

DTS6007 API 库说明

日 期	版 本	修 改 内 容	修 改 人
2023/7/6	V1.0	初始发布	谢正

概述

DTS6007 支持以下功能:

- IIC 串行数据输出接口;
- 输出原始直方图数据(仅单帧模式);
- 输出 7 种计算结果数据;

在 DTS6007lib 公共头文件中声明的 C 函数、结构体、枚举、类型定义和预处理器宏。文档将描述这些函数、结构体和类型。

DTS6007lib 提供了专用的移植接口, 简化了对 DTS6007lib 库的使用, 提高其不同平台的兼容性。

约束

由于 lib 配置接口都是基于 IIC 下发, 所以函数返回值都是同一标准, 使用的都是如下宏定义 (后面将不在赘述) :

```
#define PLS_SUCCESS 0
#define PLS_FAIL -1
#define PLS_TIMEOUT 1
```

定义在 pls_type.h 头文件下。

目录树

整个 lib 目录树如下:

```
├── dts6007_api.lib
└── include
    ├── dts6007_config.h
    ├── dts6007_def.h
    └── pls_type.h
```

在 dts6007_config.c 文件中包含所有芯片配置 API 接口。Include 目录下面主要是一些结构体, 枚举, 宏定义以及函数声明。

API 函数说明

1. int DTS6007Reg_Init

```
/**
 * @description:DTS6007Reg_Init 初始化函数
 * @param {dts6007_Callback_Fun} fun SDK 注册结构体
 * @return {*}
 * @note:
 */
int DTS6007Reg_Init(dts6007_Callback_Fun *fun);
```

初始化，注册函数接口，参数 fun 为结构体，dts6007_Callback_Fun 包含 4 个需要注册的结构体函数指针。芯片上电后调用此函数，注册成功后，才能正常使用相关的 api 接口。

2.DTS6007_I2c_ChangeAddr

```
/**
 * @description:DTS6007_I2c_ChangeAddr 修改 iic 地址
 * @param {unsigned char} new_addr 配置新的 IIC 器件地址
 * @return {*}
 * @note:
 */
int DTS6007_I2c_ChangeAddr(unsigned char new_addr); //修改 iic 地址
```

修改 iic 7bit 地址，修改完成后掉电会恢复默认地址 0x2c。

3.DTS6007_GetAllHistCal

```
/**
 * @description: 读取 hist 计算全部结果
 * @param {Cal_Res_Rawdata} *temp
 * @return {*}
 * @note:
 */
int DTS6007_GetAllHistCal(Cal_Res_Rawdata *temp);
```

读取直方图计算结果,14byte 数据。数据结构为:

- 0~1 (0 低位 1 高位, 下同): 阳光基底
- 2~3: 主峰强度

- 4~5: 主峰反射率校准
- 6~7: 主峰质心
- 8~9: 次峰强度
- 10~11: 次峰反射率校准
- 12~13: 次峰质心

4.DTS6007_GetSingleHistCal

```
/**
 * @description: 读取 hist 单个计算结果
 * @param enum hist_cal_type data 读取的类型
 * @param data 读取数据
 * @return {*}
 * @note:
 */
int DTS6007_GetSingleHistCal(enum hist_cal_type type,unsigned short *data);
```

读取单个直方图计算结果，SUN_AME MAIN_PEAK ERR_MAJ DIST_MAJ PEAK_MIN ERR_MIN DIST_MIN 分别为：阳光基底 主峰强度 主峰反射率校正 主峰质心 次峰强度 次峰反射率校正 次峰质心。

5.DTS56007_Spad_Focus^[2]

```
/**
 * @description: DTS56007_Spad_Focus
 * @param {uint16_t} num 开启单个 spad
 * @param {unsigned char} 开启对应的单个 spad
 * @param {unsigned short*} 单个 spad 的 peak 值
 * @return {*}
 * @note:
 */
int32_t DTS56007_Spad_Focus(unsigned char num,unsigned short *date);
```

采集单个 spad 的强度，共 8*16 阵列。从左到右依次计算，左上为第 0 个，右下为 127 个。

可依次扫描 0-127spad 实现对光功能。

6.DTS6007_TempConfig

```
/**
 * @description: DTS6007_TempConfig 温度初始化配置
 * @return {*}
 * @note:
```

```
*/  
int DTS6007_TempConfig(void);
```

