

*Západočeská univerzita
Fakulta aplikovaných věd
Ústav fyzikálního inženýrství*

Ing. Petr HORA, CSc.

TRACE 8608A

-

popis programování a ovladače pro TestPoint

Číslo zprávy : 148VP

Číslo kopie :

Vedoucí grantu : Ing. Petr HORA, CSc.

Ředitel ústavu : Doc. Ing. Miroslav BALDA, DrSc.

Plzeň, listopad 1996

ROZDĚLOVNÍK :

Výtisk číslo :	1	archivní výtisk	
	2	knihovna ÚFY FAV ZČU	
	3	Ing. Michal LANDA, CSc.	ÚT AV ČR
	4	Ing. Petr HORA, CSc.	ÚFY FAV ZČU

OBSAH

ÚVOD	7
ROZHRANÍ IEEE-488	9
ŽÁDOST O OBSLUHU.....	9
SÉRIOVÁ VÝZVA.....	9
STAVOVÉ SLOVO.....	10
STAVOVÉ SLOVO ZAŘÍZENÍ.....	10
PŘEPNUTÍ DO MÍSTNÍHO OVLÁDÁNÍ.....	11
SPOUŠTĚNÍ ZAŘÍZENÍ.....	11
SELEKTIVNÍ VYNULOVÁNÍ ZAŘÍZENÍ.....	11
VYNULOVÁNÍ VŠECH ZAŘÍZENÍ.....	11
SÉRIOVÉ ROZHRANÍ (RS232)	13
ŽÁDOST O SLUŽBU A SÉRIOVÁ VÝZVA.....	13
MÍSTNÍ A VZDÁLENÉ OVLÁDÁNÍ.....	13
VYNULOVÁNÍ ZAŘÍZENÍ.....	13
SPOUŠTĚNÍ ZAŘÍZENÍ.....	13
PROGRAMOVÁNÍ OSCILOSKOPU	14
ODDĚLOVAČE.....	14
PŘÍKAZY.....	14
ŘÁDKY.....	14
REPREZENTACE ČÍSEL.....	14
PROGRAMOVACÍ KÓDY	15
SYSTÉMOVÉ KÓDY.....	15
<i>Oddělovač příkazů</i>	15
<i>Oddělovač řádků</i>	15
<i>Identifikace</i>	16
<i>Kódy žádostí o obsluhu</i>	16
<i>Zjištění chyb</i>	17
NASTAVENÍ UZLU.....	18
KÓDY ZÁVISLÉ NA NASTAVENÍ UZLU.....	18
<i>Vertikální funkce</i>	18
<i>Horizontální funkce</i>	21
<i>Definiční funkce</i>	23
<i>Datové funkce</i>	26
<i>Časové funkce</i>	27
KÓDY NEZÁVISLÉ NA NASTAVENÍ UZLU.....	28
<i>Kanálové funkce</i>	28
<i>Posloupnostní funkce</i>	29
<i>Spouštěcí funkce</i>	31
<i>Zobrazovací funkce</i>	32
<i>Kurzorové funkce</i>	34
<i>Souborové funkce</i>	35
<i>Funkce rozhraní IEEE</i>	36
<i>Funkce sériového rozhraní</i>	37
<i>Funkce plotru</i>	38
<i>Časové funkce</i>	39
<i>Funkce uživatelské komunikace</i>	41
<i>Testovací funkce</i>	43
LITERATURA	44
DODATEK A: KÓDY KLÁVES	45
DODATEK B: FORMÁTY SOUBORŮ	47
FORMÁT SOUBORU VEKTORU.....	47
FORMÁT SOUBORU NASTAVENÍ.....	48

FORMÁT SOUBORU NASTAVENÍ A VŠECH STOP.....	51
DODATEK C: KÓDY UŽÍVANÉ V SOBORECH.....	53
KÓDY PŮVODNÍCH DEFINIC UZLŮ.....	53
KÓDY TYPŮ OPERANDŮ.....	54
KÓDY INTERPOLACÍ.....	54
KÓDY TRANSFOKACE.....	54
KÓDY VOLEB POTENCIOMETRICKÉHO ŘÍZENÍ.....	54
DODATEK D: OVLADAČE PRO TESTPOINT	55

Úvod

TRACE 8608A je kompaktní přenosný dvoukanálový digitální paměťový osciloskop, s plochou obrazovkou, umožňuje vzorkovat až 40 MS/s. Osciloskop má maximální ekvivalentní vzorkovací kmitočet 4 GHz.

Osciloskop je vybaven AUTO-SETem, který automaticky nastaví kanál A a B tak, aby bylo dosaženo co nejoptimálnějšího zobrazení pozorovaných signálů. Na ploché obdélníkové elektroluminiscenční obrazovce mohou být zobrazeny až 4 průběhy. Tyto průběhy mohou být také definovány matematickými funkcemi, jako např.: součet, rozdíl, násobek, podíl, integrace, derivace a vyhlazení.

Celá paměť je zálohována bateriemi. Z této paměti je vyčleněno přibližně 180 KB nazývaných elektronický disk. Tento elektronický disk funguje jako normální pružný disk, až na to, že je daleko rychlejší. Tento E-disk může pojmout až 99 normálních průběhů nebo nastavení. To znamená přibližně 9 maximálních průběhů (8K slov v režimu *A ONLY*) nebo 4-krát 4 maximální průběhy (4K slova ve dvoukanálovém režimu) s jejich parametry a kompletním nastavením.

Ve zprávě je popsáno jak programování tohoto digitálního osciloskopu, tak jeho ovladače pro prostředí TestPoint.

Součástí zprávy je disketa zahrnující ovladače pro TestPoint, ukázkové programy pro TestPoint a program pro načtení datových souborů z osciloskopu do počítače po sériové lince.

Tato zpráva vznikla na základě podpory grantu GAČR č. 101/94/0971 *Nové metody vyhodnocování signálů akustické emise* (řešitel Ing. Petr HORA, CSc.). Tento grant se řeší v ÚFY FAV ZČU v Plzni a ÚT AV ČR v Praze.

V Plzni 26. listopadu 1996

.....

Ing. Petr HORA, CSc.

ROZHRANÍ IEEE-488

Pro komunikaci mezi osciloskopem a počítačem je k dispozici rozhraní IEEE. Toto rozhraní je navrženo pro komunikaci několika přístrojů, programovatelných ale i neprogramovatelných, za účelem vytvořit měřicí systém. Z důvodů zjednodušení propojování a umožnění rozšiřování měřicího systému je rozhraní organizováno jako sběrnice systém, který obsahuje: 8 datových linek (DIO1...8) 3 linky řídicí výměnu dat mezi zařízeními (DAV-Data Valid, NRFD-Not Ready For Data, NDAC-Not Data Accepted) a 5 linek řídicích funkcí systému (REN-Remote Enable, ATN-Attention, IFC-InterFace Clear, SQR-Service ReQuest, EOI-End Or Identify). Mezi hlavní rysy rozhraní patří:

- zařízení mohou být od různých firem
- zařízení mohou používat různé přenosové rychlosti
- je možný asynchronní přenos dat bez řadiče (až do 1MB/s)
- pružnost systému umožňuje rychlou výstavbu jednoduchých i vysoce složitých měřicích systémů
- žádné problémy s kabeláží

Žádost o obsluhu

Libovolné zařízení v systému může použít linku SRQ (žádost o obsluhu) pro vyrozumění řadiče, že potřebuje obsloužit, i v případě, že datové linky sběrnice jsou zaneprázdněny jinou prací. Zařízení, které je v chybovém nebo nějakém výjimečném stavu smí požádat o obsluhu odesláním SRQ zprávy. Navíc činnost některých zařízení může trvat několik sekund a nemusí být zrovna ekonomické blokovat celý systém, dokud není tato činnost dokončena. Využitím schopností SRQ je řadiči umožněno pokračovat v jiných činnostech a přerušit je až v okamžiku, kdy dané zařízení dokončí svoji poměrně dlouho trvající činnost. Když je aktivován SRQ signál jedním zařízením, může řadič přerušit všechny ostatní aktivity a věnovat pozornost zařízení, které žádá o obsluhu. Nejprve musí řadič zjistit, které zařízení žádalo o obsluhu (vyslalo SRQ signál).

K této činnosti slouží funkce sériové výzvy. Možnou alternativou k této funkci je periodicky kontrolovat SRQ linku programem řadiče. Jestliže není funkce sériové výzvy v zařízení implementovaná, je tato alternativa jediná možná.

Sériová výzva

Až 14 zařízení může požádat o obsluhu přes SRQ linku (za předpokladu, že je každé zařízení vybaveno funkcí žádosti o obsluhu). Zařízení musí být také vybaveno funkcí 'talk' a funkcí sériové výzvy; jeho část, která dekóduje zprávy, musí být schopna dekódovat sběrnice příkazy SPE (Serial Poll Enable) a SPD (Serial Poll Disable).

Jestliže přístroj obdrží sériovou výzvu, vloží svůj stavový byte na datovou sběrnici. Ve stavovém bytu, DIO7 indikuje, zda přístroj žádal o obsluhu (SRQ=1). Ostatní bity poskytují dodatečné stavové informace týkající se přístroje, např. alarm, pohotovostní stav, právě měřím, atd.

V podstatě žádost o obsluhu a sériová výzva fungují následovně:

- zařízení žádá o obsluhu aktivací SRQ linky
- Řadič přijme zprávu s žádostí o obsluhu a spustí rutinu sériové výzvy nastavením zařízení do módu sériové výzvy sběrnicovým příkazem SPE (Serial Poll Enable). Tento univerzální příkaz je vyslán všem zařízením.
- Řadič potom adresuje každé zařízení jako mluvčího (talker).
- Zařízení adresované jako mluvčí odpoví odesláním svého stavového bytu na datovou sběrnici.
- Zařízení, které žádalo o obsluhu RQS zprávou, má ve svém stavovém bytu příslušně nastaven bit DIO7.
- Potom co řadič prověřil všechny stavové byty, ukončí mód sériové výzvy univerzálním sběrnicovým příkazem SPD (Serial Poll Disable) a obslouží zařízení, které žádalo o obsluhu (zařízení může být více). Řadič může být žádán o obsluhu z mnoha důvodů:
 - ◆ Zařízení je v chybovém stavu; vyskytla se programovací nebo provozní chyba.
 - ◆ Zařízení už dokončilo svou naprogramovanou úlohu a žádá o další povely.
 - ◆ Vstupní vyrovnávací paměť je plná nebo skoro plná. Vysílací zařízení musí zastavit přenos dat, aby se zabránilo blokování rozhraní nebo aby se předešlo ztrátě dat. Po této žádosti (plná vstupní vyrovnávací paměť) se žádá o další obsluhu v případě poklesu obsahu vyrovnávací paměti pod jistou úroveň, kdy lze opět začít s načítáním nových dat.
 - ◆ Nová platná data jsou k dispozici, ale zařízení je nemůže odeslat; např. proto, že není adresováno jako mluvčí.

Stavové slovo

Stavové slovo odráží stav funkcí rozhraní. Poněvadž funkce rozhraní mohou být také aktivovány funkcemi zařízení, může stavové slovo odrážet i některé stavy zařízení; např. žádost o obsluhu.

Bity ve stavovém slově, které jsou závislé na zařízení (osciloskopu), specifikují důvody žádosti o obsluhu a/nebo odráží stav funkcí zařízení.

Stavové slovo zařízení

Stavové slovo osciloskopu (8 bitů) má následující formát:

# bitu	Význam
8 ("exc/mess")	1: Data0...Data3 mají význam kódu výjimky 0: Data0...Data3 mají význam kódu zprávy
7 ("My_ Requ")	Žádost o obsluhu od osciloskopu
6 ("outrdy")	Data ve výstupní vyrovnávací paměti jsou k dispozici
5 ("cmdrdy")	Jeden příkaz vykonán
4 (Data 3)	Kód výjimky/zprávy
3 (Data 2)	Kód výjimky/zprávy
2 (Data 1)	Kód výjimky/zprávy
1 (Data 0)	Kód výjimky/zprávy

Dekódování kódů výjimek (DATA3...DATA0):

- 0: Základní systém vstupu/výstupu
- 1: Výpočetní modul
- 2: Souborový systém
- 3: Jádro systému
- 4: Uživatelský modul

Přepnutí do místního ovládání

Existují tři způsoby jak dostat osciloskop ze vzdáleného ovládání do místního: vypnout a zapnout přístroj, sběrniceovým příkazem GTL (Go To Local) nebo aktivováním REN linky.

