

SDK - HTHardD11.dll 说明文档

目录

一、说明..... 3

二、函数介绍..... 3

1. dsoHTDeviceConnect..... 3

2. dsoSetUSBBus..... 3

3. dsoHTAttribsCHPos 4

4. dsoHTAttribsCHEnable..... 4

5. dsoHTAttribsCHCouple..... 5

6. dsoHTAttribsCHVolt..... 5

7. dsoHTAttribsVTriggerPos 5

8. dsoHTAttribsHTriggerPos 6

9. dsoHTSetSampleRateDiv..... 6

10. dsoHTSetSampleRate..... 6

11. dsoHTStartCollectData..... 7

12. dsoHTGetState 7

13. dsoHTGetData..... 8

14. dsoHTAttribsBufferLength..... 8

15. dsoInitHard 8

16. dsoHTAttribsTriggerMode..... 9

17. dsoHTAttribsTriggerSource..... 9

18. dsoHTAttribsTriggerSlop 9

19. dsoHTAttribs AcqOption 10

三、控制流程图如下： 11

表 1：时基与索引值的对应关系..... 12

表 3：电压档位与索引值的对应关系..... 12

表 4：常用示波器设置..... 13

采集数据电压值计算..... 13

一、说明

所有 DLL 在 Vs2015 环境下编译生成。

WORD : 无符号 16bit 整型, 两个字节

BOOL : 布尔类型, 32bit 四个字节。

ULONG: 无符号 32bit 整型, 四个字节。

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
```

char bGet 所有含有 **bGet** 选项的函数 当 **bGet=1** 的时候表示从设备获取值, 0 表示设置

二、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
{
    //有设备
}
```

2. 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。

3. dsoHTAttribsCHPos

函数声明: DLL_API WORD dsoHTAttribsCHPos(WORD nDeviceIndex,
char nCH,
WORD *nPos,
char bGet = 1)

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

nCH

通道 0-3

nPos

WORD 型变量，表示当前通道 nCH 垂直位置范围 0-255。

nCH

char 型变量，表示当前设置的通道。范围 0~3

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

4. dsoHTAttribsCHEnable

函数声明: DLL_API WORD dsoHTAttribsCHEnable (WORD nDeviceIndex,
char nCH,
char *bEnable,
char bGet = 1)

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

nCH

通道 0-3

bEanbale

char 型变量，表示当前通道 1 打开通道 0 关闭通道。

nCH

char 型变量，表示当前设置的通道。范围 0~3

备注:

设定通道的开关

5. dsoHTAttribsCHCouple

函数声明: DLL_API WORD dsoHTAttribsCHCouple (WORD nDeviceIndex,
char nCH,
char *nCouple,
char bGet = 1)

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

nCH

通道 0-3

nCouple

char 型变量, 表示通道耦合方式 0 为 DC1 为 AC。

nCH

char 型变量, 表示当前设置的通道。范围 0~3

备注:

设定通道的耦合方式

6. dsoHTAttribsCHVolt

函数声明: DLL_API WORD dsoHTAttribsCHVolt (WORD nDeviceIndex,
char nCH, float *fVolt, char bGet, char nOption=0)

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

nCH

通道 0-3

fVolt

表示可以设置的通道电压单位是 mV。

nCH

char 型变量, 表示当前设置的通道。范围 0~3

备注:

设定通道的电压档位

7. dsoHTAttribsVTriggerPos

函数声明: DLL_API WORD dsoHTAttribsVTriggerPos (WORD nDeviceIndex,
WORD nCH,
char bGet,
WORD *nPos1,

WORD *nPos2=NULL)

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。

nPos1

WORD 型变量, 表示触发的垂直位置, 范围 0-255。

nCH

WORD 型变量, 表示当前设置的通道。范围 0~3

nPos2

WORD 型变量, 二次开发设置成 NULL。

备注:

设定触发的垂直位置。

8. dsoHTAttribsHTriggerPos

函数声明: DLL_API WORD dsoHTAttribsHTriggerPos(WORD nDeviceIndex, char *nPos, char bGet = 1)

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。

nPos1

char 型变量, 表示水平触发的垂直位置, 范围 0-100。

备注:

设定水平触发位置, 比如当 nPos=50 时 $50/100=0.5$ 表水平触发在中央。

9. dsoHTSetSampleRateDiv

函数声明: DLL_API WORD WINAPI dsoHTSetSampleRateDiv(WORD nDeviceIndex, 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。

nTimeDIV,

WORD 型变量, 表示用常用的几个时基值间接设置采样率即这个函数内部调用了 dsoHTSetSampleRate。实际上这个函数设置了预设的几个采样率具体见表 1

备注:

设定 FGPA 的采样率。

10. dsoHTSetSampleRate

函数声明: DLL_API WORD dsoHTSetSampleRate(WORD nDeviceIndex, double *fTimeDiv)

