

ATK-DP100DLL User Manual

DP100 Secondary development dynamic library

DP100 Digital Power Supply

ALIENTEK



History

Version	Date	Log
V1.0	2022/06/29	first release
V2.0	2023/08/21	Update the function of getting system parameters and setting system parameters Reverse connection protection, automatic output after booting requires the app version of the device ≥ 1.3

Directory

1 Description	1
2 Project description	1
2.1 Development environment	1
2.2 Project structure	1
2.3 API Reference	2
3 API usage reference	2
3.1 Instantiate	2
3.2 Device connection status	2
3.3 Operating device connection or disconnection	2
3.4 Get device information	3
3.5 Get device status information	3
3.6 Get preset output group information	3
3.7 Set preset output group information write	4
3.8 Call up the preset output group in the device	4
3.9 open output	4
3.10 close output	5
3.11 Get system parameters	5
3.12 Set system parameters	5
3.13 Read voltage Current	6
4 Demo	6
5 Serivices	10

1 Description

ATK-DP100DLL.dll is used to support Alientek's self-developed product DP100 CNC power supply. Users can use the dynamic library to customize and develop the upper computer for DP100 CNC power supply. The usage examples in this document are examples of C# projects. The following dynamic libraries or this library in this document refer to ATK-DP100DLL.dll

2 Project description

2.1 Development environment

System: windows10

Frame: .net framework 4.0

Programming language: C#

development tools: Microsoft Visual Studio 2019

2.2 Project structure

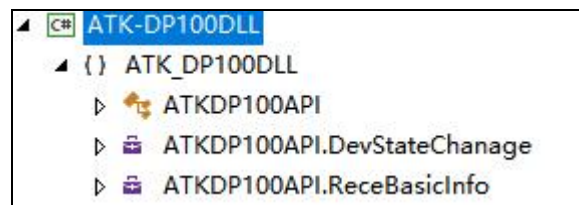


图 2.2.1

2.3 API Reference

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	调出预设设置的组参数信息

图 2.3.1

3 API usage reference

3.1 Instantiate

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

3.2 Device connection status

```
/// <summary>
/// Is the device connected
/// true means connection
/// false means disconnect
/// </summary>
public bool ConnState
```

3.3 Operating device connection or disconnection

```
/// <summary>
/// When the device is used in the connected state, the device will be disconnected
/// The device will be connected to the device when it is used in the disconnected state
/// </summary>
/// <returns> true means connection false means disconnect </returns>
public bool DevOpenOrClose()
```

3.4 Get device information

```
/// <summary>
/// Get device information
/// </summary>
/// <param name="devType">Device model </param>
/// <param name="hdwVer">hardware version </param>
/// <param name="appVer">Device app version </param>
/// <param name="devSN">device serial number </param>
/// <param name="devState">device status "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 Get device status information

```
/// <summary>
/// Obtain the current basic information of the device
/// </summary>
/// The currently used preset output group number
/// The device defaults to 0, the value range is 0-9</param>
/// <param name="state">Output state 0 off >0 on </param>
/// <param name="vset">Voltage </param>
/// <param name="iset">current </param>
/// <param name="ovp">overvoltage protection </param>
/// <param name="ocp">overcurrent protection </param>
/// <returns>normal return true / abnormal return 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 Get preset output group information

```
/// <summary>
/// Get preset output group information
/// </summary>
/// <param name="index">Group number value 0-9 Group
/// 0 can only be read, but cannot be set </param>
/// <param name="iset">Output Current Setting mA </param>
/// <param name="vset">Output voltage setting mV </param>
/// <param name="ocp">Overcurrent protection parameter setting mA </param>
/// <param name="ovp">Overvoltage protection parameter setting mV </param>
/// <returns> normal return true / abnormal return false </returns>
```

```
public bool GetGroupInfo(byte index, ref UInt16 iset, ref UInt16 vset, ref UInt16 ocp, ref
UInt16 ovp)
```

3.7 Set preset output group information write

```
/// <summary>
/// Set information to preset output group This function will only write data to the device
/// preset output group
/// </summary>
/// <param name="index"> Group number value 0-9 </param>
/// <param name="iset"> Output Current Setting mA </param>
/// <param name="vset"> Output voltage setting mV </param>
/// <param name="ocp"> Overcurrent protection parameter setting mA </param>
/// <param name="ovp"> Overvoltage protection parameter setting mV </param>
/// <returns> normal return true / abnormal return false </returns>
public bool SetGroupInfo(byte index, ushort iset, ushort vset, ushort ocp, ushort ovp)
```

