

PROGRAMOVATELNÝ KALKULÁTOR

ET-57



uživatelská příručka

Programovatelný kalkulátor ET-57

uživatelská příručka

Verze kalkulátoru 201101

listopad 2020



webové stránky: <http://www.breatharian.eu/hw/et57/>

OBSAH

1. Mapa rozložení klávesnice	6
2. Charakteristika	8
3. Popis	9
4. Jak používat kalkulátor	10
5. Odchytky od TI-57	11
6. Formát čísel	12
7. Klávesnice	14
8. Indikátory na displeji	15
9. Editor čísla	16
10. Číselné výrazy	17
11. Adresování registrů	18
12. Programování	22
13. Externí zařízení a porty	24
14. Tlačítka a instrukce	29
00 ... 09 Základní číslice, 0...9	29
10 Vypnutí kalkulátoru a kontrast displeje, OFF	29
11 Alternativní funkce, 2nd	30
12 Inverze funkce, INV	30
13 Přirozený logaritmus a exponent, $\ln x$	31
14 Oprava chyby, CE	31
15 Vymazání displeje, CLR	32
18 Dekadický logaritmus a exponent, \log	32
19 Vymazání registru T, C.t.	33
20 Tangens, \tan	33
21 Programování, LRN	34
22 Záměna registrů X a T, $x \leftrightarrow t$	34
23 Druhá mocnina čísla, x^2	34
24 Druhá odmocnina čísla, \sqrt{x}	34
25 Převrácená hodnota čísla, $1/x$	35
26 Převody minut a sekund, D.MS	35
27 Převod kartézských a polárních souřadnic, P \rightarrow R	36
28 Sinus, \sin	37
29 Kosinus, \cos	37

30	Ludolfovo číslo, pi.....	38
31	Krok programu vpřed, SST.....	38
32	Uložení čísla do datového registru, STO.....	38
33	Vyvolání čísla z datového registru, RCL	39
34	Přičtení a odečtení čísla od datového registru, SUM	39
35	Mocnina a odmocnina, y^x	39
36	Prodleva, Pause	40
37	Vložení prázdného bajtu do programu, Ins	40
38	Záměna čísla s datovým registrem, Exc	40
39	Vynásobení a vydělení datového registru, Prd	41
40	Absolutní hodnota, IxI	41
41	Krok programu zpět, BST	42
42	Režim exponentu, EE.....	42
43	Levá závorka, (.....	43
44	Pravá závorka,)	43
45	Dělení, :	43
46	Žádná operace, Nop.....	44
47	Zrušení bajtu z programu, Del.....	44
48	Zaokrouhlení, Fix.....	44
49	Celé číslo, Int.....	45
50	Stupně, Deg.....	46
51	Skok, GTO	46
55	Násobení, x	47
56	Programová smyčka, Dsz	47
57	Nepřímé uložení čísla do registru, STO*	48
58	Nepřímé vyvolání obsahu registru, RCL*	49
59	Nepřímé přičtení a odečtení čísla z registru, SUM*	49
60	Radiány, Rad.....	50
61	Podprogram, SBR	50
65	Odečtení, -	51
66	Rovno, $x=t$	51
67	Nepřímá inkrementace/dekrementace registru, Inc*	51
68	Nepřímá záměna čísla s datovým registrem, Exc*	52
69	Nepřímé násobení a dělení registru, Prd*	52
70	Grady, Grad.....	52
71	Reset, RST	53
75	Přičtení, +	53
76	Větší nebo rovno, $x \geq t$	53
77	Inkrementace a dekrementace registru, Inc.....	54
78	Výběr programového prostoru, Pgm	54
79	Generátor náhodného čísla, Rand	55
80	Rozptyl, Var	56
81	Start a stop programu, R/S.....	57
83	Desetinná tečka,	57

84	Změna znaménka, +/-	57
85	Provedení výpočtu, =	58
86	Návěští, Lbl	58
87	Faktoriál, x!	59
88	Statistika, Stat	59
89	Průměr, Mean	61
15.	Příklady programů	62
1.	Hod kostkou	62
2.	Hod kostkou 2	63
3.	Světelný LED had	64
4.	Vyčíslení polynomu	66
5.	Komplexní čísla	68
6.	Ramanujanova aproximace faktoriálu x!	73
7.	Stirlingova aproximace faktoriálu $\ln(x!)$	75
8.	Vyhledání průchodů funkce nulou	78
9.	Simpsonův integrál funkce	83
10.	Lineární regresní přímka	86

1. Mapa rozložení klávesnice

U každého tlačítka je uveden na 1. řádku základní význam a na 2. řádku alternativní význam (po stisku tlačítka **2nd**).

[11] 2nd []	[12] INV []	[13] lnx [18] log	[14] CE [19] C.t	[15] CLR [10] OFF
[21] LRN [26] D.MS	[22] x<>t [27] P->R	[23] x^2 [28] sin	[24] Vx [29] cos	[25] 1/x [20] tan
[31] SST [36] Pause	[32] STO [37] Ins	[33] RCL [38] Exc	[34] SUM [39] Prd	[35] y^x [30] pi
[41] BST [46] Nop	[42] EE [47] Del	[43] ([48] Fix	[44]) [49] Int	[45] : [40] x
[51] GTO [56] Dsz	[07] 7 [57] STO*	[08] 8 [58] RCL*	[09] 9 [59] SUM*	[55] x [50] Deg
[61] SBR [66] x=t	[04] 4 [67] Inc*	[05] 5 [68] Exc*	[06] 6 [69] Prd*	[65] = [60] Rad
[71] RST [76] x>=t	[01] 1 [77] Inc	[02] 2 [78] Pgm	[03] 3 [79] Rand	[75] + [70] Grad
[81] R/S [86] Lb 	[00] 0 [87] x!	[83] . [88] Stat	[84] +/- [89] Mean	[85] = [80] Var

ET-57

2nd ●	INV ●	lnx ● log	CE ● C.t	CLR ●
LRN ● D.MS	x\rightarrowt ● P\rightarrowR	x² ● sin	\sqrt{x} ● cos	1/x ● tan
SST ● Pause	STO ● Ins	RCL ● Exc	SUM ● Prd	y^x ● π
BST ● Nop	EE ● Del	(● Fix) ● Int	\div ● x
GTO ● Dsz	7 ● STO*	8 ● RCL*	9 ● SUM*	\times ● Deg
SBR ● x=t	4 ● Inc*	5 ● Exc*	6 ● Prd*	- ● Rad
RST ● x\geqt	1 ● Inc	2 ● Pgm	3 ● Rand	+ ● Grad
R/S ● Lbl	0 ● x!	. ● Stat	+/- ● Mean	= ● Var

2. Charakteristika

Souhrn: Přesnost výpočtů 17 číslic mantisy (BCD kód), zobrazeno 11 číslic mantisy. 80 registrů (v RAM), 500 programových kroků (v EEPROM), procesor ATmega8, LCD displej.

