

**Systemový ovladač  
PCI/PCIe karet  
pro Windows**

**Programátorská příručka**

Historie dokumentu		
datum	verze	změny
4.8.2015	---	výchozí verze
24.8.2015	08.2015	první finální verze (ovladač verze 3.10)
25.2.2016	02.2016	rozšíření podpory PCIe karet (ovladač verze 3.11)
29.2.2016	02.2016/B	drobná zpřesnění, uvolnění ovladače verze 3.12
30.3.2021	03.2021	Nové vlastnosti knihovny tedia_ep4gxa.dll... - nové funkce pro přístup k registrům 64bitovými daty - nové funkce pro čtení výrobního čísla and CardID hodnoty (knihovny tedia_ox9162.dll a tedia_ox952.dll beze změn)

### 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

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

- 2.1 Úvod
- 2.2 Terminologie
- 2.3 Karty TEDIA

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

- 3.1 Úvod
- 3.2 Součásti ovladače
- 3.3 Přehled funkcí knihoven

## **4. Popis využití ovladače**

- 4.1 Úvod
- 4.2 Nalezení karty a otevření spojení s kartou
- 4.3 Přístup k registrům karty
- 4.4 Využití přerušení systému
- 4.5 Uzavření spojení s kartou

## **Příloha A, knihovna tedia\_ox9162.dll**

- A1.1 Veřejné funkce knihovny tedia\_ox9162.dll verze 3.00 a vyšší
- A1.2 Návrátové kódy funkcí knihovny tedia\_ox9162.dll verze 3.00 a vyšší
- A2.1 OX9162\_InterfaceVersion
- A2.2 OX9162\_DriverVersion
- A2.3 OX9162\_CardsInSystem
- A2.4 OX9162\_Card\_DID
- A2.5 OX9162\_Card\_BUS
- A2.6 OX9162\_Card\_SLOT
- A2.7 OX9162\_Open
- A2.8 OX9162\_OpenMulti
- A2.9 OX9162\_Card\_OpenCount
- A2.10 OX9162\_Close
- A2.11 OX9162\_BAR4\_BaseAdr
- A2.12 OX9162\_MemRead
- A2.13 OX9162\_MemWrite
- A2.14 OX9162\_IRQ\_Wait
- A2.15 OX9162\_IRQ\_SetTimeout
- A2.16 OX9162\_IRQ\_Count

## **Příloha B, knihovna tedia\_ox952.dll**

- B1.1 Veřejné funkce knihovny tedia\_ox952.dll verze 3.00 a vyšší
- B1.2 Návrátové kódy funkcí knihovny tedia\_ox952.dll verze 3.00 a vyšší
- B2.1 OX952\_InterfaceVersion
- B2.2 OX952\_DriverVersion
- B2.3 OX952\_CardsInSystem
- B2.4 OX952\_Card\_DID
- B2.5 OX952\_Card\_BUS
- B2.6 OX952\_Card\_SLOT
- B2.7 OX952\_Open
- B2.8 OX952\_OpenMulti
- B2.9 OX952\_Card\_OpenCount
- B2.10 OX952\_Close
- B2.11 OX952\_BAR05\_BaseAdr, OX952\_BAR11\_BaseAdr

- B2.12 OX952\_MemRead
- B2.13 OX952\_MemWrite
- B2.14 OX952\_IRQ\_Wait
- B2.15 OX952\_IRQ\_SetTimeout
- B2.16 OX952\_IRQ\_Count

## **Příloha C, knihovna tedia\_ep4gxa.dll**

- C1.1 Veřejné funkce knihovny tedia\_ep4gxa.dll verze 3.10 a vyšší
- C1.2 Návrátové kódy funkcí knihovny tedia\_ep4gxa.dll verze 3.10 a vyšší
- C2.1 EP4GXA\_InterfaceVersion
- C2.2 EP4GXA\_DriverVersion
- C2.3 EP4GXA\_CardsInSystem
- C2.4 EP4GXA\_Card\_DID
- C2.5 EP4GXA\_Card\_BUS
- C2.6 EP4GXA\_Card\_SLOT
- C2.7 EP4GXA\_Card\_ID (podporováno od verze interface 3.18)
- C2.8 EP4GXA\_Card\_SerNr (podporováno od verze interface 3.18)
- C2.9 EP4GXA\_Open
- C2.10 EP4GXA\_OpenMulti
- C2.11 EP4GXA\_Card\_OpenCount
- C2.12 EP4GXA\_Close
- C2.13 EP4GXA\_BAR0\_BaseAdr
- C2.14 EP4GXA\_ReadReg8, EP4GXA\_ReadReg32
- C2.15 EP4GXA\_WriteReg8, EP4GXA\_WriteReg32
- C2.16 EP4GXA\_ReadReg64, EP4GXA\_ReadReg2x32 (podporováno od verze interface 3.18)
- C2.17 EP4GXA\_WriteReg64, EP4GXA\_WriteReg2x32 (podporováno od verze interface 3.18)
- C2.18 EP4GXA\_IRQ\_Wait
- C2.19 EP4GXA\_IRQ\_SetTimeout
- C2.20 EP4GXA\_IRQ\_Count

## **Příloha D, seznam karet a historie verzí**

- D1.1 Seznam standardních PCI karet obsluhovaných knihovnou tedia\_ox9162.dll
- D2.1 Seznam standardních PCI karet obsluhovaných knihovnou tedia\_ox952.dll
- D3.1 Seznam standardních PCI Express karet obsluhovaných knihovnou tedia\_ep4gxa.dll
- D4.1 Verze ovladače a podporované operační systémy Windows

# 1. Základní informace

---

## 1.1 Úvod

Tato programátorská příručka je věnována popisu systémových ovladačů dodávaných ke kartám TEDIA pro případ, kdy uživatel potřebuje vytvořit vlastní program přímým ovládáním registrů karty. Není určena běžnému uživateli karty, který potřebuje systémový ovladač jen nainstalovat a využívat s již hotovými programy, případně programovat aplikaci využívající některý z aplikačních ovladačů (např. TEDIA\_DAQ01, ovladače pro vývojový systém Control Web, ...).

Programátorská příručka obsahuje ...

- popis systémového ovladače, struktury souborů apod.;
- popis API knihoven interface;
- postup, jak s využitím systémového ovladače vytvořit vlastní program, případně aplikační ovladač.

Co tato příručka neobsahuje ...

- popis instalace ovladače (je uveden v samostatné příručce věnované pouze instalaci);
- popis aplikačních ovladačů (např. TEDIA\_DAQ01) vystavených nad systémovým ovladačem.

Pro využití ovladače je potřebná znalost registrové struktury zvolené karty popsané zpravidla v programátorské příručce zvolené karty. Popis registrové struktury je k dispozici pro všechny typy PCI a PCIe karet TEDIA.

## 1.2 Kde získat další informace, technická podpora

Další užitečné informace lze získat na adrese ...

URL: <https://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

telefon: <https://www.tedia.cz/kontakty>

tech. podpora: <https://www.tedia.cz/podpora>

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.*

## 2. Obecné informace a terminologie

### 2.1 Úvod

Dále uvedené odstavce jsou věnovány obecnému popisu PCI/PCle karet 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 karty v jádře Windows a jako rozhraní pro běžné programy (resp. aplikační ovladače) využívá běžnou DLL knihovnu
aplikační ovladač	je software, který využívá funkce systémového ovladače (směrem ke kartě) a v opačném směru poskytuje specifické funkce a rozhraní pro různé vývojové prostředky a systémy; žádná jeho část neběží v jádře Windows
PCI-SIG	PCI Special Interest Group; organizace standardizující PCI a PCI Express sběrnici
PCI VID	resp. PCI Vendor ID; je 16bitové identifikační číslo přidělené PCI-SIG členům této organizace; TEDIA má přiděleno číslo 1760 <sub>H</sub>
PCI DID	resp. PCI Device ID, je 16bitové číslo přidělované výrobcem (resp. držitelem PCI VID) každému typu karty
... VID + DID	sada těchto dvou hodnot tedy 100% identifikuje typ karty
PCI BUS	je číslo PCI sběrnice v počítači
PCI SLOT	je číslo pozice pro zásuvnou PCI/PCle kartu v počítači
... BUS + SLOT	sada těchto dvou hodnot tedy 100% definuje umístění karty v počítači a umožňuje identifikovat kartu zejména v případě, kdy jsou v počítači dvě nebo více karet stejného typu
PCI function	karta může obsahovat až osm samostatných funkčních částí, které systém rozpozná a obsluhuje zcela nezávisle (zjednodušeně řečeno - karta se dvěma PCI function se chová jako by v jednom slotu byly instalovány dvě samostatné karty)
BAR	Base Address Register; PCI/PCle karty obsahují funkční registry, které jsou mapovány do I/O nebo MEM prostoru počítače (operační systém po startu zjistí požadavky karty a do BAR registrů запиše počáteční adresy, pod kterými jsou mapovány registrové bloky); karta (resp. každá PCI function) může obsahovat více registrových bloků a každý pak vystupuje vlastním BAR registrem, resp. přidělenou adresou
... BAR karet TEDIA	karty používají více registrových bloků, pro uživatelské registry je však vyhrazen vždy jeden blok (BAR); číslo BARu je závislé na typu karty, resp. typu PCI řadiče použitého pro skupinu karet
I/O nebo MEM	registry karty mohou být mapovány v I/O prostoru (zastaralé, pomalé a výhodné jen pro operační systémy neumožňující 32bitové adresování paměti) a v MEM prostoru; systémový ovladač využívá registry mapované v MEM prostoru

### 2.3 Karty TEDIA

Z hlediska použitého řadiče PCI, resp. PCle sběrnice lze karty dělit do tří skupin.

PCI karty 1. generace	<ul style="list-style-type: none"> <li>osazeny řadičem OX9162 (Oxford Semiconductor, resp. PLX Technology)</li> <li>kompatibilní s PCI sběrnici s úrovněmi 5 V</li> </ul>
PCI karty 2. generace	<ul style="list-style-type: none"> <li>osazeny řadičem OXmPCI952/OXuPCI952 (Oxford Semiconductor, resp. PLX Technology)</li> <li>kompatibilní s PCI sběrnici s úrovněmi 5 V i 3,3 V</li> </ul>
PCI Express karty	<ul style="list-style-type: none"> <li>řadič řešen hradlovým polem Altera řady EP4CGX</li> <li>kompatibilní se sběrnici PCI Express (x1, Gen 1)</li> </ul>

Z hlediska systémového ovladače jsou všechny typy obsluhovány společnou systémovou částí (běží v jádře Windows) a třemi samostatnými DLL knihovnami vytvářejícími interface. Ovladač funkčně neodlišuje jednotlivé typy karet v rámci jedné skupiny.

## 3. Popis ovladače a programátorského rozhraní

### 3.1 Úvod

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

### 3.2 Součásti ovladače

Systémový ovladač je tvořen následujícími klíčovými soubory...

tediaOxPCI.sys	část běžící v jádře Windows
tedia_ox9162.dll	interface pro PCI karty 1. generace používající řadič OX9162
tedia_ox952.dll	interface pro PCI karty 2. generace používající řadič OXmPCI952 nebo OXuPCI952
tedia_ep4gxa.dll	interface pro PCI Express karty používající hradlové pole Altera řady EP4GX

Všechny uvedené soubory jsou dostupné v 32bitové i 64bitové verzi a umožňují běh 32bitovým programům instalovaným v 32bitových i 64bitových verzích Windows, resp. 64bitovým programům instalovaným v 64bitových verzích Windows.

Instalační balík obsahuje řadu dalších souborů, jejich popis však přesahuje rámec tohoto dokumentu a jejich význam není pro uživatele podstatný.

*Poznámka: Popisu API rozhraní tří DLL knihoven jsou věnovány přílohy tohoto dokumentu, rozhraní tediaOxPCI.sys není veřejně dokumentováno.*

### 3.3 Přehled funkcí knihoven

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

Jak bude z tabulek patrné, všechny knihovny obsahují (až na vyznačené výjimky) identické funkce odlišující se pouze prefixem (analogické funkce mají v závislosti na knihovně název **OX9162\_\***, **OX952\_\*** nebo **EP4GXA\_\***).

