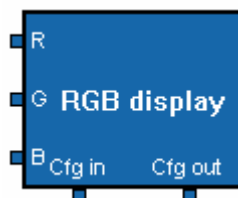


# RGB display

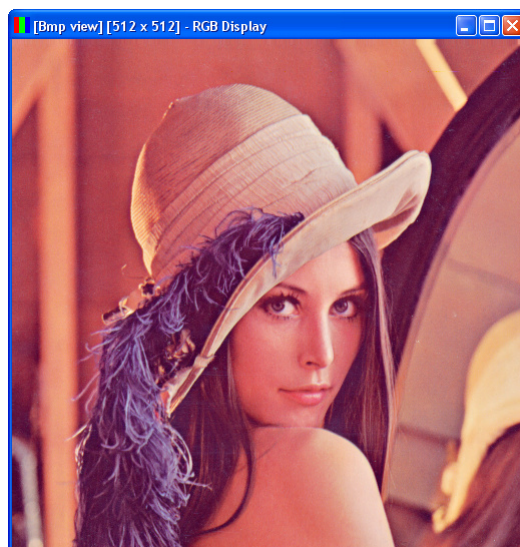
## Obsah

1. Popis.....	1
2. Závislosti .....	2
3. Implementačné informácie .....	2
4. Piny .....	2
5. Konfigurácia .....	3
5.1 Konfiguračná štruktúra.....	3
5.2 Konfiguračný dialóg.....	3
6. Definícia triedy .....	3
7. Príklad použitia .....	4

## 1. Popis



Trieda *RGB display* predstavuje farebný display. Vstupom sú tri šedo tónové obrazy rovnakého rozlíšenia v štruktúrach *TPlane*, ktoré predstavujú farebné kanály R, G a B. Vstupné roviny sa naškálujú a orežú na interval  $\langle 0, 255 \rangle$ .



V záhlaví okna sa zobrazuje nastavený názov a rozlíšenie obrazu. Kliknutím pravého tlačidla myši sa zobrazí kontextové menu okna.



Tu sú na výber dva módy okna, *Stretch* a *Fullsize*. V móde *Stretch* sa veľkosť obrazu prispôsobuje veľkosti okna. V móde *Fullsize* sa obraz zobrazuje v pôvodnej veľkosti, v prípade, že je obraz väčší ako veľkosť okna, uchopením a potiahnutím obrazu zmeníme pohľad naň. Voľbou *Close* okno zavrieme.

## 2. Závislosti

Trieda *RGB display* používa tieto externé definičné súbory a knižnice:

**Headers:** *filtergraph.h*, *datatypes.h*

**Libs:** *filtergraph.lib*

## 3. Implementačné informácie

Informácie o triede:

**Názov:** *RGB display*

**Verzia:** *1.0*

**Magic:** *30*

Informácie o definíciách:

**Header:** *rgbdisplay.h*

**Konfiguračný header:** *rgbdisplaycfg.h*

**Lib:** *rgbdisplay.lib*

Informácie o knižnici obsahujúcej triedu:

**Názov knižnice:** *RGB display Library*

**Verzia knižnice:** *1.0*

**Dll súbor knižnice:** *rgbdisplay.dll*

## 4. Piny

Piny sú popísané spôsobom: „*[index pinu na filtri] názov pinu: popis pinu*“.

**[0] Cfg in:** Konfiguračný vstup. Vstupom je serializovaná konfigurácia.

**[1] Cfg out:** Konfiguračný výstup. Výstupom je serializovaná konfigurácia.

**[2] R:** Červený vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

**[3] G:** Zelený vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

**[4] B:** Modrý vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

## 5. Konfigurácia

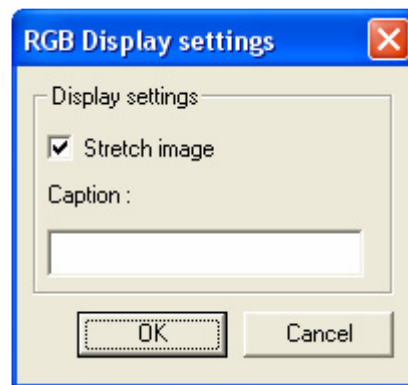
### 5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
    bool Stretch;      // mód okna, ak Stretch=TRUE, tak mód Stretch, inak mód Fullsize  
    char* Caption;     // názov okna  
} TRGBDisplayConfig;
```

Indikátor *Stretch* určuje mód okna. Ak *Stretch=TRUE*, tak je okno v móde *Stretch*, inak je v móde *Fullsize*. Reťazec *Caption* určuje názov okna, ktorý sa zobrazuje v záhlaví.

### 5.2 Konfiguračný dialóg



Konfiguračný dialóg poskytuje voľby popísané pri konfiguračnej štruktúre. Môžeme nastaviť mód okna voľbou *Stretch* a názov okna v kolonke *Caption*.

## 6. Definícia triedy

Trieda *RGB display* je definovaná nasledovne:

```
class TRGBDisplayFilter : public TFilter {  
private:  
    HINSTANCE      hInstance;    // identifikácia inštancie knižnice  
    HWND          hwnd;         // identifikácia okna  
    HBITMAP        hBitmap;     // identifikácia bitmapy  
    HMENU          hMenu;       // identifikácia menu  
  
    TBuffer DataBuffer;         // dátový buffer  
  
    char* Caption;              // názov okna  
  
    LONG Width;                 // šírka vstupného obrazu  
    LONG Height;                // výška vstupného obrazu  
  
    TPin* CfgInPin;             // vstupný konfiguračný pin
```

```

TPin* CfgOutPin;           // výstupný konfiguračný pin
TPin* RPin;               // vstupné piny...
TPin* GPin;
TPin* BPin;

bool StopFlag;            // indikátor zastavenia

int setConfigFromPin(TPin *pin);    // konfigurácia na pine...
int putConfigOnPin(TPin *pin);
int freeConfigOnPin(TPin *pin);
int clearOutputPins();              // vyčistenie výstupov...
int freeOutputData();

public:
    int ScreenMode;                // mód obrazovky
    int ScreenState;               // stav obrazovky v móde FullSize
    POINT APnt;                   // pomocné body...
    POINT BPnt;

    TRGBDisplayFilter();           // konštruktor
    TRGBDisplayFilter(HINSTANCE hInst); // konštruktor s parametrami
    ~TRGBDisplayFilter();          // deštruktor

    int setConfigData(TBuffer config,int type);    // predefinované funkcie triedy TFilter...
    TBuffer getConfigData(int type);
    int initialize();
    int run();
    int reset();
    int stop();
    int finalize();
    int showConfigDialog();

    HBITMAP getBitmap(); // pomocné funkcie...
    HMENU getMenu();
    SIZE getSize();
};

```

## 7. Príklad použitia