- Procesor ATmega8 (8MHz, 8KB ROM, 1KB RAM, 512B EEPROM)
- Napájecí napětí 5 V (z USB nabíječky nebo z USB portu)
- Výpočty v BCD kódu
- Přesnost výpočtů 17 číslic
- Přesnost základních registrů 15 číslic
- Přesnost rozšířených registrů 13 číslic
- Zobrazení údaje na 11 platných číslic
- Exponent 2 číslice, rozsah +- 99
- 10 programových prostorů
- Každý prostor 50 programových kroků (celkem 500 kroků)
- Uživatelský program uchován v EEPROM (bez baterie)
- 10 základních registrů (přístupných přímým adresováním)
- 70 rozšířených registrů (přístupných indexovým adresováním)
- Dvouřádkový LCD displej (2 x 16 alfanumerických znaků)
- 40 tlačítek
- Ovládání externího zařízení přes ISP konektor
- Kód kalkulátoru kompletně napsaný v AVR assembleru
- Exponenciální a logaritmické funkce
- Trigonometrické funkce
- Faktoriál
- Generátor náhodných čísel
- Indexový přístup k proměnným
- Vědecký mód zobrazení s exponentem
- Statistická funkce
- 10 návěští v každém programovém prostoru
- Volání funkcí a skoky mezi programovými prostory
- Hardware i software plně open source

3. Popis

Kalkulátor ET-57 vychází koncepčně z populárního kalkulátoru TI-57, vyvinutého v roce 1977 firmou Texas Instruments. Snaží se zachovat zpětnou kompatibilitu programů pro TI-57 a přitom funkčnost rozšířit s využitím možností použitého procesoru. Funkčnost rozšiřuje o více programových prostorů (10 prostorů, celkem 500 kroků), více datových registrů (80), přímé i nepřímé adresování, faktoriál, generátor náhodných čísel.

Kalkulátor je určen bývalým uživatelům kalkulátoru TI-57, zájemcům o retro techniku a jako výuková pomůcka k seznámení s principy použití a programování kalkulátoru. Z toho důvodu se snaží o maximální konstrukční zjednodušení sestávající pouze z mikropsínačů, procesoru, LCD displeje, konektoru pro externí napájení a pár drobných součástek. Slouží spíše k experimentování a výuce, při stolním použití s externím napájením z USB nabíječky nebo z USB portu.

Kromě základní varianty ET-57 s procesorem ATmega8 je kalkulátor k dispozici i ve variantě pro hardware kalkulátoru ET-58, s procesory ATmega88, ATmega168 nebo ATmega328. Tato varianta je označena ET-57B a liší se od základní varianty tím, že umožňuje vypnutí kalkulátoru (tlačítko OFF) a řízení kontrastu LCD displeje, avšak nenabízí přístup na externí zařízení (kalkulátor ET-58 neobsahuje ISP konektor).

4. Jak používat kalkulačtor

Kalkulačtor ET-57 je opatřen 2-řádkovým alfanumerickým LCD displejem, 40 mikropínači a procesorem.

Vzhledem k tomu, že kalkulačtor není napájen z baterie (je určen ke stolnímu použití s externím napájením z USB nabíječky nebo z USB portu), neobsahuje vypínač. Uživatelský program se ukládá do paměti EEPROM, jejíž obsah je uchován i bez napájení. Odpojením napájení se kalkulačtor resetuje, vymažou se registry, obsah displeje i započaté operace, jenom obsah uživatelského programu (v EEPROM) zůstane zachován.

Po připojení napájení se na displeji kalkulačtoru na 1 sekundu zobrazí název kalkulačtoru spolu s 6-místným kódem, představujícím datum verze firmware kalkulačtoru. Např. "ET-57 201101" znamená datum firmware (build) 1.11.2020.

Kalkulačtor ve variantě ET-57B (přeprogramovaný procesor kalkulačtoru ET-58) lze vypnout stiskem tlačítek **2nd CLR** (funkce **OFF**) a zapnout stiskem **CLR**. Stiskem **INV 2nd CLR** (funkce **LCD**) následovaným číslicí **0** až **9** lze řídit kontrast displeje.

5. Odchylyky od TI-57

Přestože se software kalkulátoru ET-57 snaží o maximální kompatibilitu s původním kalkulátorem TI-57, mohou se projevit odchyly a může být nutné některé programy při importu upravovat. Zde jsou uvedené známé odchyly, se kterými může být potřeba počítat.

Vyšší přesnost





Původní kalkulátor TI-57 pracuje s interní přesností 11 číslic (11 BCD číslic mantisa, 2 číslice exponent, 1 číslice znaménko, celkem 7 bajtů) a zobrazuje max. 8 číslic mantisy. Kalkulátor ET-57 počítá interně s přesností mantisy 17 číslic (10 bajtů na číslo). Výsledek výpočtu ukládá do základních registrů (tj. registry R0..R9 a X) s přesností 15 číslic (9 bajtů). Uložením do rozšířeného registru (R10..R79) se údaj zaokrouhlí na 13 číslic (8 bajtů).

Původní kalkulátor TI-57 používá k výpočtům funkcí metodu CORDIC, která umožňuje poměrně rychlé a snadné výpočty s využitím jen základních operací (posun, sčítání, odčítání) a tabulkových hodnot. Metoda CORDIC se používá u kalkulaček a interně v procesorech. Naproti tomu kalkulátor ET-57 používá k výpočtům Taylorovu řadu, která je pro použití typ procesoru vhodnější.

Důsledkem výše uvedeného počítá původní TI-57 funkce s přesností 9 až 10 číslic, další 1 až 2 číslice mantisy zahrnují nepřesnosti výpočtu. ET-57 počítá funkce s přesností 15 číslic (interně počítá na 17 číslic, uložením do registru se výsledek zaokrouhlí na platných 15 číslic).

Vyšší přesnost obvykle není na závadu, může se projevit např. u generátoru náhodných čísel nebo při porovnání výsledků kalkulátorů.

Opakování výpočtu

Kalkulátor ET-57 po stisku klávesy  opakuje naposledy zadanou aritmetickou operaci, zatímco u původní TI-57 způsobí opakovaný stisk klávesy  indikaci chyby, což některé programy využívají k indikaci chyby. V takových případech je potřeba zajistit indikaci chyby jiným způsobem, např. posloupností  .

6. Formát čísel

U původního kalkulátoru TI-57 je číslo uloženo v registrech o velikosti 14 BCD číslic D13 až D0. Jedna číslice může nabývat hodnot 0 až 9. Dvě nejnižší číslice, D1 a D0, obsahují exponent bez znaménka, v rozsahu 00 až 99. Vyšších 11 číslic, D12 až D2, obsahuje číslice mantisy. Mantisu je vždy zarovnána doleva tak, aby číslice D12 neobsahovala nulu. Nejvyšší číslice D13 obsahuje znaménkové příznaky. Bit 0 indikuje zápornou mantisu, bit 1 záporný exponent a bit 2 invertovanou mantisu (příznak přenosu z nejvyšší číslice, mantisa v záporném tvaru). Tento způsob interpretace čísla využívá podporu procesoru pro BCD operace.

Kalkulátor ET-57 používá také BCD interpretaci mantisy čísla. BCD formát zajistí vhodnější zaokrouhlování výsledků pro lidskou interpretaci. Např. číslo 0.1 se v BCD kódu uchová jako číslice '1' s exponentem -1, bez ztráty přesnosti. V binárním kódu by se takové číslo vyjádřilo s mantisou 4CCCC... (nekonečný počet číslic), kdy už samotným zapisem čísla vznikne malá chyba.

