AD14PCI

Měřicí karta - 14 bit AD/DA převodník + DIO
s DSP

Návod k obsluze

Obsah

1. Úvod **3**
2. Technické parametry **3**
3. Popis činnosti **5**
3.1. Popis měřicí sekvence *8*
4. Instalace karty **10**
5. Popis driveru AD14PCI.DLL **14**

1. Úvod

Karta AD14PCI je karta špičkové úrovně určená pro rychlá a přesná měření v laboratoři a průmyslu. Karta je vybavena signálovým procesorem ADSP2186 fy Analog Devices s výpočetním výkonem 40 MIPS, který zajišťuje řízení všech funkcí karty a zpracování naměřených dat. A/D převodník s rozlišením 14 bitů je galvanicky oddělen od obvodů PC, což výrazně snižuje problémy s rušením a výrazně rozšiřuje možnosti karty.

Karta dále obsahuje 2 D/A kanály s rozlišením 12 bitů a 16 číslicových vstupů a 16 číslicových výstupů.

2. Technické parametry

A/D převodník:

rozlišení 14 bitů, odstup s/š 74 dB, vestavěný Track/Hold

vzorkovací rychlost: 250 kHz

ef. hodnota šumu: typ. 0.6 LSB pro zesílení 1 až 4, 1.1 LSB pro zesílení 8

integrální nelinearita: typ +- 2 LSB

diferenciální nelinearita: typ +-1.5 LSB

ofset : max. 0.5 mV + 2 LSB

chyba rozsahu : max. 3 LSB

vstupní rozsah: +-5 V / 0 - 10 V, zesílení volitelné 1,2,4,8 x

počet vstupů : 16 (rozšiřitelné na 32)

spouštění : programově, TTL signálem nebo úrovní na vstupu

časovač : rozsah 16 bitů, krok časovače 0.1 μs

obvody přerušení: zdroj přerušení - zvolený počet dat,

Paměť: FIFO max. 8 Kslov

D/A převodník:

rozlišení : 12 bitů, 2 kanály

rozsah výst. napětí : 0 až 5 V nebo +-5 V volitelné propojkou

integrální nelinearita : max +-1 LSB

diferenciální nelinearita : max +-1 LSB

ofset : max +-2 LSB

chyba rozsahu : max +-3 LSB

doba ustálení: typ. 10 μs

Digitální porty:

počet vstupů: 16 (TTL/HCT kompatibilní)

počet výstupů: 16 (TTL/HCT kompatibilní)

Čítač:

16 bitů, režimy: čítání, měření periody s prog. rozlišením 0.1 až 6500 μ s , měření frekvence s hradlovací dobou 0.1 ms až 6.5 s, potlačení zákmitů, max. vst. frekvence 2.5 MHz

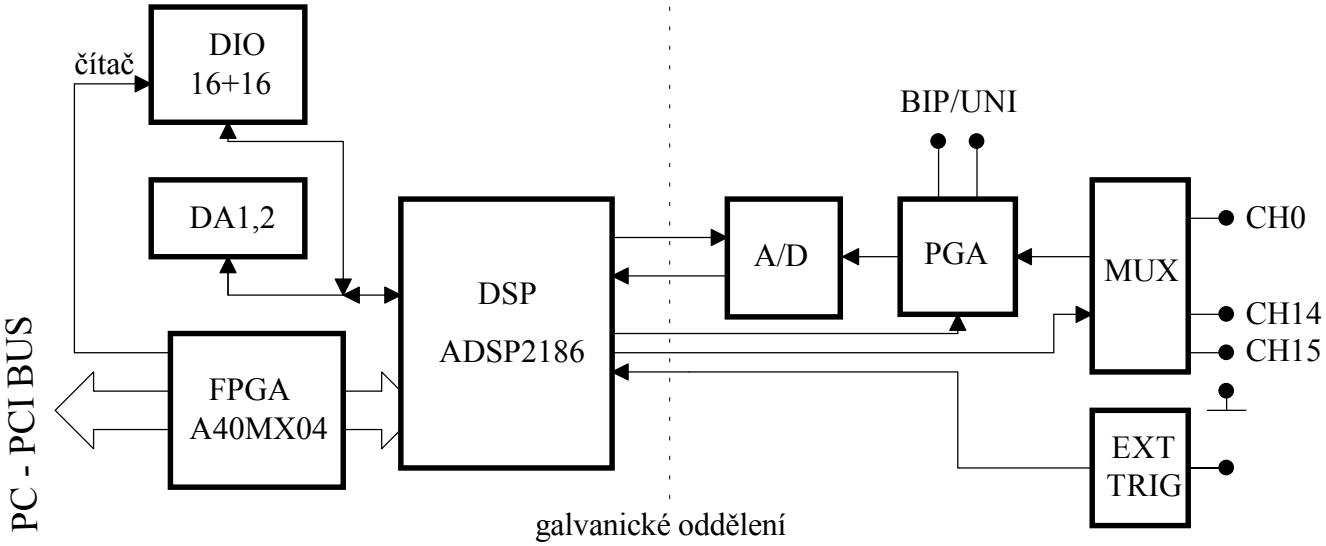
časová základna (pro čítač i ad část) : 10.002 MHz +-0.01%