3.8 Call up the preset output group in the device

```
/// <summary>
/// Call up the preset output group and let the device switch to this group
/// </summary>
/// <param name="index"> Group number value 0-9 </param>
/// <param name="iset"> Output Current Setting mA </param>
/// <param name="vset"> Output voltage setting mV </param>
/// <param name="ocp"> Overcurrent protection parameter setting mA </param>
/// <param name="ovp"> Overvoltage protection parameter setting mV </param>
/// <returns> normal return true / abnormal return false </returns>
public bool UseGroup(byte index, UInt16 iset, UInt16 vset, UInt16 ocp, UInt16 ovp)
```

3.9 open output

```
/// <summary>
/// Turn on the output Set the output voltage in the open state Set the output current
/// </summary>
/// <param name="index"> Preset output group number device open default 0 range 0-9 </param>
/// <param name="iset"> Output Current Setting mA </param>
/// <param name="vset"> Output voltage setting mV </param>
/// <returns> normal return true / abnormal return false </returns>
public bool OpenOut(byte index, ushort iset, ushort vset)
```

3.10 close output

```
/// <summary>
/// Turn off the output Set the output voltage in the off state Set the output current
/// </summary>
/// <param name="index">Preset output group number device open default 0 range 0-9 </param>
/// <param name="iset">Output Current Setting mA </param>
/// <param name="vset">Output voltage setting mV </param>
/// <returns> normal return true / abnormal return false </returns>
public bool CloseOut(byte index, ushort iset, ushort vset)
```

3.11 Get system parameters

```
/// <summary>
/// Get system parameters
/// </summary>
/// <param name="blklev">Backlight level value range 0-4</param>
/// <param name="vollev">Volume level value range 0-4</param>
/// <param name="opp">over power protection parameter 1050W unit 0.1W</param>
/// <param name="otp">Overheat protection parameter value range 50-80 degrees Celsius, unit
///0.1 degrees Celsius</param>
/// <param name="en_rep">Enable reverse polarity protection 0: off 1: on </param>
/// <param name="en_auto_out">Enable power-on automatic output 0: off 1: on </param>
/// <returns>Normal returns true, abnormal false</returns>
public bool GetSysPar(ref byte blklev, ref byte vollev, ref ushort opp, ref ushort otp, ref
byte en_rep, ref byte en_auto_out)
```

3.12 Set system parameters

```
/// <summary>
/// Set system parameters
/// </summary>
/// <param name="blklev">Backlight level value range 0-4</param>
/// <param name="vollev">Volume level value range 0-4</param>
/// <param name="opp">over power protection parameter 1050W unit 0.1W</param>
/// <param name="otp">Overheat protection parameter value range 50-80 degrees Celsius, unit
///0.1 degrees Celsius</param>
/// <param name="en_rep">Enable reverse polarity protection 0: off 1: on </param>
/// <param name="en_auto_out">Enable power-on automatic output 0: off 1: on </param>
/// <returns>Normal returns true, abnormal false</returns>
public bool SetSysPar(string appver, byte blklev, byte vollev, ushort opp, UInt16 otp, byte
en_rep, byte en_auto_out)
```

3.13 Read voltage Current

```

/// <summary>
/// Voltage and current callback Through this callback, the voltage and current can be
///obtained directly
/// </summary>
/// <param name="v">Voltage mV</param>
/// <param name="i">Current mA</param>
public delegate void ReceBasicInfo(UInt16 v, UInt16 i);
public ReceBasicInfo ReceBasicInfoEvent;

/// <summary>
/// Send command to get voltage and current
/// </summary>
/// <returns>Normal returns true, abnormal false</returns>
public bool GetBasicInfo()

```

Example:

```

{
    api.ReceBasicInfoEvent += ReceBasicInfo; //Callback
    api.GetBasicInfo()                      //Get voltage current
}
//Callback
void ReceBasicInfo(UInt16 V, UInt16 I) {
    // Voltage and current processing logic
    Debug.WriteLine("voltage " + V+"mV ");
    Debug.WriteLine("current " + I+"mA ");
}

```

4 Demo

```

static void Main(string[] args)
{
    byte
        index = 0,    // Group number Value range 0-9 Number 0 is read-only
        state = 0;    //状态
    UInt16
        iset = 1000,  // current output unit mA
        vset = 1000,  // voltage output unit mV
        ocp = 10000,  // overcurrent protection parameter unit mA

