

Obsah

1. Úvod.....	2
2. Technické parametry.....	2
3. Popis činnosti.....	3
3.1. Popis měřicí sekvence.....	5
4. Instalace modulu.....	7
5. Popis driveru AD14ETH.DLL.....	9

1. Úvod

Modul AD14ETH je měřicí modul komunikující přes Ethernet 10/100 Mbit určený pro rychlá a přesná měření v laboratoři a průmyslu. Ethernet rozhraní zajišťuje galvanické oddělení pro stejnosměrné signály což výrazně snižuje problémy s rušením a rozšiřuje možnosti modulu.

Modul dále obsahuje 7 číslicových vstupů a 7 číslicových výstupů.

2. Technické parametry

A/D převodník:

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

vzorkovací rychlost: 400 kHz

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

integrální nelinearita: typ +/- 1 LSB

diferenciální nelinearita: typ +/-0.5 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)

časovač : rozsah 16 bitů, krok časovače 0.1 μ s

Ethernet:

rychlost: 100/10 Mbit

zapojení: Automatické rozpoznání MDI/MDI-X

modul v síti pracuje jako server s nastavitelnou IP adresou, komunikuje protokolem TCP/IP

Digitální porty:

počet vstupů: 7 (TTL/HCT kompatibilní)

počet výstupů: 7 úroveň log 1 - 3.3 V

Napájení:

střídavé 12 V/50 Hz z dodávaného adaptéru.

3. Popis činnosti

Jádrem modulu AD14ETH je výkonný 32 bitový procesor Cortex M3, který realizuje veškeré potřebné funkce modulu. Vnitřní paměť RAM slouží zároveň jako vyrovnávací buffer s kapacitou 25000 vzorků. Modul komunikuje přes rozhraní Ethernet 10/100 Mbit/s protokolem TCP/IP, pro dosažení max. vzorkovacích rychlostí je nutná přenosová rychlost 100 Mbit/s. Nejvyšší vzorkovací rychlost je garantována pouze pro počet vzorků, který se vejde do vnitřní paměti, reálná dosažitelná vzorkovací rychlost pro více vzorků závisí i na zatížení sítě a výkonu PC vzhledem ke značnému datovému toku až 800 kByte/s. V praxi je však zpravidla možné v běžné síti dosahovat stabilně i nejvyšší vzorkovací rychlost.

Analogová část modulu se skládá z 14-ti bitového AD převodníku a zesilovače s programovatelným ziskem. Na vstupu zesilovače je 16-ti kanálový multiplexer.

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

Použití procesoru 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:

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

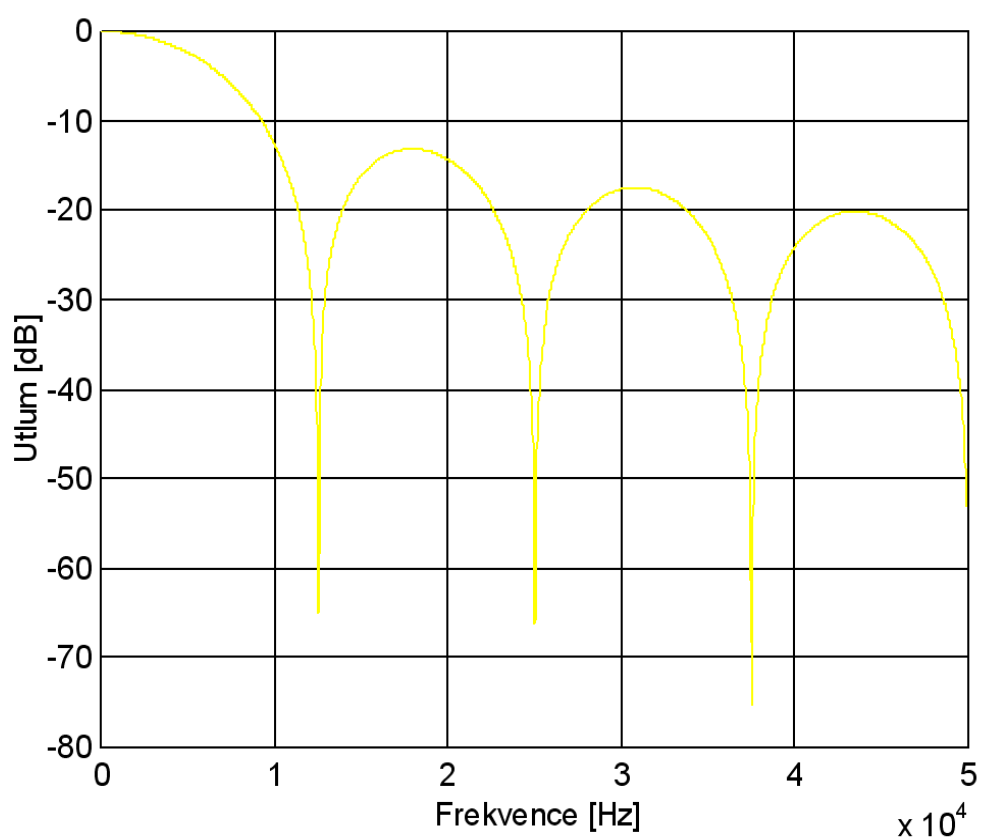
kde N je počet vzorků, z nichž je počítán průměr.