Celkový odběr : z 5 V max. 200 mA, z 12 V max. 200 mA

3. Popis činnosti

Na obr.1 je blokové schéma karty AD14PCI. Ze schématu je patrné, že hlavní součástí karty je signálový procesor ADSP2186 (DSP). Tento signálový procesor řídí všechny ostatní části karty a komunikuje přes PCI čip typu PCI2040 s PC.

Zbývající část logiky karty je realizována hradlovým polem Actel A40MX04, ve kterém je rovněž implementován i uživateli přístupný 16-ti bitový čítač pracující ve třech režimech: 1. prosté čítání, 2. měření frekvence s programovatelnou hradlovací dobou v rozsahu 0.1 ms až 6 s, 3. měření periody s programovatelným rozlišením 100 ns až 6.5 ms. Na vstupu čítače je obvod pro potlačení zákmitů, který omezuje max. vstupní frekvenci na 2.5 MHz.



Obr. 1.

A/D část karty je galvanicky oddělena od DSP i od počítače, komunikace mezi A/D převodníkem a DSP a rovněž ovládání vst. zesilovače (PGA) a multiplexeru probíhá přes optrony. Propojkou lze volit unipolární/bipolární rozsah karty a propojení kanálu 15 se zemí pro účely autokalibrace ofsetu. Pro nastavení bipolárního a unipolárního ofsetu karty a zesílení 1 jsou na kartě víceotáčkové trimry.

Nastavení ofsetu karty je pro všechny rozsahy karty společné, vzhledem k tomu, že ofset se zesílením poněkud mění, je možno nastavit přesně ofset

nejpoužívanějšího rozsahu nebo využít možnosti programové korekce ofsetu pomocí autokalibrace.

Trimrem lze rovněž dostavit rozsah převodníku při zesílení 1, ostatní zesílení jsou dána pevně vysoce stabilním a přesným destičkovým odporem.

Ve vstupní části je rovněž expanzní konektor umožňující připojení dalšího multiplexeru, který rozšiřuje počet vstupů až na 32. Tyto vstupy jsou plně podporovány logikou karty.

V plovoucí části je rovněž vstup EXTTRIG pro externí spouštění převodu. Spouštění je realizováno náběžnou nebo sestupnou hranou TTL/HCT kompatibilního signálu.

Obslužný program je do DSP nahráván při inicializaci karty, což zajišťuje vysokou flexibilitu karty, snadnou možnost upgrade a zákaznických úprav. Standardně dodávaný firmware implementuje všechny běžné funkce měřicí karty, tj . je možno zvolit které kanály s jakým zesílením je potřeba měřit, dále je možno navolit časové odstupy mezi kanály v rozsahu 0.004 až 6.553 ms pro každý kanál zvlášť. Naměřená data jsou ukládána do FIFO paměti o velikosti 8K vzorků s programovatelným příznakem $data_ready$ v rozsahu 1 až 8000. Po dosažení tohoto počtu hodnot je nastaven příslušný flag, popřípadě generováno programově nastavitelné přerušení. FIFO paměť je rovněž realizována programově.