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。

fTimeDiv,

double 型变量指针, 用来设置采样率 注意并不是所有采样率都可以设置, 实际设计的采样率以指针返回为准。

备注:

设定 FGPA 的采样率。

11. 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。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
2,3: 采集模式

备注:

没有特殊要求 nStartControl 直接设置为 5 即可。

12. dsoHTGetState

函数声明: DLL_API WORD WINAPI dsoHTGetState(WORD nDeviceIndex, PCHAR pdate)

返回值:

类型 0: 失败, 非零: 成功

参数:

DeviceIndex

WORD 型变量, 表示当前设备的索引值。

pDate

无符号 char 数据指针, 其长度为 128, 二次开发只需要了解第 0Byte 含义即可

pdate[0]&1=1 表示示波器有触发 =0 表示无触发

pdate[0]&2=2 表示示波器采集结束 =0 表示未采集完, 平常二次开发使用判断

pdate[0]&2==2 即可采集数据

备注:

获取采集状态。仅当采集结束才可读取采集数据

13. dsoHTGetData

函数声明: DLL_API WORD WINAPI dsoHTCollectDataWave(WORD nDeviceIndex,
WORD* pCH1Data,
WORD* pCH2Data,
WORD* pCH3Data,
WORD* pCH4Data,
ULONG nReadLen=0)

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%nData

WORD 型变量指针, 用来存放通道 n 的采集数据数据范围 0-255。长度: 当 nReadlen 大于 0 时长度为 nReadLen; 当 nReadLen 为 0 是长度为 dsoHTAttribsBufferLength 设置的内存长度值

nReadLen

ULONG 变量, 表示采集的长度不可以大于 dsoHTAttribsBufferLength 设置的长度, 当其为 0 时表示用读取全部

备注:

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/25, 比如设置通道 1 电压档位索引为 5(查表 3 得为 20mV), 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) *20mV/25=-50.4mV

14. dsoHTAttribsBufferLength

函数声明: DLL_API WORD dsoHTAttribsBufferLength(WORD nDeviceIndex,
ULONG *nBufferLength,
char bGet=1)

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。

nBufferLength

ULONG 型变量指针, 单个通道的数据长度

备注:

注意此函数设置的是单个通道的采集深度, 范围是 2.5K-64M(64*1024*1024), 因为设备的存储数据的内存仅有 64MB, 故仅有一个通道开启才可设置为 64MB, 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. 注意 nCHEnable=3 时其实是开了 4 个通道。

15. 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

16. dsoHTAttribsTriggerMode

函数声明: DLL_API WORD dsoHTAttribsTriggerMode(WORD nDeviceIndex, char *nTriggerMode, char bGet = 1)

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

char 型变量, 触发模式。0: 边沿 1: 脉冲 2: 视频 17: 总线触发

备注:

设置触发模式。

17. dsoHTAttribsTriggerSource

函数声明: DLL_API WORD dsoHTAttribsTriggerMode(WORD nDeviceIndex, char *nTriggerSource, char bGet = 1)

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。

nTriggerSource

char 型变量, 触发源范围 0-3

18. dsoHTAttribsTriggerSlop

函数声明: DLL_API WORD dsoHTAttribsTriggerSlop(WORD nDeviceIndex, char *nTriggerSlop, char bGet = 1)

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。

nTriggerSlop

char 型变量, 0 表示上升沿 1 表示下降沿

备注：

设置触发源通道。

19. dsoHTAttribs AcqOption

函数声明： DLL_API WORD dsoHTAttribsAcqOption (WORD nDeviceIndex, char * nOption, char bGet = 1)

