

C-Power5200 开发 API 接口手册

版本: V1.6

2015.06.24

修订记录:

日期	版本	修改内容	执行人
2009-8-11	V1.0	第一版本	
2010-01-28	V1.1	1. 增加多窗口协议数据打包 API 2. 增加多窗口协议串口和网络简单应用 API	
2010-5-22	V1.2	增加下列函数说明: 1. CP5200_Program_AddLafPict 2. CP5200_Program_AddLafVideo 3. CP5200_Program_AddVariable 4. CP5200_MakeGetTypeInfoData 5. CP5200_ParseGetTypeInfoRet 6. CP5200_MakeGetTempHumiData 7. CP5200_ParseGetTempHumiRet 8. CP5200_MakeReadConfigData 9. CP5200_ParseReadConfigRet 10. CP5200_MakeWriteConfigData 11. CP5200_ParseWriteConfigRet 12. CP5200_RS232_GetTemperature 13. CP5200_RS232_GetTypeInfo 14. CP5200_Net_GetTemperature 15. CP5200_Net_GetTypeInfo	
2011-02-24	V1.3	增加以下接口: 1. CP5200_MakeReadHWSettingData 2. CP5200_ParseReadHWSettingRet 3. CP5200_MakeWriteHWSettingData 4. CP5200_ParseWriteHWSettingRet 5. CP5200_RS232_ReadHWSetting 6. CP5200_RS232_WriteHWSetting 7. CP5200_Net_ReadHWSetting 8. CP5200_Net_WriteHWSetting	
2012-03-16		增加以下接口: 1. CP5200_RS232_RemoveFile 2. CP5200_Net_RemoveFile	
2012-08-04	V1.4	1. 增加以下接口说明: CP5200_CommData_SetParam CP5200_MakeReadRunningInfoData	

		CP5200_ParseReadRunningInfoRet CP5200_MakeScreenTestData CP5200_ParseScreenTestRet CP5200_MakeInstantMessageData1 CP5200_CmmPacket_SetParam CP5200_MakePlaySelectedPrgData1 CP5200_MakeSetZoneAndVariableData CP5200_ParseSetZoneAndVariableRet CP5200_MakeSendPureTextData CP5200_ParseSendPureTextRet CP5200_RS232_SetZoneAndVariable CP5200_RS232_SendPureText CP5200_Net_SetZoneAndVariable CP5200_Net_SendPureText CPowerBox_MakeSendClockOrTemperatureData CPowerBox_ParseSendClockOrTemperatureRet CPowerBox_MakeSetAloneProgramData CPowerBox_ParseSetAloneProgramRet CPowerBox_MakeQueryProgramData CPowerBox_ParseQueryProgramRet CPowerBox_MakeSetScheduleData CPowerBox_ParseSetScheduleRet CPowerBox_MakeDeleteScheduleData CPowerBox_ParseDeleteScheduleRet CPowerBox_MakeGetScheduleData CPowerBox_ParseGetScheduleRet CPowerBox_RS232_SendClockOrTemperature CPowerBox_RS232_SetAloneProgram CPowerBox_RS232_QueryProgram CPowerBox_RS232_SetSchedule CPowerBox_RS232_DeleteSchedule CPowerBox_RS232_GetSchedule CPowerBox_Net_SendClockOrTemperature CPowerBox_Net_SetAloneProgram CPowerBox_Net_QueryProgram CPowerBox_Net_SetSchedule CPowerBox_Net_DeleteSchedule CPowerBox_Net_GetSchedule CP5200_Net_SetBindParam 2. 完善部分接口的参数说明	
2012-05-08		增加以下接口说明： CP5200_Program_AddFormattedText CP5200_Program_AddFormattedTextW CP5200_TextToImage	

		CP5200_TextToImageW	
2013/8/5		增加以下接口： CP5200_MakeReadSoftwareSwitchInfoData CP5200_ParseReadSoftwareSwitchInfoRet CP5200_MakeWriteSoftwareSwitchInfoData CP5200_ParseWriteSoftwareSwitchInfoRet CP5200_RS232_ReadSoftwareSwitchInfo CP5200_RS232_WriteSoftwareSwitchInfo CP5200_Net_ReadSoftwareSwitchInfo CP5200_Net_WriteSoftwareSwitchInfo	
2014/1/23		增加以下接口说明： CP5200_Program_AddFormattedTextEx CP5200_TextToImageEx	
2015/3/17		增加以下接口说明： CP5200_MakeQueryControllerInfo CP5200_ParseQueryControllerInfoRet CP5200_Net_QueryControllerInfo	
2015/4/1		增加以下接口说明： CP5200_RS232_ReadNetworkParam CP5200_RS232_WriteNetworkParam CP5200_RS232_Upgrade CP5200_Net_ReadNetworkParam CP5200_Net_WriteNetworkParam CP5200_Net_Upgrade	
2015/5/18		增加以下接口说明： CP5200_MakeSendMultiProtocol CP5200_ParseSendMultiProtocolRet CP5200_Net_SendMultiProtocol CP5200_RS232_SendMultiProtocol	
2015/5/27		修改以下接口说明： CP5200_MakeSendPictureData CP5200_RS232_SendPicture CP5200_Net_SendPicture 三个函数中的 pPictureFile 和 nPictRef 的说明 增加以下接口说明：： CP5200_MakeSendSimpleImageData CP5200_ParseSendSimpleImageRet CP5200_RS232_SendSimpleImageData	
2015/6/24		增加以下接口说明：： CP5200_RS232_QueryControllerInfo	

1、概述

1.1、数据类型定义

名称	类型	定义
对象句柄	HOBJECT	void*

1.2、API 接口函数分类

- 生成节目文件的 API 接口函数
- 生成播放列表文件的 API 接口函数
- 生成通信数据和解析返回数据的 API 接口函数

1.3、一般操作步骤

- 1、生成节目文件
- 2、生成节目单文件
- 3、利用生成通信数据和解析返回数据的 API 接口函数处理通信相关的数据，实现对控制卡的操作。

注意：控制卡启动时只查找节目单文件”playbill.lpp”，若生成数据时保存为其它文件名，在将节目单文件(“.lpp”)发送到卡上的时候，需要改文件名为”playbill.lpp”。

1.4、通信约定

控制卡支持“RS232/485 通信方式”和“网络通信方式”。

1.4.1、RS232/485 通信方式

数据以 RS232/485 方式通信时，以 0xA5 作为开始码，以 0xAE 作为结束码。

在开始码和结束码之间的 0xA5, 0xAA, 0xAE 数据需要进行转码。

PC 端发送数据时将 1 个码变为 2 个码:

0xA5 → 0xAA 0x05

0xAA → 0xAA 0x0A

0xAE → 0xAA 0x0E

PC 端接收数据时将 2 个码变为 1 个码:

0xAA 0x05 → 0xA5

0xAA 0x0A → 0xAA

0xAA 0xAE → 0xAE

1.4.2、网络通信方式

发送数据时，需在数据的开始部分加上控制卡的识别码。

1.5、文字、图片显示效果编码

编码值	显示特效
0	立即显示（无特效）
1	左开
2	右开
3	横中开
4	竖中开
5	百叶（垂直）
6	左移
7	右移
8	上移
9	下移
10	向上滚动
11	向左滚动
12	向右滚动

13	闪烁
14	连续向左滚动
15	连续向右滚动
16	百叶（水平）
17	顺时针展开
18	逆时针展开
9	风车
20	风车（逆时针）
21	矩形向外
22	矩形向内
23	四角向外
24	四角向内
25	圆形向外
26	圆形向内
27	左上角展开
28	右上角展开
29	左下角展开
30	右下角展开
31	斜角展开
32	反斜角展开
33	左上角进入
34	右上角进入
35	左下角进入
36	右下角进入
37	斜角进入
38	反斜角进入
39	水平斑马线
40	垂直斑马线
41	马赛克（大）
42	马赛克（小）
43	放射线（向上）
44	放射线（向下）
45	积聚
46	下落
47	合并（水平）
48	合并（垂直）
49	旋出
50	旋入
51	棋盘（水平）
52	棋盘（垂直）
53	连续向上滚动
54	连续向下滚动
55	保留

56	保留
57	逐步变大（上）
58	逐步变大（下）
59	保留
60	逐步变大（垂直）
61	闪动（水平）
62	闪动（垂直）
63	飘雪
64	向下滚动
65	左右滚动
66	上下展开
67	扇形展开
68	保留
69	斑马条（水平）
70	斑马条（垂直）
65536	随机

1.6、文本扩展标记

含扩展标记的文本可以包含以下的扩展标记，所有扩展标记名称必须是小写

扩展标记	说明
<size>	指定文字的大小, 必须带上属性值才起作用, 否则被忽略, 属性值无效也被忽略。属性值是文字大小。有效值如下: <size=8> : 8 点阵字 <size=16> : 16 点阵字 <size=24> : 24 点阵字 <size=32> : 32 点阵字
<color>	指定文字的颜色, 必须带上属性值才起作用, 否则被忽略, 属性值无效也被忽略。属性值是颜色的 RGB 值, 十六进制。 例如: <color=#ff0000> : 红色

	<p><color=#00ff00> : 绿色</p> <p><color=#0000ff> : 蓝色</p>
<p>	换行
<align>	<p>水平对齐方式, 必须带上属性值才起作用, 否则被忽略, 属性值无效也被忽略。有效值如下:</p> <p><align=left> : 左对齐</p> <p><align=center> : 居中</p> <p><align=right> : 右对齐</p>
	<p>指定文字的字体, 必须带上属性值才起作用, 否则被忽略, 属性值无效也被忽略。支持 0~7 共 8 种字体。</p> <p>例如:</p> <p><font=0> : 缺省字体, 没有字体类型标签时</p> <p><font=1> : 第一种字体</p> <p>.....</p> <p><font=7> : 第七种字体</p>

1.7、文字大小及类型编码

1.7.1、文字大小编码

编码值	文字大小 (点)
0	8
1	12
2	16
3	24
4	32
5	40
6	48
7	56

1.7.2、文字类型编码

编码值	文字类型
0	字体 0 缺省字体
1	字体 1
2	字体 2
3	字体 3
4	字体 4
5	字体 5
6	字体 6
7	字体 7

说明：如未做特殊说明，本文档中的函数的名为“nFontSize”的参数均按如下格式定义：

Byte 0~1: 文字大小（点），如 8、12、24、32、40、48、56

Byte 2: **Bit 0~2**，文字类型编码值；

Bit 3，是否用每个字符指定的字体（0 用缺省字体，1 用每个字符指定字体），
添加标记文本节目时设置为 1，其他时候设置为 0

Bit4~7，保留。

Byte 3: 保留

1.8、文字颜色编码

单字节文字颜色编码

可表示 8 种颜色。红绿蓝颜色值各用 1 个位表示。

最低位代表红色

次低位代表绿色

第三低位代表蓝色

3 单字节文字颜色编码

可表示任意颜色。红绿蓝颜色值各用 1 个字节表示。

1.9、图片方式编码

编码值	图片处理方式
0	居中
1	按比例缩放
2	拉伸
3	平铺

1.10、时钟格式和显示内容

时钟格式:

用一个字节表示:

bit 0: 时制(0: 12 小时制; 1: 24 小时制)

bit 1: 年份位数(0: 4 位; 1: 2 位)

bit 2: 分行(0: 单行; 1: 多行)

bit 3~5: 保留(设置为 0)

bit 6: 显示时间刻度“时标、分标”

bit 7: 保留(设置为 0)

时钟显示内容:

用一个字节表示:

按位确定要显示的内容。

bit 7: 指针

bit 6: 星期

bit 5: 秒

bit 4: 分

bit 3: 时

bit 2: 日

bit 1: 月

bit 0: 年

1.11、简单图片数据格式

数据构成:

数据头	红色数据(可选)	绿色数据(可选)	蓝色数据(可选)
-----	----------	----------	----------

数据头说明:

	0	1	2	3	4	5	6	7
0x00	标识		宽度		高度		属 性	保 留

说明:

数据名称	数据大小(字节)	说明
标识	2	固定为“11”。
宽度	2	图片的宽度。低字节在前
高度	2	图片的高度。低字节在前
属性	1	图片的灰度以及颜色 Bit0: 是否存在红色数据, 为 1 时存在。 Bit1: 是否存在绿色数据, 为 1 时存在。

		<p>Bit2: 是否存在蓝色数据, 为 1 时存在。</p> <p>Bit3: 保留, 设置为 0</p> <p>Bit4~7: 灰度级别, 现支持 0 和 7 两个值</p> <p> 0: 2 级灰度, 每个点的数据用 1bit 表示</p> <p> 7: 256 级灰度, 每个点的数据用 8bit 表示</p> <p>图像的每行数据总是按字节对齐, 对于 2 级灰度的图片, 行末不足 8bit 时, 以 0 补足。</p>
保留	1	设置为 0

数据说明:

数据分颜色按照红绿蓝的顺序存放, 若属性中的某个颜色的标记位为 0, 则该颜色的数据不存在。

对于某个颜色的数据, 按照“从左到右, 从上到下”的顺序存放。先放第一行的数据, 再放第二行的数据。

1.12、全局区域信息格式

每个区域信息占 16 个字节, 其格式如下:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	类型	方式	x	y	cx	cy	ItemPropData									

说明:

数据名称	数据大小(字节)	说明
类型	1	1: 显示文本 2: 显示指定图片文件 3: 时钟 4: 温度 5: 湿度 6: 提示文字(变量里'\n'之后的) 7: 计时
方式	1	按窗口类型不同, 含义不同。 当窗口类型为 1 或 6 时, 指文本对齐方式: 0 为左对齐, 1 为右对齐, 2 为居中; 当窗口类型为 2 时, 指图片显示方式 0 为居中, 1 为比例缩放, 2 为拉伸, 3 为平铺 其他窗口类型, 暂时忽略此值, 默认为 0
x	2	区域起点 X。高字节在前
y	2	区域起点 Y。高字节在前
cx	2	区域宽度。高字节在前
cy	2	区域高度。高字节在前

ItemPropData	6	各区域属性数据，含义依赖于窗口类型
--------------	---	-------------------

ItemPropData 详细说明

1.显示文本

	A	B	C	D	E	F
0x00	起 始	结 束	停 留		字 体 颜 色	保 留

说明：

数据名称	数据大小(字节)	说明
起始	1	起始变量号，有效值 1~100
结束	1	结束变量号，有效值 1~100
停留	2	每个有效变量停留的时间，单位为秒。高字节在前
字体颜色	1	<p>Bit0~3: 字体大小，含义如下：</p> <p>0: 8 点 1: 12 点 2: 16 点 3: 24 点 4: 32 点 5: 40 点 6: 48 点 7: 56 点 8: 64 点</p> <p>Bit4~6: 颜色，含义如下：</p> <p>0: 黑色 1: 红色 2: 绿色 3: 黄色 4: 蓝色 5: 紫红色 6: 青色 7: 白色</p> <p>Bit7: 是否反色，1 为反色</p>

2.显示指定图片文件

	A	B	C	D	E	F
--	---	---	---	---	---	---

0x00	起 始	结 束	停留	保 留	保 留
------	--------	--------	----	--------	--------

说明:

数据名称	数据大小(字节)	说明
起始	1	起始变量号, 有效值 1~100
结束	1	结束变量号, 有效值 1~100
停留	2	每个有效变量停留的时间, 单位为秒。高字节在前

3.时钟

4.温度

5.湿度

6.提示文字

	A	B	C	D	E	F
0x00	提示窗口号		停留		字体颜色	保留

说明:

数据名称	数据大小(字节)	说明
提示窗口号	2	按位标记哪些窗口需要此窗口提示, 1 为提示, 0 为不提示 Bit 0: 表示 1 号窗口 Bit 1: 表示 2 号窗口 Bit 15: 表示 16 号窗口
停留	2	每个有效提示停留的时间, 单位为秒。高字节在前
字体颜色	1	含义同“1.显示文本”的字体颜色

- 说明: 1. 当“提示窗口号”指定的窗口本身就为“6.提示文字”类型窗口时将忽略;
2. 当“提示窗口号”指定的窗口里的变量值不存在'\n'或'\n'后没有数据, 该变量将不参与提示。
3. 停留时间在同步时为同步切换窗口停留时间的最大值除以该窗口提示的同步切换窗口数。

7.计时

	A	B	C	D	E	F
--	---	---	---	---	---	---

0x00	计数器号	显示格式	停留	字体颜色	保留
------	------	------	----	------	----

说明:

数据名称	数据大小(字节)	说明
计数器号	1	计数器编号
显示格式	1	0: “mm:ss” 1: “mm:ss:nn” 2: “hh:mm:ss” 3: “hh:mm:ss:nn”
停留	2	每个有效提示停留的时间，单位为秒。高字节在前
字体颜色	1	含义同“1.显示文本”的字体颜色

1.13、窗口位置及属性

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	x		y		cx		cy		窗口属性							

说明:

数据名称	数据大小(字节)	说明
x	2	窗口起点 X。高字节在前
y	2	窗口起点 Y。高字节在前
cx	2	窗口宽度。高字节在前
cy	2	窗口高度。高字节在前
窗口属性	8	窗口缺省类型以及参数

窗口属性统一用 8 字节表示:

	0	1	2	3	4	5	6	7
0x00	类型	参数						

窗口类型的定义:

类型值	类型说明
0	空白（不显示任何内容）
1	文本
2	时钟日历
3	温度、湿度
4	图片、图片引用
其它	保留

参数根据类型不同而不同。以下是各种类型的参数，所有“保留”的位置填写数据 0x00。

空白类型

	0	1	2	3	4	5	6	7
0x00	0	保留						

文本类型

	0	1	2	3	4	5	6	7
0x00	1	方式	文字大小	文字颜色	速度	停留时间		保留

数据项	数值	长度(字节)	意义描述
方式	1	1	见”1.7”节的说明
文字大小		1	Bit0~2: 文字大小 0x00: 8 点阵文字 (仅限英文) 0x01: 12 点阵文字 (仅限英文) 0x02: 16 点阵文字 0x03: 24 点阵文字 0x04: 32 点阵文字
文字颜色		1	Bit0~2: 文字颜色 0x01: 红色 0x02: 绿色 0x03: 黄色 0x04: 蓝色 红、绿、蓝颜色按位组合可组成其他颜色

时钟日历类型

	0	1	2	3	4	5	6	7
0x00	2	文字大小	文字颜色	停留时间	历法	格式	内容	

数据项	数值	长度(字节)	意义描述
文字大小		1	Bit0~2: 文字大小 0x00: 8 点阵文字 (仅限英文) 0x01: 12 点阵文字 (仅限英文) 0x02: 16 点阵文字 0x03: 24 点阵文字 0x04: 32 点阵文字
文字颜色		1	Bit0~2: 文字颜色 0x01: 红色 0x02: 绿色 0x03: 黄色 0x04: 蓝色 红、绿、蓝颜色按位组合可组成其他颜色
停留时间	0x0000~0xffff	2	高字节在前。单位是秒。
历法		1	0: 公历时间日期
格式		1	bit 0: 时制(0: 12 小时制; 1: 24 小时制) bit 1: 年份位数(0: 4 位; 1: 2 位) bit 2: 分行(0: 单行; 1: 多行) bit 3~5: 保留(设置为 0) bit 6: 显示时间刻度“时标、分标” bit 7: 保留(设置为 0)
内容		1	按位确定要显示的内容。 bit 7: 指针 bit 6: 星期 bit 5: 秒 bit 4: 分 bit 3: 时 bit 2: 日 bit 1: 月 bit 0: 年

温度、湿度类型

	0	1	2	3	4	5	6	7
0x00	3	文字大小	文字颜色	停留时间	格式	保留		

数据项	数值	长度(字节)	意义描述
-----	----	--------	------

文字大小		1	Bit0~2: 文字大小 0x00: 8 点阵文字 (仅限英文) 0x01: 12 点阵文字 (仅限英文) 0x02: 16 点阵文字 0x03: 24 点阵文字 0x04: 32 点阵文字
文字颜色			Bit0~2: 文字颜色 0x01: 红色 0x02: 绿色 0x03: 黄色 0x04: 蓝色 红、绿、蓝颜色按位组合可组成其他颜色
停留时间	0x0000~0xffff	2	高字节在前。单位是秒。
格式		1	0: 摄氏度 1: 华氏度 2: 湿度

图片、图片引用

	0	1	2	3	4	5	6	7
0x00	4	方 式	速 度	停 留 时 间	保 留			

数据项	数值	长度(字节)	意义描述
方式		1	见”1.7”节的说明
速度	0~9	1	数值越小，速度越快。立即显示时此值无效。
停留时间	0x0000~0xffff	2	高字节在前。单位是秒。

1.14、扫描参数各字节含义

扫描参数一共 16 个字节，设置扫描参数和读取扫描参数时都要用到，其各字节的含义如下：

字节	字节含义	CPower3200/2200/1200 取值说明	CPower5200/4200 取值说明
0x00	列顺序	0: 正 1: 负	0: 正 1: 负
0x01	数据极性	0: 正 1: 负	0: 正 1: 负

0x02	OE 极性	对 CPower3200/2200 无效, 设置为 0x00; 对于 CPower1200: 0: OE 正 1: OE 负	0: 正 1: 负
0x03	行调整	0: 调整-1 1: 调整 0 2: 调整 1, 3: 调整 2	0: 调整 0 1: 调整 1 2: 调整 2 3: 调整-1
0x04	消隐	0: 无消隐 1: 消隐	0: 无消隐 1: 前消隐 2: 后消隐 3: 表示前后消隐
0x05	颜色顺序	0: 红-绿 1: 绿-红	0: 红-绿-蓝 1: 红-蓝-绿 2: 绿-红-蓝 3: 绿-蓝-红 4: 蓝-红-绿 5: 蓝-绿-红
0x06	移位时钟模式	0: 6Mhz 1: 12Mhz	0~15: 表示模式 1~模式 16
0x07	时序微调	不存在此参数, 设置为 0x00	0: 微调 1 1: 微调 2 2: 微调 3 3: 微调 4
0x08	脉宽微调	不存在此参数, 设置为 0x00	0: 微调 1 1: 微调 2 2: 微调 3 3: 微调 4
0x09	扫描模式	0: 1/16 扫 1: 1/8 扫 2: 1/4 扫 3: 1/2 扫 4: 静态	0: 1/16 扫 1: 1/8 扫 2: 1/4 扫 3: 1/2 扫 4: 静态
0x0A	单位模块大小	0: 16 行 1: 8 行 2: 4 行 3: 2 行 4: 1 行	0: 16 行 1: 8 行 2: 4 行 3: 2 行 4: 1 行
0x0B	换行间隔	0: 每隔 4 点换行 1: 每隔 8 点换行 2: 每隔 16 点换行 3: 每隔 32 点换行	0: 每隔 8 点换行 1: 每隔 4 点换行 2: 每隔 16 点换行 3: 每隔 32 点换行
0x0C	换行方向	0: 正	0: 正

		1: 负	1: 负
0x0D	模块内逆序	0: 无 1: 奇行逆序 2: 偶行逆序 3: 所有	0: 无 1: 8 点逆序 2: 4 点逆序 3: 16 点逆序 4: 32 点逆序
0x0E	接口板类型	0: 标准型 1: 扩展型	0: 类型 1 1: 类型 2 2: 类型 3 3: 类型 4
0x0F	折行逆序	不存在此参数, 设置为 0x00	0: 无折行逆序 2: 偶行逆序 3: 奇行逆序

1.15、格式化文本节目参数含义

格式控制数据

数据名称	数据大小(字节)	说明
长度	2	格式定义内容字节长度 (低字节在前) (不包括此长度数据)
字体磅值	2	字体的磅值 (低字节在前)
标记	1	Bit0-7: Bit 0: 加粗 Bit 1: 斜体 Bit 2: 下划线
对齐方式	1	Bit0~1: 水平对齐 (0:左对齐, 1:水平居中, 2:右对齐) Bit2~3: 垂直对齐(0:上对齐, 1: 垂直居中, 2: 下对齐) 其他: 保留
分页方式	1	Bit0~1: 分页方式 (0: 不分页, 1: 水平分页, 2: 垂直分页) Bit 2: 分页字符串是否按字符断开 (1: 是; 0: 否) Bit 3: 是否生成覆盖整页的图片数据 (1: 是; 0: 否)
字体前景色	4	BYTE 0: RED; 1: GREEN; 2: BLUE; 3: 0
字体背景色	4	BYTE 0: RED; 1: GREEN; 2: BLUE; 3: 0
页面背景色	4	BYTE 0: RED; 1: GREEN; 2: BLUE; 3: 0
行高	1	字符串每行的高 (单位: 像素)

Y 偏移	1	垂直方向上的偏移量（单位：像素）
------	---	------------------

屏数据

数据名称	数据大小(字节)	说明
长度	2	屏参内容字节长度（低字节在前） (不包括此长度数据)
屏宽	2	像素（低字节在前）
屏高	2	像素（低字节在前）
颜色	1	Bit0~2: 颜色分量 Bit4~6: 灰度 其它: 保留

1.16、扩展格式化文本节目参数含义

扩展格式字符串内容

数据名称	数据大小(字节)	说明
长度	2	格式定义内容字节长度（低字节在前） (不包括此长度数据)
字符串编码	1	0x00: 多字节 0x01: 宽字节
子串段的个数	1	字符串的子串段的个数（一行字符串代表一个子串段）
子串段	变长	见下面的 子串段格式 定义

子串段格式

数据名称	数据大小(字节)	说明
长度	2	格式定义内容字节长度（低字节在前） (不包括此长度数据)
对齐方式	1	Bit0~1: 水平对齐（0:左对齐，1:水平居中，2:右对齐）
子串的个数	1	子串段的子串的个数
子串	变长	见下面的 子串格式 定义

子串格式

数据名称	数据大小(字节)	说明
长度	2	格式定义内容字节长度（低字节在前） (不包括此长度数据)
标记	1	Bit0-7: Bit 0: 加粗 Bit 1: 斜体 Bit 2: 下划线
字体前景色	4	BYTE 0: RED; 1: GREEN; 2: BLUE; 3: 0
字体背景色	4	BYTE 0: RED; 1: GREEN; 2: BLUE; 3: 0
字体磅值	2	字体的磅值（低字节在前）
字体名称长度	1	字体名称长度
字体名称	变长	字体名称字符串，以 0x00 结尾
字符串长度	2	字符串长度（低字节在前）
字符串	变长	字符串，以 0x00 结尾

扩展格式格式控制数据

数据名称	数据大小(字节)	说明
长度	2	格式定义内容字节长度（低字节在前） (不包括此长度数据)
对齐方式	1	Bit0~1: 垂直对齐(0:上对齐,1: 垂直居中,2: 下对齐) 其他: 保留
分页方式	1	Bit0~1: 分页方式 (0: 不分页, 1: 水平分页, 2: 垂直分页) Bit 2: 分页字符串是否按字符断开 (1: 是; 0: 否) Bit 3: 是否生成覆盖整页的图片数据 (1: 是; 0: 否)
页面背景色	4	BYTE 0: RED; 1: GREEN; 2: BLUE; 3: 0
行高	1	字符串每行的高（单位：像素）
Y 偏移	1	垂直方向上的偏移量（单位：像素）

扩展格式屏数据

数据名称	数据大	说明
------	-----	----

	小(字节)	
长度	2	屏参内容字节长度（低字节在前） (不包括此长度数据)
屏宽	2	像素（低字节在前）
屏高	2	像素（低字节在前）
颜色	1	Bit0~2: 颜色分量 Bit4~6: 灰度 其它: 保留

2、生成节目的 API 接口

2.1、节目生成 API 接口总览

序号	API 接口函数名	接口说明
1	CP5200_Program_Create	创建节目对象
2	CP5200_Program_Destroy	销毁节目对象
3	CP5200_Program_SetProperty	设置节目属性
4	CP5200_Program_SetBackgndImage	设置节目的背景图片
5	CP5200_Program_AddPlayWindow	给节目增加播放窗
6	CP5200_Program_SetWindowProperty	设置窗口属性
7	CP5200_Program_SetItemProperty	设置播放项属性
8	CP5200_Program_AddText CP5200_Program_AddText1	增加文本项到节目的指定播放窗
9	CP5200_Program_AddTagText CP5200_Program_AddTagText1	增加包含扩展标记的文本项到节目的指定播放窗
10	CP5200_Program_AddFormattedText	增加格式化的文本项到节目指定播放窗
11	CP5200_Program_AddFormattedTextW	增加格式化的文本项到节目指定播放窗(宽字符)
12	CP5200_Program_AddFormattedTextEx	增加扩展格式化的文本项到节目指定播放窗
13	CP5200_Program_AddPicture	增加图片项到节目的指定播放窗
14	CP5200_Program_AddImage	增加图片项到节目的指定播放窗
15	CP5200_Program_AddLafPict	添加 LAF 图片项到节目的指定播放窗
16	CP5200_Program_AddLafVideo	添加 LAF 动画项到节目的指定播放窗
17	CP5200_Program_AddAnimator	增加动画项到节目的指定播放窗
18	CP5200_Program_AddClock	增加时钟项到节目的指定播放窗
	CP5200_Program_AddClockEx	增加时钟项到节目的指定播放窗
19	CP5200_Program_AddTemperature	增加温度项到节目的指定播放窗
20	CP5200_Program_AddVariable	增加自定义变量数据到指定播放窗

21	CP5200_Program_AddTimeCounter	增加定时器数据到指定播放窗
22	CP5200_Program_SaveToFile	保存节目到文件

本类函数的使用步骤如下：

- 第1步：创建节目对象
- 第2步：增加播放窗
- 第3步：增加播放项到指定播放窗
- 第4步：保存节目到文件
- 第5步：销毁节目对象

2.2、节目生成 API 接口详细描述

CP5200_Program_Create

HOBJECT CP5200_Program_Create (WORD width, WORD height, BYTE color)	
作用	创建节目对象
参数	width: 显示屏宽度，以点为单位
	height: 显示屏高度，以点为单位
	color: 颜色和灰度级别。单红色 0x01; 单红 256 级灰度 0x71; 双色 0x03; 双色 256 级灰度 0x73; 七彩 0x07; 全彩 256 级灰度 0x77
返回值	节目对象句柄，供本类 API 接口函数使用 返回“空”表示创建失败
其它说明	对象创建成功并不再使用时，必须将对象销毁

CP5200_Program_Destroy

int CP5200_Program_Destroy (HOBJECT hObj)	
作用	销毁节目对象
参数	hObj: 要销毁的节目对象句柄
返回值	0: 无错误 -1: 不正确的节目对象句柄
其它说明	

CP5200_Program_SetProperty

int CP5200_Program_SetProperty(HOBJECT hObj, int nPropertyValue, DWORD nPropertyID)	
作用	设置节目属性
参数	hObj: 节目对象句柄
	nPropertyValue: 属性值, 根据“属性识别”参数的不同有不同的含义 节目循环播放次数的范围是 1~65535 节目播放时间的单位是秒, 取值范围 1~65535 编码转换: 0 不转换; 1 简体中文转繁体中文; 2 繁体中文转简体中文;
	nPropertyID: 属性识别, 可以是以下值之一 1: 节目循环播放次数 2: 节目播放时间 3: 文字编码转换。
返回值	-1: 不正确的节目对象句柄 0: 不认识的属性识别 >0: 设置成功
其它说明	“节目循环播放次数”和“节目播放时间”二者只有一个有效, 以最后一次设置为有效。

CP5200_Program_SetBackgndImage

int CP5200_Program_SetBackgndImage(HOBJECT hObj, const BYTE* pImgDat, WORD wImgWidth, WORD wImgHeight, BYTE color, int nMode, int nCompress)	
作用	设置节目的背景图片
参数	hObj: 节目对象句柄

	<p>pImgDat: 图片数据缓冲区。根据参数 color 的值确定有哪种颜色的数据，以及数据的格式。多种颜色的数据同时存在时，先红色数据，再放绿色数据，最后是蓝色数据。</p> <p>对于每个颜色的数据，先放第 1 行的数据，再放第二行的数据，每个像素的数据表示根据 color 参数的高 4 位来确定。</p>
	<p>wImgWidth: 图片宽度的点数</p>
	<p>wImgHeight: 图片高度的点数</p>
	<p>color: 图片数据包含哪些颜色的数据，以及数据的格式。</p> <p>最 4 位表示包含哪些颜色的数据，可以是以下值的组合：</p> <p>0x01: 存在红色数据</p> <p>0x02: 存在绿色数据</p> <p>0x04: 存在蓝色数据</p> <p>高 4 位表示数据的格式（灰度级别），支持 2 种格式：</p> <p>0x0: 二值图像。按照图像的水平行，每 8 个点的数据组成一个字节，行末不足 8 点时，以 0 补足一个字节；一种颜色的数据量(单位:字节) 是: $((\text{图片宽度} + 7) / 8) * \text{图片高度}$。</p> <p>0x7: 256 级灰度数据。每个点用 1 字节表示；一种颜色的数据量(单位:字节)是: $\text{图片宽度} * \text{图片高度}$。</p> <p>例如：0x71 表示 256 级灰度图片，只存在红色数据</p>
	<p>nMode: 显示时的处理方式</p> <p>0: 居中</p> <p>1: 按比例缩放</p> <p>2: 拉伸</p> <p>3: 平铺</p>
	<p>nCompress: 压缩图片数据。现只支持不压缩的方式</p> <p>0: 不压缩</p>
返回值	<p>-1: 不正确的节目对象句柄</p> <p>-4: 内存不足</p>

其它说明	
------	--

CP5200_Program_AddPlayWindow

int CP5200_Program_AddPlayWindow(HOBJECT hObj, WORD x, WORD y, WORD cx, WORD cy)	
作用	给节目增加播放窗
参数	hObj: 节目对象句柄
	x: 播放窗起始 X 坐标
	y: 播放窗起始 Y 坐标
	cx: 播放窗宽度
	cy: 播放窗高度
返回值	>=0: 播放窗号 -1: 不正确的节目对象句柄 -3: 参数错误
其它说明	

CP5200_Program_SetWindowProperty

int CP5200_Program_SetWindowProperty(HOBJECT hObj, int nWinNo, int nPropertyValue, int nPropertyID);	
作用	设置窗口属性
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号

	<p>nPropertyValue:属性值</p> <p>当属性 ID 为 1 时，其值的含义如下：</p> <p>bit0~1: 边框速度(值越小越快)</p> <p>bit2: 边框背景色(1 为绿色,0 为黑色)</p> <p>bit3: 边框前景色(1 为红色,0 为白色)</p> <p>bit4~6: 边框动作方式(0 无边框,1 单点,2 虚线,3 十字,4 追逐)</p> <p>bit7: 边框滚动方向(0 顺时针,1 逆时针)</p> <p>当属性 ID 为 2 时，其值的含义如下：</p> <p>0:静止， 1:循环播放， 2:隐藏</p>
	<p>nPropertyID:属性 ID， 可以是以下值之一：</p> <p>1: 设置窗口边框属性</p> <p>2: 设置窗口等待类型</p>
返回值	<p>>=0: 播放窗号</p> <p>-1: 不正确的节目对象句柄</p> <p>-3: 参数错误</p>
其它说明	

