 详见 HTSoftD11 中 CONTROLDATA 的定义。

备注:

获取采集模式为 NORMAL 采集数据。数组 pCH1Data 第 i 个点所代表的实际电压值为:
(pCH1Data[i]- 通道垂直位置)×电压档位/32, 比如 dsoHTSetCHAndTrigger 设置通道 1 电压档位索引为 5 (查表 3 得为 100mV), dsoHTSetCHPos 设置通道垂直位置为 128, 第 i 点数据为 65, 侧第 i 点实际电压值为 (65-128)*100mV/32=-197.7mV

18. dsoHTGetScanData

函数声明: DLL_API WORD dsoHTGetScanData(WORD nDeviceIndex,
WORD* pCH1Data,
WORD* pCH2Data,
WORD* pCH3Data,
WORD* pCH4Data,
PCONTROLDATA pControl);

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pCH%n%Data

WORD 型变量指针, 长度为 pControl-> nReadDataLen, 用来存放通道 n 的采集数据数据范围 0-255

pControl

指向 CONTROLDATA 型变量, 详见 HTSoftD11 中 CONTROLDATA 的定义。

备注:

获取采集模式为 SCAN 采集数据。

pControl->nLastAddress 用来记录上一次采集结束地址

pControl->nAlready 用来记录在一帧的采集中已经采的长度

19. dsoHTGetRollData

函数声明: DLL_API WORD dsoHTGetRollData(WORD nDeviceIndex,
WORD* pCH1Data,
WORD* pCH2Data,
WORD* pCH3Data,
WORD* pCH4Data,
PCONTROLDATA pControl)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pCH%n%Data

WORD 型变量指针, 长度为 pControl-> nReadDataLen, 用来存放通道 n 的采集数据数据范围 0-255

pControl

指向 CONTROLDATA 型变量, 详见 HTSoftD11 中 CONTROLDATA 的定义。

备注:

获取采集模式为 ROLL 采集数据。

pControl->nLastAddress 用来记录上一次采集结束地址

20. dsoHTSetHardFC

函数声明: DLL_API WORD dsoHTSetHardFC(WORD nDeviceIndex,
ULONG nTime)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nTime

ULONG 类型变量，下发的数据 nSend，时间长度单位 nS，长度越大精度越高反应越慢

备注：
请结合 dsoHTGetHardFC 查看。

21. dsoHTGetHardFC

函数声明： DLL_API ULONG dsoHTGetHardFC(WORD nDeviceIndex)

返回值：

0：失败，非零：成功

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

备注：

计算的频率= nIndata*1E9/(8*nSend);

22. dsoGetFPGAVersion

函数声明： DLL_API WORD dsoGetFPGAVersion(WORD DeviceIndex)

返回值：

FPGA 版本

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

备注：

获取 FPGA 版本。

23. ddsSetOnOff

函数声明： DLL_API WORD ddsSetOnOff(WORD DeviceIndex,
short nOnOff)

返回值：

0：失败，非零：成功

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

nOnOff

short 型变量，1 表示关闭，0 表示开启

备注：

具有 DDS 功能 DDS 开关。

24. ddsSetCmd

函数声明： DLL_API WORD ddsSetCmd(WORD DeviceIndex,
USHORT nData)

返回值：

0：失败，非零：成功

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

nData

WORD 型变量，4 表示非连续，0 表示连续

备注：

设置是否为连续波形。

25. ddsEmitSingle

函数声明： DLL_API WORD ddsEmitSingle(WORD DeviceIndex)

返回值：

0：失败，非零：成功

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

备注：

ddsSetCmd data=4 时，触发一次输出。

26. ddsSDKSetFre

函数声明： DLL_API float WINAPI ddsSDKSetFre(WORD DeviceIndex, float dFre)

返回值：

旧的值

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

dbFre

double 型变量，表示 DDS 的频率

备注：

DDS 频率设置。

27. ddsSDKSetAmp

函数声明： DLL_API WORD WINAPI ddsSDKSetAmp(WORD DeviceIndex, WORD nAmp)

