User Manual

ATK-DP100DLL 使用说明文档

DP100 数控电源上位机二次开发库

使用文档

正点原子

广州市星翼电子科技有限公司

修订历史

版本	日期	原因
V1.00	2022/06/29	第一次发布

目 录

1 说明

ATK-DP100DLL.dll 是配套正点原子产品 DP100 数控电源使用的，用户可以利用动态库来自定义开发 DP100 数控电源配套上位机.本文档使用使用示例为 C#工程的示例。本文档中以下出现动态库或者本库都是指 ATK-DP100DLL.dll

2 工程简介

2.1 开发环境

系统：windows10
环境：.net framework
语言：C#
开发工具：Microsoft Visual Studio 2019

2.2 结构如下图

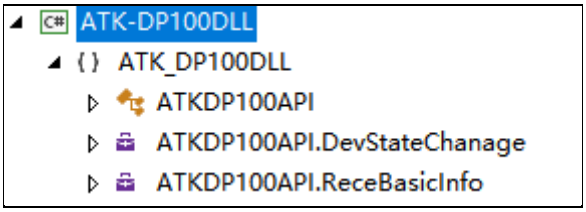

图 2.2.1

2.3 函数简介

名称	类型	修饰符	摘要
方法			
ATKDP100API		public	
CloseOut	bool	public	关闭预设输出组
DevOpenOrClose	bool	public	连接设备/断开设备
DevStateChange	void	private	
GetBasicInfo	bool	public	发送获取 基础信息
GetCurrentBasic	bool	public	获取设备当前的基础信息
GetCurrentBasic	bool	private	
GetDevInfo	bool	public	获取设备信息
GetGroupInfo	bool	public	获取
GetSysPar	bool	public	设置电流
OpenOut	bool	public	打开输出
ReceBasicInfoEventHandler	void	private	
SetBasicInfo	bool	public	设置对应分组的 输出电压 输出电流 过流保护参数，过压保护参数
SetGroupInfo	bool	public	写入
SetSysPar	bool	public	设置系统参数
UseGroup	bool	public	调出预设置的组参数信息

图 2.3.1

3 使用说明

3.1 实例化类

```
ATK_DP100DLL.ATKDP100API api = new ATK_DP100DLL.ATKDP100API();
```

3.2 设备连接状态

```
/// <summary>
/// 设备的连接状态 true 是连接
/// </summary>
public bool ConnState
```

3.3 设备连接或者断开

```
/// <summary>
/// 连接设备/断开设备 设备在连接状态下调用就是断开，设备在断开状态下调用就是连接
/// </summary>
/// <returns>设备连接 返回 true      设备断开了 返回 false</returns>
public bool DevOpenOrClose()
```

3.4 获取设备信息

```
/// <summary>
/// 获取设备信息
/// </summary>
/// <param name="devType">设备型号</param>
/// <param name="hdwVer">硬件版本</param>
/// <param name="appVer">设备 app 版本</param>
/// <param name="devSN">设备 SN</param>
/// <param name="devState">设备状态 APP/Boot</param>
/// <returns></returns>
public bool GetDevInfo(ref string devType, ref string hdwVer, ref string appVer, ref string devSN, ref string devState)
```

3.5 获取设备状态信息

```
/// <summary>
/// 获取设备当前的基础信息
/// </summary>
/// <param name="index">当前使用的预设输出组 编号 设备默认为0 取值0-9</param>
/// <param name="state">输出状态 0 关闭  >0 打开</param>
/// <param name="vset">电压</param>
/// <param name="iset">电流</param>
/// <param name="ovp">过压保护</param>
/// <param name="ocp">过流保护</param>
/// <returns>正常返回 true 异常 false</returns>
public bool GetCurrentBasic(ref byte index, ref byte state, ref UInt16 vset, ref UInt16 iset, ref UInt16 ovp, ref UInt16 ocp)
```

3.6 获取预设输出组信息

```
/// <summary>
/// 获取预设输出组的信息
/// </summary>
/// <param name="index">组号取值 0 -9  0号编组只能读取，无法设置 </param>
/// <param name="iset">输出电压设置 mV</param>
/// <param name="vset">输出电流设置 mA</param>
/// <param name="ocp">过流保护参数设置 mA</param>
/// <param name="ovp">过压保护参数设置 mA</param>
/// <returns>正常返回 true 异常 false</returns>
public bool GetGroupInfo(byte index, ref UInt16 iset, ref UInt16 vset, ref UInt16 ocp, ref UInt16 ovp)
```

