

# Udisk\_PDF 说明

## 一、功能简介

本例程以 FLASH 为存储媒介模拟 U 盘设备,自动生成自定义的 PDF 文件。

特点: 1.提供 PDF 库,可基于此库自行制作 PDF 文件

2.以温湿度结果为模版,可更改模版页面的信息

3.具备模拟 U 盘功能,可用内置 FLASH 或外挂 FLASH 为存储媒介

4.具备文件管理系统

5.PDF 部分占用 RAM 约 10K

6.内置 FLASH 模拟 U 盘存储空间为 300K

7.支持 USB2.0 和 USB1.1

## 二、模版实现效果



1. 温湿度模版首页包括页眉页脚、设备信息、配置信息以及获取数据的详细信息等,如首页①部分。

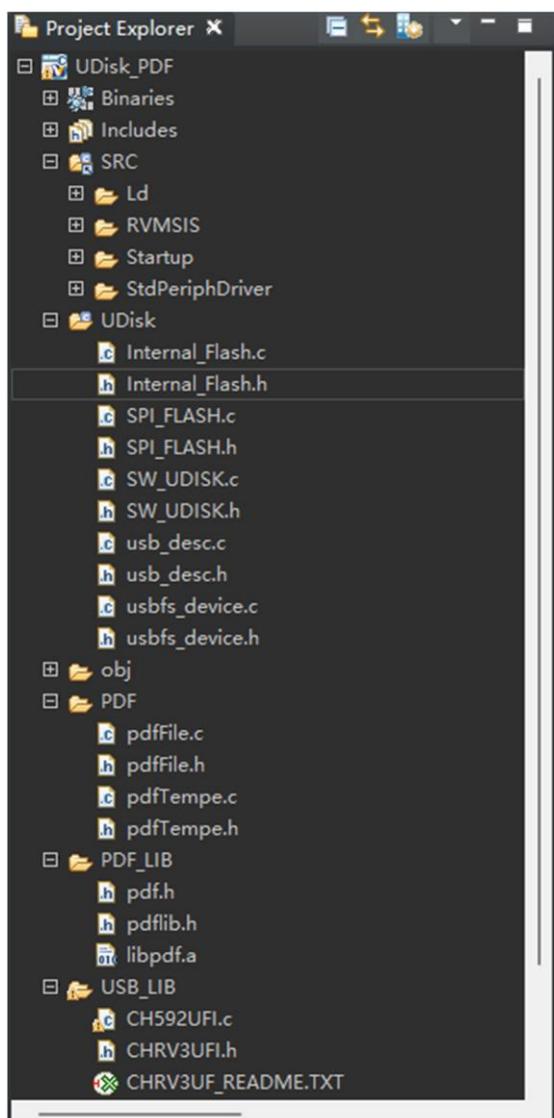
2. 首页还包括温湿度波形图: ②部分,此表格可在代码中选择显示或不显示。

3. ③为首页②部分图中详细温度所对应的时刻表和温度值。

4. ④为首页②部分图中详细湿度所对应的时刻表和湿度值。

5. ③和④的表格也可在代码中选择显示和不显示。

### 三、工程模块介绍



例程 Udisk\_PDF 主要由 SRC、UDisk、PDF、PDF\_LIB 和 User 文件夹组成，其中：