Mantisu není v kalkulátoru ET-57 uložena v absolutním tvaru s odděleným znaménkem (jako u původní kalkulačky), ale zachovává znaménkový tvar, s rozšířením do znaménkové nejvyšší číslice. Znaménková číslice obsahuje hodnotu buď 0 (indikující nezáporné číslo) nebo 9 (indikující záporné číslo). Negace čísla znamená "desítkový doplněk" číslic mantisy, neboli "devítkový doplněk" (=inverze) zvýšený o 1.

Exponent vyjadřuje opět desítkový exponent, ale je uložen v prvním bajtu čísla jako binární číslo s biasem 128. Exponent s hodnotou 0 (řád jedniček) je symbolizován binární hodnotou 128. Exponent 1 (desítky) má hodnotu 129, exponent -1 (desetiny) má hodnotu 127. Exponent má platný rozsah binárních hodnot 29 až 227, což odpovídá dekadickému exponentu -99 až +99. Kromě toho jsou používány ještě zvláštní 3 případy hodnot exponentu: 0 indikuje číslo nula, 28 indikuje podtečení do záporných exponentů (příliš malé číslo) a 228 přetečení do kladných exponentů (příliš velké číslo).

Kalkulátor používá 3 formáty čísla, lišící se přesností mantisy:

1) Během výpočtů se používají čísla 10 bajtů (přesnost 17 číslic). Prvním bajtem je exponent v binárním tvaru s biasem 128. Následujících 9 bajtů obsahuje mantisu ve znaménkovém tvaru, od vyšších číslic k nižším číslicím. To znamená 1 číslice znaménková a 17 významných číslic. Zobrazením mantisy v HEX tvaru se číslice zobrazí v lidsky čitelné formě,

jako číslice zleva doprava. Mantisa je normalizována tak, aby první (znaménková) číslice obsahovala 0 (nezáporné číslo) nebo 9 (záporné číslo). Druhá číslice (nejvyšší číslice mantisy) obsahuje číslo odlišné od znaménkové číslice - tj. mantisa je zarovnána doleva.

2) Základní registry (R0 až R9, a také registry X a LAST) obsadí 9 bajtů (přesnost 15 číslic). První bajt je exponent, následujících 8 bajtů obsahuje 15 číslic mantisy a 1 znaménkovou číslici.

3) Rozšířené registry (R10 až R79) obsadí 8 bajtů (přesnost 13 číslic), s mantisou 7 bajtů, tj. 13 číslic mantisy a 1 znaménková číslice.

Mantisu výsledku lze pro účely ladění zobrazit klávesami **INV** **+** (vypnutí s **INV** **0**). Zobrazí se 16 číslic registru X, bez exponentu.

Příklady čísel (v HEX tvaru, včetně exponentu s biasem 128):

3.14159265358979 -> 80 03 14 15 92 65 35 89 79

-3.14159265358979 -> 80 96 85 84 07 34 64 10 21

123.456 -> 82 01 23 45 60 00 00 00 00

K testu přesnosti kalkulátoru lze použít populární trigonometrický test:

9 **sin** **cos** **tan** **INV** **tan** **INV** **cos** **INV** **sin**

Při správném výpočtu by mělo být výsledkem opět číslo 9. Výpočtem se rychle ztrácí přesnost a u kalkulátorů se běžně objevuje odchylka. U původního kalkulátoru TI-57 je výsledek 9.0047464 (přesnost 3 číslice), u kalkulátoru ET-57 je výsledkem testu 8.9999999976 (přesnost 9 číslic).

Více k přesnosti kalkulátorů: <http://www.datamath.org/Forensics.htm>

7. Klávesnice

Kalkulátor lze provozovat buď v přímém (prováděcím) módu, kdy se kódy tlačítek ihned provádějí, nebo v programovacím módu, kdy se kódy tlačítek pouze zapisují do programu, ale neprovádí se.

Kalkulátor se ovládá sadou 40 tlačítek, rozložených v 8 řadách a 5 sloupcích. Řady jsou číslovány shora dolů, v pořadí 1 až 8. Sloupce jsou číslovány zleva doprava, s čísly 1 až 5. S tímto číslováním se kódy tlačítek ukládají do programu.

Po stisku tlačítka **2nd** se použije alternativní funkce tlačítka, indikovaná číslem sloupce 6 až 10 (číslo 10 je v kódu nahrazeno číslicí 0).

Při zápisu programu do paměti (tlačítkem **LRN**) se do programu zapíše kód stisknutého tlačítka jako dvojice číslic, kde první číslice představuje řádek tlačítka (typicky 1 až 9) a druhá číslice sloupec tlačítka (typicky 1 až 5 pro základní funkci nebo 6 až 0 pro alternativní funkci).

Kódy číselných tlačítek **0** až **9** se v programu neuchovávají pomocí souřadnice tlačítka, ale jako dekadická hodnota 00 až 09.

Poznámka: V textu příručky jsou názvy tlačítek uváděny bez případných prefixů **2nd**, které mohou být nutné k vyvolání funkce tlačítka. Např. kód tlačítka **Rand** (náhodné číslo) se vyvolá stiskem tlačítek **2nd****3**.

8. Indikátory na displeji

LCD displej obsahuje 2 řádky po 16 alfanumerických znacích. První řádek je využit k zobrazení indikátorů, druhý řádek k zobrazení zadávaného čísla a výsledku operace.



Deg/Rad/Grd - indikace jednotky úhlů ve stupních, radiánech nebo gradech. $360^\circ = 2 \cdot \pi$ radiánů = 400 gradů. Přepínač lze změnit tlačítky **Deg**, **Rad** nebo **Grad**.

Fix0 až Fix8 - indikuje zvolené zaokrouhlování čísel 0 až 8 desetinných míst. Nastavuje se tlačítky **Fix 0** až **Fix 8**. Posloupnost **INV Fix** (stejný význam má též **Fix 9**) vypne zaokrouhlování zobrazeného výsledku. V tomto případě se zaokrouhlení na displeji neindikuje (nahradí se mezerami).

EE - indikuje režim exponentu. Po stisku **EE** se číslo zobrazí ve vědeckém tvaru, ve formátu mantisa a exponent. Režim lze zrušit stiskem **CLR** nebo **INV EE**. Je-li mód exponentu vypnut, neindikuje se na displeji nic.

2nd - indikuje stisk tlačítka alternativních funkcí **2nd**. Je-li po jednom stisku **2nd** stisknuto nějaké tlačítko, provede se namísto jeho základní funkce jeho alternativní funkce (na klávesnici zobrazena v druhém řádku). Není-li tlačítko **2nd** stisknuto, nebo je-li stisknuto 2x, není alternativní funkce aktivní, provede se základní funkce tlačítka. Základní stav není na displeji indikován (na pozici se zobrazí mezery).

INV - indikace stisknutí tlačítka **INV**, aktivujícího inverzní funkci.

Operace - poslední pozice 1. řádku je určena k indikaci aktivní aritmetické operace: + sčítání, - odčítání, * násobení, : dělení, \ modulo, ^ mocnina, V odmocnina, (otevřená závorka.