Obecné funkce bez návaznosti na kartu (použitelné i před otevřením spojení s kartou)	
funkce	popis
OX9162_InterfaceVersion OX952_InterfaceVersion EP4GXA_InterfaceVersion	funkce určené pro zjištění verze interface ovladače
OX9162_DriverVersion OX952_DriverVersion EP4GXA_DriverVersion	funkce určené pro zjištění verze systémové části ovladače
OX9162_CardsInSystem OX952_CardsInSystem EP4GXA_CardsInSystem	funkce určené pro zjištění počtu karet obsluhovaných ovladačem

Jak je patrné z tabulky výše, API obsahuje funkce umožňující zjistit verzi interface (tzn. DLL knihovny) i systémové části ovladače (tzn. tediaOxPCI.sys).

Třetí funkcí pak umožňuje zjistit počet karet obsluhovaných každou z knihoven. Obsluhované karty jsou řazeny do tří tabulek (jedna pro každou z DLL) a funkce vrací pro každou knihovnu počet od nuly (není dostupná žádná karta) výše.

Funkce přistupující ke kartě využívají jako identifikátor karty její umístění v tabulce; tento index nabývá hodnoty od nuly (první karta v tabulce) výše.

Funkce pro zjištění vlastností karty (použitelné i před otevřením spojení s kartou)	
funkce	popis
OX9162_Card_DID OX952_Card_DID EP4GXA_Card_DID	funkce určené pro zjištění PCI Device ID zvolené karty
OX9162_Card_BUS OX952_Card_BUS EP4GXA_Card_BUS	funkce určené pro zjištění čísla PCI sběrnice, ve které je zvolená karta instalována
OX9162_Card_SLOT OX952_Card_SLOT EP4GXA_Card_SLOT	funkce určené pro zjištění čísla slotu, ve kterém je zvolená karta instalována
EP4GXA_Card_ID	funkce určená pro zjištění hodnoty CardID zvolené karty (hodnota uživatelsky nastavená DIP spínačem)
EP4GXA_Card_SerNr	funkce určená pro zjištění hodnoty výrobního čísla zvolené karty

Druhá skupina funkcí je určena ke zjištění vlastností karty, resp. umístění v počítači.

Všechny tři funkce používají jako identifikátor karty její index v tabulce (viz popis funkcí **\*\*\*\_CardsInSystem** výše) a poskytují informaci o hodnotě...

- PCI DID            v kombinaci s PCI VID TEDIA identifikují typ karty (ovladač nepodporuje karty s odlišným PCI VID)
- BUS + SLOT      dvojice konstant umožňující identifikovat kartu podle umístění v počítači (potřebné zejména v případě, kdy jsou v počítači dvě nebo více karet stejného typu)

Funkce související s otevřením/zavřením spojení s kartou	
funkce	popis
OX9162_Open OX952_Open EP4GXA_Open	funkce určené pro otevření <u>výhradního</u> spojení se zvolenou kartou
OX9162_OpenMulti OX952_OpenMulti EP4GXA_OpenMulti	funkce určené pro otevření <u>nevýhradního</u> spojení se zvolenou kartou
OX9162_Card_OpenCount OX952_Card_OpenCount EP4GXA_Card_OpenCount	funkce určené pro zjištění aktuálního počtu otevřených spojení s kartou
OX9162_Close OX952_Close EP4GXA_Close	funkce určené pro uzavření spojení s kartou

Třetí skupina funkcí je určena pro otevření spojení a uzavření spojení s kartou, tabulka obsahuje i související pomocnou funkci umožňující zjistit aktuální počet otevřených spojení s kartou.

Knihovny obsahují dvě funkce pro otevření spojení; základní funkce **\*\*\*\_Open** otevírají výhradní spojení s kartou, tzn. blokují současné otevření jiným programem.

Alternativní funkce **\*\*\*\_OpenMulti** umožňují otevřít vícenásobné (nevýhradní) spojení s kartou a přístup ke kartě více programům současně (všechny programy musejí použít **\*\*\*\_OpenMulti**). Pomocné funkce **\*\*\*\_Card\_OpenCount** pak umožňují zjistit počet programů, které kartu aktuálně využívají.



Funkce určené pro obsluhu registrů karty (vyžadují otevření spojení s kartou)	
funkce	popis
OX9162_BAR4_BaseAdr (OX952_BAR05_BaseAdr) OX952_BAR11_BaseAdr EP4GXA_BAR0_BaseAdr	funkce určené pro zjištění počáteční adresy prostoru paměti, ve kterém jsou mapovány funkční registry karty (OX952_BAR05_BaseAdr viz vysvětlující poznámka v textu pod tabulkou)
OX9162_MemRead OX952_MemRead EP4GXA_ReadReg8 EP4GXA_ReadReg32 EP4GXA_ReadReg64 *** EP4GXA_ReadReg2x32 ***	rozhraní OX9162 a OX952 nabízí funkce umožňující čtení nebo zápis osmibitového registru (rozsah adres není nijak omezen); rozhraní EP4GXA nabízí funkce s odlišnými parametry umožňující alternativně čtení nebo zápis jednoho osmibitového nebo 32bitového registru, resp. dvou 32bitových registrů naráz (oproti OX9162 a OX952 je platná adresa omezena jen na prostor registrů karty);
OX9162_MemWrite OX952_MemWrite EP4GXA_WriteReg8 EP4GXA_WriteReg32 EP4GXA_WriteReg64 *** EP4GXA_WriteReg2x32 ***	všechny uvedené funkce jsou nezbytné pro vývojová prostředí nepodporující přímý přístup k paměti, v ostatních lze využít buď uvedené funkce nebo standardní metodu přístupu k paměti *** nová funkce dostupná od verze 3.18 výše

Čtvrtá skupina funkcí je určena pro přístup k registrům karty.

První funkce **\*\*\*\_BAR\*\_BaseAdr** umožňují zjistit počáteční adresu bloku registrů karty; adresa je v závislosti na programu, resp. verzi připojené DLL knihovny 32bitová nebo 64bitová. Mapování registrů konkrétní karty je popsáno v příručce karty jako ofset od adresy předané funkcemi **\*\*\*\_BAR\*\_BaseAdr**.

*Poznámka: Ovladač DAQ karet s řadičem OXmPCI952/OXuPCI952 zpřístupňuje kromě BAR1 PCI funkce 1 (označen jako BAR11) mapujícím funkční registry navíc i BAR5 PCI funkce 0 (označen jako BAR05) obsahující sériový port určený pro servisní účely. Základní funkcí servisního rozhraní je možnost update firmware karty programy dodávanými s kartou, umožňuje však také zjistit unikátní výrobní číslo karty.*

Funkce určené pro zpracování přerušení (vyžadují otevření spojení s kartou)	
funkce	popis
OX9162_IRQ_Wait OX952_IRQ_Wait EP4GXA_IRQ_Wait	funkce pozastaví thread až do okamžiku detekce přerušení systému nebo vypršení timeoutu
OX9162_IRQ_SetTimeout OX952_IRQ_SetTimeout EP4GXA_IRQ_SetTimeout	funkce nastavuje funkci <b>***_IRQ_Wait</b> mezní dobu čekání na přerušení (timeout)
OX9162_IRQ_Count OX952_IRQ_Count EP4GXA_IRQ_Count	funkce předává počet přerušení vyvolaných kartou

Poslední skupina funkcí vytváří programátorsky jednoduchý mechanismus zachycení přerušení systému kartou.

Základní funkcí je **\*\*\*\_IRQ\_Wait**, která pozastaví vlákno až do okamžiku detekce přerušení systému nebo vypršení timeoutu. Program tedy v samostatném vlákne periodicky volá tuto funkci a v závislosti na návratovém kódu spouští obsluhu přerušení.

Pomocná funkce **\*\*\*\_IRQ\_SetTimeout** umožňuje nastavit dobu, za kterou funkce **\*\*\*\_IRQ\_Wait** propustí vlákno v případě, kdy není detekováno přerušení systému.

Poslední funkce servisního charakteru slouží ke zjištění počtu přerušení vyvolaných kartou.

## 4. Popis využití ovladače

### 4.1 Úvod

Dále uvedené odstavce jsou věnovány popisu využití ovladače uživatelskými programy.

### 4.2 Nalezení karty a otevření spojení s kartou

Je-li nezbytné zajistit současný přístup z více programů (resp. procesů) lze pro otevření spojení s kartou použít funkci **\*\*\*\_OpenMulti**. Je však potřeba si uvědomit, že přístupy všech procesů jsou symetrické a v případě ovládání stejných funkcí mohou vyvolat vzájemné kolize. Ve všech ostatních případech je vhodnější pro otevření spojení s kartou použít funkci **\*\*\*\_Open** blokující současné otevření jinému programu.

Vlastnímu otevření spojení s kartou by však vždy mělo předcházet zjištění, jaké karty a na jakých indexech tabulky jsou dostupné. Každá knihovna **tedia\_ox9162.dll**, **tedia\_ox952.dll** a **tedia\_ep4gxa.dll** vytváří vlastní tabulku.

V prvním kroku je potřeba funkcí **\*\*\*\_CardsInSystem** zjistit, kolik karet je obsluhováno zvolenou knihovnou. Je-li známý typ karty, se kterým má program pracovat, postačuje použít odpovídající z knihoven **tedia\_ox9162.dll**, **tedia\_ox952.dll** nebo **tedia\_ep4gxa.dll**, v případě různých diagnostických programů lze postupně zjistit stav pro všechny knihovny.

V druhém kroku je potřeba funkcí **\*\*\*\_Card\_DID** zjistit pro všechny dostupné karty zvolené knihovny (tzn. od 0 do hodnoty předané funkcí **\*\*\*\_CardsInSystem** zmenšenou o 1) jejich PCI Device ID a vybrat kartu/karty, se kterými bude program pracovat. Seznam podporovaných karet a jejich PCI Device ID jsou uvedeny v Příloze D této příručky. V případě více shodných karet v počítači lze pro jejich odlišení využít údaje poskytnuté funkcemi **\*\*\*\_Card\_BUS** a **\*\*\*\_Card\_SLOT**.

Až v třetím kroku lze otevřít spojení s kartou jednou z funkcí **\*\*\*\_Open** nebo **\*\*\*\_OpenMulti**.

### 4.3 Přístup k registrům karty

Podarí-li se úspěšně otevřít spojení s kartou, je potřeba zjistit počáteční adresu bloku registrů pomocí funkce **\*\*\*\_BAR\*\_BaseAdr**. Adresa je v závislosti na programu, resp. verzi připojené DLL knihovny 32bitová nebo 64bitová.

Pro vlastní přístup k registrům lze použít standardní způsob přístupu k paměti, případně přistupovat pomocí funkcí **\*\*\*\_MemRead** a **\*\*\*\_MemWrite** (resp. **EP4GXA\_ReadReg\*** a **EP4GXA\_WriteReg\***) určených primárně pro vývojová prostředí nepodporující přímý přístup k paměti (např. Visual Basic). Naopak pro funkce umožňující přenos 64bitových dat jednou instrukcí se v případě 32bitové překládaných programů jen těžko hledá náhrada.

### 4.4 Využití přerušování systému

Obsluhu přerušování lze rozdělit do tří částí...

- první část představuje vlastní zachytávání přerušování a přenesení této informace do uživatelského programu; tento krok zahrnuje i zakázání přerušování při ukončování programu
- druhou částí je vlastní zachytávání události a spouštění společného obslužného kódu
- třetí částí je obsluha registrů karty určených pro identifikaci zdroje přerušování a specifická obsluha v závislosti na zdroji přerušování.

Zatímco druhá část je unifikována pro všechny typy karet v rámci skupiny obsluhované jednou knihovnou (řešeno funkcemi **\*\*\*\_IRQ\_Wait**, **\*\*\*\_IRQ\_SetTimeout** a **\*\*\*\_IRQ\_Count**), první a třetí část je specifická pro konkrétní typ karty a typ události vyvolávající přerušování.

**Úvodem pár slov k registrům karet TEDIA souvisejících s obvody pro vyvolání přerušování systému.**

Většina karet má sadu tří registrů sloužících k povolení zdroje přerušování (IRQCfgReg), identifikaci události vyvolávající přerušování (IRQStatusReg) a nulování nastaveného příznaku události (IRQClrReg). Kromě těchto registrů mají karty ještě čtvrtý registr globálně povolující vyvolání přerušování systému (INTEnReg). Podrobný popis lze nalézt v programátorských příručkách karet.