返回值：

旧的值

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

nAmp

WORD 型变量，单位 mV 表示 DDS 的输出的幅度如果 Sinwave nAmp=1000，测波形幅度 1V，Vpp2V

备注：

DDS 幅度设置。

28. ddsSDKSetOffset

函数声明: DLL_API short WINAPI ddsSDKSetOffset(WORD DeviceIndex, short nOffset)

返回值:

旧的值

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nOffset

short 型变量, 表示信号偏移单位为 mV, 如值为 1000, 表示偏移为 1V

备注:

DDS 偏移设置。

29. ddsSDKSetBurstNum

函数声明: DLL_API WORD WINAPI ddsSDKSetBurstNum(WORD DeviceIndex, WORD nBurstNum)

返回值:

旧的值

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nBurstNum

WORD 型变量, 表示非连续波形输出的波形个数

30. ddsSDKSetWaveType

函数声明: DLL_API WORD WINAPI ddsSDKSetWaveType(WORD DeviceIndex, WORD nType)

返回值:

旧的值

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nType

WORD 型变量, 波形类型 0 正弦波; 1 三角波 2 方波; 4 直流; 8 高斯噪声; 9 白噪声

备注:

DDS 波形类型。

31. ddsSDKSetWavePhase

函数声明: DLL_API float WINAPI ddsSDKSetWavePhase(WORD DeviceIndex, float fPhase)

返回值:

旧的值

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

fPhase

表示 DDS 的信号初始相位

32. ddsSDKSetWaveDuty

函数声明: DLL_API float WINAPI ddsSDKSetWaveDuty(WORD DeviceIndex, float fDuty)

返回值:

旧的值

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

fPhase

表示 DDS 的信号的空占空比, 仅对锯齿波, 方波有效

33. dsoInitHard

函数声明: DLL_API WORD dsoInitHard(WORD DeviceIndex)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

备注:

设备初始化。硬件上电连接后需要及时调用

34. dsoHTADCCHModGain

函数声明: DLL_API WORD dsoHTADCCHModGain(WORD DeviceIndex,
WORD nCHMod)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nCHMod

WORD 型变量, 表示当前设备的通道模式。

备注:

通道模式变化时调用。

35. dsoHTSetAmpCalibrate

函数声明: DLL_API WORD dsoHTSetAmpCalibrate(WORD nDeviceIndex,
WORD nCHSet,
WORD nTimeDIV,
WORD *pLevel,
WORD *nVoltDiv,
WORD *pCHPos)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nCHSet

WORD 型变量，定义同 pControl->nCHSet。

nTimeDIV

WORD 型变量，表示时基。

pLevel

WORD 型指针，指向幅度校准数据。

nVoltDiv

WORD 型指针，长度为 4，四个通道的电压档位索引值。

pCHPos

WORD 型指针，长度为 4，四个通道的垂直位置。

备注：

设置幅度修正。

36. dsoHTSetRamAndTrigerControl

函数声明： DLL_API WORD dsoHTSetRamAndTrigerControl(WORD DeviceIndex,
WORD nTimeDiv,
WORD nCHset,
WORD nTrigerSource,
WORD nPeak)

返回值：

0: 失败，非零：成功

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

nCHSet

WORD 型变量，定义同 pControl->nCHSet。

nTimeDIV

WORD 型变量，表示时基。

nTrigerSource

WORD 型变量，表示触发通道。

nPeak

WORD 型变量，表示是否开启峰值采样。1：开启 0 关闭

备注：

设置触发源。

37. dsoHTSetTrigerMode

函数声明： DLL_API WORD dsoHTSetTrigerMode(WORD m_nDeviceIndex,
WORD nTriggerMode,
WORD nTriggerSlop,
WORD nTriggerCouple)

返回值：

0: 失败，非零：成功

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

nTriggerMode

WORD 型变量，触发模式。0：边沿 1：脉冲 2：视频

nTriggerSlop

WORD 型变量，触发时上升沿还是下降沿。0：上升沿 1：下降沿

nTriggerCouple

WORD 型变量，表示触发耦合方式。0：直流 1：交流 2：低频抑制 3：高频抑制 4：噪声抑制

备注：

设置触发模式。

38. dsoHTSetVideoTriger

函数声明： DLL_API WORD WINAPI dsoHTSetVideoTriger(WORD m_nDeviceIndex,
USHORT nStand,
USHORT nVedioSyncSelect,
USHORT nVideoHsyncNumOption,
USHORT nVideoPositive,
WORD nLevel,
WORD nLogicTriggerSource)