```

```

        ovp = 10000;    // overvoltage protection parameter unit mV

    ATKDP100API api = new ATKDP100API();           //s

    Console.WriteLine("Device connection status:" + api.ConnState);
    Console.WriteLine("-----");

    Console.WriteLine("Device connection:" + api.DevOpenOrClose());
    Console.WriteLine("-----");

    #region //Get device hardware information
    string devtype = string.Empty, hdwVer = string.Empty, appver = string.Empty,
    devsn = string.Empty, devState = string.Empty;
    Console.WriteLine("Get device information: " + api.GetDevInfo(ref devtype, ref
    hdwVer, ref appver, ref devsn, ref devState));
    Console.WriteLine("Device name: " + devtype);
    Console.WriteLine("Hardware version:" + hdwVer);
    Console.WriteLine("app version:" + appver);
    Console.WriteLine("Device SN:" + devsn);
    Console.WriteLine("Device status:" + devState);
    Console.WriteLine("-----");
    #endregion

    #region //Get device status
    Console.WriteLine("Current state of the device: " + api.GetCurrentBasic(ref
    index, ref state, ref vset, ref iset, ref ovp, ref ocp));
    Console.WriteLine("Currently used group number: " + index);
    Console.WriteLine("Current state: " + (state > 0 ? "Open" : "Close" ));
    Console.WriteLine("Output voltage: " + vset);
    Console.WriteLine("Output current: " + iset);
    Console.WriteLine("Overvoltage: " + ovp);
    Console.WriteLine("Overcurrent: " + ocp);

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

    #region //Get the information of the preset output group Get the information
    of the preset output group whose number is index
    index = 2;
    //Get the second group of preset output group information
    Console.WriteLine("Get group data: " + api.GetGroupInfo(index, ref iset, ref
    vset, ref ocp, ref ovp));
    Console.WriteLine("current output mA" + iset);
    Console.WriteLine("voltage output mV " + vset);

```

```

Console.WriteLine("overcurrent protection mA " + ocp);
Console.WriteLine("overvoltage protection mV " + ovp);
Console.WriteLine("-----");
#endregion

#region // Set the preset output group information of the second group
index = 2;
iset = 1000;           // output voltage setting 1000mA
vset = 1000;           // Set the output current to 1000mA
ocp = 2000;            // overcurrent protection setting 2000mA
ovp = 5000;            // overvoltage protection setting 5000mV
Console.WriteLine("Set preset group: " + api.SetGroupInfo(index, iset, vset,
ocp, ovp));
Console.WriteLine("Call out the preset output group: " + api.UseGroup(index,
iset, vset, ocp, ovp));
Console.WriteLine("-----");
#endregion

#region // Turn on or off output, set voltage current
// Set the voltage and current in open output or in open mode
index = 2;             // group number 2
iset = 500;            // output voltage setting 500mA
vset = 500;            // Set the output current to 500mA
// Here you can open the output and change the voltage and current output
Console.WriteLine("Open output voltage current" + api.OpenOut(index, iset,
vset));

// here is to close the output
Console.WriteLine("Close output voltage current" + api.CloseOut(index, iset,
vset));

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

#region System parameters Overheat Overpower Backlight level Volume level
byte
    blk = 0,           // backlight level value 0-4
    vol = 0,           // Volume level ranges from 0-4
    en_rep = 0,         // Enable reverse polarity protection 0: off 1: on
    en_auto_out = 0;    // Enable automatic output at power on 0: off 1:
on
    UInt16
    opp = 0,           // unit 0.1W
    otp = 0;           // Unit Celsius Value range 50-80

```

```

// Get system parameters
api.GetSysPar(ref blk, ref vol, ref opp, ref otp, ref en_rep, ref en_auto_out);
Console.WriteLine("Overheating: " + otp+"°C");
Console.WriteLine("over power: " + opp/10+"W");
Console.WriteLine("Backlight: " + blk);
Console.WriteLine("Volume: " + vol);
Console.WriteLine("Reverse connection protection switch: " + (en_rep==0 ?
"OFF":"ON"));
Console.WriteLine("Power-on automatic output: " + (en_auto_out == 0 ? "OFF" :
"ON"));
Console.WriteLine("-----");

// Setting parameters
api.SetSysPar(appver, blk, vol, opp, otp, en_rep, en_auto_out);
#endregion

#region read voltage and current
api.ReceBasicInfoEvent += ReceBasicInfo;

while (true)
{
    api.GetBasicInfo(); // Get voltage and current
    Thread.Sleep(100); // time delay
}

#endregion

Console.ReadLine();
}

// This output receives the obtained voltage and current
static void ReceBasicInfo(UInt16 V, UInt16 I) {
    Console.WriteLine("voltage "+V+"mV");
    Console.WriteLine("current "+I+"mA");
}

```

5 Services

1. After – sales Service:

DP100 host has a one-year free warranty service in the case of non-artificial damage. Please contact the dealer for warranty service.

2. Website

Download : www.alientek.com/download
Company : www.alientek.com
Aliexpress : www.aliexpress.com/store/1102909571

3. Contact US

E-mail : fae-smt@alientek.com

The logo consists of the word "ALIENTEK" in white, bold, uppercase letters, centered within a blue rounded rectangle.