DV10 用户界面

一、文件夹布局

\DV10

BMP

UI 工程 1

UI 工程 2

UI 工程 3

……

DB

RSC

Pic_color

String

1、DV10 系统运行当前目录，存放系统运行程序 PDV.EXE，写 BIN 输出文件 UI.BIN。

2、BMP 存放 UI 工程页面的缩微图，不要删除。

3、DB 所有工程的数据文件，以 Access 数据库形式存放。

4、RSC 资源文件目录，存放与目标（小机）关联的资源文件和与资源相关的资源号。

4.1、文件夹 PIC_COLOR 存放时彩色图片文件。

4.2、文件夹 string 存放字符串的图片文件。

4.3、文件 RES.H 内容是 PIC_COLOR 和 string 资源对应的文件和 ID 号，如：

////BmpResID Define Table////

#define ACTIVE 1

#define BACKDROP 2

#define BACKREC 3

#define BATTERY1 4

#define BATTERY2 5

。。。

#define S6 436

#define S7 437

#define S8 438

#define S9 439

#define S10 440

此文件基于建龙的资源工具生成，文件格式有要求。写 BIN 文件时不是写文件名，而是该资源对应的文件名对应的资源 ID 号。

4.4、文件 Color_table.H 文件是颜色对应为资源 ID 号，如：

//OSD0 Palette Table

#define PAL0_FFFFFF 0

#define PAL0_FEFFCA 1

#define PAL0_FEFE98 2

#define PAL0_FEFF67 3

……

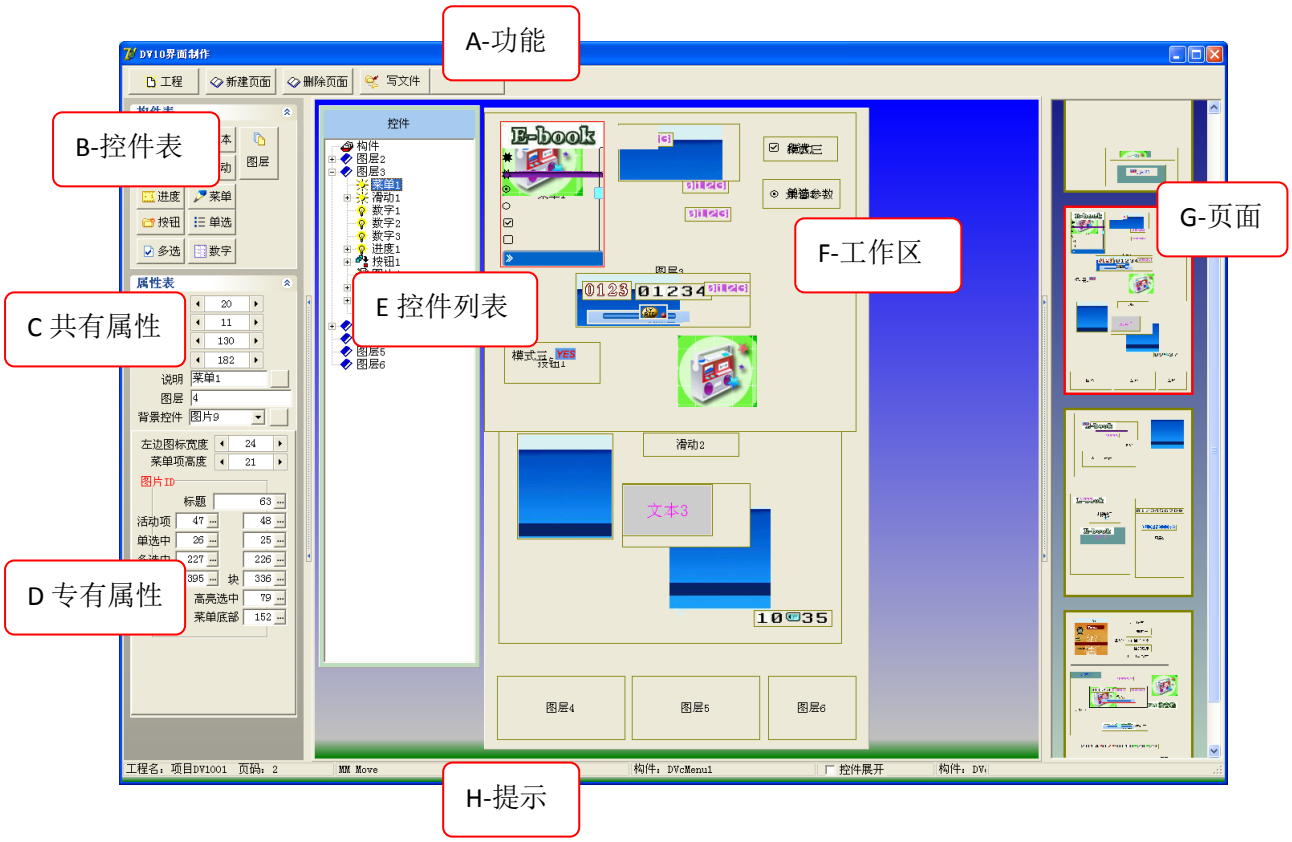
#define PAL1_999999 86

```
#define    PAL1_989964      87

#define    PAL1_989832      88

#define    PAL1_989901      89
```

