

**Nadstavbový ovladač  
UDAQ modulů  
2. generace  
pro Windows**

**Programátorská příručka**

[illegible]

**Výhrada odpovědnosti, autorských práv, ochranných známek a názvů:**

Ačkoliv byla tato programátorská příručka vytvořena s maximální pečlivostí, nelze vyloučit, že obsahuje chyby. Domníváte-li se, že jsou některé údaje uvedeny nesprávně, neúplně nebo nepřesně, prosíme, informujte technickou podporu.

Pro případ typografických nebo obsahových chyb si TEDIA® vyhrazuje právo kdykoliv provést opravy nebo zpřesnění publikovaných informací. Právě tak produkty popsané v programátorské příručce mohou být kdykoliv revidovány se záměrem zlepšení technických parametrů nebo dosažení lepších užitečných vlastností. Doporučujeme proto před každým užitím této příručky ověřit, zda není k dispozici vydání nové.

TEDIA® nezodpovídá za žádné škody vzniklé užitím této programátorské příručky nebo informací v příručce obsažených.

Programátorská příručka a její součásti jsou autorským dílem chráněným ustanovením zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

Všechna jména a názvy použité v textu mohou být chráněnými známkami nebo obchodními názvy výrobků příslušných firem.

# **OBSAH**

---

## **1. Základní informace**

- 1.1 Úvod
- 1.2 Kde získat další informace, technická podpora
- 1.3 Doporučení

## **2. Obecné informace a terminologie**

- 2.1 Úvod
- 2.2 Terminologie
- 2.3 Vnitřní struktura modulů
- 2.4 Struktura systémového a nadstavbového ovladače
- 2.5 Protokol UP4
- 2.6 Protokol UP1024/1040
- 2.7 Vlastnosti UDAQ modulů z pohledu datových přenosů

## **3. Popis ovladače a programátorského rozhraní**

- 3.1 Úvod
- 3.2 Součásti ovladače
- 3.3 Funkční návaznosti od modulu k programu
- 3.4 Přehled funkcí ovladače
- 3.5 Popis využití ovladače
- 3.6 Specifické vlastnosti USB modulů a ovladače

## **Příloha A, základní funkce**

- A1.1 Základní veřejné funkce knihovny TEDIA\_TUDFT2.DLL verze 1.0.0
- A1.2 Návrátové kódy funkcí knihovny TEDIA\_TUDFT2.DLL verze 1.0.0
- A2.1 TUDFT2\_Init
- A2.2 TUDFT2\_Done
- A2.3 TUDFT2\_InterfaceVersion
- A2.4 TUDFT2\_DriverVersion
- A2.5 TUDFT2\_DevicesInSystem
- A2.6 TUDFT2\_Device\_VID
- A2.7 TUDFT2\_Device\_PID
- A2.8 TUDFT2\_Device\_SerNo
- A2.9 TUDFT2\_Device\_TypeNo
- A2.10 TUDFT2\_Open
- A2.11 TUDFT2\_Close
- A2.12 TUDFT2\_GetDeviceStatus
- A2.13 TUDFT2\_DeviceReset
- A2.14 TUDFT2\_SendCommand
- A2.15 TUDFT2\_ReadCommandResponse
- A2.16 TUDFT2\_GetDataBuffersStatus
- A2.17 TUDFT2\_GetDeviceBuffersStatus
- A2.18 TUDFT2\_ReadDataBuffer
- A2.19 TUDFT2\_WriteDataBuffer
- A2.20 TUDFT2\_StartStopDataBufferTransfer
- A2.21 TUDFT2\_ClearDataBuffers
- A2.22 TUDFT2\_Callback

Prázdná strana

## 1. Základní informace

---

### 1.1 Úvod

Tato programátorská příručka je věnována popisu nadstavbového ovladače DAQ USB modulů osazených dvoukanálovým high-speed řadičem FT232H a implementovanými protokoly UP4 a UP1024/1040.

Příručka není určena běžnému uživateli modulů, který potřebuje ovladače nainstalovat a dále využívat s již hotovými programy.

Programátorská příručka obsahuje ...

- popis struktury UDAQ modulů s ohledem na implementaci protokolů a vlastnosti systémového ovladače;
- charakteristiku použitých datových protokolů UP4 a UP1024/1040;
- popis API nadstavbového ovladače;
- postup, jak s využitím nadstavbového ovladače vytvořit vlastní program, případně specifický ovladač.

Co tato příručka neobsahuje ...

- popis instalace systémového ani nadstavbového ovladače;
- popis vlastností konkrétních UDAQ modulů.

Pro využití nadstavbového ovladače je potřebná znalost registrové struktury zvoleného UDAQ modulu popsané zpravidla v programátorské příručce zvoleného modulu.

### 1.2 Kde získat další informace, technická podpora

Další užitečné informace lze získat na adrese...

URL: <http://www.tedia.cz>

V případě nejasností se lze obrátit na technickou podporu výrobce:

adresa: TEDIA spol. s r. o., Zábělská 12, 312 11 Plzeň, Česká republika

URL: <http://www.tedia.cz/podpora>

telefon: +420 373730426

Doporučujeme seznámit se s užitečnými pravidly pro kontaktování technické podpory (viz výše uvedená URL).

**Poznámka:** Ačkoliv byla tato programátorská příručka vytvořena s maximální pečlivostí, nelze vyloučit, že obsahuje chyby. Domníváte-li se, že jsou některé údaje uvedeny nesprávně, neúplně nebo nepřesně, prosíme, informujte technickou podporu.

### 1.3 Doporučení

Ačkoliv rozsah příručky může čtenáře-programátora svádět k přeskočení úvodních kapitol, vřele doporučujeme věnovat pár minut jejich přečtení. Ze zkušenosti můžeme říci, že bez znalostí a pochopení vnitřní struktury modulů, principu řízení přístroje a vlastností datových přenosů apod. lze jen stěží úspěšně vytvořit spolehlivě pracující program.

Snažili jsme se o maximální stručnost a srozumitelné vysvětlení, v případě nejasností nebo potřeby informace rozšířit se můžete obrátit na technickou podporu, ochotně vysvětlíme, co bylo popsáno příliš stručně nebo nedostatečně.

## 2. Obecné informace a terminologie

### 2.1 Úvod

Dále uvedené odstavce jsou věnovány obecnému popisu UDAQ modulů TEDIA, resp. popisu pojmů použitých v dalších kapitolách této příručky.

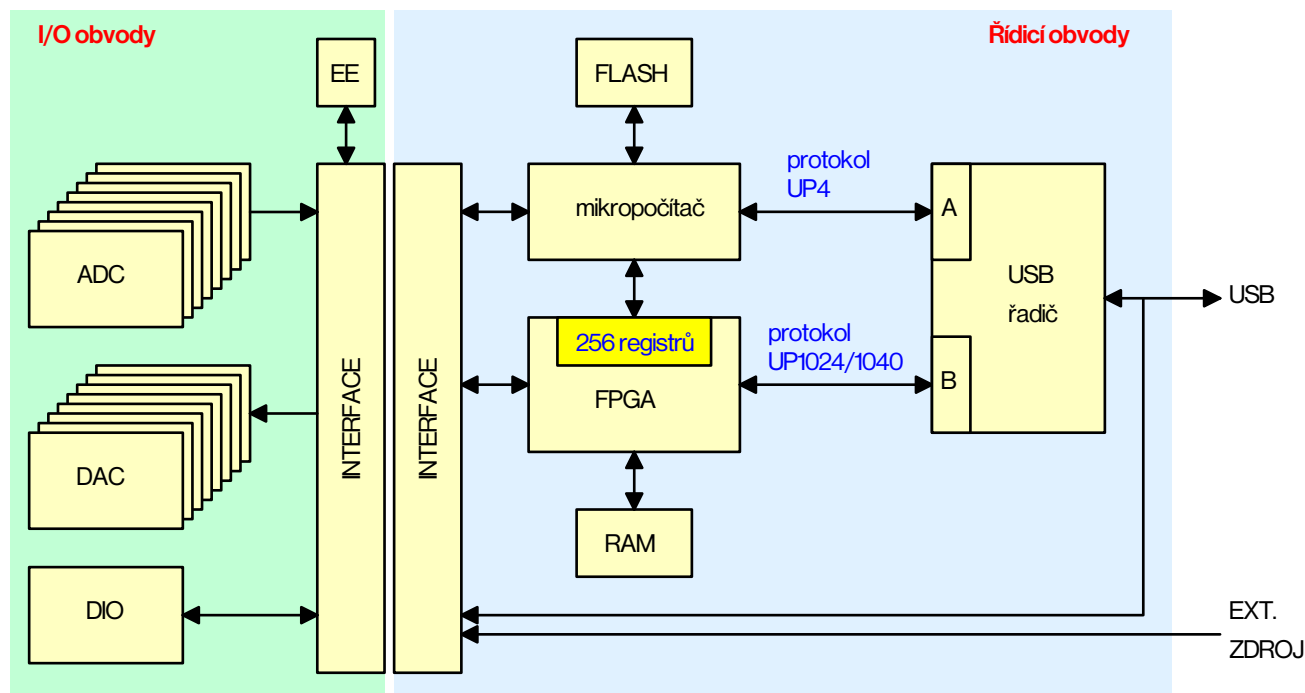
### 2.2 Terminologie

V popisu ovladače je používána řada pojmů, jejichž význam nemusí být každému uživateli zcela zřejmý. Základní z nich jsou popsány v následujících odstavcích:

systémový ovladač	je software obsahující základní obsluhu v jádře Windows a jako rozhraní pro běžné programy (resp. aplikační ovladače) využívá běžnou DLL knihovnu; systémový ovladač UDAQ modulů je modifikovaným ovladačem vyvíjeným firmou FTDI (výrobce USB řadiče)
nadstavbový ovladač	je software, který využívá funkce systémového ovladače (směrem k modulu) a v opačném směru poskytuje specifické funkce a rozhraní pro různé vývojové prostředky a systémy; žádná jeho část neběží v jádře Windows; <b>v dalším textu zkrácený výraz "ovladač" vždy znamená tento nadstavbový ovladač</b>
TEDIA_TUDFT2.DLL	je API DLL nadstavbového ovladače
VID	USB Vendor ID; číslo, které přiděluje USB Implementers Forum, Inc., svým členům UDAQ moduly podporované ovladačem používají buď VID=0403 <sub>H</sub> (VID firmy FTDI vyrábějící USB řadič FT232RL), nebo VID=A600 <sub>H</sub> (VID firmy ASIX, částečně využívá i TEDIA)
PID	USB Product ID; číslo, které USB zařízení přidělí držitel VID v kombinaci s VID=0403 <sub>H</sub> je použito výhradně PID=FB72 <sub>H</sub> v kombinaci s VID=A600 <sub>H</sub> je využívána celá sada hodnot
SerNo	osmimístné číslo (popř. textový řetězec obsahující pouze číslice 0-9) složené z třímístného čísla typu přístroje (nejvyšší tři pozice) a pětímístného pořadového čísla přístroje; číslo je naprogramováno v přístroji a je dostupné API funkcemi nadstavbového ovladače
TypeNo	trojmístné číslo definované třemi nejvyššími číslicemi z osmimístného výrobního čísla SerNo; je dostupné API funkcemi nadstavbového ovladače
Zařízení	je USB zařízení detekované FTD2xx ovladačem; v případě dvoukanálových USB řadičů je každý přístroj v systémovém ovladači reprezentován dvěma USB zařízeními označenými "nnnnnnnnA" a "nnnnnnnnB", kde "nnnnnnnn" je výrobní číslo modulu (SerNo); z pohledu popisu datových přenosů jsou rovněž použita označení "kanál A" a "kanál B"
Modul	je UDAQ přístroj využívající jeden pár A/B zařízení
UP4	komunikační protokol použitý zařízením "A" (tzn. kanál A) pro datové přenosy mikropočítače (firemní protokol TEDIA, specifikace je veřejná)
UP1024/1040	komunikační protokol použitý zařízením "B" (tzn. kanál B) pro datové přenosy FPGA (firemní protokol TEDIA, specifikace je veřejná)

## 2.3 Vnitřní struktura modulů

Moduly jsou složeny (buď jen principiálně, nebo i fakticky v případě modulární platformy) ze dvou částí - řídicími obvody a I/O obvody. Blokové schéma je zakresleno na obrázku níže.



Řídicí obvody jsou navrženy s ohledem na maximální přenositelnost mezi jednotlivými typy modulů a obsahují...

- dvoukanalový high-speed USB řadič FT2232H (dva datové kanály označené A a B) umožňující realizovat dva nezávislé obousměrné datové toky mezi počítačem a přístrojem
- řídicí mikropočítač s implementovaným protokolem UP4 komunikující s počítačem kanálem A; mikropočítač řeší i servisní funkce (zpřístupnění obsahu paměti Flash počítači, konfigurace FPGA firmware uloženým ve Flash paměti, přenos FIR koeficientů a dalších pracovních konstant z Flash paměti do FPGA, zpracování kalibračních konstant uložených v EEPROM paměti umístěné na I/O desce, systémový dohled nad činností modulu); firmware mikropočítače je v maximální míře unifikované pro všechny typy modulů
- hradlové pole typu FPGA s implementovaným protokolem UP1024/1040 komunikujícím s počítačem kanálem B; firmware FPGA je za běhu ovládán prostřednictvím 256 osmibitových registrů dostupných počítači prostřednictvím povolů UP4 protokolu
- datové zásobníky naměřených a generovaných dat tvořených pamětí RAM (obsluhovanou FPGA)
- unifikovaný interface pro připojení I/O obvodů
- napájecí obvody; primárně jsou přístroje napájeny z USB portu, alternativně (v případě I/O obvodů s vyššími nároky na příkon) souběžně z externího napájecího zdroje

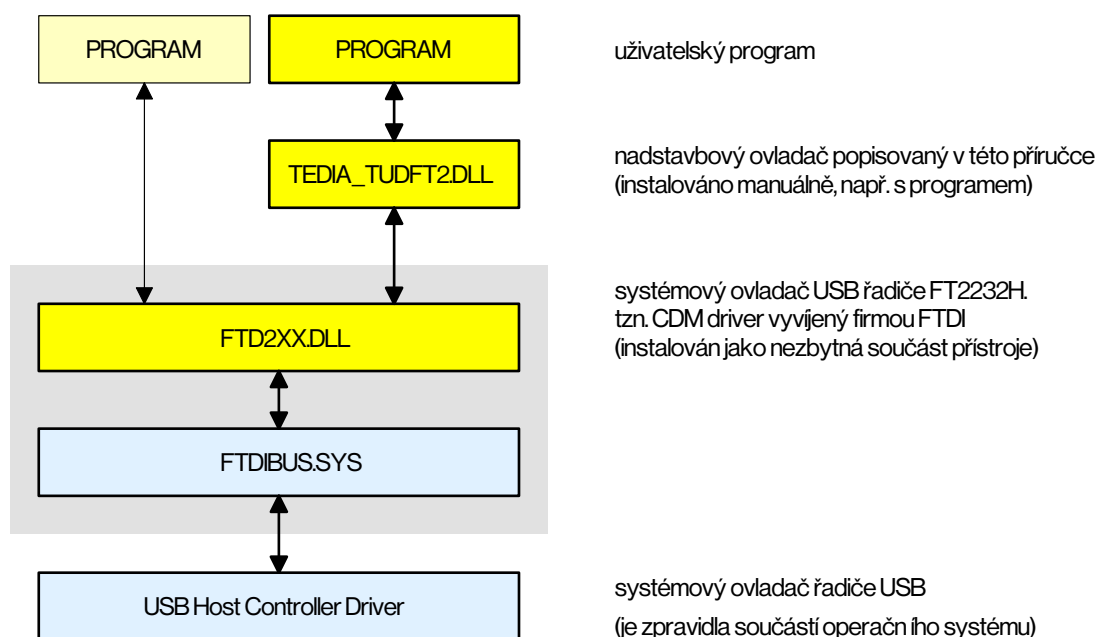
Celá struktura hardware i firmware je podřízena hlavnímu cíli - široce použitelný interface pro I/O obvody s unifikovaným řízením pomocí dvou datových kanálů tak, aby veškeré specifické vlastnosti modulů byly implementovány v FPGA na straně přístroje, resp. v uživatelském programu na straně počítače. Firmware mikropočítače, datové protokoly a ovladač přístroje jsou navrhovány společně pro všechny typy.

Ovladač poskytuje program obecné funkce umožňující přístup k identifikačním údajům a otevření spojení s vybraným modulem, přístup k registrům FPGA, spouštění procesů ovladače pro přenos naměřených a generovaných dat mezi zásobníky přístroje a datovými strukturami v ovladači, atd.

Popsané řešení usnadňuje vývoj firmware FPGA (definuje funkcionalitu modulu) a umožňuje jednoduchý update uživatelem do Flash paměti přístroje, aniž by bylo nutné modifikovat firmware mikropočítače nebo ovladač. Tyto dvě vlastnosti spolu s abstraktním řešením ovladače pak zásadně zjednodušují implementaci dodatečných zakázkových funkcí, případně vývoj nových zakázkových přístrojů.

## 2.4 Struktura systémového a nadstavbového ovladače

Návaznost jednotlivých částí ovladačů a uživatelského programu je zakreslena na obrázku níže.



Systémový ovladač UDAQ modulů je funkčně identický s originálním ovladačem vyvíjeným firmou FTDI (výrobce USB řadiče), modifikace se týká pouze seznamu podporovaných zařízení v INF souboru. Tento ovladač je tvořen FTDIBUS.SYS spuštěným v jádře Windows (komunikuje se systémovým ovladačem řadiče USB portu počítače) a knihovnou FTD2XX.DLL vytvářející API rozhraní pro běžné programy.

