



Knihovna RecDBXLib

ZÁZNAMY V DATABOXU

KNIHOVNA RecDBXLib DATABÁZE V DATABOXU

2. vydání – říjen 2008

OBSAH

1. ÚVOD	3
2. KNIHOVNA RecDBXLib DATABÁZE V DATABOXU	4
2.1. ORGANIZACE DAT V DBX.....	5
2.1.1. Seznam záznamů v databázi	5
2.1.2. Data v databázi	6
2.2. FUNKCE A FUNKČNÍ BLOKY KNIHOVNY.....	6
2.2.1. Obecný popis	6
2.2.2. Funkční blok fbBaseDBX	6
2.2.3. Funkční blok fbKeyMobil	7
2.2.4. Pomocné funkce	7
2.3. APLIKACE KNIHOVNY	8

1. ÚVOD

Knihovny funkcí a funkčních bloků jsou nedílnou součástí instalace programovacího prostředí Mosaic. Z hlediska jejich výstavby je možné knihovny rozdělit na následující typy:

- vestavěné (built-in) knihovny
- standardně dodávané externí knihovny
- uživatelsky definované knihovny

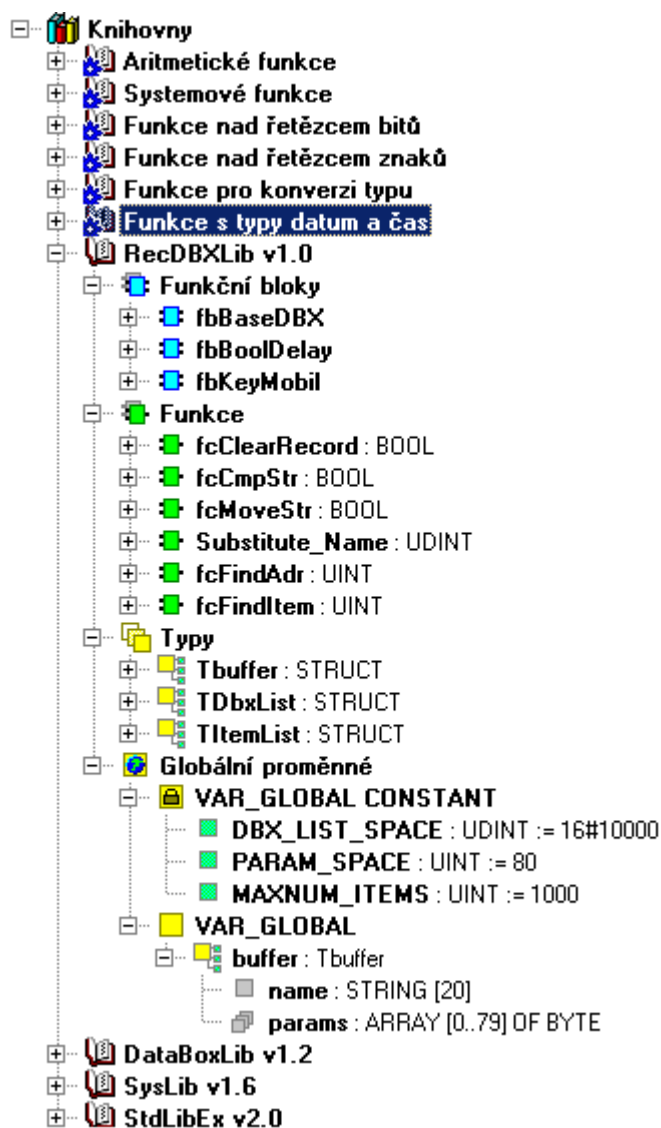
Knihovna může obsahovat deklarace funkcí, funkčních bloků, datových typů a globálních proměnných.

Velice často pro řízení strojů a technologií je třeba ukládat a obnovovat záznamy informací, které slouží pro parametrizaci řídicích algoritmu pro různé varianty výrobků. Pro tento účel je potřeba použít vnitřní paměť systému, která je typu „persistent“, t.j. paměť která je energeticky nezávislá a neztratí informaci při vypnutí napájení. V systémech Tecomat je tato paměť označována jménem „Databox“, nebo zkratkou DBX. Různé typy centrálních jednotek mají různé velikosti databoxu od 128kB až 3MB. Pro práci s databoxem jsou v systémech Tecomat základní instrukce pro identifikaci, čtení a zápis dat a programátor může tato data dále zpracovávat. Pro usnadnění práce s daty je vytvořena tato knihovna, která umožňuje vytvářet v databoxu databázy záznamů řízenou textovým řetězcem.

2. KNIHOVNA RecDBXLib DATABÁZE V DATABOXU

Knihovna RecDBXLib.mlb obsahuje základní funkční bloky, které slouží zejména při řízení databáze záznamů v Databoxu PLC Tecomat.