系统根据图层不同去查找该表，然后用 ID 号写 BIN 文件。

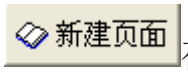
二、UI 系统使用

1、总体界面简介

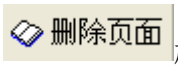


1.1、功能，对较大事件的按钮选择。

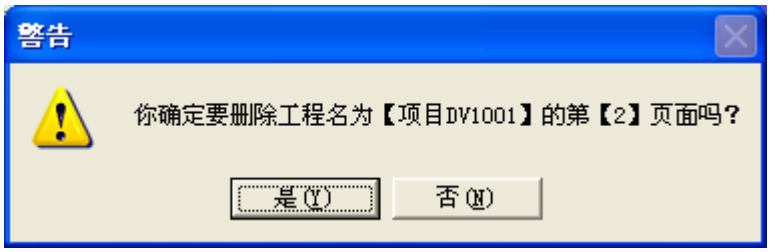
选择工程、新建工程、查看工程等。

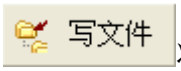


在当前工程建立新页面，同时将当前页面的图层控件的数据拷贝到新页面。



删除当前页面及页面上的全部信息，系统会提示确认

删除吗？如： 



将当前工程文件写目标机（小机）可识别的数据文件，该文件存放于\DV10\UI.BIN。

1.2 控件表，一共有 11 个 控件。点击该控件，会自动增加该控件到工作区的当前活动的控件上。图层控件例外，它是直接拖拽到工作区，图层控件只能添加到工作区，不能到其他控件上（包括图层本身）。

1.3、控件的共有属性设置，如：坐标，宽高等。

1.4、控件的专有属性，选择不同的控件，该区出现的可编辑修改数据不同，如菜单有很多属性数据，而单选一个都没有。对于时间和数字控件的数字图片在选择时只需选择一个，系统会根据资源文件命名方法自动查找其他九个数字文件的资源号。

1.5、控件列表，以树形列出工作区的控件及其控件的所属关系。点击控件可能会改变控件的共有属性和控件的专有属性的数据或显示，同时也会在工作区将该控件以高亮（红色边框）显示，如该控件在底层被其他控件挡住，该控件会从底层到顶层。

