

SDK - HTHardD11.dll 说明文档

目录

| | |
|---|---|
| 一、说明..... | 3 |
| 二、参数说明..... | 3 |
| 三、函数介绍..... | 3 |
| 1. PreMeas: 获取 pMaxData, pMinData | 3 |
| 2. FindPeriod: 获取 PeriodInfo | 4 |
| 3. CalFrequency: 计算频率 | 4 |
| 4. CalPeriod: 计算周期..... | 4 |
| 5. CalRiseTime: 上升时间..... | 4 |
| 6. CalFallTime: 下降时间..... | 5 |
| 7. CalPDutyCycle: 正占空比 | 5 |
| 8. CalNDutyCycle: 负占空比 | 5 |
| 9. CalPPulseWidth: 正脉宽..... | 5 |
| 10. CalNPulseWidth: 负脉宽 | 5 |
| 11. CalMaxVolt: 最大电压..... | 5 |
| 12. CalMinVolt: 最小电压 | 6 |
| 13. CalVpp: 峰峰值..... | 6 |
| 14. CalTopVolt: 顶部值 | 6 |
| 15. CalBaseVolt: 底部值 | 6 |
| 16. CalMidVolt: 中间值 | 6 |
| 17. CalRMS: 有效值..... | 7 |
| 18. CalCRMS: 周期有效值..... | 7 |
| 19. CalAmplitude: 幅度值 | 7 |
| 20. CalMean: 平均值 | 7 |
| 21. CalCMean: 周期平均值 | 7 |
| 22. CalPreShoot: 预冲 | 7 |
| 23. CalOverShoot: 过冲 | 8 |

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。

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

```
#define WIN_API __stdcall
```

注意, 所有函数计算结果只供参考, 如果计算结果有出入, 请自行编写函数

使用这些库函数请保证采集的波形能够至少完整的显示一个周期, 具体是

A 波形最上部分没有超出屏幕外 (采集上来的通道源数据 pCH1Data 所有数据小于 255)

B 波形最下部分没有超出屏幕外 (采集上来的通道源数据 pCH1Data 所有数据大于 0)

C 采集波形数据必须至少包含一个周期的信号

二、参数说明

pMinData: short 指针类型, 最小数据信息. (需要外部申请空间, 长度最小为 5)

pMaxData: short 指针类型, 最大数据信息. (需要外部申请空间, 长度最小为 5)

pSrcData: short 指针类型, 所要计算的数据. 注意这个数据是减去通道位置的值, 也就是说假如从 dll 获取的数组是 pData 要重新见一个数组数组是 pData 每个值都减去这个通道的垂直位置 (见 PreMeas 示例)

nSrcDataLen: ULONG 源数据的长度.

nMaxData: short 数据上限. 8Bit 精度, 直接设置为 255 即可

PeriodInfo: ULONG 指针类型, 周期信息. (需要外部申请空间, 长度为 3)

dbVoltDIV: double 实际电压档位. 比如电压档位为 3 侧这个值为 0.02 (V)

dbTimeInterval: double 数据两点之间的时间长度=1.0/dbSamplerate; dbSamplerate 即为采样率, 计算方式为 250/(时基值), 比如设置时基索引值为 15, 查表得对应时基值为 200uS, dbSamplerate=250/200uS=1.25Msa, dbTimeInterval=1/1.25Msa=0.8uS 计算时带入值为 0.8e-6;

注意, pMinData 与 pMaxData 从 PreMeas 获得, 这 2 个参数在其他函数都是输入参数; PeriodInfo 从 FindPeriod 获得, 这个参数在其他函数都是输入参数;

三、函数介绍

1. PreMeas: 获取 pMaxData, pMinData

函数声明: DLL_API void WINAPI PreMeas (short* pMaxData, short* pMinData, const short* pSrcData, ULONG nSrcDataLen);

参数:

备注:

这个函数获取了 pMaxData, pMinData。这 2 个参数是很多其他计算的输入参数

程序举例:

计算通道 1 的 pMaxData, pMinData。

```
short pMaxData[5], pMinData[5];
```

```
dsoHTGetData(WORD nDeviceIndex, WORD* pCH1Data, WORD* pCH2Data, WORD*
```

```

pCH3Data, WORD* pCH4Data, PCONTROLDATA pControl);
//假设通道 1 的垂直位置是 64 长度是 4096;
short pSrcData[4096]; //长度与 pCH1Data 长度一样
for(int i=0; i<4096; i++)
{
    pSrcData[i] = pCH1Data[i]-64; //参数 pSrcData 的获取
}
PreMeas(pMaxData, pMinData, pSrcData, 4096);

```

2. FindPeriod: 获取 PeriodInfo

函数声明: DLL_API void WINAPI FindPeriod(ULONG* PeriodInfo, const short* pMaxData, const short* pMinData, const short* pSrcData, ULONG nSrcDataLen);

参数:

备注:

计算周期频率用到的输入参数 PeriodInfo。

3. CalFrequency: 计算频率

函数声明: DLL_API double WINAPI CalFrequency(const ULONG* PeriodInfo, double dbTimeInterval);

返回值:

获取频率

4. CalPeriod: 计算周期

函数声明: DLL_API double WINAPI CalPeriod(const ULONG* PeriodInfo, double dbTimeInterval);

返回值:

获取周期

5. CalRiseTime: 上升时间

函数声明: DLL_API double WINAPI CalRiseTime(const short* pMaxData,
const short* pMinData,
const short* pSrcData,
ULONG nSrcDataLen,
double dbTimeInterval,
float fTop,
float fBottom);

返回值:

上升时间

参数:

fTop: fTop, 此值为 0.9.

fBottom: fBottom, 此值为 0.1.

6. CalFallTime: 下降时间

函数声明: DLL_API double WINAPI CalFallTime(const short* pMaxData,
const short* pMinData,
const short* pSrcData,
ULONG nSrcDataLen,
double dbTimeInterval,
float fTop,
float fBottom);

返回值:

下降时间

参数:

fTop: fTop, 此值为 0.9.

fBottom: fBottom, 此值为 0.1.

7. CalPDutyCycle: 正占空比

函数声明: DLL_API double WINAPI CalPDutyCycle(const ULONG* PeriodInfo);

返回值:

正占空比

8. CalNDutyCycle: 负占空比

函数声明: DLL_API double WINAPI CalNDutyCycle(const ULONG* PeriodInfo);

返回值:

负占空比

9. CalPPulseWidth: 正脉宽

函数声明: DLL_API double WINAPI CalPPulseWidth(const ULONG* PeriodInfo, double dbTimeInterval);

返回值:

正脉宽

10. CalNPulseWidth: 负脉宽

函数声明: DLL_API double WINAPI CalNPulseWidth(const ULONG* PeriodInfo, double dbTimeInterval);

返回值:

负脉宽

11. CalMaxVolt: 最大电压

函数声明: DLL_API double WINAPI CalMaxVolt(const short* pMaxData,
double dbVoltDIV,
short nMaxData)

返回值:

最大电压

参数:

dbVoltDIV: 实际电压档位. 比如电压档为 3 测这个值为 0.02V
nMaxData: 为 255

12. CalMinVolt: 最小电压

函数声明: DLL_API double WINAPI CalMinVolt(const short* pMaxData,
double dbVoltDIV,
short nMaxData)

返回值:

最小电压

参数:

nMinData: 为: 255

13. CalVpp: 峰峰值

函数声明: DLL_API double WINAPI CalVpp(const short* pMaxData, short*
pMinData, double dbVoltDIV, short nMaxData)

返回值:

峰峰值

参数:

14. CalTopVolt: 顶部值

函数声明: DLL_API double WINAPI CalTopVolt(const short* pMaxData, double
dbVoltDIV, short nMaxData)

返回值:

顶部值

参数:

15. CalBaseVolt: 底部值

函数声明: DLL_API double WINAPI CalBaseVolt(const short* pMinData, double
dbVoltDIV, short nMaxData)

返回值:

顶部值

参数:

16. CalMidVolt: 中间值

函数声明: DLL_API double WINAPI CalMidVolt(const short* pMaxData, const short*
pMinData, double dbVoltDIV, short nMaxData)

返回值:

中间值

参数:

17. CalRMS: 有效值

函数声明: `DLL_API double WINAPI CalRMS(const short* pSrcData, ULONG nSrcDataLen, double dbVoltDIV, short nMaxData);`

返回值:

有效值

参数:

18. CalCRMS: 周期有效值

函数声明: `DLL_API double WINAPI CalCRMS(const ULONG* PeriodInfo, const short* pSrcData, double dbVoltDIV, short nMaxData);`

返回值:

一个周期的有效值

参数:

19. CalAmplitude: 幅度值

函数声明: `DLL_API double WINAPI CalAmplitude(const short* pMaxData, const short* pMinData, double dbVoltDIV, short nMaxData);`

返回值:

幅度值

参数:

20. CalMean: 平均值

函数声明: `DLL_API double WINAPI CalMean(const short* pSrcData, ULONG nSrcDataLen, double dbVoltDIV, short nMaxData);`

返回值:

平均值

21. CalCMean: 周期平均值

函数声明: `DLL_API double WINAPI CalCMean(const ULONG* PeriodInfo, const short* pSrcData, double dbVoltDIV, short nMaxData);`

返回值:

周期平均值

22. CalPreShoot: 预冲

函数声明: `DLL_API double WINAPI CalPreShoot(const short* pMaxData, const short*`

pMinData)

返回值:

预冲

23. CalOverShoot: 过冲

函数声明: DLL_API double WINAPI CalOverShoot(const short* pMaxData, const short* pMinData);

返回值:

过冲