

Process Device

- Process Device
 - `Open(char * arg);`
 - `Close()=0;`
 - `WriteString(const char * cmd);`
 - `ReadString();`
 - `ClearBuffer();`
 - `Write(void * buffer,unsigned long len);`
 - `Read(void * buffer,int len);`
 - `SetDetectString(const char * regex,int flag = 0);`
 - `WaitForString(int iTimeOut=30*1000);`

Version History

V1.1

Author:--

Date:---

Change List:

* TBD

V1.0

Author:Ryan

Date:2017-01-20

Change List:

* 初稿

Open(char * arg);

打开一个设备

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

Close()=0;

关闭一个设备

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

WriteString(const char * cmd);

写一个字符串到设备

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

ReadString();

读设备目前的所有的输出缓冲

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

ClearBuffer();

清空当前设备的缓冲

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

Write(void * buffer,unsigned long len);

写一段数据到设备

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

Read(void * buffer,int len);

读设备输出缓冲

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

SetDetectString(const char * regex,int flag = 0);

设置需要检测设备输出缓冲的字符

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code

WaitForString(int iTimeOut=30*1000);

等待已经设备输出缓冲里出现被设置的监测字符

```
virtual int Open(char * arg)
```

参数:

arg:

返回值:

0: 打开成功

其它: error code