

上海仰邦科技股份有限公司

Led5kSDKC#Demo 函数说明

SDK 编程指南

2015-5-19

Copyright

All rights reserved. No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by ONBON.

© 2010-2015Onbon

版本历史:

版本号	日期	作者	描述
15.07.28	2015-07-28	Zhang	完成了通过广播模式、网络固定 IP 模式、网络 Modbus 模式、485 总线模式、串行 Modbus 模式发送图文节目、动态区数据，可以查询固件、设置开关机、查询控制器状态、亮度调整等等

目录

1. 错误码及说明	4
1.1 错误状态	4
2. 显示屏参数初始化说明	5
3. API 调用顺序.....	6
3.1 总体次序	6
4. 函数说明	7
4.1 连接控制卡	7
4.1.1 InitSdk 函数.....	7
4.2 CreateBroadCast 函数.....	7
4.2.1 CreateClient 函数.....	7
4.2.2 CreateTcpModbus 函数	8
4.2.3 CreateComClient 函数	8
4.2.4 CreateComModbus 函数	8
4.2.5 SetTimeout 函数	9
4.3 图文区节目	9
4.3.1 OFS_SendFileData 函数	9
4.3.2 OFS_DeleteFile 函数	10
4.3.3 SCREEN_LockProgram 函数	10
4.3.4 OFS_Formatting 函数.....	10
4.4 动态区节目	11
4.4.1 SCREEN_SendDynamicArea 函数.....	11
4.4.2 SCREEN_DelDynamicArea 函数	11
4.5 固件操作	11
4.5.1 CON_CheckCurrentFirmware 函数	11
4.5.2 CON_FirmwareActivate 函数.....	12
4.6 开关机	12
4.6.1 SCREEN_ForceOnOff 函数	12
4.6.2 SCREEN_TimeTurnOnOff 函数.....	12
4.6.3 SCREEN_CancelTimeOnOff 函数.....	13
4.7 其它命令	13
4.7.1 CON_PING 函数	13
4.7.2 CON_Reset 函数.....	13
4.7.3 CON_SytemClockCorrect 函数.....	14
4.7.4 SCREEN_SetBrightness 函数	14
4.8 查询控制卡状态	15
4.8.1 CON_ControllerStatus 函数	15
附录.....	16
控制器类型	16
串口数据位	16
串口校验模式	16
串口停止位	16
结构体定义数据	16
控制器状态查询	17
区域数据格式设置	18

Led5kSDKC#Demo 函数说明

1. 错误码及说明

动态库中每个接口函数最后都返回函数执行结果，用户可根据该执行结果查找判断该函数的执行状况。

1.1 错误状态

错误名称	代码	说明
ERR_NO	0	No Error
ERR_OUTOFGROUP	1	Command Group Error
ERR_NOCMD	2	Command Not Found
ERR_BUSY	3	The Controller is busy now
ERR_MEMORYVOLUME	4	Out of the Memory Volume
ERR_CHECKSUM	5	CRC16 Checksum Error
ERR_FILENOTEXIST	6	File Not Exist
ERR_FLASH	7	Flash Access Error
ERR_FILE_DOWNLOAD	8	File Download Error
ERR_FILE_NAME	9	Filename Error
ERR_FILE_TYPE	10	File type Error
ERR_FILE_CRC16	11	File CRC16 Error
ERR_FONT_NOT_EXIST	12	Font Library Not Exist
ERR_FIRMWARE_TYPE	13	Firmware Type Error (Check the controller type)
ERR_DATE_TIME_FORMAT	14	Date Time format error
ERR_FILE_EXIST	15	File Exist for File overwrite
ERR_FILE_BLOCK_NUM	16	File block number error
ERR_COMMUNICATE	100	通信失败
ERR_PROTOCOL	101	协议数据不正确
ERR_TIMEOUT	102	通信超时
ERR_NETCLOSE	103	通讯断开
ERR_INVALID_HAND	104	无效句柄
ERR_PARAMETER	105	参数错误
ERR_SHOULDREPEAT	106	需要重复上次的错误包
ERR_FILE	107	无效文件