Následující obrázek ukazuje strukturu knihovny v prostředí Mosaic.



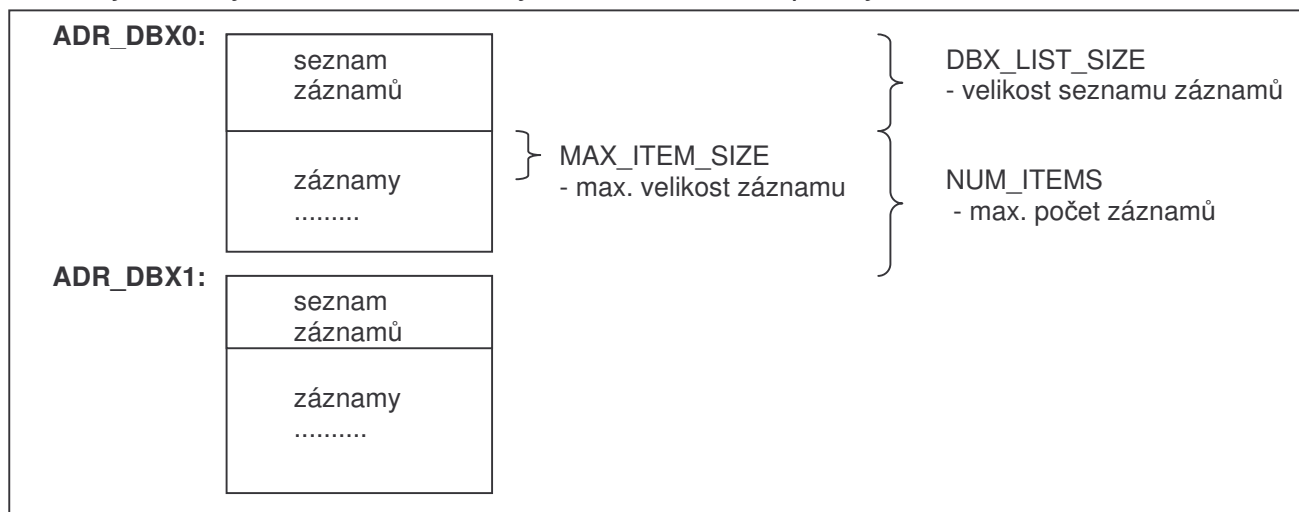
Obr. 2.1 Knihovna RecDBXLib

Základní funkční bloky můžete použít přidáním knihovny RecDBXLib.mlb do svého projektu. Nebo ve zdrojové formě jsou dostupné také v příkladu RecDBXLib.piz a každý je může použít pro další inspiraci, dále si je upravit a vygenerovat si svoji odvozenou knihovnu.

V příkladu je naznačena aplikace při použití WebMakeru jako komunikační rozhraní a také použití levného textového zobrazovače ID-14 pro ovládání databáze.

2.1. ORGANIZACE DAT V DBX

V databoxu je možné vytvořit současně více databází se záznamy. Posunutí začátku každé databáze je dán konstantou zapsanou do vstupu AdrDbx. Programátor musí zajistit odstup počátků jednotlivých databází tak, aby se data v DBX nepřekrývala!



Obr.2.1. Organizace databází v DBX

Databáze se skládá ze seznamu záznamů, který usnadňuje vyhledávání v záznamech a z vlastních záznamů, které MAX_ITEM_SIZE .

2.1.1. Seznam záznamů v databázi

Seznam záznamů je struktura skládající se z hlavičky s maximálním počtem záznamů v databázi, počtem všech již uložených záznamů a číslem aktuálního záznamů, s kterým se nyní pracuje. Následuje prázdná položka jen pro zarovnání v paměti.

Za hlavičkou následuje pole obsahující ke každému záznamu vždy dvě položky. „Hash kód“ a adresu v DBX. „Hash kód“ je CRC zakódovaný název položky, aby se v databázi snadněji vyhledávalo. Adresa ukazuje na začátek dat položky v záznamu.

```

TYPE
  TItemList : STRUCT
    hash      : UDINT;          // Hash
    addr      : UDINT;          // Address
  END_STRUCT;

  TDbxList : STRUCT
    maxItems   : UINT;          // max number of items
    numItems   : UINT;          // number of items
    curItem    : UINT;          // Current item
    dummy      : UINT;          // Padding
    item       : ARRAY[0..NUM_ITEMS] OF TItemList; // list of items
  END_STRUCT;
END_TYPE

```

Pro seznam záznamů je na počátku databáze v DBX vyčleněna paměť o velikosti dané konstantou DBX_LIST_SIZE.