Spouštění zařízení

Řadič může říci osciloskopu sběrniceovým příkazem GET (Group Executive Trigger), kdy má začít měření. Poněvadž existují dva módy měření, 'recurrent' a 'single', existují také dvě různé interpretace tohoto příkazu:

- 'Recurrent': žádná interpretace
- 'Single': nastaví stav čekávání na spouštěcí hranu

Selektivní vynulování zařízení

Osciloskop je schopen přijmou a zpracovat sběrniceový příkaz SDC (Selective Device Clear). Po obdržení tohoto příkazu spustí osciloskop svůj měkký restart (jako při zapnutí přístroje).

Vynulování všech zařízení

Osciloskop je schopen přijmou a zpracovat sběrniceový příkaz DCL (Device Clear), kterým řadič vynuluje všechna zařízení v systému. Po obdržení tohoto příkazu spustí osciloskop svůj měkký restart (jako při zapnutí přístroje).

SÉRIOVÉ ROZHRAŇÍ (RS232)

Pro komunikaci mezi osciloskopem a počítačem je kromě IEEE rozhraní k dispozici také sériové rozhraní. Využitelné rychlosti sériového rozhraní jsou následující: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800 a 9600 baudů (bitů/s). Osciloskop může dále využívat plně duplexního režimu.

Žádost o službu a sériová výzva

Protokol žádosti o obsluhu a sériové výzvy byl původně vyvinut pro IEEE rozhraní, ale byl následně rozšířen i na sériové rozhraní. Tato část pojednává o implementaci protokolu pro sériové rozhraní v osciloskopu TRACE.

Sériové rozhraní nemá pro účely žádosti o obsluhu vymezenou žádnou linku. To znamená, že schopnosti žádosti o obsluhu v zařízeních vybavených tímto rozhraním musí být poněkud omezeny. Pokud existuje nemaskovaná příčina žádosti o obsluhu, RQS-bit ve stavovém bytu označuje, že obsluha je požadována; pokud příčina žádosti o obsluhu je maskovaná, RQS-bit není nastaven. V některých případech může být příčina specifikovaná speciálním příkazem (např. ESQ%).

Řadič (počítač) obsluhující zařízení přes sériové rozhraní může periodicky kontrolovat zařízení, zda nepotřebuje obsloužit. Řadič spouští sériovou výzvu odesláním zprávy ESC7 (1B,37) k zařízení. Když zařízení obdrží tento příkaz k výzvě, odpoví odesláním ASCII ekvivalentu stavového bytu.

Místní a vzdálené ovládání

Tento protokol byl rovněž původně zamýšlen pro IEEE rozhraní, ale byl následně rozšířen i na sériové rozhraní. Tato část pojednává o implementaci pro sériové rozhraní osciloskopu TRACE.

Následující možnosti existují pro zařízení vybavená sériovým rozhraním a mající funkci rozhraní 'remote - local'.

Přepnutí z místního do vzdáleného ovládání:

Přepnutí z místního do vzdáleného stavu může být vykonáno pouze řadičem odesláním zprávy rozhraní ESC2 (1B,32).

Po obdržení této zprávy přejde zařízení bezpodmínečně do stavu vzdáleného řízení.

Přepnutí ze vzdáleného do místního:

Přepnutí ze vzdáleného do místního stavu může být vykonáno buď řadičem nebo zařízením následujícími způsoby:

- Řadič odešle zprávu ESC1 (1B,31) nebo zprávu ESC3 (1B,33).
- Po zapnutí zařízení je funkce rozhraní 'remote - local' vždy v místním ('local') stavu.

Vynulování zařízení

Zařízení vybavené sériovým rozhraním a mající funkci 'device clear', spouští tuto funkci při obdržení zprávy rozhraní ESC4(1B,34), tj. vynuluj zařízení. Tato zpráva způsobí měkký restart jako při zapnutí zařízení.

Spouštění zařízení

Tato funkce je spuštěna po obdržení zprávy rozhraní ESC8 (1B,38), tj. 'device trigger'. Tento příkaz je ekvivalentní příkazu WRT\$="WRITE".

PROGRAMOVÁNÍ OSCILOSKOPU

Oddělovače

Programovací jazyk je řádkově orientovaný. Jedna řádka se skládá z řady příkazů a je ukončena ODDĚLOVAČEM ŘÁDKŮ. Implicitním ODDĚLOVAČEM ŘÁDKŮ je CR (Carriage Return, Hex 0D, Dec 13). Příkazy jsou odděleny ODDĚLOVAČEM PŘÍKAZŮ, kterým je vždy dvojtečka (:).

Příkazy

Příkazy se skládají z kódového slova a těla.

Řádky

Řádky mohou obsahovat žádný, jeden nebo více příkazů; maximálně 256 bytů včetně řádkového oddělovače.

Reprezentace čísel

Numerická data mohou být vyjádřena v reálné nebo celočíselné notaci.

PROGRAMOVACÍ KÓDY

Jméno příkazu smí obsahovat max. 5 znaků, jméno proměnné smí mít max. 4 znaky a jeden znak pro identifikaci typu.

Např.: XYZt 't' je buď % nebo ! nebo \$

- identifikátor typu pro 16 bitovou celočíselnou proměnnou : %
- identifikátor typu pro 32 bitovou celočíselnou proměnnou : %
- identifikátor typu pro 32 bitovou reálnou proměnnou : !
- identifikátor typu pro řetězcovou proměnnou : \$

Pokud je proměnná uvedena bez identifikátoru typu, předpokládá se reálná proměnná. Avšak doporučuje se používat identifikátor typu, neboť se tím usnadňuje práce interpretru, který je pak rychlejší.

Obecně platí, že všechny proměnné s diskrétní množinou hodnot jsou řetězcové proměnné, i když jsou tyto řetězce numerické. Tato strategie odstraňuje problémy se zaokrouhlováním. Nevýhodou je, že tyto proměnné nelze přímo použít v numerických výrazech.

Proměnné se spojitou množinou hodnot, i když je jejich rozsah omezen, jsou celočíselné nebo reálné. Viz příklad pro TRL%.

Vyjímkou z tohoto pravidla jsou proměnné ATT! a SAM!, které smí reprezentovat pouze diskrétní množinu reálných hodnot.

Text v šedivém rámečku je název menu a funkce v ovladači TRACECOM pro TestPoint.

SYSTÉMOVÉ KÓDY

Oddělovač příkazů

Oddělovačem příkazů je vždy dvojtečka '!'. Nelze programovat.

Oddělovač řádků

Oddělovač řádků může být zjištěn nebo nastaven. Po restartu je implicitně nastaven na <CR>. Pro RS232 a HP-IB interface může být nastavena jedna hodnota pro vstup a jedna pro výstup. Poněvadž oddělovače řádků jsou obvykle netisknutelné znaky, jsou tyto proměnné celočíselné.

LSI% Oddělovač řádků pro RS232 vstup může být zjištěn a nastaven.

Ukázkový program pro vstup:

LSI%=13

```
TP Line Separator > Set Input Sep RS232
Get Input Sep RS232
```

LSO% Oddělovač řádků pro RS232 výstup může být zjištěn a nastaven.

Ukázkový program pro výstup:

LSO%=10

```
TP Line Separator > Set Output Sep RS232
Get Output Sep RS232
```


LII% Oddělovač řádků pro IEEE vstup může být zjištěn a nastaven.
Ukázkový program pro vstup:
LII%=13

TP	Line Separators >	Set Input Sep IEEE
		Get Input Sep IEEE

LIO% Oddělovač řádků pro IEEE výstup může být zjištěn a nastaven.
Ukázkový program pro výstup:
LIO%=10

TP	Line Separators >	Set Output Sep IEEE
		Get Output Sep IEEE

Identifikace

Pro identifikaci osciloskopu lze využít několika příkazů.

TYP\$ Zjištění typu osciloskopu.
Příklad zjištění typu osciloskopu:
?TYP\$
Odpověď je:
8608A

TP	Identity >	Get Type
-----------	----------------------	-----------------

VER\$ Zjištění verze software.
Příklad zjištění čísla verze:
?VER\$
Odpověď, je-li verze 1.12:
V 1.12

TP	Identity >	Get Version
-----------	----------------------	--------------------

SER\$ Zjištění výrobního čísla osciloskopu.
Příklad zjištění výrobního čísla:
?SER\$
Odpověď, je-li výrobní číslo 600:
600

TP	Identity >	Get Serial #
-----------	----------------------	---------------------

Kódy žádostí o obsluhu

CSQ% Umožnění synchronizace spouštění příkazů; umožnění vyslání žádosti o obsluhu při spuštění příkazu. Žádost o obsluhu je vyslána, kdykoliv je ve stavovém bytu nastaven bit "cmdrdy".
Přípustné hodnoty jsou: 0 nebo 1
Příklad umožnění vyslání žádosti o obsluhu při spuštění příkazu:
CSQ%=1

TP	Request Codes >	Set Command Req
		Get Command Rqs

ESQ% Umožnění ošetření chyb; umožnění vyslání žádosti o obsluhu při chybě. Žádost o obsluhu je vyslána, kdykoliv je ve stavovém bytu nastaven bit "exc/mess".
Přípustné hodnoty jsou: 0 nebo 1
Příklad umožnění vyslání žádosti o obsluhu při chybě:
ESQ%=1

TP	Request Codes >	Set Except Req
		Get Except Rqs

LSQ% Umožnění synchronizace dat; umožnění vyslání žádosti o obsluhu při připravenosti dat. Žádost o obsluhu je vyslána, kdykoliv je ve stavovém bytu nastaven bit "outdry".
Přípustné hodnoty jsou: 0 nebo 1
Příklad umožnění vyslání žádosti o obsluhu při připravenosti dat:
LSQ%=1

TP	Request Codes >	Set Data Req
		Get Data Rqs

TSQ% Umožnění synchronizace nahrávacích a početních cyklů; umožnění vyslání žádosti o obsluhu při dokončení nahrávání nebo výpočtu. Žádost o obsluhu je vyslána, kdykoliv je dokončen nahrávací nebo početní cyklus. (Stejná podmínka je používána pro TDR%):
Přípustné hodnoty jsou: 0 nebo 1
Příklad umožnění vyslání žádosti o obsluhu při dokončení nahrávání nebo výpočtu.:
TSQ%=1

TP	Request Codes >	Set Trace Req
		Get Trace Rqs

Zjištění chyb

IEX% Podrobná informace o chybě; zjištění aktuálního kódu chyby.
Pozor, po přečtení se vynuluje!
Příklad zjištění kódu poslední chyby:
?IEX%
Odpověď, pokud nedošlo k chybě:
0

TP	Exception >	Get Exception Code
-----------	-----------------------	---------------------------

IEX\$ Podrobná informace o chybě; zjištění aktuálního řetězce popisujícího chybu. Pozor, po přečtení se vynuluje!
Příklad zjištění řetězce popisujícího poslední chybu:
?IEX\$
Odpověď, pokud nedošlo k chybě:
OK

TP	Exception >	Get Exception Msg
-----------	-----------------------	--------------------------

NASTAVENÍ UZLU

Kanály, vypočtené průběhy a vypočtená čísla lze považovat za uzly synchronizovaného výpočetního a měřicího systému. Parametry uzlu mohou být zpřístupněny dvojím způsobem:

- specifikovat požadovaný uzel při zadávání příkazu
- určit jeden uzel a potom zadat příkaz bez specifikace uzlu

Protože kanálové, vektorové a skalární uzly mají své typické vlastnosti, ne všechny příkazy pracují na všech typech uzlů. Např. příkazy **CPL\$** nebo **PRO%** lze aplikovat pouze na kanálové uzly. Operace na kanálových uzlech ovlivňují nastavení kanálů.