Ve frekvenční oblasti je možno přenosovou funkci vyjádřit vztahem:

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

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 je spouštění měření. Odměr lze spustit 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. 25000 vzorků.

3.1. Popis měřicí sekvence

Z výše uvedeného popisu je zřejmé, že programování a průběh měření je poněkud odlišný od běžných měřicích karet. U modulu se nezadává celková perioda měření ale vždy časový odstup měřeného kanálu od předcházejícího. Tento časový odstup je nutný pro realizaci převodu předcházejícího 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 2.5 μ s ($f_s=400\text{kHz}$).

Je-li měřeno více kanálů, musí být zpoždění mezi jednotlivými kanály alespoň 2.5 μ s (3 μ s při zesílení 8) pro ustálení vst. zesilovače (10 μ s pro expander Pt100, 25 μ s 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 μ s, můžeme zvolit jednotlivá zpoždění takto: 8 μ s, 8 μ s, 9 μ s. 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: 20 μ s, 2.5 μ s, 2.5 μ s.

Pokud je aktivována filtrace 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 $\times n$. 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 ad převodníků pracujících paralelně (k - počet aktivních kanálů, max 32).

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 2.5 μ s. Zvolíme základní měřicí periodu 12.5 μ s, počet filtrací bude $20000/12.5 = 1600$ a jednotlivá zpoždění např. $3 + 3 + 3 + 3.5 = 12.5 \mu$ s.

4. Instalace modulu

Modul je napájen z dodávaného napájecího modulu 12 V/50 Hz AC. Ke komunikační síti je připojen Ethernet kabelem. Modul automaticky rozpozná přímé/překřížené zapojení Ethernet kabelu a je tedy možno jej připojit stejným kabelem k routeru i přímo k PC. Modul komunikuje protokolem TCP/IP a v síti pracuje jako server s pevnou IP adresou nastavitelnou programově. Výchozí hodnota je 169.254.162.10. S touto adresou funguje komunikace při přímém propojení PC a modulu pomocí dodaného kabelu. Jestliže budeme provozovat modul v lokální síti, je nutno adresu modulu změnit. Pokud adresu změníme, je možné se k továrnímu nastavení vrátit tak, že spojíme nejvyšší bit dig. vstupů se zemí – piny 7 a 16 na konektoru DIO a zapneme napájení modulu. Součástí dodávky je příslušně zapojený konektor pro snadný návrat do výchozího stavu.

Pro nastavení IP adresy použijte dodávaný program **AD14ethcontrol**. Doporučený postup nastavení adresy:

1. Zjistíme si vhodnou volnou adresu v síti, kde bude modul provozován. Pokud je v síti DHCP server, je vhodné tuto adresu přidat mezi vyhrazené.
2. Připojíme modul přímo k PC (pokud byla změněna adresa v modulu, zasuneme nastavovací konektor a vypneme a znovu zapneme napájení modulu).
3. Spustíme program **AD14ethcontrol**. Zvolíme menu *Konfigurace->IPadresa* a pokud není modul připojen stiskneme tlačítko *Def.addr.*, poté tlačítko *Připojit*. Pokud je modul připojen, můžeme zadat novou IP adresu a stisknutím tlačítka *Nas. v modulu* ji uložíme do modulu.
4. Odpojíme nastavovací konektor a vypneme a znovu zapneme napájení. Nyní je v modulu aktivní nová adresa a můžeme jej zapojit do sítě a komunikaci ověřit pomocí programu **AD14ethcontrol**.