Použití DSP však umožnilo i implementaci několika dalších velmi užitečných funkcí. Jsou to číslicová filtrace a spouštění s funkcí pretrigger.

Číslicová filtrace je realizována průměrováním, u kterého je možno zvolit počet vzorků, z nichž je průměr počítán a volbou dělicího koeficientu je možno zvýšit rozlišení karty až na 16 bitů. Průměrování je příkladem nejjednoduššího FIR filtru typu dolní propust s následnou decimací (snížením frekvence). Funkci číslicové filtrace je vhodné zvolit vždy, kdy není nutno měřit plnou rychlostí. Např. vzorkovací frekvence 1 kHz je možno dosáhnout buď nastavením časovače na 1 ms nebo nastavením časovače na 10 μ s a průměrem ze sta vzorků.

Druhé řešení je v naprosté většině případů mnohem vhodnější, protože filtrací se podstatně sníží šířka zpracovaného pásma a výrazně se sníží vliv šumu a rušení mimo užitečné pásmo frekvencí. Přenosová funkce tohoto typu filtru vyjádřená v z-transformaci je:

$$H(z) = \frac{1-z^{-N}}{1-z^{-1}}$$

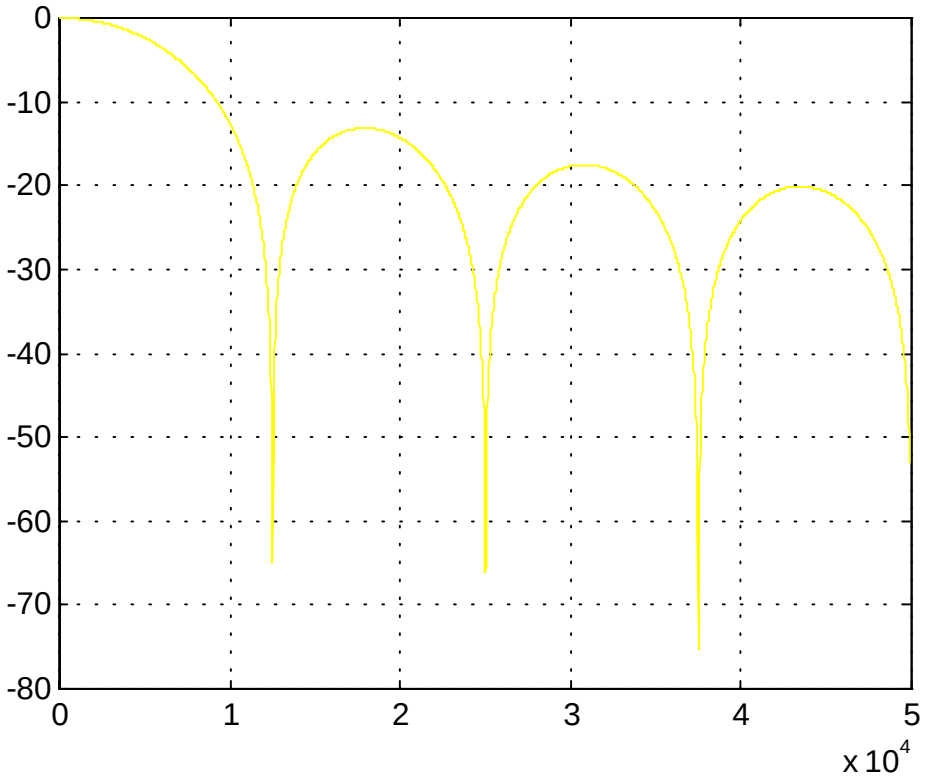
kde N je počet vzorků, z nichž je počítán průměr.

Ve frekvenční oblasti je možno přenosovou funkci vyjádřit vztahem:

$$H(f) = \frac{1}{N}\frac{\sin(N\pi fT)}{\sin(\pi fT)}$$

kde N je opět počet vzorků

T = 1/fs, fs je původní vzorkovací frekvence.



Obr. 2

Na obr. 2. je znázorněna tato funkce pro fs = 200 kHz a N = 16. Tato funkce je totožná s přenosem integračního ad převodníku s dobou integrace N x T a je velmi podobná přenosu většiny sigma/delta ad převodníků s odpovídajícím nastavením filtru.