TRS\$ Zjištění a nastavení určeného implicitního uzlu.

Platná jména uzlů jsou:

CHA | CHB | TR1 | TR2 | TR3 | TR4 | FU1 | FU2 | FU3 | FU4 | SYS

Příklad nastavení kanálu A jako implicitního uzlu:

TRS\$="CHA"

Příklad zjištění určeného jména uzlu:

?TRS\$

Odpověď, je-li určený uzel CHA:

CHA

Pokud je TR1 definován jako kanálový uzel pro kanál A, může být použito jak TR1, tak CHA. SYS určuje systémové parametry a používá se pro zjištění a nastavení času. (RTS\$, HOU%, MIN%, ...).

TP	Nodes >	Set Node
		Get Node

KÓDY ZÁVISLÉ NA NASTAVENÍ UZLU

Kódy závislé na super funkci ovlivňují parametry, které existují v každém uzlu. Můžete specifikovat požadovaný uzel spolu s příkazem jinak se, pokud jméno uzlu chybí, implicitně uvažuje určený uzel.

Vertikální funkce

ATT% Zjištění a nastavení kódu amplitudového rozsahu.

Tato proměnná je využitelná pouze pro kanálové uzly.

Přípustné hodnoty jsou: 0...9

Příklad nastavení amplitudového rozsahu určeného uzlu na úroveň 5:

ATT%=5

Příklad zjištění amplitudového rozsahu určeného uzlu:

?ATT%

Odpověď, je-li úroveň 5:

5

Příklad nastavení amplitudového rozsahu kanálu A na úroveň 5:

ATT%("CHA")=5

Příklad zjištění amplitudového rozsahu kanálu A:

?ATT%("CHA")

Odpověď, je-li úroveň 5:

5

**TP Vertical Functions > Set Atten Code
Set Atten Code N
Get Atten Code**

ATT! Zjištění plného amplitudového rozsahu nebo nastavení nejbližšího možného plného amplitudového rozsahu. LSB uzlu = ATT!/0FFFFH.
Tato proměnná je využitelná pouze pro kanálové, vektorové a skalární uzly.
Přípustné hodnoty pro kanály se sondou 1:1 jsou:
0.032, 0.064, 0.128
Přípustné hodnoty pro kanály se sondou 1:1 nebo 1:10 jsou:
0.32, 0.64, 1.28, 3.2, 6.4, 12.8, 32
Přípustné hodnoty pro kanály se sondou 1:10 jsou:
64, 128, 320
Příklad nastavení amplitudového rozsahu určeného uzlu na 32V:
ATT!=32
Příklad zjištění amplitudového rozsahu určeného uzlu:
?ATT!
Odpověď, je-li plný rozsah 32V:
32
Příklad nastavení amplitudového rozsahu kanálu A na 32V:
ATT!("CHA")=32
Příklad zjištění amplitudového rozsahu kanálu A:
?ATT!("CHA")
Odpověď, je-li rozsah 32V:
32

**TP Vertical Functions > Set Attenuation
Set Attenuation N
Get Attenuation**

CPL\$ Zjištění a nastavení vazby kanálového uzlu.
Tato proměnná je využitelná pouze pro kanálové uzly.
Přípustné hodnoty jsou: AC | DC | GND
Příklad nastavení střídavé vazby určeného uzlu:
CPL\$="AC"
Příklad zjištění vazby určeného uzlu:
?CPL\$
Odpověď, je-li vazba střídavá:
AC
Příklad nastavení střídavé vazby kanálu A:
CPL\$("CHA")="AC"
Příklad zjištění vazby kanálu A:
?CPL\$("CHA")
Odpověď, je-li vazba střídavá:
AC

**TP Vertical Functions > Set ChCoupling
Set ChCoupling N
Get ChCoupling**

OFF! Zjištění skutečné hodnoty offsetu nebo nastavení offsetu na nejbližší možnou hodnotu (podle využitelných hodnot OFF%).
 Tato proměnná je využitelná pro kanálové, vektorové a skalární uzly.
 $OFF! = (ATT! * OFF\% * 5120) / OFFFFH$.
 Příklad nastavení offsetu určeného uzlu na 32V:
 OFF!=32
 Příklad zjištění offsetu určeného uzlu:
 ?OFF!
 Odpověď, je-li offset 32V:
 32
 Příklad nastavení offsetu kanálu A na 32V:
 OFF!("CHA")=32
 Příklad zjištění offsetu kanálu A:
 ?OFF!("CHA")
 Odpověď, je-li offset 32V:
 32

TP Vertical Functions > Set Offset
Set Offset N
Get Offset

OFF% Zjištění a nastavení offsetu. Hodnota offsetu 1 odpovídá kroku 5120.
 Tato proměnná je využitelná pouze pro kanálové uzly.
 Přípustné hodnoty jsou: -6, -5, ..., 0, ..., 5, 6
 Příklad nastavení offsetu určeného uzlu asi na 30% plného rozsahu:
 OFF%=2
 Příklad zjištění offsetu určeného uzlu:
 ?OFF%
 Odpověď, je-li offset asi 30% plného rozsahu:
 2
 Příklad nastavení offsetu kanálu A asi na -90% plného rozsahu:
 OFF%("CHA")=-6
 Příklad zjištění offsetu kanálu A:
 ?OFF%("CHA")
 Odpověď, je-li offset asi -90% plného rozsahu:
 -6

TP Vertical Functions > Set Offset Code
Set Offset Code N
Get Offset Code

PRO% Zjištění typu sondy.
 Tato proměnná je využitelná pouze pro kanálové uzly.
 Příklad zjištění typu sondy určeného kanálového uzlu:
 ?PRO%
 Odpověď, jde-li o sondu 10:1:
 10
 Příklad zjištění typu sondy na kanálu A:
 ?PRO%("CHA")
 Odpověď, jde-li o sondu 100:1:
 100

TP Vertical Functions > Get Probe Factor

Horizontální funkce

DLY! Zjištění skutečné hodnoty zpoždění nebo nastavení skutečné hodnoty zpoždění na nejbližší možnou hodnotu (podle využitelných hodnot DLY%; $DLY! = DLY\% * SAM!$).

Tato proměnná je využitelná pro kanálové a vektorové uzly.

Pro záporná zpoždění jsou hodnoty spojité, ale pro kladná zpoždění mohou být nastaveny pouze hodnoty násobků 40-ti vzorků.

Příklad nastavení zpoždění určeného uzlu na $-320\mu s$. Zaznamená se již $320\mu s$ před spuštěním.

$DLY! = -320e-6$

Příklad nastavení zpoždění kanálu A na $120\mu s$. K záznamu dojde až $120\mu s$ po spuštění.

$DLY!("CHA") = 120e-6$

Příklad zjištění zpoždění kanálu A:

?DLY!("CHA")

Odpověď, je-li zpoždění $120\mu s$:

1.2E-04

Příklad zjištění zpoždění určeného uzlu:

?DLY!

Odpověď, je-li zpoždění $-320\mu s$:

-3.2e-04

TP	Horizontal Functions >	Set Delay
		Set Delay N
		Get Delay

DLY% Zjištění hodnoty zpoždění nebo nastavení hodnoty zpoždění.

Tato hodnota má význam počtu vzorků mezi spuštěním (trigger) a prvním zaznamenaným bodem.

Ne všechny hodnoty jsou nastavitelné. Pro záporná zpoždění jsou hodnoty spojité, ale pro kladná zpoždění mohou být nastaveny pouze násobky 40.

Tato proměnná je využitelná pro kanálové uzly.

Přípustné hodnoty jsou:

-39960 (=999 dílků) ... skutečná délka záznamu

Příklad nastavení zpoždění určeného uzlu na -100 časových tiků. Spouštěcí značka (trigger line) je 100 bodů napravo od začátku stopy:

$DLY\% = -100$

Příklad nastavení zpoždění kanálu A na $+120$ časových tiků. Spouštěcí značka (trigger line) je 120 bodů nalevo od začátku stopy:

$DLY\%("CHA") = 120$

Příklad zjištění zpoždění kanálu A:

?DLY%("CHA")

Odpověď, je-li zpoždění 120 časových tiků:

120

Příklad zjištění zpoždění určeného uzlu:

?DLY%

Odpověď, je-li zpoždění -100 časových tiků:

-100

TP	Horizontal Functions >	Set Delay Code
		Set Delay Code N
		Get Delay Code

SAM% Zjištění vzorkovací periody nebo její nastavení. Tento příkaz bývá používán při programování narůstající nebo klesající posloupnosti.

Tato proměnná je využitelná pouze pro kanálové uzly.

Příпустné hodnoty jsou:

0...5 vzorkovací mód 'RECURRENT' pro opakující se signály
6...22 vzorkovací mód 'RECURRENT'
23...35 vzorkovací mód 'ROLL'

Příklad nastavení vzorkovací rychlosti určeného uzlu na úroveň 12:

SAM%=12

Příklad zjištění vzorkovací rychlosti určeného uzlu:

?SAM%

Odpověď, je-li úroveň 12:

12

Příklad nastavení vzorkovací rychlosti kanálu A na úroveň 12:

SAM%("CHA")=12

Příklad zjištění vzorkovací rychlosti kanálu A:

?SAM%("CHA")

Odpověď, je-li úroveň 12:

12

TP	Horizontal Functions >	Set Sample Code
		Set Sample Code N
		Get Sample Code

SAM! Zjištění skutečné vzorkovací periody nebo její nastavení na nejbližší možnou hodnotu.

Tato proměnná je využitelná pro kanálové a vektorové uzly.

V módu RECURRENT u opakujících se signálů jsou možné hodnoty pro kanály:

.25E-09, .5E-09, 1.25E-09, 2.5E-09, 5E-09, 12.5E-09

V módu RECURRENT jsou možné hodnoty pro kanály:

25E-09, 50E-09, 125E-09, 250E-09, 500E-09, 1.25E-06, 2.5E-06, 5E-06,
12.5E-06, 25E-06, 50E-06, 125E-06, 250E-06, 500E-06, 1.25E-03, 2.5E-03
5.0E-03

V módu ROLL jsou možné hodnoty pro kanály:

12.5E-03, 25E-03, 50E-03, 125E-03, 250E-03, 500E-03, 750E-03, 1.5E+00
3E+00, 9E+00, 22.5E+00, 45E+00, 90E+00

Možné hodnoty pro vypočtené uzly jsou platné v rozsahu:

0 ... 8.43E-37 ... 3.38E38

Příklad nastavení vzorkovací periody určeného uzlu na 125ns:

SAM!=125E-09

Příklad zjištění vzorkovací periody určeného uzlu:

?SAM!

Odpověď, je-li vzorkovací perioda 125ns:

125E-09

Příklad nastavení vzorkovací periody kanálu A na 125ns:

SAM!("CHA")=125E-09

Příklad zjištění vzorkovací periody kanálu A:

?SAM!("CHA")

Odpověď, je-li vzorkovací perioda 125ns:

125E-09

PPT%

Zjištění délky vektoru nebo její nastavení na nejbližší možnou hodnotu. Tento příkaz je využitelný pouze pro vektorové uzly (zejména paměťové vektorové uzly).

Délka kanálových uzlů je řízena proměnnými ALY% a MXM%, délka vypočtených vektorových uzlů závisí na jejich operandech a délka skalárních (funkčních) uzlů je vždy 1.

Přípustné hodnoty jsou:

V šestnáctkové reprezentaci: FFFF, FFFE, ..., 1

V desítkové reprezentaci: 65535, 65534, ..., 1

Příklad nastavení délky určeného uzlu na 16384 vzorků:

PPT%=16384

Příklad zjištění délky určeného uzlu:

?PPT%

Odpověď, je-li délka 5 vzorků:

5

Příklad nastavení délky 4. vektorového uzlu na 16384 vzorků:

PPT%("TR4")=16384

Příklad zjištění délky 4. vektorového uzlu:

?PPT%("TR4")

Odpověď, je-li délka 5 vzorků:

5

Definiční funkce

ORI\$

Zjištění původní definice uzlu nebo její nastavení.