温度初始化配置，配置完成后，可通过读取 ADC 通道 0 计算芯片当前温度。

7.DTS6007_TempWWDG_Set

```
/**  
 * @description: 设置温度窗口看门狗  
 * @param {unsigned short} adc_temp_hth 设置高温度阈值  
 * @param {unsigned short} adc_temp_lth 设置低温度阈值  
 * @return {*}  
 * @note:  
 */  
int DTS5048_TempWWDG_Set(unsigned short adc_temp_hth, unsigned short adc_temp_lth);
```

配置温度报警的上限或者下限阈值。

8.DTS6007_WaitInt_Flag

```
/**  
 * @description: 轮询标志位  
 * @param {enum interr_type} int_type 轮询中断标志位  
 * @param {time}waiting time(ms) 轮询超时时间，单位 ms  
 * @return {*}  
 * @note:  
 */  
int DTS6007_WaitInt_Flag(enum interr_type int_type,unsigned short time);
```

轮询读取中断标志，中断类型如下：

- INT_FRAME_DONE：一帧完成中断；
- INT_RESULT_AVL：计算结果完成中断；
- INT_TDC_RDY： tdc raw data 准备就绪中断指示；
- INT_ADC_DONE：采样一次完成中断指示；
- INT_FAULT_OT： OT 告警中断指示；
- INT_FAULT_OV： OT 告警中断指示；
- INT_CYCLES_ERROR：周期参数错误中断指示；
- INT_COP_MP_WT_SUM_OVF_ALM：溢出指示；
- INT_COP_MP_VA_SUM_OVF_ALM：溢出指示；
- INT_COP_MP_CNT_SUM_OVF_ALM：溢出的中断指示；
- INT_COP_SP_WT_SUM_OVF_ALM：溢出的中断指示；
- INT_COP_SP_VA_SUM_OVF_ALM：溢出的中断指示；
- INT_COP_SP_CNT_SUM_OVF_ALM：溢出的中断指示
- INT_COP_SA_CNT_SUM_OVF_ALM：溢出的中断指示
- INT_COP_SA_AVG_OVF_ALM：溢出的中断指示

9.DTS6007_GetADCDData

```
*/  
/**  
 * @description: 读取 ADC 数据  
 * @param  enum adc_channel ch 读取通道  
 * @param  enum adc_channel ch 读取数据  
 * @return {*}  
 * @note:  
 */  
int DTS6007_GetADCDData(enum adc_channel ch,unsigned short *temp);
```

获取 ADC 通道 ADC 原始值

10.DTS6007_OverFault_Cfg_En

```
/**  
 * @description: 开关过温度过压保护  
 * @param {enum state_type} Vool 使能过压保护  
 * @param {enum state_type} Temp 使能过温保护  
 * @return {*}  
 * @note:  
 */  
int DTS6007_OverFault_Cfg_En(enum state_type Vool, enum state_type Temp);
```

过温，过压保护使能，如果出现 fault_ov，fault_ot 告警时，就会关闭 boost，dac。

11.DTS6007_GetIntFlag

```
/**  
 * @description: (DTS6007_GetIntFlag)读取中断标志位  
 * @return {*}  
 * @note: 在使用外部中断管脚 int 时，读取该中断标志位会将改位清零，并且中断管脚会恢复为空闲状态  
 */  
int DTS6007_GetIntFlag(enum interr_type int_type, unsigned char *flag);
```

获取中断标志位状态，中断类型见 DTS6007_WaitInt_Flag 说明，flag 为 1 表示中断置位。读清

12.DTS6007_SetInt_ENABLE

```
/**  
 * @description: 开启中断  
 * @param {enum interr_type} int_type 使能 interr_type 枚举中的中断  
 * @return {*}  
 */
```

```

* @note:
*/
int DTS6007_SetInt_ENABLE(enum interr_type int_type);
    
```

使能中断，中断类型见 DTS6007_WaitInt_Flag 说明。

13.DTS6007_SetInt_DISABLE

```

/**
* @description: 关闭中断
* @param {enum interr_type} int_type 失能 interr_type 枚举中的中断
* @return {*}
* @note:
*/
int DTS6007_SetInt_DISABLE(enum interr_type int_type);
    
```