Z uvedeného vztahu i z obr. 2 je patrná další důležitá vlastnost tohoto typu filtru a to, že frekvence fs/N a všechny její násobky jsou teoreticky nekonečně potlačeny. Zvolíme-li tedy fs a N tak, aby fs/N = 50 Hz, bude potlačen kmitočet 50 Hz a všechny jeho vyšší harmonické, což je v praxi velmi užitečné vzhledem k všudypřítomnosti rušení z rozvodné sítě při reálných měřeních.

Další užitečnou funkcí, jejíž realizaci umožnil DSP, je spouštění měření. Odměr lze spustit buď náběžnou nebo setupnou hranou na TTL vstupu, nebo náběžnou nebo sestupnou hranou a libovolně zvolenou úrovní na prvním měřeném kanálu. Navíc lze zvolit předstih (pretrigger) ukládaných dat až do maximální velikosti paměti tj. 8000 vzorků.

D/A převodníky a číslicové vstupy a výstupy jsou rovněž přímo připojeny k DSP, což umožňuje naprosto nezávislou činnost na PC např. realizovat arbitrary generátor s laděním přímou digitální syntézou (DDS) s krokem pod 0.01 Hz, generovat složité sekvence pulsů na číslicových portech, realizovat velmi rychlé regulační smyčky apod. Tyto programy nejsou standardní součástí dodávky ale vzhledem k tomu, že program pro DSP je nahráván z PC, je možno tyto úpravy realizovat bez zásahu do karty pouhou změnou software. Číslicové vstupy/výstupy je rovněž možno programovat přímo z PC.

Výstup D/A převodníků je jednak vyveden na vstupní CANNON-25 konektor, jednak na zvláštní konektor na desce spolu s napájením pro připojení eventuelního rekonstrukčního filtru. Číslicové vstupy a výstupy jsou vyvedeny na samostatné konektory DIL-20 kompatibilní s kartami Advantech PCL-812/818/1800. Vstupní a výstupní budiče těchto portů jsou umístěny v patici.

3.1. Popis měřicí sekvence

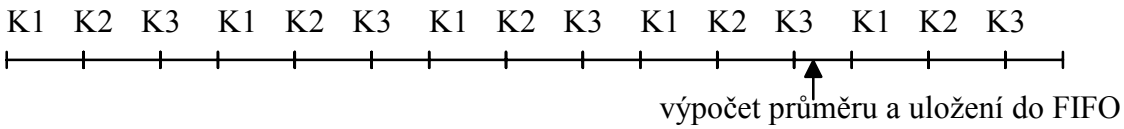
Z výše uvedeného popisu karty je zřejmé, že programování a průběh měření je poněkud odlišný od běžných karet. U karty se nezadává celková perioda měření ale vždy časový odstup měřeného kanálu od předcházejícího. Tento časový odstup je nutný pro realizaci převodu předcházejího kanálu a zároveň pro ustálení zesilovače po přepnutí na nový kanál. (Současně se startem převodu je přepnut vstupní MUX a zesílení zesilovače na novou hodnotu).

Je-li měřen pouze jeden kanál, k žádnému přepínání nedochází, a zvolené zpoždění udává zároveň periodu měření. Ta může být volena již od 4 μs (fs=250kHz).

Je-li měřeno více kanálů, musí být zpoždění mezi jednotlivými kanály alespoň 5 μs pro ustálení vst. zesilovače (10 μs pro expander Pt100, 25 μs pro expander pro termočlánky). Celková perioda měření je tedy dána součtem jednotlivých zpoždění měřených kanálů. To znamená, že chceme-li měřit např. 3 kanály s celkovou periodou 25 μs, můžeme zvolit jednotlivá zpoždění takto: 8 μs, 8 μs, 9 μs. Tato volba je optimální pro co nejdelší časy pro ustálení vst. zesilovače před každým odměrem. Je-li naopak potřeba aby kanály byly odměřeny v nejkratší možné době za sebou, je vhodnější tato volba: 15 μs, 5 μs, 5 μs.