Tato proměnná je využitelná pro vektorové a skalární uzly.

Možné řetězce jsou:

ORI\$("přijímací uzel")=funkce("zdrojový uzel")

ORI\$("přijímací uzel")=funkce("zdrojový uzel",ii)

ORI\$("přijímací uzel")=funkce("zdrojový uzel",rr)

ORI\$("přijímací uzel")=funkce("zdrojový uzel")

ORI\$("přijímací uzel")=funkce("zdrojový uzel","zdrojový uzel")

Možné vektorové přijímací uzly jsou: TR1 | TR2 | TR3 | TR4

Možné skalární přijímací uzly jsou: FU1 | FU2 | FU3 | FU4

Možné zdrojové uzly jsou: CHA | CHB | TR1 | TR2 | TR3 | TR4 | Mnn

Kde nn je číslo (00...99) vektoru uloženého na disku E.

Možné řetězce definující **jednooperandové vektorové funkce** jsou:

pro vymazání vektorového nebo skalárního uzlu:

OFF

pro normální zobrazení jednoho kanálu nebo vektoru:

EQU("zdrojový uzel")

Možné zdrojové uzly jsou: CHA | CHB | Mnn

pro derivaci jednoho vektoru:

DIF("zdrojový uzel", "sf")

Možné zdrojové uzly jsou: CHA | CHB | TR1 | TR2 | TR3 | TR4

sf=2 | 5 | 20 | 100 | 500 | 1000

pro integraci jednoho vektoru:

INT("zdrojový uzel", "sf")

Možné zdrojové uzly jsou: CHA | CHB | TR1 | TR2 | TR3 | TR4

sf=2 | 5 | 20 | 100 | 500 | 1000

pro negaci jednoho vektoru:

NEG("zdrojový uzel")

Možné zdrojové uzly jsou: CHA | CHB

pro vyhlazení jednoho vektoru:

SMO("zdrojový uzel", "sf")

Možné zdrojové uzly jsou: CHA | CHB | TR1 | TR2 | TR3 | TR4

sf=2 | 4 | 8 | 16 | 32 | 64

Možné řetězce definující **dvouoperandové vektorové funkce** jsou:

pro sečtení dvou vektorů:

ADD("zdrojový uzel 1", "zdrojový uzel 2")

pro vydělení dvou vektorů:

DIV("zdrojový uzel 1", "zdrojový uzel 2")

pro vynásobení dvou vektorů:

MUL("zdrojový uzel 1", "zdrojový uzel 2")

pro odečtení dvou vektorů:

SUB("zdrojový uzel 1", "zdrojový uzel 2")

Možné kombinace zdrojových uzlů 1 a 2 jsou:

(CHA, CHB), (CHB, CHA), (CHA, Mnn) a (CHB, Mnn)

Možné řetězce definující **skalární uzly** jsou:

Jednoduchá amplitudová měření:

Amplituda v místě normálního kurzoru vzhledem k nulové úrovni:

CZA("zdrojový uzel")

Amplituda v místě referenčního kurzoru vzhledem k nulové úrovni:

RZA("zdrojový uzel")

Rozdíl amplitud v místě normálního a referenčního kurzoru:

CRA("zdrojový uzel")

Hodnota maxima mezi kurzory (vzhledem k nulové úrovni):

MAA("zdrojový uzel")

Hodnota minima mezi kurzory (vzhledem k nulové úrovni):

MIA("zdrojový uzel")

Rozdíl maxima a minima mezi kurzory:

PPA("zdrojový uzel")

Střední hodnota amplitud v místě normálního a referenčního kurzoru:

FIA("zdrojový uzel")

Rozdíl 10% a 90% hodnoty amplitudy v místě normálního a referenčního kurzoru:

RIA("zdrojový uzel")

Integrální amplitudová měření:

Stejnoseměrná složka signálu (vypočtená z celé periody mezi kurzory):

DCA(" zdrojový uzel")

Stejnoseměrná složka signálu (vypočtená mezi kurzory):

MEA("zdrojový uzel")

Střední kvadratická hodnota amplitudy:

RMS("zdrojový uzel")

Jednoduchá amplitudová poměrová měření:

Rozdíl amplitud normálního kurzoru levého vektoru a referenčního kurzoru pravého vektoru; lineární:

CRI("zdrojový uzel", "zdrojový uzel")

Rozdíl amplitud normálního kurzoru levého vektoru a referenčního kurzoru pravého vektoru; logaritmičtí:

CRO("zdrojový uzel", "zdrojový uzel")

Integrální amplitudová poměrová měření:

Rozdíl střední kvadratické hodnoty levého a pravého vektoru; lineárně:

RMI("zdrojový uzel", "zdrojový uzel")

Rozdíl střední kvadratické hodnoty levého a pravého vektoru; logaritmičtí:

RMO("zdrojový uzel", "zdrojový uzel")

Jednoduchá časová měření:

Čas mezi normálním a referenčním kurzorem:

CRT("zdrojový uzel")

Čas mezi normálním kurzorem a spouštěcí značkou (trigger):

CTT("zdrojový uzel")

Čas mezi referenčním kurzorem a spouštěcí značkou (trigger):

RTT("zdrojový uzel")

Doba náběhu signálu z 10% na 90% mezivrcholové hodnoty:

RIT("zdrojový uzel")

Integrální časová měření:

Frekvence (vypočtená z celé periody mezi kurzory):

FRQ("zdrojový uzel")

Perioda (vypočtená z celé periody mezi kurzory):

PER("zdrojový uzel")

Jednoduchá časová poměrová měření:

Fázový rozdíl levého a pravého vektoru:

PHT("zdrojový uzel","zdrojový uzel")

Integrální časová poměrová měření:

Fázový rozdíl levého a pravého vektoru ve stupních:

PHD("zdrojový uzel","zdrojový uzel")

Možné zdrojové uzly pro řetězce definující skalární uzly jsou:

TR1 | TR2 | TR3 | TR4

Příklad nastavení původní definice určeného uzlu na kanál A:

ORI\$=EQU("CHA")

Příklad zjištění původní definice určeného uzlu:

?ORI\$

Odpověď, je-li původní definice CHA+CHB:

ADD("CHA","CHB")

Příklad nastavení původní definice uzlu TR1:

ORI\$("TR1")=SUB("CHA","M01")

Příklad zjištění původní definice uzlu FU2:

?ORI\$("FU2")

Odpověď, je-li původní definice normální kurzor - referenční kurzor vektoru TR3:

CRT("TR3")

TP	Definition Functions >	Get Definition
		Set Definition

Datové funkce

TRA!

Zjištění a nastavení skutečné hodnoty uzlu.

Tato proměnná je využitelná pro vektorové a skalární uzly.

Pro skalární uzly je možné pouze čtení (zjišťování).

Nastavování je ve skutečnosti užitečné pouze u paměťových vektorových uzlů (Mnn).

Příklad nastavení n-tého prvku určeného uzlu na 65.536V.

n může být v rozsahu 0 ... PPT% (délka uzlu):

TRA!(,n)=65.536

Příklad zjištění n-tého prvku určeného uzlu:

?TRA!(,n)

Odpověď, je-li skutečná hodnota 65.536V:

65.536

Příklad nastavení n-tého prvku kanálu A na 65.536V:

TRA!("CHA",n)=65.536

Příklad zjištění n-tého prvku kanálu A:

?TRA!("CHA",n)

Odpověď, je-li skutečná hodnota 65.536V:

65.536

TP	Data Functions >	Set Point
		Set Data Real
		Set Data Real N
		Get Data Real

TRA% Zjištění a nastavení celočíselné hodnoty uzlu.
Tato proměnná je využitelná pro vektorové a skalární uzly.
Pro skalární uzly je možné pouze čtení (zjišťování).
Nastavování je ve skutečnosti užitečné pouze u paměťových vektorových uzlů.
Příklad nastavení n-tého prvku určeného uzlu na 32767:
n může být v rozsahu 0 ... PPT% (délka uzlu):
TRA%(,n)=32767
Příklad zjištění n-tého prvku určeného uzlu:
?TRA%(,n)
Odpověď, je-li celočíselná hodnota 32767:
32767
Příklad nastavení n-tého prvku kanálu A na 32767:
TRA%("CHA",n)=32767
Příklad zjištění n-tého prvku kanálu A:
?TRA%("CHA",n)
Odpověď, je-li celočíselná hodnota 32767:
32767

TP	Data Functions >	Set Point
		Set Data Int
		Set Data Int N
		Get Data Int

Časové funkce

RTS\$ Zjištění a nastavení času a data.
Tato proměnná je využitelná pro kanálové, vektorové a skalární uzly a pro systém.
Možné hodnoty jsou: ENTER | STOP
Příklad nastavení reálného času:
RTS\$("SYS")="ENTER"
Příklad zjištění reálného času:
?RTS\$("SYS")
Odpověď může být:
ENTER
Příkaz RTS\$="STOP" uloží aktuální systémový čas a datum nebo čas a datum vektorového uzlu do vnitřního zásobníku. Potom můžete zjišťovat jednotlivé položky pomocí příkazů ?HOU%, ?MIN%, ... nebo nastavovat jednotlivé položky pomocí příkazů HOU%=12, MIN%=40, ... Upravený čas a datum se pak zapíše do systému nebo vektorového uzlu příkazem RTS\$="ENTER". Pokud upravujete pouze část času a data bez předchozího příkazu RTS\$="STOP", zbývající položky zůstanou nedefinované. Čas kanálových resp. skalárních uzlů se mění po každém záznamu resp. výpočtu.
Následující posloupnost příkazů nastaví čas a datum systémových hodin na 26.10.1987, 12:45:00:
CEN%=19:YEA%=87:MON%=10:DAY%=26:HOU%=12:MIN%=45:SEC%=0
RTS\$("SYS")="ENTER"
Následující posloupnost kopíruje čas a datum ze systémových hodin do všech vektorových uzlů:
RTS\$("SYS")="STOP":RTS\$("TR1")="ENTER":RTS\$("TR2")="ENTER"
RTS\$("TR3")="ENTER":RTS\$("TR4")="ENTER"
Poslední posloupnost zjistí čas záznamu 1. vektorového uzlu:
RTS\$("TR1")="STOP"
?CEN%,YEA%,MON%,DAY%,HOU%,MIN%,SEC%

KÓDY NEZÁVISLÉ NA NASTAVENÍ UZLU

Kanálové funkce

ALY% Zjištění a nastavení stavu *Pouze kanál A*; (v tomto režimu lze vzorkovat až 40 MS/s).
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení stavu *Pouze kanál A* na 1 (zapnutí):
ALY%=1
Příklad zjištění stavu *Pouze kanál A*:
?ALY%
Odpověď, je-li stav *Pouze kanál A* na 1:
1

**TP Channel Functions > Set A only
Get A only**

AVC\$ Zjištění a nastavení počtu bodů uvažovaných při průměrování.
Přípustné hodnoty jsou: 4 | 16 | 64 | 256 | 1024 | 4096 | 8192
Příklad nastavení počtu průměrovaných bodů na 16:
AVC\$="16"
Příklad zjištění počtu průměrovaných bodů:
?AVC\$
Odpověď, je-li počet průměrovaných bodů 16:
16

**TP Channel Functions > Set Avg Count
Get Avg Count**

AVS% Zjištění a nastavení stavu *Průměrovat*; tj. zda se má či nemá průměrovat.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení stavu *Průměrovat* na 1:
AVS%=1
Příklad zjištění stavu *Průměrovat*:
?AVS%
Odpověď, je-li stav *Průměrovat* na 1:
1

**TP Channel Functions > Set Average
Get Average**

AVM\$ Zjištění a nastavení průměrovacího módu.
Přípustné hodnoty jsou: CONT | BLOCK
Příklad nastavení spojitého průměrovacího módu:
AVM\$="CONT"
Příklad zjištění průměrovacího módu:
?AVM\$
Odpověď, je-li nastaven spojitý průměrovací mód:
CONT

