



---

## DLL knihovna pro přímou komunikaci s dekádou

---

### Stručná charakteristika:

DLL knihovna pro přímou komunikaci s dekádou zpřístupňuje funkce pro automatizované ovládání dekády prostřednictvím uživatelského software a USB rozhraní. Pomocí knihovny lze připojit a využívat neomezený počet dekád v rámci jediného aplikačního prostředí. Knihovna je plně kompatibilní se všemi typy dekád řady PROFI a BASIC včetně variant FULL a LITE. Testovací program „ELDEC Interface X.XX – Example“ umožňuje rychlé seznámení s funkcemi knihovny. Na uvolněný zdrojový kód je možné přímo navázat vývojem vlastní aplikace.

### OBSAH

|          |                                                           |          |
|----------|-----------------------------------------------------------|----------|
| <b>1</b> | <b>Knihovní funkce</b>                                    | <b>2</b> |
| 1.1      | Skupina funkcí <b>bool</b>                                | 2        |
| 1.2      | Skupina funkcí <b>String</b>                              | 3        |
| <br>     |                                                           |          |
| <b>2</b> | <b>Testovací program „ELDEC Interface X.XX – Example“</b> | <b>4</b> |
| 2.1      | Popis implementace knihovny                               | 4        |
| 2.2      | Popis testovacího programu                                | 5        |



## 1 Knihovná funkce

Knihovna obsahuje 21 funkcí přímo vázaných na hardware dekády. Návrátová hodnota funkce `GetInterfaceVersion` je jako jediná dostupná před navázáním komunikace dekády s PC prostřednictvím funkce `ConnectDevice`. Skupina funkcí, jejichž návratová hodnota je `String`, vrací parametry připojené dekády získané při navázání komunikace. Skupina funkcí, jejichž návratová hodnota je `bool`, je určena pro přímou komunikaci s dekádou.

### 1.1 Skupina funkcí `bool`

Funkce skupiny `bool` vrací `True`, pokud komunikace s dekádou proběhla úspěšně, v opačném případě vrací `False`.

#### Funkce `bool DeviceON(void)`

Funkce `DeviceON` přepne dekádu do pracovního režimu.

#### Funkce `bool DeviceOFF(void)`

Funkce `DeviceOFF` přepne dekádu do úsporného režimu.

#### Funkce `bool DeviceRestart(void)`

Funkce `DeviceRestart` restartuje dekádu. Během restartu je načítán operační systém DBOS a probíhá kompletní inicializace dekády. Po restartu je dekáda připravena k navázání komunikace pomocí funkce `ConnectDevice`.

#### Funkce `bool ConnectDevice(int USBID, unsigned char* USBFlag)`

Funkce `ConnectDevice` inicializuje komunikační propojení s dekádou. Parametr `USBID` indexuje konkrétní dekádu připojenou k USB sběrnici PC<sup>1)</sup>. Pointer `USBFlag` odkazuje na interní příznak stavu inicializace.

#### Funkce `bool DisconnectDevice(void)`

Funkce `DisconnectDevice` zruší komunikační propojení s dekádou. Dekáda je v rámci operačního systému programově odpojena a připravena pro další využití.

#### Funkce `bool SetMainValue(int value, bool write, unsigned char* protectState)`

Funkce `SetMainValue` zapíše do dekády hodnotu `value` odporu / kapacity / indukčnosti. Pokud je parametr `write` roven `True`, hodnota `value` se aktivuje přímo na výstupních svorkách dekády. Pointer `protectState` odkazuje na hodnotu stavu tepelné ochrany dekády<sup>1)</sup>.

#### Funkce `bool GetWriteValue(int* value, bool* set, unsigned char* protectState)`

Funkce `GetWriteValue` vrací pointer `value`, který odkazuje na hodnotu odporu / kapacity / indukčnosti, která je aktivovaná na výstupních svorkách dekády. Pointer `set` odkazuje na stav aktivace hodnoty na výstupních svorkách dekády, pokud je roven `True`, aktivovaná hodnota odpovídá zapsané hodnotě. Pointer `protectState` odkazuje na hodnotu stavu tepelné ochrany dekády<sup>2)</sup>.