1.6、工作区，控件的位置及所属关系，以所见即所得方式显示。

1.7、页面，显示页面的缩微图，同时点击缩微图将选择相应的页面。

1.8、状态提示区，如工作区的两个相关的图层垂直交叉时，系统会提示，以红色醒目的方式，如： 

三、资源选择

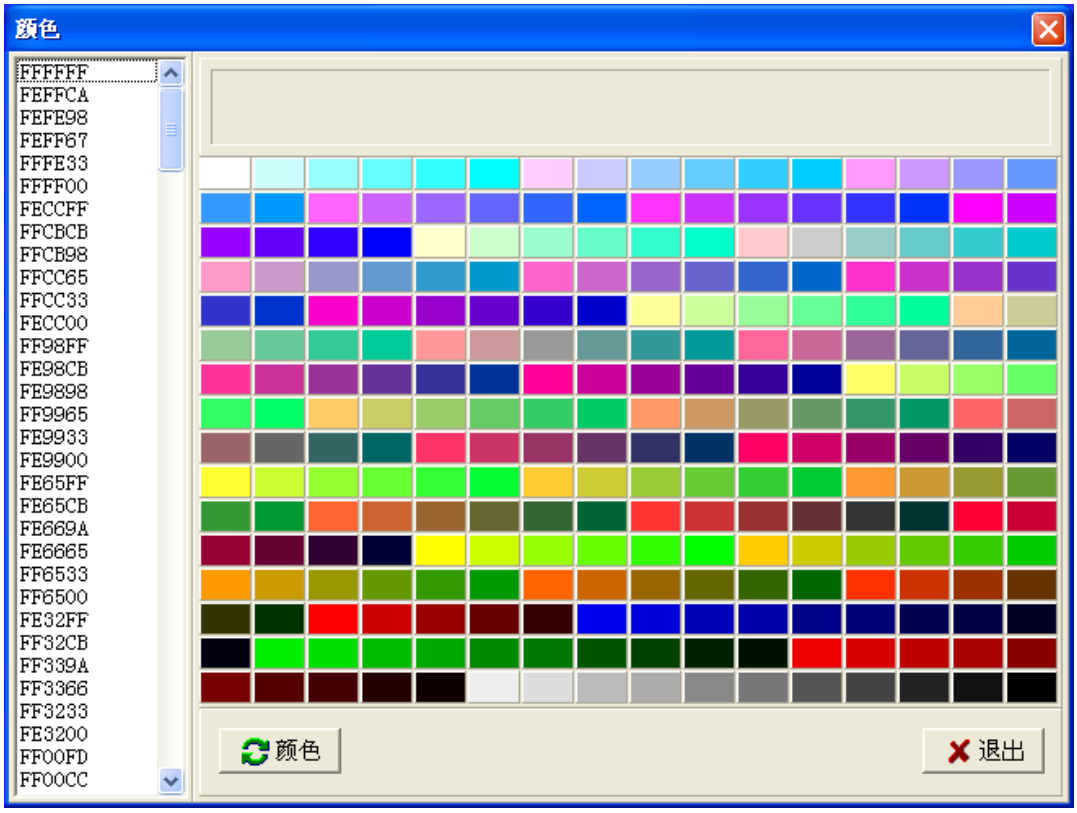
1、A-资源，列出由 D-切换（字符串、彩色图片、单色图片）的资源，资源是按实际大小显示，点击所列资源，将在 B-显示显示选中的资源。

2、C-选择，根据资源的文件名选择资源，与 A 选择的资源效果一样。

3、点击“选择”按钮，返回选中的资源，然后退出。

4、该资源的 ID 号对应是由资源表\RSC\RES.H 决定。

四、颜色资源



1、颜色主要是文本控件的前景色和背景色的选择。

2、点击相中的颜色，该色会在上部显示，鼠标放在颜色块上，会显示该颜色的十六进制的值。选择颜色也可以点击左边的颜色值选取。

3、点击“颜色”按钮将使用上部显示的颜色，然后退出。

4、颜色资源表的 ID 由\RSC\Color_table.H 决定，注：不同的图层对应的表不一样，前 5 层是 PAL0_XXXXXX，后 3 层是 PAL1_XXXXXX。