**TP Channel Functions > Set Avg Mode
Get Avg Mode**

BWL% Zjištění a nastavení stavu *Omezená šířka pásma*.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení stavu *Omezená šířka pásma* na 1:
BWL%=1
Příklad zjištění stavu *Omezená šířka pásma*:
?BWL%
Odpověď, je-li stav *Omezená šířka pásma* na 1:
1

**TP Channel Functions > Set BWL
Get BWL**

GLD% Zjištění a nastavení stavu *Detekce špiček*.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení stavu *Detekce špiček* na 1:
GLD%=1
Příklad zjištění stavu *Detekce špiček*:
?GLD%
Odpověď, je-li stav *Detekce špiček* nastaven na 1:
1

**TP Channel Functions > Set Min/Max
Get Min/Max**

MXM% Zjištění a nastavení stavu *Maximální paměť*.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení stavu *Maximální paměť* na 1:
MXM%=1
Příklad zjištění stavu *Maximální paměť*:
?MXM%
Odpověď, je-li stav *Maximální paměť* nastaven na 1:
1

**TP Channel Functions > Set Max Memory
Get Max Memory**

Posloupnostní funkce

CAL% Spustí autokalibraci.
Přípustné hodnoty jsou: 0 nebo 1
Příklad spuštění autokalibrace:
CAL%=1

TP Sequence Functions > Autocal

MOD\$ Zjištění a nastavení nahrávacího módu.
 Přípustné hodnoty jsou: RECURRENT | SINGLE | ROLL
 Příklad nastavení nahrávacího módu do režimu, kdy se nahraje pouze jedna událost:
 MOD\$="SINGLE"
 Příklad zjištění nahrávacího módu:
 ?MOD\$
 Odpověď , je-li nahrávací mód nastaven do režimu nahrávání pouze jedné události:
 SINGLE

TP	Sequence Functions >	Set Rec Mode
		Get Rec Mode

REL% Zjištění a nastavení stavu *Povolení nahrávání*.
 Přípustné hodnoty jsou: 0 nebo 1
 Příklad nastavení stavu *Povolení nahrávání* na 1:
 REL%=1

TP	Sequence Functions >	Set Release
		Get Release

SET% Spuštění automatického nastavení.
 Přípustné hodnoty jsou: 0 nebo 1
 Příklad spuštění automatického nastavení:
 SET%=1

TP	Sequence Functions >	Autoset
-----------	--------------------------------	----------------

TDR% Zjištění a nastavení stavu *Nový záznam nebo výpočet*.
 Stejná podmínka je používána pro TSQ%.
 Přípustné hodnoty jsou: 0 nebo 1
 Příklad nastavení stavu *Nový záznam nebo výpočet* na 0 (stará data):
 TDR%=0
 Příklad zjištění stavu *Nový záznam nebo výpočet*:
 ?TDR%
 Odpověď , je-li stav *Nový záznam nebo výpočet* nastaven na 1 (nová data):
 1

TP	Sequence Functions >	Set Data Ready
		Get Data Ready

WRT\$ Zjištění a nastavení stavu *Nahrávání*.
 Přípustné hodnoty jsou: WRITE | LOCK
 Příklad nastavení stavu *Nahrávání* na "WRITE" (tj. nahrávej):
 WRT\$="WRITE"
 Příklad zjištění stavu *Nahrávání*:
 ?WRT\$
 Odpověď , je-li stav *Nahrávání* nastaven na "WRITE":
 WRITE

TP	Sequence Functions >	Set Start/Stop
		Get Start/Stop

Spouštěcí funkce

ATR% Zjištění a nastavení funkce samospouštění. V módu RECURRENT a SINGLE funkce samospouštění uvolní po krátké době spouštěcí (trigger) signál. V módu ROLL spustí funkce samospouštění nahrávání v nekonečné smyčce.

Přípustné hodnoty jsou: 0 nebo 1

Příklad nastavení samospouštěcí funkce na 1:

ATR%=1

Příklad zjištění samospouštěcí funkce:

?ATR%

Odpověď, je-li samospouštěcí funkce nastavena na 1:

1

**TP Trigger Functions > Set Autotrigger
Get Autotrigger**

ECP\$ Zjištění a nastavení vazby externího spouštění.

Přípustné hodnoty jsou: AC | DC | LF | HF | AC_HF | TVF

Příklad nastavení vazby externího spouštění na televizní rozklad:

ECP\$="TVF"

Příklad zjištění vazby externího spouštění:

?ECP\$

Odpověď, je-li vazba externího spouštění nastavena na televizní rozklad:

TVF

**TP Trigger Functions > Set Trg Coupling
Get Trg Coupling**

TRL% Zjištění hodnoty spouštěcí úrovně nebo nastavení hodnoty spouštěcí úrovně na nejbližší možnou hodnotu.

Přípustné hodnoty jsou:

V šestnáctkové reprezentaci:

7F00, 7E00, ..., 100, 0, FF00, ..., 8100, 8000

V desítkové reprezentaci:

32512, 32256, ..., 256, 0, -256, ..., -32512, -32768

Příklad nastavení hodnoty spouštěcí úrovně určeného uzlu na 1/4 rozsahu:

TRL%=16384

Příklad zjištění hodnoty spouštěcí úrovně určeného uzlu:

?TRL%

Odpověď, je-li hodnota spouštěcí úrovně 1/4 rozsahu:

16384

Příklad nastavení hodnoty spouštěcí úrovně kanálu A na -1/2 rozsahu:

TRL%("CHA")=-32768

Příklad zjištění hodnoty spouštěcí úrovně kanálu A:

?TRL%("CHA")

Odpověď, je-li hodnota spouštěcí úrovně -1/2 rozsahu:

-32768

**TP Trigger Functions > Set Trg Level
Set Trg Level N
Get Trg Level**

TGH\$ Zjištění a nastavení spouštěcí hystereze.
Přípustné hodnoty jsou: 1% | 2% | 5% | 10% | 20% | 50%
Příklad nastavení spouštěcí hystereze na 10% rozsahu:
TGH\$=10%
Příklad zjištění spouštěcí hystereze:
?TGH\$
Odpověď, je-li spouštěcí hystereze 10% rozsahu:
10%

**TP Trigger Functions > Set Trg Hyster
Get Trg Hyster**

TGS\$ Zjištění a nastavení zdroje spouštění.
Přípustné hodnoty jsou: CHA | CHB | EXT | EXT_10 | LINE
Příklad nastavení externího zdroje spouštění:
TGS\$="EXT"
Příklad zjištění zdroje spouštění:
?TGS\$
Odpověď, je-li nastaven externí zdroj spouštění:
EXT

**TP Trigger Functions > Set Trg Source
Get Trg Source**

TSL% Zjištění a nastavení spouštěcí hrany.
Přípustné hodnoty jsou: 0 (sestupná) nebo 1 (náběžná).
Příklad nastavení náběžné spouštěcí hrany:
TSL%=1
Příklad zjištění spouštěcí hrany:
?TSL%
Odpověď, je-li spouštěcí hrana náběžná:
1

**TP Trigger Functions > Set Trg Slope
Get Trg Slope**

Zobrazovací funkce

D12\$ Zjištění a nastavení zobrazení vektorů TR1 a TR2.
Přípustné hodnoty jsou: YT | XY
Příklad nastavení zobrazení vektorů TR1 a TR2 na xy:
D12\$="XY"
Příklad zjištění zobrazení vektorů TR1 a TR2:
?D12\$
Odpověď, je-li zobrazení vektorů TR1 a TR2 xy:
XY

**TP Display Functions > Set Display 12
Get Display 12**

D34\$ Zjištění a nastavení zobrazení vektorů TR3 a TR4.
Přípustné hodnoty jsou: YT | XY
Příklad nastavení zobrazení vektorů TR3 a TR4 na xy:
D34\$="XY"
Příklad zjištění zobrazení vektorů TR3 a TR4:
?D34\$
Odpověď, je-li zobrazení vektorů TR3 a TR4 xy:
YT

**TP Display Functions > Set Display 34
Get Display 34**

DOT\$ Zjištění a nastavení interpolační funkce.
Přípustné hodnoty jsou: LIN | PLS | SIN | OFF
Příklad nastavení lineární interpolační funkce:
DOT\$="LIN"
Příklad zjištění interpolační funkce:
?DOT\$
Odpověď, je-li nastavena lineární interpolační funkce:
LIN

**TP Display Functions > Set Interpolation
Get Interpolation**

EXP\$ Zjištění a nastavení transfokační funkce.
Přípustné hodnoty jsou: 0.05 | 0.1 | 1 | 10
Příklad nastavení transfokační funkce na 10x:
EXP\$="10"
Příklad zjištění transfokační funkce:
?EXP\$
Odpověď, je-li transfokační funkce 10x:
10

**TP Display Functions > Set X-Zoom
Get X-Zoom**

ILL% Zjištění a nastavení zobrazení rastru.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení rastru na 1 (zobrazit):
ILL%=1
Příklad zjištění zobrazení rastru:
?ILL%
Odpověď, je-li rastr zobrazen:
1

**TP Display Functions > Set Grid
Get Grid**

SEP% Zjištění a nastavení separátního zobrazení.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení separátního zobrazení na 1 (zobrazit separátně):
SEP%=1
Příklad zjištění separátního zobrazení:
?SEP%
Odpověď, je-li separátní zobrazení zapnuto:
1

**TP Display Functions > Set Separation
Get Separation**

XPO% Zjištění a nastavení pozice vektoru na obrazovce. Závisí na transfokaci a délce vektoru.
Příklad nastavení pozice vektoru na obrazovce na hodnotu 1:
XPO%=1
Příklad zjištění pozice vektoru na obrazovce:
?XPO%
Odpověď, je-li pozice vektoru na obrazovce rovna 1:
1

TP Display Functions > Get X-Position

Kurzorové funkce

CUR% Zjištění a nastavení pozice kurzoru na obrazovce.
Závisí na transfokaci a délce vektoru.
Příklad nastavení pozice kurzoru na pozici 1:
CUR%=1
Příklad zjištění pozice kurzoru:
?CUR%
Odpověď, je-li kurzor v pozici 1:
1

**TP Cursor Functions > Set Cursor
Get Cursor**

REF% Zjištění a nastavení referenčního kurzoru na obrazovce.
Závisí na transfokaci a délce vektoru.
Příklad nastavení referenčního kurzoru na pozici 1:
REF%=1
Příklad zjištění pozice referenčního kurzoru:
?REF%
Odpověď, je-li referenční kurzor na pozici 1:
1

**TP Cursor Functions > Set Ref Cursor
Get Ref Cursor**

TRK% Posuv obou kurzorů o danou hodnotu, která závisí na transfokaci a délce vektoru.
Příklad posunu obou kurzorů o deset datových bodů vlevo:
TRK%=-10

TP Cursor Functions > Set Cursors Moving

Souborové funkce

Jména souborů:

"ALL"	pevné jméno udávající současný stav systému (nastavení + všechny existující vektory)
"SET"	pevné jméno udávající nastavení současného systému
"CHx"	řetězec "CH" následován znakem 'A' nebo 'B' identifikuje jeden kanál
"TRn"	řetězec "TR" následován číslicí v rozsahu '1' ... '4' identifikuje jeden vektor
"IEEE"	pevné jméno udávající rozhraní IEEE
"R5232"	pevné jméno udávající rozhraní RS232
"Ann"	znak "A" následován dvěma číslicemi v rozsahu "00" ... "99" identifikuje jeden 'ALL'-soubor na disku E
"Mnn"	znak "M" následován dvěma číslicemi v rozsahu "00" ... "99" identifikuje jeden 'TRn'-soubor na disku E
"Snn"	znak "S" následován dvěma číslicemi v rozsahu "00" ... "99" identifikuje jeden 'SET'-soubor na disku E

Skalární uzly (funkce - FUn) NEJSOU přístupné jako soubory, ale pouze přes funkce TRA%, TRA! atd.