Obecně lze definovat následující posloupnost operací (výchozí stav je zakázání všech zdrojů přerušování a vynulování všech identifikačních příznaků a z předchozí činnosti není žádné přerušování zachyceno, viz dále):

- globální povolení vyvolání přerušování registrem INTEnReg (musí zůstat povoleno po dobu zachytávání přerušování)
- povolení vybraných zdrojů přerušování registrem IRQCfgReg
- cyklické použití funkce **\*\*\*\_IRQ\_Wait** v samostatném vlákně; v případě návratového kódu **\*\*\*\_OK** bude spuštěna vlastní obsluha vyvolaného přerušování (popř. více současně vyvolaných přerušování) složená z kroků
  - identifikace zdroje přerušování pomocí registru IRQStatusReg
  - vlastní zpracování dat (například vyčtení naměřených dat z FIFO paměti)

- vynulování příznaku přerušení registrem IRQClrReg

Globální povolení vyvolání přerušení registrem INTEnReg musí zůstat po celou dobu povoleno!

- zakázání všech zdrojů přerušení registrem IRQCfgReg
- vynulování všech případných příznaků přerušení registrem IRQClrReg
- globální zakázání vyvolání přerušení registrem INTEnReg

Posloupnost operací popsaná v předešlých odstavcích předpokládá jako výchozí stav zakázání všech zdrojů přerušení, vynulování všech identifikačních příznaků a žádné zachycené přerušení z předchozí činnosti karty. Tohoto stavu lze dosáhnout postupem:

- zakázání všech zdrojů přerušení registrem IRQCfgReg
- globální zakázání vyvolání přerušení registrem INTEnReg
- opakované použití funkce `***_IRQ_Wait` až do okamžiku návratového kódu `***_WaitTimeout`; volání funkce proběhne jedenkrát (žádné zachycené přerušení z předchozí činnosti) nebo dvakrát (poprvé návratový kód `***_OK` a podruhé `***_WaitTimeout`); vhodné je nastavení krátkého timeoutu; tuto část není potřeba provádět, pokud bude ukončeno spojení s kartou pomocí funkce `***_Close`

## 4.5 Uzavření spojení s kartou

Pro ukončení spojení s kartou je určena funkce `***_Close`. Předcházet by však mělo:

- nastavení registrů karty do bezpečného stavu (například ukončit měření a ukládání dat do FIFO paměti), zejména pak registrů souvisejících s přerušením (viz popis v předešlých odstavcích).
- ukončení vlákna obsluhující přerušení, tzn. využívajícího funkci `***_IRQ_Wait`, v každém případě je však nezbytné vyhodnocovat návratový kód `***_StopInThread`.

Po provedení funkce `***_Close` již nelze přistupovat k registrům karty ani obsluhovat přerušení; pokud má program znovu k registrům karty přistupovat, je nezbytné znovu otevřít spojení s kartou. V tomto místě je však vhodné zdůraznit, že **při každém otevření je přidělena nová adresa bloku registrů**.

Prázdná strana

## A1.1 Veřejné funkce knihovny tedia\_ox9162.dll verze 3.00 a vyšší

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 kartu (použitelné i před otevřením spojení s kartou)	
funkce	popis
OX9162_InterfaceVersion	verze interface ovladače (tzn. knihovny tedia_ox9162.dll)
OX9162_DriverVersion	verze systémové části ovladače (tzn. tedia_OxPCI.sys)
OX9162_CardsInSystem	počet karet obsluhovaných knihovnou tedia_ox9162.dll
Funkce pro zjištění vlastností karty (použitelné i před otevřením spojení s kartou)	
funkce	popis
OX9162_Card_DID	PCI Device ID zvolené karty (v kombinaci s TEDIA VID jednoznačně definuje typ karty)
OX9162_Card_BUS	číslo PCI sběrnice, ve které je zvolená karta instalována
OX9162_Card_SLOT	číslo slotu, ve kterém je zvolená karta instalována
Funkce související s otevřením/zavřením spojení s kartou	
funkce	popis
OX9162_Open	otevře <u>výhradní</u> spojení se zvolenou kartou (tzn. neumožní otevření spojení dalším programům funkcemi OX9162_Open ani OX9162_OpenMulti)
OX9162_OpenMulti	otevře <u>nevýhradní</u> spojení se zvolenou kartou (tzn. umožní otevření spojení dalším programům funkcí OX9162_OpenMulti, ne však OX9162_Open)
OX9162_Card_OpenCount	aktuální počet otevření spojení funkcí OX9162_OpenMulti
OX9162_Close	uzavře spojení s kartou
Funkce určené pro obsluhu registrů karty (vyžadují otevření spojení s kartou)	
funkce	popis
OX9162_BAR4_BaseAdr	počáteční adresa prostoru paměti, ve kterém jsou mapovány funkční registry karty (tzn. BAR4)
OX9162_MemRead	funkce pro čtení registru karty (nutné pouze pro vývojová prostředí nepodporující přímý přístup k paměti)
OX9162_MemWrite	funkce pro zápis do registru karty (nutné pouze pro vývojová prostředí nepodporující přímý přístup k paměti)
Funkce určené pro zpracování přerušení (vyžadují otevření spojení s kartou)	
funkce	popis
OX9162_IRQ_Wait	pozastaví thread až do okamžiku přerušení nebo vypršení timeoutu
OX9162_IRQ_SetTimeout	nastavuje funkci OX9162_IRQ_Wait mezní dobu čekání na přerušení (timeout)
OX9162_IRQ_Count	předává počet přerušení vyvolaných kartou

## A1.2 Návrátové kódy funkcí knihovny tedia\_ox9162.dll verze 3.00 a vyšší

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

Přehled návratových kódů	
návratový kód	popis
OX9162_Ok	Funkce proběhla bez chyb.
OX9162_BadIndex	Zadaný index karty je buď mimo povolený rozsah, nebo v seznamu nalezených karet neexistuje karta s tímto indexem.
OX9162_AlreadyOpened	Spojení s kartou již bylo otevřeno jiným procesem.
OX9162_OpenedSameProcess	Spojení s kartou již bylo otevřeno v rámci aktuálního procesu.
OX9162_CardOpenErr	Při otvírání spojení s kartou nastala neočekávaná chyba.
OX9162_IntEnableErr	Při pokusu o povolení přerušení pro kartu nastala chyba.
OX9162_NotOpened	Spojení s kartou ještě nebylo otevřeno.
OX9162_BAR4Err	BAR4 není dostupný.
OX9162_IntDisabled	Obsluha přerušení není povolena (tzn. spojení s kartou bylo otevřeno funkcí OX9162_Open nebo OX9162_OpenMulti s parametrem enable_int=0).
OX9162_StopIntThread	Poslední přerušení systému nebo timeout, je nutné ukončit provádění threadu, protože došlo k uzavření spojení s kartou. Tento návratový kód není chybový, má pouze informativní charakter.
OX9162_NullPointer	V parametru funkce byl předán ukazatel na nulovou adresu.
OX9162_CreateEvent	Při čekání na přerušení systému nastala neočekávaná chyba.
OX9162_WaitTimeout	Funkce OX9162_IRQ_Wait byla ukončena vypršením nastaveného timeoutu.
OX9162_VersionConflict	Konflikt verze systémové části ovladače a interface ovladače.

## A2.1 OX9162\_InterfaceVersion

```
DWORD OX9162_InterfaceVersion ( VOID );
```

### Popis

Funkce je určena pro zjištění verze interface ovladače (tzn. knihovny tedia\_ox9162.dll); například v případě verze 3.06 je předána hodnota 306 a před zobrazením je tedy potřeba provést posun o dvě desetinná místa doprava.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### Návratové kódy

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

## A2.2 OX9162\_DriverVersion

```
OX9162_RET OX9162_DriverVersion (DWORD *ver);
```

**\*ver**

Ukazatel na oblast paměti, do které je funkcí uložena hodnota verze systémové části ovladače.

---

### Popis

Funkce je určena pro zjištění verze systémové části ovladače (tzn. tedia\_OxPCI.sys); například v případě verze 1.10 je předána hodnota 110 a před zobrazením je tedy potřeba provést posun o dvě desetinná místa doprava.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.



## A2.3 OX9162\_CardsInSystem

```
OX9162_RET OX9162_CardsInSystem ( DWORD *count );
```

### **\*count**

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující počet dostupných karet v systému.

### **Popis**

Funkce je určena pro zjištění počtu karet podporovaných ovladačem (resp. obsluhovaných knihovnou tedia\_ox9162.dll) instalovaných v počítači.

Proměnná nabývá hodnoty v rozsahu od nuly (není dostupná žádná karta) až do skutečného počtu karet v počítači.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### **Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.4 OX9162\_Card\_DID

```
OX9162_RET OX9162_Card_DID      ( DWORD inx,  
                                   DWORD *DID );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*DID*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující "PCI Device ID" zvolené karty.

---

### Popis

Funkce je určena pro zjištění PCI Device ID zvolené karty (v kombinaci s TEDIA VID jednoznačně definuje typ karty). Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *OX9162\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX9162\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1). Hodnota *DID* odpovídá PCI Device ID (16bitové celé číslo) přidělenému danému typu karty. Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.5 OX9162\_Card\_BUS

```
OX9162_RET OX9162_Card_BUS      ( DWORD inx,  
                                   DWORD *bus );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*bus*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující číslo PCI sběrnice, ve které je zvolená karta instalována.

---

### Popis

Funkce je určena pro zjištění čísla PCI sběrnice, ve které je zvolená karta instalována; v kombinaci s hodnotou *slot* předanou funkcí *OX9162\_Card\_SLOT* je jednoznačně definováno umístění karty v počítači (kombinace hodnot *bus* a *slot* je v počítači unikátní a umožňuje proto odlišit dvě karty stejného typu).

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *OX9162\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX9162\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.6 OX9162\_Card\_SLOT

```
OX9162_RET OX9162_Card_SLOT      ( DWORD inx,  
                                   DWORD *slot );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*slot*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující číslo PCI slotu, ve kterém je zvolená karta instalována.

---

### Popis

Funkce je určena pro zjištění čísla PCI slotu, ve kterém je zvolená karta instalována; v kombinaci s hodnotou *bus* předanou funkcí *OX9162\_Card\_BUS* je jednoznačně definováno umístění karty v počítači (kombinace hodnot *bus* a *slot* je v počítači unikátní a umožňuje proto odlišit dvě karty stejného typu).

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *OX9162\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX9162\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.7 OX9162\_Open

```
OX9162_RET OX9162_Open          ( DWORD inx,  
                                   BOOL enable_int );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *enable\_int*

Parametr povoluje hodnotou true možnost vyvolání přerušení systému kartou (false přerušení systému blokuje).

### Popis

Funkce otevře výhradní spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí být zadán v rozsahu od 0 do hodnoty předané funkcí *OX9162\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX9162\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Parametr *enable\_int* povoluje hodnotou true možnost vyvolání přerušení systému kartou; pokud karta neumožňuje přerušení vyvolat, je při pokusu o povolení přerušení předán návratový kód *OX9162\_IntEnableErr*. Přerušení systému podporují všechny standardní karty a volba false je jen rezervou pro nestandardní typy.

Přístup ke kartě je výhradní; při pokusu o otevření spojení s kartou, která je v daný okamžik používána jiným procesem (tzn. jiný proces otevřel, avšak ještě neuzavřel spojení), z dalšího procesu funkcemi *OX9162\_Open* nebo *OX9162\_OpenMulti* je předán návratový kód *OX9162\_AlreadyOpened*. Při analogickém pokusu o otevření spojení s kartou ze stejného procesu je předán návratový kód *OX9162\_OpenedSameProcess*.

V případě potřeby přistupovat ke kartě souběžně z více procesů lze použít funkci *OX9162\_OpenMulti*.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.8 OX9162\_OpenMulti (podporováno od verze interface 3.03)

**OX9162\_RET OX9162\_OpenMulti** ( **DWORD** *inx*,  
**BOOL** *enable\_int* );

### **inx**

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### **enable\_int**

Parametr povoluje hodnotou true možnost vyvolání přerušení systému kartou (false přerušení systému blokuje).

### **Popis**

Funkce otevře nevýhradní spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí být zadán v rozsahu od 0 do hodnoty předané funkcí *OX9162\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX9162\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má význam pro hodnoty 0 a 1).

Parametr *enable\_int* povoluje hodnotou true možnost vyvolání přerušení systému kartou; pokud karta neumožňuje přerušení vyvolat, je při pokusu o povolení přerušení předán návratový kód *OX9162\_IntEnableErr*. Přerušení systému podporují všechny standardní karty a volba false je jen rezervou pro nestandardní typy.