3.7 设置预设输出组信息写入

```
/// <summary>
/// 将信息设置到预设输出组 此函数只会让数据写入到设备预设输出组
/// </summary>
/// <param name="index">预设输出组编号 1-9 </param>
/// <param name="iset">输出电压设置 mV</param>
/// <param name="vset">输出电流设置 mA</param>
/// <param name="ocp">过流保护参数设置 mA</param>
/// <param name="ovp">过压保护参数设置 mA</param>
/// <returns>正常返回 true 异常 false</returns>
public bool SetGroupInfo(byte index, ushort iset, ushort vset, ushort ocp, ushort ovp)
```

3.8 调出设备里面预设输出组

```
/// <summary>
/// 调出预设输出组 让设备切换到这个分组使用
/// </summary>
/// <param name="index">预设输出组编号 1-9</param>
/// <param name="iset">输出电压设置 mV</param>
/// <param name="vset">输出电流设置 mA</param>
/// <param name="ocp">过流保护参数设置 mA</param>
/// <param name="ovp">过压保护参数设置 mA</param>
/// <returns>正常返回 true 异常 false</returns>
public bool UseGroup(byte index, UInt16 iset, UInt16 vset, UInt16 ocp, UInt16 ovp)
```

3.9 打开输出

```
/// <summary>
/// 打开输出在打开状态下设置输出电压设置输出电流
/// </summary>
/// <param name="index">预设输出组编号 设备打开默认 0 范围 0-9  </param>
/// <param name="iset">输出电压设置 mV</param>
/// <param name="vset">输出电流设置 mA</param>
/// <returns>正常返回 true 异常 false</returns>
public bool OpenOut(byte index, ushort iset, ushort vset)
```

3.10 关闭输出

```
/// <summary>
/// 关闭输出 在关闭状态下设置输出电压 设置输出电流
/// </summary>
/// <param name="index">当前打开的 预设输出组编号 设备打开默认 0 范围 0-9  </param>
/// <param name="iset">输出电压设置 mV</param>
/// <param name="vset">输出电流设置 mA</param>
/// <returns>正常返回 true 异常 false</returns>
public bool CloseOut(byte index, ushort iset, ushort vset)
```

3.11 获取系统参数

```
/// <summary>
/// 获取系统参数
/// </summary>
/// <param name="blklev">背光等级     取值范围 0-4</param>
/// <param name="vollev">音量等级     取值范围 0-4</param>
```

```
/// <param name="opp">过功率保护参数 1050W  单位 0.1W </param>
/// <param name="otp">过热保护参数   取值范围 50-80 摄氏度 单位 0.1 摄氏度</param>
/// <returns>正常返回 true 异常 false</returns>
public bool GetSysPar(ref byte blklev, ref byte vollev, ref ushort opp, ref ushort otp)
```

3.12 设置系统参数

```
/// <summary>
/// 设置系统参数
/// </summary>
/// <param name="blklev">背光等级    取值范围 0-4</param>
/// <param name="vollev">音量等级    取值范围 0-4</param>
/// <param name="opp">过功率保护参数 1050W  单位 0.1W </param>
/// <param name="otp">过热保护参数   取值范围 50-80 摄氏度 单位 0.1 摄氏度</param>
/// <returns>正常返回 true 异常 false</returns>
public bool SetSysPar(byte blklev, byte vollev, ushort opp, UInt16 otp)
```

3.13 读取电压 电流

```
/// <summary>
/// 电压电流回调 通过这个回调可以直接湖区电压电流
/// </summary>
/// <param name="v">电压 mV</param>
/// <param name="i">电流 mA</param>
public delegate void ReceBasicInfo(UInt16 v, UInt16 i);
public ReceBasicInfo ReceBasicInfoEvent;

/// <summary>
/// 发送获取电压电流的指令
/// </summary>
/// <returns>正常返回 true 异常 false</returns>
public bool GetBasicInfo()
```

代码示例：

```
 {
     api.ReceBasicInfoEvent += ReceBasicInfo; //设置回调
     api.GetBasicInfo()                       //获取电压电流

 }
 //电压 电流回调示例
 void ReceBasicInfo(UInt16 V,UInt16 I) {
     //个人根据自己软件需求做处理
     Debug.WriteLine("电压 " + V+"mV ");
```

```
    Debug.WriteLine("电流 " + I+"mA ");
}
```