Platná jména souborů pro KILL: "Ann" "Mnn" "Snn" TRn"

Platná jména souborů pro COPY jsou uvedena v následující tabulce (+ označuje platnou kombinaci, - označuje neplatnou kombinaci):

z:	ALL	SET	CHx	TRn	IEEE	R523 2	Ann	Mnn	Snn
do:									
ALL	-	-	-	-	+	+	+	-	-
SET	-	-	-	-	+	+	-	-	+
CHx	-	-	-	-	-	-	-	-	-
TRn	-	-	-	-	-	-	-	-	-
IEEE	+	+	+	+	-	-	+	+	+
RS232	+	+	+	+	-	-	+	+	+
Ann	+	-	-	-	+	+	+	-	-
Mnn	-	-	+	+	+	+	-	+	-
Snn	-	+	-	-	+	+	-	-	+

Pozor, příkaz COPY kontroluje syntaxi dat, proto **není možné** kopírovat na disk E libovolná data. Pro kopírování do souboru Mnn se vyžaduje, aby vstupující data měla správnou syntaxi vektorového souboru.

COPY Kopíruje zdrojový datový soubor nebo data na rozhraní do přijímacího datového souboru nebo posílá data na rozhraní.
Příklad kopírování uzlu TR1 do souboru M01 na E-disku
COPY "TR1" TO "M01"
COPY "TR1" "M01"

TP	File Functions >	Copy
		Save to file
		Get data from

KILL Vymazání datového souboru.
Příklad vymazání kopie uzlu TR1 (soubor M01) z E-disku
KILL "M01"

TP	File Functions >	Erase
-----------	----------------------------	--------------

CPF\$ Zjištění a nastavení formátu kopírovaných souborů.
Přípustné hodnoty jsou: ASCII_HEX | BINARY
Formát souborů na E-disku je normální binární (tak jak jsou jednotlivé byty čteny z paměti).

Pokus kopírovat takové soubory přes sériové rozhraní se zapnutým 'xon/xoff' protokolem vede k potížím, neboť všechny byty s hodnotami 17 a 19 (dekadicky) jsou interpretovány jako řídicí znaky 'handshake'. Abychom zdolali tento problém, sériové rozhraní musí použít ASCII kódované písmena a číslice. Nevýhodou této strategie je větší počet bytů při přenosu. Abychom umožnili uživateli určit jeho správný formát, zavedli jsme proměnnou CPF\$. V "BINARY" módu je veškeré kopírování přes rozhraní RS232 a IEEE prováděno s minimálním počtem bytů a maximální rychlostí. Tento mód je vhodný pro IEEE přenosy a sériové přenosy používající pouze modemové řízení. Přenos je ukončen EOI na IEEE a počtem bloků a bytů na RS232. Pro sériové přenosy používající protokol 'xon/xoff' je využitelný "ASCII_HEX" mód, při kterém každý přenášený byte je ASCII znak v rozsahu '0'...'9' a 'A'...'F'. Přenos je zakončen ^Z (90 dekadicky).

Příklad nastavení formátu kopírovaných souborů na ASCII:

CPF\$="ASCII_HEX"

Příklad zjištění formátu kopírovaných souborů:

?CPF\$

Odpověď , je-li formát kopírovaných souborů binární:

BINARY

TP	File Functions >	Set Format
		Get Format

Funkce rozhraní IEEE

ADD% Zjištění a nastavení adresy IEEE.
Přípustné hodnoty jsou: 0 ... 30
Příklad nastavení adresy IEEE na hodnotu 8:

ADD%=8

Příklad zjištění adresy IEEE:

?ADD%

Odpověď , je-li adresa IEEE rovna 8:

8

TP	IEEE Functions >	Set IEEE Address
		Get IEEE Address

TLM\$ Zjištění a nastavení módu IEEE.
Přípustné hodnoty jsou: TO | TL | LO
Příklad nastavení módu IEEE na 'talk only':
TLM\$="TO"
Příklad nastavení módu IEEE na 'listen only':
TLM\$="LO"
Příklad zjištění módu IEEE:
?TLM\$
Odpověď , je-li mód IEEE 'talk/listen':
TL

**TP IEEE Functions > Set IEEE Mode
Get IEEE Mode**

Funkce sériového rozhraní

BAU\$ Zjištění a nastavení rychlosti přenosu sériového rozhraní.
Přípustné hodnoty jsou: 50 | 75 | 110 | 150 | 300 | 600 | 1200 | 2400 | 4800 | 9600
Příklad nastavení rychlosti přenosu sériového rozhraní:
BAU\$="9600"
Příklad zjištění rychlosti přenosu sériového rozhraní:
?BAU\$
Odpověď může být:
9600

**TP RS232 Functions > Set Baud Rate
Get Baud Rate**

DAT\$ Zjištění a nastavení počtu datových bitů.
Přípustné hodnoty jsou: 7 | 8
Příklad nastavení počtu datových bitů na 7:
DAT\$="7"
Příklad zjištění počtu datových bitů:
?DAT\$
Odpověď může být:
7

**TP RS232 Functions > Set Data Bits
Get Data Bits**

MCT% Zjištění a nastavení řízení sériového rozhraní.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení řízení sériového rozhraní (zapnuto):
MCT%=1
Příklad zjištění řízení sériového rozhraní:
?MCT%
Odpověď může být:
1

**TP RS232 Functions > Set Handshake
Get Handshake**

PAR\$ Zjištění a nastavení parity.
Přípustné hodnoty jsou: ODD | EVEN | NO
Příklad nastavení parity na lichou paritu:
PAR\$="ODD"
Příklad zjištění parity:
?PAR\$
Odpověď může být:
ODD

**TP RS232 Functions > Set Parity
Get Parity**

STO\$ Zjištění a nastavení počtu stop bitů.
Přípustné hodnoty jsou: 1 | 2
Příklad nastavení počtu stop bitů na 1:
STO\$="1"
Příklad zjištění počtu stop bitů:
?STO\$
Odpověď může být:
1

**TP RS232 Functions > Set Stop Bits
Get Stop Bits**

XON% Zjištění a nastavení 'xon/xoff' protokolu.
Přípustné hodnoty jsou: 0 nebo 1
Příklad nastavení 'xon/xoff' protokolu (zapnuto):
XON%=1
Příklad zjištění 'xon/xoff' protokolu:
?XON%
Odpověď může být:
1

**TP RS232 Functions > Set Protocol
Get Protocol**

Funkce plotru

PIN\$ Zjištění a nastavení rozhraní plotru.
Přípustné hodnoty jsou: IEEE | RS232
Příklad nastavení rozhraní plotru:
PIN\$="IEEE"
Příklad zjištění rozhraní plotru:
?PIN\$
Odpověď může být:
IEEE

**TP Plot Functions > Set Plot Interface
Get Plot Interface**

PLA\$ Zjištění a nastavení jazyku plotru.
 Přípustné hodnoty jsou: HPGL | STANDARD
 Příklad nastavení jazyku plotru:
 PLA\$="HPGL"
 Příklad zjištění jazyku plotru:
 ?PLA\$
 Odpověď může být:
 HPGL

TP	Plot Functions >	Set Plot Language Get Plot Language
-----------	----------------------------	--

PLT% Zjištění a nastavení stavu *Výpis obrazovky* plotru.
 Přípustné hodnoty jsou: 0 nebo 1
 Příklad nastavení stavu *Výpis obrazovky* plotru (kopie obrazovky):
 PLT%=1
 Příklad zjištění stavu *Výpis obrazovky* plotru:
 ?PLT%
 Odpověď, je-li stav *Výpis obrazovky* plotru nastaven:
 1

TP	Plot Functions >	Start Plot Stop Plot Get Plot
-----------	----------------------------	--

PSI\$ Zjištění a nastavení velikosti papíru plotru.
 Přípustné hodnoty jsou: A3 | A4
 Příklad nastavení velikosti papíru plotru:
 PSI\$="A3"
 Příklad zjištění velikosti papíru plotru:
 ?PSI\$
 Odpověď může být:
 A3

TP	Plot Functions >	Set Plot Size Get Plot Size
-----------	----------------------------	--

Časové funkce

Příkaz `RTS$="STOP"` uloží aktuální systémový čas a datum nebo čas a datum vektorového uzlu do vnitřního zásobníku. Potom můžete zjišťovat jednotlivé položky pomocí příkazů `?HOU%`, `?MIN%`, ... nebo nastavovat jednotlivé položky příkazy `HOU%=12`, `MIN%=40`, ... Upravený čas a datum se pak zapíše do systému nebo vektorového uzlu příkazem `RTS$="ENTER"`. Pokud upravujete pouze části času a data bez předchozího příkazu `RTS$="STOP"`, zbývající položky zůstanou nedefinované. Čas kanálových resp. skalárních uzlů se mění po každém záznamu resp. výpočtu.

Následující posloupnost nastaví čas a datum systémových hodin na 26.10.1987, 12:45:00:

```
CEN%=19:YEA%=87:MON%=10:DAY%=26:HOU%=12:MIN%=45:SEC%=0
RTS$("SYS")="ENTER"
```

Následující posloupnost kopíruje čas a datum ze systémových hodin do všech vektorových uzlů:

```
RTS$("SYS")="STOP":RTS$("TR1")="ENTER":RTS$("TR2")="ENTER"
RTS$("TR3")="ENTER":RTS$("TR4")="ENTER"
```

Poslední posloupnost zjistí čas záznamu 1. vektorového uzlu:

```
RTS$("TR1")="STOP"
```


?CEN%,YEA%,MON%,DAY%,HOU%,MIN%,SEC%

TP	Time Functions >	Set Real Time
		Set Real Time N
		Get Real Time

CEN%

Zjištění a nastavení století.

Přípustné hodnoty jsou: 0 ... 99

Příklad nastavení století pro příští příkaz RTS\$="ENTER":

CEN%=1

Příklad zjištění století z posledního příkazu RTS\$="STOP":

?CEN%

Odpověď může být:

1

TP	Time Functions >	Set Century
		Get Century

DAY%

Zjištění a nastavení dne.

Přípustné hodnoty jsou: 1 ... 31

Příklad nastavení dne pro příští příkaz RTS\$="ENTER":

DAY%=1

Příklad zjištění dne z posledního příkazu RTS\$="STOP":

?DAY%

Odpověď může být:

1

TP	Time Functions >	Set Day
		Get Day

HOU%

Zjištění a nastavení hodiny.

Přípustné hodnoty jsou: 0 ... 23

Příklad nastavení hodiny pro příští příkaz RTS\$="ENTER":

HOU%=1

Příklad zjištění hodiny z posledního příkazu RTS\$="STOP":

?HOU%

Odpověď může být:

1

TP	Time Functions >	Set Hour
		Get Hour

MIN%

Zjištění a nastavení minut.

Přípustné hodnoty jsou: 0 ... 59

Příklad nastavení minut pro příští příkaz RTS\$="ENTER":

MIN%=1

Příklad zjištění minut z posledního příkazu RTS\$="STOP":

?MIN%

Odpověď může být:

1

TP	Time Functions >	Set Minute
		Get Minute

MON% Zjištění a nastavení měsíce.
 Přípustné hodnoty jsou: 1 ... 12
 Příklad nastavení měsíce pro příští příkaz RTSS\$="ENTER":
 MON%=1
 Příklad zjištění měsíce z posledního příkazu RTSS\$="STOP":
 ?MON%
 Odpověď může být:
 1

TP	Time Functions >	Set Month
		Get Month

SEC% Zjištění a nastavení sekund.
 Přípustné hodnoty jsou: 0 ... 59
 Příklad nastavení sekund pro příští příkaz RTSS\$="ENTER":
 SEC%=1
 Příklad zjištění sekund z posledního příkazu RTSS\$="STOP":
 ?SEC%
 Odpověď může být:
 1

TP	Time Functions >	Set Second
		Get Second

YEA% Zjištění a nastavení roku.
 Přípustné hodnoty jsou: 0 ... 99
 Příklad nastavení roku pro příští příkaz RTSS\$="ENTER":
 YEA%=1
 Příklad zjištění roku z posledního příkazu RTSS\$="STOP":
 ?YEA%
 Odpověď může být:
 1

TP	Time Functions >	Set Year
		Get Year

Funkce uživatelské komunikace

MES\$ Nastavení řádky se zprávou.
 Přípustné hodnoty jsou:
 Libovolný ASCII řetězec s maximálně 44 tisknutelnými znaky.
 Prvním indexem se volí řádek. K dispozici jsou dva řádky, 1 a 2. Hodnota 0 znamená smazání obou řádek. Druhý index určuje atributy řádku:

- 0=normální reprezentace,
- 1=dvojnásobná výška, horní polovina zprávy,
- 2=dvojnásobná výška, dolní polovina zprávy,
- 3=dvojnásobná výška, horní polovina zprávy, inverzně,
- 4=dvojnásobná výška, dolní polovina zprávy, inverzně,
- 5=podtržení,
- 7=normální reprezentace, inverzně

Příklad nastavení první řádky se zprávou s normální reprezentací:

MES\$(1,0)="Press softkey 1 if ready"

Příklad nastavení první a druhé řádky se zprávou s dvojnásobnou výškou:

MES\$(1,1)="Press softkey 1 if ready"

MES\$(2,2)="Press softkey 1 if ready"

Příklad nastavení první řádky se zprávou s normální reprezentací, inverzně.