Pokud je aktivována filtrace v DSP s počtem průměrů *n*, je výše uvedená sekvence *n* krát zopakována, poté je pro každý kanál je vypočten průměr, který je uložen jako výsledek měření do FIFO paměti. Výsledná perioda měření je dána jako součet zpoždění x *n*. Na obr. 2a je znázorněna sekvence měření pro 3 kanály a průměr ze 4 vzorků.

Díky tomuto způsobu měření i při měření více kanálů platí vztahy pro přenos filtru uvedené výše a karta má obdobné vlastnosti jako *k* integračních a převodníků pracujících paralelně (*k* - počet aktivních kanálů, max 32).



Obr. 2a

Pro názornost příklad: chceme měřit 4 kanály s celkovou periodou 20 ms. (Na všech kanálech bude potlačeno rušení 50 Hz a jeho násobky). Vzhledem k vysokému počtu průměrů má smysl zvýšit rozlišení karty na 16 bitů a proto pro dokonalé ustálení je vhodné zvolit zpoždění mezi kanály více než 5 μs Zvolíme základní měřicí periodu 25 μs, počet filtrací bude 20000/25 = 800 a jednotlivá zpoždění např. 6 + 6 + 6 + 7 = 25 μs.

4. Instalace karty

Před instalací karty je nutno zvolit propojkou bipolární/unipolární rozsah karty. Karta je ve výrobě nastavena na bipolární rozsah. Na obr. 3 jsou znázorněny příslušné propojky. Kartu je možno zasunout do počítače pouze při vypnutém napájení! Vzhledem k citlivosti a rychlosti karty je vhodné ji umístit tak, aby byla ostatními deskami co nejméně rušena. Všechny konektory jsou zapojeny totožně jako u předcházejícího typu AD14DSP. Instalace software viz soubor **readme**.

Zapojení konektorů:

K1 Cannon-25 zásuvka

plovoucí analog. zem **13**

25 kanál 0

kanál 1 **12**

24 kanál 2

kanál 3 **11**

23 kanál 4

kanál 5 **10**

22 kanál 6

kanál 7 **9**

21 kanál 8

kanál 9 **8**

20 kanál 10

kanál 11 **7**

19 kanál 12

kanál 13 **6**

18 kanál 14

kanál 15 **5**

17 ext. spouštění

log. zem **4**

16 výstup 1. D/A převodníku

2. D/A **3**

15 zem pro D/A převodníky

+15 V izolovaných **2**

14 -15 V izolovaných

zem pro napájení **1**

K2 - DIL20 číslicové výstupy:

O0 **1**	**2** O1
O2 **3**	**4** O3
O4 **5**	**6** O5
O6 **7**	**8** O7
O8 **9**	**10** O9
O10 **11**	**12** O11
O12 **13**	**14** O13
O14 **15**	**16** O15
GND **17**	**18** GND
+5V **19**	**20** +12V

K3 - DIL20 číslicové vstupy:

IN0 **1**	**2** IN1
IN2 **3**	**4** IN3
IN4 **5**	**6** IN5
IN6 **7**	**8** IN7
IN8 **9**	**10** IN9
IN10 **11**	**12** IN11
IN12 **13**	**14** IN13

IN14 **15** **1** **6** IN15 (Na vstup IN15 je přes propojku přiveden i vstup čítače/měřiče kmitočtu)