Uživatelské programy mohou využívat přímo rozhraní FTD2XX.DLL (v takovém případě však musí přímo podporovat protokoly UP4 a UP1024/1040), nebo využít jako mezivrstvu knihovnu TEDIA\_TUDFT2.DLL (tzn. nadstavbový ovladač popisovaný v této příručce) "zabaluující" oba protokoly do uživatelsky přívětivějšího rozhraní bez potřeby jejich znalosti (zcela postačuje obecná představa u jejich vlastnostech popsáná v této příručce).

## 2.5 Protokol UP4

Protokol UP4 je jednoduchý protokol pracující metodou dotaz - odpověď mezi zařízením typu "master" (typicky počítač, zařízení zahajuje komunikaci vysláním dotazu) a zařízením typu "slave" (v našem případě UDAQ modul). Typicky je použitelný pro jednorázové přístupy k I/O linkám, konfiguraci streamovaného měření/generování, apod.

**Protokol UP4 se vyznačuje...**

- komunikací typu dotaz - odpověď mezi zařízením typu "master" (typicky počítač; zahajuje komunikaci vysláním dotazu) a zařízením typu "slave" (v našem případě UDAQ modul; odpovídá na dotazy)
- sadou povelů umožňujícími přistupovat k interním obvodům přístroje
- strukturou umožňující nenáročnou implementaci do mikropočítačů a využití celé řady komunikačních tras (sériová linka, USB port, LAN při využití UDP nebo TCP)

**Protokol UP4 obsahuje povel...**

- **pro podpůrné funkce** (restart přístroje, načtení identifikačních údajů, vložení prodlevy mezi instrukce, atd.)  
tyto povelvy využívá interně ovladač a výsledky zpřístupňuje programu prostřednictvím API funkcí
- **pro servisní funkce** (přístup k pamětem Flash a EEPROM s adresovým prostorem 4 GB; obsahují firmware FPGA, kalibrační konstanty, identifikační údaje, atd.)  
tyto povelvy jsou využívány jen speciálními programy pro upgrade firmware, kalibraci modulů apod.
- **pro provozní funkce** (přístup k registrům FPGA s adresovým prostorem 256 B; pro každý typ modulu je k dispozici podrobný popis registrové struktury); tento typ povelvy umožňuje sdružovat mnoho desítek instrukcí čtení a zápisu do jednoho povelvy, komunikaci tak zásadně urychlit a zajistit integritu operací s registry  
tyto povelvy využívá program prostřednictvím API funkcí ovladače

Přenosy jsou zajištěny (kromě dalších mechanismů zajištění integrity dat na úrovni přenosové linky) kódováním znaků a kontrolním součtem celé zprávy. V případě zachování elementární syntaxe (tzn. start/stop značky a platný kontrolní součet) koncové zařízení vždy odpovídá na dotaz a eliminuje tak prodlevy v komunikaci vlivem timeoutů.

## 2.6 Protokol UP1024/1040

Protokol UP1024/1040 je jednoduchý duplexní a symetrický protokol pro streamovaný přenos dat vyžadující zpravidla druhý datový kanál pro řízení přenosů (není-li uvažován trvalý kontinuální tok dat).

**Protokol UP1024/1040 se vyznačuje...**

- duplexním datovým tokem mezi dvěma zařízeními, z nichž ani jedno není nadřazeno druhému (z tohoto pohledu je komunikace symetrická), datové toky probíhají v každém směru nezávisle
- paketem pevné délky 1040 B děleným na datovou část délky 1024 B (tzv. payload; datová část nemusí být nezbytně plně využita daty) a kontrolní patičku délky 16 B
- kontrolní patička obsahuje diagnostické informace vysílače paketu, zejména:
  - kontrolní pořadové číslo paketu
  - kontrolní součet datové části paketu
  - zaplnění datové části paketu
  - zaplnění datových zásobníků vysílající strany pro oba směry přenosů (význam především pro stranu přístroje)
  - kritické stavové příznaky (přetečení/podtečení zásobníků, fatální chyba I/O obvodů, apod.)
- strukturou umožňující implementaci zejména do FPGA nebo výkonných mikropočítačů a využití komunikačních tras s dostatečným zajištěním integrity dat (USB, LAN při využití TCP)

Protokol neposkytuje žádné mechanismy přijímající straně, kterými by mohla ovládat obsah nebo časování paketů vysílající strany; jedinou výjimkou je informování vysílající strany o stavu zaplnění vlastních zásobníků vysíláním paketů (nemusí obsahovat žádná data; tedy nevyužitý payload a patičku). V ideálním případě je přijímací strana schopna přijímat pakety v kterýkoliv okamžik včetně zpracování kontinuálního datového toku.

*Poznámka: V případě kombinace UDAQ modulu a počítače lze předpokládat (vzhledem k neporovnatelně vyššímu výkonu a dostupné paměti počítače), že počítač bude schopen přijímat data z modulu kdykoliv a bez omezení datového toku, zatímco modul bude pakety přenášenými do počítače informovat o zaplnění svých zásobníků a počítač pak přizpůsobí datový tok vysílaných dat možností modulu.*

Pro konfiguraci datového toku (není-li uvažován trvalý kontinuální tok) je nutné využít jiný komunikační kanál, v případě UDAQ modulů jde datový kanál A s protokolem UP4.

## 2.7 Vlastnosti UDAQ modulů z pohledu datových přenosů

Z pohledu uživatelského programu lze najít několik parametrů podstatných pro řešení uživatelského programu.

- délka datového zásobníku v přístroji pro zpracování zpráv UP4 paketu  
všechny moduly disponují pamětí umožňující provést v rámci jednoho povelu až 200 operací čtení nebo zápis (resp. libovolné kombinace čtení a zápisů) do registrů FPGA
- velikost zásobníků pro datové přenosy kanálem B  
moduly jsou osazeny RAM s kapacitou závislou na typu modulu (lze zjistit funkcemi ovladače; aktuálně vyráběné moduly mají zásobníky velikosti od 256 kB do 4 MB)
- pakety ve směru z přístroje do PC jsou přenášeny "vždy"  
pakety směřující do PC obsahují informace o zaplnění zásobníků modulu a základní chybové příznaky, proto jsou přenášeny i v případě, že probíhá jen generování (cca 10x za sekundu), resp. probíhá současně měření s nízkým datovým tokem (jsou přenášeny pakety s částečným zaplněním); tato vlastnost garantuje, že proces ovladače obsluhující přenosy dat B kanálem má k dispozici stavové informace vždy, aniž by musel občasně blokovat komunikaci programem kanálem A

### 3. Popis ovladače a programátorského rozhraní

#### 3.1 Úvod

Dále uvedené odstavce jsou věnovány obecnému popisu ovladače a implementovaným funkcím.

#### 3.2 Součásti ovladače

Vlastní ovladač je tvořen následujícími soubory...

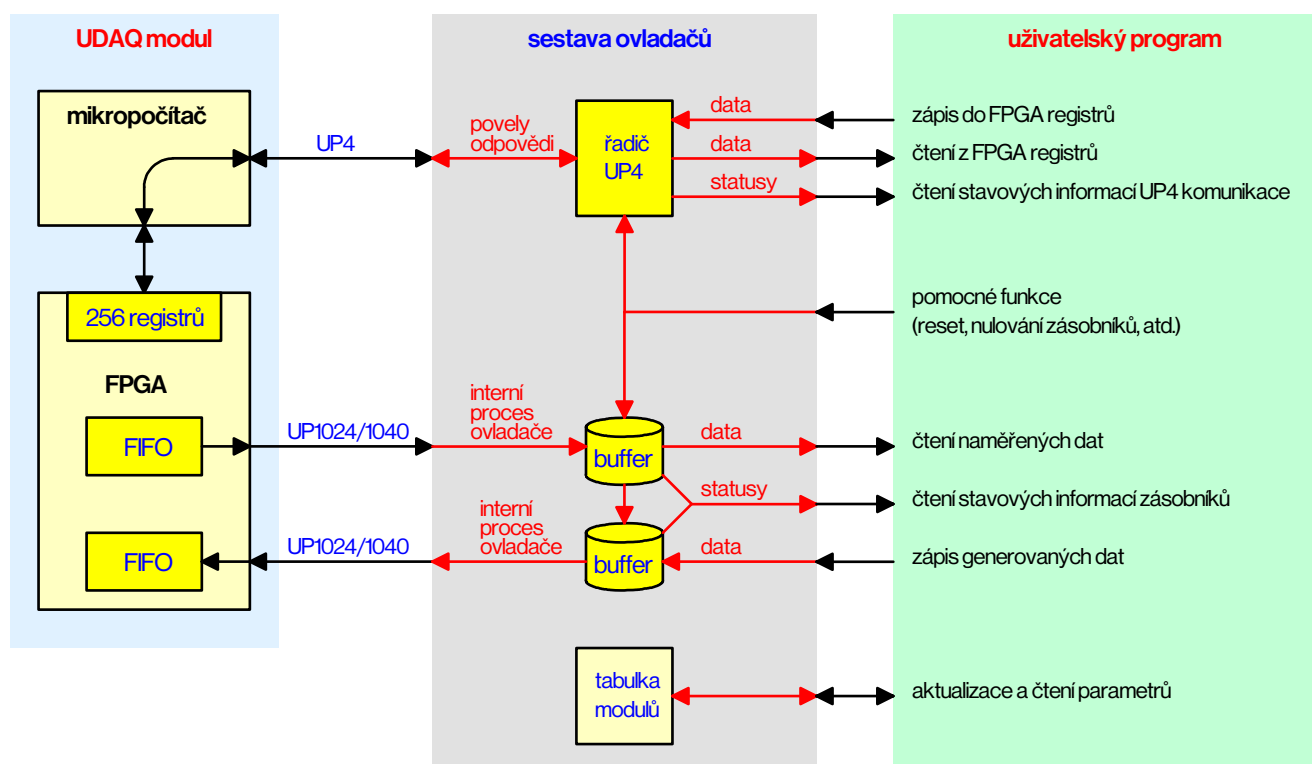
TEDIA\_TUDFT2.DLL      interface pro uživatelský program (DLL dostupná v 32bitové i 64bitové verzi)

hlavičkové soubory      hlavičkové soubory pro C++ a Delphi

Soubory jsou k dispozici volně, nejsou integrovány do instalačního balíku.