返回值：

0：失败，非零：成功

参数：

DeviceIndex

WORD 型变量，表示当前设备的索引值。

nStand

WORD 型变量，表示视频触发制式 0：PALSECAM 1：NTSC。

nVedioSyncSelect

WORD 型变量，表示视频触发模式下的同步选择，默认值为 0。

值：

所有行： 0

行： 1

奇数场： 2

偶数场： 3

所有场： 4

nVideoHsyncNumOption,

WORD 型变量，表示视频触发的线数值。

nVideoPositive

WORD 型变量，表示触发的极性选择。

nLevel

WORD 型变量，表示触发的实际电平

nLogicTriggerSource

WORD 型变量，表示视频触发的触发源。

备注:

设定视频触发时，调用完 dsoHTSetTrigerMode 需调用此函数。

39. dsoHTSetPulseTriger

函数声明: DLL_API WORD WINAPI dsoHTSetPulseTriger(WORD m_nDeviceIndex,
ULONG nPW,
WORD nPWCondition)

返回值:

0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

nPW

ULONG 型变量, 表示 Pulse 宽度 nS 单位。

nPWCondition

WORD 型变量, 表示 Pulse 触发条件。

值 :

等于 0 .

不等于 1 .

大于 2 .

小于 3 .

备注:

设定 Pulse 触发时，调用完 dsoHTSetTrigerMode 需调用此函数。

四、控制流程图如下：

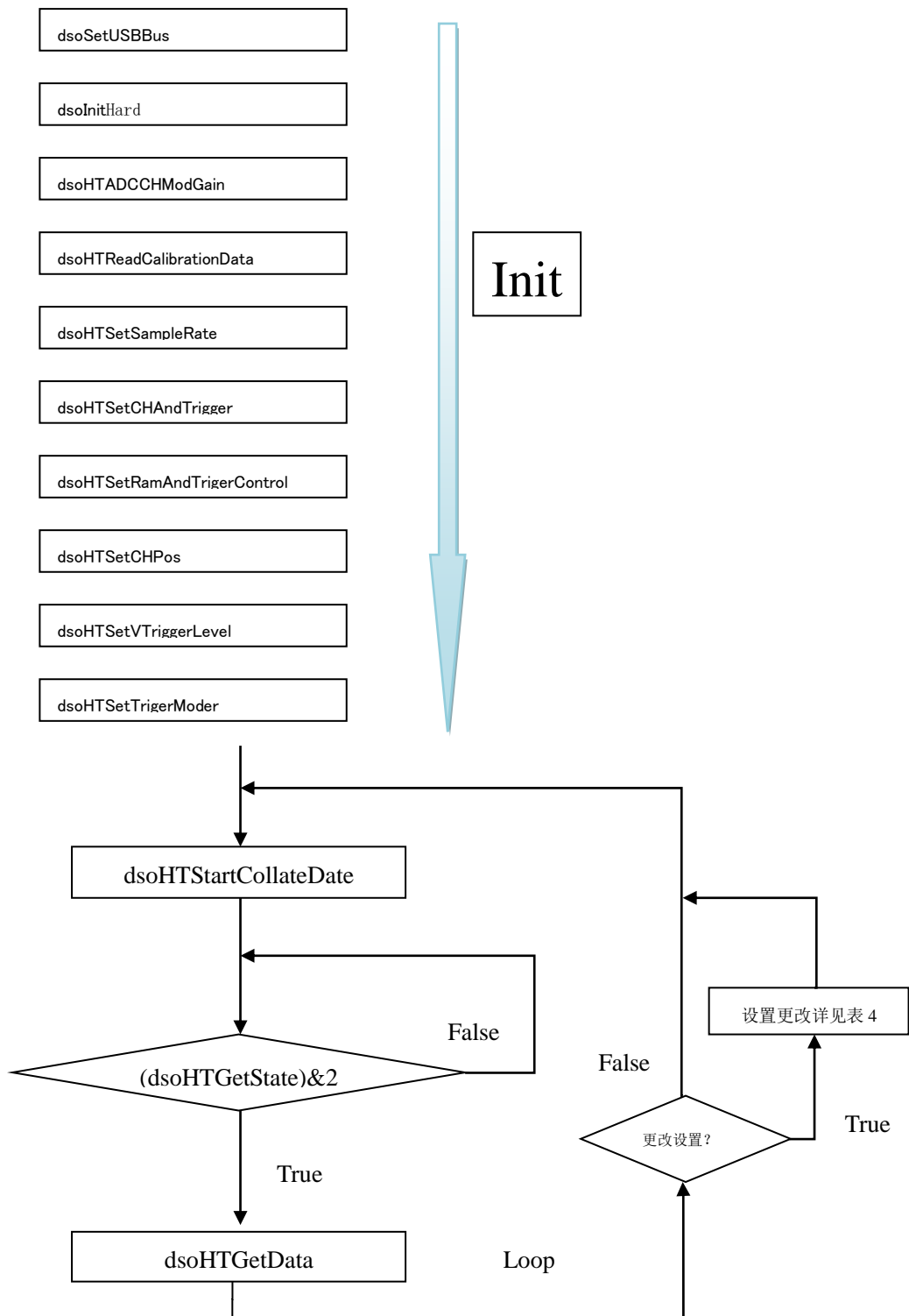


表 1：时基与索引值的对应关系

时基索引	时基值	采样率 (Sa)	时基索引	时基值	采样率 (Sa)
0	2nS	单:1G 双:0.5G 四:250M	19	5mS	50K
1	5nS	单:1G 双:0.5G 四:250M	20	10mS	25K
2	10nS	单:1G 双:0.5G 四:250M	21	20mS	12.5K
3	20nS	单:1G 双:0.5G 四:250M	22	50mS	5K
4	50nS	单:1G 双:0.5G 四:250M	23	100mS	2.5K