Přístup ke kartě je nevýhradní; při pokusu o otevření spojení s kartou, která je v daný okamžik používána nevýhradně jiným procesem (tzn. jiný proces otevřel pomocí funkce *OX9162\_OpenMulti*, avšak ještě neuzavřel spojení), z dalšího procesu funkcí *OX9162\_OpenMulti* je otevřeno další souběžné spojení. Při pokusu o otevření spojení s kartou funkcí *OX9162\_Open* je otevření spojení odmítnuto s návratovým kódem *OX9162\_AlreadyOpened*.

### **Podporované karty**

Funkce je implementována od verze 3.03 pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

**Upozornění:** *Přístupy všech procesů jsou symetrické a v případě ovládání stejných funkcí mohou vyvolat vzájemné kolize. Není-li přístup z více procesů nezbytný, je vhodné otevřít s kartou výhradní spojení (tedy s jistotou, že jiný proces nemůže ovlivnit stav registrů karty) pomocí funkce *OX9162\_Open*.*

## A2.9 OX9162\_Card\_OpenCount (podporováno od verze interface 3.03)

```
OX9162_RET OX9162_Card_OpenCount ( DWORD inx,  
                                     DWORD *count );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*count*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota proměnné udávající aktuální počet otevřených spojení s požadovanou kartou.

---

### Popis

Funkce je určena pro zjištění aktuálního počtu otevřených spojení s požadovanou kartou pomocí funkce *OX9162\_OpenMulti*, případně *OX9162\_Open* (počet otevřených spojení je vždy roven jedné).

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována od verze 3.03 pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.10 OX9162\_Close

```
OX9162_RET OX9162_Close ( DWORD inx );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

---

### Popis

Funkce uzavře spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

*Poznámka: Při ukončení programu jsou automaticky uzavřena všechna otevřená spojení tohoto programu s kartou (v případě otevřených spojení s více kartami jsou uzavřena všechna spojení).*

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.



## A2.11 OX9162\_BAR4\_BaseAdr

```
OX9162_RET OX9162_BAR4_BaseAdr ( DWORD inx,  
                                   size_t *base_adr,  
                                   DWORD *range );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*base\_adr*

Ukazatel na oblast paměti, do které je funkcí uložena počáteční adresa bloku paměti pro přístup k registrům karty.

### *\*range*

Ukazatel na oblast paměti, do které je funkcí uložena velikost bloku paměti pro přístup k registrům karty.

---

### Popis

Funkce je určena pro zjištění počáteční adresy a velikosti bloku paměti pro přístup k funkčním registrům karty.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

*Poznámka:* Pro přístup k registrům lze použít funkce *OX9162\_MemRead* a *OX9162\_MemWrite*.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.12 OX9162\_MemRead

```
BYTE OX9162_MemRead ( size_t base_adr,  
                      DWORD offset );
```

### *base\_adr*

Specifikuje počáteční adresu bloku paměti pro přístup k registrům karty (tzn. adresa zjištěná funkcí *OX9162\_BAR4\_BaseAdr* po otevření spojení s požadovanou kartou).

### *offset*

Specifikuje adresu požadovaného registru karty v rámci bloku paměti.

---

### Popis

Funkce slouží pro čtení obsahu registrů karty zejména v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic).

Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot *base\_adr* a *offset*.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

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

**Upozornění:** *Funkce neblokuje možnost přístupu na adresy mimo přidělený prostor kartě a tedy způsobení fatální chyby běhu programu; veškerá odpovědnost za korektní adresu je proto na straně programu.*

## A2.13 OX9162\_MemWrite

```
VOID OX9162_MemWrite          ( size_t base_adr,  
                                DWORD offset,  
                                BYTE data );
```

### **base\_adr**

Specifikuje počáteční adresu bloku paměti pro přístup k registrům karty (tzn. adresa zjištěná funkcí *OX9162\_BAR4\_BaseAdr* po otevření spojení s požadovanou kartou).

### **offset**

Specifikuje adresu požadovaného registru karty v rámci bloku paměti.

### **data**

Předává data určená pro zápis do požadovaného registru karty.

---

### **Popis**

Funkce slouží pro zápis do registrů karty zejména v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic).

Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot *base\_adr* a *offset*.

### **Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### **Návratové kódy**

Funkce nevyužívá žádný návratový kód.

**Upozornění:** *Funkce neblokuje možnost přístupu na adresy mimo přidělený prostor kartě a tedy způsobení fatální chyby běhu programu; veškerá odpovědnost za korektní adresu je proto na straně programu.*

## A2.14 OX9162\_IRQ\_Wait

```
OX9162_RET OX9162_IRQ_Wait      ( DWORD inx,  
                                   DWORD * );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

\*

Druhý parametr není od verze 1.55 využit (zůstal zachován z důvodu zpětné kompatibility).

### Popis

Funkce pozastaví thread pro zpracování přerušení až do okamžiku, kdy ovladač zachytí přerušení systému vyvolané kartou nebo do vypršení mezní doby čekání na vyvolání přerušení (viz popis funkce *OX9162\_IRQ\_SetTimeout*).

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

Funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule) v případě, že proběhla bez chyb a bylo vyvoláno přerušení systému, v opačném případě pak předává některý z nenulových chybových návratových kódů.

Návratový kód *OX9162\_WaitTimeout* signalizuje, že funkce proběhla bez chyb, avšak nebylo vyvoláno přerušení systému a funkce skončila vypršením mezní doby čekání na vyvolání přerušení (timeoutem).

Pokud však funkce předá návratový kód *OX9162\_StopInThread*, je nutné ukončit provádění threadu, protože došlo k uzavření spojení s kartou.

## A2.15 OX9162\_IRQ\_SetTimeout

```
OX9162_RET OX9162_IRQ_SetTimeout ( DWORD inx,  
                                     DWORD timeout );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *timeout*

Definuje mezní dobu čekání na přerušení, po kterou funkce *OX9162\_IRQ\_Wait* pozastaví thread. Hodnota je zadávána v milisekundách.

---

### Popis

Funkce nastaví mezní dobu čekání na vyvolání přerušení. Není-li přerušení vyvoláno před mezní dobu, je funkce *OX9162\_IRQ\_Wait* ukončena (tzn. pozastavený thread uvolněn) s návratovým kódem *OX9162\_WaitTimeout*.

Je-li hodnota *timeout* nastavena na nulu, funkce *OX9162\_IRQ\_Wait* detekuje, zda bylo vyvoláno přerušení od okamžiku předešlého provedení funkce *OX9162\_IRQ\_Wait* a bez dalšího čekání je běh této funkce ukončen s návratovým kódem *OX9162\_OK* nebo *OX9162\_WaitTimeout*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

Funkce předává návratový kód *OX9162\_OK* (hodnota rovna nule) v případě, že proběhla bez chyb, v opačném případě pak předává některý z nenulových chybových návratových kódů.

## A2.16 OX9162\_IRQ\_Count

**DWORD OX9162\_IRQ\_Count (DWORD *inx*);**

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### Popis

Funkce je určena pro zjištění počtu ovladačem zachycených přerušení od okamžiku otevření spojení s kartou některou z funkcí *OX9162\_Open* nebo funkcí *OX9162\_OpenMulti*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

Funkci *OX9162\_IRQ\_Count* je možné použít i v případě, kdy nebylo při otevření spojení funkcí *OX9162\_OpenMulti* vyvolání přerušení povoleno; může totiž nastat situace, kdy jiný program otevře (nebo již otevřel) spojení s kartou s povoleným přerušením.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox9162.dll.

### Návratové kódy

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

V případě platného parametru *inx* je předáván počet ovladačem zachycených přerušení, v případě neplatného parametru *inx* je předávána hodnota 0.

## B1.1 Veřejné funkce knihovny tedia\_ox952.dll verze 3.00 a vyšší

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 kartu (použitelné i před otevřením spojení s kartou)	
funkce	popis
OX952_InterfaceVersion	verze interface ovladače (tzn. knihovny tedia_ox952.dll)
OX952_DriverVersion	verze systémové části ovladače (tzn. tedia_OxPCI.sys)
OX952_CardsInSystem	počet karet obsluhovaných knihovnou tedia_ox952.dll
Funkce pro zjištění vlastností karty (použitelné i před otevřením spojení s kartou)	
funkce	popis
OX952_Card_DID	PCI Device ID zvolené karty (v kombinaci s TEDIA VID jednoznačně definuje typ karty)
OX952_Card_BUS	číslo PCI sběrnice, ve které je zvolená karta instalována
OX952_Card_SLOT	číslo slotu, ve kterém je zvolená karta instalována
Funkce související s otevřením/zavřením spojení s kartou	
funkce	popis
OX952_Open	otevře <u>výhradní</u> spojení se zvolenou kartou (tzn. neumožní otevření spojení dalším programům funkcemi OX952_Open ani OX952_OpenMulti)
OX952_OpenMulti	otevře <u>nevýhradní</u> spojení se zvolenou kartou (tzn. umožní otevření spojení dalším programům funkcí OX952_OpenMulti, ne však OX952_Open)
OX952_Card_OpenCount	aktuální počet otevření spojení funkcí OX952_OpenMulti
OX952_Close	uzavře spojení s kartou
Funkce určené pro obsluhu registrů karty (vyžadují otevření spojení s kartou)	
funkce	popis
OX952_BAR05_BaseAdr	počáteční adresa prostoru paměti, ve kterém jsou mapovány registry sériového portu určeného pro servisní účely (tzn. BAR5, PCI funkce 0) (tato funkce není pro běžné aplikace potřebná)
OX952_BAR11_BaseAdr	počáteční adresa prostoru paměti, ve kterém jsou mapovány funkční registry karty (tzn. BAR1, PCI funkce 1)
OX952_MemRead	funkce pro čtení registru karty (nutné pouze pro vývojová prostředí nepodporující přímý přístup k paměti)
OX952_MemWrite	funkce pro zápis do registru karty (nutné pouze pro vývojová prostředí nepodporující přímý přístup k paměti)
Funkce určené pro zpracování přerušení (vyžadují otevření spojení s kartou)	
funkce	popis
OX952_IRQ_Wait	pozastaví thread až do okamžiku přerušení nebo vypršení timeoutu
OX952_IRQ_SetTimeout	nastavuje funkci OX952_IRQ_Wait mezní dobu čekání na přerušení (timeout)
OX952_IRQ_Count	předává počet přerušení vyvolaných kartou

## B1.2 Návrátové kódy funkcí knihovny tedia\_ox952.dll verze 3.00 a vyšší

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

Přehled návratových kódů	
návratový kód	popis
OX952_Ok	Funkce proběhla bez chyb.
OX952_BadIndex	Zadaný index karty je buď mimo povolený rozsah, nebo v seznamu nalezených karet neexistuje karta s tímto indexem.
OX952_AlreadyOpened	Spojení s kartou již bylo otevřeno jiným procesem.
OX952_OpenedSameProcess	Spojení s kartou již bylo otevřeno v rámci aktuálního procesu.
OX952_CardOpenErr	Při otevírání spojení s kartou nastala neočekávaná chyba.
OX952_IntEnableErr	Při pokusu o povolení přerušení pro kartu nastala chyba.
OX952_NotOpened	Spojení s kartou ještě nebylo otevřeno.
OX952_BAR05Err	BAR5 PCI funkce 0 není dostupný.
OX952_BAR11Err	BAR1 PCI funkce 1 není dostupný.
OX952_IntDisabled	Obsluha přerušení není povolena (tzn. spojení s kartou bylo otevřeno funkcí OX952_Open nebo OX952_OpenMulti s parametrem enable_int=0).
OX952_StopIntThread	Poslední přerušení systému nebo timeout, je nutné ukončit provádění threadu, protože došlo k uzavření spojení s kartou. Tento návratový kód není chybový, má pouze informativní charakter.
OX952_NullPointer	V parametru funkce byl předán ukazatel na nulovou adresu.
OX952_CreateEvent	Při čekání na přerušení systému nastala neočekávaná chyba.
OX952_WaitTimeout	Funkce OX952_IRQ_Wait byla ukončena vypršením nastaveného timeoutu.
OX952_VersionConflict	Konflikt verze systémové části ovladače a interface ovladače.
OX952_NoCoupledDevice	Chyba při párování PCI funkcí v rámci jedné karty (pravděpodobně nebyl korektně nainstalovaný systémový ovladač pro obě PCI funkce).



## B2.1 OX952\_InterfaceVersion

```
DWORD OX952_InterfaceVersion ( VOID );
```

### Popis

Funkce je určena pro zjištění verze interface ovladače (tzn. knihovny tedia\_ox952.dll); například v případě verze 3.06 je předána hodnota 306 a před zobrazením je tedy potřeba provést posun o dvě desetinná místa doprava.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### Návratové kódy

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

## B2.2 OX952\_DriverVersion

```
OX952_RET OX952_DriverVersion ( DWORD *ver );
```

**\*ver**

Ukazatel na oblast paměti, do které je funkcí uložena hodnota verze systémové části ovladače.

---

### Popis

Funkce je určena pro zjištění verze systémové části ovladače (tzn. tedia\_OxPCI.sys); například v případě verze 1.10 je předána hodnota 110 a před zobrazením je tedy potřeba provést posun o dvě desetinná místa doprava.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód **OX952\_OK** (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.3 OX952\_CardsInSystem

```
OX952_RET OX952_CardsInSystem ( DWORD *count );
```

### **\*count**

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující počet dostupných karet v systému.

### **Popis**

Funkce je určena pro zjištění počtu karet podporovaných ovladačem (resp. obsluhovaných knihovnou tedia\_ox952.dll) instalovaných v počítači.

Proměnná nabývá hodnoty v rozsahu od nuly (není dostupná žádná karta) až do skutečného počtu karet v počítači.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### **Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód **OX952\_OK** (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.4 OX952\_Card\_DID

```
OX952_RET OX952_Card_DID          ( DWORD inx,  
                                   DWORD *DID );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*DID*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující "PCI Device ID" zvolené karty.