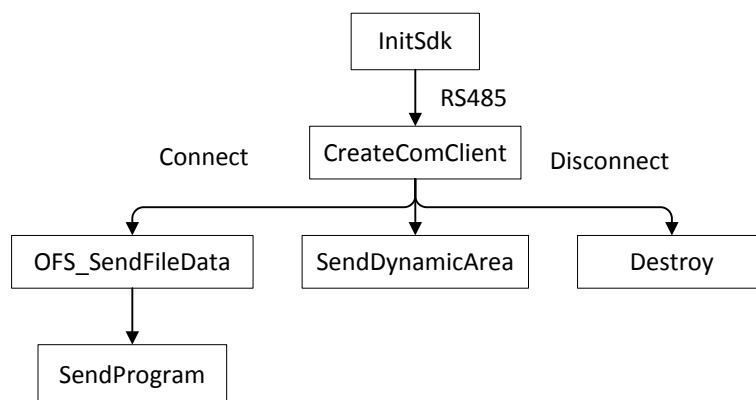
2. 显示屏参数初始化说明

控制器接入动态库使用的系统前需要首先使用我司提供的 LedshowTW2013 软件来进行控制卡地址、通讯波特率、网络 IP(如有网口)、端口地址(如有网口)、扫描方式等设置；设置好后就可以再接入本动态库的系统中按照之前设定好的相关参数来设定显示屏的其它参数、信息和命令了。

3. API调用顺序

3.1 总体次序

在执行其他操作之前，首先执行函数[InitSDK](#)初始化动态库，然后选择通讯模式，设置不同模式下的参数，连接控制器(广播模式执行函数[CreateBroadCast](#)，网络固定ip模式执行函数[CreateClient](#)，网络modbus模式执行函数[CreateTcpModbus](#)，485总线模式执行函数[CreateComClient](#)，串口modbus模式执行函数[CreateComModbus](#))，设置网络超时时间执行函数[SetTimeout](#)。之后可以执行图文区节目（发送节目[OFS_SendFileData](#)、删除节目[OFS_DeleteFile](#)、锁定/解锁节目[SCREEN_LockProgram](#)）、动态区节目（发送节目[SCREEN_SendDynamicArea](#)、删除节目[SCREEN_DelDynamicArea](#)）、开关（强制开关机[SCREEN_ForceOnOff](#)，定时开关机[SCREEN_TimeTurnOnOff](#)，取消定时开关机[SCREEN_CancelTimeOnOff](#)）、固件查询（查询固件信息[CON_CheckCurrentFirmware](#)，激活固件[CON_FirmwareActivate](#)）、控制器状态查询（[CON_ControllerStatus](#)）、（亮度调整[SCREEN_SetBrightness](#)，复位[CON_Reset](#)，ping [CON_PING](#)，校时[CON_SytemClockCorrect](#)）等操作。例如RS485模式中函数的调用：



4. 函数说明

4.1 连接控制卡

对屏幕进行任何操作前，都要通过 [InitSdk](#) 函数初始化动态库。该函数不对显示屏进行任何通讯。

4.1.1 InitSdk函数

函数：

```
public static extern void InitSdk(byte minorVer , byte minorVer );
```

参数：

minorVer=2 、 minorVer=2 为固定值，不能改变，无意义。

说明：

初始化动态库

4.2 CreateBroadCast函数

函数：

```
public static extern uint CreateBroadCast(byte[] broad_ip, uint broad_port, bx_5k_card_type card_type);
```

参数

参数	描述
broad_ip	广播 IP
broad_port,	端口号
card_type	控制卡 枚举类型

说明：

通过广播模式连接控制卡，设置广播 IP、端口号、控制卡类型。

4.2.1 CreateClient函数

函数：

```
public static extern uint CreateClient(byte[] led_ip, uint led_port, bx_5k_card_type card_type, CallbackClientClose pCloseFunc);
```

参数：

参数	描述
led_ip	控制卡 IP
led_port,	端口号
card_type	控制卡 枚举类型
pCloseFunc	回调函数，参数值为null

说明：

通过网络固定 IP 模式连接控制卡，设置控制卡 IP、端口号、控制卡类型，回调函数：

```
[System.Runtime.InteropServices.UnmanagedFunctionPointerAttribute(System.Runtime.InteropServices.CallingConvention.StdCall)]
```

```
public delegate void CallbackClientClose(uint hand, int err);
```