**Funkce** `bool GetTemperature(int* tmp1, int* tmp2, int* tmp3)`

Funkce `GetTemperature` vrací pointer `tmpX`, který odkazuje na hodnotu teploty dané oblasti dekády.

**Funkce** `bool GetExternalInput(bool* input)`<sup>3)</sup>

Funkce `GetExternalInput` vrací pointer `input`, který odkazuje na stav externího vstupu.

**Funkce** `bool EnableProtect(void)`

Funkce `EnableProtect` aktivuje tepelnou ochranu dekády.

**Funkce** `bool DisableProtect(void)`

Funkce `DisableProtect` deaktivuje tepelnou ochranu dekády.

1) Dekády jsou indexovány od 0 inkrementálně, tzn. pokud jsou k PC připojeny 2 dekády, jejich USBID je 0 a 1.

2) Stavy tepelné ochrany dekády: 2 dekáda přetížena, 1 dekáda se zotavuje z přetížení, 0 dekáda připravena

3) Externí vstup je přístupný jen u dekád řady PROFI, stav externího vstupu dekád řady BASIC je vždy `False`.

## 1.2 Skupina funkcí `String`

**Funkce** `String^ GetInterfaceVersion(void)`

Funkce `GetInterfaceVersion` vrací verzi aktuálně použité DLL knihovny.

**Funkce** `String^ GetDeviceConnect(void)`

Funkce `GetDeviceConnect` vrací úplný název připojené dekády.

**Funkce** `String^ GetDeviceLine(void)`

Funkce `GetDeviceLine` vrací název typové řady, ze které dekáda pochází.

**Funkce** `String^ GetDeviceName(void)`

Funkce `GetDeviceName` vrací označení dekády z hlediska její primární veličiny.

**Funkce** `String^ GetDeviceForm(void)`

Funkce `GetDeviceForm` vrací typ konstrukce dekády.

**Funkce** `String^ GetDeviceType(void)`

Funkce `GetDeviceType` vrací konkrétní typ dekády dané typové řady.

**Funkce** `String^ GetDeviceVersion(void)`

Funkce `GetDeviceVersion` vrací označení vývojové verze připojené dekády.

**Funkce** `String^ GetDeviceID(void)`

Funkce `GetDeviceID` vrací jedinečné identifikační označení dekády.

**Funkce** `String^ GetDeviceBooting(void)`

Funkce `GetDeviceBooting` vrací počet spuštění operačního systému dekády.

**Funkce** `String^ GetDeviceTime(void)`

Funkce `GetDeviceTime` vrací stav vnitřních hodin při posledním programovém připojení dekády.



## 2 Testovací program „ELDEC Interface X.XX – Example“

Testovací program „ELDEC Interface X.XX – Example“ slouží pro rychlé seznámení se s funkcemi DLL knihovny. X.XX označuje verzi aktuálně připojené knihovny k testovacímu programu. Zdrojový kód testovacího programu je volně šiřitelný a je ho tak možné využít k vývoji své vlastní aplikace.

### 2.1 Popis implementace knihovny

Vlastní funkce dekády se nacházejí ve jmeném prostoru `EldecInterface`, o který je nutné kód vyvíjené aplikace rozšířit.

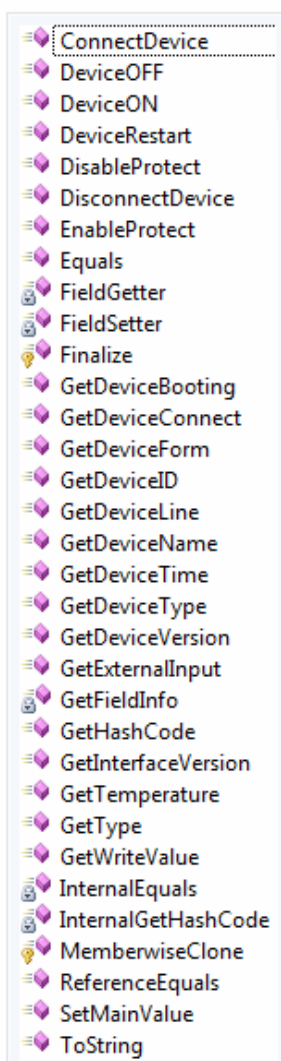
```
using namespace EldecInterface;
```

Následně je nutné deklarovat a vytvořit datový typ `EldecDevice`, který bude svázaný s připojenou dekádou.

```
EldecDevice ^ eldecDevice = gcnew EldecDevice();
```

Dále již můžeme využívat všechny přístupné funkce dekády.