---

### Popis

Funkce je určena pro zjištění PCI Device ID zvolené karty (v kombinaci s TEDIA VID jednoznačně definuje typ karty).

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *OX952\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX952\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Hodnota *DID* odpovídá PCI Device ID (16bitové celé číslo) přidělenému danému typu karty.

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.5 OX952\_Card\_BUS

```
OX952_RET OX952_Card_BUS          ( DWORD inx,  
                                     DWORD *bus );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*bus*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující číslo PCI sběrnice, ve které je zvolená karta instalována.

---

### Popis

Funkce je určena pro zjištění čísla PCI sběrnice, ve které je zvolená karta instalována; v kombinaci s hodnotou *slot* předanou funkcí *OX952\_Card\_SLOT* je jednoznačně definováno umístění karty v počítači (kombinace hodnot *bus* a *slot* je v počítači unikátní a umožňuje proto odlišit dvě karty stejného typu).

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *OX952\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX952\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.6 OX952\_Card\_SLOT

```
OX952_RET OX952_Card_SLOT          ( DWORD inx,  
                                     DWORD *slot );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*slot*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující číslo PCI slotu, ve kterém je zvolená karta instalována.

---

### Popis

Funkce je určena pro zjištění čísla PCI slotu, ve kterém je zvolená karta instalována; v kombinaci s hodnotou *bus* předanou funkcí *OX952\_Card\_BUS* je jednoznačně definováno umístění karty v počítači (kombinace hodnot *bus* a *slot* je v počítači unikátní a umožňuje proto odlišit dvě karty stejného typu).

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *OX952\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX952\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.7 OX952\_Open

**OX952\_RET OX952\_Open** ( **DWORD** *inx*,  
**BOOL** *enable\_int* );

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *enable\_int*

Parametr povoluje hodnotou true možnost vyvolání přerušení systému kartou (false přerušení systému blokuje).

### Popis

Funkce otevře výhradní spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí být zadán v rozsahu od 0 do hodnoty předané funkcí *OX952\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX952\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Parametr *enable\_int* povoluje hodnotou true možnost vyvolání přerušení systému kartou; pokud karta neumožňuje přerušení vyvolat, je při pokusu o povolení přerušení předán návratový kód *OX952\_IntEnableErr*. Přerušení systému podporují všechny standardní karty a volba false je jen rezervou pro nestandardní typy.

Přístup ke kartě je výhradní; při pokusu o otevření spojení s kartou, která je v daný okamžik používána jiným procesem (tzn. jiný proces otevřel, avšak ještě neuzavřel spojení), z dalšího procesu funkcemi *OX952\_Open* nebo *OX952\_OpenMulti* je předán návratový kód *OX952\_AlreadyOpened*. Při analogickém pokusu o otevření spojení s kartou ze stejného procesu je předán návratový kód *OX952\_OpenedSameProcess*.

V případě potřeby přistupovat ke kartě souběžně z více procesů lze použít funkci *OX952\_OpenMulti*.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.8 OX952\_OpenMulti (podporováno od verze interface 3.03)

**OX952\_RET OX952\_OpenMulti** ( **DWORD** *inx*,  
**BOOL** *enable\_int* );

### **inx**

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### **enable\_int**

Parametr povoluje hodnotou true možnost vyvolání přerušení systému kartou (false přerušení systému blokuje).

### **Popis**

Funkce otevře nevýhradní spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí být zadán v rozsahu od 0 do hodnoty předané funkcí *OX952\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *OX952\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má význam pro hodnoty 0 a 1).

Parametr *enable\_int* povoluje hodnotou true možnost vyvolání přerušení systému kartou; pokud karta neumožňuje přerušení vyvolat, je při pokusu o povolení přerušení předán návratový kód *OX952\_IntEnableErr*. Přerušení systému podporují všechny standardní karty a volba false je jen rezervou pro nestandardní typy.

Přístup ke kartě je nevýhradní; při pokusu o otevření spojení s kartou, která je v daný okamžik používána nevýhradně jiným procesem (tzn. jiný proces otevřel pomocí funkce *OX952\_OpenMulti*, avšak ještě neuzavřel spojení), z dalšího procesu funkcí *OX952\_OpenMulti* je otevřeno další souběžné spojení. Při pokusu o otevření spojení s kartou funkcí *OX952\_Open* je otevření spojení odmítnuto s návratovým kódem *OX952\_AlreadyOpened*.

### **Podporované karty**

Funkce je implementována od verze 3.03 pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

**Upozornění:** *Přístupy všech procesů jsou symetrické a v případě ovládání stejných funkcí mohou vyvolat vzájemné kolize. Není-li přístup z více procesů nezbytný, je vhodné otevřít s kartou výhradní spojení (tedy s jistotou, že jiný proces nemůže ovlivnit stav registrů karty) pomocí funkce *OX952\_Open*.*



## B2.9 OX952\_Card\_OpenCount (podporováno od verze interface 3.03)

```
OX952_RET OX952_Card_OpenCount ( DWORD inx,  
                                   DWORD *count );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*count*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota proměnné udávající aktuální počet otevřených spojení s požadovanou kartou.

---

### Popis

Funkce je určena pro zjištění aktuálního počtu otevřených spojení s požadovanou kartou pomocí funkce *OX952\_OpenMulti*, případně *OX952\_Open* (počet otevřených spojení je vždy roven jedné).

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována od verze 3.03 pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.10 OX952\_Close

```
OX952_RET OX952_Close ( DWORD inx );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

---

### Popis

Funkce uzavře spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

*Poznámka: Při ukončení programu jsou automaticky uzavřena všechna otevřená spojení tohoto programu s kartou (v případě otevřených spojení s více kartami jsou uzavřena všechna spojení).*

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.11 OX952\_BAR05\_BaseAdr, OX952\_BAR11\_BaseAdr

```
OX952_RET OX952_BAR05_BaseAdr    ( DWORD inx,
                                     size_t *base_adr,
                                     DWORD *range );

OX952_RET OX952_BAR11_BaseAdr    ( DWORD inx,
                                     size_t *base_adr,
                                     DWORD *range );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*base\_adr*

Ukazatel na oblast paměti, do které je funkcí uložena počáteční adresa bloku paměti pro přístup k registrům karty.

### *\*range*

Ukazatel na oblast paměti, do které je funkcí uložena velikost bloku paměti pro přístup k registrům karty.

### Popis

Funkce *OX952\_BAR05\_BaseAdr* je určena pro zjištění počáteční adresy a velikosti bloku paměti pro přístup k registrům sériového portu určeného pro servisní účely (tato funkce není pro běžné aplikace potřebná).

*Poznámka: Sériový port je primárně určen pro servisní účely (například uživatelská aktualizace firmware), ale slouží i k zjištění výrobního čísla karty.*

Funkce *OX952\_BAR11\_BaseAdr* je určena pro zjištění počáteční adresy a velikosti bloku paměti pro přístup k funkčním registrům karty.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

*Poznámka: Pro přístup k registrům lze použít funkce *OX952\_MemRead* a *OX952\_MemWrite*.*

### Podporované karty

Funkce *OX952\_BAR05\_BaseAdr* je implementována pro většinu typů karet podporované knihovnou tedia\_ox952.dll (viz příloha D této příručky).

Funkce *OX952\_BAR11\_BaseAdr* je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *OX952\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.12 OX952\_MemRead

```
BYTE OX952_MemRead          ( size_t base_adr,  
                               DWORD offset );
```

### *base\_adr*

Specifikuje počáteční adresu bloku paměti pro přístup k registrům karty (tzn. adresa zjištěná funkcí *OX952\_BAR05\_BaseAdr* nebo *OX952\_BAR11\_BaseAdr*).

### *offset*

Specifikuje adresu požadovaného registru karty v rámci bloku paměti.

---

### Popis

Funkce slouží pro čtení obsahu registrů karty zejména v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic).

Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot *base\_adr* a *offset*.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

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

**Upozornění:** *Funkce neblokuje možnost přístupu na adresy mimo přidělený prostor kartě a tedy způsobení fatální chyby běhu programu; veškerá odpovědnost za korektní adresu je proto na straně programu.*

## B2.13 OX952\_MemWrite

```
VOID OX952_MemWrite          ( size_t base_adr,  
                               DWORD offset,  
                               BYTE data );
```

### **base\_adr**

Specifikuje počáteční adresu bloku paměti pro přístup k registrům karty (tzn. adresa zjištěná funkcí *OX952\_BAR05\_BaseAdr* nebo *OX952\_BAR11\_BaseAdr*).

### **offset**

Specifikuje adresu požadovaného registru karty v rámci bloku paměti.

### **data**

Předává data určená pro zápis do požadovaného registru karty.

---

### **Popis**

Funkce slouží pro zápis do registrů karty zejména v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic).

Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot *base\_adr* a *offset*.

### **Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### **Návratové kódy**

Funkce nevyužívá žádný návratový kód.

**Upozornění:** *Funkce neblokuje možnost přístupu na adresy mimo přidělený prostor kartě a tedy způsobení fatální chyby běhu programu; veškerá odpovědnost za korektní adresu je proto na straně programu.*

## B2.14 OX952\_IRQ\_Wait

```
OX952_RET OX952_IRQ_Wait          ( DWORD inx,  
                                     DWORD * );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

\*

Druhý parametr není od verze 1.55 využit (zůstal zachován z důvodu zpětné kompatibility).

### Popis

Funkce pozastaví thread pro zpracování přerušení až do okamžiku, kdy ovladač zachytí přerušení systému vyvolané kartou nebo do vypršení mezní doby čekání na vyvolání přerušení (viz popis funkce *OX952\_IRQ\_SetTimeout*).

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

Funkce předává návratový kód *OX952\_OK* (hodnota rovna nule) v případě, že proběhla bez chyb a bylo vyvoláno přerušení systému, v opačném případě pak předává některý z nenulových chybových návratových kódů.

Návratový kód *OX952\_WaitTimeout* signalizuje, že funkce proběhla bez chyb, avšak nebylo vyvoláno přerušení systému a funkce skončila vypršením mezní doby čekání na vyvolání přerušení (timeoutem).

Pokud však funkce předá návratový kód *OX952\_StopInThread*, je nutné ukončit provádění threadu, protože došlo k uzavření spojení s kartou.