4.2.2 CreateTcpModbus函数

函数:

```
public static extern uint CreateTcpModbus(byte[] led_ip, bx_5k_card_type card_type,
CallbackClientClose pCloseFunc);
```

参数:

参数	描述
led_ip	控制卡 IP
card_type	控制卡 枚举类型
pCloseFunc	回调函数，参数值为null

说明:

通过网络 modbus 模式连接控制卡，设置控制卡 IP，选择控制卡，回调函数:

```
[System.Runtime.InteropServices.UnmanagedFunctionPointerAttribute(System.Runtime.InteropServices.
CallingConvention.StdCall)]
```

```
public delegate void CallbackClientClose(uint hand, int err);
```

4.2.3 CreateComClient函数

函数:

```
public static extern uint CreateComClient(byte com, uint baudrate, bx_5k_card_type card_type, ushort
ScreenID);
```

参数:

参数	描述
com	控制卡 IP
baudrate	波特率
card_type	控制卡 枚举类型
ScreenID	屏号

说明:

通过网络 modbus 模式连接控制卡，设置控制卡 IP，选择波特率和控制卡、设置屏号

4.2.4 CreateComModbus函数

函数:

```
public static extern uint CreateComModbus(byte com, uint baudrate, serial_parity Parity, serial_databits
DataBits, serial_stopbits StopBits, bx_5k_card_type card_type, ushort ScreenID);
```

参数:

参数	描述
com	控制卡 IP
baudrate	波特率
Parity	串口校验模式 枚举类型 serial_parity
DataBits	串口数据位 枚举类型 serial_databits
StopBits	串口停止位 枚举类型 serial_stopbits
card_type	控制卡 枚举类型 bx_5k_card_type
ScreenID	屏号

说明:

通过串口 Modbus 模式连接控制卡，设置控制卡 IP、波特率、校验模式、串口数据位、以及停止位、并设置屏号。

4.2.5 SetTimeout 函数

函数：

```
public static extern void SetTimeout(uint dwHand, uint nSec);
```

参数：

dwHand 连接控制卡函数的返回值、nSec 设置超时时间。

说明：

连接到控制卡后，调用该函数设置发送节目信息的超时时间

4.3 图文区节目

图文节目要求不频繁更新，以保护控制器的使用寿命，该类节目支持字库和词库，断电后保存。

4.3.1 OFS_SendFileData 函数

函数：

```
public static extern int OFS_SendFileData(uint dwHand, byte overwrite, byte[] pFileName, ushort DisplayType, byte PlayTimes, byte[] ProgramLife, byte ProgramWeek, byte ProgramTime, byte[] Period, byte AreaNum, byte[] AreaDataList, int AreaDataListLen);
```

返回值：

详见[错误码及说明](#)。

参数：

参数	描述
dwHand	dwHand 连接控制卡函数的返回值
overwrite	设置默认参数为 1
pFileName	文件名 PXXX XXX 为文件编号，ASCII 码表示 文件名为字符串，发送按顺序发送，如“P123”则 先发送‘P’，最后发送‘3’。
DisplayType	节目播放方式默认设置为 0——顺序播放
PlayTimes	节目重复播放次数默认设置为 1
ProgramLife	节目生命周期即有效时间范围，默认设置为 ProgramLife = newbyte[8]; ProgramLife[0] ~ ProgramLife[7] = 0xff; 表示播放始终有效
ProgramWeek	节目的星期属性，默认设置为 1 表示一周都有效
ProgramTime	节目定时播放参数，当前为非 0，表示非定时
Period	节目播放时段，当前设置为 null 表示全天有效
AreaNum	节目播放个数
AreaDataList	区域数据参考 区域数据格式设置
AreaDataListLen	区域 数据长度

说明：

通过指定的通讯模式，连接到控制卡，向指定的图文区域发送图文节目，该节目断电后会自动保存。

4.3.2 OFS_DeleteFile 函数

函数:

```
public static extern int OFS_DeleteFile(uint dwHand, ushort FileNumber, byte[] pFileNameList);
```

返回值:

详见[错误码及说明](#)。

参数:

参数	描述
dwHand	连接控制卡函数的返回值
FileNumber	删除当前图文节目个数为 1，删除所有节目该参数值为 0
pFileNameList	删除当前图文节目的节目名，删除所有节目该参数值为 null

说明:

该函数用来删除图文区节目，若删除当前显示的节目，函数的参数 FileNumber=1，pFileNameList 的值为要删除的图文节目名；若删除所有节目，则函数的参数 FileNumber=0，pFileNameList=null

4.3.3 SCREEN_LockProgram 函数

函数:

```
public static extern int SCREEN_LockProgram(uint dwHand, byte LockFlag, byte StoreMode, byte[] ProgramFileName);
```

返回值:

详见[错误码及说明](#)。

参数:

参数	描述
dwHand	连接控制卡函数的返回值
LockFlag	当前图文节目锁定为 1，解锁为 0
StoreMode	默认为 0
ProgramFileName	当前显示的图文节目名

说明:

该函数是解锁/锁定当前的图文节目，若为是锁定节目则函数参数 LockFlag=1；解锁 LockFlag=0；

4.3.4 OFS_Formatting 函数

函数:

```
public static extern int OFS_Formatting(uint dwHand);
```

返回值:

详见[错误码及说明](#)。

参数:

参数	描述
dwHand	连接控制卡函数的返回值

说明:

该函数是格式化控制卡数据，一般不要轻易格式化操作。

4.4 动态区节目

动态区节目，掉电后该类节目不保存，但该类节目可无限次更新。

4.4.1 SCREEN_SendDynamicArea 函数

函数：

```
public static extern uint SCREEN_SendDynamicArea(uint dwHand, bx_5k_area_header header, ushort TextLen, byte[] AreaText);
```

返回值：

详见[错误码及说明](#)。

参数：

参数	说明
dwHand	连接控制卡函数的返回值
header	结构体类型 bx_5k_area_header ，动态区域数据参数，参数设置见 区域数据格式设置
TextLen	动态区数据长度
AreaText	动态区数据

说明：

该函数向指定的动态区域发送动态区节目，参考“区域数据格式设置”设置相应的参数，添加节目内容，计算内容长度。节目断电后不会保存。

4.4.2 SCREEN_DelDynamicArea 函数

函数：

```
public static extern int SCREEN_DelDynamicArea(uint dwHand, byte DeleteAreald);
```

返回值：

详见[错误码及说明](#)。

参数：

参数	说明
dwHand	连接控制卡函数的返回值
DeleteAreald	删除动态区节目序号

说明：

该函数是根据节目序号删除指定的动态节目。

4.5 固件操作

4.5.1 CON_CheckCurrentFirmware 函数

函数：

```
public static extern int CON_CheckCurrentFirmware(uint dwHand, byte[] FirmwareName, byte[] FirmwareVersion, byte[] FirmwareDateTime);
```

返回值：

详见[错误码及说明](#)。

参数:

参数	说明
dwHand	连接控制卡函数的返回值
FirmwareName	固件名称
FirmwareVersion	固件版本
FirmwareDateTime	固件日期

说明:

查询控制卡版本信息。

4.5.2 CON_FirmwareActivate 函数

函数:

```
public static extern int CON_FirmwareActivate(uint dwHand, byte[] FirmwareName);
```

返回值:

详见[错误码及说明](#)。

参数:

参数	说明
dwHand	连接控制卡函数的返回值
FirmwareName	固件名

说明:

激活固件。

4.6 开关机

4.6.1 SCREEN_ForceOnOff 函数

函数:

```
public static extern int SCREEN_ForceOnOff(uint dwHand, byte OnOffFlag);
```

返回值:

详见[错误码及说明](#)。

参数:

参数	说明
dwHand	连接控制卡函数的返回值。
nProgramOrd	参数值为 1 开机，2 关机

说明:

强制开关机

4.6.2 SCREEN_TimeTurnOnOff 函数

函数:

```
public static extern int SCREEN_TimeTurnOnOff(uint dwHand, byte[] pTimer, int nGroup);
```

返回值:

详见[错误码及说明](#)。

参数:

参数	说明
dwHand	连接控制卡函数的返回值。
pTimer	定时器 定时开机时间，BCD 码表示，格式如下： 时（1）+分（1） 定时关机时间，BCD 码表示，格式如下： 时（1）+分（1）
nGroup	分组编号

说明:

设置定时开关机

4.6.3 SCREEN_CancelTimeOnOff函数**函数:**

```
public static extern int SCREEN_CancelTimeOnOff(uint dwHand);
```

返回值:

详见[错误码及说明](#)。

参数:

参数	说明
dwHand	连接控制卡函数的返回值。

说明:

取消定时开关机设置

4.7 其它命令**4.7.1 CON_PING 函数****函数:**

```
public static extern int CON_PING(uint dwHand)
```

返回值:

详见[错误码及说明](#)。

参数:

参数	说明
dwHand	连接控制卡函数的返回值。

说明:

若 ping 命令执行成功，说明与控制卡通信成功。

4.7.2 CON_Reset 函数**函数:**

```
public static extern int CON_Reset(uint dwHand);
```

返回值:

详见[错误码及说明](#)。

参数：

参数	说明
dwHand	连接控制卡函数的返回值。

说明：

控制器接收到系统复位命令后，擦除除 Firmware 文件外的所有数据文件，然后复位 CPU。

4.7.3 CON_SytemClockCorrect 函数

函数：

```
public static extern int CON_SytemClockCorrect(uint dwHand);
```

返回值：

详见[错误码及说明](#)。

参数：

参数	说明
dwHand	连接控制卡函数的返回值。

说明：

校正系统时间。

4.7.4 SCREEN_SetBrightness 函数

函数：

```
public static extern int SCREEN_SetBrightness(uint dwHand, byte BrightnessType, byte CurrentBrightness, byte[] BrightnessValue);
```

返回值：

详见[错误码及说明](#)。

参数：

参数	说明
dwHand	连接控制卡函数的返回值。
BrightnessType	亮度调整方式 0x01——强制调节亮度 0x02——定时调节亮度
CurrentBrightness	当前亮度，强制调节亮度时有效。亮度值为 0-15 共 16 级。亮度值为 15 时亮度最高
BrightnessValue	亮度值列表，用于定时调亮 把一天分为 48 个时段，每 30 分钟为一个时段，此 处 48 个字节为每个时段对应的亮度值

说明：

调整屏幕亮度。

4.8 查询控制卡状态

4.8.1 CON_ControllerStatus函数

函数：

```
public static extern int CON_ControllerStatus(uint dwHand, byte[] pStatus, ref ushort len);
```

返回值：

详见[错误码及说明](#)。

参数：

参数	说明
dwHand	连接控制卡函数的返回值。
pStatus	具体见代码 控制器状态查询
len	设置默认值为 0

说明：

查询控制器当前的图文节目(个数和节目名)、动态节目(列表和个数)、开机状态、亮度值、以及控制卡时间。

附录

控制器类型

```
public enum bx_5k_card_type : byte{
    BX_Any = 0xFE,
    BX_5K1 = 0x51,
    BX_5K2 = 0x58,
    BX_5MK2 = 0x53,
    BX_5MK1 = 0x54,
}
```

串口数据位

```
public enum serial_databits : byte{
    COM_4BITS = 4, //4 位
    COM_5BITS = 5, //5 位
    COM_6BITS = 6, //6 位
    COM_7BITS = 7, //7 位
    COM_8BITS = 8, //8 位
}
```

串口校验模式

```
public enum serial_parity : byte{
    COM_NOPARITY = 0,    //无校验
    COM_ODDPARITY = 1,   //奇校验
    COM_EVENPARITY = 2,  //偶校验
    COM_MARKPARITY = 3,  //标记校验
    COM_SPACEPARITY = 4, //空格校验
}
```

串口停止位

```
public enum serial_stopbits : byte{
    COM_ONESTOPBIT = 0,    //1 位
    COM_ONE5STOPBITS = 1,  //1.5 位
    COM_TWOSTOPBITS = 2,   //2 位
}
```

结构体定义数据

```
public struct bx_5k_area_header{
    public byte AreaType;
    public short AreaX;
    public short AreaY;
    public short AreaWidth;
    public short AreaHeight;
    public byte DynamicAreaLoc;
    public byte Lines_sizes;
    public byte RunMode;
    public short Timeout;
    public byte Reserved1;
    public byte Reserved2;
}
```

```

public byte Reserved3;
public byte SingleLine;
public byte NewLine;
public byte DisplayMode;
public byte ExitMode;
public byte Speed;
public byte StayTime;
public int DataLen;
}