Zapojení konektorů

Na přední straně modulu je 25 pinový Canon pro připojení měřených signálů, vedle něj je 20 pinový konektor určený pro případné rozšíření počtu vstupů na 32, oba tyto konektory jsou kompatibilní s kartou AD14PCI. Na zadní straně modulu je 20 pinový konektor s digitálními vstupy/výstupy označený DIO, dále Ethernet konektor a napájecí konektor.

K1 Canon-25 zásuvka

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
zem	4	16 nezapojeno
nezapojeno	3	15 zem
+12.7 V izolovaných	2	14 -12.7 V izolovaných
zem pro napájení	1	

K2 - DIL20 DIO – zadní strana modulu

IN0	1	2	IN1
IN2	3	4	IN3
IN4	5	6	IN5
IN6	7	8	O0
O1	9	10	O2
O3	11	12	O4
O5	13	14	O6
3.3V	15	16	GND
CLK	17	18	GND
CSTRT	19	20	TRIG

5. Popis driveru AD14ETH.DLL

Pro snadnou tvorbu vlastních programů s AD14ETH pod Windows je spolu s kartou dodáván i driver **AD14ETH.DLL** pro ovládání v libovolném jazyce umožňujícím volání funkcí z DLL knihoven. Postup instalace a příklady použití viz CD.

K dispozici jsou následující procedury a funkce:

```
function ConnectAD(modul:integer;IPAdress:pchar) : integer;stdcall; export;
procedure DisconnectAD(modul:longword);stdcall; export;
procedure ProgEn(modul:integer;chan,unip:byte);stdcall; export;
procedure ProgCh(modul:integer;channel,gain : byte;delay:word);stdcall;
export;
procedure Filter(modul:integer;filt,divid:integer);stdcall; export;
function DataLength(modul:integer):integer;stdcall; export;
function
ReadData(modul:integer;buff:pointer;MaxLength:integer):integer;stdcall;
export;
function GetIWritePointer(modul:integer): integer;stdcall; export;
function GetEWritePointer(modul:integer): integer;stdcall; export;
procedure SetEWritePointer(modul,value:integer);stdcall; export;
procedure
ReadIBuffer(modul:integer;buff:pointer;MaxInd:integer;kruh:WordBool);stdca
ll; export;
procedure StartAD(modul:integer);stdcall; export;
procedure StopAD(modul:integer);stdcall; export;
procedure SetTrigMode(modul:integer; mode : byte; level, pretrig:
integer);stdcall; export;
function GetSerialN(modul:integer) : integer;stdcall; export;
procedure GetMACAddress(modul:integer;buff:pointer);stdcall; export;
```

```

procedure SetPacketSize(modul,psize:integer);stdcall; export;
procedure GetIPAdress(modul:integer;var IPAdress,mask:longword);stdcall;
export;
procedure SetIPAdress(modul,IPAdress,mask:longword);stdcall; export;
procedure DigOut(modul,hodnota:integer);stdcall; export;
function DigIn(modul:integer):integer;stdcall; export;

```

```

function ConnectAD(modul:integer;IPAddress:pchar) : integer;stdcall; export;

```

Tato funkce je volána jako první, inicializuje paměť a navazuje komunikaci s modulem, má tyto parametry: *modul* – číslo modulu – od 0, *IPAdress* – ukazatel na pole typu byte s IP adresou příslušného modulu. Funkce vrací 0 pokud je úspěšně navázáno spojení, 1 pokud funguje TCP/IP komunikace ale modul nereaguje na povely, popřípadě číslo chyby, pokud není vůbec navázána TCP/IP komunikace. Při opakovaném volání této funkce bez odpovídajícího volání *DisconnectAD* může modul přestat správně komunikovat, pomůže krátké odpojení modulu od zdroje.

```

procedure DisconnectAD(modul:longword);stdcall; export;

```

Ukončí spojení s modulem a uvolní paměť.

```

procedure ProgEn(modul:integer;chan,unip: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)

unip - 1 - unipolární rozsah,

```

procedure ProgCh(modul: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 25 až 65535 (v desetinách μ s)

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

Tato procedura slouží k nastavení parametrů filtrace. Parametr *filt* určuje, z kolika vzorků bude počítán průměr, parametr *divid* určuje, jakým koeficientem bude výsledný součet vydělen. Je-li *divid* roven *filt*, je výsledné rozlišení 14 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. Rozsah naměřených hodnot je -8192 až 8191 pro 14 bitů bipolárně, 0 až 16383 pro 14 bit unipolárně, resp. -32768 až 32764 nebo 0 až 65532 pro 16 bitů.