#### 3.3 Funkční návaznosti od modulu k programu

Na obrázku níže je zakreslena návaznost jednotlivých částí řetězce od modulu přes ovladače až k uživatelskému programu včetně principiálních datových toků a způsobů řízení.



Jak je z obrázku patrné, software v PC komunikuje jedním kanálem s mikropočítačem a druhým kanálem s FPGA, resp. datovými zásobníky typu FIFO vytvořenými z RAM a obslužného firmwaru.

Na straně PC obsluhuje komunikaci s mikropočítačem **řadič UP4** protokolu implementovaný v ovladači; uživatelský program má možnost využít jeho funkce pro přístup k registrům FPGA pomocí dvou implementovaných funkcí (bez potřeby znalosti UP4 protokolu) a volitelné callback funkcionality. Rozšířené API (další dvě funkce) umožňuje pracovat s UP4 protokolem i na obecné úrovni.

Druhý komunikační kanál s protokolem UP1024/1040 je kompletně obsluhován interními procesy ovladače (tzn. přenos dat mezi FIFO zásobníky modulu a interními zásobníky ovladače) a uživatelský program má k dispozici pouze funkce pro zápis a čtení dat ze zásobníků ovladače, resp. funkce pro zjištění stavu zaplnění jednotlivých zásobníků a stavu procesů přenášejících data. Uživatelský program je také zodpovědný za spuštění/zastavení těchto procesů ovladače (vyhrazená funkce API) v návaznosti na start/stop měření a generování v modulu (obsluha registrů FPGA).

Další část funkcí umožňuje uživatelskému programu vytvořit v ovladači tabulku dostupných modulů a následně z ní načítat identifikační údaje tak, aby mohl vybrat požadovaný modul a otevřít spojení.

Poslední skupina funkcí umožňuje vyvolat reset modulu, nulovat příznaky vytvářené v ovladači a nulovat zásobníky.

### 3.4 Přehled funkcí ovladače

V tabulkách uveden níže jsou podle významu seřazeny funkce knihoven, jejich podrobnému popisu jsou věnovány přílohy této příručky.

Obecné funkce bez návaznosti na modul (použitelné i před otevřením spojení s modulem)	
funkce	popis
TUDFT2_Init	párové funkce pro vytvoření (resp. zrušení) instance ovladače v paměti pro program
TUDFT2_Done	
TUDFT2_InterfaceVersion	funkce pro zjištění verze nadstavbového ovladače a verze systémového ovladače
TUDFT2_DriverVersion	
Funkce pro zjištění vlastností modulu (použitelné i před otevřením spojení s modulem)	
funkce	popis
TUDFT2_DevicesInSystem	funkce pro zjištění počtu připojených modulů obsluhovaných ovladačem (vytváří tabulku parametrů uvnitř ovladače) a zjištění parametrů jednotlivých modulů z tabulky
TUDFT2_Device_VID	
TUDFT2_Device_PID	
TUDFT2_Device_SerNo	
TUDFT2_Device_TypeNo	
Funkce související s otevřením/zavřením spojení s modulem	
funkce	popis
TUDFT2_Open	párové funkce pro otevření a uzavření spojení se zvoleným modulem
TUDFT2_Close	
Funkce pro obsluhu modulu (vyžadují otevření spojení s modulem)	
funkce	popis
TUDFT2_GetDeviceStatus	funkce pro zjištění aktuálního stavu interních obvodů modulu
TUDFT2_DeviceReset	funkce pro vyvolání interního restartu modulu
Funkce pro komunikaci servisním rozhraním (protokol UP4, vyžadují otevření spojení s modulem)	
funkce	popis
TUDFT2_SendCommand	párové funkce pro vyslání UP4 paketu vytvořeného ze sady instrukcí zadaných programem a přijetí odpovědi
TUDFT2_ReadCommandResponse	
Funkce pro přenosy datovým rozhraním (protokol UP1024/1040, vyžadují otevření spojení s modulem)	
funkce	popis
TUDFT2_GetDataBuffersStatus	funkce pro zjištění stavu zásobníků ovladače a zásobníků modulu
TUDFT2_GetDeviceBuffersStatus	
TUDFT2_ReadDataBuffer	funkce pro čtení dat ze zásobníku ovladače, resp. zápis dat do zásobníku ovladače
TUDFT2_WriteDataBuffer	
TUDFT2_StartStopDataBufferTransfer	funkce pro spouštění/zastavení procesů obsluhujících zásobníky
TUDFT2_ClearDataBuffers	funkce pro nulování zásobníků
Funkce pro callback funkcionalitu	
funkce	popis
TUDFT2_Callback	definice funkce umístěné v uživatelském programu volané vybranými událostmi ovladače

### 3.5 Popis využití ovladače

Ovladač má párové funkce *TUDFT2\_Init* a *TUDFT2\_Done*. První funkce vytvoří v paměti instanci ovladače pro program a inicializuje ovladač, druhá funkce ukončí instanci ovladače a uvolní prostor v paměti.

Po vytvoření instance ovladače program použije sadu funkcí pro zjištění aktuálně připojených modulů. Funkce *TUDFT2\_DeviceInSystem* zajistí vytvoření tabulky dostupných modulů v paměti ovladače a předá programu informaci o počtu dostupných modulů. Sada funkcí *TUDFT2\_Device\_VID*, *TUDFT2\_Device\_PID*, *TUDFT2\_SerNo*, *TUDFT2\_TypeNo* a *TUDFT2\_Status* slouží k načtení všech identifikačních údajů všech dostupných modulů z tabulky.

Program zvolí modul a otevře s ním spojení funkcí *TUDFT2\_Open* (funkce vrací handle) a od tohoto okamžiku může program využívat všechny funkce pro přístup k registrům FPGA (popis registrů a postup jejich obsluhy je popsán v programátorské příručce konkrétního modulu), resp. přístup k datovým zásobníkům uvnitř ovladače.

Pro uzavření spojení s modulem je vyhrazena funkce *TUDFT2\_Close*.

Následně může program buď vybrat nový modul a otevřít s ním spojení (tzn. vrátit se v postupu popsáném v předešlých odstavcích k funkci *TUDFT2\_DeviceInSystem*), nebo ukončit instanci ovladače funkcí *TUDFT2\_Done*.

*Poznámka: V rámci jedné instance ovladače může být otevřeno v jeden okamžik spojení jen s jedním modulem. Potřebuje-li program pracovat s více moduly současně, musí vytvořit odpovídající počet instancí ovladače.*

Znázornění posloupnosti využití základních funkcí je zobrazeno v tabulce níže; žlutou barvou je vyznačena existence instance ovladače, zelenou barvou pak otevřené spojení s modulem.

Sekvence volání funkcí v programu	
<i>TUDFT2_Init</i>	první volaná funkce, spustí ovladač v programu
<i>TUDFT2_InterfaceVersion</i> <i>TUDFT2_DriverVersion</i>	program může zjistit verze ovladačů
<i>TUDFT2_DeviceInSystem</i>	program zjistí počet připojených modulů (funkce vytvoří v ovladači tabulku identifikačních údajů a programů poskytne informaci o počtu)
<i>TUDFT2_Device_VID</i> <i>TUDFT2_Device_PID</i>	program může pro každý modul zjistit USB údaje z tabulky (větší význam ale mají SerNo, resp. TypeNo)
<i>TUDFT2_Device_SerNo</i> <i>TUDFT2_Device_TypeNo</i>	program pro každý modul zjistí typ a výrobní číslo a vybere "svůj" modul (údaje z tabulky)
<i>TUDFT2_Open</i>	program otevře spojení s vybraným modulem (pokud to lze)
<i>TUDFT2_SendCommand</i> <i>TUDFT2_ReadCommandResponse</i>	program se znalostí registrové mapy zvoleného modulu obsluhuje registry modulu (ovládá DIO porty, konfiguruje měření a generování, spouští měření a generování, atd.)
<i>TUDFT2_StartStopDataBufferTransfer</i> <i>TUDFT2_ReadDataBuffer</i> <i>TUDFT2_WriteDataBuffer</i>	program může spustit obslužná vlákna ovladače pro měření a generování, přijímat naměřená data a vysílat generovaná data
<i>TUDFT2_Close</i>	program uzavře spojení s modulem
.....	program může otevřít spojení s jiným modulem (tzn. vrátit se na začátek k vytvoření tabulky)
<i>TUDFT2_Done</i>	...nebo ukončit běh ovladače v programu

### 3.6 Specifické vlastnosti USB modulů a ovladače

Tento odstavec je určen zejména současným uživatelům DAQ karet pro sběrnici PCI nebo PCI Express.