9. Editor čísla

Zadávané číslo, stejně jako výsledky výpočtů, se zobrazí na 2. řádku displeje. Mantisa se zobrazí s přesností max. 11 číslic.

Před mantisou je rezervována 1 pozice pro znaménko. Na tomto místě se pro záporná čísla zobrazí '-', pro kladná čísla se ponechá mezera.

Za mantisou se zobrazí exponent (je-li aktivní režim s exponentem). Exponent je od mantisy oddělen znaménkem + nebo -. Exponent se zobrazí na 2 číslice.

Součástí mantisy je desetinná tečka. V módu exponentu ve vědecké notaci (mantisa a exponent) se desetinná tečka zobrazí vždy za první číslicí. Není-li aktivní mód exponentu, zobrazí se desetinná tečka za číslicí jednotek.

Tlačítko **CE** maže poslední znak mantisy nebo exponentu (podle toho kam právě probíhá zápis číslic).

Tlačítko **EE** zahájí zadávání exponentu. K zadávání mantisy se lze vrátit stiskem tlačítka tečky **.** nebo **INV EE**. Tlačítko **EE** slouží též k zahájení editace zobrazeného výsledku operace. To lze využít k odstranění skrytých číslic čísla.

Příklad - zarovnání čísla na 4 číslice:

PI [3.1415926536] ... příklad čísla s desetinnými místy

Fix 4 [3.1416] ... zobrazení se zaokrouhlí na 4 desetinná místa

EE [3.1416+00] ... zahájení editace, oříznutí skrytých číslic

INV EE [3.1416] ... vypnutí módu s exponentem

INV Fix [3.1416] ... vypnutí zaokrouhlení, výsledek je stále 3.1416

10. Číselné výrazy

Kalkulátor během výpočtů udržuje prioritu operací ve 3 stupních:

1. + sčítání, - odečítání
2. * násobení, : dělení, \ modulo
3. ^ mocnina, $\sqrt{}$ odmocnina

Při výpočtech se nejdříve vyhodnotí úroveň 3) mocnina a odmocnina, poté 2) násobení a dělení, a nakonec 1) sčítání a odčítání.

Ve výrazu lze libovolně používat závorky, a to až do 7. úrovně.

Po provedení výpočtu lze opětovným stiskem tlačítka $\boxed{=}$ opakovat výpočet nejnižší úrovně. Zadáním čísla a stiskem $\boxed{=}$ se zopakuje operace. Zadané číslo se použije jako první operand operace, druhý operand zůstane zachován původní.

Poznámka: Výpočty se interně provádí s přesností 17 číslic. Uložením do registru výsledku X se výsledek zaokrouhlí na 15 číslic.

Příklad:

$$\boxed{3} \boxed{+} \boxed{2} \boxed{=} [5]$$

$$\boxed{4} \boxed{=} [6]$$

$$\boxed{1} \boxed{0} \boxed{=} [12]$$

$$\boxed{1} \boxed{0} \boxed{+} \boxed{2} \boxed{*} \boxed{3} \boxed{y^x} \boxed{4} \boxed{=} [172] \dots \text{výpočet výrazu } 10 + (2 * (3^4)) = 172$$

11. Adresování registrů

Kalkulátor obsahuje 10 základních registrů (označených R0 až R9) a 70 rozšiřujících registrů (označených R10 až R79), celkem to je 80 datových registrů (R0 až R79).

Základní registry R0 až R9 mají přesnost mantisy 15 číslic. Uložením výsledku operace do základního registru se přesnost údaje nesníží. Základní registry se používají jako hlavní pracovní registry. Adresují se přímým adresováním, pomocí instrukcí **STO**, **RCL**, **SUM**, **Exc**, **Prd** a **Inc**, za kterými se uvede index registru 0 až 9. Základní registry jsou adresovány též u inverzních operací **INV SUM**, **INV Prd** a **INV Inc**.

Instrukce **INV STO**, **INV RCL** a **INV Exc** mají stejnou funkci jako instrukce bez **INV**, ovšem namísto základních registrů R0 až R9 adresují rozšířené registry R10 až R19.

Většina základních registrů má ještě doplňkovou funkci:

R0 ... počet položek N (statistika Stat), čítač smyčky Dsz

R1 ... součet y (statistika Stat)

R2 ... součet y^2 (statistika Stat)

R3 ... součet x (statistika Stat)

R4 ... součet x^2 (statistika Stat)

R5 ... součet $x \cdot y$ (statistika Stat)

R6

R7 ... registr T

R8 ... indexový registr pro nepřímé adresování

R9 ... alternativní indexový registr pro nepřímé adresování

Rozšiřující datové registry R10 až R79 mají sníženou přesnost 13 číslic. Jejich použitím se uložený údaj zkrátí o 2 číslice. Rozšiřující registry nelze adresovat přímým adresováním, je nutné použít nepřímé adresování - instrukce **STO***, **RCL***, **SUM***, **Exc***, **Prd*** a **Inc***. Při nepřímém adresování není číslo registru součástí instrukce, ale načte se z registru R8 (registr pro nepřímé adresování). Podobně jsou adresovány i inverzní operace **INV SUM***, **INV Prd*** a **INV Inc***.

Instrukce **INV STO***, **INV RCL*** a **INV Exc*** mají stejnou funkci jako instrukce bez **INV**, ovšem namísto indexového registru R8 se použije

alternativní indexový registr R9.

Poznámka: *Nepřímým adresováním lze adresovat též základní registry R0 až R9.*

Příklad, převrácení pořadí obsahu registrů:

1) Zaplnění registrů R10 až R79 číslem 0 až 69

Pgm 1 ... přepnutí na programový prostor 1 (a reset ukazatele)

LRN ... aktivace programovacího módu

00 **Lbl 1** ... návěští 1 (podprogram 1 = zaplnění registrů)

01 **70 STO 0** ... příprava čítače registrů (70 registrů)

04 **9 STO 8** ... příprava indexového registru - 1 (ukazatel na R10-1)

06 **CLR** ... [0] příprava ukládané hodnoty X = 0

07 **Lbl 9** ... návěští začátku smyčky

08 **Inc 8** ... inkrementace indexového registru R8

09 **STO*** ... uložení hodnoty X do registru indexovaného s R8

10 **+ 1 =** ... inkrementace údaje na displeji (registr X)

13 **Dsz** ... dekrementace R0 a přeskočení následující instrukce při 0

14 **GTO 9** ... pokračování smyčky pokud R0 není ještě 0

15 **INV SBR** ... konec programu

2) Reverze obsahu registrů (R10<->R79, R11<->R78,...)