GND **17**	**18** GND
+5V **19**	**20** +12V

K4 - DIL20 expanzní konektor

K5 - SIL6 výstup DA převodníků

DA2 1

GND 2

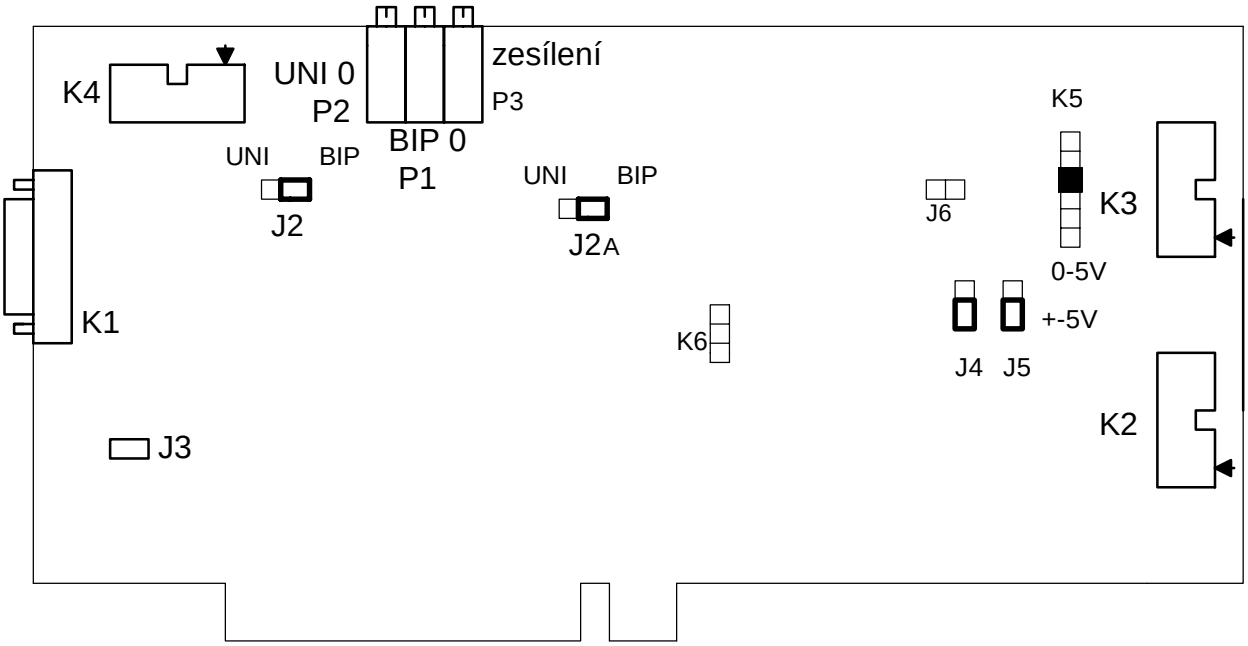
nic 3

DA1 4

12 V 5

-12 V 6

K6 - SIL3 - synchronizace



Obr. 3

Nastavovací prvky:

J2 - unipolární/bipolární rozsah A/D převodníku

J3 - propojí kanál 15 se zemí pro kalibraci ofsetu

J4 - nastavení rozsahu 1. DA převodníku

J5 - nastavení rozsahu 2. DA převodníku

J6 - propojí vstup čítače s IN15 na konektoru K3-1

P2 - nastavení unipolární nuly A/D převodníku

P1 - nastavení bipolární nuly A/D převodníku

P3 - nastavení rozsahu A/D převodníku při zesílení 1

5. Popis driveru AD14PCI.DLL

Pro snadnou tvorbu vlastních programů s AD14PCI pod Windows je spolu s kartou dodáván i driver **AD14PCI.DLL** pro ovládání v libovolném ja umožňujícím volání funkcí z DLL knihoven. Postup instalace a příklady použití viz dodaná disketa.

K dispozici jsou následující procedury a funkce:

function InitializeDevice(CardCount:integer):integer; stdcall;

procedure FinalizeDevice; stdcall;

function Boot(card: word;bootname:pchar): byte;stdcall;

procedure SetFIFOIndex(card:integer;index : word); stdcall;

procedure SetTrigMode(card:integer;mode: byte;level,pretrig:integer);stdcall;

function SetIRQ():WordBool;stdcall;

procedure ProgEn(card:integer;chan:byte);stdcall;

procedure ProgCh(card:integer;channel,gain : byte;delay:word);stdcall;

procedure Filter(card:integer;filt,divid:integer);stdcall;

function GetStatus(card:integer) : byte;stdcall;

function GetSample(card:integer) : integer;stdcall;

function GetIWritePointer(card:integer) : integer;stdcall;

function GetEWritePointer(card:integer) : integer;stdcall;

procedure ReadIBuffer(buff:pointer;MaxInd:integer;kruh:WordBool);stdcall;

procedure AutoCal(card:integer;res:byte;ofset:pointer);stdcall;

procedure StartAD(card:integer);stdcall;

procedure StopAD(card:integer);stdcall;

procedure ActFlags(card:integer);stdcall;

procedure TimeBase(card:integer;time:word);stdcall;

procedure StartCounter(card:integer);stdcall;

procedure StopCounter(card:integer);stdcall;

procedure ClearCounter(card:integer);stdcall;

procedure StartFMeas(card:integer);stdcall;

procedure StartPMeas(card:integer);stdcall;

function FPMeasDone(card:integer):WordBool;stdcall;

function GetCounter(card:integer):word;stdcall;