Při používání DAQ modulů pro rozhraní USB je potřeba počítat i s chováním vyplývajícím z principu jejich připojení k počítači, resp. způsobu napájení.

- modul může v průběhu měření "zmizet"

Zatímco v případě karty "ztráta" karty v systému znamená fatální chybu, v případě USB jde o provozní stav, kdy dojde k odpojení USB kabelu, vypnutí USB hubu, ale také k narušení komunikace a následnému odpojení ze systému vlivem elektromagnetického rušení v okolí kabelu a přístroje.

- tabulka se může změnit

Mezi voláním funkce *TUDFT2\_DeviceInSystem* a *TUDFT2\_Open* může dojít k odpojení zvoleného modulu a na jeho index může být přesunut jiný; je potřeba, aby časová prodleva mezi vykonáním těchto funkcí nebyla dlouhá. Funkce *TUDFT2\_Open* vytváří interně novou tabulku zařízení a ověřuje, že aktuálně dostupný modul odpovídá modulu ve "staré" programem vytvořené tabulce.

- I/O linky modulu mohou přestat fungovat

Problém je podobný předešlému; pokud USB modul využívá pro napájení externí zdroj, je nutné předpokládat možnost jeho vypnutí nebo odpojení.

- doba přístupu do registrů je dlouhá a není přesně definovatelná

Přístup do registrů FPGA (resp. jakákoliv komunikace) může proběhnout za půl milisekundy, ale může také trvat desítky milisekund podle okamžité vytíženosti systému a "ochoty" se věnovat obsluze USB komunikace.

Prázdná strana

## A1.1 Základní veřejné funkce knihovny TEDIA\_TUDFT2.DLL verze 1.0.0

V tabulce níže je uveden přehled všech veřejných funkcí knihovny, podrobný popis je pak uveden na následujících stranách této přílohy.

Obecné funkce bez návaznosti na modul (použitelné i před otevřením spojení s modulem)	
funkce	popis
TUDFT2_Init	vytvoří v paměti instanci ovladače pro program, inicializuje ovladač
TUDFT2_Done	ukončí instanci ovladače a uvolní prostor v paměti
TUDFT2_InterfaceVersion	poskytne verzi ovladače (tzn. verzi TEDIA_TUDFT2.DLL)
TUDFT2_DriverVersion	poskytne verzi systémového ovladače (tzn. ovladače FTD2xx)
Funkce pro zjištění vlastností modulu (použitelné i před otevřením spojení s modulem)	
funkce	popis
TUDFT2_DevicesInSystem	zjistí počet připojených modulů obsluhovaných ovladačem, načte identifikační + stavové údaje o modulech do interní tabulky ovladače a programu poskytne informaci o počtu modulů
TUDFT2_Device_VID	poskytne hodnotu USB VID vybraného modulu z tabulky
TUDFT2_Device_PID	poskytne hodnotu USB PID vybraného modulu z tabulky
TUDFT2_Device_SerNo	poskytne hodnotu SerNo vybraného modulu z tabulky
TUDFT2_Device_TypeNo	poskytne hodnotu TypeNo vybraného modulu z tabulky
Funkce související s otevřením/zavřením spojení s modulem	
funkce	popis
TUDFT2_Open	otevře spojení se zvoleným modulem
TUDFT2_Close	uzavře spojení se zvoleným modulem
Funkce pro obsluhu modulu (vyžadují otevření spojení s modulem)	
funkce	popis
TUDFT2_GetDeviceStatus	je určena pro zjištění aktuálního stavu interních obvodů modulu
TUDFT2_DeviceReset	je určena pro vyvolání interního restartu modulu
Funkce pro komunikaci servisním rozhraním (protokol UP4, vyžadují otevření spojení s modulem)	
funkce	popis
TUDFT2_SendCommand	vytvoří a vyšle UP4 paket ze sady instrukcí zadaných programem
TUDFT2_ReadCommandResponse	předá výsledky komunikace, tzn. reakce na vyslaný UP4 paket
Funkce pro přenosy datovým rozhraním (protokol UP1024/1040, vyžadují otevření spojení s modulem)	
funkce	popis
TUDFT2_GetDataBuffersStatus	zjištění stavu zásobníků ovladače
TUDFT2_GetDeviceBuffersStatus	zjištění stavu zásobníků modulu
TUDFT2_ReadDataBuffer	čtení dat ze zásobníku ovladače
TUDFT2_WriteDataBuffer	zápis dat do zásobníku ovladače
TUDFT2_StartStopDataBufferTransfer	spouštění/zastavení procesů obsluhujících zásobníky ovladače
TUDFT2_ClearDataBuffers	nulování zásobníků ovladače
Funkce pro callback funkcionalitu	
funkce	popis
TUDFT2_Callback	definice funkce umístěné v uživatelském programu volané vybranými událostmi ovladače

## A1.2 Návrátové kódy funkcí knihovny TEDIA\_TUDFT2.DLL verze 1.0.0

V tabulce níže je uveden přehled všech návratových kódů používaných funkcemi knihovny.

Přehled návratových kódů			
návratový kód (typu DWORD)		popis	
TUDFT2_RET_OK	0	O.K.	Funkce proběhla bez chyb.
TUDFT2_RET_InvalidHandle	1	ERROR	Neplatný handle předaný funkci.
TUDFT2_RET_ParameterError	2	ERROR	Chyba některého z parametrů funkce.
TUDFT2_RET_DeviceNotFound	101	ERROR	Zvolené zařízení nebylo nalezeno.
TUDFT2_RET_DeviceOpenFailure	102	ERROR	Při otevírání spojení s modulem nastala chyba.
TUDFT2_RET_FTD2xxError	103	ERROR	Ovladač narazil na chybu systémového ovladače.
TUDFT2_RET_CannotBePerformedYet	201	WARNING	Funkce nemůže být v daný okamžik provedena.

Návratový kód typu ERROR představuje fatální chybu, mj. chybu parametru nebo syntaxe. Návratový kód typu WARNING znamená chybu, která může být vyřešena pozdějším zopakováním volání (např. proces nedokončil činnost při předešlém spuštění).

*Poznámka: Narozdíl od karet je chyba při otevírání spojení, resp. chyba za běhu běžným provozním stavem, protože moduly lze odpojovat a připojovat za běhu.*

## A2.1 TUDFT2\_Init

```
TUDFT2_HANDLE TUDFT2_Init ( VOID );
```

### Popis

Funkce vytvoří v paměti instanci ovladače pro program a inicializuje ovladač.

### Podporované moduly

Funkce je implementována pro všechny typy modulů, resp. není na přítomnosti a vlastnostech modulů nijak závislá.

### Návratové kódy

Funkce nevyužívá žádný návratový kód, protože návratová hodnota představuje přímo *handle*.

Pokud funkce vrátí nulový *handle*, vytvoření instance ovladače se nepodařilo

Funkci TUDFT2\_Init lze volat opakovaně až do okamžiku, kdy se podaří vytvořit tolik instancí, kolik má být současně obsluhováno modulů.

## A2.2 TUDFT2\_Done

```
TUDFT2_RET TUDFT2_Done ( TUDFT2_HANDLE handle );
```

### *handle*

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### Popis

Funkce ukončí instanci ovladače a uvolní prostor v paměti.

Funkce by měla být volána jako poslední, tedy až po ukončení činnosti modulu a provedení funkce *TUDFT2\_Close*.

Pokud však nebyla funkce *TUDFT2\_Close* použita, funkce *TUDFT2\_Done* ji provede interně.

### Podporované moduly

Funkce je implementována pro všechny typy modulů, resp. není na přítomnosti a vlastnostech modulů nijak závislá.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.3 TUDFT2\_InterfaceVersion

```
DWORD TUDFT2_InterfaceVersion ( VOID );
```

### Popis

Funkce je určena pro zjištění verze ovladače (tzn. knihovny TEDIA\_TUDFT2.DLL); například v případě verze "1.02.14" je předána hodnota 0x00010214.

Funkce nevyžaduje předchozí otevření spojení s jakýmkoliv modulem.

### Podporované moduly

Funkce je implementována pro všechny typy modulů, resp. není na přítomnosti a vlastnostech modulů nijak závislá.

### Návratové kódy

Funkce nevyužívá žádný návratový kód, protože návratová hodnota představuje přímo předávanou proměnnou.

