



NOO DLL knihovna pro přímou komunikaci s dekádou NI LabVIEW implementace

Poslední úpravy 1.12.2016 (vydání NOO DLL knihovny verze 1.02)

Stručná charakteristika:

NOO DLL knihovna pro přímou komunikaci s dekádou pro neobjektově orientovaná vývojová prostředí. 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. Ukázkovým využitím NOO DLL knihovny je její implementace v prostředí NI LabVIEW.

OBSAH

1	Knihovní funkce	2
1.1	Funkce pro komunikaci s dekádou	2
1.2	Funkce pro komunikaci s DLL knihovnou	3
2	Ukázková implementace v LabVIEW	4



1 Knihovní funkce

Funkce jsou rozděleny do dvou skupin: a) funkce pro komunikaci s dekádou, b) funkce pro komunikaci s DLL knihovnou.

1.1 Funkce pro komunikaci s dekádou

Všechny funkce pro komunikaci s dekádou vrací hodnotu **1**, pokud komunikace s dekádou proběhla úspěšně, v opačném případě vrací hodnotu **0**. Návratová hodnota je rovna **0** také v případě, že daná funkce není podporována konkrétní verzí přístroje (je nutný upgrade FW). Všechny funkce obsahují shodně dva základní parametry. Parametr `USBID` indexuje konkrétní dekádu připojenou k USB sběrnici PC¹⁾. Pointer `USBFlag` odkazuje na interní příznak stavu komunikace.

Funkce `int DevicePWR(int USBID, int* USBFlag, int mode)`

Funkce `DevicePWR` umožňuje přepínat mezi pracovním a úsporným režimem nebo restartovat zařízení. Požadovaná činnost je daná parametrem `mode`. Přejít do úsporného režimu `mode = 0`. Přejít do pracovního režimu `mode = 1`. Restart zařízení `mode = 2`.

Funkce `int SetMainValue(int USBID, int* USBFlag, int value, int write, int* protectState)`

Funkce `SetMainValue` zapíše do dekády hodnotu `value` odporu / kapacity / indukčnosti. Pokud je parametr `write` roven **1**, hodnota `value` se aktivuje přímo na výstupních svorkách dekády. Pointer `protectState` odkazuje na hodnotu stavu tepelné ochrany dekády²⁾.

Funkce `int GetWriteValue(int USBID, int* USBFlag, int* value, int* set, int* 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 **1**, aktivovaná hodnota odpovídá zapsané hodnotě. Pointer `protectState` odkazuje na hodnotu stavu tepelné ochrany dekády²⁾.

Funkce `int GetTemperature(int USBID, int* USBFlag, int* tmp1, int* tmp2, int* tmp3)`

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

Funkce `int GetExternalInput(int USBID, int* USBFlag, int* input)`³⁾

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

Funkce `int GetDeviceID(int USBID, int* USBFlag, int* deviceID)`

Funkce `GetDeviceID` vrací pointer `deviceID`, který odkazuje na jedinečné identifikační označení dekády.



Funkce `int SetDisplay(int USBID, int* USBFlag, int contrast)`⁴⁾

Funkce `SetDisplay` zapíše do dekády hodnotu `contrast`, která je svázána s konkrétním parametrem displeje určeného typovou řadou a variantou přístroje.

PROFI FULL	kontrast displeje v rozsahu 0 - 20
PROFI LITE	intenzita svitu indikační LED 0 - 20
BASIC FULL	intenzita podsvětlení displeje 0 - 1
BASIC LITE	intenzita svitu indikační LED 0 - 1

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 PROFÍ, stav externího vstupu dekád řady BASIC je vždy 0.

4) Funkce je podporovaná od verze přístroje 1.07, pokud máte přístroj verze nižší, je nutný upgrade FW.

1.2 Funkce pro komunikaci s DLL knihovnou

Funkce pro komunikaci s DLL knihovnou nejsou přímo vázány na přítomnost zařízení v komunikačním kanálu, jejich vykonání je vždy úspěšné.