附件 A、控件定义（来自建龙）

以此定义设计控件和写 BIN 文件

1.资源生成工具生成图片格式

jpg 格式默认指定图片层　　打勾

bmp 格式默认 osd 层　24 位色　8 位色　单色　不打勾

2.UI 编辑器增加图层

增加画布，相当于图层

```
typedef struct PIC
{
    u16 x;
    u16 y;
    u16 id;
};

//1.图片控件
typedef struct CPIC
{
    u8 layer;       //图层
    u16 x;          //x 坐标
    u16 y;          //y 坐标
    u16 width;      //图片宽度
    u16 height;     //图片高度
```

```
    u16 id;          //图片 id 号
}; //11 Bytes

//2.文本控件

typedef enum DISP_MODE
{
    UI_STATIC_TEXT,        //静态显示
    UI_SCROLL_TEXT,        //滚动显示
    UI_KLOK_TEXT,          //卡拉 OK 滚动显示
};

typedef enum FONT_SIZE
{
    SMALL_FONT,      //小(16x16)
    MIDDLE_FONT,     //中(20x20)
    BIG_FONT,        //大(24x24)
};

typedef struct CTEXT
{
    u8 layer;            //图层
```

u16 x;　　//起始 x 坐标

u16 y;　　//起始 y 坐标

u16 width;　　//显示区域的宽

u16 height;　　//显示区域的高

u16 backpicture;　//背景图片　0:无背景图片,背景颜色有效

其它:背景图片控件 ID 号,背景颜色无效（图片控件 id 号）

u8 backcolor;　　//背景颜色

u8 forecolor;　　//前景颜色

u8 FONT_SIZE　font_size;　　//字号,支持小(16x16),中(20x20),大(24x24)三种字号

u8　DISP_MODE dispmode;　//显示方式

}; //15 Bytes

//说明:背景颜色和前景颜色为 8 位色

//3.时间控件

//type:时间控件的类型

//0:hh:mm　　时钟:分钟

//1:hh:mm:ss　　时钟:分钟:秒钟

//2:　　yyyy-mm-dd hh:mm:ss　年-月-日　　时钟:分钟:秒钟

```
typedef struct CTIME
{
    u8 layer;
    u16 x;              //起始 x 坐标
    u16 y;              //起始 y 坐标
    u16 width;          //控件宽度
    u16 height;         //控件高度
    u8   type;          //时间显示类型
    u16 num_id[12];     //图片数字 0~9 的 ID 号，10~11 为分隔符 ID 号,前两种类型记录到 num_id[10],最后一种记录到 num_id[11]
};// 34 Bytes


//4.菜单控件
typedef struct CMENU
{
    u8 layer;
    u16 x;          //起始 x 坐标
    u16 y;          //起始 y 坐标
    u16 width;      //菜单宽度
    u16 height;             //菜单高度
```

```
    u16 mItemWidth;      //菜单项左边图标宽度
    u16 mItemHeight;     //菜单项高度
    u16 backpic;         //背景图片(图片控件 id 号)
    u16 titlepic;        //标题栏
    u16 bottompic;       //菜单底部状态栏
    u16 activepic;       //活动项图标
    u16 unactivepic;     //非活动项图标
    u16 OneChoiceSel;    //单选已选中图标
    u16 OneChoiceNoSel; //单选未选中图标
    u16 MultiChoiceSel;  //多选已选中图标
    u16 MultiChoiceNoSel;    //多选未选中图标
    u16 selpic;                  //高亮选中图片
    u16 scrollbar;       //滚动条图片
    u16 scrollbar_p;     //滚动块图片
}; //37 Bytes


//5.数字控件
typedef struct CNUMBER
{
    u8 layer;
    u16 x;           //起始 x 坐标
```

```
    u16 y;          //起始 y 坐标
    u16 width;      //控件宽度
    u16 height;         //控件高度
    u16 num_id[10];             //数字 0-9 的图片 ID 号
    u16 num_bits;           //数字有效位数
}; // 31


//6.滑动块控件
typedef struct CSLIDER
{
    u8 layer;
    u16 x;          //起始 x 坐标
    u16 y;          //起始 y 坐标
    u16 width;      //控件宽度
    u16 height;     //控件高度
    PIC bar;        //滑动块图片