## A2.4 TUDFT2\_DriverVersion

```
TUDFT2_RET TUDFT2_DriverVersion  
            ( TUDFT2_HANDLE handle,  
              DWORD *version );
```

### *handle*

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *\*version*

Ukazatel na oblast paměti, do které je ovladačem uložena hodnota definující verzi systémového driveru FTD2xx.

---

### Popis

Funkce je určena pro zjištění verze systémového driveru FTD2xx; například v případě verze "3.01.15" je předána hodnota 0x00030115.

Funkce nevyžaduje předchozí otevření spojení s jakýmkoliv modulem.

### Podporované moduly

Funkce je implementována pro všechny typy modulů, resp. není na přítomnosti a vlastnostech modulů nijak závislá.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.5 TUDFT2\_DevicesInSystem

```
TUDFT2_RET TUDFT2_DevicesInSystem ( TUDFT2_HANDLE handle,  
                                     DWORD *count );
```

### *handle*

Hodnota *handle* (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *\*count*

Ukazatel na oblast paměti, do které je ovladačem uložena hodnota definující počet dostupných modulů.

---

### Popis

Funkce zjistí počet připojených modulů obsluhovaných ovladačem, načte jejich identifikační a stavové údaje do interní tabulky ovladače a programu poskytne informaci o počtu modulů..

Proměnná *count* nabývá hodnoty v rozsahu od nuly (není dostupný žádný modul) až do skutečného počtu modulů připojených k počítači (resp. enumerovaných v systému).

Funkce nevyžaduje předchozí otevření spojení s jakýmkoliv modulem.

Funkce využívá interní procesy systémového ovladače, doba trvání je typicky 200 ms.

### Podporované moduly

Funkce je implementována pro všechny typy modulů, resp. není na přítomnosti a vlastnostech modulů nijak závislá.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.6 TUDFT2\_Device\_VID

```
TUDFT2_RET TUDFT2_Device_VID  
  
    ( TUDFT2_HANDLE handle,  
      DWORD inx,  
      DWORD *VID );
```

### *handle*

Hodnota *handle* (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *inx*

Specifikuje zvolený modul z tabulky vytvořené funkcí *TUDFT2\_DevicesInSystem* nalezených modulů.

### *\*VID*

Ukazatel na oblast paměti, do které je ovladačem uložena hodnota definující USB VID zvoleného modulu.

---

### Popis

Funkce je určena pro zjištění USB VID modulu zvoleného parametrem *inx* (je platný v rozsahu 0 až *count* -1, kde *count* představuje počet modulů zjištěný funkcí *TUDFT2\_DevicesInSystem*).

Funkce nevyžaduje předchozí otevření spojení s jakýmkoliv modulem.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.7 TUDFT2\_Device\_PID

```
TUDFT2_RET TUDFT2_Device_PID( TUDFT2_HANDLE handle,  
                                DWORD inx,  
                                DWORD *PID );
```

### *handle*

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *inx*

Specifikuje zvolený modul z tabulky vytvořené funkcí *TUDFT2\_DevicesInSystem* nalezených modulů.

### *\*PID*

Ukazatel na oblast paměti, do které je ovladačem uložena hodnota definující USB PID zvoleného modulu.

---

### Popis

Funkce je určena pro zjištění USB PID modulu zvoleného parametrem *inx* (je platný v rozsahu 0 až *count*-1, kde *count* představuje počet modulů zjištěný funkcí *TUDFT2\_DevicesInSystem*).

Funkce nevyžaduje předchozí otevření spojení s jakýmkoliv modulem.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.8 TUDFT2\_Device\_SerNo

```
TUDFT2_RET TUDFT2_Device_SerNo  
            ( TUDFT2_HANDLE handle,  
              DWORD inx,  
              DWORD *SerNo );
```

### **handle**

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### **inx**

Specifikuje zvolený modul z tabulky vytvořené funkcí *TUDFT2\_DevicesInSystem* nalezených modulů.

### **\*SerNo**

Ukazatel na oblast paměti, do které je ovladačem uložena hodnota definující výrobní číslo zvoleného modulu.

---

### **Popis**

Funkce je určena pro zjištění SerNo (osmimístné výrobní číslo) modulu zvoleného parametrem *inx* (je platný v rozsahu 0 až *count* -1, kde *count* představuje počet modulů zjištěný funkcí *TUDFT2\_DevicesInSystem*).

Funkce nevyžaduje předchozí otevření spojení s jakýmkoliv modulem.

### **Podporované moduly**

Funkce je implementována pro všechny typy modulů.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.9 TUDFT2\_Device\_TypeNo

```
TUDFT2_RET TUDFT2_Device_TypeNo  
            ( TUDFT2_HANDLE handle,  
              DWORD inx,  
              DWORD *TypeNo );
```

### *handle*

Hodnota *handle* (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *inx*

Specifikuje zvolený modul z tabulky vytvořené funkcí *TUDFT2\_DevicesInSystem* nalezených modulů.

### \**TypeNo*

Ukazatel na oblast paměti, do které je ovladačem uložena hodnota definující typové číslo zvoleného modulu.

---

### Popis

Funkce je určena pro zjištění *TypeNo* (trojmístné číslo definované třemi nejvyššími číslicemi z osmimístného výrobního čísla *SerNo*) modulu zvoleného parametrem *inx* (je platný v rozsahu 0 až *count*-1, kde *count* představuje počet modulů zjištěný funkcí *TUDFT2\_DevicesInSystem*).

Funkce nevyžaduje předchozí otevření spojení s jakýmkoliv modulem.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.10 TUDFT2\_Open

```
TUDFT2_RET TUDFT2_Open( TUDFT2_HANDLE handle,  
                        DWORD inx,  
                        VOID *driverCallback );
```

### *handle*

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *inx*

Specifikuje zvolený modul z tabulky vytvořené funkcí *TUDFT2\_DevicesInSystem* nalezených modulů.

### *\*driverCallback*

Ukazatel na funkci vytvořenou v uživatelském programu volanou vybranými událostmi.

---

### Popis

Funkce je určena pro otevření spojení s modulem zvoleným pomocí parametru *inx* (je platný v rozsahu 0 až *count* -1, kde *count* představuje počet modulů zjištěný funkcí *TUDFT2\_DevicesInSystem*).

Parametr *driverCallback* je ukazatelem na funkci vytvořenou v uživatelském programu volanou vybranými událostmi v ovladači; pokud nemá být callback funkcionalita využita, je potřeba namísto ukazatele na funkci zadat null.

Funkce využívá v prvním kroku interní procesy systémového ovladače a následně komunikuje s modulem, doba trvání je typicky 500 ms.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.11 TUDFT2\_Close

```
TUDFT2_RET TUDFT2_Close ( TUDFT2_HANDLE handle );
```

### *handle*

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

---

### Popis

Funkce uzavře spojení s modulem.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

Funkce využívá interní procesy systémového ovladače, doba trvání je typicky 200 ms.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.12 TUDFT2\_GetDeviceStatus

**TUDFT2\_RET TUDFT2\_GetDeviceStatus** ( TUDFT2\_HANDLE *handle*,  
DWORD *\*deviceStatus* );

### *handle*

Hodnota *handle* (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *\*deviceStatus*

Ukazatel na oblast paměti, do které je ovladačem uložena hodnota se stavovými příznaky zvoleného modulu.

### Popis

Funkce je určena pro zjištění aktuálního stavu modulu, ke kterému má program již otevřené spojení.

Hodnota *deviceStatus* signalizuje různé stavové informace interních obvodů modulu (viz tabulka níže; hodnota nula znamená běžný provozní stav).

Funkce komunikuje s modulem, doba trvání jsou typicky jednotky až desítky milisekund.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

Popis příznaků <i>deviceStatus</i>		
bit parametru	význam hodnoty 0	význam hodnoty 1
D00	firmware FPGA modulu běží	běh firmware FPGA modulu je zastaven <i>příčina:</i> neočekávaný restart modulu (např. watchdog vlivem rušení nebo poklesu napájecího napětí USB), popř. interní závada <i>řešení:</i> program uzavře spojení s modulem a znovu jej otevře (tzn. funkce close/open)
D01	FPGA modulu je naprogramované (= funkční)	FPGA modulu není naprogramované <i>příčina:</i> přepsaný obsah Flash paměti, popř. interní závada <i>řešení:</i> konfiguračním programem obnovte firmware FPGA ze souboru
D02	kontrolní součet bloku kalibračních konstant je platný	kontrolní součet bloku kalibračních konstant je neplatný <i>příčina:</i> přepsaný obsah Flash paměti, popř. interní závada <i>řešení:</i> konfiguračním programem obnovte konstanty ze záložní paměti
D03	napájecí napětí bloků vyžadujících externí zdroj je v pořádku	napájecí napětí funkčních bloků vyžadujících externí zdroj není v pořádku <i>příčina:</i> vypnutý nebo přetížený zdroj, popř. interní závada <i>řešení:</i> zkontrolujte napájecí zdroj
<i>Poznámka: Ovladač nastavuje i více příznaků současně.</i>		

## A2.13 TUDFT2\_DeviceReset

```
TUDFT2_RET TUDFT2_DeviceReset ( TUDFT2_HANDLE handle );
```

### *handle*

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### Popis

Funkce je určena pro vyvolání interního restartu modulu, vyžaduje proto předchozí otevření spojení modulem.