MES\$(1,7)="Press softkey 1 if ready"

TP User Communication > Set message line

MEV%

Zjištění a nastavení viditelnosti řádky se zprávou.

Přípustné hodnoty jsou: 0 nebo 1

Příklad nastavení viditelnosti řádky se zprávou (je vidět):

MEV%=1

Příklad zjištění viditelnosti řádky se zprávou:

?MEV%

Odpověď může být:

1

**TP User Communication > Set Msg Visible
Get Msg Visible**

RCS\$

Zjištění a nastavení volby potenciometrického řízení.

Přípustné hodnoty jsou: TRIG | DELAY | CURSOR | REFER | TRACK

Příklad nastavení volby potenciometrického řízení na kurzor:

RCS\$="CURSOR"

Příklad zjištění volby potenciometrického řízení:

?RCS\$

Odpověď, je-li volba potenciometrického řízení nastavena na kurzor:

CURSOR

**TP User Communication > Set Rotary
Get Rotary**

SKV%

Zjištění a nastavení viditelnosti funkčních klíčů.

Přípustné hodnoty jsou: 0 nebo 1

Příklad nastavení viditelnosti funkčních klíčů (je vidět):

SKV%=1

Příklad zjištění viditelnosti funkčních klíčů:

?SKV%

Odpověď, je-li viditelnost funkčních klíčů zapnuta:

1

**TP User Communication > Set Softkey Visible
Get Softkey Visible**

SKY\$

Nastavení popisu funkčních klíčů.

Přípustné hodnoty jsou:

Libovolný ASCII řetězec s maximálně 21 tisknutelnými znaky.

Prvním indexem se volí funkční klíč. K dispozici je 7 funkčních klíčů. Hodnota 0 znamená smazání všech funkčních klíčů. Druhý index určuje atributy klíče:

0=dvojnásobná výška,

1=dvojnásobná výška, inverzně

Příklad nastavení popisu prvního funkčního klíče s dvojnásobnou výškou:

SKY\$(1,0)="I am ready"

TP User Communication > Set key line

TIK% Zjištění stavu kláves na osciloskopu (pro synchronizaci akcí).
Přípustné hodnoty jsou v příloze *Kódy kláves*.
Příklad zjištění poslední stisknuté klávesy:
?TIK%
Odpověď, nebyla-li stisknuta žádná klávesa:
0

TP User Communication > Get Keyboard

Testovací funkce

TST% Spuštění a čtení testů.
Přípustné hodnoty jsou: ACQUCTRL | ACQUMEM | DISPMEM | INTEPROC |
RAM | ROM | RS232 | RTC
Možné odpovědi jsou: 0 nebo 1
Příklad spuštění a čtení testu systémové paměti:
?TST%("RAM")
Odpověď, prošla-li systémová paměť testem:
1

TP Test Functions > Get Test

NUL! Zjištění a nastavení reálné proměnné pro obecné využití.
Příklad nastavení obecně využitelné reálné proměnné:
NUL!=12345.678
Příklad čtení obecně využitelné reálné proměnné:
?NUL!
Odpověď může být:
1.2346e+04

**TP Test Functions > Set Tmp Real
Get Tmp Real**

NUL\$ Zjištění a nastavení obecně využitelné řetězcové proměnné.
Příklad nastavení obecně využitelné řetězcové proměnné:
NUL\$="null string"
Příklad zjištění obecně využitelné řetězcové proměnné:
?NUL\$
Odpověď může být:
null string

**TP Test Functions > Set Tmp String
Get Tmp String**

NUL% Zjištění a nastavení obecně využitelné dlouhé celočíselné proměnné.
Příklad nastavení obecně využitelné dlouhé celočíselné proměnné:
NUL%=987654321
Příklad zjištění obecně využitelné dlouhé celočíselné proměnné:
?NUL%
Odpověď může být:
987654321

**TP Test Functions > Set Tmp Long
Get Tmp Long**

LITERATURA

TRACE: Operating Manual, Programming Manual. Vienna, 1989

TestPoint: Quick Start. CEC, Billerica, USA, 1994

TestPoint: Techniques & Reference. CEC, Billerica, USA, 1994

TestPoint: Application Notes. CEC, Billerica, USA, 1994

DODATEK A: KÓDY KLÁVES

Pokud je celočíselná hodnota získaná příkazem TIK% interpretována hexadecimálně, ve vyšší polovině je kód třídy a v nižší polovině je kód funkce nebo hodnota.

Název klávesy	TIK%/100H	TIK% AND OFFH
DISPLAY #1	010H	001H
DISPLAY #2	010H	011H
DISPLAY #3	010H	002H
DISPLAY #4	010H	012H
DISPLAY 1vs2	010H	021H
DISPLAY 2vs4	010H	022H
DISPLAY INTERP	010H	031H
DISPLAY GRAT	010H	041H
DISPLAY X-ZOOM	010H	032H
DISPLAY SEPAR	010H	042H
INPUT A-ONLY	010H	003H
INPUT BWL	010H	013Fi
INPUT MIN/MAX	010H	023H
INPUT AUTOSET	010H	033H
INPUT AVG	010H	043H
VERTICAL CHA COUPL	010H	004H
VERTICAL CHB COUPL	010H	014H
VERTICAL CHA OFFSET	009H	signed value
VERTICAL CHB OFFSET	029H	signed value
VERTICAL CHA ATTEN	00AH	signed value
VERTICAL CHB ATTEN	049H	signed value
TRIGGER HYSTERESIS	010H	054H
TRIGGER SOURCE	010H	064H
TRIGGER SLOPE	010H	053H
TRIGGER EXT COUPL	010H	063H
TIME MAX MEM	010H	024H
TIME WRITE/LOCK	010H	044H
TIME MODE	010H	034H
TIME RELEASE	010H	074H
TIME TIMEBASE	069H	signed value
MEASURE SELECT	010H	051H
MEMORY SELECT	010H	071H
SETTING SELECT	010H	025H
SETTING PLOT	010H	035H
CONTROLS TRIG	010H	073H
CONTROLS DELAY	010H	056H
CONTROLS CURSOR	010H	017H
CONTROLS REFER	010H	027H
CONTROLS TRACK	010H	007H
CONTROLS ENCODER	091H	signed value
SOFT F1	010H	075H
SOFT F2	010H	045H
SOFT F3	010H	065H
SOFT F4	010H	066H

SOFT F5
SOFT F6
RETURN

010H
010H
010H

005H
046H
006H

DODATEK B: FORMÁTY SOUBORŮ

FORMÁT SOUBORU VEKTORU

<hlavička souboru> <blok uzlu> [<blok parametrů> <blok dat> ...]

Proměnná	Typ	Bytů	Hodnota
Hlavička souboru:			
délka bloku	unsigned word	2	10
typ bloku=File ID	unsigned word	2	5a81h
počet bloků v souboru	unsigned word	2	-
verze software	unsigned word	2	112h
konec bloku	unsigned word	2	a55ah
Blok uzlu:			
délka bloku	unsigned word	2	24
typ bloku	unsigned word	2	5a01h
samples per file	unsigned word	2	-
versions of file	unsigned word	2	1
data first index	unsigned word	2	0
data actual index	unsigned word	2	0
data wrap flag	unsigned byte	1	-
(spare)		1	0
record century	BCD coded byte	1	-
record year	BCD coded byte	1	-
record month	BCD coded byte	1	-
record day	BCD coded byte	1	-
record hour	BCD coded byte	1	-
record minute	BCD coded byte	1	-
record second	BCD coded byte	1	-
(spare)		1	0
konec bloku	unsigned word	2	a55ah
Blok parametrů:			
délka bloku	unsigned word	2	118
typ bloku	unsigned word	2	5a02h
(spare)		2	0
(spare)		2	0
data type	unsigned word	2	0
addr axis scale type	unsigned word	2	0
addr axis lsb value	real	4	-
addr axis offset value	real	4	-
addr axis dimension V exp.	signed word	2	0
add axis dimension m exp.	signed word	2	0
addr axis dimension s exp.	signed word	2	1
addr axis dimension A exp.	signed word	2	0
(spare)		16	
addr axis dimension code	unsigned byte	1	-
(spare)		15	
value axis scale type	unsigned word	2	0
value axis lsb value	real	4	-

value axis offset value	real	4	-
value axis dimension V exp.	signed word	2	-
value axis dimension m exp.	signed word	2	-
value axis dimension s exp.	signed word	2	-
value axis dimension A exp.	signed word	2	-
(spare)		16	
value axis dimension code	unsigned byte	1	-
(spare)		15	
value axis overflow code	unsigned byte	1	-
trigger mode code	unsigned byte	1	-
trigger level	unsigned word	2	-
(spare)	unsigned word	2	
konec bloku	unsigned word	2	a55ah

Blok dat:

délka bloku	unsigned word	2	2xsamples/file+6
typ bloku	unsigned word	2	5a03h
data	unsigned words	2xN	-
konec bloku	unsigned word	2	a55ah
kontrolní součet	unsigned word	2	

Kontrolní součet (checksum) je 16-ti bitová suma (mod 10000h) všech bytů zapsaných do souboru včetně všech hlavičkových, typových, datových a koncových záznamů, ale s výjimkou samotné kontrolní sumy.

N je počet vzorků v souboru.