CP5200_Program_SetItemProperty

<pre>int CP5200_Program_SetItemProperty(HOBJECT hObj, int nWinNo, int nItem, int nPropertyValue, int nPropertyID);</pre>	
作用	设置播放项属性
参数	hObj: 节目对象句柄
	nWinNo:播放窗号
	nItem:播放项编号， AddText 等函数返回
	nPropertyValue: 颜色值， RGB， 负值取消透明颜色
	nPropertyID: 值为 1 时， 表示设置透明颜色
返回值	<p>1: 成功</p> <p>-1: 不正确的节目对象句柄</p> <p>-3: 参数错误</p>

其它说明	
------	--

CP5200_Program_AddText

(CP5200_Program_AddText1)

<pre>int CP5200_Program_AddText(HOBJECT hObj, int nWinNo, const char* pText, int nFontSize, COLORREF crColor, int nEffect, int nSpeed, int nStay)</pre>	
作用	增加文本项到指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pText: 文本字符串
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	crColor: 文字颜色
	nEffect: 文字显示效果
	nSpeed: 文字显示速度
nStay: 停留时间	
返回值	>=0: 播放项编号 -1: 不正确的节目对象句柄 -3: 播放窗号无效 -4: 内存不足
其它说明	CP5200_Program_AddText1 接口用于纯粹单字节文字(英文、欧洲文字)节目

CP5200_Program_AddTagText

(CP5200_Program_AddTagText1)

<pre>int CP5200_Program_AddTagText(HOBJECT hObj, int nWinNo, const char* pText, int nFontSize, COLORREF crColor, int nEffect, int nSpeed, int nStay)</pre>	
作用	增加文本项到指定播放窗

参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pText: 文本字符串, 可包含扩展标记。参见”1.6”节
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	crColor: 文字颜色
	nEffect: 文字显示效果
	nSpeed: 文字显示速度
	nStay: 停留时间
返回值	<p>>=0: 播放项编号</p> <p>-1: 不正确的节目对象句柄</p> <p>-3: 播放窗号无效</p> <p>-4: 内存不足</p>
其它说明	<p>缺省使用参数中的文字大小、颜色等属性值, 若文本中包含扩展标记, 则从扩展标记开始, 使用扩展标记指定的值。</p> <p>CP5200_Program_AddTagText1 接口用于纯粹单字节文字 (英文、欧洲文字) 节目</p>

CP5200_Program_AddFormattedText

<pre>int CP5200_Program_AddFormattedText (HOBJECT hObj, int nWinNo, const char *pText, const char *pFontFaceName, const byte *pFormatData, const byte *pScreenData, int nMode, int nEffect, int nSpeed, int nStay, int nCompress)</pre>	
作用	增加格式化的文本项到节目指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pText: 文本字符串
	pFontFaceName: 字体名称
	pformatData: 格式控制数据
	pScreenData: 屏数据

	nMode: 显示时的处理方式 0: 居中 1: 按比例缩放 2: 拉伸 3: 平铺 4: 左上角对齐 8: 垂直分页 11: 水平分页 (左对齐) 12: 水平分页 (右对齐)
	nEffect: 图片显示效果
	nSpeed: 效果展示速度。0 最快
	nStay: 停留时间, 单位为秒
	nCompress: 压缩图片数据 0: 不压缩 1: 转换为 256 色图片并压缩数据
返回值	>=0: 播放项编号 -1: 不正确的节目对象句柄 -3: 播放窗号无效 -4: 内存不足 -5: 格式字符串相关数据不合法
其它说明	

CP5200_Program_AddFormattedTextW

<pre>int CP5200_Program_AddFormattedTextW(HOBJECT hObj, int nWinNo, const wchar_t *pText, const char *pFontFaceName, const byte *pFormatData, const byte *pScreenData, int nMode, int nEffect, int nSpeed, int nStay, int nCompress)</pre>	
作用	增加格式化的文本项到节目指定播放窗 (宽字符)
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号

	pText: 文本字符串
	pFontFaceName: 字体名称
	pformatData: 格式控制数据
	pScreenData: 屏数据
	nMode: 显示时的处理方式 0: 居中 1: 按比例缩放 2: 拉伸 3: 平铺 4: 左上角对齐 8: 垂直分页 11: 水平分页（左对齐） 12: 水平分页（右对齐）
	nEffect: 图片显示效果
	nSpeed: 效果展示速度。0 最快
	nStay: 停留时间，单位为秒
	nCompress: 压缩图片数据 0: 不压缩 1: 转换为 256 色图片并压缩数据
	返回值
其它说明	

CP5200_Program_AddFormattedTextEx

```
int CP5200_Program_AddFormattedTextEx( HOBJECT hObj, int nWinNo, const byte
*pTextContent, const byte *pFormatData, const byte *pScreenData, int nMode, int nEffect,
```

int nSpeed, int nStay, int nCompress);	
作用	增加扩展格式化的文本项到节目指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pTextContent: 扩展格式字符串内容
	pFormatData: 扩展格式格式控制数据
	pScreenData: 扩展格式屏数据
	nMode: 显示时的处理方式 0: 居中 1: 按比例缩放 2: 拉伸 3: 平铺 4: 左上角对齐 8: 垂直分页 11: 水平分页（左对齐） 12: 水平分页（右对齐）
	nEffect: 图片显示效果
	nSpeed: 效果展示速度。0 最快
	nStay: 停留时间，单位为秒
	nCompress: 压缩图片数据 0: 不压缩 1: 转换为 256 色图片并压缩数据
	返回值
其它说明	

CP5200_Program_AddPicture

<pre>int CP5200_Program_AddPicture(HOBJECT hObj, int nWinNo, const char* pPictFile, int nMode, int nEffect, int nSpeed, int nStay, int nCompress)</pre>		
作用	增加图片项到指定播放窗	
参数	hObj: 节目对象句柄	
	nWinNo: 播放窗号	
	pPictFile: 图片文件路径和名称	
	nMode: 显示时的处理方式 0: 居中 1: 按比例缩放 2: 拉伸 3: 平铺 4: 左上角对齐 8: 垂直分页 11: 水平分页（左对齐） 12: 水平分页（右对齐）	
	nEffect: 图片显示效果	
	nSpeed: 效果展示速度。0 最快	
	nStay: 停留时间，单位为秒	
	nCompress: 压缩图片数据 0: 不压缩 1: 转换为 256 色图片并压缩数据	
	返回值	>=0: 播放项编号 -1: 不正确的节目对象句柄 -3: 播放窗号无效 -4: 内存不足
	其它说明	

CP5200_Program_AddImage

<pre>int CP5200_Program_AddImage(HOBJECT hObj, int nWinNo, const BYTE* pImgDat, WORD wImgWidth, WORD wImgHeight, BYTE color, int nMode, int nEffect, int nSpeed, int nStay, int nCompress, int nPageCount)</pre>	
作用	添加图片项到节目的指定播放窗口
参数	hObj : 节目对象句柄
	nWinNo : 播放窗口号
	pImgDat : 图片数据缓冲区。根据参数 color 的值确定有哪种颜色的数据以及灰度。按红绿蓝的顺序存放各颜色数据，没有则跳过。对于每个颜色的数据，其格式为：按照图像的水平行，每 8 个点的数据组成一个字节，行末不足 8 点时，以 0 补足一个字节；先放第 1 行的数据，再放第二行的数据；一种颜色的数据量是： : 二级灰度为 $((\text{图片宽度} + 7) / 8) * \text{图片高度}$, 256 级灰度为 $\text{图片宽度} * \text{图片高度}$;
	wImgWidth : 图片宽度的点数
	wImgHeight : 图片高度的点数
	color : : 图片颜色和灰度: bit 0: 是否存在红色数据, 0 不存在, 1 存在 bit 1: 是否存在绿色数据, 0 不存在, 1 存在 bit 2: 是否存在蓝色数据, 0 不存在, 1 存在 bit 4~7: 0 为 2 级灰度 7 为 256 级灰度
	nMode : 显示时的处理方式 0: 居中 1: 按比例缩放 2: 拉伸 3: 平铺
	nEffect : 图片显示效果

	nSpeed: 效果展示速度。0 最快
	nStay: 停留时间，单位为秒，高位在前。
	nCompress: 压缩图片数据 0: 不压缩 1: 转换为 256 色图片并压缩数据
	nPageCount: 页数
返回值	>=0: 播放项编号 -1: 不正确的节目对象句柄 -3: 播放窗号无效 -4: 内存不足
其它说明	图片大小必须与播放窗大小一致

CP5200_Program_AddLafPict

int CP5200_Program_AddLafPict(HOBJECT hObj, int nWinNo, const char* pLafFile, int nMode, int nEffect, int nSpeed, int nStay, int nCompress)	
作用	添加 LAF 图片项到节目的指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pLafFile: 图片文件路径和名称
	nMode: 显示时的处理方式 0: 居中 1: 按比例缩放 2: 拉伸 3: 平铺
	nEffect: 图片显示效果
	nSpeed: 效果展示速度。0 最快
	nStay: 停留时间，单位为秒

	<p>nCompress: 压缩图片数据</p> <p>0: 不压缩</p> <p>1: 转换为 256 色图片并压缩数据</p>
返回值	<p>>=0: 播放项编号</p> <p>-1: 不正确的节目对象句柄</p> <p>-3: 播放窗号无效</p> <p>-4: 内存不足</p>
其它说明	

CP5200_Program_AddLafVideo

<pre>int CP5200_Program_AddLafVideo(HOBJECT hObj, int nWinNo, const char* pLafFile, int nMode, int nRepeat)</pre>	
作用	添加 LAF 动画项到节目的指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pLafFile: 动画文件路径和名称
	nMode: 显示时的处理方式
	<p>0: 居中</p> <p>1: 按比例缩放</p> <p>2: 拉伸</p> <p>3: 平铺</p>
	nRepeat: 动画循环播放的次数
返回值	<p>>=0: 播放项编号</p> <p>-1: 不正确的节目对象句柄</p> <p>-3: 播放窗号无效</p> <p>-4: 内存不足</p>
其它说明	

CP5200_Program_AddAnimator

int CP5200_Program_AddAnimator(HOBJECT hObj, int nWinNo, const char* pAniFile, int nMode, int nRepeat)	
作用	增加动画项到指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pAniFile: 动画文件(.gif)路径和名称
	nMode: 显示时的处理方式 0: 居中 1: 按比例缩放 2: 拉伸 3: 平铺
	nRepeat: 动画循环播放的次数
返回值	>=0: 播放项编号 -1: 不正确的节目对象句柄 -3: 播放窗号无效 -4: 内存不足
其它说明	

CP5200_Program_AddClock

int CP5200_Program_AddClock(HOBJECT hObj, int nWinNo, const char* pText, int nFontSize, COLORREF crColor, int nStay, WORD wAttr)	
作用	增加时钟项到指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pText: 文本字符串
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	crColor: 文字颜色

	<p>nStay: 停留时间</p> <hr/> <p>wAttrib: 时钟属性,按位确定要显示的内容</p> <p>bit 0: 显示年</p> <p>bit 1: 显示月</p> <p>bit 2: 显示日</p> <p>bit 3: 显示小时</p> <p>bit 4: 显示分钟</p> <p>bit 5: 显示秒</p> <p>bit 6: 显示星期</p> <p>bit 7: 显示指针</p> <p>bit 8: 时制(0: 12 小时制; 1: 24 小时制)</p> <p>bit 9: 年份位数(0: 4 位; 1: 2 位)</p> <p>bit 10: 分行(0: 单行; 1: 多行)</p> <p>bit 11~13: 格式控制, 如 2010 年 11 月 12 日星期五 的格式根据取值不同分别表示如下:</p> <p>0: 2010 年 11 月 12 日 星期五 16:20:30</p> <p>1: Fri, 12/11/2010 16:20:30</p> <p>2: 2010-11-12 Fri. 16:20:30</p> <p>3: Friday, 12 November 2010 16:20:30</p> <p>4: Fri, Nov 12,2010 16:20:30</p> <p>5: Friday, November 12 2010 16:20:30</p> <p>6: Fri, 11/12/2010 16:20:30</p> <p>7: 2010/11/12, Fri.16:20:30</p> <p>bit 14: 显示时标、分标</p> <p>bit 15: 透明显示</p>
返回值	<p>>=0: 播放项编号</p> <p>-1: 不正确的节目对象句柄</p> <p>-3: 播放窗号无效</p>

	-4: 内存不足
其它说明	

CP5200_Program_AddClockEx

<pre>int CP5200_Program_AddClockEx(HOBJECT hObj, int nWinNo, const char* pText, int nFontSize, COLORREF crColor, int nStay, WORD wAttr, WORD wAttrEx)</pre>	
作用	增加时钟项到指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pText: 文本字符串
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	crColor: 文字颜色
	nStay: 停留时间

	<p>wAttrib: 时钟属性,按位确定要显示的内容</p> <p>bit 0: 显示年</p> <p>bit 1: 显示月</p> <p>bit 2: 显示日</p> <p>bit 3: 显示小时</p> <p>bit 4: 显示分钟</p> <p>bit 5: 显示秒</p> <p>bit 6: 显示星期</p> <p>bit 7: 显示指针</p> <p>bit 8: 时制(0: 12 小时制; 1: 24 小时制)</p> <p>bit 9: 年份位数(0: 4 位; 1: 2 位)</p> <p>bit 10: 分行(0: 单行; 1: 多行)</p> <p>bit 11~13: 格式控制, 如 2010 年 11 月 12 日星期五 的格式根据取值不同分别表示如下:</p> <p>0: 2010 年 11 月 12 日 星期五 16:20:30</p> <p>1: Fri, 12/11/2010 16:20:30</p> <p>2: 2010-11-12 Fri. 16:20:30</p> <p>3: Friday, 12 November 2010 16:20:30</p> <p>4: Fri, Nov 12,2010 16:20:30</p> <p>5: Friday, November 12 2010 16:20:30</p> <p>6: Fri, 11/12/2010 16:20:30</p> <p>7: 2010/11/12, Fri.16:20:30</p> <p>bit 14: 显示时标、分标</p> <p>bit 15: 透明显示</p>
--	---

	<p>wAttrEx: 调整时间值。</p> <p>BYTE1: 正负、分钟值</p> <p> bit0~5: 分值 0~59</p> <p> bit6: “:” 闪烁</p> <p> bit7: 数值符号, 0 正值, 1 负值</p> <p>BYTE2: 小时值</p> <p> bit0~4: 小时值 0~31</p> <p> bit5~7: 保留</p>
返回值	<p>>=0: 播放项编号</p> <p>-1: 不正确的节目对象句柄</p> <p>-3: 播放窗号无效</p> <p>-4: 内存不足</p>
其它说明	

CP5200_Program_AddTemperature

<pre>int CP5200_Program_AddTemperature(HOBJECT hObj, int nWinNo, const char* pText, int nFontSize, COLORREF crColor, int nStay, WORD wAttr)</pre>	
作用	增加温度项到指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	pText: 文本字符串
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	crColor: 文字颜色
	nStay: 停留时间
	<p>wAttrib: 温度属性</p> <p> 0: 设置温度</p> <p> 1: 华氏温度</p>
返回值	<p>>=0: 播放项编号</p> <p>-1: 不正确的节目对象句柄</p>

	-3: 播放窗号无效 -4: 内存不足
其它说明	

CP5200_Program_AddVariable

<pre>int CP5200_Program_AddVariable(HOBJECT hObj, int nWinNo, int nFontSize, COLORREF crColor, int nStay, WORD wAttr)</pre>	
作用	增加自定义变量数据到指定播放窗
参数	hObj: 节目对象句柄
	nWinNo: 播放窗号
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	crColor: 文字颜色
	nStay: 停留时间
	wAttr: 自定义数据属性 高字节为格式: 意义如下: 0: 文本数据。直接显示变量文本 1: 图片数据。显示变量文本指定的图片 低字节为变量编号,有效值 1~100, 指定使用哪个用户变量
返回值	>=0: 播放项编号 -1: 不正确的节目对象句柄 -3: 播放窗号无效 -4: 内存不足
其它说明	

CP5200_Program_AddTimeCounter

<pre>int CP5200_Program_AddTimeCounter(HOBJECT hObj, int nWinNo, int nFontSize, COLORREF crColor, int nStay, int nOption, const int* pBaseTime, const char* pContent)</pre>	
作用	增加定时器数据到指定播放窗
参数	hObj: 节目对象句柄

	nWinNo: 播放窗号
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	crColor: 文字颜色
	nStay: 停留时间
	nOption: 显示属性 第 1 字节: 指定“格式”, Bit0: 计时类型。0 正计时, 1 倒计时 Bit1~7: 保留 第 2 字节: 指定“对齐”, Bit0~1: 水平对齐; Bit2~3: 垂直对齐
	pBaseTime: 长度为 6 的整形数组首地址, 分别存储年月日时分秒
	pContent:
返回值	>=0: 播放项编号 -1: 不正确的节目对象句柄 -3: 播放窗号无效 -4: 内存不足
其它说明	

CP5200_Program_SaveToFile

int CP5200_Program_SaveToFile(HOBJECT hObj, const char* pFilename)	
作用	保存节目到文件
参数	hObj: 节目对象句柄
	pFilename: 文件路径和名称
返回值	>=0: 文件大小 -1: 不正确的节目对象句柄 -3: 文件操作失败
其它说明	

3、生成节目单的 API 接口

3.1、节目单生成 API 接口总览

序号	API 接口函数名	接口说明
1	CP5200_Playbill_Create	创建节目单对象
2	CP5200_Playbill_Destroy	销毁节目单对象
3	CP5200_Playbill_SetProperty	设置节目单属性
4	CP5200_Playbill_AddFile	增加节目文件到节目单
5	CP5200_Playbill_DelFile	从节目单中删除节目文件
6	CP5200_Playbill_SaveToFile	保存节目单到文件

本类函数的使用步骤如下：

- 第1步：创建节目单对象
- 第2步：增加节目文件到节目单
- 第3步：保存节目单到文件
- 第4步：销毁节目单对象

3.2、节目单生成 API 接口详细描述

CP5200_Playbill_Create

HOBJECT CP5200_Playbill_Create(WORD width, WORD height, BYTE color)	
作用	创建节目单对象
参数	width: 显示屏宽度，以点为单位
	height: 显示屏高度，以点为单位
	color: 颜色和灰度级别。单红色 0x01; 单红 256 级灰度 0x71; 双色 0x03; 双色 256 级灰度 0x73; 七彩 0x07; 全彩 256 级灰度 0x77
返回值	节目单对象句柄，供本类 API 接口函数使用 返回“空”表示创建失败
其它说明	对象创建成功并不再使用时，必须将对象销毁

CP5200_Playbill_Destroy

int CP5200_Playbill_Destroy(HOBJECT hObj)	
作用	销毁节目单对象
参数	hObj: 要销毁的节目单对象句柄
返回值	0: 无错误 -1: 节目单对象句柄空 -2: 不正确的节目单对象句柄
其它说明	

CP5200_Playbill_SetProperty

int CP5200_Playbill_SetProperty(HOBJECT hObj, int nPropertyValue, DWORD nPropertyID)	
作用	设置节目单属性
参数	hObj: 节目对象句柄
	nPropertyValue: 属性值, 根据“属性识别”参数的不同有不同的含义 旋转: 0: 不旋转; 1: 旋转 90 度。
	nPropertyID: 属性识别, 可以是以下值之一 1: 旋转
返回值	-1: 不正确的节目对象句柄 0: 不认识的属性识别 >0: 设置成功
其它说明	。

CP5200_Playbill_AddFile

int CP5200_Playbill_AddFile(HOBJECT hObj, const char* pFilename)	
作用	增加节目到节目单的末尾
参数	hObj: 节目单对象句柄
	pFilename: 节目文件路径和名称

返回值	>=0: 添加成功 -1: 不正确的节目单对象句柄 -3: 节目文件非短文件名
其它说明	

CP5200_Playbill_DelFile

int CP5200_Playbill_DelFile(HOBJECT hObj, const char* pFilename)	
作用	从节目单中删除节目
参数	hObj: 节目单对象句柄
	pFilename: 节目文件路径和名称
返回值	>=0: 删除成功 -1: 不正确的节目单对象句柄 -3: 节目文件非短文件名
其它说明	

CP5200_Playbill_SaveToFile

int CP5200_Playbill_SaveToFile(HOBJECT hObj, const char* pFilename)	
作用	保存节目单到文件
参数	hObj: 节目单对象句柄
	pFilename: 文件路径和名称
返回值	0: 无错误 -1: 不正确的节目单对象句柄 -3: 文件操作失败
其它说明	

4、数据通信 API 接口

4.1、数据通信 API 接口总览

序号	API 接口函数名	接口说明
1	CP5200_CommData_Create	创建数据对象
2	CP5200_CommData_Destroy	销毁数据对象
3	CP5200_CommData_SetParam	设置数据包参数
4	CP5200_MakeCreateFileData	生成创建文件所需的数据
5	CP5200_ParseCreateFileRet	解析返回的创建文件命令的数据
6	CP5200_MakeWriteFileData	生成写文件命令的数据
7	CP5200_ParseWriteFileRet	解析返回的写文件命令的数据
8	CP5200_MakeCloseFileData	生成关闭文件命令的数据
9	CP5200_ParseCloseFileRet	解析返回的关闭文件命令的数据
10	CP5200_MakeDeleteFileNoData	生成根据文件号删除文件命令的数据
11	CP5200_ParseDeleteFileNoRet	解析返回的根据文件号删除文件命令的数据
12	CP5200_MakeDeleteFileNameData	生成根据文件名删除文件命令的数据
13	CP5200_ParseDeleteFileNameRet	解析返回的根据文件名删除文件命令的数据
14	CP5200_MakeReadTimeData	生成查询控制卡上当前时间命令的数据
15	CP5200_ParseReadTimeRet	解析返回的查询控制卡当前时间命令的数据
16	CP5200_MakeWriteTimeData	生成设置时间命令的数据
17	CP5200_ParseWriteTimeRet	解析返回的设置控制卡上时间命令的数据
18	CP5200_MakeReadBrightnessData	生成查询控亮度控制信息命令的数据
19	CP5200_ParseReadBrightnessRet	解析返回的查询亮度控制信息命令的数据
20	CP5200_MakeWriteBrightnessData	生成设置亮度控制信息命令的数据
21	CP5200_ParseWriteBrightnessRet	解析返回的设置亮度控制信息命令的数据
22	CP5200_MakeWriteIOOnOffTimeData	生成设置 IO 信号定时开关命令的数据
23	CP5200_ParseWriteIOOnOffTimeRet	解析返回的 IO 信号定时开关命令的数据
24	CP5200_MakeReadIOOnOffTimeData	生成查询 IO 开关时间控制信号命令的数据
25	CP5200_ParseReadIOOnOffTimeRet	解析返回的查询 IO 开关定时控制命令的数据
26	CP5200_MakeWriteOnOffTimeData	生成设置定时开关屏命令的数据
27	CP5200_ParseWriteOnOffTimeRet	解析返回的设置定时开关屏命令的数据
28	CP5200_MakeReadOnOffTimeData	生成查询定时开关屏命令的数据
29	CP5200_ParseReadOnOffTimeRet	解析返回的查询定时开关屏命令的数据
30	CP5200_MakeReadVersionData	生成查询版本信息命令的数据

31	CP5200_ParseReadVersionRet	解析返回的查询版本信息命令的数据
32	CP5200_MakeFormatData	生成格式化控制卡文件系统命令的数据
33	CP5200_ParseFormatRet	解析返回的格式化控制卡文件系统命令的数据
34	CP5200_MakeRestartAppData	生成重启 App 命令的数据
35	CP5200_ParseRestartAppRet	解析返回的重启 App 命令的数据
36	CP5200_MakeRestartSysData	生成重启控制卡命令的数据
37	CP5200_ParseRestartSysRet	解析返回的重启控制卡命令的数据
38	CP5200_MakeGetFreeSpaceData	生成查询控制卡上可用空间命令的数据
39	CP5200_ParseGetFreeSpaceRet	解析返回的查询可用空间命令的数据
40	CP5200_MakeGetFileInfoData	生成取得文件信息命令的数据
41	CP5200_ParseGetFileInfoRet	解析返回的取得文件信息命令的数据
42	CP5200_ParseGetFirstFileInfoRet	解析返回的取得文件信息命令的数据, 并取得第一个文件的信息
43	CP5200_ParseGetNextFileInfoRet	解析返回的取得文件信息命令的数据, 取得下一个文件的信息
44	CP5200_MakeBeginFileUploadData	生成开始上载文件的数据
45	CP5200_ParseBeginFileUploadRet	解析返回的开始上载文件的数据
46	CP5200_MakeFileUploadData	生成上载文件的数据
47	CP5200_ParseFileUploadRet	解析返回的生成上载文件的数据
48	CP5200_MakeEndFileUploadData	生成结束上载文件的数据
49	CP5200_ParseEndFileUploadRet	解析返回的结束上载文件的数据
50	CP5200_MakeGetTypeInfoData	生成获得类型信息数据
51	CP5200_ParseGetTypeInfoRet	解析返回的获得类型信息的数据
52	CP5200_MakeGetTempHumiData	生成获得温度、湿度信息的数据
53	CP5200_ParseGetTempHumiRet	解析返回的获得温度、湿度信息的数据
54	CP5200_MakeReadConfigData	生成读取配置信息的数据
55	CP5200_ParseReadConfigRet	解析返回的读取配置信息的数据
56	CP5200_MakeWriteConfigData	生成写配置信息的数据
57	CP5200_ParseWriteConfigRet	解析返回的写配置信息的数据
58	CP5200_MakeReadRunningInfoData	生成查询控制卡运行信息的数据
59	CP5200_ParseReadRunningInfoRet	解析返回的查询控制卡运行信息的数据
60	CP5200_MakeScreenTestData	生成显示测试图案的数据
61	CP5200_ParseScreenTestRet	解析返回的显示测试图案的数据
62	CP5200_MakeInstantMessageData CP5200_MakeInstantMessageData1	生成即时信息数据
63	CP5200_MakeSendInstantMessageData	生成发送即时信息数据
64	CP5200_ParseSendInstantMessageRet	解析返回的发送即时信息的数据
65	CP5200_MakeReadHWSettingData	生成读取扫描参数的数据
66	CP5200_ParseReadHWSettingRet	解析返回的读取扫描参数的数据
67	CP5200_MakeWriteHWSettingData	生成写扫描参数的数据

68	CP5200_ParseWriteHWSettingRet	解析返回的写扫描参数的数据
69	CP5200_MakeReadSoftwareSwitchInfoData	生成查询软件开关屏信息命令的数据
70	CP5200_ParseReadSoftwareSwitchInfoRet	解析返回的查询软件开关屏信息命令的数据
71	CP5200_MakeWriteSoftwareSwitchInfoData	生成设置软件开关屏信息命令的数据
72	CP5200_ParseWriteSoftwareSwitchInfoRet	解析返回的设置软件开关屏信息命令的数据
73	CP5200_MakeQueryControllerInfo	生成查询控制卡信息的数据
74	CP5200_ParseQueryControllerInfoRet	解析查询控制卡信息命令的返回值
75	CP5200_MakeOpenFileData	生成打开控制卡上文件的数据
76	CP5200_ParseOpenFileRet	解析返回的打开控制卡文件的数据
77	CP5200_MakeGetDirentryData	生成获取控制卡上文件信息的数据
78	CP5200_ParseGetDirentryRet	解析返回的获取控制卡上文件信息的数据
79	CP5200_MakeReadFileNoData	生成读取控制卡上文件数据的数据
80	CP5200_ParseReadFileNoRet	解析返回的读取控制卡上文件数据的数据
81	CP5200_MakeCloseFileNoData	生成关闭控制卡上文件的数据
82	CP5200_ParseCloseFileNoRet	解析返回的关闭控制卡文件的数据

本类函数的使用步骤如下：

- 第1步：创建数据对象
- 第2步：生成通信所需的数据，包含了RS232/485的转码处理 (0xa5 => 0xaa 0x05, ...)，或者网络通信的识别码信息
- 第3步：将数据发送到控制卡
- 第4步：接收返回的数据，接收时进行转码处理 (0xaa 0x05 => 0xa5, ...)
- 第5步：解析返回的数据得到返回结果
- 第6步：销毁数据对象

4.2、数据通信 API 接口详细描述

CP5200_CommData_Create

HOBJECT CP5200_CommData_Create(int nCommType, BYTE byCardID, DWORD dwIDCode)	
作用	创建数据对象
参数	nCommType: RS232/485 或者网络通信方式 0: RS232/485 1: 网络

	byCardID: 控制卡 ID
	dwIDCode: 控制卡的网络识别码。RS232/485 通信方式时忽略此参数。
返回值	数据对象句柄，供本类 API 接口函数使用 返回“空”表示创建失败
其它说明	对象创建成功并不再使用时，必须将对象销毁

CP5200_CommData_Destroy

HOBJECT CP5200_CommData_Destroy (HOBJECT hObj)	
作用	创建数据对象
参数	hObj: 要销毁的数据对象句柄
返回值	0: 无错误 -1: 不正确的数据对象句柄
其它说明	

CP5200_CommData_SetParam

HOBJECT CP5200_CommData_SetParam (HOBJECT hObj, int nParamType, const char *pParamString)	
作用	设置数据包参数
参数	hObj: 数据对象句柄
	nParamType: 参数类型，目前只有 1 有效
	pParamString: 参数字符串 当参数类型为 1 时，pParamString 为控制卡的设备 ID，这个参数只有控制卡作为客户端通信或者控制卡板载 GPRS 通信时才需要设置
返回值	1: 无错误 0: 参数类型不对 -1: 不正确的数据对象句柄 -2: 参数字符串值不对
其它说明	

CP5200_MakeCreateFileData

<pre>int CP5200_MakeCreateFileData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, const char* pFilename, long lFileSize, const BYTE* pTimeBuffer)</pre>	
作用	生成创建文件所需的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pFilename: 要建立的文件名，必须是短文件名
	lFileSize: 要建立的文件的大小（字节）
	pTimeBuffer: 文件时间信息，长度 6 字节 字节 0: 年(00~99), 实际年份-2000 字节 1: 月(1~12) 字节 2: 日(1~31) 字节 3: 时 (0~23) 字节 4: 分(0~56) 字节 5: 秒(0~59)
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	若控制卡上存在同名的文件，旧文件将被覆盖 可支持的最大文件大小是 1.5M 字节 对每个控制卡，每次只能对 1 个文件进行操作

CP5200_ParseCreateFileRet

<pre>int CP5200_ParseCreateFileRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的创建文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区

	nLength: 返回数据的长度
返回值	1: 成功建立文件 0: 不能建立文件 -2: 返回的数据错误 -3: 返回的数据长度不足
其它说明	

CP5200_MakeWriteFileData

int CP5200_MakeWriteFileData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, const BYTE *pData, WORD wDatLen, WORD *pwChecksum)	
作用	生成写文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pData: 文件数据缓冲区
	wDataLen: 本次写入的数据长度
	pwChecksum: 指向 WORD 变量的指针，传入原有的数据校验和，并返回新的数据校验和
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	若文件较大，需要分多次写入数据，每次写入数据的不大于 1000 字节

CP5200_ParseWriteFileRet

int CP5200_ParseWriteFileRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的写文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度

返回值	1: 数据接收成功 0: 数据接收失败 -2: 返回的数据错误 -3: 返回的数据长度不足
其它说明	控制卡接收到数据后, 保存在临时缓冲区内, 直到关闭文件时才把数据写到文件中

CP5200_MakeCloseFileData

int CP5200_MakeCloseFileData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, WORD wChecksum)	
作用	生成关闭文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
	wChecksum: 文件数据的校验和
返回值	>0: 生成数据的长度 (字节) -1: 不正确的数据对象句柄
其它说明	关闭文件时, 需要将此文件的所有数据写入文件, 这个过程需要较长时间, 所需时间大约如下: $((\text{文件大小} / 4096) + 1) * 200 + 100$ 单位:毫秒

CP5200_ParseCloseFileRet

int CP5200_ParseCloseFileRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的关闭文件命令的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功关闭文件 0: 不能关闭文件

	255: 校验和错误 -2: 返回的数据错误 -3: 返回的数据长度不足
其它说明	

CP5200_MakeDeleteFileNoData

int CP5200_MakeDeleteFileNoData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, int fno)	
作用	生成根据文件号删除文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	fno: 文件号
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseDeleteFileNoRet

int CP5200_ParseDeleteFileNoRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的根据文件号删除文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功删除文件 0: 不能删除文件 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeDeleteFileNameData

int CP5200_MakeDeleteFileNameData (HOBJECT hObj, BYTE *pBuffer, int nBufSize, const char *pFilename)	
作用	生成根据文件名删除文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
	pFilename: 文件名
返回值	>0: 生成数据的长度 (字节) -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseDeleteFileNameRet

int CP5200_ParseDeleteFileNameRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的根据文件名删除文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功删除文件 0: 不能删除文件 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeReadTimeData

int CP5200_MakeReadTimeData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成查询控制卡上当前时间命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseReadTimeRet

int CP5200_ParseReadTimeRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pTimeBuffer, int nTimeBufSize)	
作用	解析返回的查询控制卡当前时间命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pTimeBuffer: 时间信息缓冲区，信息含义如下 0 字节: 秒 1 字节: 分 2 字节: 时 3 字节: 星期 4 字节: 日 5 字节: 月 6 字节: 年(2 位数字，加上 2000 是实际年份值)
	nTimeBufSize: 时间信息缓冲区的长度，要求不小于 7 字节
返回值	1: 成功获取时间信息

	0: 不能获取时间信息 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeWriteTimeData

<pre>int CP5200_MakeWriteTimeData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, const BYTE* pTimeBuffer)</pre>	
作用	生成设置时间命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pTimeBuffer: 时间信息数据缓冲区，信息数据含义如下 0 字节: 秒 1 字节: 分 2 字节: 时 3 字节: 星期 4 字节: 日 5 字节: 月 6 字节: 年(2 位数字，加上 2000 是实际年份值)
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseWriteTimeRet

<pre>int CP5200_ParseWriteTimeRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的设置控制卡上时间命令的数据

参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功设置时间 0: 不能设置时间 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeReadBrightnessData

int CP5200_MakeReadBrightnessData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成查询控亮度控制信息命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseReadBrightnessRet

int CP5200_ParseReadBrightnessRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pBrightnessBuffer, int nBrightBufSize)	
作用	解析返回的查询亮度控制信息命令的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度

	<p>pBrightnessBuffer: 亮度信息缓冲区，每小时用 1 个字节控制其亮度值，共 24 字节。每个字节的含义如下</p> <p>字节值 0~31: 亮度级别</p> <p>字节值 >31: 根据光敏传感器调节亮度</p>
	nBrightBufSize: 亮度信息缓冲区的长度,要求不小于 24 字节
返回值	<p>1: 成功获得亮度控制信息</p> <p>0: 不能获得亮度控制信息</p> <p>-2: 返回数据类型错误</p> <p>-3: 返回数据长度不足</p>
其它说明	