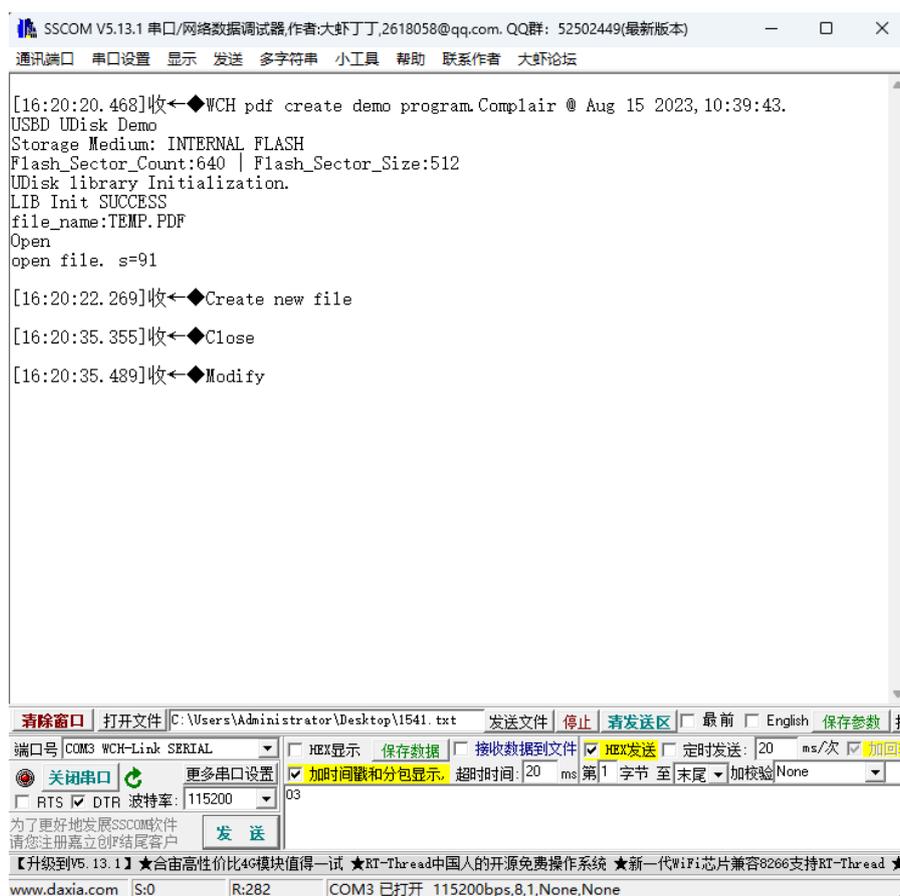
- SRC：驱动代码
- Udisk：模拟 U 盘相关代码
- PDF：PDF 生成相关代码
- PDF\_LIB：PDF 库
- User：主函数 Main.c

### 四、代码执行过程

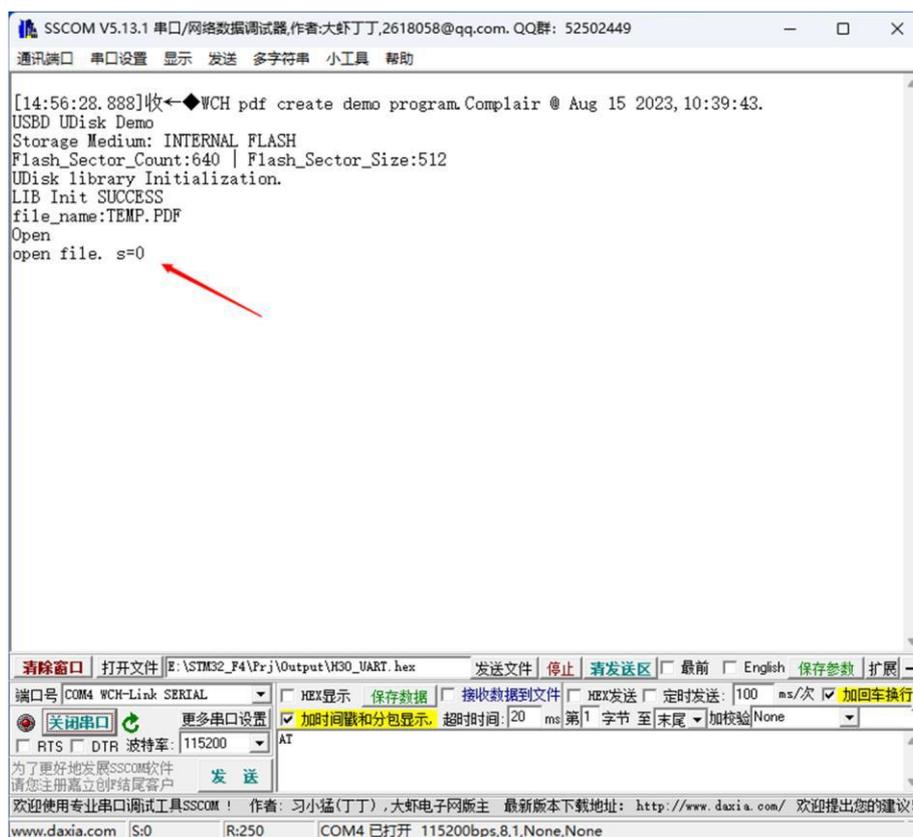
编译并烧录例程，由串口打印信息：

1. 打印例程相关信息，初始化库，输出时间为最后编译时间
2. 若 U 盘未格式化会触发错误码 0x91 “磁盘的主引导记录无效,可能磁盘尚未分区或者尚未格式化”，进而进入格式化 U 盘函数。

## 3. U 盘格式化成功后执行创建 PDF 文件、关闭文件和做最后的调整。



## 4. 若不清空 CodeFlash 重新烧录或只是重新上下电，打印结果如下：



## 五、代码生成结果

PDF 生成的内容基本来源于 pdfFile.c 文件中 pdf\_temperature\_init 函数。

### 1. 页眉信息和 WebUrl

```
strcpy(info->FileName, "template.pdf");  
strcpy(info->FileVersion, "v1.4");  
strcpy(info->WebUrl, "www.wch.cn");
```