16 **Lbl 2** ... návěští 2 (podprogram 2 = reverze registrů)

17 **35 STO 0** ... příprava čítače registrů (70 registrů / 2 = 35)

20 **9 STO 8** ... příprava prvního indexového registru - 1

22 **80 STO 9** ... příprava druhého indexového registru + 1

25 **Lbl 8** ... návěští začátku smyčky

- 26 **Inc 8** ... inkrementace prvního indexového registru R8
- 27 **INV Inc 9** ... dekrementace druhého indexového registru R9
- 28 **RCL*** ... načtení hodnoty registru indexovaného registrem R8
- 29 **INV Exc*** ... záměna s obsahem registru indexovaného R9
- 30 **STO*** ... uložení hodnoty do registru indexovaného R8
- 31 **Dsz** ... dekrementace R0 a přeskočení následující instrukce při 0
- 32 **GTO 8** ... pokračování smyčky pokud R0 není ještě 0
- 33 **INV SBR** ... konec programu

3) Kontrolní zobrazení obsahu registrů R10 až R79

- 34 **Lbi 3** ... návěští 3 (podprogram 3 = zobrazení registrů)
- 35 **7 0 STO 0** ... příprava čítače registrů (70 registrů)
- 38 **9 STO 8** ... příprava indexového registru - 1 (ukazatel na R10-1)
- 40 **Lbi 7** ... návěští začátku smyčky
- 41 **Inc 8** ... inkrementace indexového registru R8
- 42 **RCL*** ... načtení hodnoty z registru indexovaného s R8
- 43 **Pause** ... pauza k zobrazení údaje
- 44 **Dsz** ... dekrementace R0 a přeskočení následující instrukce při 0
- 45 **GTO 7** ... pokračování smyčky pokud R0 není ještě 0
- 46 **INV SBR** ... konec programu

4) Test programu

- LRN** ... vypnutí programovacího módu
- SBR 1** ... zaplnění registrů R10 až R79 číslem 0 až 69
- INV RCL 9 [9]** ... zobrazení obsahu registru R19, tj. číslo 9
- SBR 3** ... kontrola obsahu registrů, zobrazí se čísla 0 až 69

SBR **2** ... reverze obsahu registrů

INV **RCL** **9** [60] ... zobrazení obsahu registru R19, tj. číslo 60

SBR **3** ... kontrola obsahu registrů, zobrazí se čísla 69 až 0

12. Programování

Zápis posloupnosti tlačítek do programové paměti nazýváme programem. Programem se z kalkulátoru stává mocný nástroj. Programová paměť sestává z 10 nezávislých programových prostorů, přepínaných instrukcí **Pgm** s parametrem **0** až **9**. Každý programový prostor obsahuje 50 programových kroků, celkem je tedy k dispozici 500 programových kroků. V každém programovém prostoru lze použít 10 návěstí **Lbl**, číslovaných 0 až 9. Mezi programovými prostory lze volat podprogramy nebo provádět skoky použitím instrukce **Pgm** v programu - instrukce nepřepne programový prostor trvale, ale pouze pro jeden následující příkaz **GTO** nebo **SBR**.

Programy jsou uloženy v paměti EEPROM procesoru, jejíž obsah je zachován i po odpojení napájení kalkulačky.

Režim programování se zahájí tlačítkem **LRN**. Obsah programu se zobrazuje na dvou řádcích displeje. Na spodním řádku zleva je uveden aktuální programový prostor (Pgm0 až Pgm9), za kterým následuje aktuální ukazatel v programu, tj. adresa 00 až 49. Za adresou následuje číselný kód instrukce. Kód instrukce sestává ze 2 číslic. První číslice představuje řádek s klávesou 1 až 8, druhá číslice je sloupec s klávesou 1 až 5 nebo u alternativní funkce sloupec 6 až 0. Číselné klávesy se zobrazí s kódem 00 až 09. Za kódem instrukce může následovat parametr 0 až 9. Před kódem instrukce může být uveden znak mínus '-' znamenající inverzní funkci **INV**. Na horním řádku je uveden textový formát instrukce.



Klávesy během programování

SST (Single Step) - Zvýší ukazatel programu o 1 ("další krok"). Tlačítko **SST** lze používat i během normálního (prováděcího) režimu. Po jeho stisku se provede kód instrukce, na kterou je nastaven ukazatel programu.

BST (Back Step) - Sníží ukazatel programu o 1 ("zpětný krok").

Ins (Insert) - Vloží na aktuální pozici programu prázdnou instrukci **Nop** a zbytek programu odsune.

Del (Delete) - Zruší bajt na aktuální pozici programu a přisune následující část programu.

LRN (Learn) - Ukončí režim editace programu a navrátí kalkulač do prováděcího režimu.

GTO (Go To) - Tlačítko **GTO** se běžně používá k přesunu ukazatele programu na zadané návěští 0 až 9. V tomto režimu nelze instrukci použít k přesunu ukazatele během režimu programování, protože by se příslušný kód uložil do programu. Je nutné ukončit režim programování stiskem **LRN**, provést skok **GTO** na zadané návěští **0** až **9** a stiskem **LRN** se vrátit do režimu programování.

Kromě skoku na návěští umožňuje tlačítko **GTO** provést skok na absolutní adresu v programu. Skok se provede stiskem **INV GTO**, za kterým následují 2 číslice adresy 00 až 49. Instrukci skoku na absolutní adresu nelze uložit do programu, slouží pouze k přesunu ukazatele během programování a proto může být zadána jak v prováděcím režimu, tak i během programování,

Poznámka: Skok na absolutní adresu se u ET-57 provádí jiným způsobem (s jiným pořadím kláves) než tomu bylo u původní TI-57. U původní TI-57 se stiskla nejdříve klávesa **GTO**, potom **INV** a nakonec 2 číslice adresy. U ET-57 je nutné nejdříve stisknout **INV**, poté **GTO** a nakonec 2 číslice adresy.

RST (Reset) - Podobně jako **GTO** nelze použít přímo v programovacím módu, ale lze využít v prováděcím módu k převinutí ukazatele programu na adresu 00 v aktuálním programovém prostoru.

R/S (Run/Stop) - Start programu nebo zastavení programu (použije se v prováděcím módu).

13. Externí zařízení a porty

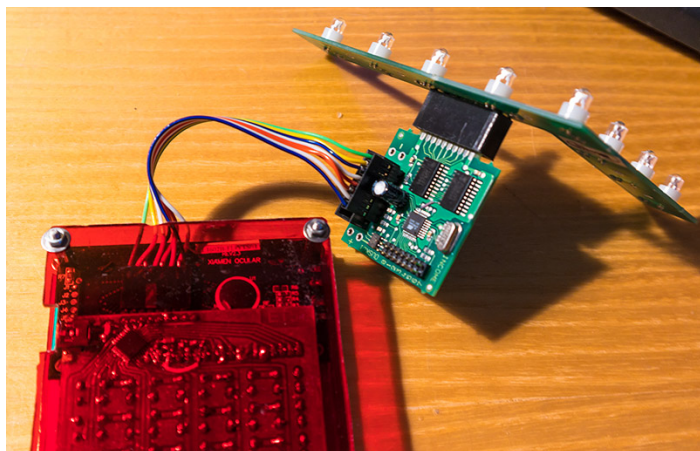


Kalkulátor ET-57 umožňuje připojení externího zařízení pomocí ISP konektoru, který jinak slouží k programování procesoru kalkulátoru. Komunikace probíhá pomocí SPI protokolu. ISP konektor je 8-pinový konektor KONPC-SPK-8 s následujícím obsazením pinů:

- 1 SCK** (sériové hodiny, vysílané kalkulaátorem)
- 2 MISO** (vstup dat do kalkulátoru, výstup dat ze zařízení)
- 3 MOSI** (výstup dat z kalkulátoru, vstup dat do zařízení)
- 4** orientační klíč, pin chybí, v konektoru zaslepen aby nešel zasunout
- 5 /RESET** (reset procesoru během ISP programování)
- 6 GND** (zem, 0V)
- 7 VCC** (napájení, +5V)
- 8** nepoužito, v kalkulátoru připojeno na +5V, ale v budoucnu tak nemusí být nebo může být využito pro SS signál.