CP5200_MakeWriteBrightnessData

<pre>int CP5200_MakeWriteBrightnessData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, const BYTE *pBrightnessBuffer)</pre>	
作用	生成设置亮度控制信息命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	<p>pBrightnessBuffer: 亮度设置信息缓冲区，每小时用 1 个字节控制其亮度值，共 24 字节。每个字节的含义如下</p> <p>字节值 0~31: 亮度级别</p> <p>字节值 >31: 根据光敏传感器调节亮度</p>
返回值	<p>>0: 生成数据的长度（字节）</p> <p>-1: 不正确的数据对象句柄</p> <p>-4: 缓冲区长度不足</p>
其它说明	

CP5200_ParseWriteBrightnessRet

int CP5200_ParseWriteBrightnessRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的设置亮度控制信息命令的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功设置亮度控制信息 0: 不能设置亮度控制信息 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeWriteIOOnOffTimeData

int CP5200_MakeWriteIOOnOffTimeData (HOBJECT hObj, BYTE *pBuffer, int nBufSize, const BYTE *pOnOffBuffer)	
作用	生成设置 IO 信号定时开关命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pOnOffBuffer: 开关时间缓冲区，信息需要 4 个字节 0~1 字节：分别是“开”的时、分 2~3 字节：分别是“关”的时、分 当“开”、“关”的时间相同，表示一直“开”；若时>23 或分>59 则表示该时间无效
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	IO 信号从 J3 的 5、6 脚输出

CP5200_ParseWriteIOOnOffTimeRet

int CP5200_ParseWriteIOOnOffTimeRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的 IO 信号定时开关命令的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功设置 IO 开关时间 0: 不能设置 IO 开关时间 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeReadIOOnOffTimeData

int CP5200_MakeReadIOOnOffData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成查询 IO 开关时间控制信号命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseReadIOOnOffTimeRet

int CP5200_ParseReadIOOnOffTimeRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pOnOffBuffer, int nOnOffBufSize)	
作用	解析返回的查询 IO 开关定时控制命令的数据

参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pOnOffBuffer: IO 开关定时信息缓冲区，信息共 4 个字节 0~1 字节：分别是“开”的时、分 2~3 字节：分别是“关”的时、分 当“开”、“关”的时间相同，表示一直“开”；若时>23 或分>59 则表示该时间无效
	nOnOffBufSize: IO 开关定时信息缓冲区的长度，要求不小于 4 字节
返回值	1: 成功获得 IO 开关时间 0: 不能获得 IO 开关时间 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 信息缓冲区长度不足 -5: 校验和错误
其它说明	

CP5200_MakeWriteOnOffTimeData

<pre>int CP5200_MakeWriteOnOffTimeData (HOBJECT hObj, BYTE *pBuffer, int nBufSize, const BYTE *pOnOffBuffer)</pre>	
作用	生成设置定时开关屏命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）

	<p>pOnOffBuffer: 开关屏时间缓冲区，信息需要 6 个字节</p> <p>0~1 字节：分别是“开”的时、分</p> <p>2~3 字节：分别是“关”的时、分</p> <p>4~5 字节：保留，填 0</p> <p>当“开”、“关”的时间相同，表示一直“开”；若时>23 或分>59 则表示该时间无效</p>
返回值	<p>>0: 生成数据的长度（字节）</p> <p>-1: 不正确的数据对象句柄</p> <p>-4: 缓冲区长度不足</p>
其它说明	

CP5200_ParseWriteOnOffTimeRet

int CP5200_ParseWriteOnOffTimeRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的设置定时开关屏命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	<p>1: 成功设置定时开关屏</p> <p>0: 不能设置定时开关屏</p> <p>-2: 返回数据类型错误</p> <p>-3: 返回数据长度不足</p>
其它说明	

CP5200_MakeReadOnOffTimeData

int CP5200_MakeReadOnOffTimeData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成查询定时开关屏命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区

	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseReadOnOffTimeRet

<pre>int CP5200_ParseReadOnOffTimeRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pOnOffBuffer, int nOnOffBufSize)</pre>	
作用	解析返回的查询定时开关屏命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	<p>pOnOffBuffer: 定时开关信息缓冲区，信息共 6 个字节</p> <p>0~1 字节：分别是“开”的时、分</p> <p>2~3 字节：分别是“关”的时、分</p> <p>4~5 字节：保留，默认为 0</p> <p>当“开”、“关”的时间相同，表示一直“开”；若时>23 或分>59 则表示该时间无效</p>
	nOnOffBufSize: 定时开关屏信息缓冲区的长度，要求不小于 6 字节
返回值	<p>1: 成功获得定时开关屏信息</p> <p>0: 不能获得定时开关屏信息</p> <p>-2: 返回数据类型错误</p> <p>-3: 返回数据长度不足</p> <p>-4: 信息缓冲区长度不足</p> <p>-5: 校验和错误</p>
其它说明	

CP5200_MakeReadVersionData

int CP5200_MakeReadVersionData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成查询版本信息命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
返回值	>0: 生成数据的长度 (字节) -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseReadVersionRet

int CP5200_ParseReadVersionRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)	
作用	解析返回的查询版本信息命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 版本信息缓冲区, 版本信息包括 3 字节, 每个字节的高 4 位是主版本, 低 4 位是次版本(0x10 = V1.0) 0 字节: Bios 版本 1 字节: 逻辑版本 2 字节: 软件版本
	nInfoBufSize: 版本信息缓冲区长度, 要求不小于 3 字节
返回值	1: 成功获得版本信息 0: 不能获得版本 -2: 返回数据类型错误 -3: 返回数据长度不足

	-4: 版本信息缓冲区长度不足
其它说明	

CP5200_ParseReadVersionRet2

<pre>int CP5200_ParseReadVersionRet2(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的查询版本信息命令的数据,包括卡型号,每个版本信息用 2 字节表示
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 版本信息缓冲区,返回值定义如下: 0 字节: 有效数据长度,包括本字节 1 字节: 控制卡类型 2~3 字节: BIOS 版本 4~5 字节: 逻辑版本 6~7 字节: APP(程序)版本 BIOS、逻辑、APP 版本信息用 2 字节表示。
	nInfoBufSize: 版本信息缓冲区长度
返回值	1: 成功获得版本信息 0: 不能获得版本 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 版本信息缓冲区长度不足
其它说明	

CP5200_MakeFormatData

<pre>int CP5200_MakeFormatData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)</pre>

作用	生成格式化控制卡文件系统命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	格式化将删除控制卡的所有文件！ 由于格式化过程需要一定的时间，格式化命令数据往控制卡发送之后，需要等待大约 1 秒钟，然后才能接收到返回信息。

CP5200_ParseFormatRet

<code>int CP5200_ParseFormatRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</code>	
作用	解析返回的格式化控制卡文件系统命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功格式化 0: 不能格式化 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeRestartAppData

<code>int CP5200_MakeRestartAppData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)</code>	
作用	生成重启 App 命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区

	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseRestartAppRet

int CP5200_ParseRestartAppRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的重启 App 命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功重启 App 0: 不能重启 App -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeRestartSysData

int CP5200_MakeRestartSysData (HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成重启控制卡命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseRestartSysRet

int CP5200_ParseRestartSysRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的重启控制卡命令的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功重启控制卡 0: 不能重启控制卡 -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeGetFreeSpaceData

int CP5200_MakeGetFreeSpaceData (HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成查询控制卡上可用空间命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseGetFreeSpaceRet

int CP5200_ParseGetFreeSpaceRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的查询可用空间命令的数据
参数	hObj:数据对象句柄

	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	>=0: 可用空间的大小（字节） -2: 返回数据类型错误 -3: 返回数据长度不足
其它说明	

CP5200_MakeGetFileInfoData

int CP5200_MakeGetFileInfoData(HOBJECT hObj, BYTE* pBuffer, int nLength, int nNewSearch)	
作用	生成取得文件信息命令的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	nNewSearch: 1 从头开始查找；0 继续上次的查找
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	此命令取得控制卡上最多 4 个文件的信息，接收返回数据的缓冲区长度不小于 300 字节 可通过多次调用本函数，并给参数 nNewSearch 赋合适的值来获得所有文件的信息。

CP5200_ParseGetFileInfoRet

int CP5200_ParseGetFileInfoRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, int pos, BYTE* pInfoBuffer, int nInfoBufSize)	
作用	解析返回的取得文件信息命令的数据，取得下一个文件的信息
参数	hObj:数据对象句柄

	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pos: 当前文件序号
	<p>pInfoBuffer: 文件信息缓冲区</p> <p>字节 含义</p> <p>0~44: 文件名及扩展名, 二者之间含分割点'.'. 例如"a1.txt"</p> <p>45~45: 年高两位。例如: 2009 年, 本字节的值是 20; 1999 年, 本字节的值是 19</p> <p>46~46: 年低两位。例如: 2009 年, 本字节的值是 9; 1999 年, 本字节的值是 99</p> <p>47~47: 月</p> <p>48~48: 日</p> <p>49~49: 时</p> <p>50~50: 分</p> <p>51~51: 秒</p> <p>52~55: 文件大小, 低字节在前</p>
	nInfoBufSize: 文件信息缓冲区的长度, 应不小于 64
返回值	<p>>0: pBuffer 中存在的文件信息数量</p> <p>0: 没有所需的信息</p> <p>-1: 参数 hObj 无效</p> <p>-2: 返回数据类型错误</p> <p>-3: 返回数据长度不足</p> <p>-4: 信息缓冲区长度不足</p>
其它说明	<p>文件序号从 0 开始, 从返回值可得知本次获取的文件数量。</p> <p>若返回值小于 4, 则说明已经获取了卡上的所有文件信息。</p> <p>若返回值等于 4, 可再次调用 CP5200_MakeGetFileInfoData()来获取更多的文件信息。</p>

CP5200_ParseGetFirstFileInfoRet

int CP5200_ParseGetFirstFileInfoRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)	
作用	解析返回的取得文件信息命令的数据，并取得第一个文件的信息
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 文件信息缓冲区 0~11 字节: 文件名 12~15: 文件扩展名 16~18: 文件日期的年、月、日，每个字节一个值。其中年份的值范围是 0~99,加上 2000 得到实际值 19~21: 文件时间的时、分、秒，每个字节一个值。 22~25: 文件大小，低字节在前 26~31: 保留
	nInfoBufSize: 文件信息缓冲区的长度，应不小于 32
返回值	1: 成功获得需要的信息 0: 没有所需的信息，即控制卡上没有任何文件 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 信息缓冲区长度不足
其它说明	与函数 CP5200_ParseGetNextFileInfoRet 配合使用

CP5200_ParseGetNextFileInfoRet

int CP5200_ParseGetNextFileInfoRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, int pos, BYTE* pInfoBuffer, int nInfoBufSize)	
作用	解析返回的取得文件信息命令的数据，取得下一个文件的信息
参数	hObj:数据对象句柄

	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pos: 当前文件序号
	pInfoBuffer: 文件信息缓冲区 0~11 字节: 文件名 12~15: 文件扩展名 16~18: 文件日期的年、月、日, 每个字节一个值。其中年份的值范围是 0~99,加上 2000 得到实际值 19~21: 文件时间的时、分、秒, 每个字节一个值 22~25: 文件大小, 低字节在前 26~31: 保留
	nInfoBufSize: 文件信息缓冲区的长度, 应不小于 32
返回值	1: 成功获得需要的信息 0: 没有所需的信息 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 信息缓冲区长度不足
其它说明	文件序号从 0 开始, 当 pos=0 时, 取下一个文件就是第二个文件的信息 与函数 CP5200_ParseGetFirstFileInfoRet 配合使用 若返回值小于 4, 则说明已经获取了卡上的所有文件信息。 若返回值等于 4, 可再次调用 CP5200_MakeGetFileInfoData()来获取更多的文件信息。

CP5200_MakeBeginFileUploadData

int CP5200_MakeBeginFileUploadData(HOJECT hObj, BYTE *pBuffer, int nBufSize, const char* pFilename, long lFilesize, const BYTE* pTimeBuffer)	
作用	生成开始上传文件所需的数据, 通知控制卡做好接收数据的准备
参数	hObj: 数据对象句柄

	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pFilename: 要建立的文件名，必须是短文件名
	lFileSize: 要建立的文件的大小（字节）
	pTimeBuffer: 文件时间信息，长度 6 字节 字节 0: 年(00~99), 实际年份-2000 字节 1: 月(1~12) 字节 2: 日(1~31) 字节 3: 时 (0~23) 字节 4: 分(0~56) 字节 5: 秒(0~59)
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	若控制卡上存在同名的文件，旧文件将被覆盖 可支持的最大文件大小是 1.5M 字节 对每个控制卡，每次只能对 1 个文件进行操作

CP5200_ParseBeginFileUploadRet

int CP5200_ParseBeginFileUploadRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的开始上传文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功建立文件，可开始上传 0: 不能建立文件 -2: 返回的数据错误 -3: 返回的数据长度不足
其它说明	

CP5200_MakeFileUploadData

<pre>int CP5200_MakeFileUploadData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, const BYTE *pData, WORD wDatLen, WORD wSegNo, WORD wSegLen, int nWantRet)</pre>	
作用	生成上传文件的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pData: 本次上传的文件数据缓冲区
	wDataLen: 本次上传的实际数据长度。不能大于 wSegLen。
	wSegNo: 数据段编号，从 0 开始
	wSegLen: 数据段的长度，要求不大于 1024，一般设置为 512 即可。对于每次上传，此参数需保持一致。
返回值	nWantRet: 是否要立即返回确认信息。 0 不要返回 1 要返回
	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	若文件较大，需要分多次上传数据。 两次上传之间的时间间隔需不少于 50 毫秒。 必须根据 nWantRet 的值来确定是否处理返回信息。

CP5200_ParseFileUploadRet

<pre>int CP5200_ParseFileUploadRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的上传文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 数据接收成功

	0: 数据接收失败 -2: 返回的数据错误 -3: 返回的数据长度不足
其它说明	只有传递给 CP5200_MakeWriteFileData 函数的 nWantRet 参数非 0 时，才需要利用本函数。

CP5200_MakeEndFileUploadData

<pre>int CP5200_MakeEndFileUploadData(HOBJECT hObj, BYTE *pBuffer, int nBufSize, WORD wTotalSeg)</pre>	
作用	生成结束上传文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	wTotalSeg: 总的的数据段数量
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseEndFileUploadRet

<pre>int CP5200_ParseEndFileUploadRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的结束上传文件命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 上传结果信息缓冲区，记录不成功的数据段。每 2 字节代表 1 个数据段的段号。低字节在前。
	nInfoBufSize: 上传结果信息缓冲区的长度，要求 72 字节以上

返回值	<p>0: 成功上传了文件</p> <p>>0 且不大于 36: 错误的数数据段数量</p> <p>255: 没有文件要关闭</p> <p>-2: 返回的数据错误</p> <p>-3: 返回的数据长度不足</p>
其它说明	返回的错误数据段数量不表示全部的错误数据段, 可重发本次已知的错误数据段, 然后获得新的结果信息, 直到没有错误为止。

CP5200_MakeGetTypeInfoData

int CP5200_MakeGetTypeInfoData(HOBJECT hObj, BYTE* pBuffer, int nBufSize)	
作用	生成获得类型信息数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
返回值	<p>>0: 生成数据的长度 (字节)</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseGetTypeInfoRet

int CP5200_ParseGetTypeInfoRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)	
作用	解析返回的获得类型信息的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度

	<p>pInfoBuffer: 类型信息缓冲区（10 字节）</p> <p>字节 0: 控制卡类型</p> <p>字节 1: FPGA 版本</p> <p>字节 2-5: BIOS 版本</p> <p>字节 6-9: APP 版本</p>
	nInfoBufSize: 获得类型信息缓冲区的长度，要求 10 字节以上
返回值	<p>0: 成功</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 返回的数据错误</p> <p>-3: 返回的数据长度不足</p> <p>-5: 校验和错误</p>
其它说明	

CP5200_MakeGetTempHumiData

<pre>int CP5200_MakeGetTempHumiData(HOBJECT hObj, BYTE* pBuffer, int nBufSize , byte byFlag)</pre>	
作用	生成获得温度信息的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	byFlag: 查询标记 Bit0: 是否要查询温度(0 不要,1 要) Bit1: 是否要查询湿度(0 不要,1 要)
返回值	<p>>0: 生成数据的长度（字节）</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseGetTempHumiRet

<pre>int CP5200_ParseGetTempHumiRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
--	--

作用	解析返回的获得温度信息的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 温度、湿度信息缓冲区, 长度为 8 字节, 信息含义如下: byte 0: 查询标记, 含义同发送包 byte 1~2: 温度值(摄氏度): 第 1 字节: Bit7: 数值符号。1 负值, 0 正值。 Bit6~0: 温度绝对值整数部分的高 7 位 第 2 字节: Bit7~4: 温度绝对值整数部分的低 4 位 Bit3~0: 小数部分, 单位是 1/16(即 0.0625) byte 3~4: 温度值(华氏度): byte 5: 温度调整值, Bit7: 1 华氏度, 0 摄氏度 Bit6: 1 负值, 0 正值 Bit5~0: 温度调整的绝对值 byte 6: 湿度值。有效值 0~100 byte 7: 湿度调整值 Bit7: 保留 Bit6: 1 负值, 0 正值 Bit5~0: 湿度调整的绝对值
	nInfoBufSize: 温度信息缓冲区的长度
返回值	-1
其它说明	此函数功能暂未实现

CP5200_MakeReadConfigData

int CP5200_MakeReadConfigData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, int nFlag)	
作用	生成读取配置信息的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
	nFlag: 要读取的配置信息的标记

返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseReadConfigRet

<pre>int CP5200_ParseReadConfigRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的读取配置信息的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 配置信息缓冲区
	nInfoBufSize: 配置信息缓冲区大小，要求 6 字节以上
返回值	6: 成功 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -5: 校验和错误
其它说明	

CP5200_MakeWriteConfigData

<pre>int CP5200_MakeWriteConfigData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, const BYTE* pConfig, int nCfgLength)</pre>	
作用	生成写配置信息的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pConfig: 配置信息内容指针

	nCfgLength: 配置信息长度
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseWriteConfigRet

int CP5200_ParseWriteConfigRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的写配置信息的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功 0: 失败 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -5: 校验和错误
其它说明	

CP5200_MakeReadRunningInfoData

int CP5200_MakeReadRunningInfoData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, int nFlag)	
作用	生成读取运行信息的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）

	<p>nFlag: 要读取的运行信息的标记</p> <p>0: (无效)</p> <p>1: 当前播放节目号</p> <p>2: 读当前字库信息</p> <p>其它: 保留</p>
返回值	<p>>0: 生成数据的长度 (字节)</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseReadRunningInfoRet

<pre>int CP5200_ParseReadRunningInfoRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的读取运行信息的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 运行信息缓冲区
	<p>byte 0: 确认信息, 0 失败; >0 成功</p> <p>如果查询标记为 1: 当前播放节目号, 则 Byte1~Byte5 如下:</p> <p>第 1 字节: 播放类型</p> <p>第 2~3 字节: 节目总数。高字节在前</p> <p>第 4~5 字节: 节目号。高字节在前</p> <p>如果查询标记为 2: 读当前字库信息, 则 Byte1~Byte6 如下:</p> <p>第 1 字节: 字库类型</p> <p>第 2 字节: 保留</p> <p>第 3~4 字节: ASCII 字库可用大小。高字节在前</p> <p>第 5~6 字节: 扩展字库可用大小。高字节在前</p>
	nInfoBufSize: 运行信息缓冲区大小, 要求 7 字节以上
返回值	<p>>=6: 成功</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 返回的数据错误</p>

	-3: 返回的数据长度不足 -5: 校验和错误
其它说明	

CP5200_MakeScreenTestData

<pre>int CP5200_MakeScreenTestData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, BYTE* pInfoBuffer, int nInfLength)</pre>		
作用	生成显示测试图案的数据	
参数	hObj:数据对象句柄	
	pBuffer: 结果数据缓冲区	
	nBufSize: 结果数据缓冲区的大小（字节）	
	pInfoBuffer: 测试图案信息内容指针，详细如下： byte 0: 选项: Bit7: 0 立即取消测试, 1 立即进入测试。 Bit6: 是否自动返回。1 自动返回, 0 不自动返回 Bit0~5: 自动返回前测试的次数。 取消测试时, 后面的参数无效。 byte 1~2: 屏宽: 指定屏的宽度。高字节在前 0: 缺省, >0 屏宽度 byte 3~4: 屏高: 指定屏的高度。高字节在前 0: 缺省, >0 屏高度 byte 5: 测试颜色: Bit0~2: 基本颜色 Bit3: 是否组合基本颜色 Bit4~7: 保留, 填 0。 byte 6: 测试灰度: Bit0~3: 要测试的灰度级别 Bit4~7: 保留, 填 0。 byte 7: 测试图案: 0: 整屏 1: 单斜线左移 2: 斜网格线左移 其它: 保留, 填 0。 byte 8~9: 切换时间: 高字节在前。单位 10 毫秒。 0 为缺省时间（整屏 3 秒, 移动 20 毫秒）	
	nInfLength: 测试图案信息长度, 目前为 10 字节	
	返回值	>0: 生成数据的长度（字节）

	-1: 不正确的数据对象句柄
其它说明	

CP5200_ParseScreenTestRet

int CP5200_ParseScreenTestRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的显示测试图案的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1 成功 0 失败 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -5: 校验和错误
其它说明	

CP5200_MakeInstantMessageData

int CP5200_MakeInstantMessageData(BYTE* pBuffer, int nBufSize, byte byPlayTimes , int x , int y , int cx , int cy , byte byFontSizeColor , int nEffect , byte nSpeed , byte byStayTime , const char* pText);	
作用	生成即时信息数据
参数	pBuffer:结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	byPlayTimes: 播放次数。0~255。数值 0 表示一直播放，直到有新的指令。
	x:显示起点 x，显示区域的左上角横坐标。
	y:显示起点 y，显示区域的左上角纵坐标。

	Cx:显示宽度，显示区域的宽度。等于 0 含义为屏最大宽度。
	Cy:显示高度，显示区域的高度。等于 0 含义为屏最大高度。
	byFontSizeColor:字体颜色。文字显示的字体大小。 Bit0~3: 字体大小 Bit4: 颜色的红色分量 Bit5: 颜色的绿色分量 Bit6: 颜色的蓝色分量 Bit7: 保留
	nEffect:显示效果
	nSpeed:速度。0~255。数值越小，速度越快。对立即显示的方式无效。
	byStayTime:停留时间。高字节在前，单位是秒。显示每页之间的停留时间。对滚动方式无效。
	pText:文本数据。
返回值	>0: 生成数据的长度 <=0: 结果数据缓冲区不足
其它说明	此函数只是将即时信息打包，并不进行发送。需要配合函数 CP5200_MakeSendInstantMessageData 和 CP5200_ParseSendInstantMessageRet 使用。

CP5200_MakeInstantMessageData1

<pre>int MakeInstantMessageData1(BYTE* pBuffer, int nBufSize, BYTE byPlayTimes, int x, int y, int cx, int cy, int nFontSize, byte byColorAlign, int nEffect, BYTE nSpeed, BYTE byStayTime, const char* pText);</pre>	
作用	生成即时信息数据
参数	pBuffer:结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	byPlayTimes: 播放次数。0~255。数值 0 表示一直播放，直到有新的指令。
	x:显示起点 x，显示区域的左上角横坐标。
	y:显示起点 y，显示区域的左上角纵坐标。
	Cx:显示宽度，显示区域的宽度。等于 0 含义为屏最大宽度。

	Cy:显示高度，显示区域的高度。等于 0 含义为屏最大高度。
	nFontSize: 文字大小及类型，参见 1.7、文字大小及类型编码
	byColorAlign: 颜色及对齐方式 Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~5: 水平对齐。0 左，1 中，2 右 Bit6~7: 垂直对齐。0 上，1 中，2 下
	nEffect:显示效果
	nSpeed:速度。0~255。数值越小，速度越快。对立即显示的方式无效。
	byStayTime:停留时间。高字节在前，单位是秒。显示每页之间的停留时间。对滚动方式无效。
	pText:文本数据。
返回值	>0: 生成数据的长度 <=0: 结果数据缓冲区不足
其它说明	此函数只是将即时信息打包，并不进行发送。需要配合函数 CP5200_MakeSendInstantMessageData 和 CP5200_ParseSendInstantMessageRet 使用。

CP5200_MakeSendInstantMessageData

<pre>int CP5200_MakeSendInstantMessageData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, const BYTE* pData, int nDataLen, byte byLastPacket, long lDataOffset);</pre>	
作用	生成发送即时信息数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pData: 由函数 CP5200_MakeInstantMessageDat 生成的即时信息数据
	nDataLen: 数据长度。低字节在前
	byLastPacket: 是否最后 1 包数据，1 是，0 否
	lDataOffset: 数据的偏移量,低字节在前
返回值	>0: 生成数据的长度（字节）

	-1: 不正确的数据对象句柄
其它说明	本函数将 CP5200_MakeInstantMessageData 制作出来的数据封包进行发送，每个包的最大长度不超过 1024 字节，建议 200 字节。

CP5200_ParseSendInstantMessageRet

<pre>int CP5200_ParseSendInstantMessageRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize);</pre>	
作用	解析返回的发送即时消息命令数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer:返回结果缓冲区
	nInfoBufSize: 5 个字节。 第 1 个字节: 0x00: 失败 非零: 成功 后面 4 个字节: 低字节在前 成功时返回数据偏移。 失败时第一字节返回错误码，另外 3 字节保留 1: 内部缓冲区不足 其它: 保留 非零: 错误码
返回值	> 0:成功 0: 失败 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -5: 校验和错误
其它说明	

CP5200_MakeReadHWSettingData

<pre>int CP5200_MakeReadHwSettingData(HOBJECT hObj, BYTE* pBuffer, int nBufSize)</pre>	
作用	生成读取配置信息的数据

参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseReadHWSettingRet

<pre>int CP5200_ParseReadHWSettingRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize , int nPassword)</pre>	
作用	解析返回的读取配置信息的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 扫描信息缓冲区, 不少于 16 个字节, 各字节含义参见 1.14 、 扫描参数各字节含义
	nInfoBufSize: 扫描信息缓冲区大小, 要求 16 字节以上
	nPassword: 解析密码, 根据不同控制卡填入不同密码, 否则不予受理
返回值	16: 成功 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -5: 校验和错误
其它说明	

CP5200_MakeWriteHWSettingData

<pre>int CP5200_MakeWriteHWSettingData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, const BYTE* pSetting, int nPassword)</pre>	
--	--

作用	生成写配置信息的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	pSetting: 扫描信息内容指针，长度为 16 字节，各字节含义参见 1.14 、 扫描参数各字节含义
	nPassword: 解析密码，根据不同控制卡填入不同密码，否则不予受理
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseWriteHWSettingRet

int CP5200_ParseWriteConfigRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的写配置信息的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	>0: 成功 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -5: 校验和错误
其它说明	

CP5200_MakeReadSoftwareSwitchInfoData

int CP5200_MakeReadSoftwareSwitchInfoData(HOBJECT hObj, BYTE *pBuffer, int nBufSize)	
作用	生成查询软件开关屏信息命令的数据
参数	hObj: 数据对象句柄

	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
返回值	>0: 生成数据的长度 (字节) -1: 不正确的数据对象句柄 -4: 缓冲区长度不足
其它说明	

CP5200_ParseReadSoftwareSwitchInfoRet

int CP5200_ParseReadSoftwareSwitchInfoRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)		
作用	解析返回的查询软件开关屏信息命令的数据	
参数	hObj:数据对象句柄	
	pBuffer: 返回数据缓冲区	
	nLength: 返回数据的长度	
	pInfoBuffer: 软件开关屏信息缓冲区, 软件开关屏信息包括 9 字节	
	数据项	长度
开关状态	1	0: 关 1: 开
值	8	1~2 字节: 开机时、分 3~4 字节: 关机时、分 5~8 字节: 保留
	nInfoBufSize: 软件开关屏信息缓冲区长度,要求不小于 9 字节	
返回值	1: 成功获得软件开关屏信息 0: 不能获得软件开关屏信息 -1: 不正确的数据对象句柄 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 软件开关屏信息缓冲区长度不足	

	-5: 校验和错误
其它说明	

CP5200_MakeWriteSoftwareSwitchInfoData

<pre>int CP5200_MakeWriteSoftwareSwitchInfoData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, const BYTE *pSoftwareSwitchInfoBuf)</pre>										
作用	生成设置软件开关屏信息命令的数据									
参数	hObj: 数据对象句柄									
	pBuffer: 结果数据缓冲区									
	nBufSize: 结果数据缓冲区的大小（字节）									
	pSoftwareSwitchInfoBuf: 软件开关屏信息									
	<table border="1"> <thead> <tr> <th>数据项</th> <th>长度</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>开关命令</td> <td>1</td> <td>0: 立即关屏 1: 立即开屏</td> </tr> <tr> <td>值</td> <td>8</td> <td>默认为 0</td> </tr> </tbody> </table>	数据项	长度	说明	开关命令	1	0: 立即关屏 1: 立即开屏	值	8	默认为 0
数据项	长度	说明								
开关命令	1	0: 立即关屏 1: 立即开屏								
值	8	默认为 0								
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄 -4: 缓冲区长度不足									
其它说明										

CP5200_ParseWriteSoftwareSwitchInfoRet

<pre>int CP5200_ParseWriteSoftwareSwitchInfoRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的设置软件开关屏信息命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度

	pInfoBuffer: 软件开关屏信息缓冲区, 软件开关屏信息包括 1 字节以上		
	数据项	长度	说明
	状态	>=1	第一字节: 0 当前关屏, 1 当前开屏 其它: 忽略
	nInfoBufSize: 软件开关屏信息缓冲区长度, 要求不小于 1 字节		
返回值	1: 成功设置软件开关屏信息 0: 不能设置软件开关屏信息 -1: 不正确的数据对象句柄 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 软件开关屏信息缓冲区长度不足 -5: 校验和错误		
其它说明			

CP5200_MakeQueryControllerInfo

<pre>int CP5200_MakeQueryControllerInfo(HOBJECT hObj, BYTE* pBuffer, int nBufSize, byte byInfoFlag, const BYTE *pAppendBuf, int nAppendLen)</pre>	
作用	生成查询控制卡信息命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
	byInfoFlag: 查询标记, 目前只支持 0x0b 0x00: (无效) 0x01: 当前播放节目号 0x02: 读当前字库信息 0x0a: 查询节目播放中的检查结果 0x0b: 获取当前显示屏在播状态及数据 其它: 保留

	<p>pAppendBuf:</p> <p>所需的附加值根据“信息标识”不同而不同:</p> <p>信息标识 0x01: 无附加值</p> <p>信息标识 0x02: 无附加值</p> <p>信息标识 0x0a: 1个字节的附加值, 现在支持的有效值是 0x31</p> <p>信息标识 0x0b: >=1个字节的附加值, 要读取的信息和参数</p> <p> 第一字节=0 返回截图数据标识, 节目号, 节目已播放时长。</p> <p> 第一字节=1 返回截图数据标识, 节目号, 节目已播放时长, 此外还返回在播画面图片的数据。第二字节乘以 8 为期望的每包数据包含的图片字节数, 0 表示由控制卡确定。</p> <p> 第一字节=2 返回实际图像数据。第二、三字节为图像数据包序号, 高字节在前。</p>
	nAppendLen: 附加值数据长度
返回值	<p>>0: 生成数据的长度 (字节)</p> <p>-1: 不正确的数据对象句柄</p> <p>-4: 缓冲区长度不足</p>
其它说明	

CP5200_ParseQueryControllerInfoRet

<pre>int CP5200_ParseQueryControllerInfoRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, byte byInfoFlag, byte byAppendFlag, byte *pInfoBuf, int nInfoBufLen)</pre>	
作用	解析返回的查询控制卡信息命令的数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	byInfoFlag: 查询标记, 目前只支持 0x0b
	0x00: (无效)
	0x01: 当前播放节目号
	0x02: 读当前字库信息
	0x0a: 查询节目播放中的检查结果
0x0b: 获取当前显示屏在播状态及数据	
其它: 保留	
byAppendFlag: 当 byInfoFlag 为 0x0b 是, 取值 0x01	
pInfoBuf: 见下文 结果缓存说明	
nInfoBufLen: 结果缓存长度	

返回值	>0: 结果缓存有效数据长度 0: 附加标记无效 -1: 不正确的数据对象句柄 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 软件开关屏信息缓冲区长度不足 -5: 校验和错误 -6: 返回数据错误 -7: 查询标记不支持
其它说明	

结果缓存说明

信息标识	返回的信息值长度及含义
0x01: 当前播放节目号	5 字节。 第 0 字节: 播放类型。 Bit0~3: 0 一般节目, bit4: 0 第一套节目, 1 第二套节目 Bit5~7: 保留 第 1~2 字节: 节目总数。高字节在前 第 3~4 字节: 节目号。高字节在前
0x02: 读当前字库信息	6 字节。 第 0 字节: 字库类型 第 1 字节: 保留 第 2~3 字节: ASCII 字库可用大小。高字节在前 第 4~5 字节: 扩展字库可用大小。高字节在前
0x0a: 查询节目播放中的检查结果。 信息标识之后的附加值 (1 个字节) 还要设置为 0x31	变长, 不超过 150 字节。 第 0 字节: 附加值 (同发送值) 第 1~2 字节: 节目播放表里的节目总数。高字节在前 第 3~4 字节: 已检查的节目数量.有播放计划时, 该值可能有误。高字节在前 第 5~6 字节: 节目错误信息的数据长度。高字节在前 第 7~ 字节: 节目错误信息的数据。按节目顺序, 每 8 个节目的信息用 1 个字节表示 (每个节目 1 位), 0 标识未发现错误 (也可能还没检查到), 1 标识有错。
0x0b: 获取当前显示屏在播状态。 返回值根据信息标识之后的附加值确定	附加值 0 和 1: 附加值 0 时返回 1~18 字节的信息; 附加值为 1 时, 还返回 19 字节及之后的数据: 第 0~5 字节: 当前的截图数据标识 第 6~7 字节: 节目号。高字节在前, 第一个节目从 1 开始, 0 表示没有节目在播放或者播放临时信息

	<p>第 8~9 字节：播放项号。</p> <p>第 10~13 字节：节目已播放时长，单位为 1/10 秒，高字节在前</p> <p>第 14~17 字节：播放项已播放时长，单位为 1/10 秒，高字节在前</p> <p>以下</p> <p>第 18~19 字节：图像宽度，高字节在前</p> <p>第 20~21 字节：图像高度，高字节在前</p> <p>第 22 字节：颜色及灰度</p> <p>第 23 字节：乘以 8 为一个数据包的图像数据长度</p> <p>第 24~27 字节：图像数据长度，高字节在前</p> <p>附加值 2:</p> <p>第 0~5 字节：当前的截图数据标识</p> <p>第 6~7 字节：图像数据包总数，高字节在前</p> <p>第 8~9 字节：图像数据包序号，高字节在前</p> <p>第 10~13 字节：图像数据偏移，高字节在前</p> <p>第 14~15 字节：图像数据长度，高字节在前</p> <p>第 16~ 字节：图像数据</p>
--	--