Funkce komunikuje s modulem (doba trvání jsou typicky jednotky až desítky milisekund), předá požadavek na vyvolání interního restartu a se zpožděním nezbytné komunikace (avšak bez čekání na provedení restartu) předá programu návratový kód.

Restart běžného modulu trvá typicky 1 sekundu; po tuto dobu modul nereaguje na žádné povely z programu, zůstává však enumerován v systému. Program musí po vyvolání resetu uzavřít spojení s modulem pomocí *TUDFT2\_Close* a následně (nejdříve po jedné sekundě) spojení znovu otevřít.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.14 TUDFT2\_SendCommand

**TUDFT2\_RET TUDFT2\_SendCommand** ( TUDFT2\_HANDLE *handle*,  
DWORD *commandID*,  
TUDFT2\_CommandBuffer \**command*,  
DWORD *commandLen* );

### *handle*

Hodnota *handle* (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### *commandID*

Je pomocná identifikační hodnota použitá v odpovědi na povel (z pohledu komunikace nemá význam).

### \**command*

Ukazatel na oblast paměti, do které program uložil pole s instrukcemi povelu (pole délky 200 DWORD).

### *commandLen*

Hodnota definuje zaplnění pole *command* instrukcemi a daty.

## Popis

Funkce je určena pro vytvoření a vyslání UP4 povelu obsahujícího výhradně instrukce pro přístup do registrů FPGA a pomocné zpoždění vložené mezi dvě instrukce.

Pole *command* obsahuje až 200 DWORD parametrů definujících operaci čtení/zápis, adresu pro operaci čtení/zápis a data pro operaci zápis, resp. parametr definující zpoždění (viz popis parametrů pole v tabulce níže).

Ovladač verifikuje obsah pole v délce definované *commandLen*, konvertuje jej na UP4 povel, povel vyšle do přístroje a následně čeká na odpověď. Odpověď je programu zpřístupněna funkcí *TUDFT2\_ReadCommandResponse* obsluhovanou buď periodickým čtením, nebo callback funkcionalitou (viz popis funkce *TUDFT2\_Open*).

Funkce nekomunikuje přímo s modulem (komunikaci řeší ovladač v samostatném vlákně) a její vykonání proto nečeká na reakci modulu.

## Podporované moduly

Funkce je implementována pro všechny typy modulů.

## Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

Popis parametru pole <i>command</i> , resp. pole <i>response</i>			
D31..D24 (= povel)	D23..D16 (= rezerva)	D15..D08 (= adresa)	D07..D00 (= data)
0010 0000 (= čtení)	00 <sub>H</sub>	00 <sub>H</sub> ÷ FF <sub>H</sub>	00 <sub>H</sub> ÷ FF <sub>H</sub> (viz poznámka)
0100 0000 (= zápis)	00 <sub>H</sub>	00 <sub>H</sub> ÷ FF <sub>H</sub>	00 <sub>H</sub> ÷ FF <sub>H</sub>
1000 0000 (= zpoždění)	00 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub> ÷ 0F <sub>H</sub> (1÷16 ms)
ostatní hodnoty jsou rezervovány		hodnoty budou definovány v případě rozšíření sady povelů	

V případě operace čtení nemají pro pole *command* bity D07..D00 význam a jsou doplněny ovladačem po provedení operace čtení do pole *response*.

Operace zpoždění umožňuje vložit prodlevu mezi dvě operace čtení/zápis v případech, kdy firmware modulu takové zpoždění vyžaduje (např. čekání na naměření dat).

Ovladač má pevně nastavenou maximální dobu čekání na odpověď na 1 sekundu, UP4 povel tedy nesmí generovat mnohonásobný požadavek o vložené zpoždění.

## A2.15 TUDFT2\_ReadCommandResponse

```
TUDFT2_RET TUDFT2_ReadCommandResponse ( TUDFT2_HANDLE handle,  
                                           DWORD timeout,  
                                           DWORD *commandID,  
                                           TUDFT2_ResponseBuffer *response,  
                                           DWORD responseLen,  
                                           DWORD *responseStatus );
```

### **handle**

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### **timeout**

Je hodnota zadaná programem (čas v milisekundách 0+2000) definující mezní dobu, po kterou funkce čeká na odpověď z modulu.

### **\*commandID**

Je identifikační hodnota převzatá ovladačem z povelu vyslaného *TUDFT2\_SendCommand*.

### **\*response**

Ukazatel na oblast paměti, do které je ovladačem uloženo pole s reakcí modulu (pole délky 200 DWORD).

### **responseLen**

Hodnota definuje zaplnění pole *response* instrukcemi a daty.

### **\*responseStatus**

Ukazatel na oblast paměti, do které je ovladačem uložena stavová informace o průběhu komunikace.

## Popis

Funkce je určena pro načtení odpovědi modulu na UP4 povel vyslaný funkcí *TUDFT2\_SendCommand* buď periodickým čtením, nebo callback funkcionalitou (viz popis funkce *TUDFT2\_Open*).

Funkce obsahuje parametr *timeout* zadaný programem; funkce čeká na příchod odpovědi z modulu maximálně po dobu *timeout* a při návratu informuje o výsledku pomocí *responseStatus*.

Pole *response* obsahuje až 200 DWORD parametrů definujících operaci čtení/zápis, adresu pro operaci čtení/zápis a data pro operaci zápis, resp. parametr definující zpoždění (viz popis u funkce *TUDFT2\_SendCommand*).

Parametr *responseLen* předává v aktuální implementaci povelů shodnou hodnotu jako *commandLen* zadaný programem ve funkci *TUDFT2\_SendCommand*.

Parametr *responseStatus* předává stavové informace o úspěšnosti nebo důvodu neúspěšnosti komunikace, resp. stále probíhající komunikaci.

Funkce nekomunikuje přímo s modulem (komunikaci řeší ovladač v samostatném vlákne) a její vykonání proto nečeká na reakci modulu.

## Podporované moduly

Funkce je implementována pro všechny typy modulů.

## Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

Popis příznaků <i>responseStatus</i>	
hodnota / bit	význam
hodnota = 0	komunikace úspěšně dokončena, odpověď je dostupná v datovém poli
D00 = 1	komunikace ještě nebyla dokončena (tzn. zatím nepřišla odpověď, ale přijímací proces stále běží)
D01 = 1	komunikace dokončena neúspěšně - odpověď nepřišla v mezním čase nastaveném v ovladači
D02 = 1	komunikace dokončena neúspěšně - odpověď přišla s chybou
<i>Poznámka: Aktuální verze ovladače nenastavuje více příznaků současně.</i>	

## A2.16 TUDFT2\_GetDataBuffersStatus

```
TUDFT2_RET TUDFT2_GetDataBuffersStatus
( TUDFT2_HANDLE handle,
  DWORD *outBufferSize,
  DWORD *outBufferUsedSpace,
  DWORD *outBufferFreeSpace,
  DWORD *inBufferSize,
  DWORD *inBufferUsedSpace,
  DWORD *inBufferFreeSpace,
  DWORD *globalStatus );
```

### **handle**

Hodnota handle poskytnutá funkcí *TUDFT2\_Init*.

### **\*outBufferSize**

Ukazatel na oblast paměti, do které je ovladačem uložena celková velikost zásobníku v ovladači pro generovaná data.

### **\*outUsedSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost zaplněného prostoru v zásobníku v ovladači pro generovaná data.

### **\*outFreeSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost volného prostoru v zásobníku v ovladači pro generovaná data.

### **\*inBufferSize**

Ukazatel na oblast paměti, do které je ovladačem uložena celková velikost zásobníku v ovladači pro naměřená data.

### **\*inUsedSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost zaplněného prostoru v zásobníku v ovladači pro naměřená data.

### **\*inFreeSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost volného prostoru v zásobníku v ovladači pro naměřená data.

### **\*globalStatus**

Ukazatel na oblast paměti, do které je ovladačem uložena stavová informace o průběhu měření a generování.

## **Popis**

Funkce je určena pro zjištění velikosti a aktuálního zaplnění dvou datových zásobníků vytvořených uvnitř ovladače (ne tedy uvnitř USB modulu).

Platí  $outUsedSize + outFreeSize = outBufferSize$ . Velikost zásobníku je u verze 1.0.0 nastavena pevně na 20 MB.

Platí  $inUsedSize + inFreeSize = inBufferSize$ . Velikost zásobníku je u verze 1.0.0 nastavena pevně na 20 MB.

## **Podporované moduly**

Funkce je implementována pro všechny typy modulů.