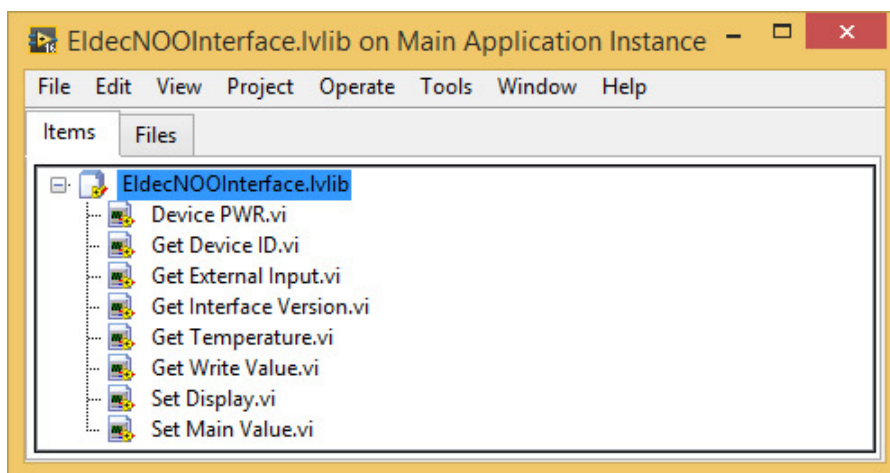
Funkce `int GetInterfaceVersion(void)`

Funkce `GetInterfaceVersion` vrací verzi aktuálně použité DLL knihovny.



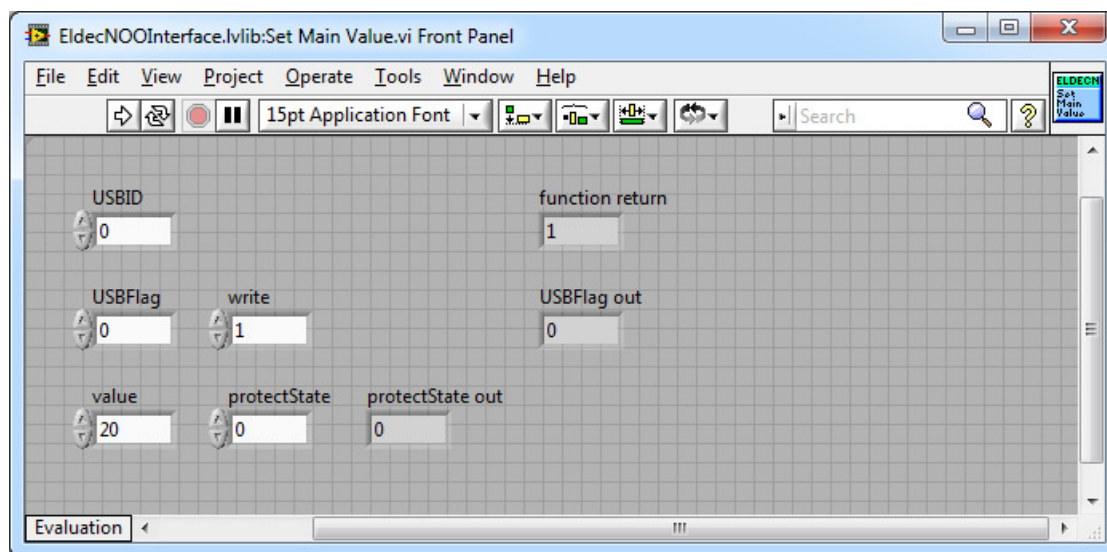
2 Ukázková implementace v LabVIEW

Jednou z typických cílových aplikací NOO DLL knihovny je vývojové prostředí LabVIEW. Pomocí menu Tools/Import/SharedLibrary vygenerujeme z DLL knihovny typ LVLIB přímo podporovaný prostředím LabVIEW.



Obr. 2.1 LVLIB knihovna s výpisem implementovaných funkcí

V oknech jednotlivých funkcí můžeme pro prvotní vyzkoušení komunikace spustit samotnou funkci a ověřit její návratové hodnoty. Dále již dle zvyklostí prostředí LabVIEW.



Obr. 2.2 Spuštění funkce Set Main Value – úspěšné nastavení hodnoty odporu 20ohm



MARZ
MIKRO **ARZ**.com

www.odporove-dekady.cz



Copyright © 2016