`eldecDevice->`

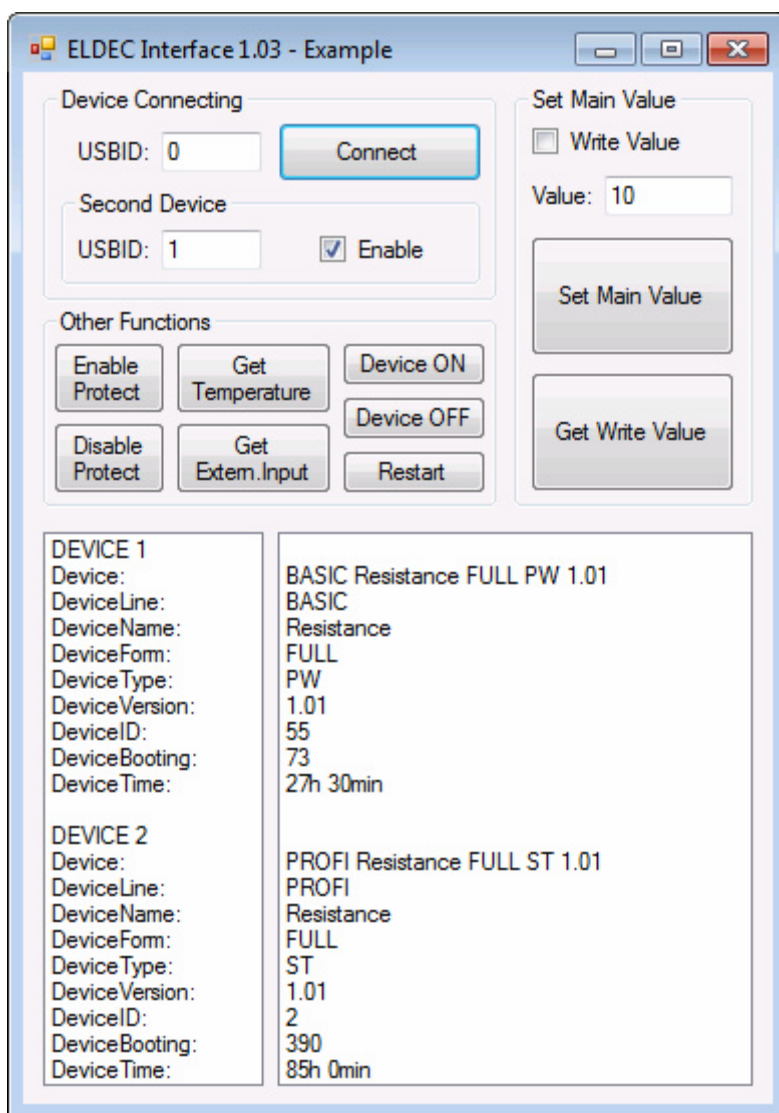


Obr. 2.1 Knihovní funkce



## 2.2 Popis testovacího programu

Uživatelské prostředí testovacího programu umožňuje vytvořit spojení s dvěma z celkového počtu dekád připojených k počítači (jednotlivé dekády jsou identifikovány parametrem USBID). Navázání komunikačního spojení s vyšším počtem dekád probíhá obdobně, proto není demonstrováno. Po inicializaci spojení jsou vypsány kompletní informace o obou typech dekád (pokud je aktivováno použití druhé dekády). Pro samotné testování funkcí dekády je vytvořeno 9 tlačítek, jejichž funkce odpovídají popsaným funkcím knihovny. Pokud je aktivováno použití druhé dekády, jsou všechny funkce testovacího programu prováděny simultánně nad oběma připojenými dekádami. Kód testovacího programu je uvolněn pro vývoj vlastního aplikačního prostředí.



Obr. 2.2 Screen testovacího programu



**[www.odporove-dekady.cz](http://www.odporove-dekady.cz)**



**Copyright © 2013**