## **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

Popis příznaků <i>globalStatus</i>	
bit parametru	význam
D00 = 1	signalizuje fatální chybu v paketu protokolu UP1024/1040, ovladač již další naměřená data nepřenáší do zásobníku ovladače (již přenesená zůstávají dostupná programu)
D01 = 1	signalizuje chybu přetečení zásobníku naměřených dat v ovladači; od okamžiku přetečení ovladač žádná další data z modulu nezpracovává, již uložená data v zásobníku jsou programu dostupná
D24 = 1	signalizuje fatální chybu měření v modulu, modul ukončil přenos dat do PC
D25 = 1	signalizuje chybu podtečení zásobníku v modulu (po příchodu dalších dat modul pokračuje v generování), nebo fatální chybu generování v modulu (modul dalších data z PC už nepřijímá)
D26 = 1	signalizuje fatální chybu modulu (např. výpadek napájecího napětí I/O signálů)

## A2.17 TUDFT2\_GetDeviceBuffersStatus

```
TUDFT2_RET TUDFT2_GetDeviceBuffersStatus ( TUDFT2_HANDLE handle,
DWORD *outBufferSize,
DWORD *outBufferUsedSpace,
DWORD *outBufferFreeSpace,
DWORD *inBufferSize,
DWORD *inBufferUsedSpace,
DWORD *inBufferFreeSpace,
DWORD *globalStatus );
```

### **handle**

Hodnota handle poskytnutá funkcí *TUDFT2\_Init*.

### **\*outBufferSize**

Ukazatel na oblast paměti, do které je ovladačem uložena celková velikost zásobníku v modulu pro generovaná data.

### **\*outUsedSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost zaplněného prostoru v zásobníku v modulu pro generovaná data.

### **\*outFreeSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost volného prostoru v zásobníku v modulu pro generovaná data.

### **\*inBufferSize**

Ukazatel na oblast paměti, do které je ovladačem uložena celková velikost zásobníku v modulu pro naměřená data.

### **\*inUsedSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost zaplněného prostoru v zásobníku v modulu pro naměřená data.

### **\*inFreeSize**

Ukazatel na oblast paměti, do které je ovladačem uložena velikost volného prostoru v zásobníku v modulu pro naměřená data.

### **\*globalStatus**

Ukazatel na oblast paměti, do které je ovladačem uložena stavová informace o průběhu měření a generování.  
(viz popis funkce *TUDFT2\_GetDataBuffersStatus*)

## Popis

Funkce je analogická předešlé *TUDFT2\_GetDataBuffersStatus* s tím rozdílem, že poskytuje informace o zásobnících modulu.

Ačkoliv jsou předávány informace o zaplnění zásobníků v modulu, funkce nekomunikuje s modulem UP4 protokolem. Informace o zaplnění jsou odvozeny od interních proměnných ovladače, jejichž hodnota je převzata z posledního korektně přijatého paketu UP1024/1040 protokolu přicházejících do počítače. Je na odpovědnosti programu, aby byl modul nastaven do režimu, ve kterém jsou pakety vysílány, v opačném případě budou předávány neaktualizované konstantní hodnoty.

Hodnoty jsou navíc nulovány funkcí *TUDFT2\_ClearDataBuffers*.

## Podporované moduly

Funkce je implementována pro všechny typy modulů.

## Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.18 TUDFT2\_ReadDataBuffer

```
TUDFT2_RET TUDFT2_ReadDataBuffer ( TUDFT2_HANDLE handle,  
                                     BYTE *dataBuffer,  
                                     DWORD dataRequested,  
                                     DWORD *dataConfirmed );
```

### **handle**

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### **\*dataBuffer**

Ukazatel na oblast paměti, do které jsou ovladačem přenesena naměřená data ze zásobníku *inBuffer*.

### **dataRequested**

Množství dat, které chce program ze zásobníku ovladače přečíst.

### **\*dataConfirmed**

Ukazatel na oblast paměti, do které je ovladačem zapsán počet skutečně předaných dat.

---

### Popis

Funkce je určena pro načtení naměřených dat z datového zásobníku ovladače.

Program zadává požadavek o množství dat a ovladač jej splní (tzn. přenese z interního zásobníku ovladače do *\*dataBuffer* programu) v případě, že je zásobník dostatečně zaplněný (v takovém případě jsou *dataRequested* a *dataConfirmed* shodné. Je-li v zásobníku méně dat, než požaduje program, ovladač přenese jen dostupný počet dat a program informuje nižší hodnotou *dataConfirmed*.

Doba trvání funkce jsou typicky milisekundy pro předání bloku dat velikosti 1 MB, resp. desítky milisekund pro předání bloku dat maximální velikosti 20 MB.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.19 TUDFT2\_WriteDataBuffer

```
TUDFT2_RET TUDFT2_WriteDataBuffer ( TUDFT2_HANDLE handle,
                                     BYTE *dataBuffer,
                                     DWORD dataRequested,
                                     DWORD *dataConfirmed );
```

### **handle**

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### **\*dataBuffer**

Ukazatel na oblast paměti, ze které jsou ovladačem přenesena generovaná data do zásobníku *outBuffer*.

### **dataRequested**

Množství dat, které chce program do zásobníku ovladače zapsat.

### **\*dataConfirmed**

Ukazatel na oblast paměti, do které je ovladačem zapsán počet skutečně převzatých dat.

---

### Popis

Funkce je určena pro zapsání generovaných dat do datového zásobníku ovladače.

Program zadává požadavek o množství dat a ovladač jej splní (tzn. přenese z *\*dataBuffer* programu do interního zásobníku ovladače) v případě, že je v zásobníku dostatečně velký volný prostor (v takovém případě jsou *dataRequested* a *dataConfirmed* shodné). Je-li v zásobníku méně volného prostoru, než požaduje program, ovladač přenese jen omezený počet dat a program informuje nižší hodnotou *dataConfirmed*.

Doba trvání funkce jsou typicky milisekundy pro předání bloku dat velikosti 1 MB, resp. desítky milisekund pro předání bloku dat maximální velikosti 20 MB.

### Podporované moduly

Funkce je implementována pro všechny typy modulů.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.20 TUDFT2\_StartStopDataBufferTransfer

```
TUDFT2_RET TUDFT2_StartStopDataBufferTransfer ( TUDFT2_HANDLE handle,  
                                                DWORD runOut,  
                                                DWORD runIn );
```

### **handle**

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### **runOut**

Parametr definující běh interního vlákna generování v ovladači.

(D00=0 definuje, že obsluha je zastavena, D00=1 definuje, že obsluha probíhá; ostatní bity jsou rezervovány)

### **runIn**

Parametr definující běh interního vlákna měření v ovladači.

(D00=0 definuje, že obsluha je zastavena, D00=1 definuje, že obsluha probíhá; ostatní bity jsou rezervovány)

---

### **Popis**

Funkce je určena pro spouštění a zastavování interních procesů obsluhujících oba datové zásobníky ovladače a datové přenosy protokolem UP1024/1040 oběma směry.

Funkce nenuluje zásobníky, je vždy potřeba použít funkci *TUDFT2\_ClearDataBuffers*.

### **Podporované moduly**

Funkce je implementována pro všechny typy modulů.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.21 TUDFT2\_ClearDataBuffers

```
TUDFT2_RET TUDFT2_ClearDataBuffers ( TUDFT2_HANDLE handle,  
                                       DWORD clearOut,  
                                       DWORD clearIn );
```

### **handle**

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### **clearOut**

Parametr definující nulování zásobníku generování v ovladači (resp. nulování počtu uložených byte).  
(D00=0 neprovádí žádnou funkci, D00=1 provede nulování zásobníku; ostatní bity jsou rezervovány)

### **clearIn**

Parametr definující nulování zásobníku měření v ovladači (resp. nulování počtu uložených byte).  
(D00=0 neprovádí žádnou funkci, D00=1 provede nulování zásobníku; ostatní bity jsou rezervovány)

---

### **Popis**

Funkce je určena pro nulování zásobníků generování a měření uvnitř ovladače.

Po nulování (tzn. zápisu hodnoty 1) není potřeba zapisovat hodnotu 0.

Funkce komunikuje s řadičem modulu interními procesy systémového ovladače, doba trvání je typicky 200 ms.

### **Podporované moduly**

Funkce je implementována pro všechny typy modulů.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule),  
v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).

## A2.22 TUDFT2\_Callback

```
VOID *TUDFT2_Callback ( TUDFT2_HANDLE handle,  
                        DWORD typeCB,  
                        DWORD commandID );
```

### **handle**

Hodnota handle (typu ukazatel) poskytnutá funkcí *TUDFT2\_Init*.

### **typeCB**

Parametr definující událost, která vyvolala callback funkci.

0 = callback od procesu *TUDFT2\_SendCommand*

ostatní hodnoty jsou rezervovány

### **commandID**

Je pomocná identifikační hodnota použitá v povelu UP4 (viz popis funkce *TUDFT2\_SendCommand*).

---

### **Popis**

Typ funkce definované uživatelem pro obsluhu callback funkcionality, tělo se musí nacházet v uživatelském programu (tzn. není součástí ovladače).

Tato funkce bude volána pouze v případě povolení callback funkcionality (viz popis funkce *TUDFT2\_Open*).

### **Podporované moduly**

Funkce je implementována pro všechny typy modulů.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *TUDFT2\_RET\_OK* (DWORD hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů (viz kapitola A1.2).