## B2.15 OX952\_IRQ\_SetTimeout

```
OX952_RET OX952_IRQ_SetTimeout      ( DWORD inx,  
                                       DWORD timeout );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *timeout*

Definuje mezní dobu čekání na přerušení, po kterou funkce *OX952\_IRQ\_Wait* pozastaví thread. Hodnota je zadávána v milisekundách.

---

### Popis

Funkce nastaví mezní dobu čekání na vyvolání přerušení. Není-li přerušení vyvoláno před mezní dobu, je funkce *OX952\_IRQ\_Wait* ukončena (tzn. pozastavený thread uvolněn) s návratovým kódem *OX952\_WaitTimeout*.

Je-li hodnota *timeout* nastavena na nulu, funkce *OX952\_IRQ\_Wait* detekuje, zda bylo vyvoláno přerušení od okamžiku předešlého provedení funkce *OX952\_IRQ\_Wait* a bez dalšího čekání je běh této funkce ukončen s návratovým kódem *OX952\_OK* nebo *OX952\_WaitTimeout*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

Funkce předává návratový kód *OX952\_OK* (hodnota rovna nule) v případě, že proběhla bez chyb, v opačném případě pak předává některý z nenulových chybových návratových kódů.

## B2.16 OX952\_IRQ\_Count

**DWORD OX952\_IRQ\_Count (DWORD *inx*);**

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### Popis

Funkce je určena pro zjištění počtu ovladačem zachycených přerušení od okamžiku otevření spojení s kartou některou z funkcí *OX952\_Open* nebo funkcí *OX952\_OpenMulti*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

Funkci *OX952\_IRQ\_Count* je možné použít i v případě, kdy nebylo při otevření spojení funkcí *OX952\_OpenMulti* vyvolání přerušení povoleno; může totiž nastat situace, kdy jiný program otevře (nebo již otevřel) spojení s kartou s povoleným přerušením.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ox952.dll.

### Návratové kódy

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

V případě platného parametru *inx* je předáván počet ovladačem zachycených přerušení, v případě neplatného parametru *inx* je předávána hodnota 0.



## C1.1 Veřejné funkce knihovny tedia\_ep4gxa.dll verze 3.10 a vyšší

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 kartu (použitelné i před otevřením spojení s kartou)	
funkce	popis
EP4GXA_InterfaceVersion	verze interface ovladače (tzn. knihovny tedia_ep4gxa.dll)
EP4GXA_DriverVersion	verze systémové části ovladače (tzn. tedia_OxPCI.sys)
EP4GXA_CardsInSystem	počet karet obsluhovaných knihovnou tedia_ep4gxa.dll
Funkce pro zjištění vlastností karty (použitelné i před otevřením spojení s kartou)	
funkce	popis
EP4GXA_Card_DID	PCI Device ID zvolené karty (v kombinaci s TEDIA VID jednoznačně definuje typ karty)
EP4GXA_Card_BUS	číslo PCI sběrnice, ve které je zvolená karta instalována
EP4GXA_Card_SLOT	číslo slotu, ve kterém je zvolená karta instalována
EP4GXA_Card_ID (od verze 3.18)	hodnota CardID nastavená DIP spínačem na kartě
EP4GXA_Card_SerNr (od verze 3.18)	hodnota výrobního čísla karty
Funkce související s otevřením/zavřením spojení s kartou	
funkce	popis
EP4GXA_Open	otevře <u>výhradní</u> spojení se zvolenou kartou (tzn. neumožní otevření spojení dalším programům funkcemi EP4GXA_Open ani EP4GXA_OpenMulti)
EP4GXA_OpenMulti	otevře <u>nevýhradní</u> spojení se zvolenou kartou (tzn. umožní otevření spojení dalším programům funkcí EP4GXA_OpenMulti, ne však EP4GXA_Open)
EP4GXA_Card_OpenCount	aktuální počet otevření spojení funkcí EP4GXA_OpenMulti
EP4GXA_Close	uzavře spojení s kartou
Funkce určené pro obsluhu registrů karty (vyžadují otevření spojení s kartou)	
funkce	popis
EP4GXA_BAR0_BaseAdr	počáteční adresa a velikost bloku paměti, ve kterém jsou mapovány funkční registry karty (tzn. BAR0); nutné pro alternativu přímého mapování registrů
EP4GXA_ReadReg8 EP4GXA_ReadReg32	funkce pro čtení registru karty ve variantách pro 8bitová nebo 32bitová data (funkce nevyžaduje přímý přístup k paměti z programu)
EP4GXA_WriteReg8 EP4GXA_WriteReg32	funkce pro zápis do registru karty ve variantách pro 8bitová nebo 32bitová data (funkce nevyžaduje přímý přístup k paměti z programu)
EP4GXA_ReadReg64 EP4GXA_ReadReg2x32 (od verze 3.18)	funkce pro současné čtení dvou registrů karty ve variantách předávajících 64bitová data nebo dvojici 32bitových dat (funkce nevyžaduje přímý přístup k paměti z programu)
EP4GXA_WriteReg64 EP4GXA_WriteReg2x32 (od verze 3.18)	funkce pro současný zápis do dvou registrů karty ve variantách předávajících 64bitová data nebo dvojici 32bitových dat (funkce nevyžaduje přímý přístup k paměti z programu)
Funkce určené pro zpracování přerušení (vyžadují otevření spojení s kartou)	
funkce	popis
EP4GXA_IRQ_Wait	pozastaví thread až do okamžiku přerušení nebo vypršení timeoutu
EP4GXA_IRQ_SetTimeout	nastavuje funkci EP4GXA_IRQ_Wait mezní dobu čekání na přerušení (timeout)
EP4GXA_IRQ_Count	předává počet přerušení vyvolaných kartou

## C1.2 Návrátové kódy funkcí knihovny tedia\_ep4gxa.dll verze 3.10 a vyšší

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

Přehled návratových kódů	
návratový kód	popis
EP4GXA_Ok	Funkce proběhla bez chyb.
EP4GXA_BadIndex	Zadaný index karty je buď mimo povolený rozsah, nebo v seznamu nalezených karet neexistuje karta s tímto indexem.
EP4GXA_AlreadyOpened	Spojení s kartou již bylo otevřeno jiným procesem.
EP4GXA_OpenedSameProcess	Spojení s kartou již bylo otevřeno v rámci aktuálního procesu.
EP4GXA_CardOpenErr	Při otvírání spojení s kartou nastala neočekávaná chyba.
EP4GXA_IntEnableErr	Při pokusu o povolení přerušení pro kartu nastala chyba.
EP4GXA_NotOpened	Spojení s kartou ještě nebylo otevřeno.
EP4GXA_BARErr	BAR0 není dostupný (popř. další BAR prostory pro servisní funkce).
EP4GXA_IntDisabled	Obsluha přerušení není povolena (tzn. spojení s kartou bylo otevřeno funkcí EP4GXA_Open nebo EP4GXA_OpenMulti s parametrem enable_int=0).
EP4GXA_StopIntThread	Poslední přerušení systému nebo timeout, je nutné ukončit provádění threadu, protože došlo k uzavření spojení s kartou. Tento návratový kód není chybový, má pouze informativní charakter.
EP4GXA_NullPointer	V parametru funkce byl předán ukazatel na nulovou adresu.
EP4GXA_CreateEvent	Při čekání na přerušení systému nastala neočekávaná chyba.
EP4GXA_WaitTimeout	Funkce EP4GXA_IRQ_Wait byla ukončena vypršením nastaveného timeoutu.
EP4GXA_VersionConflict	Konflikt verze systémové části ovladače a interface ovladače.
EP4GXA_InvalidAddress	Neplatná adresa při volání funkcí EP4GXA_ReadReg8, EP4GXA_ReadReg32, EP4GXA_WriteReg8 nebo EP4GXA_WriteReg32

## C2.1 EP4GXA\_InterfaceVersion

```
DWORD EP4GXA_InterfaceVersion ( VOID );
```

### Popis

Funkce je určena pro zjištění verze interface ovladače (tzn. knihovny tedia\_ep4gxa.dll); například v případě verze 3.06 je předána hodnota 306 a před zobrazením je tedy potřeba provést posun o dvě desetinná místa doprava.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### Návratové kódy

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

## C2.2 EP4GXA\_DriverVersion

```
EP4GXA_RET EP4GXA_DriverVersion (DWORD *ver);
```

**\*ver**

Ukazatel na oblast paměti, do které je funkcí uložena hodnota verze systémové části ovladače.

---

### Popis

Funkce je určena pro zjištění verze systémové části ovladače (tzn. tedia\_OxPCI.sys); například v případě verze 1.10 je předána hodnota 110 a před zobrazením je tedy potřeba provést posun o dvě desetinná místa doprava.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód **EP4GXA\_OK** (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.3 EP4GXA\_CardsInSystem

```
EP4GXA_RET EP4GXA_CardsInSystem ( DWORD *count );
```

### **\*count**

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující počet dostupných karet v systému.

### **Popis**

Funkce je určena pro zjištění počtu karet podporovaných ovladačem (resp. obsluhovaných knihovnou tedia\_ep4gxa.dll) instalovaných v počítači.

Proměnná nabývá hodnoty v rozsahu od nuly (není dostupná žádná karta) až do skutečného počtu karet v počítači.

Funkce nevyžaduje předchozí otevření spojení s jakoukoliv kartou.

### **Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll, resp. není na přítomnosti a vlastnostech karet nijak závislá.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.4 EP4GXA\_Card\_DID

```
EP4GXA_RET EP4GXA_Card_DID      ( DWORD inx,  
                                DWORD *DID );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*DID*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující "PCI Device ID" zvolené karty.

---

### Popis

Funkce je určena pro zjištění PCI Device ID zvolené karty (v kombinaci s TEDIA VID jednoznačně definuje typ karty). Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *EP4GXA\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *EP4GXA\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1). Hodnota *DID* odpovídá PCI Device ID (16bitové celé číslo) přidělenému danému typu karty. Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.5 EP4GXA\_Card\_BUS

```
EP4GXA_RET EP4GXA_Card_BUS      ( DWORD inx,  
                                DWORD *bus );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*bus*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující číslo PCI Express sběrnice, ve které je zvolená karta instalována.

---

### Popis

Funkce je určena pro zjištění čísla PCI Express sběrnice, ve které je zvolená karta instalována; v kombinaci s hodnotou *slot* předanou funkcí *EP4GXA\_Card\_SLOT* je jednoznačně definováno umístění karty v počítači (kombinace hodnot *bus* a *slot* je v počítači unikátní a umožňuje proto odlišit dvě karty stejného typu).

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *EP4GXA\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *EP4GXA\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.6 EP4GXA\_Card\_SLOT

```
EP4GXA_RET EP4GXA_Card_SLOT      ( DWORD inx,  
                                   DWORD *slot );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*slot*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující číslo PCI Express slotu, ve kterém je zvolená karta instalována.

---

### Popis

Funkce je určena pro zjištění čísla PCI Express slotu, ve kterém je zvolená karta instalována; v kombinaci s hodnotou *bus* předanou funkcí *EP4GXA\_Card\_BUS* je jednoznačně definováno umístění karty v počítači (kombinace hodnot *bus* a *slot* je v počítači unikátní a umožňuje proto odlišit dvě karty stejného typu).

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *EP4GXA\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *EP4GXA\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.



## C2.7 EP4GXA\_Card\_ID (podporováno od verze interface 3.18)

```
EP4GXA_RET EP4GXA_Card_ID      ( DWORD inx,  
                                DWORD *id );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*id*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota definující hodnotu CardID.

---

### Popis

Funkce je určena pro zjištění hodnoty CardID (je nastavena DIP spínačem na kartě v rozsahu 0÷3) umožňující identifikovat kartu zejména v případě více karet stejného typu.

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *EP4GXA\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *EP4GXA\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.8 EP4GXA\_Card\_SerNr (podporováno od verze interface 3.18)

```
EP4GXA_RET EP4GXA_Card_SerNr      ( DWORD inx,  
                                   DWORD *sernr );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*sernr*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota výrobního čísla karty.

---

### Popis

Funkce je určena pro zjištění výrobního čísla karty umožňující 100% identifikovat kartu.