4 源码示例

```
static void Main(string[] args)
        {
            byte
               index = 0,       //组编号         取值范围 0-9  编号 0 为只读
               state = 0;       //状态
            UInt16
                iset = 1000,    //电流输出       单位 mA
                vset = 1000,    //电压输出       单位 mV
                ocp = 10000,    //过流保护参数   单位 mA
                ovp = 10000;    //过压保护参数   单位 mV

            ATKDP100API api = new ATKDP100API();            //s

            Console.WriteLine("设备连接状态:" + api.ConnState);
            Console.WriteLine("---------------------------------");

            Console.WriteLine("设备连接:" + api.DevOpenOrClose());
            Console.WriteLine("---------------------------------");

            #region  //获取设备硬件信息
            string devtype = string.Empty, hdwVer = string.Empty, appver = string.Empty,
devsn = string.Empty, devState = string.Empty;
            Console.WriteLine("获取设备信息：" + api.GetDevInfo(ref devtype, ref hdwVer,
ref appver, ref devsn, ref devState));
            Console.WriteLine("设备名字：" + devtype);
            Console.WriteLine("硬件版本:" + hdwVer);
            Console.WriteLine("app 版本:" + appver);
            Console.WriteLine("设备 SN:" + devsn);
            Console.WriteLine("设备状态:" + devState);
            Console.WriteLine("---------------------------------");
            #endregion

            #region  //获取设备状态
            Console.WriteLine("设备当前状态：" + api.GetCurrentBasic(ref index, ref state,
ref vset, ref iset, ref ovp, ref ocp));
            Console.WriteLine("当前使用的分组编号：" + index);
            Console.WriteLine("当前状态："+(state > 0?"开启":"关闭") );
```

```
            Console.WriteLine("输出电压：" + vset);
            Console.WriteLine("输出电流：" + iset);
            Console.WriteLine("过压：" + ovp);
            Console.WriteLine("过流：" + ocp);

            Console.WriteLine("---------------------------------");
            #endregion

            #region //获取预设输出组信息  获取预设输出组 编号为 index 的信息
            index = 2;
            //获取第二组  预设输出组信息
            Console.WriteLine("获取分组数据：" + api.GetGroupInfo(index, ref iset, ref vset,
ref ocp, ref ovp));
            Console.WriteLine("电流输出 mA " + iset);
            Console.WriteLine("电压输出 mV " + vset);
            Console.WriteLine("过流保护 mA " + ocp);
            Console.WriteLine("过压保护 mV " + ovp);
            Console.WriteLine("---------------------------------");
            #endregion

            #region //设置第二组的  预设输出组信息
            index = 2;
            iset = 1000;          //输出电压设置   1000mA
            vset = 1000;          //输出电流设置   1000mV
            ocp = 2000;           //过流保护设置   2000mA
            ovp = 5000;           //过压保护设置   5000mV
            Console.WriteLine("设置预设组：" + api.SetGroupInfo(index, iset, vset, ocp,
ovp));
            Console.WriteLine("调出预设输出组：" + api.UseGroup(index, iset, vset, ocp,
ovp));
            Console.WriteLine("---------------------------------");
            #endregion

            #region //打开或关闭输出  设置电压 电流
            //在打开输出 或者在打开模式下设置电压电流
            index = 2;            //组号 2
            iset = 500;          //输出电压设置  500mA
            vset = 500;           //输出电流设置  500mV
            //这里可以打开输出  和改变电压电流输出
            Console.WriteLine("打开输出电压 电流" + api.OpenOut(index, iset, vset));

            //这里是关闭输出
            Console.WriteLine("关闭输出电压 电流" + api.CloseOut(index, iset, vset));
```

```
        Console.WriteLine("--------------------------------");
        #endregion

        #region 系统参数  过热 过功率 背光等级 音量登记
        byte
            blk = 0,                //背光等级 取值 0-4
            vol = 0;                //音量等级 取值 0-4
        UInt16
            opp = 0,                //单位 0.1W
            otp = 0;                //单位摄氏度 取值范围 50-80
        //获取系统参数
        api.GetSysPar(ref blk, ref vol, ref opp, ref otp);
        Console.WriteLine("过热：" + otp+"℃");
        Console.WriteLine("过功率：  " + opp/10+"W");
        Console.WriteLine("背光：" + blk);
        Console.WriteLine("音量：" + vol);
        Console.WriteLine("--------------------------------");

        //设置参数
        api.SetSysPar(blk, vol, opp, otp);
        #endregion

        #region 读取电压电流
        api.ReceBasicInfoEvent += ReceBasicInfo;
        //帧率可以控制在 1-100hz，根据需求改
        while (true)
        {
            api.GetBasicInfo();                 //获取电压电流
            Thread.Sleep(100);
        }

        #endregion

        Console.ReadLine();
    }
    //此出接收获取的电压 电流
    static void ReceBasicInfo(UInt16 V, UInt16 I) {
        Console.WriteLine("电压"+V+"mV");
        Console.WriteLine("电流"+I+"mA");
    }
```

5 其他

1、 购买地址：

天猫：https://zhengdianyuanzi.tmall.com

淘宝：https://openedv.taobao.com

2、 资料下载

模块资料下载地址：http://www.openedv.com/docs/index.html（更新模块地址，找运营要）

3、 技术支持

公司网址：www.alientek.com

技术论坛：http://www.openedv.com/forum.php

在线教学：www.yuanzige.com

B 站视频：https://space.bilibili.com/394620890

传真：020-36773971

电话：020-38271790

正点原子