失能中断，中断类型见 DTS6007_WaitInt_Flag 说明。

14.DTS6007_SpadSetUnit

```

/**
* @description: 设置 spad 阵列开关
* @param {spad_cfg} *cfg 成员 spad_mode: 配置 SPAD 与或模式，spad_uint_deal 为开关状态
* @return {*}
* @note:
*/
int DTS6007_SpadSetUnit(spad_cfg *cfg);
    
```

设置 spad 开启阵列。spad_comb_mode 可以设置 spad 模式：

SPAD_COMB_MODE_AND 为 spad 与模式

SPAD_COMB_MODE_OR 为 spad 或模式

spad_uint_deal 为 SPAD 的 spad 开关状态。Eg:

```

spad_uint[]={
    _BIN_(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
    _BIN_(0,1,1,1,1,1,1,1,1,1,1,1,1,1,0),
    _BIN_(0,1,1,1,1,1,1,1,1,1,1,1,1,0),
    _BIN_(0,1,1,1,1,1,1,1,1,1,1,1,1,0),
    _BIN_(0,1,1,1,1,1,1,1,1,1,1,1,1,0),
    _BIN_(0,1,1,1,1,1,1,1,1,1,1,1,1,0),
    _BIN_(0,1,1,1,1,1,1,1,1,1,1,1,1,0),
    _BIN_(0,1,1,1,1,1,1,1,1,1,1,1,1,0),
    _BIN_(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
};
    
```

};对应关 spad 外圈。

15.DTS6007_Set_Fps

```
/**
 * @description: 设置 fps-激光重频
 * @param {unsigned int} fps      设置输出帧率
 * @param {unsigned int} laser_frq 设置激光重频
 * @return {*}
 * @note:
 */
int DTS6007_Set_Fps(unsigned int fps,unsigned int laser_frq);
```

设置输出帧率和激光重频。单帧模式下，需要考虑 iic 传输时间。

16.DTS6007_Set_TdcSubFix

```
/**
 * @description:
 * @param {unsigned char} tdc_sub_fix
 * @return {*}
 */
int DTS6007_Set_TdcSubFix(unsigned char tdc_sub_fix);
```

tdc 输出数据，换算为 bin 位置时，固定扣除部分

17.DTS6007_Loop_maa_cfg

```
/**
 * @description: 循环测量配置，根据设置的 FPS 循环计算测量数据
 * @return {*}
 * @note:
 */
int DTS6007_Loop_maa_cfg(void);
```

配置采集模式为自动循环采集

18.DTS6007_Single_maa_cfg

```
/**
 * @description: 配置单次测量配置，采集一次后自动停止测量
 * @return {*}
 * @note:
 */
int DTS6007_Single_maa_cfg(void);
```

配置采集模式为手动单次采集

19.DTS6007_Stop_maa

```
/**
 * @description: 停止测量和 DTS6007_Start_maa 相对应
 * @return {*}
 * @note:
 */
int DTS6007_Stop_maa(void);
```

停止采集

20.DTS6007_Start_maa

```
/**
 * @description: 开始测量和 DTS6007_Stop_maa 相对应
 * @return {*}
 * @note:
 */
int DTS6007_Start_maa(void);
```

开始采集

21.DTS6007_Hist_Out

```
/**
 * @description: 输出直方图还是，计算结果(单帧模式)
 * @param {enum frame_type} type 配置单帧采集输出模式 FRAME_HIST_OEN:直方图数
据 FRAME_COP_OEN_MAN: 输出计算结果
 * @return {*}
 * @note: 此配置只有在单帧模式下才能生效
 */
int DTS6007_Hist_Out(enum frame_type type);
```

配置单帧数据输出直方图，还是计算结果。FRAME_HIST_OEN:直方图数据
FRAME_COP_OEN_MAN: 输出计算结果

22.DTS6007_ReadHist_buff

```
/**
 * @description: 读取 hist 直方图结果
 * @param {unsigned short *} temp 读取的直方图数据
 * @return {*}
 */
```

```
* @note:直方图数据配置为 1024bin, 连续读取 1024 次就是直方图数据
*/
int DTS6007_ReadHist_buff(unsigned short *buff);
```

读取直方图数据循环读取 1024 次，即为一帧直方图数据。

NOTES:

[1] 此接口会修改芯片默认参数，建议仅用于生产测试。

API 应用 DEMO

1.循环采集模式

```
dts6007_Callback_Fun dts6007;
dts6007.IO_Reset=DTS6007_IO_Reset;    //硬件复位
dts6007.Delay_ms=DTS6007_Delay_ms;    //延时函数 ms
dts6007.i2c_Rev=i2c_master_Rev;       //iic 接收数据接口
dts6007.i2c_transmit=i2c_master_transmit;//iic 发送数据接口
DTS6007Reg_Init(&dts6007);//初始化配置
DTS6007_Set_Fps(30,5000000);
ret = DTS6007_Histout_Mode(1);/*配置 IIC 输出*/
DTS6007_SetInt_ENABLE(INT_FRAME_DONE);/*开启*/
DTS6007_SetInt_DISABLE(INT_RESULT_AVL);/*关闭*/
DTS6007_SetInt_DISABLE(INT_TDC_RDY);/*关闭*/
DTS6007_SetInt_DISABLE(INT_ADC_DONE);/*关闭*/
DTS6007_SetInt_DISABLE(INT_FAULT_OT);/*关闭*/
DTS6007_SetInt_DISABLE(INT_CYCLES_ERROR);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_MP_WT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_MP_VA_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_MP_CNT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SP_WT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SP_VA_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SP_CNT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SA_CNT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SA_AVG_OVF_ALM);/*关闭*/
DTS6007_Loop_maa_cfg(); //配置循环模式
DTS6007_GetIntFlag(INT_FRAME_DONE,&res);//清除 INT_FRAME_DONE
DTS6007_Start_maa();//开始测量
While(1);
```

```
/*外部 int 中断函数*/
Void exit_irqhandler()
{
    Unsigned char flag;
    Cal_Res_Rawdata temp;
    DTS6007_GetIntFlag(INT_FRAME_DONE,&flag);//读取 INT_FRAME_DONE 是否置位
    if(flag==1)
    {
        DTS6007_GetAllHistCal(&temp);//读取计算结果
    }
}
```

2.单帧模式

```
int ret;
/*配置 Idd 模式 0，激光脉宽 0.9ns 驱动输出电流调节 200ma*/
dts6007_Callback_Fun dts6007;
dts6007.IO_Reset=DTS6007_IO_Reset; //硬件复位
dts6007.Delay_ms=DTS6007_Delay_ms; //延时函数 ms
dts6007.i2c_Rev=i2c_master_Rev; //iic 接收数据接口
dts6007.i2c_transmit=i2c_master_transmit;//iic 发送数据接口
DTS6007Reg_Init(&dts6007);//初始化配置
DTS6007_Set_Fps(30,5000000);
DTS6007_Histout_Mode(1);/*配置 IIC 输出*/
DTS6007_SetInt_DISABLE(INT_FRAME_DONE);/*关闭*/
DTS6007_SetInt_DISABLE(INT_RESULT_AVL);/*关闭*/
DTS6007_SetInt_DISABLE(INT_TDC_RDY);/*关闭*/
DTS6007_SetInt_DISABLE(INT_ADC_DONE);/*关闭*/
DTS6007_SetInt_DISABLE(INT_FAULT_OT);/*关闭*/
DTS6007_SetInt_DISABLE(INT_CYCLES_ERROR);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_MP_WT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_MP_VA_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_MP_CNT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SP_WT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SP_VA_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SP_CNT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SA_CNT_SUM_OVF_ALM);/*关闭*/
DTS6007_SetInt_DISABLE(INT_COP_SA_AVG_OVF_ALM);/*关闭*/
DTS6007_Single_maa_cfg(); //配置循环模式
DTS6007_GetIntFlag(INT_FRAME_DONE,&res);//清除 INT_FRAME_DONE
While(1)
{
    Cal_Res_Rawdata temp;
```

```
DTS6007_Start_maa();//开始测量
ret = DTS6007_WaitInt_Flag(INT_FRAME_DONE,20);
if(ret==PLS_SUCCESS)
{
    DTS6007_Hist_Out(FRAME_HIST_OEN);
}
ret = DTS6007_WaitInt_Flag(INT_FRAME_DONE,20);
if(ret==PLS_SUCCESS)
{
    DTS6007_GetAllHistCal(&temp);//读取计算结果
}
}
```