Syntaxe a použití těchto procedur a funkcí je následující:

function InitializeDevice(CardCount:integer):integer; stdcall;

Inicializuje driver , musí být volána jako první.

parametry: *CardCount* - počet použitých karet.

Pokud funkce proběhla úspěšně vrací CardCount, jinak číslo menší než CardCount. U všech ostatních procedur a funkcí udává parametr *card* číslo karty ke které se daný povel vztahuje, první karta má číslo 0.

procedure FinalizeDevice; stdcall;

Uvolňuje driver z paměti.

function Boot(card: word;bootname:pchar): byte;stdcall;

Tato funkce nahraje obslužný program se jménem *bootname* do DSP.Pokud proběhne bootování úspěšně, vrací funkce hodnotu *0* , pokud není nalezen soubor *bootname*, vrací hodnotu *1* , pokud není přítomna, vrací hodnotu Tato funkce musí být volána před jakoukoliv další funkcí po zavolání InitializeDevice.

procedure SetFIFOIndex(card:integer;index : word); stdcall;

tato procedura nastaví FIFO index přímo na požadovaný počet vzorků daný parametrem *index*. Pro jeho velikost platí:

index = počet vzorků na kanál x počet aktivních kanálů

max. velikost je 8000 vzorků.

procedure SetTrigMode(card:integer;mode: byte;level,pretrig:integer);stdcall;

Tato procedura nastaví režim spouštění podle hodnoty *mode* takto:

0 - ihned po povelu start

1 - vzestupnou hranou TTL + pretrigger

2 - sestupnou hranou TTL + pretrigger

3 - vzestupnou hranou na 1. kanálu + úroveň + pretrigger

4 - sestupnou hranou na 1. kanálu + úroveň + pretrigger

level - úroveň spouštění pro typy 3 a 4 - rozsah -8192 až +8191 (-32768 až 32767 pro 16-ti bitové rozlišení

pretrig - předstih spouštění v rozsahu 0 až 8000 vzorků pro typy 1 až 4

function SetIRQ():WordBool;stdcall;

aktivuje přerušovací systém, zavádí obslužný podprogram volaný z kernel módu procesoru, okud je zavedení úspešné vrací true, jinak false.

procedure ProgEn(card:integer;chan:byte);stdcall;

Tato procedura slouží k zahájení programování požadovaných kanálů. Po zavolání této procedury musí následovat příslušný počet volání procedury *ProgCh*.

chan - počet kanálů (max. 32)

procedure ProgCh(card:integer;channel,gain : byte;delay:word);stdcall;

Tato procedura nastaví požadovaný kanál, jeho zesílení a zpoždění vůči předcházejícímu kanálu.

channel - adresa kanálu v rozsahu 0 až 31

gain - zesílení 0 - 1 x, 1 - 2 x, 3 - 4 x, 3 - 8 x

delay - zpoždění v rozsahu 40 až 65535 (v desetinách μs)

procedure Filter(card:integer;filt,divid:integer);stdcall;

Tato procedura slouží k nastavení paremetrů filtrace. Parametr *filt* určuje, z kolika vzorků bude počítán průměr, parametr *divid* určuje, jakým koeficentem bude výsledný součet vydělen. Je-li *divid* roven *filt*, je výsledné rozlišení bitů, je-li dvakrát nebo čtyřikrát menší, je výsledné rozlišení 15 respektive 16

bitů. Rozsah použitelných hodnot pro filt je 1 až 16383, pro divid je 2 až 16383. Je-li *filt* = 1, parametr *divid* je ignorován.

function GetStatus(card:integer) : byte;stdcall;

Tato fukce vrací status FIFO paměti a může nabývat těchto hodnot:

0 - empty - FIFO je prázdné

1 - ve FIFO paměti je alespoň jeden vzorek

2 - v paměti je alespoň *data_ready* hodnot

3 - FIFO je zaplněno, tj. v paměti je 8000 hodnot

Při testování statusu je třeba mít na zřeteli, že k automatické aktualizaci dochází pouze zápisem, aktualizaci po čtení je nutno zajistit voláním procedury *ActFlags*. Tuto funkci má smysl volat pouze není-li aktivován přerušovací systém.

function GetSample(card:integer) : integer;stdcall;

Tato funkce vrací naměřenou hodnotu. Rozlišení závisí na nastavení procedurou *Filter* a může být 14 až 16 bitů. Tuto funkci má smysl volat pouze není-li aktivován přerušovací systém.

function GetIWritePointer(card:integer) : integer;stdcall;

vrací hodnotu zápisového pointeru v interním kruhovém bufferu 250 000 vzorků, ukazuje vždy na 1. nazapsanou pozici, tj. poslední naměřená hodnota je na pozici o 1 nižší. Použití této a následujících fukcí viz demo na disketě.

function GetEWritePointer(card:integer) : integer;stdcall;

vrací hodnotu zápisového pointeru v uživatelském bufferu (tento buffer je deklarován v uživatelském programu, doporučená velikost je alespoň na cca 1 sec zaznamu pro kontinualní čtení dat bez výpadků), ukazuje vždy na 1. nezapsanou pozici, tj. poslední naměřená hodnota je na pozici o 1 nižší.

procedure ReadIBuffer(buff:pointer;MaxInd:integer;kruh:WordBool);stdcall;

tato funkce kopíruje data z interního do uživatelského bufferu. Vždy jsou prečtena všechna dosud nenačtená data. MaxInd udává maximální přípustný index uživatelského bufferu, proměnná *kruh* určuje, zda je uživatelský buffer kruhový. Pokud má hodnotu True, pak při překročení *MaxInd* se data zapisují znovu od 0, pokud má hodnotu False, pak se zápis dat zastaví na hodnotě *MaxInd.*

procedure AutoCal(card:integer;res:byte;Uo:pointer);stdcall;

Tato procedura zajistí změření vlastního ofsetu karty pro jednotlivá zesílení a jejich uložení do pole *Uo* deklarovaného v uživatelském programu takto: *Uo : array[0..3] of integer*. Pro využití této funkce je nutné, aby byla zapojena propojka zkratující kanál 15 s plovoucí zemí! Parametr *res* určuje rozlišení, je-li roven 0, je rozlišení 14-ti bitové, jinak je 16-ti bitové.

procedure StartAD(card:integer);stdcall;

Tato procedura odstartuje měření, popřípadě povolí externí spouštění.

procedure StopAD(card:integer);stdcall;

Tato procedura zastaví měření.

procedure ActFlags(card:integer);stdcall;

Tato procedura zajišťuje aktualizaci flagů indikujících stav FIFO paměti. Je nutno ji volat po skončení čtení dat před opětovným testováním statusu. Tuto funkci má smysl volat pouze není-li aktivován přerušovací systém.

procedure TimeBase(card:integer;time:word);stdcall;

tato procedura zajištuje generování hodinového signálu pro čítač pro měření periody/frekvence. Při měření frekvence udává parametr *time* hradlovací dobu s krokem 0.1 ms, při měření periody rozlišení v čase s krokem 100 ns.

procedure StartCounter(card:integer);stdcall;

tato pocedura spustí prosté čítání čítače.

procedure StopCounter(card:integer);stdcall;

tato procedura zastaví čítač.

procedure ClearCounter(card:integer);stdcall;

tato procedura vynuluje čítač.

procedure StartFMeas(card:integer);stdcall;

tato procedura spustí měření frekvence s hradlovací dobou nastavenou procedurou *TimeBase*.

procedure StartPMeas(card:integer);stdcall;

tato procedura spustí měření periody s rozlišením nastaveným procedurou *TimeBase*.

function FPMeasDone(card:integer):WordBool;stdcall;

tato funkce vrací *true* jestliže je dokončeno měření periody nebo frekvence.

function GetCounter(card:integer):word;stdcall;

tato funkce vrací obsah čítače.