```

控制器状态查询

```

private void button35_Click(object sender, EventArgs e){
    listBox1.Items.Clear();
    byte[] pStatus = new byte[1024];
    for (int i = 0; i < pStatus.Length; i++)
    {
        pStatus[i] = 0;
    }
    ushort len = 0;
    int err = ONNONLed5KSDKD.Led5kSDK.CON_ControllerStatus(m_dwCurHand,
pStatus, ref len);
    if (err == 0)
    {
        byte onoff = pStatus[0];
        comboBox6.SelectedIndex = onoff - 1;
        byte brightness = pStatus[1];
        byte[] date = new byte[20];

        ushort year = Convert.ToUInt16(pStatus[3] / 16 * 10 + pStatus[3] % 16
+ pStatus[2] / 16 * 1000 + pStatus[2] % 16 * 100);
        byte month = Convert.ToByte(pStatus[4] / 16 * 10 +
pStatus[4] % 16);
        byte day = Convert.ToByte(pStatus[5] / 16 * 10 +
pStatus[5] % 16);
        byte hour = Convert.ToByte(pStatus[7] / 16 * 10 +
pStatus[7] % 16);
        byte minute = Convert.ToByte(pStatus[8] / 16 * 10 +
pStatus[8] % 16);
        byte sec = Convert.ToByte(pStatus[6] / 16 * 10 +
pStatus[6] % 16);
        byte week = Convert.ToByte(pStatus[9] / 16 * 10 +
pStatus[9] % 16);
        string str;
        str = brightness.ToString();
        textBox24.Text = str;
        textBox23.Text = string.Format("{0}-{1}-{2} {3}:{4}:{5}",
year, month, day, hour, minute, sec);

        byte programnum = pStatus[10];
        textBox20.Text = string.Format("{0:d}", programnum);
        byte[] name = new byte[5];
        for (int i = 0; i < name.Length; i++)
        {
            name[i] = 0;

```

```
    }
    for (int j = 0; j < 5; j++)
    {
        name[j] = pStatus[j + 11]; ;
    }
    byte[] cur_program = new byte[5];
    name.CopyTo(cur_program, 0);
    textBox21.Text =
System.Text.Encoding.Default.GetString(cur_program);

    //-----
    byte SpecialDynaArea = pStatus[16];
    int z = pStatus[16];
    comboBox12.SelectedIndex = z;

    byte PageNum = pStatus[18];
    textBox26.Text = PageNum.ToString();

    byte DynaAreaNum = pStatus[17];
    textBox22.Text = DynaAreaNum.ToString();

    for (int i = listBox1.Items.Count; i > 0; i--)
    {
        listBox1.Items.Remove(i - 1);
    }
    for (int i = 0; i < DynaAreaNum; i++)
    {
        str = pStatus[18 + i].ToString();
        listBox1.Items.Add(str);
    }
}
}
```

区域数据格式设置

参数	数据长度	默认值	描述
AreaType	1	0x00	区域类型
AreaX	2		区域 X 坐标，以字节(8 个像素点)为单位
AreaY	2		区域 Y 坐标，以像素点为单位
AreaWidth	2		区域宽度，以字节(8 个像素点)为单位
AreaHeight	2		区域高度，以像素点为单位
DynamicAreaLoc	1	0xFF	动态区域编号注意：该参数只对动态区有效，其他区域为默认值，动态区必须统一编号，编号从 0 开始递增。
Lines_sizes	1		行间距
RunMode	1		动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时删除动态区信息。

Timeout	2		动态区数据超时时间，单位为秒
Reserved	3	0x00	保留字
SingleLine	1	0x02	是否单行显示 0x01——单行显示 0x02——多行显示
NewLine	1		是否自动换行 0x01——不自动换行，显示数据在换行时必须插入换行符 0x02——自动换行，显示内容不需要换行符，但是只能使用统一的中文体和英文字体
DisplayMode	1		显示方式，其定义如下： 0x01——静止显示 0x02——快速打出 0x03——向左移动 0x04——向右移动 0x05——向上移动 0x06——向下移动
ExitMode	1	0x00	退出方式
Speed	1		显示速度，定义如下： 0x00——最快 0x01—— 0x17—— 0x18——最慢
StayTime	1		显示特技停留时间，单位为 0.5s
DataLen			数据长度（包括换行，颜色等转义参数）