SDK - HTHardDll.dll 说明文档

阅读须知：

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，四个字节。

ULONG：unsigned long，无符号 32bit 整型，八个字节。

MAX_CH_NUM：见在 DefMacro 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
```

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 nCHVoltDIV[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。

nCHVoltDIV[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;

函数介绍

1. **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. **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]!=0。

备注：

遍历端口。

程序举例：

```
short DevInfo[32];
WORD nConnectedDevNum = 0;
//DevInfo Initial
//…
//Call this function
nConnectedDevNum = dsoHTSearchDevice(DevInfo);
```

When DevInfo[n] is 0 , there are devices. When DevInfo[n] is -1, there is no device.

3. **函数声明：**

```
DLL_API WORD WINAPI dsoHTReadCalibrationData(
    WORD DeviceIndex,
    WORD* pLevel,
    WORD nLen
    )
```

4. **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。

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。

程序举例：

```
WORD DeviceIndex = 0;
WORD pLevel[317];
WORD nLen = 317;
//…
//调用函数
if（0 = dsoHTReadCalibrationData (DeviceIndex, pLevel, nLen)）
{
    //失败
```

```
}
else
{
    //成功
}
```

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

6. **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

nLen：

WORD 型变量，表示数组的大小

备注：

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

程序举例:

```
WORD DeviceIndex = 0;
WORD pAmpCoe [317];
WORD nLen = 317;
//…
//调用函数
if（0 = dsoHTRDAmpCali (DeviceIndex, pAmpCoe, nLen)）
{
    //失败
}
else
{
    //成功
}
```

7. **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，暂未起到时基控制。

8. **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至少为 577。

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。

nCHMode

WORD 型变量，表示当前的通道模式（1、2、4）。

备注：

设定通道的垂直位置。

9. **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。

备注：

设定触发的垂直位置。

10. **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 HTSoftDll 中 CONTROLDATA 的定义。

nCHMode

WORD 型变量，表示当前的通道模式（1、2、4）。

备注：

设定水平触发位置，以及预触发触发后长度。

11. **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。

12. **函数声明：**DLL_API WORD dsoHTSetCHAndTriggerVB(WORD nDeviceIndex,
WORD* pCHEnable,
WORD* pCHVoltDIV,
WORD* pCHCoupling,
WORD* pCHBWLimit,
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。

pCHVoltDIV

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：

dsoHTSetCHAndTrigger 的变形，用来设定继电器状态。

13. **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 HTSoftDll 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. **函数声明：**DLL_API WORD dsoHTSetSampleRateVi(WORD nDeviceIndex,
WORD *pAmpLevel,