5	100nS	单:1G 双:0.5G 四:250M	24	200mS	1.25K
6	200nS	单:1G 双:0.5G 四:250M	25	500mS	500
7	500nS	单双: 0.5G 四:250M	26	1S	250
8	1uS	250M	27	2S	125
9	2uS	125M	28	5S	50
10	5uS	50M	29	10S	25
11	10uS	25M	30	20S	12.5
12	20uS	12.5M	31	50S	5
13	50uS	5M	32	100S	2.5
14	100uS	2.5M	33	200S	1.25
15	200uS	1.25M	34	500S	0.5
16	500uS	500K	35	1000S	0.25
17	1mS	250K			
18	2mS	125K			

注意：

1 采样率列中粗体表示需要插值

2 当示波器不需要插值时，采样率=250/时基；其中 250 是水平一个大格的采集点数，例如时基为 1uS 时采样率=250/(1e-6)=250MSa

3 “单”表示单通道模式；“双”表示双通道模式；“四”表示四通道模式。关于通道模式详见表：时基与通道模式

表 2：时基与通道模式

时基索引	时基值	通道开启数目/通道模式
0	2nS	1/单；2/双；3、4/四
...	...	1/单；2/双；3、4/四
6	200nS	1/单；2/双；3、4/四
7	500nS	1、2/双；3、4/四
8	1uS	*/四
...	...	*/四
35	1000S	*/四

注意：

1 例如在 200nS 时基的时候，开启了 1 个通道为单通道模式；开启 2 个通道为双通道模式；开启 3 个和 4 个为四通道模式。

表 3：电压档位与索引值的对应关系

索引	电压档位	量程	索引	电压档位	量程
0	2mV	16mV	6	200mV	1.6V
1	5mV	40mV	7	500mV	4V
2	10mV	80mV	8	1V	8V
3	20mV	160mV	9	2V	16V
4	50mV	400mV	10	5V	40V
5	100mV	800mV	11	10V	80V

注意：

1 “电压档位”表示垂直 1 个大格对应的电压值，更确切的说是表示采集的波形数据 32 个数据差对应的值

2 “量程”表示用 1:1 探头对应的量程，如果用 n 倍衰减探头，测量程扩大 n 倍。例如 100mV 用 1:1 探头量程是 800mV；用 1:10 探头量程是 8V