CP5200_MakeOpenFileData

<pre>int CP5200_MakeOpenFileData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, const char* chFileName)</pre>	
作用	生成打开控制卡文件的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	chFileName: 要打开的文件名称，如果在系统盘，名称前需加上“S:”
返回值	<p>>0: 生成数据的长度（字节）</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseOpenFileRet

<pre>int CP5200_ParseOpenFileRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的打开控制卡文件的数据

参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 配置信息缓冲区 字节 0~1: 表示文件信息编号, 低字节在前
	nInfoBufSize: 配置信息缓冲区大小, 要求 2 字节以上
返回值	<p>>=0: 成功 , 表示打开的文件号</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 返回的数据错误</p> <p>-3: 返回的数据长度不足</p> <p>-4: 接收缓冲区长度不足</p> <p>-5: 校验和错误</p>
其它说明	

CP5200_MakeGetDirentryData

<pre>int CP5200_MakeGetDirentryData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, WORD dno, int nPath)</pre>	
作用	生成获取文件信息的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小 (字节)
	dno: 要获取的文件信息的编号, 由 CP5200_ParseOpenFileRet 获得
	nPath: 路径信息, 用户盘为 1, 系统盘为 0
返回值	<p>>0: 生成数据的长度 (字节)</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseGetDirentryRet

<pre>int CP5200_ParseGetDirentryRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的获取文件信息的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
参数	nLength: 返回数据的长度
	pInfoBuffer: 配置信息缓冲区
	字节 0~31: 文件名称
	字节 32~43: 文件扩展名
	字节 44~45: 文件属性
	字节 46~47: 文件校验和
	字节 48~49: 保留
	字节 50~53: 文件生成时间
	字节 54~57: 文件生成日期
	字节 58~59: 意义未知
字节 60~63: 文件大小	
	各数据均为低字节在前
	nInfoBufSize: 配置信息缓冲区大小, 要求 64 字节以上
返回值	>=0: 文件信息的编号 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -4: 接收缓冲区长度不足 -5: 校验和错误
其它说明	

CP5200_MakeReadFileNoData

<pre>int CP5200_MakeReadFileNoData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, WORD wdCount, byte fno)</pre>	
作用	生成读取文件数据的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	wdCount: 要读取的数据长度
	fno: 读取的文件号，由 CP5200_ParseOpenFileRet 的返回值获取
返回值	<p>>0: 生成数据的长度（字节）</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseReadFileNoRet

<pre>int CP5200_ParseReadFileNoRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的读取文件数据的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer: 配置信息缓冲区，存储读取的数据
	nInfoBufSize: 配置信息缓冲区大小，一般要求 512 字节以上
返回值	<p>>=0: 读取数据的长度</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 返回的数据错误</p> <p>-3: 返回的数据长度不足</p> <p>-4: 接收缓冲区长度不足</p> <p>-5: 校验和错误</p>

其它说明	
------	--

CP5200_MakeCloseFileNoData

int CP5200_MakeCloseFileNoData(HOBJECT hObj, BYTE* pBuffer, int nBufSize, byte fno)	
作用	生成关闭文件的数据
参数	hObj:数据对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	fno: 读取的文件号，由 CP5200_ParseOpenFileRet 的返回值获取
返回值	>0: 生成数据的长度（字节） -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseCloseFileNoRet

int CP5200_ParseCloseFileNoRet(HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE*)	
作用	解析返回的关闭文件的数据
参数	hObj:数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	1: 成功 -1: 不正确的数据对象句柄 -2: 返回的数据错误 -3: 返回的数据长度不足 -4: 接收缓冲区长度不足 -5: 校验和错误
其它说明	

5、多窗口数据通信 API 接口

5.1、数据通信 API 接口总览

序号	API 接口函数名	接口说明
1	CP5200_CmmPacker_Create	创建数据包对象
2	CP5200_CmmPacker_Destroy	销毁数据包对象
3	CP5200_CmmPacket_SetParam	设置通信数据参数
4	CP5200_CmmPacker_Count	得到数据包对象中包的数量
5	CP5200_CmmPacker_Data	取得数据包对象中具体包的数据
6	CP5200_MakeSplitScreenData	生成分窗口命令的数据
7	CP5200_ParseSplitScreenRet	解析返回的分窗口命令的数据
8	CP5200_MakeSendTextData CP5200_MakeSendTextData1	生成发送文本命令的数据
9	CP5200_ParseSendTextRet	解析返回的发送文本命令的数据
10	CP5200_MakeSendTagTextData CP5200_MakeSendTagTextData1	生成发送文本命令的数据
11	CP5200_ParseSendTagTextRet	解析返回的发送文本命令的数据
12	CP5200_MakeSendPictureData	生成发送图片命令的数据
13	CP5200_ParseSendPictureRet	解析返回的发送图片命令的数据
14	CP5200_MakeSendSimpleImageData	生成发送简单图像数据命令的数据
15	CP5200_ParseSendSimpleImageRet	解析返回的发送简单图像命令的数据
16	CP5200_MakeSendStaticData	生成发送静态文本命令的数据
17	CP5200_ParseSendStaticRet	解析返回的发送静态文本命令的数据
18	CP5200_MakeSendClockData	生成发送时钟命令的数据
19	CP5200_ParseSendClockRet	解析返回的发送时钟命令的数据
20	CP5200_MakeExitSplitScreenData	生成退出分窗口命令的数据
21	CP5200_ParseExitSplitScreenRet	解析返回的退出分窗口命令的数据
22	CP5200_MakeSaveClearWndData	生成保存或清空窗口数据命令的数据
23	CP5200_ParseSaveClearWndRet	解析返回的保存或清空窗口数据命令的数据
24	CP5200_MakePlaySelectedPrgData CP5200_MakePlaySelectedPrgData1	生成选择播放节目命令的数据
25	CP5200_ParsePlaySelectedPrgRet	解析返回的选择播放节目命令的数据
26	CP5200_MakeSetUserVarData	生成设置用户变量命令的数据
27	CP5200_ParseSetUserVarRet	解析返回的设置用户变量命令的数据
28	CP5200_MakeSelectedAndUserVarData	生成选择节目及用户变量数据
29	CP5200_ParseSelectedAndUserVarRet	解析返回的选择播放节目及用户变量命令的数据
30	CP5200_MakeSetGlobalZoneData	生成设置全局区域的数据

31	CP5200_ParseSetGlobalZoneRet	解析返回的设置全局区域命令的数据
32	CP5200_MakePushUserVarData	生成推移用户变量的数据
33	CP5200_ParsePushUserVarRet	解析返回的推移用户变量命令的数据
34	CP5200_MakeTimerCtrlData	生成计时器控制数据
35	CP5200_ParseTimerCtrlRet	解析返回的计时器控制命令数据
36	CP5200_MakeSetZoneAndVariableData	生成设置全局显示区域及变量值的数据
37	CP5200_ParseSetZoneAndVariableRet	解析返回的设置全局显示区域及变量值的数据
38	CP5200_MakeSendPureTextData	生成发送纯文本命令的数据
39	CP5200_ParseSendPureTextRet	解析返回的发送纯文本命令的数据

本类函数的使用步骤如下：

第1步：创建数据包对象

第2步：生成通信所需的数据，包含了RS232/485的转码处理(0xa5 => 0xaa 0x05, ...)，或者网络通信的识别码信息

第3步：得到数据包对象的包数

第4步：逐一将每一个数据包的数据按下列方式处理：

1. 将该包数据发送到控制卡上
2. 接收该包返回的数据，接收时进行转码处理(0xaa 0x05 => 0xa5, ...)
3. 解析该包返回的数据得到返回结果

第5步：销毁数据包对象

5.2、数据通信 API 接口详细描述

CP5200_CmmPacker_Create

HOBJECT CP5200_CmmPacker_Create(int nCommType, BYTE byCardID, DWORD dwIDCode)	
作用	创建数据包对象
参数	nCommType: RS232/485 或者网络通信方式 0: RS232/485 1: 网络
	byCardID: 控制卡 ID
	dwIDCode: 控制卡的网络识别码。RS232/485 通信方式时忽略此参数。
返回值	数据对象句柄，供本类 API 接口函数使用 返回“空”表示创建失败

其它说明	对象创建成功并不再使用时，必须将对象销毁
------	----------------------

CP5200_CmmPacker_Destroy

int CP5200_CmmPacker_Destroy(HOBJECT hObj)	
作用	销毁数据包对象
参数	hObj: 要销毁的数据包对象句柄
返回值	0: 无错误 -1: 不正确的数据包对象句柄
其它说明	

CP5200_CmmPacket_SetParam

HOBJECT CP5200_CmmPacket_SetParam (HOBJECT hObj, int nParamType, const char *pParamString)	
作用	设置数据包参数
参数	hObj: 数据对象句柄
	nParamType: 参数类型，目前只有 1 有效
	pParamString: 参数字符串 当参数类型为 1 时，pParamString 为控制卡的设备 ID，这个参数只有控制卡作为客户端通信或者控制卡板载 GPRS 通信时才需要设置
返回值	1: 无错误 0: 参数类型不对 -1: 不正确的数据对象句柄 -2: 参数字符串值不对
其它说明	

CP5200_CmmPacker_Count

int CP5200_CmmPacker_Count(HOBJECT hObj)	
作用	得到数据包对象中包的数量

参数	hObj:数据包对象句柄
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_CmmPacker_Data

int CP5200_CmmPacker_Data(HOBJECT hObj , BYTE *pBuffer, int nBufSize, int nPackIndex)	
作用	取得数据包对象中具体包的数据
参数	hObj: 数据包对象句柄
	pBuffer: 结果数据缓冲区
	nBufSize: 结果数据缓冲区的大小（字节）
	nPackIndex: 要取得数据的子包索引，从 0 开始
返回值	>=0: 取得的数据的长度 -1: 不正确的数据对象句柄
其它说明	

CP5200_MakeSplitScreenData

int CP5200_MakeSplitScreenData(HOBJECT hObj, int nScrWidth, int nScrHeight, int nWndCnt, const int *pWndRects);	
作用	生成分窗口命令的数据
参数	hObj: 数据包对象句柄
	nScrWidth: 显示屏宽度
	nScrHeight: 显示屏高度
	nWndCnt: 要划分的窗口数量，有效值 1~8。
	pWndRects: 窗口的坐标，每个窗口用 4 个整数表示其“左、上、右、下”的坐标值。与 Windows 的 RECT 数据结构相同。
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄

其它说明	
------	--

CP5200_ParseSplitScreenRet

int CP5200_ParseSplitScreenRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的分窗口命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendTextData

(CP5200_MakeSendTextData1)

int CP5200_MakeSendTextData(HOBJECT hObj, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment);	
作用	生成发送文本命令的数据
参数	hObj: 数据包对象句柄
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码 , 本参数只支持文字大小, 不支持多种字体
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。

	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	CP5200_MakeSendTextData1 接口用于纯粹单字节文字（英文、欧洲文字）的发送

CP5200_ParseSendTextRet

int CP5200_ParseSendTextRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的发送文本命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendTagTextData

(CP5200_MakeSendTagTextData1)

int CP5200_MakeSendTagTextData(HOBJECT hObj, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment)	
作用	生成发送标签内容的数据
参数	hObj: 数据包对象句柄

	nWndNo: 窗口号
	pText: 文本字符串, 可包含扩展标记。参见”1.6”节
	crColor:颜色
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码 , 本参数只支持文字大小, 不支持多种字体
	nSpeed: 显示速度
	nEffect:显示效果。见 1.5、文字、图片显示效果编码
	nStayTime:停留时间。高字节在前。单位是秒。
	nAlignment: 对齐方式。 0: 左对齐 1: 水平居中 2: 右对齐
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	CP5200_MakeSendTagTextData1 接口用于纯粹单字节文字(英文、欧洲文字)的发送

CP5200_ParseSendTagTextRet

int CP5200_ParseSendTagTextRet(HOJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的标签数据
参数	hObj: 数据对象句柄
	pBuffer: 返回数据缓冲区
	nLength 返回的数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendPictureData

<pre>int CP5200_MakeSendPictureData(HOBJECT hObj, int nWndNo, int nPosX, int nPosY, int nCx, int nCy, const char *pPictureFile, int nSpeed, int nEffect, int nStayTime, int nPictRef);</pre>	
作用	生成发送图片命令的数据
参数	hObj: 数据包对象句柄
	nWndNo: 窗口的顺序号, 有效值 0~7
	nPosX: 图片显示的起点 X 偏移, 相对窗口左边。
	nPosY: 图片显示的起点 Y 偏移, 相对窗口上边。
	nCx: 图片宽度
	nCy: 图片高度
	pPictureFile: 图片文件, 其含义由参数 nPictRef 确定: 当 nPictRef 为 0 时: pPictureFile 为电脑存在的实际图片的路径名 当 nPictRef 为 1 时: pPictureFile 为在控制卡上预存的 gif 图像文件的文件名。 当 nPictRef 为 2 时: pPictureFile 为电脑存在的实际图片的路径名 当 nPictRef 为 3 时: pPictureFile 为控制卡上预存的图像包文件名 以及图片序号。中间以空格分隔。例如" images.rpk 1"
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
nPictRef: 图片发送方式及含义 0: 显示用户指定的本地图片文件, 将转换成 GIF 格式发送 1: 显示控制卡上预存的 GIF 图片文件 2: 显示用户指定的本地图片文件, 将转换成简单图片格式发送 3: 显示控制上预存的图像包中的图片文件 其他值: 当 0 值处理	
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseSendPictureRet

int CP5200_ParseSendPictureRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的发送图片命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendSimpleImageData

int CP5200_MakeSendSimpleImageData (HOBJECT hObj, int nWndNo, int nPosX, int nPosY, int nSpeed, int nEffect, int nStayTime, BYTE* pPicData, long lPicDataLen);	
作用	生成发送简单图像数据命令的数据
参数	hObj: 数据包对象句柄
	nWndNo: 窗口的顺序号, 有效值 0~7
	nPosX: 图片显示的起点 X 偏移, 相对窗口左边。
	nPosY: 图片显示的起点 Y 偏移, 相对窗口上边。
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	pPicData: 简单图片数据, 格式参见 1.11、简单图片数据格式 。
	lPicDataLen: 图片数据长度
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄

其它说明	
------	--

CP5200_ParseSendSimpleImageRet

int CP5200_ParseSendSimpleImageRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的发送简单图像数据命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendStaticData

int CP5200_MakeSendStaticData(HOBJECT hObj, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nAlignment, int x, int y, int cx, int cy);	
作用	生成发送静态文本命令的数据
参数	hObj: 数据包对象句柄
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	nAlignment: 文字对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
x: 显示区域起点 X	

	y: 显示区域起点 Y
	cx: 显示区域宽度
	cy: 显示区域高度
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseSendStaticRet

int CP5200_ParseSendStaticRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的发送静态文本命令的数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendClockData

int CP5200_MakeSendClockData (HOBJECT hObj, int nWinNo, int nStayTime, int nCalendar, int nFormat, int nContent, int nFont, int nRed, int nGreen, int nBlue, LPCSTR pTxt);	
作用	生成发送时钟命令的数据
参数	hObj: 数据包对象句柄
	nWndNo: 窗口的顺序号, 有效值 0~7。
	nStayTime: 停留时间, 单位是秒。

	nCalendar: 历法 0: 公历时间日期 1: 农历时间日期 2: 农历节气 3: 农历时间日期+节气	
	nFormat: 格式 bit 0: 时制(0: 12 小时制; 1: 24 小时制) bit 1: 年份位数(0: 4 位; 1: 2 位) bit 2: 分行(0: 单行; 1: 多行) bit 3~5: 格式控制, 如 2010 年 11 月 12 日星期五 的格式根据取值不同分别表示如下: 0: 2010 年 11 月 12 日 星期五 16:20:30 1: Fri, 12/11/2010 16:20:30 2: 2010-11-12 Fri. 16:20:30 3: Friday, 12 November 2010 16:20:30 4: Fri, Nov 12, 2010 16:20:30 5: Friday, November 12 2010 16:20:30 6: Fri, 11/12/2010 16:20:30 7: 2010/11/12, Fri. 16:20:30 bit 6: 显示时标、分标 bit 7: 透明显示	
	nContent: 内容 按位确定要显示的内容。 bit 7: 指针 bit 6: 星期 bit 5: 秒 bit 4: 分 bit 3: 时 bit 2: 日 bit 1: 月 bit 0: 年	
	nFont: 字体, Bit0~3: 文字大小	
	nRed: 颜色的红色分量	
	nGreen: 颜色的绿色分量	
	nBlue: 颜色的蓝色分量	
	pTxt: 以 0x00 结束的文字串	
	返回值	>=0: 包的数量 -1: 不正确的数据对象句柄

其它说明	
------	--

CP5200_ParseSendClockRet

int CP5200_ParseSendClockRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的发送时钟命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeExitSplitScreenData

CP5200_MakeExitSplitScreenData (HOBJECT hObj)	
作用	生成退出分窗口命令的数据
参数	hObj: 数据包对象句柄
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseExitSplitScreenRet

int CP5200_ParseExitSplitScreenRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的退出分窗口命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度

返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSaveClearWndData

CP5200_MakeSaveClearWndData(HOBJECT hObj, int nSavaOrClear);	
作用	生成保存或清空窗口数据命令的数据
参数	hObj: 数据包对象句柄
	nSavaOrClear: 保存还是清楚 0: 保存数据到 flash。 1: 清除 flash 保存的数据。
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseSaveClearWndRet

int CP5200_ParseSaveClearWndRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的保存或清空窗口数据命令的数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakePlaySelectedPrgData

int CP5200_MakePlaySelectedPrgData (HOBJECT hObj, const WORD *pSelected, int nSelCnt, int nOption)	
作用	生成分窗口命令的数据
参数	hObj: 数据包对象句柄
	pSelected: 选播的节目号数组
	nSelCnt: 选择的节目数量
	nOption: 是否保存选播信息到 FLASH 0: 不保存 1: 保存
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_MakePlaySelectedPrgData1

int CP5200_MakePlaySelectedPrgData1 (HOBJECT hObj, const WORD *pSelected, int nSelCnt, int nOption, int nScrWidth, int nScrHeight, byte byColorGray, byte nWndCnt)	
作用	生成分窗口命令的数据
参数	hObj: 数据包对象句柄
	pSelected: 选播的节目号数组
	nSelCnt: 选择的节目数量
	nOption: 是否保存选播信息到 FLASH 0: 不保存 1: 保存
	nScrWidth: 屏体宽度
	nScrHeight: 屏体高度
	byColorGray: 颜色灰度
	nWndCnt: 窗口数
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	本函数中发送的节目号在协议中用双字节表示

CP5200_ParsePlaySelectedPrgRet

int CP5200_ParsePlaySelectedPrgRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的选择播放节目命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSetUserVarData

int CP5200_MakeSetUserVarData(HOBJECT hObj, int nOption, int nVarNum, int bAstride, int* nVarLen, byte* byNoData);	
作用	生成分窗口命令的数据
参数	hObj: 数据包对象句柄
	nOption: Bit0: 是否保存所有变量值到 FLASH 0 不保存, 1 保存。 Bit1: 是否先清除原来的所有变量值 0 不清除, 1 清除。 Bit2~7: 保留, 设置为 0
	nVarNum: 要设置值的变量数量
	bAstride: 是否允许跨变量区设置。0 不允许; 1 允许
	nVarLen: 按顺序指定每个变量数据的字节
	byNoData: 按顺序指定每个变量的变量号及变量数据, 对于每一变量第一字节是变量号, 后面接着是指定长度的变量数据
	返回值

其它说明	<p>每个变量号对应一个变量区，每个变量区的大小是 32 字节，可将多个连续的变量区连在一起作为一个变量使用，被占用了变量区的变量号不能再使用。</p> <p>不允许跨变量区时，超过 32 字节的数据被丢弃；允许跨变量区时，根据数据长度计算需要使用的变量区数量</p> <p>变量号有效值是 1~100。每个变量号对应的变量区可存放 32 字节数据，连续的多个变量区可合起来供一个变量使用，被占用了变量区的变量号不能再使用。</p> <p>不更新变量值，只想执行保存变量值到 FLASH 动作时可“数量”的值为 0，设置“选项”为保存</p>
------	---

CP5200_ParseSetUserVarRet

int CP5200_ParseSetUserVarRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的选择播放节目命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	<p>0: 成功</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 接收数据命令码不对</p> <p>-3: 接收数据长度有误</p>
其它说明	

CP5200_MakeSelectedAndUserVarData

int CP5200_MakeSelectedAndUserVarData(HOBJECT hObj, int nOption, int nVarNum, int bAstride, int* nVarLen, byte* byNoData, int nSelPrg)	
作用	生成选播节目及用户变量数据。
参数	hObj:数据包对象句柄

	<p>nOption: 选项。</p> <p>Bit0: 是否保存节目号到 FLASH 0 不保存, 1 保存。</p> <p>Bit1: 是否保存全部变量值到 FLASH 0 不保存, 1 保存。</p> <p>Bit2: 是否保存先清除原来的所有变量值 0 不清除, 1 清除。</p> <p>Bit3~7: 保留, 设置为 0</p>
	nVarNum: 变量数量。Bit0~6: 要设置值的变量数量
	<p>bAstride:跨区域允许。</p> <p>Bit7: 是否允许跨变量区设置。0 不允许; 1 允许 每个变量号对应一个变量区, 每个变量区的大小是 32 字节, 可将多个连续的变量区连在一起作为一个变量使用, 被占用了变量区的变量号不能再使用。</p> <p>不允许跨变量区时, 超过 32 字节的数据被丢弃; 允许跨变量区时, 根据数据长度计算需要使用的变量区数量</p>
	nVarLen:变量数据长度表。按书序指定每个变量数据的字节。变量号及数据一起的长度是(1+n)字节。
	byNoData:变量号及数据。第一字节是变量号, 后面接着是指定长度的变量数据
	<p>nSelPrg: 节目号。要选播的节目号列表。每个节目号 2 个字节, 高字节在前</p> <p>超出预存节目数量的节目号被忽略</p>
返回值	<p>>=0: 包的数量</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseSelectedAndUserVarRet

int CP5200_ParseSelectedAndUserVarRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的选播节目及变量值
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据长度

返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSetGlobalZoneData

<pre>int CP5200_MakeSetGlobalZoneData(HOBJECT hObj, byte byConfig, byte bySynchro, byte byZoneNum, byte *byZoneMsg)</pre>	
作用	生成设置全局显示区域消息的数据
参数	hObj:数据包对象句柄
	byConfig: 选项。Bit0: 是否保存设置到 FLASH 0 不保存, 1 保存。 Bit1~7: 保留, 设置为 0
	bySynchro:同步。Bit0: 同步显示。0 不同步, 1 同步。 Bit1~7: 保留
	byZoneNum:区域数量。要设置的全局显示区域数量 区域数量为 0 取消所有区域
	byZoneMsg: 区域定义, 大小为区域数量*16 字节。全局显示区域的具体定义见: 1.7、全局区域信息格式
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseSetGlobalZoneRet

<pre>int CP5200_ParseSetGlobalZoneRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的全局区域数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据长度

返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakePushUserData

<pre>int CP5200_MakePushUserData(HOBJECT hObj, byte byOption , byte byVarZoonNum , byte byVarDataLen , byte* pVarNoAndData)</pre>	
作用	生成推移及用户变量数据
参数	hObj:数据包对象句柄
	byOption: 选项。Bit0: 是否保存所有变量值到 FLASH 0 不保存, 1 保存。 Bit1: 方向。0 往后推移,1 往前推移 Bit 2~3: 保留, 设置为 0 Bit4~7: 推移量。+1 为推移的变量区数量
	byVarZoonNum:变量区数量。Bit0~6: 要推移的变量数量区数量 1~100 Bit7: 保留, 填 0
	byVarDataLen:变量数据长度。指定变量数据的字节。变量号及数据一起的长度是(1+n)字节。
	pVarNoAndData: 变量号及数据。第一字节是变量号, 后面接着是指定长度的变量数据
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParsePushUserVarRet

<pre>int CP5200_ParsePushUserVarRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的推移及用户变量数据
参数	hObj: 数据包对象句柄

	pBuffer: 返回数据缓冲区
	nLength: 返回数据长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeTimerCtrlData

int CP5200_MakeTimerCtrlData(HOBJECT hObj, byte byTimerNo , byte byCmd , byte byProp , byte byValue[4])	
作用	生成计时器控制数据
参数	hObj:数据包对象句柄
	byTimerNo: 计时器编号。按位标识要设置的计时器。位值为 1 有效 Bit0: 计时器 1。 Bit1: 计时器 2。 Bit3: 计时器 3。 Bit4: 计时器 4。 Bit5: 计时器 5。 Bit6: 计时器 6。 Bit7: 计时器 7。
	byCmd: 动作。1: 初始化计时器 2: 复位计时器 3: 启动计时器 4: 暂停计时器 其它: 保留
	byProp:属性。根据动作的不同, 含义不同。 动作为初始化计时器时: Bit0: 0 正计时, 1 倒计时 Bit1: 0 暂停, 1 立即启动 Bit2~3: 保留 Bit4~7: 计时量 动作为其他值时, 保留填 0

	<p>byValue:值。根据动作的不同, 含义不同。</p> <p>动作为初始化计时器时:</p> <p>倒计时的初始值, 单位是毫秒</p> <p>高字节在前</p> <p>正计时此值填 0</p> <p>动作为其他值时, 保留填 0</p>
返回值	<p>>=0: 包的数量</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CP5200_ParseTimerCtrlRet

<pre>int CP5200_ParseTimerCtrlRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的计时器控制数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据长度
返回值	<p>0: 成功</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 接收数据命令码不对</p> <p>-3: 接收数据长度有误</p>
其它说明	

CP5200_MakeSetZoneAndVariableData

<pre>int CP5200_MakeSetZoneAndVariableData(HOBJECT hObj, const BYTE* pZoneData, int nZoneLen, const BYTE* pVariableData, int nVarLen, WORD wCtrl, WORD wReserved)</pre>	
作用	设置全局显示区域及变量数据
参数	hObj:数据包对象句柄
	pZoneData: 全局区域数据, 包括区域选项、区域数量、区域编号、区域定义
	nZoneLen: 区域数据长度

	pVariableData: 变量数据, 包括变量选项、变量数据及跨区允许、变量数据长度表、变量号及数据
	nVarLen: 变量数据长度
	wCtrl: 有效控制参数 播放次数。高字节在前。 该值为 0 表示一直有效。 Bit15: 保留, 填 0 Bit0~14: 显示次数
	wReserved: 保留
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	使用本指令之后, 全局区域自动变为同步显示

CP5200_ParseSetZoneAndVariableRet

<pre>int CP5200_ParseSetZoneAndVariableRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的设置全局显示区域及用户变量数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendPureTextData

<pre>int CP5200_MakeSendPureTextData(HOBJECT hObj, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment);</pre>	
作用	生成发送纯文本命令的数据

参数	hObj: 数据包对象句柄
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseSendPureTextRet

int CP5200_ParseSendPureTextRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的发送文本命令的数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CP5200_MakeSendMultiProtocol

int CP5200_MakeSendMultiProtocol (HOBJECT hObj, int nItem, const BYTE *pText, int nLength)	
作用	生成发送多协议的数据
	hObj: 数据包对象句柄
	nItem: 数据项数量
	pText: 数据项数据文本, 参见 《C-Power 控制卡外部调用通信协议》发送多协议数据: CC=0x60
	nLength: 数据项数据长度
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CP5200_ParseSendMultiProtocolRet

int CP5200_ParseSendMultiProtocolRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的发送多协议命令的数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

6、模板数据通信 API 接口

6.1、接口总览

序号	API 接口函数名	接口说明
1	CPowerBox_MakeSetProgramTemplateData CPowerBox_MakeSetProgramTemplateData l	生成节目模板数据
2	CPowerBox_ParseSetProgramTemplateRet	解析返回的节目模板数据
3	CPowerBox_MakeInOutProgramTemplateData	生成进出节目模板数据
4	CPowerBox_ParseInOutProgramTemplateRet	解析返回的进出节目模板数据
5	CPowerBox_MakeQueryProgramTemplateData CPowerBox_MakeQueryProgramTemplateData l	生成查询节目模板数据
6	CPowerBox_ParseQueryProgramTemplateRet	解析返回的查询节目模板数据
7	CPowerBox_MakeDeleteProgramData	生成删除节目数据
8	CPowerBox_ParseDeleteProgramRet	解析返回的删除节目数据
9	CPowerBox_MakeSendTextData	生成发送文本数据
10	CPowerBox_ParseSendTextRet	解析返回的发送文本数据
11	CPowerBox_MakeSendPictureData	生成发送的图片数据
12	CPowerBox_ParseSendPictureRet	解析返回的发送图片数据
13	CPowerBox_MakeSendClockOrTemperatureData	生成发送时钟或温度的数据
14	CPowerBox_ParseSendClockOrTemperatureRet	解析返回的发送时钟或温度的数据
15	CPowerBox_MakeSetAloneProgramData	生成设置独立节目的数据
16	CPowerBox_ParseSetAloneProgramRet	解析返回的设置独立节目的数据
17	CPowerBox_MakeQueryProgramData	生成查询节目信息的数据
18	CPowerBox_ParseQueryProgramRet	解析返回的查询节目信息的数据
19	CPowerBox_MakeSetProgramPropertyData	生成设置节目属性数据
20	CPowerBox_ParseSetProgramPropertyRet	解析返回的节目属性数据
21	CPowerBox_MakeSetScheduleData	生成设置播放计划的数据
22	CPowerBox_ParseSetScheduleRet	解析返回的设置播放计划的数据
23	CPowerBox_MakeDeleteScheduleData	生成删除播放计划的数据
24	CPowerBox_ParseDeleteScheduleRet	解析返回的删除播放计划的数据

25	CPowerBox_MakeGetScheduleData	生成查询播放计划的数据
26	CPowerBox_ParseGetScheduleRet	解析返回的查询播放计划的数据

本类函数的使用步骤如下：

- 第1步：创建数据包对象
- 第2步：生成通信所需的数据，包含了RS232/485的转码处理 (0xa5 => 0xaa 0x05, ...)，或者网络通信的识别码信息
- 第3步：得到数据包对象的包数
- 第4步：逐一将每一个数据包的数据按下列方式处理：
 4. 将该包数据发送到控制卡上
 5. 接收该包返回的数据，接收时进行转码处理 (0xaa 0x05 => 0xa5, ...)
 6. 解析该包返回的数据得到返回结果
- 第5步：销毁数据包对象

6.2、接口详细描述

CPowerBox_MakeSetProgramTemplateData

<pre>int CPowerBox_MakeSetProgramTemplateData(HOBJECT hObj, byte byColor ,USHORT nWidth , USHORT nHeight , byte nWndNum , byte *byDefParam , byte* pWndParam);</pre>	
作用	生成节目模板数据
参数	hObj:数据包对象句柄
	byColor: Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~6: 灰度级别 0: 2 级灰度, 7: 256 级灰度 Bit7: 保留
	nWidth: 屏宽, 高字节在前
	nHeight: 屏高, 高字节在前
	nWndNum:播放窗口数量, 最多支持 10 个窗口

	<p>byDefParam:缺省参数。Byte0~1: 停留时间, 单位是秒。高字节在前</p> <p>Byte2: 速度。该值越小, 移动越快</p> <p>Byte3: 文字大小。参见 1.7、文字大小编码</p> <p>Byte4: 文字颜色。参见单字节 1.8、文字颜色编码</p> <p>Byte5: 显示特效。参见 1.5、文字、图片显示效果编码</p> <p>Byte6: 图片方式。参见 1.9、图片方式编码</p> <p>Byte7: 时钟格式。参见 1.10、时钟格式和显示内容中的”时钟格式”</p> <p>Byte8: 时钟内容。参见 1.10、时钟格式和显示内容中的”时钟显示内容”</p>
	<p>pWndParam: 窗口参数。每个窗口 16 字节的参数。</p> <p>数据总长度是: 窗口数量*16。</p> <p>详见 1.13、窗口位置及属性</p>
返回值	<p>>=0: 包的数量</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CPowerBox_MakeSetProgramTemplateData1

<pre>int CPowerBox_MakeSetProgramTemplateData(HOBJECT hObj, byte byColor ,USHORT nWidth , USHORT nHeight , byte nWndNum , BYTE byOption, byte *byDefParam , byte* pWndParam);</pre>	
作用	生成节目模板数据
参数	hObj:数据包对象句柄
	<p>byColor: Bit0: 红色标记</p> <p>Bit1: 绿色标记</p> <p>Bit2: 蓝色标记</p> <p>Bit3: 保留</p> <p>Bit4~6: 灰度级别</p> <p>0: 2 级灰度, 7: 256 级灰度</p> <p>Bit7: 保留</p>
	nWidth: 屏宽, 高字节在前
	nHeight: 屏高, 高字节在前
	nWndNum:播放窗口数量, 最多支持 10 个窗口

	<p>byOption: 选项:</p> <p>Bit0: 强制进入节目模板方式运行</p> <p>Bit1: 模板保存位置。0 用户盘, 1 系统盘。 若模板保存到系统盘, 则用户盘原有的模板被清除; 若模板保存到用户盘, 则系统盘原有的模板被清除。</p> <p>Bit2~7: 保留</p>
	<p>byDefParam: 缺省参数。Byte0~1: 停留时间, 单位是秒。高字节在前</p> <p>Byte2: 速度。该值越小, 移动越快</p> <p>Byte3: 文字大小。参见 1.7、文字大小编码</p> <p>Byte4: 文字颜色。参见单字节 1.8、文字颜色编码</p> <p>Byte5: 显示特效。参见 1.5、文字、图片显示效果编码</p> <p>Byte6: 图片方式。参见 1.9、图片方式编码</p> <p>Byte7: 时钟格式。参见 1.10、时钟格式和显示内容中的”时钟格式”</p> <p>Byte8: 时钟内容。参见 1.10、时钟格式和显示内容中的”时钟显示内容”</p>
	<p>pWndParam: 窗口参数。每个窗口 16 字节的参数。 数据总长度是: 窗口数量*16。 详见 1.13、窗口位置及属性</p>
返回值	<p>>=0: 包的数量</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CPowerBox_ParseSetProgramTemplateRet

<pre>int CPowerBox_ParseSetProgramTemplateRet(HOBJECT hObj, const BYTE* pBuffer, int nLength);</pre>	
作用	解析返回的节目模板数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLenght: 返回数据的长度
返回值	<p>0: 成功</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 接收数据命令码不对</p> <p>-3: 接收数据长度有误</p>
其它说明	

CPowerBox_MakeInOutProgramTemplateData

<code>int CPowerBox_MakeInOutProgramTemplateData(HOBJECT hObj, byte byInOrOut);</code>	
作用	生成进出节目的模板数据
参数	<code>hObj</code> :数据包对象句柄
	<code>byInOrOut</code> : 进入或退出。 1: 进入节目模板方式 0: 退出节目模板方式
返回值	<code>>=0</code> : 包的数量 <code>-1</code> : 不正确的数据对象句柄
其它说明	