①

### 2. 第一栏 Device Information

```
// Device Information  
info->DeviceInfo.pInfo = device_info;  
strcpy(info->DeviceInfo.ProductType, "ZZ-10A");  
strcpy(info->DeviceInfo.SerialNumber, "ABCD1234");  
strcpy(info->DeviceInfo.ProtectionGrade, "IP67");  
strcpy(info->DeviceInfo.ProbeType, "temperature&humidity");  
info->DeviceInfo.StorageSpace = 10000;  
info->DeviceInfo.MinMeauRange = -40;  
info->DeviceInfo.MaxMeauRange = 70;
```

②

### 3. 第二栏 Trip Information

```
// Trip Information  
info->TripInfo.pInfo = trip_info;  
strcpy(info->TripInfo.Order, "A-0-B-1-001");  
strcpy(info->TripInfo.Shipper, "anonymity");  
strcpy(info->TripInfo.City, "NanJing");  
strcpy(info->TripInfo.Carrier, "lobster");  
strcpy(info->TripInfo.Receiver, "abc");  
strcpy(info->TripInfo.Signature, "xxx");
```

③

### 4. 第三栏 Logger Configuration

```
// Logger Configuration  
info->ConfigInfo.pInfo = logger_conf;  
strcpy(info->ConfigInfo.StartMode, "Key Start");  
strcpy(info->ConfigInfo.StopMode, "Key Stop + Full Stop");  
info->ConfigInfo.IsKeyStop = LPDF_TRUE;  
info->ConfigInfo.IsTone = LPDF_FALSE;  
info->ConfigInfo.MaxTempAlarm = 20;  
info->ConfigInfo.MinTempAlarm = -10;  
  
info->ConfigInfo.StartDelayed = 60;  
info->ConfigInfo.Interval = 100;  
strcpy(info->ConfigInfo.TempUnit, centigrade);  
info->ConfigInfo.IsAlarmTone = LPDF_FALSE;  
info->ConfigInfo.MaxHumiAlarm = 70;  
info->ConfigInfo.MinHumiAlarm = 30;
```

④

## 5. 第四栏 Statistical Information

```

// Statistical Information
info->StatInfo.RecordNum = info->TempTotalDataNum + info->HumiTotalDataNum;
info->StatInfo.FirstRecordYear = timeStart[YEAR_OFFSET];
info->StatInfo.FirstRecordMonth = timeStart[MONTH_OFFSET];
info->StatInfo.FirstRecordDay = timeStart[DAY_OFFSET];
info->StatInfo.FirstRecordHour = timeStart[HOURL_OFFSET];
info->StatInfo.FirstRecordMinute = timeStart[MINUTE_OFFSET];
info->StatInfo.FirstRecordSec = timeStart[SECOND_OFFSET];

```

## 6. 图表绘制开关:

1. 波形图 2. PDF 后两页表格

```

info->isGraphEnable = LPDF_TRUE;
info->isTableEnable = LPDF_TRUE;

```

## 7. PDF 温湿度数据获取函数

```

LPDF_REAL 'pdf_get_temperature_data( LPDF_REAL *pBuf, LPDF_UINT offset, LPDF_UINT *num )
{
    for (int i = 0; i < *num; i++) {
        pBuf[i] = *((uint32_t*)((offset+i)*sizeof(LPDF_REAL)));
        pBuf[i] = (((uint64_t)pBuf[i]*(45-1))>>32)+1;
    }
    return pBuf;
}

.....
* @fn      pdf_get_humidity_data
* @brief   The purpose of this function is to obtain a segment of humidity data from the specified memory address offset, process it, and store it in the given buffer.
*
*          pBuf: Pointer to the buffer where temperature data is stored;
*          offset: The starting offset address of temperature data in memory;
*          num: Pointer to an integer pointer used to store the number of temperature data obtained.
*
* @param   pBuf - 数据缓冲区
*          offset - 偏移量
*          num - 数据量
*
* @return  a pointer to the buffer where temperature data is stored, i.e. pBuf
*/
LPDF_REAL 'pdf_get_humidity_data( LPDF_REAL *pBuf, LPDF_UINT offset, LPDF_UINT *num )
{
    for (int i = 0; i < *num; i++) {
        pBuf[i] = *((uint32_t*)((offset+i)*sizeof(LPDF_REAL)));
        pBuf[i] = (((uint64_t)pBuf[i]*(70-30))>>32)+30;
    }
    return pBuf;
}

```