    PIC slider;     //滑动条图片
    u16 cnum[3];  //数字控件的 id 号,分别表示最小值/当前值/最大值（数字控件 id 号）
    u16 backpic;   //背景图片（图片控件 id 号）
}; // 29
```

```
//7.进度条控件
typedef struct CPROGRESS
{
    u8 layer;
    u16 x;          //起始 x 坐标
    u16 y;          //起始 y 坐标
    u16 width;      //控件宽度
    u16 height;     //控件高度
    PIC bar;        //进度块
    PIC progress;   //进度条
    u16 num;        //数字，用于显示百分比（数字控件 id 号）
    u16 backpicture; //背景图片（图片控件 id 号）
}; // 25


//8.单选控件
typedef struct CRADIO
{
    u8 layer;
    u16 x;      //x 坐标
    u16 y;          //y 坐标
```

```
    u16 width;      //控件宽度
    u16 height;     //控件高度
    PIC sel;        //选中图标
    PIC unsel;      //未选中图标
    PIC strid;      //字符串 id 号
}; // 27

//9.多选控件
typedef struct CHECKBOX
{
    u8 layer;
    u16 x;          //x 坐标
    u16 y;          //y 坐标
    u16 width;      //控件宽度
    u16 height;     //控件高度
    PIC sel;        //选中图标
    PIC unsel;      //未选中图标
    PIC strid;      //字符串 id 号
    u8 status;      //默认是选中状态还是未选中状态  0,1
}; // 28

//10.按钮控件
```

```
typedef struct CBUTTON
{
    u8 layer;
    u16 x;          //x 坐标
    u16 y;          //y 坐标
    u16 width;      //控件宽度
    u16 height;     //控件高度
    PIC picid;      //按钮图片
    PIC strid;      //按钮文本     ---> strID  字符串 id 号 ?
}; // 21

//11.图层(所有控件都在图层之上，必须先建立图层，才能在该图层上画控件)

typedef struct LAYER
{
    LAYER_TYPE type;     //图层序号    u8
    u8 alpha;            //图层透明度，64 level(0~63)
    u16 x;               //x 坐标
    u16 y;               //y 坐标
    u16 width;           //图层宽度
    u16 height;          //图层高度
```

```
}; // 10


typedef enum LAYER_TYPE
{
    LAYER_IMAGE0,        //图片层 0(没有透明度)
    LAYER_IMAGE1,        //图片层 1
    LAYER_OSD0_WIN0,   //OSD 层 0  窗口 0
    LAYER_OSD0_WIN1,   //OSD 层 0  窗口 1
    LAYER_OSD0_WIN2,   //OSD 层 0  窗口 2
    LAYER_OSD1_WIN0,   //OSD 层 1  窗口 0
    LAYER_OSD1_WIN1,   //OSD 层 1  窗口 1
    LAYER_OSD1_WIN2,   //OSD 层 1  窗口 2
};//图层标号
```

说明：

1.osd 层一共有两层,每层 osd 最多有三个窗口，这三个窗口在垂直方向上不能交叉,不同层的 osd 在垂直方向可以相互交叉

2.除了图片层 0 没有透明度，其它各层都有透明度

3.各图层还需要遵循对齐的原则

图片层 x,y,width,height 都需要 4 对齐(4 的倍数)

OSD 层 x,y,width,height 都需要 2 对齐(2 的倍数)

```
//.STY 文件头结构
typedef struct STYFILEHEADER
{
    u32 flag;       //*.sty 文件标志 0xFF 0xFE 0x55 0xAA
    u16 fileversion;   //文件版本 0x0101,版本 1.01
    u16 filecount;     //==11  控件类型总数  1:CPIC  2.TEXT  3.TIME  4.MENU  7.NUMBER  5.SLIDER  6.PROGRESS  8.RADIO  9.CHECKBOX  10.BUTTON 11.LAYER
}; //8 bytes

//控件信息
typedef struct
{
    u32 num;      //该类控件总数
    u32 offset;         //该类控件偏移地址,相对于风格文件的起始地址的偏移
}CONTROL_HEADER; //8 bytes


//控件类型
enum
```

```
{
    TYPE_PIC,
    TYPE_TEXT,
    TYPE_TIME,
    TYPE_MENU,
    TYPE_SLIDER,
    TYPE_PROGRESS,
    TYPE_NUMBER,
};
```