Propojovací kabel propojí stejné piny v kalkulátoru i zařízení (neprovádí se křížení signálů).

Poznámka: Signál SS na straně slave je nutné spojit s GND.



Zařízení lze ovládat pomocí adresovatelného 256-bajtového pole portů. Do registru R8 kalkulátoru se uloží záporné číslo -1 až -256 představující adresu portu 0 až 255. Instrukcí **STO*** se na zvolený port odešle číslo 0 až 255. Naopak instrukcí **RCL*** se načte hodnota ze zvoleného portu jako číslo 0 až 255.

Podobně lze použít i instrukce **INV STO*** a **INV RCL***, u kterých se namísto registru R8 použije registr R9 s adresou portu. Jiné instrukce než **STO*** a **RCL*** neumožňují přistupovat na porty zařízení.

Není-li zařízení připojeno nebo v jiném případě chyby komunikace, se při použití instrukcí **STO*** a **RCL*** program zastaví a indikuje chybu (blikající 'E'). U instrukcí **INV STO*** a **INV RCL*** se chyba komunikace ignoruje, program chybu neindikuje a pokračuje v činnosti. Zpravidla se v průběhu ladění programu použijí nejdříve instrukce **STO*** a **RCL***, které zajistí indikaci problémů s komunikací. Po odladění programu se použijí instrukce s **INV** které zajistí, že program pojede dlouhodobě nepřerušeně, a to i v případech krátkodobých výpadků zařízení.

Komunikace probíhá pomocí SPI komunikačního protokolu, s rychlostí hodin 250 kHz. Kalkulátor vystupuje v roli master (řídící jednotka), externí zařízení je slave (podřízená jednotka). Master (kalkulátor) vysílá na linku MOSI sériová data, od vyššího bitu k nižšímu. Na linku SCK vysílá hodinový signál, se vzorkováním dat v náběžné vstoupné hraně.

Současně slave (zařízení) vysílá zpět data po lince MISO. Přenos jednoho bajtu (8 bitů) trvá 32 us. Master po odeslání každého bajtu doplní prodlevu 10 us, aby měl slave čas na vyhodnocení přijatého bajtu.

Synchronizace komunikace mezi master a slave není řízena SS signálem, ale softwarově, resetováním příjmu na straně slave. Na začátku každého přenosu vysílá master synchronizační bajt 0x53 (písmeno 'S', bitově 01010011b). Není-li přenos synchronizován, jsou data na straně slave bitově posunutá a slave přijímá jinou hodnotu než 0x53. V takovém případě vyšle slave bajt 0x55 jako indikaci chyby synchronizace, resetuje spojení a po prodlevě 50 us inicializuje nové spojení. Přijme-li slave správný synchronizační bajt 0x53, odpoví stejnou hodnotou 0x53 a pokračuje v komunikaci.

Master vysílá synchronizační bajty 0x53 opakovaně. Přijme-li od slave správnou odpověď 0x53, pokračuje v komunikaci. Přijme-li bajt 0xff nebo 0x00, považuje to za příznak nepřípojeného zařízení a přeruší komunikaci s indikací chyby. V případech jiných odpovědí doplní prodlevu 100 us a opakuje pokus o synchronizaci.

Po úspěšné synchronizaci pokračuje master vysláním příkazu 0x52 (písmeno 'R') pro čtení portu ze zařízení nebo 0x57 (písmeno 'W') pro zápis portu do zařízení. Po příkazu vyšle master bajt s adresou portu 0 až 255. Po vyslání adresy vyšle master třetí bajt příkazu - data. V případě zápisu do portu vyšle data 0 až 255 k zápisu do portu. V případě čtení z portu vyšle master bajt 0xFF, současně s vysíláním bajtu slave zpětně odpovídá bajtem dat načtených z portu.

Ve všech ostatních případech, kdy není bajt od slave určený, odpovídá slave zpět opakováním bajtu přijatého od master (echo).

Příklad komunikace (master/slave):

- Master zapíše bajt 0x24 na port 0x01

0x53 / 0x?? ... master vyšle 5x SYNC povel pro synchronizaci,
odpověď od slave zpočátku nedefinovaná, ignoruje se

0x53 / 0x53 ... slave odpovídá správně 0x53, ale master odpověď
zatím ignoruje, protože se může jednat o zbytek předešlé
komunikace

0x53 / 0x53

0x53 / 0x53

0x53 / 0x53 ... master detekuje správnou odpověď 0x53, pokračuje

0x57 / 0x53 ... master vyšle povel pro zápis 0x57, slave pokračuje v echo

0x01 / 0x57 ... master vysílá adresu portu 0x01, slave detekoval příkaz
pro zápis, vysílá echo předešlého bajtu a načítá adresu
0x24 / 0x01 ... master vysílá data 0x24 k zápisu na port

- Master čte zpět data z portu 0x01

0x53 / 0x01 ... master opět vyšle 5x SYNC povel pro synchronizaci

0x53 0x53 0x53 0x53 / 0x53 0x53 0x53 0x53 ... slave připaven

0x52 / 0x53 ... master vyšle povel pro čtení 0x52

0x01 / 0x52 ... master vysílá adresu portu 0x01, slave připraví data
z portu k odeslání

0xFF / 0x24 ... master vysílá 0xFF a čte od slave data 0x24

Příklad zdrojového kódu pro obsluhu komunikace Slave:

```
u8 SPIData[256];          // SPI slave data
u8 SPINx;                 // index of SPI command
u8 SPICmd;                // 1st byte - SPI command
u8 SPIAddr;               // 2nd byte - SPI address

#define SPICMD_SYNC        0x53    // synchro command ('S')
#define SPICMD_READ        0x52    // read command ('R')
#define SPICMD_WRITE       0x57    // write command ('W')
#define SPICMD_ERR         0x55    // invalid synchronization ('U')

ISR(SPI_STC_vect)         // SPI interrupt
{
    u8 d = SPDR;           // read data byte
    if (SPINx == 0) // receive command
    {
        // check known commands
        if ( (d != SPICMD_SYNC) &&
              (d != SPICMD_READ) &&
              (d != SPICMD_WRITE))
        {
            // error, try to re-synchronize
            SPDR = SPICMD_ERR; // report error to master
            SPI_Term(); // terminate SPI
            _delay_us(50); // short delay 50 us
            SPI_SlaveInit(); // re-initialize SPI
            return;
        }

        // shift to next byte of the command
        if (d != SPICMD_SYNC)
```

```

        {
            SPICmd = d;
            SPINx = 1; // next index = address
        }
    }
    else if (SPINx == 1)    // receive address
    {
        SPIAddr = d; // save address
        SPINx = 2; // next index = data

        // read data
        if (SPICmd == SPICMD_READ) SPDR = SPIData[d];
    }
    else    // receive data
    {
        // next index = command
        SPINx = 0;

        // write data
        if (SPICmd == SPICMD_WRITE) SPIData[SPIAddr] = d;
    }
}

```

14. Tlačítka a instrukce

U každého tlačítka je uveden programový BCD kód, název tlačítka a posloupnost stisků tlačítek k jeho vyvolání.