WORD* pCHEnable,
WORD* pCHVoltDIV,

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。

pCHVoltDIV

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 HTSoftDll 中 CONTROLDATA 的定义。

备注：

dsoHTSetSampleRate 的变形，设定 FGPA 的采样率。

15. **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. **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。

17. **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 HTSoftDll 中 CONTROLDATA 的定义。

备注：

获取采集模式为 NORMAL 采集数据。

18. **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 HTSoftDll 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. **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 HTSoftDll 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. **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. **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. **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 版本储存到 pControl->nFPGAVersion。

23. **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. **函数声明：**DLL_API WORD ddsSetCmd(WORD DeviceIndex,
USHORT nSingle)

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。

nSingle

WORD 型变量，1 表示单次，0 表示连续

备注：

设置是否为单次触发。

25. **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：

单次触发时，触发一次波形。

26. **函数声明**：DLL_API WORD ddsSetFrequency(WORD DeviceIndex,
double dbFre,
WORD* pWaveNum,
WORD* pPeriodNum)

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。

dbFre

double 型变量，表示 DDS 的频率

pWaveNum

WORD 型指针，为返回值 ，总共的数据点数

pPeriodNum

WORD 型指针，为返回值 ，总共的周期数

备注：

DDS 频率设置。

27. **函数声明**：DLL_API WORD ddsDownload(WORD DeviceIndex,
WORD iWaveNum,
WORD* pData)

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。

iWaveNum

WORD 型变量，ddsSetFrequency 返回值 ，总共的数据点数

pData

WORD 型指针，指向存储 DDS 下发值 Buffer 的指针

备注：

DDS 数据下发。

28. **函数声明**：DLL_API WORD dsoInitADCOnce(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。

29. **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。

30. **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。

31. **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。

32. **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。

33. **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。

34. **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 .

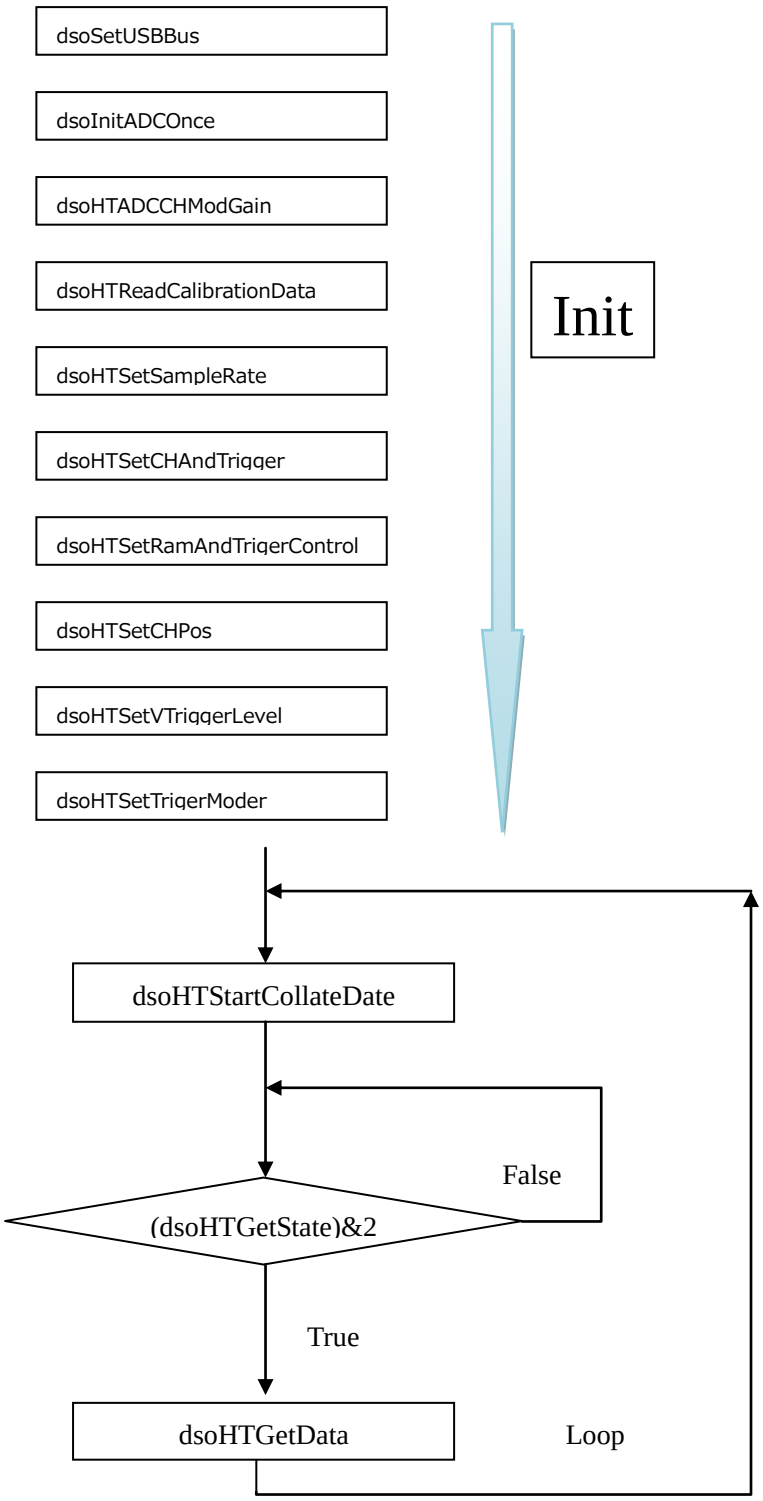
大于 2 .

小于 3 .

备注：

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

控制流程图如下：


dsoSetUSBBus
dsoInitADCOnce
dsoHTADCCHModGain
dsoHTReadCalibrationData
dsoHTSetSampleRate
dsoHTSetCHAndTrigger
dsoHTSetRamAndTrigerControl
dsoHTSetCHPos
dsoHTSetVTriggerLevel
dsoHTSetTrigerModer
Init
dsoHTStartCollateDate
False
(dsoHTGetState)&2
True
dsoHTGetData
Loop