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。

nOption

char 型变量，表示采集模式无特殊要求设置成 5

备注：

设置采集模式。

三、控制流程图如下：

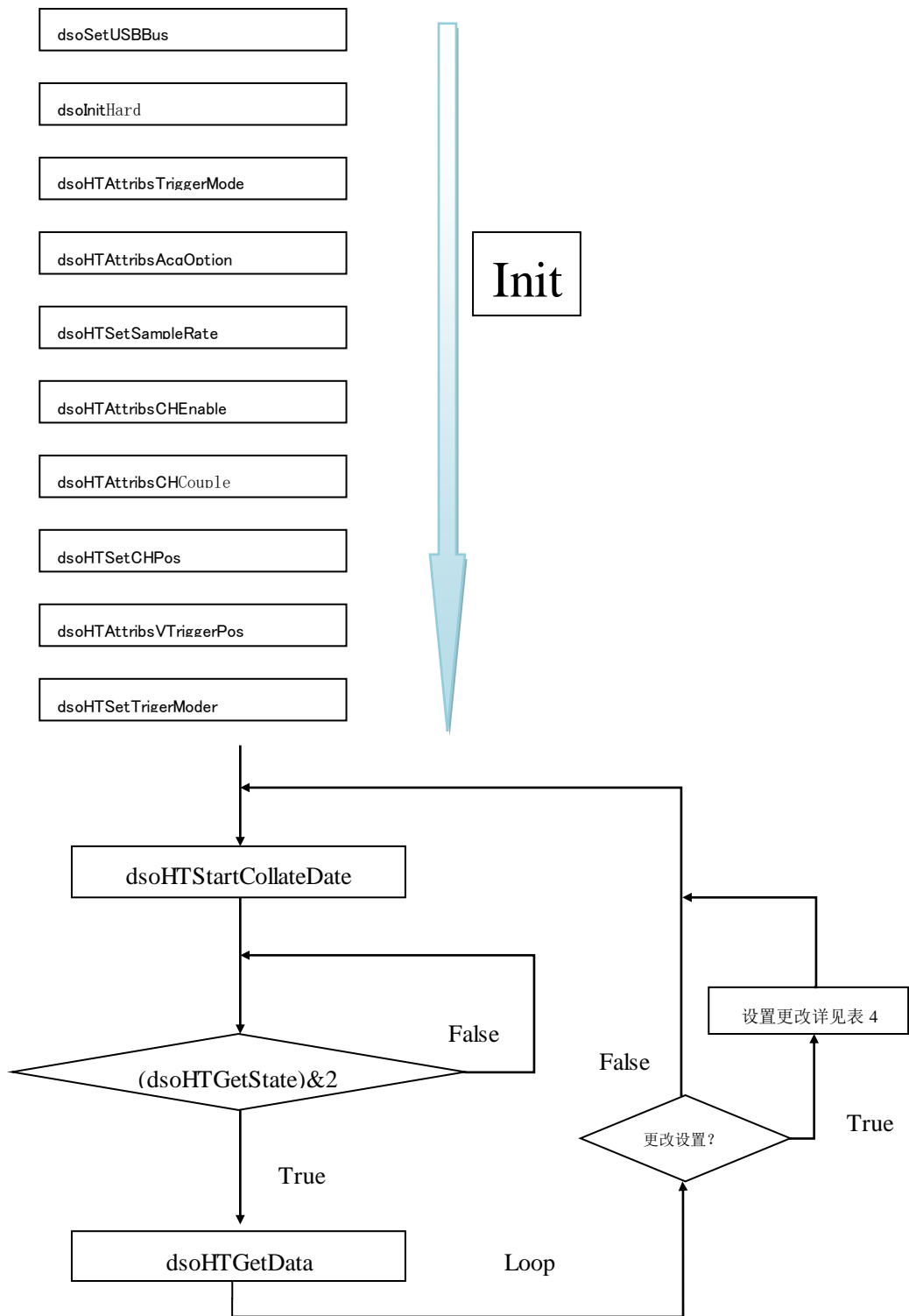


表 1：时基与索引值的对应关系

| 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”表示开一个通道；“双”表示开 2 个通道；“四”表示开 3-4 个通道。

4 此表只是表示用 dsoHTSetSampleRateDiv 设置采样率时使用输入参数时时基索引值可以直接使用 dsoHTSetSampleRate 函数设置采样率

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 | 0.5mV | 4mV | 7 | 100mV | 800mV |
| 1 | 1mV | 8mV | 8 | 200mV | 1.6V |
| 2 | 2mV | 16mV | 9 | 500mV | 4V |
| 3 | 5mV | 40mV | 10 | 1V | 8V |
| 4 | 10mV | 80mV | 11 | 2V | 16V |
| 5 | 20mV | 160mV | 12 | 5V | 40V |
| 6 | 50mV | 400mV | 13 | 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 | dsoHTAttribsCHVolt |
| 2 | 采样率 | dsoHTSetSampleRateDiv: 或者 dsoHTSetSampleRateDiv |
| 3 | 通道开启/关闭 | dsoHTAttribsCHEnable |
| 4 | 垂直触发位置 | dsoHTAttribsVTriggerPos |
| 5 | 水平触发位置 | dsoHTAttribsHTriggerPos |
| 6 | 带宽限制 | dsoHTAttribsCHBW |
| 7 | 交直流输入耦合 | dsoHTAttribsCHCouple |
| 8 | 触发模式 | dsoHTAttribsTriggerMode |
| 9 | 触发源 | dsoHTAttribsTriggerSource |
| 10 | 触发上升沿/下降沿 | dsoHTAttribsTriggerSlop |
| 11 | 通道垂直位置 | dsoHTAttribsCHpos |
| | | |

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
dsoHTGetDataWave(WORD nDeviceIndex, WORD pCHData[0], WORD pCHData[1], WORD
pCHData[2], WORD pCHData[3]); //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; 则 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 5mV，那么这个点的电压值为 $-14/25*5\text{mV}=-2.8\text{mV}$ 。
故计算通道 i 第 j 个点实际电压公式： $(\text{pCHData}[i][j]-\text{nPos}[i])/25.0f*$ 索引对应的电压值。

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