00 ... 09 Základní číslice, 0...9


Symbol  ... 

Vyvolání tlačítka  ... 





Základní číslice slouží k zadání číslic v rozsahu 0 až 9. Používají se k zadání mantisy čísla, zadání exponentu, číslo paměťového registru, číslo návěští a jiné.






Číslice se do programu ukládají s kódem 00 až 09.




10 Vypnutí kalkulatoru a kontrast displeje, OFF

Symbol 

Vyvolání tlačítka  

Tlačítko  (přesněji posloupnost  ) slouží k vypnutí kalkulatoru. Opětovné zapnutí kalkulatoru je možné stiskem samotného tlačítka .





Uvedením prefixu  před instrukcí  lze nastavit kontrast displeje (funkce ). Instrukce si vyžádá jako parametr číselný kód  až . 0 nastaví nejmenší kontrast displeje (světlé písmo na světlém pozadí), 9 nastaví nejvyšší kontrast displeje (tmavé písmo na tmavém pozadí).


Funkce  a   je dostupná pouze u varianty kalkulatoru ET-57B (verze s baterií). Kalkulátor varianty ET-57 nelze vypnout tlačítkem ani řídit kontrast displeje.

11 Alternativní funkce, 2nd

Symbol 

Vyvolání tlačítkem 

Tlačítko  slouží ke změně významu následujícího tlačítka na alternativní funkci. Po stisku tlačítka  se provede alternativní funkce následujícího tlačítka. Druhým stiskem   se navrátí zpět k základním funkcím.


Kód tlačítka  se do programu neukládá. Vždy se ukládá alternativní kód následujícího tlačítka.

Příklad:




  [0.6931...] ... vypočte přirozený logaritmus čísla



   [0.3010...] ... dekadický logaritmus čísla (instrukce )

12 Inverze funkce, INV

Symbol 

Vyvolání tlačítkem 

Tlačítko , stisknuté před jiným kódem tlačítka, u mnoha tlačítek vyvolá jejich obrácenou funkci. V některých případech jde o doplňkovou alternativní funkci. Druhým stiskem   se navrátí zpět k základním funkcím.

Kód tlačítka  se do programu neukládá. Ukládá se jako příznak instrukce. V programu se indikuje znakem '-' před kódem instrukce. Některé instrukce prefix  neakceptují, do programu se uloží kód bez prefixu.

Příklad:

   [0.1736...] ... výpočet $\sin(10) = 0.1736...$

INV **sin** [10] ... výpočet arkus sinus, $\text{asin}(0.1736...) = 10$

13 Přirozený logaritmus a exponent, **lnx**

Symbol **lnx**

Vyvolání tlačítkem **lnx**

Tlačítkem **lnx** se vypočte přirozený logaritmus čísla na displeji. Přirozený logaritmus používá jako základ Eulerovu konstantu s hodnotou 2.718281828459. Stiskne-li se nejdříve tlačítko **INV**, provede se inverzní funkce, přirozený exponent.

Argumentem funkce **lnx** musí být kladné, nenulové číslo. V případě nuly se displej rozbliká s hodnotou -9.9999+99, jako příznak chyby. Pro záporné číslo se provede výpočet absolutní hodnoty čísla a displej se opět rozbliká s indikací chyby.

Argumentem funkce **INV** **lnx** může být jak kladné, tak i záporné číslo, v rozsahu zhruba -227 až +227. Číslo mimo tento rozsah způsobí přetečení údaje a indikaci chyby.

Příklad:

5 **lnx** [1.6094...] ... vypočte přirozený logaritmus čísla 5 (tj. 1.6094...)

INV **lnx** [5] ... vypočte přirozený exponent čísla na displeji (tj. 5)

14 Oprava chyby, **CE**

Symbol **CE**

Vyvolání tlačítkem **CE**

Tlačítkem **CE** lze zrušit indikaci chyby 'E', projevující se blikáním displeje.



Během editace čísla na displeji je tlačítkem **CE** smazán poslední znak zadávaného údaje. Probíhá-li editace mantisy, je smazán poslední znak

mantisy. Probíhá-li editace exponentu, je smazán poslední znak exponentu. Je-li mazán exponent s hodnotou 0, exponent se zruší a přejde se na editaci mantisy.


15 Vymazání displeje, CLR

Symbol 

Vyvolání tlačítkem 

Tlačítko  provede několik inicializačních operací. Resetuje započaté aritmetické operace, vynuluje indikaci chyby, vypne mód exponentu , vynuluje registr X a zahájí editaci nového čísla s výchozí hodnotou 0.



Tlačítko  nenuluje registr T ani datové registry.


U kalkulátoru varianty ET-57B slouží tlačítko  k zapnutí kalkulátoru.



18 Dekadický logaritmus a exponent, log

Symbol 

Vyvolání tlačítky  

Tlačítkem  se vypočte dekadický logaritmus čísla na displeji. Dekadický logaritmus používá jako základ číslo 10. Stiskne-li se nejdříve tlačítko , provede se inverzní funkce, dekadický exponent.

Argumentem funkce  musí být kladné, nenulové číslo. V případě nuly se displej rozblíká s hodnotou -9.9999+99, jako indikace chyby. Pro záporné číslo se provede výpočet absolutní hodnoty čísla a displej se opět rozblíká s indikací chyby.

Argumentem funkce   může být jak kladné, tak i záporné číslo, v rozsahu -99 až +99. Číslo mimo tento rozsah způsobí přetečení údaje a indikaci chyby.

Příklad:

5 **log** [.69897...] ... dekadický logaritmus čísla 5 (tj. 0.69897...)

INV **log** [5] ... dekadický exponent čísla na displeji (tj. 5)

19 Vymazání registru T, C.t

Symbol **C.t**

Vyvolání tlačítka **2nd** **CE**

Tlačítkem **C.t** lze vymazat registr T (tj. datový registr R7). Uvedením prefixu **INV** před tlačítkem se provede vymazání všech datových registrů R0 až R79 (včetně registru T).

20 Tangens, tan

Symbol **tan**

Vyvolání tlačítka **2nd** **1/x**

Funkce **tan** vypočte tangens úhlu. Úhel je zadán v jednotkách nastavených přepínači **Deg**, **Rad** nebo **Grad**.

Zadáním prefixu **INV** před instrukcí **tan** se provede opačná funkce - arkus tangens. Výsledkem je úhel v aktuálně nastavené úhlové míře.

Příklad:


5 **0** **tan** [1.19175...] ... tangens úhlu 50° je hodnota 1.19175...

INV **tan** [50] ... arkus tangens

21 Programování, LRN

Symbol 


Vyvolání tlačítkem 

Tlačítko  aktivuje nebo deaktivuje režim programování. Režim programování byl blíže popsán v sekci [Programování](#).

22 Záměna registrů X a T, x<>t


Symbol 

Vyvolání tlačítkem 


Tlačítkem  je možné zaměnit registry X a T. Registr X je pracovní registr neboli též obsah displeje. Registr T je pomocný registr (temporary), odpovídá registru R7. Slouží k porovnávání čísel a k převodu polárních a kartézských souřadnic.