CPowerBox_ParseInOutProgramTemplateRet

<code>int CPowerBox_ParseInOutProgramTemplateRet(HOBJECT hObj, const BYTE* pBuffer, int nLength);</code>	
作用	解析返回的进出节目的模板数据
参数	<code>hObj</code> :数据包对象句柄
	<code>pBuffer</code> : 返回数据缓冲区
	<code>nLength</code> : 返回数据的长度
返回值	0: 成功 <code>-1</code> : 不正确的数据对象句柄 <code>-2</code> : 接收数据命令码不对 <code>-3</code> : 接收数据长度有误
其它说明	

CPowerBox_MakeQueryProgramTemplateData

<code>int CPowerBox_MakeQueryProgramTemplateData(HOBJECT hObj);</code>	
作用	生成查询节目模板数据
参数	<code>hObj</code> :数据包对象句柄

返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_MakeQueryProgramTemplateData1

<code>int CPowerBox_MakeQueryProgramTemplateData(HOBJECT hObj , byte byFlag);</code>	
作用	生成查询节目模板数据
参数	hObj:数据包对象句柄 byFlag: 选项 Bit0: 是否查询模板状态 Bit1: 返回模板定义的颜色灰度、的屏大小信息 Bit1~7: 保留
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseQueryProgramTemplateRet

<code>int CPowerBox_ParseQueryProgramTemplateRet(HOBJECT hObj, const BYTE* pBuffer, int nLength ,BYTE* pInfoBuffer, int nInfoBufSize);</code>	
作用	解析返回的查询节目模板数据
参数	hObj:数据包对象句柄 pBuffer: 返回数据缓冲区 nLength: 返回数据的长度 pInfoBuffer:返回信息缓冲区 nInfoBufSize:返回信息的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

pInfoBuffer 数据含义如下:

数据项	数值	长度(字节)	意义描述
CC	0x83	1	说明本数据包是查询节目模板方式相关参数
选项		1	同发送值
模板方式		1	0: 非节目模板方式 1: 节目模板方式
模板状态		1	Bit0~1: 模板可用性 0: 模板不可用 1: 模板可用 其它: 保留 Bit2~7: 保留
颜色灰度		1	颜色及灰度。同“设置节目模板”的定义
屏宽度		2	高字节在前
屏高度		2	高字节在前
窗口数		1	播放窗口数量。最多支持 10 个播放窗口

CPowerBox_MakeDeleteProgramData

<pre>int CPowerBox_MakeDeleteProgramData(HOBJECT hObj, byte byConfig , byte byProNum , byte* pDelPro);</pre>	
作用	生成删除节目数据
参数	hObj:数据包对象句柄 byConfig:选项。 Bit0: 删除的节目范围 0: 删除全部 1: 删除指定节目 其它: 保留
	byProNum:节目数量。删除全部节目不需要此数据项
	pDelPro:节目列表。长度字节数等于节目数量。 每个节目号用 1 字节表示, 节目号从 1 开始
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseDeleteProgramRet

int CPowerBox_ParseDeleteProgramRet(HOBJECT hObj, const BYTE* pBuffer, int nLength);	
作用	解析返回的删除节目数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

CPowerBox_MakeSendTextData

int CPowerBox_MakeSendTextData(HOBJECT hObj, DWORD dwAppendCode , byte byProNo , byte byWndNo , byte byProp , byte *byShowFormat , char* pText);	
作用	生成发送文本数据
参数	hObj:数据包对象句柄
	dwAppendCode:用户给出的附加码, 高字节在前。
	byProNo: 节目号, 有效值 1~255
	byWndNo:窗口号。有效值 1~10, 超出节目模板定义时无效
	byProp:属性, Bit0~3: 文本类型 0: 普通文本 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留

	<p>byShowFormat: 显示格式。 当”属性”的显示格式为 0 时不需要此项数据</p> <p>Byte0~1: 停留时间。高字节在前</p> <p>Byte2: 速度。该值越小，速度越快</p> <p>Byte3: 文字大小。参见 1.7、文字大小编码</p> <p>Byte4: 文字颜色。参见单字节 1.8、文字颜色编码</p> <p>Byte5: 显示特效。参见 1.5、文字、图片显示效果编码</p> <p>Byte6: 保留</p> <p>Byte7: 保留</p>
	pText:文字数据。文本内容，以 0x00 结束。
返回值	<p>>=0: 包的数量</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CPowerBox_ParseSendTextRet

<pre>Int CPowerBox_ParseSendTextRet(HOBJECT hObj, const BYTE* pBuffer, int nLength);</pre>	
作用	解析返回的发送文本数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength:返回数据的长度
返回值	<p>0: 成功</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 接收数据命令码不对</p> <p>-3: 接收数据长度有误</p>
其它说明	

CPowerBox_MakeSendPictureData

<pre>int CPowerBox_MakeSendPictureData(HOBJECT hObj, DWORD dwAppendCode, byte byProNo, byte byWndNo, byte byPicType, byte *byShowFormat, byte* pPicData, long lPicDataLen);</pre>	
作用	生成发送图片信息
参数	hObj:数据包对象句柄

	dwAppendCode:用户给出的附加码,高字节在前。
	byProNo: 节目号,有效值 1~255
	byWndNo:窗口号。有效值 1~10,超出节目模板定义时无效
	byPicType: 图片类型。Bit0~3: 图片类型 1: GIF 图像文件的实际数据,里面包含图像的宽度、高度等信息; 2: 在控制卡上预存的 GIF 图像文件的文件名。 4: 简单图片数据,格式见 1.11、简单图片数据格式 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留
	byShowFormat: 显示格式。 当”属性”的显示格式为 0 时不需要此项数据 Byte0~1: 停留时间高字节在前 Byte2: 速度,该值越小,移动越快 Byte3: 显示特效,参见 1.5、文字、图片显示效果编码 Byte4: 图片方式(缩放、拉伸等),参加 1.9、图片方式编码 Byte5: 保留 Byte6: 保留 Byte7: 保留
	pPicData:图片数据。
	lPicDataLen:图片长度。
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseSendPictureRet

int CPowerBox_ParseSendPictureRet (HOBJECT hObj, const BYTE* pBuffer, int nLength);	
作用	解析返回的图片数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对

	-3: 接收数据长度有误
其它说明	

CPowerBox_MakeSendClockOrTemperatureData

<pre>int CPowerBox_MakeSendClockOrTemperatureData(HOBJECT hObj,DWORD dwAppendCode , BYTE byProNo , BYTE byWndNo , BYTE byProgramType , UINT nPropLen , BYTE* pProgramProp)</pre>	
作用	生成发送时钟和温度的数据
参数	hObj:数据包对象句柄
	dwAppendCode:用户给出的附加码，高字节在前。
	byProNo: 节目号，有效值 1~255
	byWndNo:窗口号。有效值 1~10，超出节目模板定义时无效
	byProgramType: 节目类型。 Bit0~3: 类型。2 时钟；3 温度，其它保留。 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留备用。填 0
	nPropLen: 属性长度
	pProgramProp: 节目属性 属性数据含义根据类型不同而不同 类型=2: 同 1.13 的“时钟日历”属性 类型=3: 同 1.13 的“温度、湿度”属性
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseSendClockOrTemperatureRet

<pre>int CPowerBox_ParseSendClockOrTemperatureRet(HOBJECT hObj, const BYTE* pBuffer, int nLength , BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的发送时钟和温度数据
参数	hObj:数据包对象句柄

	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer:返回信息缓冲区
	nInfoBufSize:返回信息的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

pInfoBuffer 数据含义如下:

数据项	数值	长度(字节)	意义描述
CC	0x83	1	说明本数据包是查询节目模板方式相关参数
用户附加码		4	用户给出的附加码。高字节在前
节目号		1	同发送值。有效值 1~100。
窗口号		1	同发送值。有效值 1~10, 超出节目模板定义时无效
丢包数		1	尚未接收到的包的数量。发送第一包时的丢包数量是总包数减去 1。
丢包的包号		变长	丢包的包号。总是按照从小到大排列; 第一包的包号是 0。每个包号一个字节。

CPowerBox_MakeSetAloneProgramData

<pre>int CPowerBox_MakeSetAloneProgramData (HOBJECT hObj, DWORD dwAppendCode , BYTE byProgramNo , BYTE byWindowCnt ,BYTE* pWndParam, BYTE* pWndData)</pre>	
作用	生成设置独立节目数据
参数	hObj:数据包对象句柄
	dwAppendCode:用户给出的附加码, 高字节在前。
	byProgramNo: 节目号, 有效值 1~100
	byWindowCnt:窗口数。有效值 1~10,

	<p>pWndParam: 窗口参数</p> <p>每个窗口的信息表 22 字节, 前 16 字为窗口位置及属性 (参见 1.13、窗口位置及属性), 17~19 字节为窗口数据偏移, 20~22 字节为窗口数据长度。高字节在前。无数据则偏移和长度均为 0。</p>
	<p>pWndData: 窗口数据</p> <p>窗口要播放的数据, 如“文本”、“图片”等。</p> <p>第 1 字节: 数据类型(1 文本; 4 图片)</p> <p>第 2 字节: 数据格式 (同发文本或发图片的定义)</p> <p>第 3 字节起: 文本或图片数据。</p>
返回值	<p>>=0: 包的数量</p> <p>-1: 不正确的数据对象句柄</p>
其它说明	

CPowerBox_ParseSetAloneProgramRet

<pre>int CPowerBox_ParseSetAloneProgramRet (HOBJECT hObj, const BYTE* pBuffer, int nLength , BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的设置独立节目数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer:返回信息缓冲区
	nInfoBufSize:返回信息的长度
返回值	<p>0: 成功</p> <p>-1: 不正确的数据对象句柄</p> <p>-2: 接收数据命令码不对</p> <p>-3: 接收数据长度有误</p> <p>0x01 节目模板无效</p> <p>0x11 节目号超出范围</p> <p>0x12 窗口号超出范围</p> <p>0x13 定义的窗口在节目模板定义的屏大小之外</p> <p>0x80 当前不是节目模板方式</p>

其它说明	
------	--

pInfoBuffer 含义如下:

数据项	数值	长度(字节)	意义描述
CC	0x88	1	说明本数据包是发送独立节目的返回包
用户附加码		4	用户给出的附加码。高字节在前
节目号		1	同发送值。有效值 1~100。
保留		1	保留备用
丢包数		1	尚未接收到的包的数量。发送第一包时的丢包数量是总包数减去 1。
丢包的包号		变长	丢包的包号。总是按照从小到大排列；第一包的包号是 0。每个包号一个字节。

***必须要先发送第一个数据包。最好能确认第一个数据包发送成功，然后再发送后续的数据包。**

*返回数据包中的“返回值”非零时的含义:

- 0x01 节目模板无效
- 0x11 节目号超出范围
- 0x12 窗口号超出范围
- 0x13 定义的窗口在节目模板定义的屏大小之外
- 0x80 当前不是节目模板方式

CPowerBox_MakeQueryProgramData

int CPowerBox_MakeQueryProgramData(HOBJECT hObj , byte byFlag , byte* pParam)	
作用	生成查询节目信息的数据
参数	hObj:数据包对象句柄
	byFlag: 查询标记, 指定要查询哪一项节目信息 0: 保留 1: 查询有效节目数量及节目号 2: 查询指定节目信息 其它: 保留
	pParam: 当查询标记为 1 时: byte1~5, 保留, 填 0 当查询标记为 2 时: byte1, 节目号; byte2~5, 保留, 填 0
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseQueryProgramRet

<pre>int CPowerBox_ParseQueryProgramRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)</pre>	
作用	解析返回的查询节目信息数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer:返回信息缓冲区
	nInfoBufSize:返回信息的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误
其它说明	

pInfoBuffer 中数据含义如下:

- 查询“有效节目数量及节目号”

数据项	数值	长度(字节)	意义描述
CC	0x89	1	说明本数据包是查询节目信息的返回包
信息标识		1	同发送值。
参数		5	同发送值。
有效节目数		1	有效节目的数量
有效节目号		变长	每一个字节标识一个有效的节目号。有效值 1~100。

*返回数据包中的“返回值”非零时的含义:

0x01 控制卡没有运行在节目模板方式

0x10 不认识的信息标识

- 查询指定节目信息

返回数据

数据项	数值	长度(字节)	意义描述
CC	0x89	1	说明本数据包是查询节目信息的返回包
信息标识		1	同发送值。
参数		5	同发送值。
信息数		1	现在仅返回 1 项信息
节目号		1	节目号

用户附加码		4	用户发送节目信息时指定的用户附加码
-------	--	---	-------------------

*返回数据包中的“返回值”非零时的含义：

- 0x01 控制卡没有运行在节目模板方式
- 0x10 不认识的信息标识
- 0x11 无效的节目
- 0x12 不能取得需要的信息

CPowerBox_MakeSetProgramPropertyData

<pre>int CPowerBox_MakeSetProgramPropertyData(HOBJECT hObj, byte byOption , byte byProgramCnt , byte* pPrograms , byte byPropertyID1 , byte byPropertyID2 , byte byProgramLevel , USHORT nLoopCnt , USHORT nTime , byte* pDuetime , byte* pTimeInterval);</pre>	
作用	生成设置节目属性数据
参数	hObj:数据包对象句柄
	byOption:选项, Bit0: 设置节目属性的范围 0: 全部节目 1: 指定节目 其它: 保留
	byProgramCnt:要生成的节目数量
	pPrograms:节目列表
	ByPropertyID1: 属性标示 1, 按位标识要设置节目的哪些属性, 标识为 0 则该项数据不存在。 Bit0: 节目级别。节目的重要级别 Bit1: 循环次数。节目每次播放时循环的次数 Bit2: 有效时长。从当前时间开始多长时间内有效播放 Bit3: 有效期限 Bit4~7: 保留
	ByPropertyID2: 属性标识 2。Bit0~4: 有效时段。>0 时为有效时段数.<=4 Bit5~7: 保留
	byProgramLevel: 节目级别。1~3 级。优先播放级别高的节目
	nLoopCnt:循环次数。高字节在前。 0:不播放该节目, 用于临时屏蔽节目 1~255:该节目每次播放时循环的次数

	<p>nTime: 有效时长。高字节在前。单位为分钟 0: 不限定播放时间 >0: 指定播放时间。单位是分钟。</p>
	pDuetime:有效时限。
	pTimeInterval:有效时段。时段开始的“时分秒”和结束的“时分秒”时分秒各用一个字节表示
返回值	<p>>=0: 包的数量 -1: 不正确的数据对象句柄</p>
其它说明	

CPowerBox_ParseSetProgramPropertyRet

<pre>int CPowerBox_ParseSetProgramPropertyRet(HOBJECT hObj, const BYTE* pBuffer, int nLength);</pre>	
作用	解析返回的设置节目属性数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	<p>0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误</p>
其它说明	

CPowerBox_MakeSetScheduleData

<pre>int CPowerBox_MakeSetScheduleData (HOBJECT hObj, DWORD dwAppendCode, BYTE byScheduleNo, const BYTE* pProperty, const BYTE* pBoxes, BYTE byBoxCnt);</pre>	
作用	生成设置播放计划数据
参数	hObj:数据包对象句柄
	dwAppendCode:用户给出的附加码，高字节在前。

	byScheduleNo: 计划项编号, 有效值 1~100。总共支持 100 个计划项, 同一个项的新数据覆盖旧数据。
	pProperty: 播放属性, 共 14 字节: byte 0: 格式及级别: Bit0~3: 表示数据格式, 填 0x01 Bit4~7: 表示优先级别。优先级别数值越大, 则越优先播放, 0 为最低优先级。 byte 1: 星期: Bit0~6: 7 个位分别标识星期日到星期六 byte 2~4: 开始日期, 3 个字节分别表示“年”“月”“日”。年份的值取 0~99, 表示范围是 2000~2099 byte 5~7: 结束日期, 3 个字节分别表示“年”“月”“日”。年份的值取 0~99, 表示范围是 2000~2099 byte 8~10: 开始时间, 3 个字节分别表示“时”“分”“秒” byte 11~13: 结束时间, 3 个字节分别表示“时”“分”“秒”
	pBoxes: 节目编号, 每个字节表示 1 个编号。编号从小到大排列, 不重复。
	byBoxCnt: 节目编号数量, 有效值 1~100
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseSetScheduleRet

int CPowerBox_ParseSetScheduleRet (HOBJECT hObj, const BYTE* pBuffer, int nLength)	
作用	解析返回的设置播放计划数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄

	-2: 接收数据命令码不对 -3: 接收数据长度有误 0x01: 节目模板无效 0x80: 当前不是节目模板方式
其它说明	

CPowerBox_MakeDeleteScheduleData

<pre>int CPowerBox_MakeDeleteScheduleData(HOBJECT hObj, DWORD dwAppendCode, const BYTE* pSchs, BYTE bySchCnt)</pre>	
作用	生成删除播放计划数据
参数	hObj: 数据包对象句柄
	dwAppendCode: 用户给出的附加码, 高字节在前。
	pSchs: 计划项编号, 编号有效值 1~100。每个字节表示 1 个编号。删除全部项时, 本项数据长度为 1 字节, 值是 0xff, 作为删除的确认标记。
	bySchCnt: 要删除的计划项的数量。0 表示删除全部
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseDeleteScheduleRet

<pre>int CPowerBox_ParseDeleteScheduleRet(HOBJECT hObj, const BYTE* pBuffer, int nLength)</pre>	
作用	解析返回的删除播放计划数据
参数	hObj: 数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对

	-3: 接收数据长度有误 0x01 节目模板无效 0x11 要删除的计划项数量为 0 0x80 当前不是节目模板方式
其它说明	

CPowerBox_MakeGetScheduleData

int CPowerBox_MakeGetScheduleData (HOBJECT hObj, DWORD dwAppendCode, BYTE byType, BYTE byScheduleNo)	
作用	生成获取播放计划的数据
参数	hObj:数据包对象句柄
	dwAppendCode:用户给出的附加码，高字节在前。
	byType: 0: 查询有效的计划项编号 1: 查询指定编号的计划项内容 其它: 保留
	byScheduleNo: 编号有效值 1~100。 当查询类型为 0 时，本项数据填 0。
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

CPowerBox_ParseGetScheduleRet

int CPowerBox_ParseGetScheduleRet (HOBJECT hObj, const BYTE* pBuffer, int nLength, BYTE* pInfoBuffer, int nInfoBufSize)	
作用	解析返回的获取播放计划数据
参数	hObj:数据包对象句柄
	pBuffer: 返回数据缓冲区
	nLength: 返回数据的长度
	pInfoBuffer:返回信息缓冲区
	nInfoBufSize:返回信息的长度

返回值	0: 成功 -1: 不正确的数据对象句柄 -2: 接收数据命令码不对 -3: 接收数据长度有误 0x01 节目模板无效 0x11 不支持的查询类型 0x12 指定的计划项编号超出有效范围 0x80 当前不是节目模板方式
其它说明	

pInfoBuffer 中数据格式如下:

数据项	数值	长度(字节)	意义描述
CC	0x8d	1	说明本数据包是查询节目模板方式相关参数
用户附加码		4	同发送数据
查询类型		1	同发送值
数量/编号		1	查询类型为 0, 该值是有效计划项数量 查询类型为 1, 该值是计划项编号
标号表/计划项内容		变长	查询类型为 0, 是有效计划项的编号表 查询类型为 1, 是计划项内容, 数据格式同“设置播放计划”的数据格式

要区分查询类型来处理返回数据

返回数据包中的“返回值”非零时的含义:

- 0x01 节目模板无效
- 0x11 不支持的查询类型
- 0x12 指定的计划项编号超出有效范围
- 0x80 当前不是节目模板方式

7、通信基本接口函数

7.1、RS232 通信基本 API 接口总览

序号	API 接口函数名	接口说明
1	CP5200_RS232_Init	初始化串口参数
2	CP5200_RS232_InitEx	初始化串口参数, 超时设置
3	CP5200_RS232_Open	打开串口
4	CP5200_RS232_OpenEx	打开串口, 指定读、写的超时时间
5	CP5200_RS232_Close	关闭串口
6	CP5200_RS232_IsOpened	测试串口是否已经打开

7	CP5200_RS232_Write	往串口写数据
8	CP5200_RS232_Read	从串口读数据
9	CP5200_RS232_WriteEx	往串口写数据，并作转码处理
10	CP5200_RS232_ReadEx	从串口读数据，并作转码处理

本类函数的使用步骤如下：

- 第1步：初始化串口
- 第2步：打开串口
- 第3步：对串口进行读写操作
- 第4步：关闭串口

7.2、RS232 通信基本 API 接口详细描述

CP5200_RS232_Init

<code>int CP5200_RS232_Init(const char *fName, int nBaudrate)</code>	
作用	初始化串口参数
参数	fName: RS232 串口名字，例如“COM1”、“COM2”、...
	nBaudrate: 波特率，例如 115200、57600、...
返回值	1: 成功 0: 失败
其它说明	其它串口参数是固定的： 校验位：无校验 数据位：8 停止位：1 流控：无

CP5200_RS232_InitEx

<code>int CP5200_RS232_InitEx(const char *fName, int nBaudrate, DWORD dwTimeout);</code>	
作用	初始化串口参数
参数	fName: RS232 串口名字，例如“COM1”、“COM2”、...

	nBaudrate: 波特率, 例如 115200、57600、...
	dwTimeout; 超时时间
返回值	1: 成功 0: 失败
其它说明	其它串口参数是固定的: 校验位: 无校验 数据位: 8 停止位: 1 流控: 无

CP5200_RS232_Open

int CP5200_RS232_Open(void)	
作用	打开串口
参数	无
返回值	1: 成功 0: 失败
其它说明	使用串口之后, 需要调用 CP5200_RS232_Close()将其关闭 读、写的超时时间均设置为 600 毫秒

CP5200_RS232_OpenEx

int CP5200_RS232_OpenEx(DWORD dwReadTimeout, DWORD dwWriteTimeout)	
作用	打开串口
参数	dwReadTimeout: 读操作的超时时间。单位毫秒
	dwWriteTimeout: 写操作的超时时间。单位毫秒
返回值	1: 成功 0: 失败
其它说明	使用串口之后, 需要调用 CP5200_RS232_Close()将其关闭

CP5200_RS232_Close

int CP5200_RS232_Close(void)	
作用	关闭串口
参数	无
返回值	1: 成功 0: 失败或者串口本来就是关闭状态
其它说明	

CP5200_RS232_IsOpened

int CP5200_RS232_IsOpened(void)	
作用	测试串口是否已经打开
参数	无
返回值	1: 已经打开 0: 没有打开
其它说明	

CP5200_RS232_Write

int CP5200_RS232_Write(const void* pBuf, int nLength)	
作用	往串口写数据
参数	pBuf: 数据缓冲区指针
	nLength: 数据长度
返回值	1: 成功 0: 失败或者串口本来就是关闭状态
其它说明	

CP5200_RS232_Read

int CP5200_RS232_Read(void* pBuf, int nBufSize)	
---	--

作用	从串口读数据
参数	pBuf: 数据缓冲区指针, 存放读到的数据
	nBufSize: 数据缓冲区的大小
返回值	数据长度
其它说明	

CP5200_RS232_WriteEx

int CP5200_RS232_WriteEx(const void* pBuf, int nLength)	
作用	往串口写数据, 并作转码处理
参数	pBuf: 数据缓冲区指针
	nLength: 数据长度
返回值	1: 成功 0: 失败或者串口本来就是关闭状态
其它说明	在数据开头加上开始码 0xA5, 在数据末尾加上结束码 0xAE。对发送的数据做了转码处理: 0xa5 => 0xaa 0x05 0xaa => 0xaa 0x0a 0xae => 0xaa 0x0e

CP5200_RS232_ReadEx

int CP5200_RS232_ReadEx(void* pBuf, int nBufSize)	
作用	从串口读数据, 并作转码处理
参数	pBuf: 数据缓冲区指针, 存放读到的数据
	nBufSize: 数据缓冲区的大小
返回值	数据长度
其它说明	读取开始码 0xA5 结束码 0xAE 之间的数据, 返回的数据不包括开始码和结束码, 并对开始码和结束码之间的数据做了转码处理: 0xaa 0x05 => 0xa5

	0xaa 0x0a => 0xaa
	0xaa 0x0e => 0xae

7.3、Network 通信基本 API 接口总览

序号	API 接口函数名	接口说明
1	CP5200_Net_Init	初始化网络参数
2	CP5200_Net_SetBindParam	绑定通信的客户端 IP 和端口
3	CP5200_Net_Connect	打开网络连接
4	CP5200_Net_IsConnected	判断网口是否已打开
5	CP5200_Net_Disconnect	关闭网络连接
6	CP5200_Net_Write	利用网络写数据
7	CP5200_Net_Read	利用网络读数据

本类函数的使用步骤如下：

- 第1步：初始化网络参数
- 第2步：打开网络连接
- 第3步：对网络进行读写操作
- 第4步：关闭网络连接

7.4、Network 通信基本 API 接口详细描述

CP5200_Net_Init

int CP5200_Net_Init(DWORD dwIP, int nIPPort, DWORD dwIDCode, int nTimeOut)	
作用	初始化网络参数
参数	dwIP:IP 地址。例如： 192.168.1.100 为 0xc0a80164
	nIPPort:端口
	dwIDCode:识别码
	nTimeOut:超时时间
返回值	0
其它说明	

CP5200_Net_SetBindParam

int CP5200_Net_SetBindParam(DWORD dwClientIP , int nClientPort)	
作用	绑定通信的客户端 IP 和端口
参数	dwClientIP: 要绑定的 IP 地址。例如: 192.168.1.100 为 0xc0a80164
	nClientPort:要绑定的端口
返回值	0
其它说明	

CP5200_Net_Connect

int CP5200_Net_Connect(void)	
作用	打开网络连接
参数	无
返回值	1: 成功 0: 失败 -1: IP 不合法
其它说明	

CP5200_Net_IsConnected

int CP5200_Net_IsConnected(void)	
作用	判断网口是否已打开
参数	无
返回值	1: 已经打开 0: 未打开
其它说明	

CP5200_Net_Disconnect

<code>int CP5200_Net_Disconnect(void)</code>	
作用	关闭网络连接
参数	无
返回值	1: 成功 0: 失败或者网络本来就是关闭状态
其它说明	

CP5200_Net_Write

<code>int CP5200_Net_Write(const BYTE* pBuf, int nLength);</code>	
作用	利用网络写数据
参数	pBuf: 数据缓冲区指针
	nLength: 数据长度
返回值	1: 成功 0: 失败 -1: 网络未连接
其它说明	

CP5200_Net_Read

<code>int CP5200_Net_Read(BYTE* pBuf, int nSize)</code>	
作用	利用网络读数据
参数	pBuf: 数据缓冲区指针, 存放读到的数据
	nSize: 数据缓冲区的大小
返回值	>0: 读取数据长度 0: 失败 -1: 网络未连接

其它说明	
------	--

8、运行计划接口函数

C-Power5200 控制卡支持按日期、星期的运行计划控制。运行计划以文件保存在控制卡上，文件名称固定是“playbill.rsf”。

运行计划 API 接口函数用于生成 playbill.rsf 文件。

8.1、运行计划 API 接口总览

序号	API 接口函数名	接口说明
1	CP5200_Runsch_Create	创建运行计划对象
2	CP5200_Runsch_Destroy	销毁运行计划对象
3	CP5200_Runsch_AddItem	添加运行计划项
4	CP5200_Runsch_SaveToFile	保存运行计划到文件

本类函数的使用步骤如下：

- 第1步：创建运行计划对象
- 第2步：添加运行计划项
- 第3步：保存运行计划到文件
- 第4步：销毁运行计划对象

8.2、运行计划 API 接口详细描述

CP5200_Runsch_Create

HOBJECT CP5200_Runsch_Create(int nPrgSum, int nAttrib)	
作用	创建运行计划对象
参数	nPrgSum: 节目总数
	nAttrib: 属性 0: 缺省时间段不播放任何节目 1: 缺省时间段播放全部节目
返回值	运行计划对象句柄，供本类 API 接口函数使用

	返回“空”表示创建失败
其它说明	对象创建成功并不再使用时，必须将对象销毁

CP5200_Runsch_Destroy

int CP5200_Runsch_Destroy(HOBJECT hObj)	
作用	销毁运行计划对象
参数	hObj: 要销毁的运行计划对象句柄
返回值	0: 无错误 -1: 对象句柄空 -2: 不正确的对象句柄
其它说明	

CP5200_Runsch_AddItem

int CP5200_Runsch_AddItem(HOBJECT hObj, int nGrade, int nWeekDateRelative, int nWeeks, const int* pBeginDate, const int* pEndDate, const int* pBeginTime, const int* pEndTime, int nItemCnt, const int *pItems)	
作用	增加运行计划项
参数	hObj: 运行计划对象句柄
	nGrade: 计划项级别, 0~9 级别越高越优先。
	nWeekDateRelative: 星期与日期的关系 0: 星期与日期同时满足才执行本项计划 1: 星期或者日期之一满足就执行本项计划

	<p>nWeeks: 星期的标记,可以是以下值之一或多个值的组合</p> <p>1: 星期天</p> <p>2: 星期一</p> <p>4: 星期二</p> <p>8: 星期三</p> <p>16: 星期四</p> <p>32: 星期五</p> <p>64: 星期六</p>
	pBeginDate: 开始日期。3 个整数值分别表示“年”“月”“日”
	pEndDate: 结束日期。3 个整数值分别表示“年”“月”“日”
	pBeginTime: 开始时间。3 个整数值分别表示“时”“分”“秒”
	pEndTime: 结束时间。3 个整数值分别表示“时”“分”“秒”
	<p>nItemCnt: 本计划项要播放的节目数量。数量不能大于 CP5200_RunSch_Create 第一个参数指定的数量。</p>
	<p>pItems: 本计划要播放的节目号,长度是”nItemCnt”个整数。每个整数是要播放的节目号,节目号从 0 开始编号,编号小于 CP5200_RunSch_Create 第一个参数指定的数量</p>
返回值	<p>>=0: 成功, 计划项序号</p> <p>-1: 不正确的对象句柄</p> <p>-2: 参数非法</p> <p>-3: 内存不足</p> <p>-4: 内存错误</p>
其它说明	

CP5200_RunSch_SaveToFile

int CP5200_RunSch_SaveToFile(HOBJECT hObj, const char* pFilename)	
作用	保存运行计划到文件
参数	hObj: 运行计划对象句柄
	pFilename: 文件路径和名称

返回值	0: 无错误 -1: 不正确的对象句柄 -3: 文件操作失败
其它说明	

9、按星期限时播放信息

控制卡支持节目按照时段播放的功能，限时播放信息保存在独立的文件中，文件名称是”playbill.lpt”。文件的格式如下：

文件头
第 1 个记录
...
第 n 个记录

文件头的长度为 7 字节，每个记录的长度也是 7 个字节。

文件里只记录需要限定时间播放的节目的限时信息，总是播放的节目不需要在本文件里记录任何信息。

9.1、文件头详细定义

	0	1	2	3	4	5	6
0x00	文件标识	格式版本号	记录数量	保留			

说明：

数据名称	数据大小(字节)	说明
文件标识	2	固定为“LT”。
格式版本号	2	0x0100(第一字节 0x00,第二字节 0x01)
记录数量	2	限时播放信息的记录数量，低字节在前。

9.2、按星期限时播放记录详细定义

	0	1	2	3	4	5	6
0x00	节目号	星期	开始分	开始时	结束分	结束时	

说明：

数据名称	数据大小(字节)	说明

节目号	2	节目编号，从 0 开始编号
星期	1	按星期限定，使用 7 个位，每个位表示一天，需要在哪天播放将对应的位设置为 1。 星期天：0x01 星期一：0x02 星期二：0x04 星期三：0x08 星期四：0x10 星期五：0x20 星期六：0x40
开始分	1	开始播放的时间分值(0~59)
开始时	1	开始播放的时间时值(0~23)
结束分	1	结束播放的时间分值(0~59)
结束时	1	结束播放的时间时值(0~23)

10、多窗口控制接口

10.1、RS232 多窗口控制接口总览

序号	API 接口函数名	接口说明
1	CP5200_RS232_SplitScreen	发送分窗口命令
2	CP5200_RS232_SendText CP5200_RS232_SendText1	发送文本到指定窗口
3	CP5200_RS232_SendTagText CP5200_RS232_SendTagText1	发送标签文本到指定窗口
4	CP5200_RS232_SendPicture CP5200_RS232_SendSimpleImageData	发送图片到指定窗口 发送简单图像数据到指定窗口
5	CP5200_RS232_SendStatic	发送静态文字图片到指定窗口的指定位置
6	CP5200_RS232_SendClock	发送时钟到指定窗口
7	CP5200_RS232_ExitSplitScreen	退出分窗口命令
8	CP5200_RS232_SaveClearWndData	保存或清楚分窗口信息
9	CP5200_RS232_PlaySelectedPrg CP5200_RS232_PlaySelectedPrg1	选播预存节目
10	CP5200_RS232_SetUserVarData	设置用户变量
11	CP5200_RS232_SetSelectedAndUserVarData	设置选播节目及用户变量数据
12	CP5200_RS232_SetGlobalZone	设置全局显示区域
13	CP5200_RS232_PushUserVarData	推移及用户变量数据

14	CP5200_RS232_TimerCtrl	计时器设置
15	CP5200_RS232_SetZoneAndVariable	设置全局显示区域和变量
16	CP5200_RS232_SendPureText	发送纯文本

本类函数的使用步骤如下：

第1步：初始化串口参数

串口参数初始化只记录下参数信息，不对串口进行实际的操作。

第2步：发送分窗口命令，

若已经确认屏上窗口的划分已经符合要求，则这一步可省掉，否则要发送分窗口命令。

第3步：发送文本到指定窗口，或者发送图片到指定窗口。

备注：本类接口事先不必考虑串口是否已经打开，只要初始化好串口参数即可。

10.2、RS232 多窗口控制接口详细描述

CP5200_RS232_SplitScreen

<pre>int CP5200_RS232_SplitScreen(int nCardID, int nScrWidth, int nScrHeight, int nWndCnt, const int *pWndRects)</pre>	
作用	发送分窗口命令
参数	nCardID: 控制卡 ID
	nScrWidth: 显示屏宽度
	nScrHeight: 显示屏高度
	nWndCnt: 要划分的窗口数量，有效值 1~8。
	pWndRects: 窗口的坐标，每个窗口用 4 个整数表示其“左、上、右、下”的坐标值。与 Windows 的 RECT 数据结构相同。
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据

	-6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误 -9: 窗口数量太大
其它说明	本函数设置了分窗口信息, 并且发送分窗口命令。

CP5200_RS232_SendText

(CP5200_RS232_SendText1)

<pre>int CP5200_RS232_SendText(int nCardID, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment);</pre>	
作用	发送文本到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码 , 本参数只支持文字大小, 不支持多种字体
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口

	-4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	发送文本到指定窗口

CP5200_RS232_SendTagText (CP5200_RS232_SendTagText)

<pre>int CP5200_RS232_SendTagText(int nCardID, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment)</pre>	
作用	发送标签文本到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。支持扩展标记, 详见 1.6、文本扩展标记
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码 , 本参数只支持文字大小, 不支持多种字体
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见 1.5、文字、图片显示效果编码
	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误