2.1.2. Data v databázi

Každý záznam v databázi má rezervovaný prostor v paměti DBX o velikosti dané konstantou `MAX_ITEM_SIZE`. I když můj vlastní záznam nezaplní zcela tento prostor paměti, je vhodné ponechat tuto rezervu pro případné budoucí přidání parametrů. Nebude pak třeba starší záznamy znovu všechny vstupovat do databáze. Proto program pracuje s proměnnou „buffer“, do které je můj záznam, „myRecord“, nasměrován definicí:

```
myRecord AT buffer : TMyRecord;    // record is located to start of buffer
```

Tato definice představuje vlastně jiný pohled (přes brýle jiné datové struktury) na tu stejnou oblast paměti.

2.2. FUNKCE A FUNKČNÍ BLOKY KNIHOVNY

2.2.1. Obecný popis

Knihovna obsahuje hlavní funkční bloky:

fbBaseDBX a **fbKeyMobil**.

a pomocné funkce:

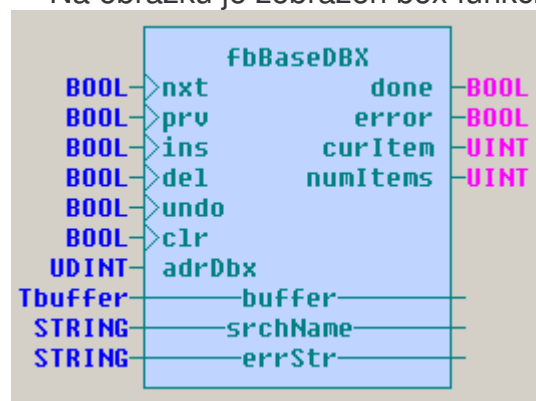
fcCmpStr, **fcMoveStr**, **fcCoutAdr**, **Substitute_Name**, **fcFindAdr**, **fcFindItem**.

Dále definuje datové typy:

Tbuffer, **TDbxList**, **TItemList**

2.2.2. Funkční blok fbBaseDBX

Na obrázku je zobrazen box funkčního bloku **fbBaseDBX** a typy jeho vstupů a výstupů.



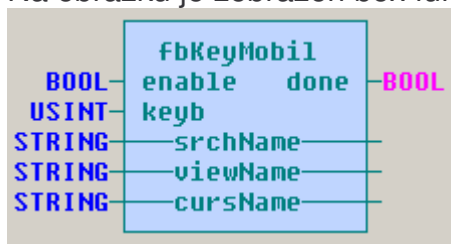
Popis proměnných :

Název	Význam	Typ	Formát
nxt	přechod na následující záznam	vstupní	BOOL
prv	přechod na předchozí záznam		BOOL
ins	uložit nový záznam za aktuální záznam nebo přepsat záznam, když jméno již existuje		BOOL
del	vypustit aktuální záznam		BOOL
undo	obnovit naposledy vypuštěný záznam před aktuální pozici		BOOL
clr	smazat všechny záznamy z databáze (pozor není již návratu!)		BOOL
adrDBX	adresa v DBX, kde začíná tato databáze		UDINT
buffer	odkaz na pracovní záznam v registrech	vstupně/ výstupní	Tbuffer
srchName	odkaz na jméno, které bude vyhledáváno v databázi(je-li nové,hledá se)		STRING[20]
errStr	odkaz na chybové hlášení ve formě textového řetězce		STRING[20]
done	hotovo	výstupní	BOOL
error	příznak chyby		BOOL
curItem	číslo aktuálně zpracovávaného záznamu		UINT
numItems	počet uložených záznamů		UINT

Všechny řídicí vstupy typu bool vykonávají svoji funkci na náběžnou hranu signálu.

2.2.3. Funkční blok fbKeyMobil

Na obrázku je zobrazen box funkčního bloku fbKeyMobil a typy jeho vstupů a výstupů.



Popis proměnných :

Název	Význam	Typ	Formát
enable	povolení funkce	vstupní	BOOL
keyb	kód klávesy z klávesnice		BOOL
srchName	odkaz na jméno, do kterého se zapíše výsledný string po stisku Enter	výstupní	STRING[20]
viewName	odkaz na string, který se zobrazuje během editace na displeji	výstupní	STRING[20]
cursName	odkaz na string, který zobrazuje kurzor ^ pod editovaným textem		STRING[20]
done	hotovo	výstupní	BOOL

Tento FB zpracovává kódy kláves (0,1,2,3,4,5,6,7,8,9, Left,Right,Esc,Enter,+) a opakovaným stiskem číslic rotuje následující řetězce ASCII znaků, jako v mobilním telefonu :

```

'0 mezerá '
'1 . , ? ! " ' - ( ) @ / : _ ; + & % * = < > [ ] '
'2 A B C a b c '
'3 D E F d e f '
'4 G H I g h i '
'5 J K L j k l '
'6 M N O m n o '
'7 P Q R S p q r s '
'8 T U V t u v '
'9 W X Y Z w x y z '

```

Kurzorové šipky umožňují pohyb kurzoru po textu. Escape klávesa umožňuje vypouštět znak na pozici kurzoru. Klávesa + vloží do textu mezeru. Číselné klávesy opakovaným stiskem přepisují aktuální pozici. Po prodlevě 1,5 sec, nebo při stisku jiné klávesy s číslem se provádí automatický posun kurzoru. klávesa Enter ukončí edici a přepíše výsledek do stringu srchName.