Program volí požadovanou kartu ze seznamu přítomných karet (jsou číslovány od nuly), parametr *inx* tedy musí být v rozsahu od 0 do hodnoty předané funkcí *EP4GXA\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *EP4GXA\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Funkce nevyžaduje předchozí otevření spojení se zvolenou kartou.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.9 EP4GXA\_Open

```
EP4GXA_RET EP4GXA_Open ( DWORD inx,  
                          BOOL enable_int );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *enable\_int*

Parametr povoluje hodnotou true možnost vyvolání přerušení systému kartou (false přerušení systému blokuje).

### Popis

Funkce otevře výhradní spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí být zadán v rozsahu od 0 do hodnoty předané funkcí *EP4GXA\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *EP4GXA\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má platný význam pro hodnoty 0 a 1).

Parametr *enable\_int* povoluje hodnotou true možnost vyvolání přerušení systému kartou; pokud karta neumožňuje přerušení vyvolat, je při pokusu o povolení přerušení předán návratový kód *EP4GXA\_IntEnableErr*. Přerušení systému podporují všechny standardní karty a volba false je jen rezervou pro nestandardní typy.

Přístup ke kartě je výhradní; při pokusu o otevření spojení s kartou, která je v daný okamžik používána jiným procesem (tzn. jiný proces otevřel, avšak ještě neuzavřel spojení; není podstatné, zda spojení bylo otevřeno funkcí *EP4GXA\_Open* nebo *EP4GXA\_OpenMulti*) je předán návratový kód *EP4GXA\_AlreadyOpened*.

Při analogickém pokusu o otevření spojení s kartou, se kterou již bylo ve stejném procesu otevřeno spojení pomocí *EP4GXA\_Open*, je předán návratový kód *EP4GXA\_OpenedSameProcess* a spojení není otevřeno.

V případě potřeby přistupovat ke kartě souběžně z více procesů lze použít funkci *EP4GXA\_OpenMulti* namísto funkce *EP4GXA\_Open*.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.10 EP4GXA\_OpenMulti

```
EP4GXA_RET EP4GXA_OpenMulti ( DWORD inx,  
                               BOOL enable_int );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *enable\_int*

Parametr povoluje hodnotou true možnost vyvolání přerušení systému kartou (false přerušení systému blokuje).

### Popis

Funkce otevře nevýhradní spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí být zadán v rozsahu od 0 do hodnoty předané funkcí *EP4GXA\_CardInSystem* zmenšenou o 1 (tzn. v případě dvou karet v počítači funkce *EP4GXA\_CardInSystem* vrátí hodnotu 2 a parametr *inx* má význam pro hodnoty 0 a 1).

Parametr *enable\_int* povoluje hodnotou true možnost vyvolání přerušení systému kartou; pokud karta neumožňuje přerušení vyvolat, je při pokusu o povolení přerušení předán návratový kód *EP4GXA\_IntEnableErr*. Přerušení systému podporují všechny standardní karty a volba false je jen rezervou pro nestandardní typy.

Přístup ke kartě je nevýhradní; při pokusu o otevření spojení s kartou, která je v daný okamžik používána nevýhradně jiným procesem (tzn. jiný proces otevřel spojení pomocí funkce *EP4GXA\_OpenMulti* a ještě jej neuzavřel), z dalšího procesu funkcí *EP4GXA\_OpenMulti* je otevřeno další souběžné spojení. Pokud však bylo první spojení s kartou otevřeno jako výhradní (tzn. pomocí funkce *EP4GXA\_Open*), pokus o druhé otevření pomocí *EP4GXA\_OpenMulti* bude neúspěšné s návratovým kódem *EP4GXA\_AlreadyOpened*.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

**Upozornění:** *Přístupy všech procesů jsou symetrické a v případě ovládání stejných funkcí mohou vyvolat vzájemné kolize. Není-li přístup z více procesů nezbytný, je vhodné otevřít s kartou výhradní spojení (tedy s jistotou, že jiný proces nemůže ovlivnit stav registrů karty) pomocí funkce *EP4GXA\_Open*.*

## C2.11 EP4GXA\_Card\_OpenCount

```
EP4GXA_RET EP4GXA_Card_OpenCount ( DWORD inx,  
                                     DWORD *count );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*count*

Ukazatel na oblast paměti, do které je funkcí uložena hodnota proměnné udávající aktuální počet otevřených spojení s požadovanou kartou.

---

### Popis

Funkce je určena pro zjištění aktuálního počtu otevřených spojení s požadovanou kartou pomocí funkce *EP4GXA\_OpenMulti*, případně *EP4GXA\_Open* (počet otevřených spojení je vždy roven jedné).

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.12 EP4GXA\_Close

```
EP4GXA_RET EP4GXA_Close ( DWORD inx );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

---

### Popis

Funkce uzavře spojení s kartou se zadaným indexem *inx*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

*Poznámka: Při ukončení programu jsou automaticky uzavřena všechna otevřená spojení tohoto programu s kartou (v případě otevřených spojení s více kartami jsou uzavřena všechna spojení).*

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.13 EP4GXA\_BAR0\_BaseAdr

```
EP4GXA_RET EP4GXA_BAR0_BaseAdr ( DWORD inx,  
                                size_t *base_adr,  
                                DWORD *range );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *\*base\_adr*

Ukazatel na oblast paměti, do které je funkcí uložena počáteční adresa bloku paměti pro přístup k registrům karty.

### *\*range*

Ukazatel na oblast paměti, do které je funkcí uložena velikost bloku paměti pro přístup k registrům karty.

---

### Popis

Funkce je určena pro zjištění počáteční adresy a velikosti bloku paměti pro přístup k funkčním registrům karty.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

*Poznámka: Jednodušší a bezpečnější variantou řízení karty je použití funkcí EP4GXA\_ReadReg8, EP4GXA\_ReadReg32, EP4GXA\_WriteReg8 a EP4GXA\_WriteReg32.*

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.14 EP4GXA\_ReadReg8, EP4GXA\_ReadReg32

<b>EP4GXA_RET EP4GXA_ReadReg8</b>	<b>( DWORD <i>inx</i>, DWORD <i>offset</i>, BYTE <i>*data</i> );</b>
<b>EP4GXA_RET EP4GXA_ReadReg32</b>	<b>( DWORD <i>inx</i>, DWORD <i>offset</i>, DWORD <i>*data</i> );</b>

### ***inx***

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### ***offset***

Specifikuje adresu požadovaného registru karty v rámci bloku paměti s počáteční adresou BAR0 (tzn. offset proti adrese BAR0 uváděný v příručce). Zadaná hodnota musí být celočíselně dělitelná čtyřmi.

### ***\*data***

Ukazatel na oblast paměti, do které je funkcí uložena hodnota čteného registru.

### **Popis**

Funkce ve variantách pro 8bitová nebo 32bitová data slouží pro čtení obsahu registrů karty; nezbytná je v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic).

Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot počáteční adresy BAR0 (tuto hodnotu udržuje knihovna) a hodnoty *offset* zadané programem. Funkce proto umožňuje přistupovat k registrům výhradně z prostoru BAR0 a brání přístupu mimo platný rozsah adres.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### **Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

**Poznámka:** Doba vykonání obou funkcí činí cca 2,5  $\mu$ s, pro dosažení vyšší propustnosti při čtení dvou sousedních registrů lze použít funkce *EP4GXA\_ReadReg64* nebo *EP4GXA\_ReadReg2x32*.



## C2.15 EP4GXA\_WriteReg8, EP4GXA\_WriteReg32

<b>EP4GXA_RET EP4GXA_WriteReg8</b>	<b>( DWORD <i>inx</i>, DWORD <i>offset</i>, BYTE <i>data</i> );</b>
<b>EP4GXA_RET EP4GXA_WriteReg32</b>	<b>( DWORD <i>inx</i>, DWORD <i>offset</i>, DWORD <i>data</i> );</b>

### ***inx***

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### ***offset***

Specifikuje adresu požadovaného registru karty v rámci bloku paměti s počáteční adresou BAR0 (tzn. offset proti adrese BAR0 uváděný v příručce). Zadaná hodnota musí být celočíselně dělitelná čtyřmi.

### ***data***

Hodnota zapisovaná do registru.

## **Popis**

Funkce ve variantách pro 8bitová nebo 32bitová data slouží pro zápis do registrů karty; nezbytná je v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic).

Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot počáteční adresy BAR0 (tuto hodnotu udržuje knihovna) a hodnoty *offset* zadané programem. Funkce proto umožňuje přistupovat k registrům výhradně z prostoru BAR0 a brání přístupu mimo platný rozsah adres.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

## **Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

## **Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

**Poznámka:** Doba vykonání obou funkcí činí cca 0,2  $\mu$ s, pro dosažení vyšší propustnosti při zápisu do dvou sousedních registrů lze použít funkce *EP4GXA\_WriteReg64* nebo *EP4GXA\_WriteReg2x32*.

**C2.16 EP4GXA\_ReadReg64, EP4GXA\_ReadReg2x32 (podporováno od verze interface 3.18)**

```
EP4GXA_RET EP4GXA_ReadReg64      ( DWORD inx,  
                                   DWORD offset,  
                                   DWORD64 *data );
```

```
EP4GXA_RET EP4GXA_ReadReg2x32    ( DWORD inx,  
                                   DWORD offset,  
                                   DWORD *dataHi,  
                                   DWORD *dataLo );
```

***inx***

Specifikuje požadovanou kartu ze seznamu nalezených karet.

***offset***

Specifikuje adresu požadovaného registru karty v rámci bloku paměti s počáteční adresou BAR0 (tzn. offset proti adrese BAR0 uváděný v příručce). Zadaná hodnota musí být celočíselně dělitelná osmi.

***\*data***

Ukazatel na oblast paměti, do které je funkcí uložena 64bitová hodnota složená ze dvou 32bitových registrů (dolních 32 bitů přenáší obsah registru s adresou *offset*, horních 32 bitů obsah registru s adresou *offset+4*).

***\*dataLo***

Ukazatel na oblast paměti, do které je funkcí uložena 32bitová hodnota registru s adresou *offset*.

***\*dataHi***

Ukazatel na oblast paměti, do které je funkcí uložena 32bitová hodnota registru s adresou *offset+4*.

**Popis**

Funkce ve variantách pro jednu 64bitovou nebo dvě 32bitové hodnoty slouží pro čtení obsahu registrů karty; nezbytná je v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic). Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot počáteční adresy BAR0 (tuto hodnotu udržuje knihovna) a hodnoty *offset* zadané programem. Funkce proto umožňuje přistupovat k registrům výhradně z prostoru BAR0 a brání přístupu mimo platný rozsah adres.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

**Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

**Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

**Poznámka:** Doba vykonání obou funkcí je téměř identická jako doba vykonání funkce *EP4GXA\_ReadReg32* (cca 2,5 μs), datová propustnost je tedy dvojnásobná a to v 64bitově i 32bitově překládaných programech.

**C2.17 EP4GXA\_WriteReg64, EP4GXA\_WriteReg2x32 (podporováno od verze interface 3.18)**

<b>EP4GXA_RET EP4GXA_WriteReg64</b>	<b>( DWORD <i>inx</i>, DWORD <i>offset</i>, DWORD64 <i>data</i> );</b>
<b>EP4GXA_RET EP4GXA_WriteReg2x32</b>	<b>( DWORD <i>inx</i>, DWORD <i>offset</i>, DWORD <i>dataHi</i>, DWORD <i>dataLo</i> );</b>

***inx***

Specifikuje požadovanou kartu ze seznamu nalezených karet.

***offset***

Specifikuje adresu požadovaného registru karty v rámci bloku paměti s počáteční adresou BAR0 (tzn. offset proti adrese BAR0 uváděný v příručce). Zadaná hodnota musí být celočíselně dělitelná osmi.

***data***

Hodnota zapisovaná do dvou registrů karty (dolních 32 bitů přenáší obsah registru s adresou *offset*, horních 32 bitů obsah registru s adresou *offset+4*).

***dataLo***

Hodnota zapisovaná do registru karty s adresou *offset*.

***dataHi***

Hodnota zapisovaná do registru karty s adresou *offset+4*.

**Popis**

Funkce ve variantách pro jednu 64bitovou nebo dvě 32bitové hodnoty slouží pro zápis do registrů karty; nezbytná je v případě vývojových prostředí, která nepodporují přímý přístup k paměti (např. Visual Basic).

Skutečná adresa, na kterou funkce přistupuje, je dána součtem hodnot počáteční adresy BAR0 (tuto hodnotu udržuje knihovna) a hodnoty *offset* zadané programem. Funkce proto umožňuje přistupovat k registrům výhradně z prostoru BAR0 a brání přístupu mimo platný rozsah adres.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

**Podporované karty**

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

**Návratové kódy**

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

**Poznámka:** Doba vykonání obou funkcí je téměř identická jako doba vykonání funkce *EP4GXA\_WriteReg32* (cca 0,2  $\mu$ s), datová propustnost je tedy dvojnásobná a to v 64bitové i 32 překládaných programech.

## C2.18 EP4GXA\_IRQ\_Wait

```
EP4GXA_RET EP4GXA_IRQ_Wait      ( DWORD inx,  
                                   DWORD * );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

\*

Druhý parametr není od verze interface 1.55 využit (zůstal zachován z důvodu zpětné kompatibility s PCI kartami).