Procedure SetTrigMode(modul: integer; mode: byte;level, pretrig: integer);stdcall;

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

0 - ihned po povelu start

1 - rezervováno

2 – rezervováno

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 25000 vzorků pro typy 1 až 4

function DataLength(modul:integer):integer;stdcall; export;

tato funkce vrací počet vzorků ve vyrovnávací paměti.

function

ReadData(modul:integer;buff:pointer;MaxLength:integer):integer;stdcall;
export;

tato funkce načítá data z interního bufferu do uživatelského na adrese *buff*. Tento buffer je deklarován jako pole hodnot typu signed short tj. 2 byte se znaménkem. parametr **MaxLength** udává maximální počet hodnot (jedna hodnota jsou dva byte). Rozsah hodnot je -8192 až 8191 pro 14 bitů bipolárně, 0 až 16383 pro 14 bit unipolárně, resp. -32768 až 32764 nebo 0 až 65532 pro 16 bitů. Funkce vrací počet skutečně přečtených hodnot. Příklad použití viz dodané demo.

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

vrací hodnotu zápisového pointeru v interním kruhovém bufferu vzorků, ukazuje vždy na 1. nezapsanou pozici, tj. poslední naměřená hodnota je na pozici o 1 nižší. Tato a následující funkce jsou zde hlavně kvůli kompatibilitě s AD14PCI, pro nové aplikace doporučuji používat jednodušší funkce *DataLength* a *ReadData*.

function GetEWritePointer(modul: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 záznamu pro kontinuální čtení dat bez výpadků), ukazuje vždy na 1. nezapsanou pozici, tj. poslední naměřená hodnota je na pozici o 1 nižší.

procedure SetEWritePointer(modul,value:integer);stdcall;

Nastaví zápisový pointer v uživatelském bufferu.

procedure

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

tato funkce kopíruje data z interního do uživatelského bufferu. Vždy jsou přečtena 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. Rozsah hodnot je -8192 až 8191 pro 14 bitů bipolárně, 0 až 16383 pro 14 bit unipolárně, resp. -32768 až 32764 nebo 0 až 65532 pro 16 bitů.

procedure StartAD(modul:integer);stdcall;

Tato procedura odstartuje měření, popřípadě povolí externí spouštění.

procedure StopAD(modul:integer);stdcall;

Tato procedura zastaví měření.

function GetSerialN(modul:integer) : integer;stdcall; export;

vrací sériové číslo modulu.

procedure GetMACaddress(modul:integer;buff:pointer);stdcall; export;

vrací MAC adresu modulu v poli, na které ukazuje *buff*

procedure SetPacketSize(modul,psize:integer);stdcall; export;

nastavuje velikost TCP/IP paketu v bytech, standardní velikost je 1460, pokud je však použita nízká vzorkovací rychlost, je vhodné velikost paketu snížit – modul posílá data v celých paketech a dokud není odměřen celý paket dat, modul data nepošle.

procedure GetIPaddress(modul:integer;var IPaddress,mask:longword);stdcall; export;

vrací aktuální IP adresu a masku podsítě

procedure SetIPaddress(modul,IPaddress,mask:longword);stdcall; export;

nastavuje IP adresu a masku podsítě modulu. Nová hodnota se projeví až po odpojení a znovu připojení napájení modulu.

procedure DigOut(modul,hodnota:integer);stdcall; export;

zápis na dig. výstup, *hodnota* v rozsahu 0 až 127. Jen pokud neběží AD převod!

function DigIn(modul:integer):integer;stdcall; export;

přečte stav z dig. vstupu *cislo* 1 nebo 2, vrací hodnotu v rozsahu 0 až 255. Jen pokud neběží AD převod!

