

Binary encoder

Obsah

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

1. Popis



Trieda *Binary encoder* zakóduje pole celých nezáporných čísel do zhustenej binárnej podoby. Analyzuje pole čísel, zistí potrebný počet bitov na zakódovanie maximálneho prvku, týmto počtom bitov zakóduje každý prvok poľa, a zapíše ho do postupnosti bitov. Pole čísel je kódované nasledovne (počet bitov - položka):

```
32          Count          // počet prvkov poľa
8           BitCount        // počet bitov použitý na každý prvok
BitCount    Number[0]      // nultý prvok zakódovaný na BitCount bitov
...
BitCount    Number[Count-1] // posledný prvok zakódovaný na BitCount bitov
```

2. Závislosti

Trieda *Binary encoder* používa tieto externé definičné súbory a knižnice:

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

Libs: *filtergraph.lib*, *bitstream.lib*

3. Implementačné informácie

Informácie o triede:

Názov: *Binary encoder*

Verzia: *1.0*

Magic: *170*

Informácie o definíciách:

Header: *binaryencoder.h*

Konfiguračný header: *binaryencodercfg.h*

Lib: *binaryencoder.lib*

Informácie o knižnici obsahujúcej triedu:

Názov knižnice: *Binary encoder Library*

Verzia knižnice: *1.0*

Dll súbor knižnice: *binaryencoder.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] In: Dátový vstup. Vstupom je pole celých čísel zapísané v štruktúre *TIntArray*.

[3] Out: Dátový výstup. Výstupom je buffer binárnych dát zakódovaný hore uvedeným spôsobom.

5. Konfigurácia

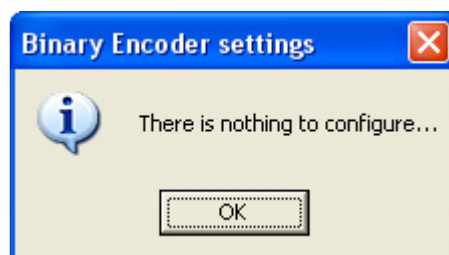
5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
} TBinaryEncoderConfig;
```

Táto trieda nemá žiadne nastavenia. Výstup deterministicky závisí od vstupných dát.

5.2 Konfiguračný dialóg



Konfiguračný dialóg neposkytuje žiadne voľby.

6. Definícia triedy

Trieda *Binary encoder* je definovaná nasledovne:

```
class TBinaryEncoderFilter : public TFilter {  
private:
```

```

HINSTANCE hInstance;           // identifikácia inštancie knižnice

TIntArray *Input;              // vstupné pole celých čísel
TBuffer OutPut;               // výstupné binárne dáta

TPin* CfgInPin;               // vstupný konfiguračný pin
TPin* CfgOutPin;              // výstupný konfiguračný pin
TPin* InPin;                  // vstupný pin
TPin* OutPin;                 // výstupný pin

bool StopFlag;                // indikátor zastavenia

int getNumBits();              // zistí potrebný počet bitov
int encode(int bitcount);      // zakóduje prvky poľa daným počtom bitov

int setConfigFromPin(TPin *pin); // nastaví konfiguráciu z pinu
int putConfigOnPin(TPin *pin);   // sprístupní konfiguráciu na pine
int freeConfigOnPin(TPin *pin);  // uvoľní konfiguráciu na pine
int clearOutputPins();           // vyčistí výstupné piny
int freeOutputData();           // uvoľní výstupné dáta

public:
    TBinaryEncoderFilter();      // konštruktor
    TBinaryEncoderFilter(HINSTANCE hInst); // konštruktor s identifikáciou knižnice
    ~TBinaryEncoderFilter();     // deštruktor

int setConfigData(TBuffer config,int type); // nastaví konfiguračné dáta
TBuffer getConfigData(int type);           // vráti konfiguračné dáta
int initialize();                          // inicializácia
int run();                                // beh
int reset();                              // reset stavu
int stop();                              // ukončenie behu
int finalize();                           // finalizácia
int showConfigDialog();                   // vyvolanie konfiguračného dialógu
};

```

7. Príklad použitia