表 4：常用示波器设置

序号	设置选项	函数调用
1	电压档位	dsoHTSetCHAndTrigger: 必须 dsoHTSetAmpCalibrate: 不调用幅度可能不对
2	时基	dsoHTSetSampleRate: 必须 dsoHTSetRamAndTrigerControl: 必须 SetADCCHModGain: 通道模式改变需下发，不调用幅度不准 SetAmpCalibrate: 通道模式改变需下发，不调用幅度不准
3	通道开启/关闭	dsoHTSetCHAndTrigger: 必须 SetADCCHModGain: 通道模式改变需下发，不调用幅度不准 SetAmpCalibrate: 通道模式改变需下发，不调用幅度不准
4	垂直触发位置	dsoHTSetVTriggerLevel: 必须
5	水平触发位置	dsoHTSetHTriggerLength: 必须
6	带宽限制	dsoHTSetCHAndTrigger: 必须
7	交直流输入耦合	dsoHTSetCHAndTrigger: 必须
8	触发模式	dsoHTSetTrigerMode: 必须 dsoHTSetVideoTriger: 视频触发额外调用 dsoHTSetPulseTriger: 脉宽触发额外调用
9	触发源	dsoHTSetRamAndTrigerControl: 必须
10	触发上升沿/下降沿	dsoHTSetTrigerMode: 必须
11	通道垂直位置	dsoHTSetCHPos 必须 dsoHTSetAmpCalibrate 必须

注意：

1 归根到底设置是改变硬件寄存器所存储的值，因此所有设置即可以重复下发，也可在不需要下发时下发。

采集数据电压值计算

```
WORD pCHData[4][4096]; //申请空间每个通道 4096 个数据
dsoHTGetData(WORD nDeviceIndex, WORD pCHData[0], WORD pCHData[1], WORD
pCHData[2], WORD pCHData[3], PCONTROLDATA pControl); //调用函数进行数据采集
//假设通道 1 的垂直位置是 64;
short pSrcData[4][4096]; //长度与 pCHData 长度一样存放减去基准位置的数据
WORD nPos[4]; //通道的垂直位置
for(int i=0; i<4; i++)
{
    for(int j=0; j<4096; j++)
    {
        pSrcData[i][j] = pCHData[i][j] - nPos[i];
    }
}
```

我们假设计算通道 1 第 1000 个点对应的实际电压值

假设 $nPos[0]=64$; $pCHData[0][999]=50$; 侧 $pSrcData[0][999]=-14$; 假设通道 1 设置的电压档位索引值为“3”，查表 3 得“3”对应 20mV，那么这个点的电压值为 $-14/32*20mV=-8.75mV$ 。故计算通道 i 第 j 个点实际电压公式： $(pCHData[i][j] - nPos[i])/32.0f*$ 索引对应的电压值。

注意：要计算的点 $pCHData[i][j]$ 必须在 [1-254] 之间否则数据超出量程计算的肯定不正确

DDS 输出示例

1. 输出正弦波

频率 10k; 幅度 1.5V; 偏移-0.5V; 连续

```
ddsSetCmd(nIndex, 0);
ddsSDKSetFre(nIndex, 10000);
ddsSDKSetAmp(nDIndex, 1500);
ddsSDKSetOffset(nIndex, -500);
ddsSDKSetWaveType(nIndex, 0);
ddsSetOnOff(nIndex, 1)
```

2. 输出非连续正弦波

频率 10k; 幅度 1.5V; 每隔 10s 输出一次; 每次输出 11 个波形

```
ddsSetCmd(nIndex, 4);
ddsSDKSetFre(nIndex, 10000);
ddsSDKSetAmp(nDIndex, 1500);
ddsSDKSetBurstNum(nIndex, 11);
ddsSDKSetWaveType(nIndex, 0);
ddsSetOnOff(nIndex, 1);
while(true) {
    sleep(10S);
}
```

```
        ddsEmitSingle(nIndex);  
    }
```

3. 输出方波

频率 1k; 幅度 2V; 占空比 0.3

```
ddsSetCmd(nIndex, 0);  
ddsSDKSetFre(nIndex, 1000);  
ddsSDKSetAmp(nIndex, 2000);  
ddsSDKSetWaveDuty(nIndex, 0.3);  
ddsSDKSetWaveType(nIndex, 2);  
ddsSetOnOff(nIndex, 1);
```