### Popis

Funkce pozastaví thread pro zpracování přerušení až do okamžiku, kdy ovladač zachytí přerušení systému vyvolané kartou nebo do vypršení mezní doby čekání na vyvolání přerušení (viz popis funkce *EP4GXA\_IRQ\_SetTimeout*).

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

Funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule) v případě, že proběhla bez chyb a bylo vyvoláno přerušení systému, v opačném případě pak předává některý z nenulových chybových návratových kódů.

Návratový kód *EP4GXA\_WaitTimeout* signalizuje, že funkce proběhla bez chyb, avšak nebylo vyvoláno přerušení systému a funkce skončila vypršením mezní doby čekání na vyvolání přerušení (timeoutem).

Pokud však funkce předá návratový kód *EP4GXA\_StopInThread*, je nutné ukončit provádění threadu, protože zřejmě došlo k uzavření spojení s kartou.

## C2.19 EP4GXA\_IRQ\_SetTimeout

```
EP4GXA_RET EP4GXA_IRQ_SetTimeout ( DWORD inx,  
                                   DWORD timeout );
```

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### *timeout*

Definuje mezní dobu čekání na přerušení, po kterou funkce *EP4GXA\_IRQ\_Wait* pozastaví thread. Hodnota je zadávána v milisekundách.

---

### Popis

Funkce nastaví mezní dobu čekání na vyvolání přerušení. Není-li přerušení vyvoláno před mezní dobu, je funkce *EP4GXA\_IRQ\_Wait* ukončena (tzn. pozastavený thread uvolněn) s návratovým kódem *EP4GXA\_WaitTimeout*.

Je-li hodnota *timeout* nastavena na nulu, funkce *EP4GXA\_IRQ\_Wait* detekuje, zda bylo vyvoláno přerušení od okamžiku předešlého provedení funkce *EP4GXA\_IRQ\_Wait* a bez dalšího čekání je běh této funkce ukončen s návratovým kódem *EP4GXA\_OK* nebo *EP4GXA\_WaitTimeout*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

V případě bezchybného provedení funkce předává návratový kód *EP4GXA\_OK* (hodnota rovna nule), v opačném případě pak předává některý z nenulových chybových návratových kódů.

## C2.20 EP4GXA\_IRQ\_Count

**DWORD EP4GXA\_IRQ\_Count (DWORD *inx*);**

### *inx*

Specifikuje požadovanou kartu ze seznamu nalezených karet.

### Popis

Funkce je určena pro zjištění počtu ovladačem zachycených přerušení od okamžiku otevření spojení s kartou některou z funkcí *EP4GXA\_Open* nebo funkcí *EP4GXA\_OpenMulti*.

Parametr *inx* musí odpovídat indexu karty, s níž již proces otevřel spojení.

Funkci *EP4GXA\_IRQ\_Count* je možné použít i v případě, kdy nebylo při otevření spojení funkcí *EP4GXA\_OpenMulti* vyvolání přerušení povoleno; může totiž nastat situace, kdy jiný program otevře (nebo již otevřel) spojení s kartou s povoleným přerušením.

### Podporované karty

Funkce je implementována pro všechny typy karet podporované knihovnou tedia\_ep4gxa.dll.

### Návratové kódy

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

V případě platného parametru *inx* je předáván počet ovladačem zachycených přerušení, v případě neplatného parametru *inx* je předávána hodnota 0.

## D1.1 Seznam standardních PCI karet obsluhovaných knihovnou tedia\_ox9162.dll

Karty obsluhované knihovnou tedia\_ox9162.dll jsou osazeny řadičem OX9162 (Oxford Semiconductor). Jelikož jsou kompatibilní s PCI sběrnici pouze s úrovní 5 V, jsou označeny jako PCI karty 1. generace.

Přehled karet je uveden v tabulce níže.

Standardní PCI karty obsluhované tedia_ox9162.dll		
typ karty	DID (hex) *	popis
PCD-7004	0101	PCI karta digitálních portů
PCD-7104	0102	PCI karta digitálních portů
PCT-7303A	0121	PCI karta IRC čítačů
PCT-7408A	0122	PCI karta čítačů/časovačů
PCT-7424	0123	PCI karta čítačů
PCA-7208AL	0141	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7208AS	0142	multifunkční PCI karta (AIN, AOOUT, DIO, CNT)
PCA-7408AL	0143	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7408AS	0144	multifunkční PCI karta (AIN, AOOUT, DIO, CNT)
PCA-7228AL	0145	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7228AS	0146	multifunkční PCI karta (AIN, AOOUT, DIO, CNT)
PCA-7428AL	0147	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7428AS	0148	multifunkční PCI karta (AIN, AOOUT, DIO, CNT)
PCA-7228EL	0149	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7428EL	0150	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7628AL	0151	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7628AS	0152	multifunkční PCI karta (AIN, AOOUT, DIO, CNT)
PCA-7288A	0161	PCI karta analogových výstupů
PCI-1052	0180	speciální komunikační PCI karta

**Poznámka:** Sloupec DID uvádí hodnotu, kterou vrací funkce *OX9162\_Card\_DID* (umožňuje jednoznačně identifikovat typ karty).

## D2.1 Seznam standardních PCI karet obsluhovaných knihovnou tedia\_ox952.dll

Karty obsluhované knihovnou tedia\_ox952.dll jsou osazeny řadičem OXmPCI952 nebo OXuPCI952 (PLX Technology, dříve Oxford Semiconductor). Jelikož jsou kompatibilní s PCI sběrnici s úrovněmi 5 V i 3,3 V, jsou označeny jako PCI karty 2. generace.

Přehled karet je uveden v tabulce níže.

Standardní PCI karty obsluhované tedia_ox952.dll s podporou obou PCI funkcí (funkce <b>OX952_BAR05_BaserAdr</b> poskytuje platnou adresu)		
typ karty	DID (hex) *	popis
PCA-7428CL	0240	multifunkční PCI karta (AIN, DIO, CNT)
PCA-7428CS	0242	multifunkční PCI karta (AIN, AOUT, DIO, CNT)
PCA-7428CE	0244	multifunkční PCI karta (AIN, DIO, CNT)
PCD-7006C	0302	PCI karta digitálních portů
PCD-7106C	0304	PCI karta digitálních portů
PCT-7303B	0200	PCI karta IRC čítačů
PCT-7303C	0210	PCI karta IRC čítačů
PCT-7303E	0212	PCI karta IRC čítačů
PCT-7424C	0214	PCI karta čítačů
PCT-7424E	0216	PCI karta čítačů
PCI karty obsluhované tedia_ox952.dll s podporou pouze PCI funkce 0 (PCI funkce 0 je obsluhována ovladačem COM portů, funkci <b>OX952_BAR05_BaserAdr</b> nelze použít)		
typ karty	DID (hex) *	popis
PCI-1054U	0401	speciální komunikační PCI karta

**Poznámka:** Sloupec DID uvádí hodnotu, kterou vrací funkce **OX952\_Card\_DID** (umožňuje jednoznačně identifikovat typ karty).



### D3.1 Seznam standardních PCI Express karet obsluhovaných knihovnou tedia\_ep4gxa.dll

Karty obsluhované knihovnou tedia\_ep4gxa.dll jsou osazeny hradlovým polem řady EP4GX (Intel, dříve Altera; integruje řadič PCI Express a většinu I/O funkcí). Karty jsou kompatibilní se sběrnici PCI Express (x1, Gen 1.0a/1.1).

Přehled karet je uveden v tabulce níže.

Standardní PCI Express karty obsluhované tedia_ep4gxa.dll		
typ karty	DID (hex) *	popis
PCD-8006	0800	PCI karta digitálních portů
PCD-8104	0804	PCI karta digitálních portů
PCD-8105	0805	PCI karta digitálních portů
PCD-8106	0806	PCI karta digitálních portů
PCT-8303	0810	PCI karta IRC čítačů
PCT-8306	0811	PCI karta IRC čítačů
PCT-8363	0812	PCI karta IRC čítačů a SSI rozhraní
PCT-8360	0820	PCI karta SSI rozhraní
PCT-8424	0830	PCI karta čítačů/časovačů
PCT-8425	0831	PCI karta čítačů/časovačů
PCT-8426	0832	PCI karta čítačů/časovačů
PCA-8428	0840	multifunkční PCI karta (AIN, AOUT, DIO, CNT)
PCA-8429	0841	multifunkční PCI karta (AIN, DIO, CNT)
PCA-8438	0842	multifunkční PCI karta (AIN, AOUT, DIO, CNT)
PCA-8439	0843	multifunkční PCI karta (AIN, DIO, CNT)
PCA-8628	0850	multifunkční PCI karta (AIN, AOUT, DIO, CNT)
PCA-8629	0851	multifunkční PCI karta (AIN, DIO, CNT)
PCA-8638	0852	multifunkční PCI karta (AIN, AOUT, DIO, CNT)
PCA-8639	0853	multifunkční PCI karta (AIN, DIO, CNT)
PCA-8288	0860	multifunkční PCI karta (AOUT, DIO)
PCA-8688	0861	multifunkční PCI karta (AOUT, DIO)
PC8K-LOADER	08FF	jakákoliv karta ze seznamu výše s naprogramovaným servisním firmware (umožňuje programování platného firmware karty)

**Poznámka:** Sloupec DID uvádí hodnotu, kterou vrací funkce *EP4GXA\_Card\_DID* (umožňuje jednoznačně identifikovat typ karty).

## D4.1 Verze ovladače a podporované operační systémy Windows

### Historie verzí ovladače od svého vzniku:

- 1.x uvolněn v roce 2001  
ovladač vystavěný na vývojovém kitu Jungo Windriver verze 5.x  
podpora PCI karet a 32bitových systémů Windows 98/Me, Windows 2000 a Windows XP
- 2.x uvolněn v roce 2007  
ovladač vystavěný na vývojovém kitu Jungo Windriver verze 9.x  
podpora PCI karet a 32bitových systémů Windows 2000, Windows XP, Windows Vista a Windows 7
- 3.0x uvolněn v roce 2010  
ovladač vytvořen jako vlastní kernel software  
podpora 32bitových i 64bitových operačních systémů  
podpora Windows 2000, Windows XP, Windows Vista a Windows 7  
podpora Windows Server 2003, Windows Server 2008 a Windows Server 2008-R2  
později doplněna podpora Windows 8/8.1 a Windows Server 2012/2012-R2
- 3.1x uvolněn v roce 2015  
doplněna podpora PCIe karet  
doplněna podpora Windows 10 (Windows Server 2016 a vyšší nejsou podporovány)

### Historie verzí ovladače 3.1x:

- 3.10 dostupný od října 2015  
betaverze s podporou PCIe karet, obsahuje SHA-1/SHA-2 certifikáty
- 3.12 dostupný od března 2016  
první uvolněná verze s podporou PCIe karet, obsahuje SHA-1/SHA12 certifikáty
- 3.15 dostupný od září 2020  
vyřešena podpora přechodů mezi úspornými režimy včetně "rychlého spuštění"  
doplněna podpora MSI režimu přerušení (pouze PCIe karty)  
obsahuje již pouze SHA-2 certifikáty (nelze použít ve Windows Vista a Windows Server 2008)
- 3.18 dostupný od března 2021  
pro PCIe karty doplněny funkce...
  - pro zjištění hodnoty CardID (poloha DIP spínače) a výrobního čísla bez nutnosti otevření spojení
  - funkce pro 64bitové přístupy k registrům ve 64bitové i 32bitové překládaných programech (umožňují cca 2x vyšší datový tok než 32bitové přístupy)





**Výroba, prodej, servis a technická podpora:**

adresa: TEDIA® spol. s r. o.  
Zábělská 12  
31211 Plzeň  
Česká republika

internet: <http://www.tedia.cz>  
<http://www.tedia.eu>

telefon: aktuální informace najdete na adrese  
<http://www.tedia.cz/kontakty>

e-mail: aktuální informace najdete na adresách  
<http://www.tedia.cz/kontakty>  
<http://www.tedia.cz/podpora>