FORMÁT SOUBORU NASTAVENÍ

<hlavička souboru> <blok kanálů> <blok vektorů> <blok funkcí> <blok zobrazení>

Proměnná	Typ	Bytů	Hodnota
Hlavička souboru:			
délka bloku	unsigned word	2	10
typ bloku=File ID	unsigned word	2	5a82h
počet bloků v souboru	unsigned word	2	-
verze software	unsigned word	2	112h
konec bloku	unsigned word	2	a55ah
Blok kanálů:			
délka bloku	unsigned word	2	30
typ bloku	unsigned word	2	5a11h
trigger level	signed word	2	-
trigger slope	boolean byte	1	-
trigger source	unsigned byte	1	-
trigger mode	unsigned byte	1	-
autotrigger	boolean byte	1	-
coupling a	unsigned byte	1	-
coupling b	unsigned byte	1	-

coupling ext	unsigned byte	1	-
attenu a	unsigned byte	1	-
attenu b	unsigned byte	1	-
offset a	unsigned byte	1	-
offset b	unsigned byte	1	-
timebase	unsigned byte	1	-
recording mode	unsigned byte	1	-
maximum memory active	unsigned byte	1	-
delay length	unsigned word	2	-
glitch	unsigned byte	1	-
chanlink (A-only)	unsigned byte	1	-
bandwidth	boolean byte	1	-
average	boolean byte	1	-
average mode	signed byte	1	-
average number	unsigned byte	1	-
konec bloku	unsigned word	2	a55ah

Blok vektorů:

délka bloku	unsigned word	2	26
typ bloku	unsigned word	2	5a12h
trace # 1 origin code	byte	1	-
trace # 1 operand 1 typ	byte	1	-
trace # 1 operand 1 index	byte	1	-
trace # 1 operand 2 typ	byte	1	-
trace # 1 operand 2 index	byte	1	-
trace # 2 origin code	byte	1	-
trace # 2 operand 1 typ	byte	1	-
trace # 2 operand 1 index	byte	1	-
trace # 2 operand 2 typ	byte	1	-
trace # 2 operand 2 index	byte	1	-
trace # 3 origin code	byte	1	-
trace # 3 operand 1 typ	byte	1	-
trace # 3 operand 1 index	byte	1	-
trace # 3 operand 2 typ	byte	1	-
trace # 3 operand 2 index	byte	1	-
trace # 4 origin code	byte	1	-
trace # 4 operand 1 typ	byte	1	-
trace # 4 operand 1 index	byte	1	-
trace # 4 operand 2 typ	byte	1	-
trace # 4 operand 2 index	byte	1	-
konec bloku	unsigned word	1	a55ah

Blok funkcí:

délka bloku	unsigned word	2	26
typ bloku	unsigned word	2	5a13h
scalar # 1 origin code	byte	1	-
scalar # 1 operand 1 typ	byte	1	-
scalar # 1 operand 1 index	byte	1	-
scalar # 1 operand 2 typ	byte	1	-
scalar # 1 operand 2 index	byte	1	-
scalar # 2 origin code	byte	1	-
scalar # 2 operand 1 typ	byte	1	-

scalar # 2 operand 1 index	byte	1	-
scalar # 2 operand 2 typ	byte	1	-
scalar # 2 operand 2 index	byte	1	-
scalar # 3 origin code	byte	1	-
scalar # 3 operand 1 typ	byte	1	-
scalar # 3 operand 1 index	byte	1	-
scalar # 3 operand 2 typ	byte	1	-
scalar # 3 operand 2 index	byte	1	-
scalar # 4 origin code	byte	1	-
scalar # 4 operand 1 typ	byte	1	-
scalar # 4 operand 1 index	byte	1	-
scalar # 4 operand 2 typ	byte	1	-
scalar # 4 operand 2 index	byte	1	-
konec bloku	unsigned word	2	a55ah

Blok zobrazení:

délka bloku	unsigned word	2	32
typ bloku	unsigned word	2	5a14h
interpol	unsigned word	2	-
x-zoom	unsigned word	2	-
x-position	unsigned dword	4	-
y-seperation	unsigned word	2	-
xy12	unsigned word	2	-
xy34	unsigned word	2	-
cursor position	unsigned dword	4	-
reference position	unsigned dword	4	-
graticule	boolean word	2	-
rotary select	unsigned word	2	-
konec bloku	unsigned word	2	a55ah
kontrolní součet	unsigned word	2	-

Kontrolní součet (checksum) je 16-ti bitová suma (mod 10000h) všech bytů zapsaných do souboru včetně všech hlavičkových, typových, datových a koncových záznamů, ale s výjimkou samotné kontrolní sumy.

FORMÁT SOUBORU NASTAVENÍ A VŠECH STOP

<hlavička souboru> <blok kanálů> <blok vektorů> <blok funkcí> <blok zobrazení> [<blok uzlu>
<blok parametrů> <blok dat>]...

[] ... pro každý vektor definovaný v bloku vektorů

Proměnná	Typ	Bytů	Hodnota
Hlavička souboru:			
délka bloku	unsigned word	2	-
typ bloku=File ID	unsigned word	2	5a83h
počet bloků v souboru	unsigned word	2	-
verze software	unsigned word	?	112h
konec bloku	unsigned word	2	a55ah

DODATEK C: KÓDY UŽÍVANÉ V SOUBORECH

KÓDY PŮVODNÍCH DEFINIC UZLŮ

Vymazání definice:	00	OFF
Přiřazení definice:	01	EQU (" <code><zdrojový uzel></code> ")
Derivace vektoru:	16	DIF (" <code><zdrojový uzel></code> ",sf)
Integrace vektoru:	15	INT (" <code><zdrojový uzel></code> ",sf)
Negace vektoru:	17	NEG (" <code><zdrojový uzel></code> ")
Vyhlazení vektoru:	18	SMU (" <code><zdrojový uzel></code> ",sf)
Sečtení vektorů:	11	ADD (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
Vydělení vektorů:	14	DIV (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
Vynásobení vektorů:	13	MUL (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
Odečtení vektorů:	12	SUB (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
Amplituda mezi kurzorem a referencí:	22	CRA (" <code><zdrojový uzel></code> ")
Amplituda mezi kurzorem a nulou :	20	CZA (" <code><zdrojový uzel></code> ")
50% amplituda vektoru:	29	FIA (" <code><zdrojový uzel></code> ")
Maximální amplituda vektoru:	23	MAA (" <code><zdrojový uzel></code> ")
Minimální amplituda vektoru:	24	MIA (" <code><zdrojový uzel></code> ")
Mezivrcholová amplituda vektoru:	25	PPA (" <code><zdrojový uzel></code> ")
Amplituda mezi 10% a 90% vektoru:	30	RIA (" <code><zdrojový uzel></code> ")
Amplituda mezi referencí a nulou:	21	RZA (" <code><zdrojový uzel></code> ")
Stejnosečná složka vektoru:	26	DCA (" <code><zdrojový uzel></code> ")
Střední hodnota vektoru:	28	MEA (" <code><zdrojový uzel></code> ")
Root mean square amplitude of trace:	27	RMS (" <code><zdrojový uzel></code> ")
Amplituda mezi kurzorem a referencí:	39	CRI (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
Amplituda mezi kurzorem a referencí:	40	CRO (" <code><source nede></code> ",&code><zdrojový uzel>")
RMS mezi vektory; lineárně:	37	RMI (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
RMS mezi vektory; logaritmicky:	38	RMO (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
Doba mezi kurzorem a referencí:	31	CRT (" <code><zdrojový uzel></code> ")
Doba mezi kurzorem a triggerem:	32	CTT (" <code><zdrojový uzel></code> ")
Doba mezi referencí a triggerem:	33	RTT (" <code><zdrojový uzel></code> ")
Náběžná hrana vektoru; z 10% na 90%:	34	RIT (" <code><zdrojový uzel></code> ")
Frekvence vektoru:	36	FRQ (" <code><zdrojový uzel></code> ")
Perioda vektoru:	35	PER (" <code><zdrojový uzel></code> ")
Fázový rozdíl vektorů; v čase:	41	PHT (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")
Fázový rozdíl vektorů; ve stupních:	42	PHD (" <code><zdrojový uzel></code> ",&code><zdrojový uzel>")

KÓDY TYPŮ OPERANDŮ

1=Operand je kanál:	index 0 = kanál A index 1 = kanál B
2=Operand je vektor:	index 0 =vektor #1 ... index 3 = vektor #4
3=Operand je soubor:	index 0 = M00 ... index 99 = M99
4=Operand is konstanta:	hodnota indexu = konstanta (např. DIF(...,10)

KÓDY INTERPOLACÍ

0 = OFF
1 = LINEAR
2 = SINE
3 = PULSE

KÓDY TRANSFOKACE

0 = *0.05
1 = *0.1
2 = *1
3 = *10

KÓDY VOLEB POTENCIOMETRICKÉHO ŘÍZENÍ

0 = TRIG
1 = DELAY
2 = CURSOR
3 = REFER
4 = TRACK

DODATEK D: OVLADAČE PRO TESTPOINT

Disketa přiložená ke zprávě zahrnuje kromě programu pro načtení datových souborů z osciloskopu do počítače po sériové lince (popis programu uveden na disketě) také ovladače pro TestPoint a ukázkové programy pro TestPoint (popis taktéž uveden na disketě).

Na disketě jsou obsaženy dva druhy ovladačů pro TestPoint:

- **TRACECOM.TST** - ovladač obsahující jednotlivé příkazy pro komunikaci a nastavení osciloskopu a
- **TRACEWIN.TST** - ovladač, který má příbuzné funkce integrovány do samostatných panelů TestPointu, (vyvoláním jediné funkce se rozbalí panel s prvky pro ovládání dané skupiny funkcí osciloskopu, prvky v panelu jsou nastaveny na aktuální hodnoty).

Dále následuje menu jednotlivých příkazů z ovladače **TRACECOM** :

Set Remote	
Set Local	
Nodes >	Set Node Get Node
Exception >	Get Exception Msg Get Exception Code
Line Separators >	Set Input Sep RS232 Get Input Sep RS232 Set Output Sep RS232 Get Output Sep RS232 Set Input Sep IEEE Get Input Sep IEEE Set Output Sep IEEE Get Output Sep IEEE
Identity >	Get Type Get Version Get Serial #
Request Codes >	Set Command Req Get Command Rqs Set Except Req Get Except Rqs Set Data Req Get Data Rqs Set Trace Req Get Trace Rqs

Vertical Functions >	Set Atten Code Set Atten Code N Get Atten Code Set Attenuation Set Attenuation N Get Attenuation Set ChCoupling Set ChCoupling N Get ChCoupling Set Offset Code Set Offset Code N Get Offset Code Set Offset Set Offset N Get Offset Get Probe Factor
Horizontal Functions >	Set Delay Code Set Delay Code N Get Delay Code Set Delay Set Delay N Get Delay Set Sample Code Set Sample Code N Get Sample Code Set Sample Set Sample N Get Sample Set Points Set Points N Get Points
Definition Functions >	Get Definition Set Definition

Data Functions >	Get Data Int Get Data Real Set Point Set Data Int Set Data Int N Set Data Real Set Data Real N
Channel Functions >	Set A only Get A only Set Avg Count Get Avg Count Set Average Get Average Set Avg Mode Get Avg Mode Set BWL Get BWL Set Min/Max Get Min/Max Set Max Memory Get Max Memory
Sequence Functions >	Autocal Set Rec Mode Get Rec Mode Set Release Get Release Autoset Set Data Ready Get Data Ready Set Start/Stop Get Start/Stop

Trigger Functions >	Set Autotrigger Get Autotrigger Set Trg Coupling Get Trg Coupling Set Trg Level Set Trg Level N Get Trg Level Set Trg Hyster Get Trg Hyster Set Trg Source Get Trg Source Set Trg Slope Get Trg Slope
Display Functions >	Set Display 12 Get Display 12 Set Display 34 Get Display 34 Set Interpolation Get Interpolation Set X-Zoom Get X-Zoom Set Grid Get Grid Set Separation Get Separation Get X-Position
Cursor Functions >	Set Cursor Get Cursor Set Ref Cursor Get Ref Cursor Set Cursors Moving
File Functions >	Erase Copy Save to file Get data from Set Format Get Format

IEEE Functions >	Set IEEE Address Get IEEE Address Set IEEE Mode Get IEEE Mode
RS232 Functions >	Set Baud Rate Get Baud Rate Set Data Bits Get Data Bits Set Parity Get Parity Set Stop Bits Get Stop Bits Set Protocol Get Protocol Set Handshake Get Handshake
Time Functions >	Set Real Time Set Real Time N Get Real Time Set Century Get Century Set Year Get Year Set Month Get Month Set Day Get Day Set Hour Get Hour Set Minute Get Minute Set Second Get Second

Plot Functions >	Set Plot Interface Get Plot Interface Set Plot Language Get Plot Language Set Plot Size Get Plot Size Start Plot Stop Plot Get Plot
User Communication >	Set message line Set Msg Visible Get Msg Visible Set key line Set Softkey Visible Get Softkey Visible Set Rotary Get Rotary Get Keyboard
Test Functions >	Get Test Set Tmp Real Get Tmp Real Set Tmp Long Get Tmp Long Set Tmp String Get Tmp String

Dále následuje menu jednotlivých příkazů z ovladače **TRACEWIN** :

Show Node Selection
Show Line Separator
Show Identity
Show Request Codes
Show Vertical Functions
Show Horizontal Functions
Show Channel Functions
Show Sequence Functions
Show Trigger Functions
Show Display Functions
Show Cursor Functions
Show File Functions
Show IEEE Functions
Show RS232 Functions
Show Plot Functions
Show Time Functions
Show User Communication
Show Test Functions
Show Definition Functions