	-3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_RS232_SendPicture

<pre>int CP5200_RS232_SendPicture(int nCardID, int nWndNo, int nPosX, int nPosY, int nCx, int nCy, const char *pPictureFile, int nSpeed, int nEffect, int nStayTime, int nPictRef)</pre>	
作用	发送图片到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7。
	nPosX: 图片显示的起点 X 偏移, 相对窗口左边。
	nPosY: 图片显示的起点 Y 偏移, 相对窗口上边。
	nCx: 图片宽度
	nCy: 图片高度
	<p>pPictureFile: 图片文件, 其含义由参数 nPictRef 确定:</p> <p>当 nPictRef 为 0 时: pPictureFile 为电脑存在的实际图片的路径名</p> <p>当 nPictRef 为 1 时: pPictureFile 为在控制卡上预存的 gif 图像文件的文件名。</p> <p>当 nPictRef 为 2 时: pPictureFile 为电脑存在的实际图片的路径名</p> <p>当 nPictRef 为 3 时: pPictureFile 为控制卡上预存的图像包文件名以及图片序号。中间以空格分隔。例如” images.rpk 1”</p>
	<p>nSpeed: 特效展示速度</p> <p>0~100: 取值为 0 时速度最快。</p>
nEffect: 显示效果。参见“1.5”节。	
nStayTime: 停留时间。单位为秒。	

	<p>nPictRef: 图片发送方式及含义</p> <p>0: 显示用户指定的本地图片文件, 将转换成 GIF 格式发送</p> <p>1: 显示控制卡上预存的 GIF 图片文件</p> <p>2: 显示用户指定的本地图片文件, 将转换成简单图片格式发送</p> <p>3: 显示控制上预存的图像包中的图片文件</p> <p>其他值: 当 0 值处理</p>
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p> <p>-9: 窗口参数错误</p> <p>-10: 图片文件不存在</p> <p>-11: 指定的文件不是可支持的图片文件</p>
其它说明	<p>最终图片被转换为 256 色图片发送, 若给出的是真彩色图片, 可能会有颜色的变化。</p> <p>图片大小将被拉伸或压缩以适应指定窗口的大小。</p>

CP5200_RS232_SendSimpleImageData

<pre>int CP5200_RS232_SendSimpleImageData(int nCardID, int nWndNo, int nPosX, int nPosY, const char *pPictureFile, int nSpeed, int nEffect, int nStayTime, BYTE* pPicData , long lPicDataLen)</pre>	
作用	发送简单图像数据到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	nPosX: 图片显示的起点 X 偏移, 相对窗口左边。

	nPosY: 图片显示的起点 Y 偏移, 相对窗口上边。
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	pPicData: 简单图片数据, 格式参见 1.11、简单图片数据格式 。
	lPicDataLen: 图片数据长度
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误 -9: 窗口参数错误 -10: 图片文件不存在 -11: 指定的文件不是可支持的图片文件
其它说明	最终图片被转换为 256 色图片发送, 若给出的是真彩色图片, 可能会有颜色的变化。 图片大小将被拉伸或压缩以适应指定窗口的大小。

CP5200_RS232_SendStatic

<pre>int CP5200_RS232_SendStatic(int nCardID, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nAlignment, int x, int y, int cx, int cy)</pre>	
作用	发送静态文本到指定窗口的指定区域
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。

	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	nAlignment: 文字对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
	x: 显示区域起点 X
	y: 显示区域起点 Y
	cx: 显示区域宽度
	cy: 显示区域高度
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	区域之外的内容保留不变

CP5200_RS232_SendClock

<pre>CP5200_RS232_SendClock(int nCardID, int nWinNo , int nStayTime , int nCalendar , int nFormat , int nContent , int nFont , int nRed , int nGreen , int nBlue , LPCSTR pTxt);</pre>	
作用	发送时钟到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	nStayTime: 停留时间, 单位是秒。

	nCalendar: 历法 0: 公历时间日期 1: 农历时间日期 2: 农历节气 3: 农历时间日期+节气	
	nFormat: 格式 bit 0: 时制(0: 12 小时制; 1: 24 小时制) bit 1: 年份位数(0: 4 位; 1: 2 位) bit 2: 分行(0: 单行; 1: 多行) bit 3~5: 格式控制, 如 2010 年 11 月 12 日星期五 的格式根据取值不同分别表示如下: 0: 2010 年 11 月 12 日 星期五 16:20:30 1: Fri, 12/11/2010 16:20:30 2: 2010-11-12 Fri. 16:20:30 3: Friday, 12 November 2010 16:20:30 4: Fri, Nov 12, 2010 16:20:30 5: Friday, November 12 2010 16:20:30 6: Fri, 11/12/2010 16:20:30 7: 2010/11/12, Fri. 16:20:30 bit 6: 显示时标、分标 bit 7: 透明显示	
	nContent: 内容 按位确定要显示的内容。 bit 7: 指针 bit 6: 星期 bit 5: 秒 bit 4: 分 bit 3: 时 bit 2: 日 bit 1: 月 bit 0: 年	
	nFont: 字体, Bit0~3: 文字大小	
	nRed: 颜色的红色分量	
	nGreen: 颜色的绿色分量	
	nBlue: 颜色的蓝色分量	
	pTxt: 以 0x00 结束的文字串	
	返回值	0: 无错误 -1: 不能生成命令数据

	-2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_RS232_ExitSplitScreen

<code>int CP5200_RS232_ExitSplitScreen(int nCardID);</code>	
作用	退出分窗口命令
参数	nCardID: 控制卡 ID
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_RS232_SaveClearWndData

<code>int CP5200_RS232_SaveClearWndData(int nCardID , int nSavaOrClear);</code>	
作用	保存或清楚分窗口信息
参数	nCardID: 控制卡 ID nSavaOrClear: 保存还是清楚 0: 保存数据到 flash。 1: 清除 flash 保存的数据。

返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CP5200_RS232_PlaySelectedPrg

<pre>int CP5200_RS232_PlaySelectedPrg(int nCardID, const WORD *pSelected, int nSelCnt, int nOption)</pre>	
作用	发送播放指定的节目
参数	nCardID: 控制卡 ID
	pSelected: 指向要选播的节目的指针。
	nSelCnt: 要选播的节目数量。
	nOption: 是否保存选播信息, 0 不保存, 1 保存
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CP5200_RS232_PlaySelectedPrg1

<pre>int CP5200_RS232_PlaySelectedPrg1(int nCardID, const WORD *pSelected, int nSelCnt, int nOption, int nScrWidth, int nScrHeight, byte byColorGray, byte nWndCnt)</pre>	
作用	发送播放指定的节目
参数	nCardID: 控制卡 ID
	pSelected: 指向要选播的节目的指针。
	nSelCnt: 要选播的节目数量。
	nOption: 是否保存选播信息, 0 不保存, 1 保存
	nScrWidth: 屏体宽度
	nScrHeight: 屏体高度
	byColorGray: 颜色灰度
	nWndCnt: 窗口数
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CP5200_RS232_SetUserVarData

<pre>int CP5200_Rs232_SetUserVarData(int nCardID, int bSave, int nVarNum, int bAstride, int* nVarLen, byte* byNoData);</pre>	
作用	设置用户变量值
参数	nCardID: 控制卡 ID
	bSave: Bit0: 是否保存所有变量值到 FLASH 0 不保存, 1 保存。
	Bit1~7: 保留, 设置为 0

	nVarNum: 要设置值的变量数量
	bAstride: 是否允许跨变量区设置。0 不允许; 1 允许
	nWarLen: 按顺序指定每个变量数据的字节
	byNoData: 按顺序指定每个变量的变量号及变量数据, 对于每一变量第一字节是变量号, 后面接着是指定长度的变量数据
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	<p>每个变量号对应一个变量区, 每个变量区的大小是 32 字节, 可将多个连续的变量区连在一起作为一个变量使用, 被占用了变量区的变量号不能再使用。</p> <p>不允许跨变量区时, 超过 32 字节的数据被丢弃; 允许跨变量区时, 根据数据长度计算需要使用的变量区数量</p> <p>变量号有效值是 1~100。每个变量号对应的变量区可存放 32 字节数据, 连续的多个变量区可合起来供一个变量使用, 被占用了变量区的变量号不能再使用。</p> <p>不更新变量值, 只想执行保存变量值到 FLASH 动作时可“数量”的值为 0, 设置“选项”为保存</p>

CP5200_RS232_SetSelectedAndUserVarData

<pre>int CP5200_RS232_SetSelectedAndUserVarData(int nCardID, int bSave, int nVarNum, int bAstride, int* nWarLen, byte* byNoData, int nSelPrg)</pre>	
作用	设置已选择的用户变量值
参数	nCardID: 控制卡 ID
	bSave: Bit0: 是否保存所有变量值到 FLASH 0 不保存, 1 保存。
	Bit1~7: 保留, 设置为 0
	nVarNum: 要设置值的变量数量
	bAstride: 是否允许跨变量区设置。0 不允许; 1 允许

	nWarLen: 按顺序指定每个变量数据的字节
	byNoData: 按顺序指定每个变量的变量号及变量数据, 对于每一变量第一字节是变量号, 后面接着是指定长度的变量数据
	nSelPrg:
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	<p>每个变量号对应一个变量区, 每个变量区的大小是 32 字节, 可将多个连续的变量区连在一起作为一个变量使用, 被占用了变量区的变量号不能再使用。</p> <p>不允许跨变量区时, 超过 32 字节的数据被丢弃; 允许跨变量区时, 根据数据长度计算需要使用的变量区数量</p> <p>变量号有效值是 1~100。每个变量号对应的变量区可存放 32 字节数据, 连续的多个变量区可合起来供一个变量使用, 被占用了变量区的变量号不能再使用。</p> <p>不更新变量值, 只想执行保存变量值到 FLASH 动作时可设置“数量”的值为 0, 设置“选项”为保存</p>

CP5200_RS232_SetGlobalZone

<pre>int CP5200_RS232_SetGlobalZone(int nCardID, byte byConfig, byte bySynchro, byte byZoneNum, byte *byZoneMsg)</pre>	
作用	设置全局显示区域
参数	nCardID: 控制卡 ID
	byConfig: 选项。Bit0: 是否保存设置到 FLASH 0 不保存, 1 保存。 Bit1~7: 保留, 设置为 0
	bySynchro: 同步。Bit0: 同步显示。0 不同步, 1 同步。 Bit1~7: 保留

	byZoneNum: 区域数量。要设置的全局显示区域数量 区域数量为 0 取消所有区域
	byZoneMsg: 区域定义, 大小为区域数量*16 字节。全局显示区域的具体定义见: 1.7、全局区域信息格式
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_RS232_PushUserVarData

int CP5200_RS232_PushUserVarData(int nCardID, byte byOption , byte byVarZoonNum , byte byVarDataLen , byte* pVarNoAndData)	
作用	推移用户变量数据
参数	nCardID: 控制卡 ID
	byOption:选项。Bit0: 是否保存所有变量值到 FLASH 0 不保存, 1 保存。 Bit1: 方向。0 往后推移,1 往前推移 Bit 2~3: 保留, 设置为 0 Bit4~7: 推移量。+1 为推移的变量区数量
	byVarZoonNum:变量区数量。Bit0~6: 要推移的变量数量区数量 1~100 Bit7: 保留, 填 0
	byVarDataLen: 变量数据长度。指定变量数据的字节。变量号及数据一起的长度是(1+n)字节。
	pVarNoAndData: 变量号及数据。第一字节是变量号, 后面接着是指定长度的变量数据
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口

	-4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_RS232_TimerCtrl

<pre>int CP5200_RS232_TimerCtrl(int nCardID, byte byTimerNo , byte byCmd , byte byProp , DWORD dwValue);</pre>	
作用	秒表计时设置
参数	nCardID: 控制卡 ID
	byTimerNo: 计时器选择。按位标识要设置的计时器。位值为 1 有效 Bit0: 计时器 1。 Bit1: 计时器 2。 Bit3: 计时器 3。 Bit4: 计时器 4。 Bit5: 计时器 5。 Bit6: 计时器 6。 Bit7: 计时器 7。
	byCmd: 动作。1: 初始化计时器 2: 复位计时器 3: 启动计时器 4: 暂停计时器 其它: 保留
	byProp:属性。根据动作的不同, 含义不同。 动作为初始化计时器时: Bit0: 0 正计时, 1 倒计时 Bit1: 0 暂停, 1 立即启动 Bit2~3: 保留 Bit4~7: 计时量 动作为其他值时, 保留填 0

	<p>dwValue: 值。根据动作的不同, 含义不同。</p> <p>动作为初始化计时器时:</p> <p>倒计时的初始值, 单位是毫秒</p> <p>高字节在前</p> <p>正计时此值填 0</p> <p>动作为其他值时, 保留填 0</p>
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CP5200_RS232_SetZoneAndVariable

<pre>int CP5200_RS232_SetZoneAndVariable(int nCardID, const BYTE* pZoneData, int nZoneLen, const BYTE* pVariableData, int nVarLen, WORD wCtrl, WORD wReserved)</pre>	
作用	设置全局显示区域及变量数据
参数	nCardID: 控制卡 ID
	pZoneData: 全局区域数据, 包括区域选项、区域数量、区域编号、区域定义
	nZoneLen: 区域数据长度
	pVariableData: 变量数据, 包括变量选项、变量数据及跨区允许、变量数据长度表、变量号及数据
	nVarLen: 变量数据长度
	<p>wCtrl: 有效控制参数</p> <p>播放次数。高字节在前。</p> <p>该值为 0 表示一直有效。</p> <p>Bit15: 保留, 填 0</p> <p>Bit0~14: 显示次数</p>

	wReserved: 保留
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	使用本指令之后, 全局区域自动变为同步显示

CP5200_RS232_SendPureText

<pre>int CP5200_RS232_SendPureText(int nCardID, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment)</pre>	
作用	发送纯文本数据
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	0: 无错误

	-1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_RS232_SendMultiProtocol

<pre>int CP5200_Net_SendMultiProtocol(int nCardID, int nItem, const BYTE *pText, int nLength)</pre>	
作用	发送多协议数据
参数	nCardID: 控制卡 ID
	nItem: 数据项数量
	pText: 数据项数据文本, 参见 《C-Power 控制卡外部调用通信协议》发送多协议数据: CC=0x60
	nLength: 数据项数据长度
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

10.3、Network 多窗口控制接口总览

序号	API 接口函数名	接口说明
1	CP5200_Net_SplitScreen	发送分窗口命令
2	CP5200_Net_SendText CP5200_Net_SendText1	发送文本到指定窗口
3	CP5200_Net_SendTagText CP5200_Net_SendTagText1	发送标签文本到指定窗口
4	CP5200_Net_SendPicture CP5200_Net_SendSimpleImageData	发送图片到指定窗口 发送简单图像数据到指定窗口
5	CP5200_Net_SendStatic	发送静态文字图片到指定窗口的指定位置
6	CP5200_Net_SendClock	发送时钟到指定窗口
7	CP5200_Net_ExitSplitScreen	退出分窗口命令
8	CP5200_Net_SaveClearWndData	保存或清楚分窗口信息
9	CP5200_Net_PlaySelectedPrg CP5200_Net_PlaySelectedPrg1	选播预存节目
10	CP5200_Net_SetUserVarData	设置用户变量
11	CP5200_Net_SetSelectedAndUserVarData	设置选播及用户变量数据
12	CP5200_Net_SetGuvvMsg	设置全局显示信息
13	CP5200_Net_PushUserVarData	推移及用户变量数据
14	CP5200_Net_TimerCtrl	设置计时器
15	CP5200_Net2_SetZoneAndVariable	设置全局显示区域和变量
16	CP5200_Net_SendPureText	发送纯文本

本类函数的使用步骤如下：

第1步：初始化网口参数

网口参数初始化只记录下参数信息，不对网口进行实际的操作。

第2步：发送分窗口命令，

若已经确认屏上窗口的划分已经符合要求，则这一步可省掉，否则要发送分窗口命令。

第3步：发送文本到指定窗口，或者发送图片到指定窗口。

备注：本类接口事先不必考虑网口是否已经打开，只要初始化好网口参数即可。

10.4、Network 多窗口控制接口详细描述

CP5200_Net_SplitScreen

<pre>int CP5200_Net_SplitScreen(int nCardID, int nScrWidth, int nScrHeight, int nWndCnt, const int *pWndRects)</pre>	
作用	发送分窗口命令
参数	nCardID: 控制卡 ID
	nScrWidth: 显示屏宽度
	nScrHeight: 显示屏高度
	nWndCnt: 要划分的窗口数量，有效值 1~8。
	pWndRects: 窗口的坐标，每个窗口用 4 个整数表示其“左、上、右、下”的坐标值。与 Windows 的 RECT 数据结构相同。
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足，或者数据标识错误</p> <p>-7: 数据校验错误</p> <p>-9: 窗口数量太大</p>
其它说明	本函数设置了分窗口信息，并且发送分窗口命令。

CP5200_Net_SendText (CP5200_Net_SendText1)

<pre>int CP5200_Net_SendText(int nCardID, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment);</pre>	
作用	发送文本到指定窗口

参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码 , 本参数只支持文字大小, 不支持多种字体
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	<p>发送文本到指定窗口</p> <p>CP5200_Net_SendText1 接口用于纯粹单字节文字 (英文、欧洲文字) 的发送</p>

CP5200_Net_SendTagText (CP5200_Net_SendTagText1)

```
int CP5200_Net_SendTagText(int nCardID, int nWndNo, const char *pText,
COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime,
```

int nAlignment)	
作用	发送标签文本到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码 , 本参数只支持文字大小, 不支持多种字体
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	发送标签文本到指定窗口

CP5200_Net_SendPicture

```
int CP5200_Net_SendPicture(int nCardID, int nWndNo, int nPosX, int nPosY, int nCx, int
```

nCy, const char *pPictureFile, int nSpeed, int nEffect, int nStayTime, int nPictRef)	
作用	发送图片到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	nPosX: 图片显示的起点 X 偏移, 相对窗口左边。
	nPosY: 图片显示的起点 Y 偏移, 相对窗口上边。
	nCx: 图片宽度
	nCy: 图片高度
	pPictureFile: 图片文件, 其含义由参数 nPictRef 确定: 当 nPictRef 为 0 时: pPictureFile 为电脑存在的实际图片的路径名 当 nPictRef 为 1 时: pPictureFile 为在控制卡上预存的 gif 图像文件的文件名。 当 nPictRef 为 2 时: pPictureFile 为电脑存在的实际图片的路径名 当 nPictRef 为 3 时: pPictureFile 为控制卡上预存的图像包文件名以及图片序号。中间以空格分隔。例如" images.rpk 1"
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	nPictRef: 图片发送方式及含义 0: 显示用户指定的本地图片文件, 将转换成 GIF 格式发送 1: 显示控制卡上预存的 GIF 图片文件 2: 显示用户指定的本地图片文件, 将转换成简单图片格式发送 3: 显示控制上预存的图像包中的图片文件 其他值: 当 0 值处理
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据

	<p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p> <p>-9: 窗口参数错误</p> <p>-10: 图片文件不存在</p> <p>-11: 指定的文件不是可支持的图片文件</p>
其它说明	<p>最终图片被转换为 256 色图片发送, 若给出的是真彩色图片, 可能会有颜色的变化。</p> <p>图片大小将被拉伸或压缩以适应指定窗口的大小。</p>

CP5200_Net_SendSimpleImageData

<pre>int CP5200_Net_SendSimpleImageData(int nCardID, int nWndNo, int nPosX, int nPosY, const char *pPictureFile, int nSpeed, int nEffect, int nStayTime, BYTE* pPicData, long lPicDataLen)</pre>	
作用	发送简单图像数据到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	nPosX: 图片显示的起点 X 偏移, 相对窗口左边。
	nPosY: 图片显示的起点 Y 偏移, 相对窗口上边。
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。
	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	pPicData: 简单图片数据, 格式参见 1.11、简单图片数据格式 。
	lPicDataLen: 图片数据长度
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p>

	<p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p> <p>-9: 窗口参数错误</p> <p>-10: 图片文件不存在</p> <p>-11: 指定的文件不是可支持的图片文件</p>
其它说明	<p>最终图片被转换为 256 色图片发送, 若给出的是真彩色图片, 可能会有颜色的变化。</p> <p>图片大小将被拉伸或压缩以适应指定窗口的大小。</p>

CP5200_Net_SendStatic

<pre>int CP5200_Net_SendStatic(int nCardID, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nAlignment, int x, int y, int cx, int cy)</pre>	
作用	发送静态文本到指定窗口的指定区域
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	nAlignment: 文字对齐方式
	0: 左对齐
	1: 水平居中
	2: 右对齐
	x: 显示区域起点 X
y: 显示区域起点 Y	
cx: 显示区域宽度	
cy: 显示区域高度	
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p>

	-2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	区域之外的内容保留不变

CP5200_Net_SendClock

<pre>CP5200_Net_SendClock(int nCardID, int nWinNo , int nStayTime , int nCalendar , int nFormat , int nContent , int nFont , int nRed , int nGreen , int nBlue , LPCSTR pTxt);</pre>	
作用	发送时钟到指定窗口
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	nStayTime: 停留时间, 单位是秒。
	nCalendar: 历法 0: 公历时间日期 1: 农历时间日期 2: 农历节气 3: 农历时间日期+节气

	<p>nFormat: 格式</p> <p>bit 0: 时制(0: 12 小时制; 1: 24 小时制)</p> <p>bit 1: 年份位数(0: 4 位; 1: 2 位)</p> <p>bit 2: 分行(0: 单行; 1: 多行)</p> <p>bit 3~5: 格式控制, 如 2010 年 11 月 12 日星期五 的格式根据取值不同分别表示如下:</p> <p>0: 2010 年 11 月 12 日 星期五 16:20:30</p> <p>1: Fri, 12/11/2010 16:20:30</p> <p>2: 2010-11-12 Fri. 16:20:30</p> <p>3: Friday, 12 November 2010 16:20:30</p> <p>4: Fri, Nov 12, 2010 16:20:30</p> <p>5: Friday, November 12 2010 16:20:30</p> <p>6: Fri, 11/12/2010 16:20:30</p> <p>7: 2010/11/12, Fri. 16:20:30</p> <p>bit 6: 显示时标、分标</p> <p>bit 7: 透明显示</p>
	<p>nContent: 内容</p> <p>按位确定要显示的内容。</p> <p>bit 7: 指针</p> <p>bit 6: 星期</p> <p>bit 5: 秒</p> <p>bit 4: 分</p> <p>bit 3: 时</p> <p>bit 2: 日</p> <p>bit 1: 月</p> <p>bit 0: 年</p>
	nFont: 字体, Bit0~3: 文字大小
	nRed: 颜色的红色分量
	nGreen: 颜色的绿色分量
	nBlue: 颜色的蓝色分量
	pTxt: 以 0x00 结束的文字串
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p>

	-5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_Net_ExitSplitScreen

<code>int CP5200_Net_ExitSplitScreen(int nCardID);</code>	
作用	退出分窗口命令
参数	nCardID: 控制卡 ID
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_Net_SaveClearWndData

<code>int CP5200_Net_SaveClearWndData(int nCardID , int nSavaOrClear);</code>	
作用	保存或清楚分窗口信息
参数	nCardID: 控制卡 ID
	nSavaOrClear: 保存还是清楚 0: 保存数据到 flash。 1: 清除 flash 保存的数据。
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误

	-3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_Net_PlaySelectedPrg

<pre>int CP5200_Net_PlaySelectedPrg(int nCardID, const WORD *pSelected, int nSelCnt, int nOption)</pre>	
作用	发送播放指定的节目
参数	nCardID: 控制卡 ID
	pSelected: 指向要选播的节目的指针。
	nSelCnt: 要选播的节目数量。
	nOption: 是否保存选播信息, 0 不保存, 1 保存
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_Net_PlaySelectedPrg1

<pre>int CP5200_Net_PlaySelectedPrg1(int nCardID, const WORD *pSelected, int nSelCnt, int nOption, int nScrWidth, int nScrHeight, byte byColorGray, byte nWndCnt)</pre>

作用	发送播放指定的节目
参数	nCardID: 控制卡 ID
	pSelected: 指向要选播的节目的指针。
	nSelCnt: 要选播的节目数量。
	nOption: 是否保存选播信息, 0 不保存, 1 保存
	nScrWidth: 屏体宽度
	nScrHeight: 屏体高度
	byColorGray: 颜色灰度
	nWndCnt: 窗口数
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CP5200_Net_SetUserVarData

<pre>int CP5200_Net_SetUserVarData(int nCardID, int bSave, int nVarNum, int bAstride, int* nWarLen, byte* byNoData);</pre>	
作用	设置用户变量值
参数	nCardID: 控制卡 ID
	bSave: Bit0: 是否保存所有变量值到 FLASH 0 不保存, 1 保存。 Bit1~7: 保留, 设置为 0
	nVarNum: 要设置值的变量数量
	bAstride: 是否允许跨变量区设置。0 不允许; 1 允许
	nWarLen: 按顺序指定每个变量数据的字节
	byNoData: 按顺序指定每个变量的变量号及变量数据, 对于每一变量第一字节是变量号, 后面接着是指定长度的变量数据

返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	<p>每个变量号对应一个变量区, 每个变量区的大小是 32 字节, 可将多个连续的变量区连在一起作为一个变量使用, 被占用了变量区的变量号不能再使用。</p> <p>不允许跨变量区时, 超过 32 字节的数据被丢弃; 允许跨变量区时, 根据数据长度计算需要使用的变量区数量</p> <p>变量号有效值是 1~100。每个变量号对应的变量区可存放 32 字节数据, 连续的多个变量区可合起来供一个变量使用, 被占用了变量区的变量号不能再使用。</p> <p>不更新变量值, 只想执行保存变量值到 FLASH 动作时可“数量”的值为 0, 设置“选项”为保存</p>

CP5200_Net_SetSelectedAndUserVarData

<pre>int CP5200_Net_SetSelectedAndUserVarData(int nCardID, int bSave , int nVarNum , int bAstride , int* nWarLen , byte* byNoData, int nSelPrg)</pre>	
作用	设置选播单个节目及变量值
参数	<p>nCardID: 控制卡 ID</p> <p>bSave:选项。Bit0: 是否保存节目号到 FLASH 0 不保存, 1 保存。</p> <p>Bit1: 是否保存全部变量值到 FLASH 0 不保存, 1 保存。</p> <p>Bit2: 是否保存先清除原来的所有变量值 0 不清除, 1 清除。</p> <p>Bit3~7: 保留, 设置为 0</p> <p>nVarNum: 变量数量</p>

	<p>bAstride: 数量及跨区域允许。Bit0~6: 要设置值的变量数量 Bit7: 是否允许跨变量区设置。0 不允许; 1 允许 每个变量号对应一个变量区, 每个变量区的大小是 32 字节, 可将多个连续的变量区连在一起作为一个变量使用, 被占用了变量区的变量号不能再使用。 不允许跨变量区时, 超过 32 字节的数据被丢弃; 允许跨变量区时, 根据数据长度计算需要使用的变量区数量</p>
	<p>nWarLen: 变量数据长度。按书序指定每个变量数据的字节。变量号及数据一起的长度是(1+n)字节。</p>
	<p>byNoData: 变量号及数据。第一字节是变量号, 后面接着是指定长度的变量数据</p>
	<p>nSelPrg: 节目号。要选播的节目号列表。每个节目号 2 个字节, 高字节在前 超出预存节目数量的节目号被忽略</p>
返回值	<p>0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误</p>
其它说明	<p>变量号有效值是 1~100。每个变量号对应的变量区可存放 32 字节数据, 连续的多个变量区可合起来供一个变量使用, 被占用了变量区的变量号不能再使用。 不更新变量值, 只想执行保存变量值到 FLASH 动作时可设置“数量”的值为 0, 设置“选项”为保存。</p>

CP5200_Net_SetGlobalZone

<pre>int CP5200_Net_SetGlobalZone(int nCardID, byte byConfig , byte bySynchro , byte byZoneNum , byte *byZoneMsg);</pre>	
作用	设置全局显示区域
参数	nCardID: 控制卡 ID

	<p>byConfig:选项。Bit0: 是否保存设置到 FLASH 0 不保存, 1 保存。 Bit1~7: 保留, 设置为 0</p>
	<p>bySynchro:同步。Bit0: 同步显示。0 不同步, 1 同步。 Bit1~7: 保留</p>
	<p>byZoneNum: 区域数量。要设置的全局显示区域数量 区域数量为 0 取消所有区域</p>
	<p>byZoneMsg:区域定义, 大小为区域数量*16 字节。全局显示区域的具体定义见: 1.7、全局区域信息格式</p>
返回值	<p>0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误</p>
其它说明	

CP5200_Net_PushUserVarData

<pre>int CP5200_Net_PushUserVarData(int nCardID, byte byOption , byte byVarZoonNum , byte byVarDataLen , byte* pVarNoAndData)</pre>	
作用	推移及用户变量值数据
参数	nCardID: 控制卡 ID
	<p>byOption:选项。Bit0: 是否保存所有变量值到 FLASH 0 不保存, 1 保存。 Bit1: 方向。0 往后推移,1 往前推移 Bit 2~3: 保留, 设置为 0 Bit4~7: 推移量。+1 为推移的变量区数量</p>
	<p>byVarZoonNum:变量区数量。Bit0~6: 要推移的变量数量区数量 1~100 Bit7: 保留, 填 0</p>

	byVarDataLen: 变量数据长度。指定变量数据的字节。变量号及数据一起的长度是(1+n)字节。
	pVarNoAndData: 变量号及数据。第一字节是变量号, 后面接着是指定长度的变量数据
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CP5200_Net_TimerCtrl

int CP5200_Net_TimerCtrl(int nCardID, byte byTimerNo, byte byCmd, byte byProp, DWORD dwValue)	
作用	设置计时器控制数据
参数	nCardID: 控制卡 ID
	byTimerNo: 计时器编号。按位标识要设置的计时器。位值为 1 有效 Bit0: 计时器 1。 Bit1: 计时器 2。 Bit3: 计时器 3。 Bit4: 计时器 4。 Bit5: 计时器 5。 Bit6: 计时器 6。 Bit7: 计时器 7。
	byCmd: 动作。1: 初始化计时器 2: 复位计时器 3: 启动计时器 4: 暂停计时器 其它: 保留

	<p>byProp:属性。根据动作的不同,含义不同。</p> <p>动作为初始化计时器时:</p> <p>Bit0: 0 正计时, 1 倒计时</p> <p>Bit1: 0 暂停, 1 立即启动</p> <p>Bit2~3: 保留</p> <p>Bit4~7: 计时量</p> <p>动作为其他值时, 保留填 0</p>
	<p>byValue:值。根据动作的不同,含义不同。</p> <p>动作为初始化计时器时:</p> <p>倒计时的初始值, 单位是毫秒</p> <p>高字节在前</p> <p>正计时此值填 0</p> <p>动作为其他值时, 保留填 0</p>
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CP5200_Net_SetZoneAndVariable

<pre>int CP5200_Net_SetZoneAndVariable(int nCardID, const BYTE* pZoneData, int nZoneLen, const BYTE* pVariableData, int nVarLen, WORD wCtrl, WORD wReserved)</pre>	
作用	设置全局显示区域及变量数据
参数	nCardID: 控制卡 ID
	pZoneData: 全局区域数据, 包括区域选项、区域数量、区域编号、区域定义
	nZoneLen: 区域数据长度

	pVariableData: 变量数据, 包括变量选项、变量数据及跨区允许、变量数据长度表、变量号及数据
	nVarLen: 变量数据长度
	wCtrl: 有效控制参数 播放次数。高字节在前。 该值为 0 表示一直有效。 Bit15: 保留, 填 0 Bit0~14: 显示次数
	wReserved: 保留
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	使用本指令之后, 全局区域自动变为同步显示

CP5200_Net_SendPureText

<pre>int CP5200_Net_SendPureText(int nCardID, int nWndNo, const char *pText, COLORREF crColor, int nFontSize, int nSpeed, int nEffect, int nStayTime, int nAlignment)</pre>	
作用	发送纯文本数据
参数	nCardID: 控制卡 ID
	nWndNo: 窗口的顺序号, 有效值 0~7
	pText: 文本字符串。
	crColor: 文本颜色。
	nFontSize: 文字大小及类型, 参见 1.7、文字大小及类型编码
	nSpeed: 特效展示速度 0~100: 取值为 0 时速度最快。

	nEffect: 显示效果。参见“1.5”节。
	nStayTime: 停留时间。单位为秒。
	nAlignment: 对齐方式 0: 左对齐 1: 水平居中 2: 右对齐
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足，或者数据标识错误 -7: 数据校验错误
其它说明	

CP5200_Net_SendMultiProtocol

int CP5200_Net_SendMultiProtocol(int nCardID, int nItem, const BYTE *pText, int nLength)	
作用	发送多协议数据
参数	nCardID: 控制卡 ID
	nItem: 数据项数量
	pText: 数据项数据文本，参见 《C-Power 控制卡外部调用通信协议》发送多协议数据：CC=0x60
	nLength: 数据项数据长度
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误

	-5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

11、节目模板接口

11.1、RS232 节目模板接口总览

序号	API 接口函数名	接口说明
1	CPowerBox_RS232_SetProgramTemplate CPowerBox_RS232_SetProgramTemplatel	设置节目模板
2	CPowerBox_RS232_InOutProgramTemplate	进入或退出模板方式
3	CPowerBox_RS232_QueryProgramTemplate CPowerBox_RS232_QueryProgramTemplatel	查询节目模板方式
4	CPowerBox_RS232_DeleteProgram	删除节目
5	CPowerBox_RS232_SendText	发送文本到指定节目的指定窗口
6	CPowerBox_RS232_SendPicture	发送图片到指定节目的指定窗口
7	CPowerBox_RS232_SendClockOrTemperature	发送时钟或温度到指定的窗口
8	CPowerBox_RS232_SetAloneProgram	设置独立节目
9	CPowerBox_RS232_QueryProgram	查询节目信息
10	CPowerBox_RS232_SetProgramProperty	设置节目属性
11	CPowerBox_RS232_SetSchedule	设置播放计划
12	CPowerBox_RS232_DeleteSchedule	删除播放计划
13	CPowerBox_RS232_GetSchedule	查询播放计划

本类函数的使用步骤如下:

第1步: 初始化串口参数

串口参数初始化只记录下参数信息, 不对串口进行实际的操作。

第2步: 设置节目模板,

第3步: 发送文本到指定窗口, 或者发送图片到指定窗口。

备注: 本类接口事先不必考虑串口是否已经打开, 只要初始化好串口参数即可。

11.2、RS232 节目模板接口详细描述

CPowerBox_RS232_SetProgramTemplate

<pre>int CPowerBox_RS232_SetProgramTemplate(int nCardID, byte byColor, USHORT nWidth, USHORT nHeight, byte nWndNum, byte *byDefParam, byte* pWndParam)</pre>	
作用	设置节目模板
参数	nCardID: 卡 ID
	byColor: Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~6: 灰度级别 0: 2 级灰度, 7: 256 级灰度 Bit7: 保留
	nWidth: 屏宽, 高字节在前
	nHeight: 屏高, 高字节在前
	nWndNum: 播放窗口数量, 最多支持 10 个窗口
	byDefParam: 缺省参数。Byte0~1: 停留时间, 单位是秒。高字节在前 Byte2: 速度。该值越小, 移动越快 Byte3: 文字大小。参见 1.7、文字大小编码 Byte4: 文字颜色。参见单字节 1.8、文字颜色编码 Byte5: 显示特效。参见 1.5、文字、图片显示效果编码 Byte6: 图片方式。参见 1.9、图片方式编码 Byte7: 时钟格式。参见 1.10、时钟格式和显示内容 中的”时钟格式” Byte8: 时钟内容。参见 1.10、时钟格式和显示内容 中的”时钟显示内容
	pWndParam: 窗口参数。每个窗口 16 字节的参数。 数据总长度是: 窗口数量*16。 详见 1.13、窗口位置及属性
	返回值

	-5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_RS232_SetProgramTemplate1

<pre>int CPowerBox_RS232_SetProgramTemplate1 (int nCardID, BYTE byColor , USHORT nWidth , USHORT nHeight , BYTE nWndNum , BYTE byOption, BYTE* pDefParam , BYTE* pWndParam)</pre>	
作用	设置节目模板
参数	nCardID: 卡 ID
	byColor: Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~6: 灰度级别 0: 2 级灰度, 7: 256 级灰度 Bit7: 保留
	nWidth: 屏宽, 高字节在前
	nHeight: 屏高, 高字节在前
	nWndNum: 播放窗口数量, 最多支持 10 个窗口
	byOption: 选项: Bit0: 强制进入节目模板方式运行 Bit1: 模板保存位置。0 用户盘, 1 系统盘。 若模板保存到系统盘, 则用户盘原有的模板被清除; 若模板保存到用户盘, 则系统盘原有的模板被清除。 Bit2~7: 保留
	byDefParam: 缺省参数。Byte0~1: 停留时间, 单位是秒。高字节在前 Byte2: 速度。该值越小, 移动越快 Byte3: 文字大小。参见 1.7、文字大小编码 Byte4: 文字颜色。参见单字节 1.8、文字颜色编码 Byte5: 显示特效。参见 1.5、文字、图片显示效果编码 Byte6: 图片方式。参见 1.9、图片方式编码 Byte7: 时钟格式。参见 1.10、时钟格式和显示内容 中的”时钟格式” Byte8: 时钟内容。参见 1.10、时钟格式和显示内容 中的”时钟显示内容”

	<p>pWndParam: 窗口参数。每个窗口 16 字节的参数。 数据总长度是：窗口数量*16。 详见 1.13、窗口位置及属性</p>
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足，或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CPowerBox_RS232_InOutProgramTemplate

<pre>int CPowerBox_RS232_InOutProgramTemplate(int nCardID,byte byInOrOut)</pre>	
作用	设置进入或退出模板方式
参数	nCardID: 卡 ID
	byInOrOut: 进入或退出。 1: 进入节目模板方式 0: 退出节目模板方式
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足，或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CPowerBox_RS232_QueryProgramTemplate

<code>int CPowerBox_RS232_QueryProgramTemplate(int nCardID , byte* pState);</code>	
作用	查询节目模板方式
参数	nCardID: 卡 ID
	pState: 返回数据, 当前是否为模板方式, 1 是, 0 不是
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_RS232_QueryProgramTemplate1

<code>int CPowerBox_RS232_QueryProgramTemplate1(int nCardID , byte byFlag , BYTE* pStateBuf , int nBufSize)</code>	
作用	查询节目模板
参数	nCardID: 卡 ID
	byFlag: 选项 Bit0: 是否查询模板状态 Bit1: 返回模板定义的颜色灰度、的屏大小信息 Bit1~7: 保留
	pStateBuf: 结果缓冲区
	nBufSize: 结果缓冲区的大小, 不小于 10 字节
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

pStateBuf 数据含义如下:

数据项	数值	长度(字节)	意义描述
CC	0x83	1	说明本数据包是查询节目模板方式相关参数
选项		1	同发送值
模板方式		1	0: 非节目模板方式 1: 节目模板方式
模板状态		1	Bit0~1: 模板可用性 0: 模板不可用 1: 模板可用 其它: 保留 Bit2~7: 保留
颜色灰度		1	颜色及灰度。同“设置节目模板”的定义
屏宽度		2	高字节在前
屏高度		2	高字节在前
窗口数		1	播放窗口数量。最多支持 10 个播放窗口

CPowerBox_RS232_DeleteProgram

<pre>int CPowerBox_RS232_DeleteProgram(int nCardID,byte byConfig , byte byProNum , byte* pDelPro);</pre>	
作用	删除节目
参数	nCardID: 卡 ID
	byConfig: Bit0: 删除的节目范围 0: 删除全部 1: 删除指定节目 其它: 保留
	byProNum: 节目数量。删除全部节目不需要此数据项
	pDelPro: 要删除的节目列表
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误

其它说明	
------	--

CPowerBox_RS232_SendText

<pre>int CPowerBox_RS232_SendText(int nCardID, DWORD dwAppendCode , byte byProNo , byte byWndNo , byte byProp , byte *byShowFormat , char* pText);</pre>	
作用	发送指定文本到指定节目的指定窗口
参数	nCardID: 卡 ID
	dwAppendCode:用户给出的附加码，高字节在前。
	byProNo: 节目号，有效值 1~255
	byWndNo:窗口号。有效值 1~10，超出节目模板定义时无效
	byProp:属性，Bit0~3: 文本类型 0: 普通文本 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留
	byShowFormat: 显示格式。 当”属性”的显示格式为 0 时不需要此项数据 Byte0~1: 停留时间。高字节在前 Byte2: 速度。该值越小，速度越快 Byte3: 文字大小。参见 1.7、文字大小编码 Byte4: 文字颜色。参见单字节 1.8、文字颜色编码 Byte5: 显示特效。参见 1.5、文字、图片显示效果编码 Byte6: 保留 Byte7: 保留
	pText:文字数据。文本内容，以 0x00 结束。
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足，或者数据标识错误</p> <p>-7: 数据校验错误</p>

其它说明	
------	--

CPowerBox_RS232_SendPicture

<pre>int CPowerBox_RS232_SendPicture(int nCardID, DWORD dwAppendCode , byte byProNo , byte byWndNo , byte byPicType , byte *byShowFormat , byte* pPicData , long lPicDataLen);</pre>	
作用	发送图片到指定节目的指定窗口
参数	<p>nCardID: 卡 ID</p> <p>dwAppendCode:用户给出的附加码，高字节在前。</p> <p>byProNo: 节目号，有效值 1~255</p> <p>byWndNo:窗口号。有效值 1~10，超出节目模板定义时无效</p> <p>byPicType: 图片类型。Bit0~3: 图片类型 1: GIF 图像文件的实际数据，里面包含图像的宽度、高度等信息； 2: 在控制卡上预存的 GIF 图像文件的文件名。 4: 简单图片数据，格式见 1.11、简单图片数据格式 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留</p> <p>byShowFormat: 显示格式。 当”属性”的显示格式为 0 时不需要此项数据 Byte0~1: 停留时间高字节在前 Byte2: 速度, 该值越小，移动越快 Byte3: 显示特效,参见 1.5、文字、图片显示效果编码 Byte4: 图片方式(缩放、拉伸等)，参加 1.9、图片方式编码 Byte5: 保留 Byte6: 保留 Byte7: 保留</p> <p>pPicData:图片数据。</p> <p>lPicDataLen:图片长度。</p>
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p>

	-6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_RS232_SendClockOrTemperature

<pre>CPowerBox_RS232_SendClockOrTemperature(int nCardID, DWORD dwAppendCode , BYTE byProNo , BYTE byWndNo , BYTE byProgramType , UINT nPropLen , BYTE* pProgramProp , byte* pBuf , int nBufSize)</pre>	
作用	发送时钟和温度
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码, 高字节在前。
	byProNo: 节目号, 有效值 1~255
	byWndNo: 窗口号。有效值 1~10, 超出节目模板定义时无效
	byProgramType: 节目类型。 Bit0~3: 类型。2 时钟; 3 温度, 其它保留。 Bit4: 显示格式。0 缺省格式, 1 指定格式 Bit5~7: 保留备用。填 0
	nPropLen: 属性长度
	pProgramProp: 节目属性 属性数据含义根据类型不同而不同 类型=2: 同 1.13 的“时钟日历”属性 类型=3: 同 1.13 的“温度、湿度”属性
	pBuf: 结果数据
nBufSize: 结果缓冲区大小	
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误

	-7: 数据校验错误
其它说明	

pBuf 数据含义如下:

数据项	数值	长度(字节)	意义描述
CC	0x83	1	说明本数据包是查询节目模板方式相关参数
用户附加码		4	用户给出的附加码。高字节在前
节目号		1	同发送值。有效值 1~100。
窗口号		1	同发送值。有效值 1~10, 超出节目模板定义时无效
丢包数		1	尚未接收到的包的数量。发送第一包时的丢包数量是总包数减去 1。
丢包的包号		变长	丢包的包号。总是按照从小到大排列; 第一包的包号是 0。每个包号一个字节。

CPowerBox_RS232_SetAloneProgram

<pre>int CPowerBox_RS232_SetAloneProgram(int nCardID, DWORD dwAppendCode , BYTE byProgramNo , BYTE byWindowCnt , BYTE* pWndParam, BYTE* pWndData)</pre>	
作用	设置独立节目
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码, 高字节在前。
	byProgramNo: 节目号, 有效值 1~100
	byWindowCnt: 窗口数。有效值 1~10,
	pWndParam: 窗口参数 每个窗口的信息表 22 字节, 前 16 字为窗口位置及属性 (参见 1.13、窗口位置及属性), 17~19 字节为窗口数据偏移, 20~22 字节为窗口数据长度。高字节在前。无数据则偏移和长度均为 0。
pWndData: 窗口数据 窗口要播放的数据, 如“文本”、“图片”等。 第 1 字节: 数据类型(1 文本; 4 图片) 第 2 字节: 数据格式 (同发文本或发图片的定义) 第 3 字节起: 文本或图片数据。	
返回值	0: 无错误

	-1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_RS232_QueryProgram

<pre>Int CPowerBox_RS232_QueryProgram(int nCardID ,byte byFlag , byte* pParam , BYTE* pBuf , int nBufSize);</pre>	
作用	查询节目信息
参数	nCardId: 控制卡 ID
	byFlag: 查询标记, 指定要查询哪一项节目信息 0: 保留 1: 查询有效节目数量及节目号 2: 查询指定节目信息 其它: 保留
	pParam: 当查询标记为 1 时: byte1~5, 保留, 填 0 当查询标记为 2 时: byte1, 节目号; byte2~5, 保留, 填 0
	pBuf: 结果缓冲区
	nBufSize: 结果缓冲区大小
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据

	-6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

pBuf 中数据含义如下:

- 查询“有效节目数量及节目号”

返回数据

数据项	数值	长度(字节)	意义描述
CC	0x89	1	说明本数据包是查询节目信息的返回包
信息标识		1	同发送值。
参数		5	同发送值。
有效节目数		1	有效节目的数量
有效节目号		变长	每一个字节标识一个有效的节目号。有效值 1~100。

*返回数据包中的“返回值”非零时的含义:

0x01 控制卡没有运行在节目模板方式

0x10 不认识的信息标识

- 查询指定节目信息

返回数据

数据项	数值	长度(字节)	意义描述
CC	0x89	1	说明本数据包是查询节目信息的返回包
信息标识		1	同发送值。
参数		5	同发送值。
信息数		1	现在仅返回 1 项信息
节目号		1	节目号
用户附加码		4	用户发送节目信息时指定的用户附加码

*返回数据包中的“返回值”非零时的含义:

0x01 控制卡没有运行在节目模板方式

0x10 不认识的信息标识

0x11 无效的节目

0x12 不能取得需要的信息

CPowerBox_RS232_SetProgramProperty

int CPowerBox_RS232_SetProgramProperty(int nCardID, BYTE byOption, BYTE byProgramCnt, BYTE* pPrograms, BYTE byPropertyID1, BYTE byPropertyID2, BYTE byProgramLevel, USHORT nLoopCnt, USHORT nTime, BYTE* pDuetime, BYTE* pTimeInterval)	
作用	设置节目属性
参数	nCardID:卡 ID

	<p>byOption:选项, Bit0: 设置节目属性的范围 0: 全部节目 1: 指定节目 其它: 保留</p>
	byProgramCnt:要生成的节目数量
	pPrograms:节目列表
	<p>ByPropertyID1: 属性标示 1, 按位标识要设置节目的哪些属性, 标识为 0 则该项数据不存在。 Bit0: 节目级别。节目的重要级别 Bit1: 循环次数。节目每次播放时循环的次数 Bit2: 有效时长。从当前时间开始多长时间内有效播放 Bit3: 有效期限 Bit4~7: 保留</p>
	<p>ByPropertyID2: 属性标识 2。Bit0~4: 有效时段。>0 时为有效时段数.<=4 Bit5~7: 保留</p>
	byProgramLevel: 节目级别。1~3 级。优先播放级别高的节目
	<p>nLoopCnt:循环次数。高字节在前。 0:不播放该节目, 用于临时屏蔽节目 1~255:该节目每次播放时循环的次数</p>
	<p>nTime: 有效时长。高字节在前。单位为分钟 0: 不限定播放时间 >0: 指定播放时间。单位是分钟。</p>
	pDuetime:有效时限。
	<p>pTimeInterval:有效时段。时段开始的“时分秒”和结束的“时分秒” 时分秒各用一个字节表示</p>
返回值	<p>0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误</p>

其它说明	
------	--

CPowerBox_RS232_SetSchedule

<pre>int CPowerBox_RS232_SetSchedule(int nCardID, DWORD dwAppendCode, BYTE byScheduleNo, const BYTE* pProperty, const BYTE* pBoxes, BYTE byBoxCnt)</pre>	
作用	设置播放计划
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码，高字节在前。
	byScheduleNo: 计划项编号，有效值 1~100。总共支持 100 个计划项，同一个项的新数据覆盖旧数据。
	pProperty: 播放属性，共 14 字节： byte 0: 格式及级别： Bit0~3: 表示数据格式，填 0x01 Bit4~7: 表示优先级别。优先级别数值越大，则越优先播放，0 为最低优先级。
	byte 1: 星期: Bit0~6: 7 个位分别标识星期日到星期六
	byte 2~4: 开始日期，3 个字节分别表示“年”“月”“日”。年份的值取 0~99, 表示范围是 2000~2099
	byte 5~7: 结束日期, 3 个字节分别表示“年”“月”“日”。年份的值取 0~99, 表示范围是 2000~2099
byte 8~10: 开始时间, 3 个字节分别表示“时”“分”“秒”	
byte 11~13: 结束时间, 3 个字节分别表示“时”“分”“秒”	
pBoxes: 节目编号，每个字节表示 1 个编号。编号从小到大排列，不重复。	
byBoxCnt: 节目编号数量，有效值 1~100	
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能打开串口</p>

	-4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误-1: 不正确的数据对象句柄
其它说明	

CPowerBox_RS232_DeleteSchedule

<pre>int CPowerBox_RS232_DeleteSchedule(int nCardID, DWORD dwAppendCode, const BYTE* pSchs, BYTE bySchCnt)</pre>	
作用	删除播放计划
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码, 高字节在前。
	pSchs: 计划项编号, 编号有效值 1~100。每个字节表示 1 个编号。 删除全部项时, 本项数据长度为 1 字节, 值是 0xff, 作为删除的确认标记。
	bySchCnt: 要删除的计划项的数量。0 表示删除全部
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误-1: 不正确的数据对象句柄
其它说明	

CPowerBox_RS232_GetSchedule

<pre>int CPowerBox_RS232_GetSchedule(int nCardID, DWORD dwAppendCode, BYTE byType, BYTE</pre>

byScheduleNo , byte* pBuf , int nBufSize)	
作用	查询播放计划
参数	nCardID:卡 ID
	dwAppendCode:用户给出的附加码, 高字节在前。
	byType: 0: 查询有效的计划项编号 1: 查询指定编号的计划项内容 其它: 保留
	byScheduleNo: 编号有效值 1~100。 当查询类型为 0 时, 本项数据填 0。
	pBuf:结果数据缓冲区 nBufSize: 结果数据缓冲区大小
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能打开串口 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误-1: 不正确的数据对象句柄
其它说明	

pBuf 中数据格式如下:

数据项	数值	长度(字节)	意义描述
CC	0x8d	1	说明本数据包是查询节目模板方式相关参数
用户附加码		4	同发送数据
查询类型		1	同发送值
数量/编号		1	查询类型为 0, 该值是有效计划项数量 查询类型为 1, 该值是计划项编号
标号表/计划项内容		变长	查询类型为 0, 是有效计划项的编号表 查询类型为 1, 是计划项内容, 数据格式同“设置播放计划”的数据格式

要区分查询类型来处理返回数据

返回数据包中的“返回值”非零时的含义:

- 0x01 节目模板无效
- 0x11 不支持的查询类型
- 0x12 指定的计划项编号超出有效范围
- 0x80 当前不是节目模板方式

11.3、Network 节目模板接口总览

序号	API 接口函数名	接口说明
1	CPowerBox_Net_SetProgramTemplate CPowerBox_Net_SetProgramTemplatel	设置节目模板
2	CPowerBox_Net_InOutProgramTemplate	进入或退出模板方式
3	CPowerBox_Net_QueryProgramTemplate CPowerBox_Net_QueryProgramTemplatel	查询节目模板方式
4	CPowerBox_Net_DeleteProgram	删除节目
5	CPowerBox_Net_SendText	发送文本到指定节目的指定窗口
6	CPowerBox_Net_SendPicture	发送图片到指定节目的指定窗口
7	CPowerBox_Net_SendClockOrTemperature	发送时钟或温度到指定的窗口
8	CPowerBox_Net_SetAloneProgram	设置独立节目
9	CPowerBox_Net_QueryProgram	查询节目信息
10	CPowerBox_Net_SetProgramProperty	设置节目属性
11	CPowerBox_Net_SetSchedule	设置播放计划
12	CPowerBox_Net_DeleteSchedule	删除播放计划
13	CPowerBox_Net_GetSchedule	查询播放计划

本类函数的使用步骤如下：

第1步：初始化网口参数

网口参数初始化只记录下参数信息，不对网口进行实际的操作。

第2步：设置节目模板，

第3步：发送文本到指定窗口，或者发送图片到指定窗口。

备注：本类接口事先不必考虑网口是否已经打开，只要初始化好网口参数即可。

11.4、Network 节目模板接口详细描述

CPowerBox_Net_SetProgramTemplate

<pre>int CPowerBox_Net_SetProgramTemplate(int nCardID, byte byColor, USHORT nWidth, USHORT nHeight, byte nWndNum, byte *byDefParam, byte* pWndParam)</pre>	
作用	设置节目模板
参数	nCardID: 卡 ID

	<p>byColor: Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~6: 灰度级别 0: 2 级灰度, 7: 256 级灰度 Bit7: 保留</p> <p>nWidth: 屏宽, 高字节在前</p> <p>nHeight: 屏高, 高字节在前</p> <p>nWndNum: 播放窗口数量, 最多支持 10 个窗口</p> <p>byDefParam: 缺省参数。Byte0~1: 停留时间, 单位是秒。高字节在前 Byte2: 速度。该值越小, 移动越快 Byte3: 文字大小。参见 1.7、文字大小编码 Byte4: 文字颜色。参见单字节 1.8、文字颜色编码 Byte5: 显示特效。参见 1.5、文字、图片显示效果编码 Byte6: 图片方式。参见 1.9、图片方式编码 Byte7: 时钟格式。参见 1.10、时钟格式和显示内容中的”时钟格式” Byte8: 时钟内容。参见 1.10、时钟格式和显示内容中的”时钟显示内容</p> <p>pWndParam: 窗口参数。每个窗口 16 字节的参数。 数据总长度是: 窗口数量*16。 详见 1.13、窗口位置及属性</p>
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CPowerBox_Net_SetProgramTemplate1

```
int CPowerBox_Net_SetProgramTemplate1(int nCardID, BYTE byColor, USHORT nWidth, USHORT nHeight, BYTE nWndNum, BYTE byOption, BYTE* pDefParam, BYTE* pWndParam)
```

作用	设置节目模板
参数	nCardID: 卡 ID
	byColor: Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~6: 灰度级别 0: 2 级灰度, 7: 256 级灰度 Bit7: 保留
	nWidth: 屏宽, 高字节在前
	nHeight: 屏高, 高字节在前
	nWndNum: 播放窗口数量, 最多支持 10 个窗口
	byOption: 选项: Bit0: 强制进入节目模板方式运行 Bit1: 模板保存位置。0 用户盘, 1 系统盘。 若模板保存到系统盘, 则用户盘原有的模板被清除; 若模板保存到用户盘, 则系统盘原有的模板被清除。 Bit2~7: 保留
	byDefParam: 缺省参数。Byte0~1: 停留时间, 单位是秒。高字节在前 Byte2: 速度。该值越小, 移动越快 Byte3: 文字大小。参见 1.7、文字大小编码 Byte4: 文字颜色。参见单字节 1.8、文字颜色编码 Byte5: 显示特效。参见 1.5、文字、图片显示效果编码 Byte6: 图片方式。参见 1.9、图片方式编码 Byte7: 时钟格式。参见 1.10、时钟格式和显示内容中的”时钟格式” Byte8: 时钟内容。参见 1.10、时钟格式和显示内容中的”时钟显示内容”
	pWndParam: 窗口参数。每个窗口 16 字节的参数。 数据总长度是: 窗口数量*16。 详见 1.13、窗口位置及属性
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据

	-6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_Net_InOutProgramTemplate

<code>int CPowerBox_Net_InOutProgramTemplate(int nCardID, byte byInOrOut)</code>	
作用	设置进入或退出模板方式
参数	nCardID: 卡 ID
	byInOrOut: 进入或退出。 1: 进入节目模板方式 0: 退出节目模板方式
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_Net_QueryProgramTemplate

<code>int CPowerBox_Net_QueryProgramTemplate(int nCardID , byte* pState);</code>	
作用	查询节目模板方式
参数	nCardID: 卡 ID
	pState:保留
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误

	-3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_Net_QueryProgramTemplate1

<pre>int CPowerBox_Net_QueryProgramTemplate1(int nCardID, byte byFlag, BYTE* pStateBuf, int nBufSize)</pre>	
作用	查询节目模板
参数	nCardID: 卡 ID
	byFlag: 选项 Bit0: 是否查询模板状态 Bit1: 返回模板定义的颜色灰度、的屏大小信息 Bit1~7: 保留
	pStateBuf: 结果缓冲区
	nBufSize: 结果缓冲区的大小, 不小于 10 字节
返回值	>=0: 包的数量 -1: 不正确的数据对象句柄
其它说明	

pStateBuf 数据含义如下:

数据项	数值	长度(字节)	意义描述
CC	0x83	1	说明本数据包是查询节目模板方式相关参数
选项		1	同发送值
模板方式		1	0: 非节目模板方式 1: 节目模板方式
模板状态		1	Bit0~1: 模板可用性 0: 模板不可用 1: 模板可用 其它: 保留 Bit2~7: 保留
颜色灰度		1	颜色及灰度。同“设置节目模板”的定义
屏宽度		2	高字节在前
屏高度		2	高字节在前

窗口数		1	播放窗口数量。最多支持 10 个播放窗口
-----	--	---	----------------------

CPowerBox_Net_DeleteProgram

<pre>int CPowerBox_Net_DeleteProgram(int nCardID,byte byConfig , byte byProNum , byte* pDelPro);</pre>	
作用	删除节目
参数	nCardID: 卡 ID
	byConfig: Bit0: 删除的节目范围 0: 删除全部 1: 删除指定节目 其它: 保留
	byProNum: 节目数量。删除全部节目不需要此数据项
	pDelPro: 要删除的节目列表
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_Net_SendText

<pre>int CPowerBox_Net_SendText(int nCardID, DWORD dwAppendCode , byte byProNo , byte byWndNo , byte byProp , byte *byShowFormat , char* pText);</pre>	
作用	发送指定文本到指定节目的指定窗口
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码, 高字节在前。

	byProNo: 节目号, 有效值 1~255
	byWndNo:窗口号。有效值 1~10, 超出节目模板定义时无效
	byProp:属性, Bit0~3: 文本类型 0: 普通文本 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留
	byShowFormat: 显示格式。 当”属性”的显示格式为 0 时不需要此项数据 Byte0~1: 停留时间。高字节在前 Byte2: 速度。该值越小, 速度越快 Byte3: 文字大小。参见 1.7、文字大小编码 Byte4: 文字颜色。参见单字节 1.8、文字颜色编码 Byte5: 显示特效。参见 1.5、文字、图片显示效果编码 Byte6: 保留 Byte7: 保留
	pText:文字数据。文本内容, 以 0x00 结束。
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_Net_SendPicture

<pre>int CPowerBox_Net_SendPicture(int nCardID, DWORD dwAppendCode , byte byProNo , byte byWndNo , byte byPicType , byte *byShowFormat , byte* pPicData , long lPicDataLen);</pre>	
作用	发送图片到指定节目的指定窗口
参数	nCardID: 卡 ID
	dwAppendCode:用户给出的附加码, 高字节在前。

	byProNo: 节目号, 有效值 1~255
	byWndNo:窗口号。有效值 1~10, 超出节目模板定义时无效
	byPicType: 图片类型。Bit0~3: 图片类型 1: GIF 图像文件的实际数据, 里面包含图像的宽度、高度等信息; 2: 在控制卡上预存的 GIF 图像文件的文件名。 4: 简单图片数据, 格式见 1.11、简单图片数据格式 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留
	byShowFormat: 显示格式。 当”属性”的显示格式为 0 时不需要此项数据 Byte0~1: 停留时间高字节在前 Byte2: 速度, 该值越小, 移动越快 Byte3: 显示特效,参见 1.5、文字、图片显示效果编码 Byte4: 图片方式(缩放、拉伸等), 参加 1.9、图片方式编码 Byte5: 保留 Byte6: 保留 Byte7: 保留
	pPicData:图片数据。
	lPicDataLen:图片长度。
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

CPowerBox_Net_SendClockOrTemperature

```
CPowerBox_Net_SendClockOrTemperature( int nCardID, DWORD dwAppendCode , BYTE byProNo ,
BYTE byWndNo , BYTE byProgramType , UINT nPropLen , BYTE* pProgramProp , byte* pBuf , int
nBufSize )
```

作用	发送时钟和温度
参数	nCardID: 卡 ID
	dwAppendCode:用户给出的附加码，高字节在前。
	byProNo: 节目号，有效值 1~255
	byWndNo:窗口号。有效值 1~10，超出节目模板定义时无效
	byProgramType: 节目类型。 Bit0~3: 类型。2 时钟；3 温度，其它保留。 Bit4: 显示格式。0 缺省格式,1 指定格式 Bit5~7: 保留备用。填 0
	nPropLen: 属性长度
	pProgramProp: 节目属性 属性数据含义根据类型不同而不同 类型=2: 同 1.13 的“时钟日历”属性 类型=3: 同 1.13 的“温度、湿度”属性
	pBuf:结果数据
	nBufSize: 结果缓冲区大小
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足，或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

pBuf 数据含义如下:

数据项	数值	长度(字节)	意义描述
CC	0x83	1	说明本数据包是查询节目模板方式相关参数
用户附加码		4	用户给出的附加码。高字节在前
节目号		1	同发送值。有效值 1~100。
窗口号		1	同发送值。有效值 1~10，超出节目模板定义时无效
丢包数		1	尚未接收到的包的数量。发送第一包时的丢包数量是总包数减去 1。
丢包的包号		变长	丢包的包号。总是按照从小到大排列；

			第一包的包号是 0。每个包号一个字节。
--	--	--	---------------------

CPowerBox_Net_SetAloneProgram

<pre>int CPowerBox_Net_SetAloneProgram(int nCardID, DWORD dwAppendCode , BYTE byProgramNo , BYTE byWindowCnt , BYTE* pWndParam, BYTE* pWndData)</pre>	
作用	设置独立节目
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码，高字节在前。
	byProgramNo: 节目号，有效值 1~100
	byWindowCnt: 窗口数。有效值 1~10，
	pWndParam: 窗口参数 每个窗口的信息表 22 字节，前 16 字为窗口位置及属性（参见 1.13、窗口位置及属性 ），17~19 字节为窗口数据偏移，20~22 字节为窗口数据长度。高字节在前。无数据则偏移和长度均为 0。
pWndData: 窗口数据 窗口要播放的数据，如“文本”、“图片”等。 第 1 字节：数据类型(1 文本；4 图片) 第 2 字节：数据格式（同发文本或发图片的定义） 第 3 字节起：文本或图片数据。	
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足，或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CPowerBox_Net_QueryProgram

<pre>Int CPowerBox_Net_QueryProgram(int nCardID ,byte byFlag , byte* pParam , BYTE* pBuf , int nBufSize);</pre>	
作用	查询节目信息
参数	hObj:数据包对象句柄
	byFlag: 查询标记, 指定要查询哪一项节目信息 0: 保留 1: 查询有效节目数量及节目号 2: 查询指定节目信息 其它: 保留
	pParam: 当查询标记为 1 时: byte1~5, 保留, 填 0 当查询标记为 2 时: byte1, 节目号; byte2~5, 保留, 填 0
	pBuf: 结果缓冲区
	nBufSize: 结果缓冲区大小
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误
其它说明	

pBuf 中数据含义如下:

- 查询“有效节目数量及节目号”

返回数据

数据项	数值	长度(字节)	意义描述
CC	0x89	1	说明本数据包是查询节目信息的返回包
信息标识		1	同发送值。
参数		5	同发送值。

有效节目数		1	有效节目的数量
有效节目号		变长	每一个字节标识一个有效的节目号。有效值1~100。

*返回数据包中的“返回值”非零时的含义：

0x01 控制卡没有运行在节目模板方式

0x10 不认识的信息标识

● 查询指定节目信息

返回数据

数据项	数值	长度(字节)	意义描述
CC	0x89	1	说明本数据包是查询节目信息的返回包
信息标识		1	同发送值。
参数		5	同发送值。
信息数		1	现在仅返回 1 项信息
节目号		1	节目号
用户附加码		4	用户发送节目信息时指定的用户附加码

*返回数据包中的“返回值”非零时的含义：

0x01 控制卡没有运行在节目模板方式

0x10 不认识的信息标识

0x11 无效的节目

0x12 不能取得需要的信息

CPowerBox_Net_SetProgramProperty

```
int CPowerBox_Net_SetProgramProperty( int nCardID, BYTE byOption, BYTE byProgramCnt,
BYTE* pPrograms, BYTE byPropertyID1, BYTE byPropertyID2, BYTE byProgramLevel, USHORT
nLoopCnt, USHORT nTime, BYTE* pDuetime, BYTE* pTimeInterval)
```

作用	设置节目属性
参数	nCardID:卡 ID
	byOption:选项, Bit0: 设置节目属性的范围 0: 全部节目 1: 指定节目 其它: 保留
	byProgramCnt:要生成的节目数量
	pPrograms:节目列表

	<p>ByPropertyID1: 属性标示 1, 按位标识要设置节目的哪些属性, 标识为 0 则该项数据不存在。</p> <p>Bit0: 节目级别。节目的重要级别</p> <p>Bit1: 循环次数。节目每次播放时循环的次数</p> <p>Bit2: 有效时长。从当前时间开始多长时间内有效播放</p> <p>Bit3: 有效期限</p> <p>Bit4~7: 保留</p>
	<p>ByPropertyID2: 属性标识 2。Bit0~4: 有效时段。>0 时为有效时段数.<=4</p> <p>Bit5~7: 保留</p>
	<p>byProgramLevel: 节目级别。1~3 级。优先播放级别高的节目</p>
	<p>nLoopCnt:循环次数。高字节在前。</p> <p>0:不播放该节目, 用于临时屏蔽节目</p> <p>1~255:该节目每次播放时循环的次数</p>
	<p>nTime: 有效时长。高字节在前。单位为分钟</p> <p>0: 不限定播放时间</p> <p>>0: 指定播放时间。单位是分钟。</p>
	<p>pDuetime:有效时限。</p>
	<p>pTimeInterval:有效时段。时段开始的“时分秒”和结束的“时分秒”</p> <p>时分秒各用一个字节表示</p>
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误</p>
其它说明	

CPowerBox_Net_SetSchedule

```
int CPowerBox_Net_SetSchedule(int nCardID, DWORD dwAppendCode, BYTE byScheduleNo, const
BYTE* pProperty, const BYTE* pBoxes, BYTE byBoxCnt)
```

作用	设置播放计划
----	--------

参数	nCardID:卡 ID
	dwAppendCode:用户给出的附加码, 高字节在前。
	byScheduleNo: 计划项编号, 有效值 1~100。总共支持 100 个计划项, 同一个项的新数据覆盖旧数据。
	pProperty:播放属性, 共 14 字节: byte 0: 格式及级别: Bit0~3: 表示数据格式, 填 0x01 Bit4~7: 表示优先级别。优先级别数值越大, 则越优先播放, 0 为最低优先级。 byte 1: 星期: Bit0~6: 7 个位分别标识星期日到星期六 byte 2~4: 开始日期, 3 个字节分别表示“年”“月”“日”。年份的值取 0~99,表示范围是 2000~2099 byte 5~7: 结束日期, 3 个字节分别表示“年”“月”“日”。年份的值取 0~99,表示范围是 2000~2099 byte 8~10: 开始时间, 3 个字节分别表示“时”“分”“秒” byte 11~13: 结束时间, 3 个字节分别表示“时”“分”“秒”
	pBoxes: 节目编号, 每个字节表示 1 个编号。编号从小到大排列, 不重复。
	byBoxCnt: 节目编号数量, 有效值 1~100
返回值	<p>0: 无错误</p> <p>-1: 不能生成命令数据</p> <p>-2: 命令数据打包错误</p> <p>-3: 不能连接控制卡</p> <p>-4: 数据分包错误</p> <p>-5: 超时未接收到返回数据</p> <p>-6: 返回数据长度不足, 或者数据标识错误</p> <p>-7: 数据校验错误-1: 不正确的数据对象句柄</p>
其它说明	

CPowerBox_Net_DeleteSchedule

<pre>int CPowerBox_Net_DeleteSchedule(int nCardID, DWORD dwAppendCode, const BYTE* pSchs, BYTE bySchCnt)</pre>	
作用	删除播放计划
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码，高字节在前。
	pSchs: 计划项编号，编号有效值 1~100。每个字节表示 1 个编号。 删除全部项时，本项数据长度为 1 字节，值是 0xff，作为删除的确认标记。
	bySchCnt: 要删除的计划项的数量。0 表示删除全部
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足，或者数据标识错误 -7: 数据校验错误-1: 不正确的数据对象句柄
其它说明	