2.2.4. Pomocné funkce

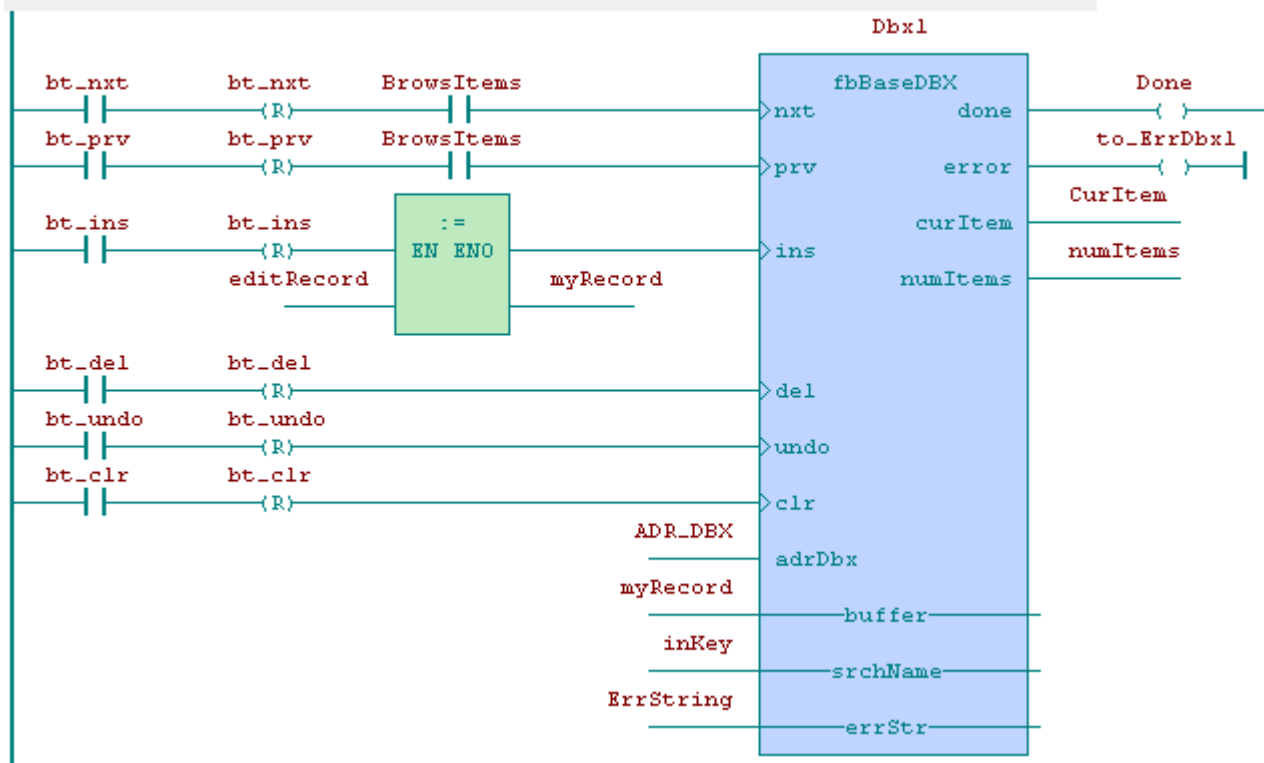
fcCmpStr - porovnání dvou textových řetězců
fcMoveStr - přenesení textového řetězce

interní pomocné funkce použité uvnitř fbBaseDBX :

fcCoutAdr - výpočítává adresu v DBX
fcSubstit_Name - nazení jména kódem Hash
fcFindAdr - vyhledání položky podle adresy v seznamu
fcFindItem - vyhledání položky podle Hash v seznamu

2.3. APLIKACE KNIHOVNY

0002



Poznámky :



teco

Objednávky a informace:

Teco a. s. Havlíčkova 260, 280 58 Kolín 4, tel. 321 737 611, fax 321 737 633

TXV 003 49.01

Výrobce si vyhrazuje právo na změny dokumentace. Poslední aktuální vydání je k dispozici na internetu
www.tecomat.cz