CPowerBox_Net_GetSchedule

<pre>int CPowerBox_Net_GetSchedule(int nCardID, DWORD dwAppendCode, BYTE byType, BYTE byScheduleNo, byte* pBuf, int nBufSize)</pre>	
作用	查询播放计划
参数	nCardID: 卡 ID
	dwAppendCode: 用户给出的附加码，高字节在前。
	byType: 0: 查询有效的计划项编号 1: 查询指定编号的计划项内容 其它: 保留

	byScheduleNo: 编号有效值 1~100。 当查询类型为 0 时, 本项数据填 0。
	pBuf: 结果数据缓冲区
	nBufSize: 结果数据缓冲区大小
返回值	0: 无错误 -1: 不能生成命令数据 -2: 命令数据打包错误 -3: 不能连接控制卡 -4: 数据分包错误 -5: 超时未接收到返回数据 -6: 返回数据长度不足, 或者数据标识错误 -7: 数据校验错误-1: 不正确的数据对象句柄
其它说明	

pBuf 中数据格式如下:

数据项	数值	长度(字节)	意义描述
CC	0x8d	1	说明本数据包是查询节目模板方式相关参数
用户附加码		4	同发送数据
查询类型		1	同发送值
数量/编号		1	查询类型为 0, 该值是有效计划项数量 查询类型为 1, 该值是计划项编号
标号表/计划项内容		变长	查询类型为 0, 是有效计划项的编号表 查询类型为 1, 是计划项内容, 数据格式同“设置播放计划”的数据格式

要区分查询类型来处理返回数据

返回数据包中的“返回值”非零时的含义:

- 0x01 节目模板无效
- 0x11 不支持的查询类型
- 0x12 指定的计划项编号超出有效范围
- 0x80 当前不是节目模板方式

12、简单应用接口

12.1、RS232 简单应用接口总览

序号	API 接口函数名	接口说明
1	CP5200_RS232_UploadFile	上传文件到控制卡

2	CP5200_RS232_DownloadFile	下载控制卡文件
3	CP5200_RS232_RemoveFile	删除控制卡文件
4	CP5200_RS232_TestController	测试控制卡是否连接到 PC
5	CP5200_RS232_TestCommunication	测试控制卡是否可以正常通讯
6	CP5200_RS232_GetTime	得到控制卡时间
7	CP5200_RS232_SetTime	设置控制卡时间
8	CP5200_RS232_GetTempHumi	得到控制卡温度、湿度
9	CP5200_RS232_RestartApp	重启控制卡 App
10	CP5200_RS232_RestartSys	重启控制卡系统
11	CP5200_RS232_GetTypeInfo	得到控制卡类型信息
12	CP5200_RS232_SendInstantMessage CP5200_RS232_SendInstantMessage1	发送即时信息
13	CP5200_RS232_ReadHWSetting	读取扫描参数
14	CP5200_RS232_WriteHWSetting	写扫描参数
15	CP5200_RS232_ReadSoftwareSwitchInfo	读取软件开关屏信息参数
16	CP5200_RS232_WriteSoftwareSwitchInfo	设置开关屏信息参数
17	CP5200_RS232_QueryControllerInfo	查询控制卡信息
18	CP5200_RS232_ReadNetworkParam	读网络连接参数
19	CP5200_RS232_WriteNetworkParam	写网络连接参数
20	CP5200_RS232_Upgrade	升级控制卡

本类函数的使用步骤如下：

第1步：初始化串口参数

串口参数初始化只记录下参数信息，不对串口进行实际的操作。

第2步：使用上述简单应用接口

备注：本类接口事先不必考虑串口是否已经打开，只要初始化好串口参数即可。

12.2、RS232 简单应用接口详细描述

CP5200_RS232_UploadFile

<pre>int CP5200_RS232_UploadFile(int nCardID, const char* pSourceFilename, const char* pTargetFilename);</pre>	
作用	上传文件到控制卡
参数	nCardID: 控制卡 ID

	pSourceFilename: 源文件名称
	pTargetFilename: 目的文件名称
返回值	<p>0: 成功</p> <p>-1: 源文件读取错误</p> <p>-2: 不能生成命令数据</p> <p>-3: 制作开始文件上传数据错误或开始文件上传返回数据错误</p> <p>-5: 不能打开串口</p> <p>-7: 文件上传返回数据错误</p> <p>-8: 文件上传没有返回数据</p> <p>-9: 制作结束文件上传数据错误</p> <p>-10: 开始文件上传或结束文件上传没有返回数据</p> <p>-11: 结束文件上传返回数据错误</p>
其它说明	

CP5200_RS232_DownloadFile

<pre>int CP5200_RS232_DownloadFile(int nCardID, const char* pSourceFilename, const char* pTargetFilename);</pre>	
作用	下载控制卡文件
参数	nCardID: 控制卡 ID
	pSourceFilename: 源文件名称, 如果该文件是系统盘文件, 文件名前面需加上 "S:"
	pTargetFilename: 目的文件名称
返回值	<p>0: 成功</p> <p>-1: 失败</p> <p>-2: 不能生成命令数据</p> <p>-3: 打开控制卡文件失败</p> <p>-4: 得到控制卡文件信息失败</p> <p>-5: 不能打开串口</p> <p>-6: 分配文件缓冲区失败</p> <p>-7: 读取控制卡文件数据失败</p> <p>-8: 保存文件失败</p>

其它说明	
------	--

CP5200_RS232_RemoveFile

<code>int CP5200_RS232_RemoveFile(int nCardID, const char* pFilename);</code>	
作用	删除控制卡文件
参数	nCardID: 控制卡 ID
	pFilename: 文件名称, 如果该文件是系统盘文件, 文件名前面需加上“S:”
返回值	1: 成功删除文件 0: 不能删除文件 -1: 不正确的数据对象句柄 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 缓冲区长度不足 -5: 不能打开串口 -6: 不能生成命令数据 -7: 得到控制卡文件信息失败
其它说明	

CP5200_RS232_TestController

<code>int CP5200_RS232_TestController(int nCardID);</code>	
作用	测试控制卡是否连接到 PC
参数	nCardID: 控制卡 ID
返回值	>0: 说明 ID 为 nCardID 的控制卡已连接
其它说明	

CP5200_RS232_TestCommunication

<code>int CP5200_RS232_TestCommunication(int nCardID);</code>	
作用	测试控制卡是否可以正常通讯
参数	nCardID: 控制卡 ID
返回值	1: 通讯成功 0: 通讯失败
其它说明	本函数不负责的串口的开与关, 可作为测试串口是否已打开之用

CP5200_RS232_GetTime

<code>int CP5200_RS232_GetTime(int nCardID, BYTE *pBuf, int nBufSize);</code>	
作用	得到控制卡时间
参数	nCardID: 控制卡 ID
	pBuf: 时间信息缓冲区, 信息含义如下 0 字节: 秒 1 字节: 分 2 字节: 时 3 字节: 星期 4 字节: 日 5 字节: 月 6 字节: 年(2 位数字, 加上 2000 是实际年份值)
	nBufSize: 时间信息缓冲区的长度, 要求不小于 7 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_SetTime

<code>int CP5200_RS232_SetTime(byte nCardID, const BYTE *pInfo);</code>	
作用	设置控制卡时间
参数	nCardID: 控制卡 ID
	<p>pInfo: 时间信息数据缓冲区，信息数据含义如下</p> <p>0 字节: 秒</p> <p>1 字节: 分</p> <p>2 字节: 时</p> <p>3 字节: 星期</p> <p>4 字节: 日</p> <p>5 字节: 月</p> <p>6 字节: 年(2 位数字，加上 2000 是实际年份值)</p>
返回值	<p>1: 成功</p> <p>0: 失败</p>
其它说明	

CP5200_RS232_GetTempHumi

<code>int CP5200_RS232_GetTempHumi (int nCardID, BYTE byFlag , BYTE *pBuf, int nBufSize)</code>	
作用	得到温度、湿度信息
参数	nCardID: 控制卡 ID
	<p>byFlag: 查询标记</p> <p>Bit0: 是否要查询温度(0 不要,1 要)</p> <p>Bit1: 是否要查询湿度(0 不要,1 要)</p>

	<p>pBuf: 温度、湿度信息缓冲区, 长度为 8 字节, 信息含义如下:</p> <p>byte 0: 查询标记, 含义同发送包</p> <p>byte 1~2: 温度值(摄氏度): 第 1 字节: Bit7: 数值符号。1 负值, 0 正值。 Bit6~0: 温度绝对值整数部分的高 7 位 第 2 字节: Bit7~4: 温度绝对值整数部分的低 4 位 Bit3~0: 小数部分, 单位是 1/16(即 0.0625)</p> <p>byte 3~4: 温度值(华氏度):</p> <p>byte 5: 温度调整值, Bit7: 1 华氏度, 0 摄氏度 Bit6: 1 负值, 0 正值 Bit5~0: 温度调整的绝对值</p> <p>byte 6: 湿度值。有效值 0~100</p> <p>byte 7: 湿度调整值 Bit7: 保留 Bit6: 1 负值, 0 正值 Bit5~0: 湿度调整的绝对值</p>
	nBufSize: 温度信息缓冲区的长度,至少 8 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_RestartApp

int CP5200_RS232_RestartApp(byte nCardID);	
作用	重启 App 程序
参数	nCardID: 控制卡 ID
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_RestartSys

<code>int CP5200_RS232_RestartSys(byte nCardID);</code>	
作用	重启控制卡系统
参数	nCardID: 控制卡 ID
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_GetTypeInfo

<code>int CP5200_RS232_GetTypeInfo(byte nCardID, BYTE *pBuf, int nBufSize);</code>	
作用	得到控制卡类型
参数	nCardID: 控制卡 ID
	pBuf: 控制卡类型信息，信息含义如下 字节 0: 控制卡类型 字节 1: FPGA 版本 字节 2-5: BIOS 版本 字节 6-9: APP 版本
	nBufSize: 控制卡类型信息的长度，要求不小于 10 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_SendInstantMessage

<code>int CP5200_RS232_SendInstantMessage(byte nCardID, byte byPlayTimes , int x , int y , int cx , int cy , byte byFontSizeColor , int nEffect , byte nSpeed , byte byStayTime , const char* pText);</code>	
作用	发送即时信息

参数	nCardID:控制卡 ID
	byPlayTimes:播放次数
	x:显示起点 x, 显示区域的左上角横坐标。
	y:显示起点 y, 显示区域的左上角纵坐标。
	Cx:显示宽度, 显示区域的宽度。等于 0 含义为屏最大宽度。
	Cy:显示高度, 显示区域的高度。等于 0 含义为屏最大高度。
	byFontSizeColor:字体颜色。文字显示的字体大小。 Bit0~3: 字体大小 Bit4: 颜色的红色分量 Bit5: 颜色的绿色分量 Bit6: 颜色的蓝色分量 Bit7: 保留
	nEffect:显示效果
	nSpeed:速度。0~255。数值越小, 速度越快。对立即显示的方式无效。
	byStayTime:停留时间。高字节在前, 单位是秒。显示每页之间的停留时间。对滚动方式无效。
	pText:文本数据。
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_SendInstantMessage1

CP5200_RS232_SendInstantMessage1(BYTE nCardID, BYTE byPlayTimes , int x , int y , int cx , int cy , int nFontSize , byte byColorAlign , int nEffect , BYTE nSpeed , BYTE byStayTime ,const char* pText)	
作用	发送即时信息
参数	nCardID:控制卡 ID
	byPlayTimes:播放次数
	x:显示起点 x, 显示区域的左上角横坐标。
	y:显示起点 y, 显示区域的左上角纵坐标。
	Cx:显示宽度, 显示区域的宽度。等于 0 含义为屏最大宽度。

	Cy:显示高度，显示区域的高度。等于 0 含义为屏最大高度。
	nFontSize: 文字大小及类型，参见 1.7、文字大小及类型编码
	byColorAlign: 颜色及对齐方式 Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~5: 水平对齐。0 左，1 中，2 右 Bit6~7: 垂直对齐。0 上，1 中，2 下
	nEffect:显示效果
	nSpeed:速度。0~255。数值越小，速度越快。对立即显示的方式无效。
	byStayTime:停留时间。高字节在前，单位是秒。显示每页之间的停留时间。对滚动方式无效。
	pText:文本数据。
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_ReadHWSetting

<code>int CP5200_RS232_ReadHWSetting(byte nCardID, BYTE *pBuf, int nBufSize, int nPassword);</code>	
作用	读取扫描参数
参数	nCardID: 控制卡 ID
	pBuf: 扫描参数信息，各字节含义参见 1.14、扫描参数各字节含义
	nBufSize: 扫描参数信息的长度，要求不小于 16 字节
	nPassword: 解析密码，根据不同控制卡填入不同密码，否则不予受理
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_WriteHWSetting

<code>int CP5200_RS232_WriteHWSetting(byte nCardID, BYTE *pSetting, int nPassword);</code>	
作用	写控制卡扫描参数
参数	nCardID: 控制卡 ID
	pSetting: 扫描参数缓冲区, 其各字节含义参见: 1.14、扫描参数各字节含义
	nPassword: 设置密码, 根据不同控制卡填入不同密码, 否则不予受理
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_ReadSoftwareSwitchInfo

<code>int CP5200_RS232_ReadSoftwareSwitchInfo(BYTE nCardID, BYTE *pBuf, int nBufSize)</code>	
作用	读取软件开关屏信息参数
参数	nCardID: 控制卡 ID
	pBuf: 软件开关屏信息缓冲区, 各字节含义参见: CP5200_ParseReadSoftwareSwitchInfoRet 参数 pInfoBuffer 的说明
	nBufSize: 软件开关屏信息的长度, 要求不小于 9 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_WriteSoftwareSwitchInfo

<code>int CP5200_RS232_WriteSoftwareSwitchInfo(BYTE nCardID, const BYTE *pBuf)</code>	
作用	设置软件开关屏信息参数
参数	nCardID: 控制卡 ID

	<p>pBuf: 软件开关屏信息缓冲区, 其各字节含义参见: CP5200_MakeWriteSoftwareSwitchInfoData 参数</p> <p>pSoftwareSwitchInfoBuf 的说明</p>
返回值	<p>1: 成功</p> <p>0: 失败</p>
其它说明	

CP5200_RS232_QueryControllerInfo

<pre>int CP5200_RS232_QueryControllerInfo(BYTE nCardID, byte byInfoFlag, byte *pInfoBuf, int nInfoBufLen, const char *szSavePath)</pre>	
作用	查询控制卡信息
参数	nCardID: 控制卡 ID
	byInfoFlag: 查询标记, 暂只支持信息标志: 0x0b
	pInfoBuf: 查询结果缓存
	第 0~1 字节: 节目号。高字节在前, 第一个节目从 1 开始, 0 表示没有节目在播放或者播放临时信息
	第 2~3 字节: 播放项号。
	第 4~7 字节: 节目已播放时长, 单位为 1/10 秒, 高字节在前
	第 8~11 字节: 播放项已播放时长, 单位为 1/10 秒, 高字节在前
	第 12~13 字节: 图像宽度, 高字节在前
	第 14~15 字节: 图像高度, 高字节在前
	第 16 字节: 颜色及灰度
	第 17~20 字节: 图像数据长度, 高字节在前
	nInfoBufLen: 查询结果缓存长度, 至少 21 字节
	szSavePath: 显示屏截屏图片保存路径
返回值	<p>1: 成功</p> <p>0: 失败</p>
其它说明	

CP5200_RS232_ReadNetworkParam

CP5200_RS232_ReadNetworkParam(BYTE nCardID, BYTE *pBuf, int nBufSize)	
作用	读取网络连接参数
参数	nCardID: 控制卡 ID
	pBuf: 网络连接参数缓冲区, 各字节含义如下: Byte 0~3: IP 地址 Byte 4~7: 网关 Byte 9~11: 子网掩码 Byte 12~13: IP 端口号 Byte 14~17: 网络识别码
	nBufSize: 网络连接参数信息的长度, 要求不小于 18 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_WriteNetWorkParam

CP5200_RS232_WriteNetworkParam(BYTE nCardID, DWORD dwIP , DWORD dwGateway , DWORD dwIPMast , WORD nPort , DWORD dwIDCode)	
作用	设置网络连接参数
参数	nCardID: 控制卡 ID
	dwIP: IP 地址
	dwGateway: 网关
	dwIPMast: 子网掩码
	nPort: 通信端口
	dwIDCode: 网络识别码
返回值	1: 成功 0: 失败
其它说明	

CP5200_RS232_Upgrade

CP5200_RS232_Upgrade(int nCardID, int nProgramType, const char* pProgramFilename)	
作用	上传文件到控制卡
参数	nCardID: 控制卡 ID
	nProgramType: 升级程序类型 3: BIOS 4: APP 5: SCAN 6: NET 8: BAS 9: GRAPH
	pProgramFilename: 程序文件名称
返回值	0: 成功 -1: 源文件读取错误或者程序类型不匹配 -2: 不能生成命令数据 -3: 制作开始文件上传数据错误或开始文件上传返回数据错误 -4: 制作文件上传数据错误 -5: 不能打开串口 -7: 文件上传返回数据错误 -8: 文件上传没有返回数据 -9: 制作结束文件上传数据错误 -10: 开始文件上传或结束文件上传没有返回数据 -11: 结束文件上传返回数据错误
其它说明	

12.3、Network 简单应用接口总览

序号	API 接口函数名	接口说明
----	-----------	------

1	CP5200_Net_UploadFile	上传文件到控制卡
2	CP5200_Net_DownloadFile	下载控制卡文件
3	CP5200_Net_RemoveFile	删除控制卡文件
4	CP5200_Net_TestController	测试控制卡是否连接到在网络中
5	CP5200_Net_TestCommunication	测试控制卡是否可以正常通讯
6	CP5200_Net_GetTime	得到控制卡时间
7	CP5200_Net_SetTime	设置控制卡时间
8	CP5200_Net_GetTempHumi	得到控制卡温度、湿度
9	CP5200_Net_RestartApp	重启控制卡 App
10	CP5200_Net_RestartSys	重启控制卡系统
11	CP5200_Net_GetTypeInfo	得到控制卡类型
12	CP5200_Net_SendInstantMessage CP5200_Net_SendInstantMessage1	发送即时信息
13	CP5200_Net_ReadHWSetting	读取扫描参数
14	CP5200_Net_WriteHWSetting	写扫描参数
15	CP5200_Net_ReadSoftwareSwitchInfo	读取软件开关屏信息参数
16	CP5200_Net_WriteSoftwareSwitchInfo	设置软件开关屏信息参数
17	CP5200_Net_QueryControllerInfo	查询控制卡信息
18	CP5200_Net_ReadNetworkParam	读网络连接参数
19	CP5200_Net_WriteNetworkParam	写网络连接参数
20	CP5200_Net_Upgrade	升级控制卡

本类函数的使用步骤如下：

第1步：初始化网口参数

网口参数初始化只记录下参数信息，不对网口进行实际的操作。

第2步：使用上述简单应用接口

备注：本类接口事先不必考虑网口是否已经打开，只要初始化好网口参数即可

12.4、Network 简单应用接口详细描述

CP5200_Net_UploadFile

<pre>int CP5200_Net_UploadFile(int nCardID, const char* pSourceFilename, const char* pTargetFilename)</pre>	
作用	上传文件到控制卡
参数	nCardID: 控制卡 ID

	pSourceFilename: 源文件名称
	pTargetFilename: 目的文件名称
返回值	<p>0: 成功</p> <p>-1: 源文件读取错误</p> <p>-2: 不能生成命令数据</p> <p>-3: 制作开始文件上传数据错误或开始文件上传返回数据错误</p> <p>-5: 不能连接控制卡</p> <p>-7: 文件上传返回数据错误</p> <p>-8: 文件上传没有返回数据</p> <p>-9: 制作结束文件上传数据错误</p> <p>-10: 开始文件上传或结束文件上传没有返回数据</p> <p>-11: 结束文件上传返回数据错误</p>
其它说明	

CP5200_Net_DownloadFile

<pre>int CP5200_Net_DownloadFile(int nCardID, const char* pSourceFilename, const char *pTargetFilename);</pre>	
作用	下载控制卡文件
参数	nCardID: 控制卡 ID
	pSourceFilename: 源文件名称, 如果该文件是系统盘文件, 文件名前面需加上 "S:"
	pTargetFilename: 目的文件名称
返回值	<p>0: 成功</p> <p>-1: 失败</p> <p>-2: 不能生成命令数据</p> <p>-3: 打开控制卡文件失败</p> <p>-4: 得到控制卡文件信息失败</p> <p>-5: 不能连接控制卡</p> <p>-6: 分配文件缓冲区失败</p> <p>-7: 读取控制卡文件数据失败</p> <p>-8: 保存文件失败</p>

其它说明	
------	--

CP5200_Net_RemoveFile

<code>int CP5200_Net_RemoveFile(int nCardID, const char* pFilename);</code>	
作用	删除控制卡文件
参数	nCardID: 控制卡 ID
	pFilename: 文件名称, 如果该文件是系统盘文件, 文件名前面需加上“S:”
返回值	1: 成功删除文件 0: 不能删除文件 -1: 不正确的数据对象句柄 -2: 返回数据类型错误 -3: 返回数据长度不足 -4: 缓冲区长度不足 -5: 不能连接控制卡 -6: 不能生成命令数据 -7: 得到控制卡文件信息失败
其它说明	

CP5200_Net_TestController

<code>int CP5200_Net_TestController(int nCardID);</code>	
作用	测试控制卡是否连接
参数	nCardID: 控制卡 ID
返回值	>0: 说明 ID 为 nCardID 的控制卡已连接
其它说明	

CP5200_Net_TestCommunication

<code>int CP5200_Net_TestCommunication(int nCardID);</code>	
作用	测试控制卡是否可以正常通讯
参数	nCardID: 控制卡 ID
返回值	1: 成功 0: 失败
其它说明	本函数不负责的网口的开与关，可作为测试网口是否已打开之用

CP5200_Net_GetTime

<code>int CP5200_Net_GetTime(int nCardID, BYTE *pBuf, int nBufSize);</code>	
作用	得到控制卡时间
参数	nCardID: 控制卡 ID
	pBuf: 时间信息缓冲区，信息含义如下 0 字节: 秒 1 字节: 分 2 字节: 时 3 字节: 星期 4 字节: 日 5 字节: 月 6 字节: 年(2 位数字，加上 2000 是实际年份值)
	nBufSize: 时间信息缓冲区的长度，要求不小于 7 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_SetTime

<code>int CP5200_Net_SetTime(byte nCardID, const BYTE *pInfo);</code>

作用	设置控制卡时间
参数	nCardID: 控制卡 ID
	pInfo: 时间信息数据缓冲区, 信息数据含义如下 0 字节: 秒 1 字节: 分 2 字节: 时 3 字节: 星期 4 字节: 日 5 字节: 月 6 字节: 年(2 位数字, 加上 2000 是实际年份值)
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_GetTempHumi

int CP5200_Net_GetTempHumi(int nCardID, byte byFlag , BYTE *pBuf, int nBufSize)	
作用	得到温度信息
参数	nCardID: 控制卡 ID
	byFlag: 查询标记 Bit0: 是否要查询温度(0 不要,1 要) Bit1: 是否要查询湿度(0 不要,1 要)

	<p>pBuf: 温度、湿度信息缓冲区，长度为 8 字节，信息含义如下：</p> <p>byte 0: 查询标记，含义同发送包</p> <p>byte 1~2: 温度值(摄氏度): 第 1 字节: Bit7: 数值符号。1 负, 0 正值。 Bit6~0: 温度绝对值整数部分的高 7 位 第 2 字节: Bit7~4: 温度绝对值整数部分的低 4 位 Bit3~0: 小数部分, 单位是 1/16(即 0.0625)</p> <p>byte 3~4: 温度值(华氏度):</p> <p>byte 5: 温度调整值, Bit7: 1 华氏度, 0 摄氏度 Bit6: 1 负值, 0 正值 Bit5~0: 温度调整的绝对值</p> <p>byte 6: 湿度值。有效值 0~100</p> <p>byte 7: 湿度调整值 Bit7: 保留 Bit6: 1 负值, 0 正值 Bit5~0: 湿度调整的绝对值</p>
	nBufSize: 温度信息缓冲区的长度
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_RestartApp

int CP5200_Net_RestartApp(byte nCardID);	
作用	重启控制卡 App
参数	nCardID: 控制卡 ID
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_RestartSys

<code>int CP5200_Net_RestartSys(byte nCardID);</code>	
作用	重启控制卡系统
参数	nCardID: 控制卡 ID
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_GetTypeInfo

<code>int CP5200_Net_GetTypeInfo(byte nCardID, BYTE *pBuf, int nBufSize);</code>	
作用	得到控制卡类型
参数	nCardID: 控制卡 ID
	pBuf: 控制卡类型信息，信息含义如下 字节 0: 控制卡类型 字节 1: FPGA 版本 字节 2-5: BIOS 版本 字节 6-9: APP 版本
	nBufSize: 控制卡类型信息的长度，要求不小于 10 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_SendInstantMessage

<code>int CP5200_Net_SendInstantMessage(byte nCardID, byte byPlayTimes, int x, int y, int cx, int cy, byte byFontSizeColor, int nEffect, byte nSpeed, byte byStayTime, const char* pText);</code>	
作用	发送即时信息

参数	nCardID:控制卡 ID
	byPlayTimes:播放次数
	x:显示起点 x, 显示区域的左上角横坐标。
	y:显示起点 y, 显示区域的左上角纵坐标。
	Cx:显示宽度, 显示区域的宽度。等于 0 含义为屏最大宽度。
	Cy:显示高度, 显示区域的高度。等于 0 含义为屏最大高度。
	byFontSizeColor:字体颜色。文字显示的字体大小。 Bit0~3: 字体大小 Bit4: 颜色的红色分量 Bit5: 颜色的绿色分量 Bit6: 颜色的蓝色分量 Bit7: 保留
	nEffect:显示效果
	nSpeed:速度。0~255。数值越小, 速度越快。对立即显示的方式无效。
	byStayTime:停留时间。高字节在前, 单位是秒。显示每页之间的停留时间。对滚动方式无效。
	pText:文本数据。
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_SendInstantMessage1

CP5200_Net_SendInstantMessage1 (BYTE nCardID, BYTE byPlayTimes , int x , int y , int cx , int cy , int nFontSize , byte byColorAlign , int nEffect , BYTE nSpeed , BYTE byStayTime , const char* pText)	
作用	发送即时信息
参数	nCardID:控制卡 ID
	byPlayTimes:播放次数
	x:显示起点 x, 显示区域的左上角横坐标。
	y:显示起点 y, 显示区域的左上角纵坐标。

	Cx:显示宽度，显示区域的宽度。等于 0 含义为屏最大宽度。
	Cy:显示高度，显示区域的高度。等于 0 含义为屏最大高度。
	nFontSize: 文字大小及类型，参见 1.7、文字大小及类型编码
	byColorAlign: 颜色及对齐方式 Bit0: 红色标记 Bit1: 绿色标记 Bit2: 蓝色标记 Bit3: 保留 Bit4~5: 水平对齐。0 左，1 中，2 右 Bit6~7: 垂直对齐。0 上，1 中，2 下
	nEffect:显示效果
	nSpeed:速度。0~255。数值越小，速度越快。对立即显示的方式无效。
	byStayTime:停留时间。高字节在前，单位是秒。显示每页之间的停留时间。对滚动方式无效。
	pText:文本数据。
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_ReadHWSetting

<code>int CP5200_Net_ReadHWSetting(byte nCardID, BYTE *pBuf, int nBufSize , int nPassword);</code>	
作用	读取扫描参数
参数	nCardID: 控制卡 ID
	pBuf: 扫描参数信息，各字节含义参见 1.14、扫描参数各字节含义
	nBufSize: 扫描参数信息的长度，要求不小于 16 字节
	nPassword: 解析密码，根据不同控制卡填入不同密码，否则不予受理
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_WriteHWSetting

int CP5200_Net_WriteHWSetting(byte nCardID, BYTE *pSetting, int nPassword);	
作用	写控制卡扫描参数
参数	nCardID: 控制卡 ID
	pSetting: 扫描参数缓冲区, 其各字节含义参见: 1.14、扫描参数各字节含义
	nPassword: 设置密码, 根据不同控制卡填入不同密码, 否则不予受理
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_ReadSoftwareSwitchInfo

int CP5200_Net_ReadSoftwareSwitchInfo(BYTE nCardID, BYTE *pBuf, int nBufSize)	
作用	读取软件开关屏信息参数
参数	nCardID: 控制卡 ID
	pBuf: 软件开关屏信息缓冲区, 各字节含义参见: CP5200_ParseReadSoftwareSwitchInfoRet 参数 pInfoBuffer 的说明
	nBufSize: 软件开关屏信息的长度, 要求不小于 9 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_WriteSoftwareSwitchInfo

int CP5200_Net_WriteSoftwareSwitchInfo(BYTE nCardID, const BYTE *pBuf)	
作用	设置软件开关屏信息参数
参数	nCardID: 控制卡 ID

	<p>pBuf: 软件开关屏信息缓冲区, 其各字节含义参见: CP5200_MakeWriteSoftwareSwitchInfoData 参数</p> <p>pSoftwareSwitchInfoBuf 的说明</p>
返回值	<p>1: 成功</p> <p>0: 失败</p>
其它说明	

CP5200_Net_QueryControllerInfo

<pre>int CP5200_Net_QueryControllerInfo(BYTE nCardID, byte byInfoFlag, byte *pInfoBuf, int nInfoBufLen, const char *szSavePath)</pre>	
作用	查询控制卡信息
参数	nCardID: 控制卡 ID
	byInfoFlag: 查询标记, 暂只支持信息标志: 0x0b
	pInfoBuf: 查询结果缓存
	第 0~1 字节: 节目号。高字节在前, 第一个节目从 1 开始, 0 表示没有节目在播放或者播放临时信息
	第 2~3 字节: 播放项号。
	第 4~7 字节: 节目已播放时长, 单位为 1/10 秒, 高字节在前
	第 8~11 字节: 播放项已播放时长, 单位为 1/10 秒, 高字节在前
	第 12~13 字节: 图像宽度, 高字节在前
	第 14~15 字节: 图像高度, 高字节在前
	第 16 字节: 颜色及灰度
	第 17~20 字节: 图像数据长度, 高字节在前
	nInfoBufLen: 查询结果缓存长度, 至少 21 字节
	szSavePath: 显示屏截屏图片保存路径
返回值	<p>1: 成功</p> <p>0: 失败</p>
其它说明	

CP5200_Net_ReadNetworkParam

CP5200_Net_ReadNetworkParam(BYTE nCardID, BYTE *pBuf, int nBufSize)	
作用	读取网络连接参数
参数	nCardID: 控制卡 ID
	pBuf: 网络连接参数缓冲区, 各字节含义如下: Byte 0~3: IP 地址 Byte 4~7: 网关 Byte 9~11: 子网掩码 Byte 12~13: IP 端口号 Byte 14~17: 网络识别码
	nBufSize: 网络连接参数信息的长度, 要求不小于 18 字节
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_WriteNetworkParam

CP5200_Net_WriteNetworkParam(BYTE nCardID, DWORD dwIP , DWORD dwGateway , DWORD dwIPMast , WORD nPort , DWORD dwIDCode)	
作用	设置网络连接参数
参数	nCardID: 控制卡 ID
	dwIP: IP 地址
	dwGateway: 网关
	dwIPMast: 子网掩码
	nPort: 通信端口
	dwIDCode: 网络识别码
返回值	1: 成功 0: 失败
其它说明	

CP5200_Net_Upgrade

CP5200_Net_Upgrade(int nCardID, int nProgramType , const char* pProgramFilename)	
作用	上传文件到控制卡
参数	nCardID: 控制卡 ID
	nProgramType: 升级程序类型 3: BIOS 4: APP 5: SCAN 6: NET 8: BAS 9: GRAPH
	pProgramFilename: 程序文件名称
返回值	0: 成功 -1: 源文件读取错误或者程序类型不匹配 -2: 不能生成命令数据 -3: 制作开始文件上传数据错误或开始文件上传返回数据错误 -4: 制作文件上传数据错误 -5: 不能连接控制卡 -7: 文件上传返回数据错误 -8: 文件上传没有返回数据 -9: 制作结束文件上传数据错误 -10: 开始文件上传或结束文件上传没有返回数据 -11: 结束文件上传返回数据错误
其它说明	

13、其它应用接口

13.1、其它应用接口总览

序号	API 接口函数名	接口说明
1	CP5200_CalcImageDataSize	计算 Image 数据大小
2	CP5200_MakeImageDataFromFile	从图片文件得到 Image 数据
3	CP5200_TextToImage	获取格式化文本的图片
4	CP5200_TextToImageW	获取格式化文本的图片（宽字符）
5	CP5200_TextToImageEx	获取扩展格式化文本的图片

13.2、其它应用接口详细描述

CP5200_CalcImageDataSize

int CP5200_CalcImageDataSize(WORD imgw, WORD imgh, BYTE color)	
作用	计算 Image 数据大小
参数	imgw: Image 宽度
	imgh: Image 高度
	color: Image 颜色
返回值	>=0: Image 数据大小
其它说明	

CP5200_MakeImageDataFromFile

int CP5200_MakeImageDataFromFile(WORD imgw, WORD imgh, BYTE color, BYTE *pDatBuf, int nBufSize, const char* pFilename, int nMode)	
作用	从图片文件得到 Image 数据
参数	imgw: Image 宽度
	imgh: Image 高度

	color: Image 颜色
	pDatBuf: Image 数据缓冲区
	nBufSize: Image 数据缓冲区大小
	pFilename: 图片文件名
	nMode: 图片模式, 见 1.9、图片方式编码
返回值	>=0: Image 数据的大小 -1: 图片文件找不到或加载不成功 -2: 图片转换不成功 -3: 图片模式不对 -4: Image 数据缓冲区不足
其它说明	

CP5200_TextToImage

<pre>int CP5200_TextToImage(const char *pSavePath, const char *pText, const char *pFontFaceName, const byte *pFormatData, const byte *pScreenData)</pre>	
作用	获取格式化文本的图片
参数	pSavePath: 保存路径
	pText: 文本字符串
	pFontFaceName: 字体名称
	pformatData: 格式控制数据
	pScreenData: 屏数据
返回值	0: 成功 -1: 失败
其它说明	

CP5200_TextToImageW

<pre>int CP5200_TextToImageW(const char *pSavePath, const wchar_t *pText, const char *pFontFaceName, const byte *pFormatData, const byte *pScreenData)</pre>	
--	--

作用	获取格式化文本的图片（宽字符）
参数	pSavePath: 保存路径
	pText: 文本字符串
	pFontFaceName: 字体名称
	pformatData: 格式控制数据
	pScreenData: 屏数据
返回值	0: 成功 -1: 失败
其它说明	

CP5200_TextToImageEx

<pre>int CP5200_TextToImageEx(const char *pSavePath, const byte *pTextContent, const byte *pFormatData, const byte *pScreenData);</pre>	
作用	获取扩展格式化文本的图片
参数	pSavePath: 保存路径
	pTextContent: 扩展格式字符串内容
	pFormatData: 扩展格式格式控制数据
	pScreenData: 扩展格式屏数据
返回值	0: 成功 -1: 失败
其它说明	