Registr X je nulován tlačítkem . Registr T je nulován tlačítkem .

23 Druhá mocnina čísla, x^2

Symbol 

Vyvolání tlačítkem 

Tlačítkem  se vypočte druhá mocnina čísla, nebo-li násobek čísla samo se sebou.

24 Druhá odmocnina čísla, Vx

Symbol 

Vyvolání tlačítkem 

Tlačítkem **Vx** se vypočte druhá odmocnina čísla. Číslo nesmí být záporné. Je-li proveden výpočet záporného čísla, vypočte se odmocnina absolutní hodnoty čísla a nastaví se indikace chyby 'E' (displej bliká).

25 Převrácená hodnota čísla, 1/x

Symbol **1/x**

Vyvolání tlačítkem **1/x**

Tlačítkem **1/x** se vypočte převrácená hodnota čísla. Je-li číslem nula, zobrazí se hodnota 9.9999+99 a nastaví se indikace chyby 'E' (displej bliká). To se často používá v programech k zapnutí indikace chyby a k indikaci chybného běhu programu.

26 Převody minut a sekund, D.MS

Symbol **D.MS**

Vyvolání tlačítky **2nd** **LRN**

Instrukcí **D.MS** lze převést čas nebo úhel vyjádřený pomocí minut a sekund (vteřin) na desetinné číslo. Vstupem funkce je desetinné číslo DD.MMSS, mající na pozici celých čísel celý počet hodin či stupně, na prvních dvou desetinných místech je počet minut a na dalších dvou desetinných místech je počet sekund (vteřin). Desetinná místa sekund lze doplnit jako další desetinné číslice. Výstupem funkce je desetinné číslo představující počet hodin či stupně DD.DDDD, vyjádřených desetinným číslem.

Uvedením prefixu **INV** před instrukcí **D.MS** se provede opačná operace - čas nebo úhel vyjádřený pomocí desetinného čísla se převede na údaj minut a sekund (vteřin). Vstupem funkce je desetinné číslo představující počet hodin či stupně DD.DDDD. Výstupem je číslo DD.MMSS, mající na pozici celých čísel celý počet hodin či stupně, na prvních dvou desetinných místech je počet minut a na dalších dvou desetinných místech je počet sekund (vteřin). Není-li výsledkem celý počet sekund, jsou desetinná místa sekund doplněna jako další desetinné číslice.

Příklad, součet času:

12:30:23 ... čas je 12:30:23 (12 hodin, 30 minut a 23 sekund)

D.MS [12.50638...] ... převod na hodiny vyjádřené v desetínách

+ **3:45:12** **D.MS** [3.7533...] ... plus 3 hodiny, 45 minut a 12 sekund

= **INV D.MS** [16.1535] ... výsledný čas 16 hodin, 15 minut a 35 sekund

27 Převod kartézských a polárních souřadnic, P->R

Symbol **P->R**

Vyvolání tlačítka **2nd x<>t**

Tlačítkem **P->R** se převedou souřadnice z polárního vyjádření na souřadnice kartézské. Před operací obsahuje registr T (tj. pomocný registr R7) radius (poloměr) a registr X (obsah displeje) obsahuje úhel. Úhel se udává v aktuálně vybrané úhlové míře (tlačítka **Deg**, **Rad** a **Grad**). Po operaci obsahuje registr T (pomocný registr R7) souřadnici X, registr X (obsah displeje) obsahuje souřadnici Y.

Uvedením prefixu **INV** před instrukcí **P->R** se provede opačná operace, převod kartézských souřadnic na polární. Před operací obsahuje registr T souřadnici X, registr X (displej) obsahuje souřadnici Y. Po operaci obsahuje registr T radius (poloměr) a registr X (displej) obsahuje úhel. Úhel je udán v aktuálně vybrané úhlové míře.

Příklad.

10 x<>t ... zadání radiusu 10 do registru T

30 ... zadání úhlu 30° do registru X

P->R [5] ... převod polárních souřadnic na kartézské. Na displeji se zobrazí souřadnice Y = 5

x<>t [8.6602...] ... přepnutím X a T se zobrazí souřadnice X = 8.6602...

x<>t [5] ... přepnutí registrů X a T zpět

INV **P->R** [30] ... zpětný přepočít, zobrazí se úhel 30°

x<>t [10] ... zobrazení registru T s radiusem 10

28 Sinus, sin

Symbol **sin**

Vyvolání tlačítka **2nd** **x^2**

Funkce **sin** vypočte sinus úhlu. Úhel je zadán v jednotkách nastavených přepínači **Deg**, **Rad** nebo **Grad**.

Zadáním prefixu **INV** před instrukcí **sin** se provede opačná funkce - arkus sinus. Výsledkem je úhel v aktuálně nastavené úhlové míře.

Úhel vypočtený funkcí arkus sinus je v rozsahu -90° až +90°. Vstupní hodnota funkce arkus sinus musí být v rozsahu -1 až +1. Leží-li mimo uvedený rozsah, údaj se omezí do platného rozsahu a indikuje se chyba 'E' (displej bliká).

29 Kosinus, cos

Symbol **cos**

Vyvolání tlačítka **2nd** **Vx**

Funkce **cos** vypočte kosinus úhlu. Úhel je zadán v jednotkách nastavených přepínači **Deg**, **Rad** nebo **Grad**.

Zadáním prefixu **INV** před instrukcí **cos** se provede opačná funkce - arcus kosinus. Výsledkem je úhel v aktuálně nastavené úhlové míře.

Úhel vypočtený funkcí arkus kosinus je v rozsahu 0° až +180°. Vstupní hodnota funkce arkus kosinus musí být v rozsahu -1 až +1. Leží-li mimo uvedený rozsah, údaj se omezí do platného rozsahu a indikuje se chyba 'E' (displej bliká).

30 Ludolfovo číslo, pi

Symbol **pi**

Vyvolání tlačítka **2nd** **y^x**

Tlačítko **pi** slouží k zadání konstanty "Ludolfovo číslo pi", které má hodnotu 3.14159265358979.

31 Krok programu vpřed, SST

Symbol **SST**

Vyvolání tlačítkem **SST**

Tlačítkem **SST** (Single Step) se v režimu programování zvýší ukazatel adresy programu o 1.

V prováděcím režimu se provede 1 instrukce programu, čímž je možné program krokovat z důvodu ladění. Bude-li kombinováno krokování programu se spouštěním podprogramů, nemusí docházet ke správnému navracení z podprogramů (kalkulátor si nebude pamatovat návratovou adresu z podprogramu).

Bližší informace naleznete v kapitole [Programování](#).

32 Uložení čísla do datového registru, STO

Symbol **STO**

Vyvolání tlačítkem **STO**

Pomocí tlačítka **STO** (Store) lze uložit číslo na displeji do datového registru R0 až R9. Jako parametr instrukce se zadává číslo registru 0 až 9.

Uvedením prefixu **INV** před instrukcí **STO** se provede obdobná funkce, ale namísto do registrů R0 až R9 se číslo uloží do registrů R10 až R19. Registry R10 až R19 patří do skupiny rozšířených datových registrů se sníženou přesností 13 číslic. Z mantisy se uložením odstraní poslední 2

číslice.

33 Vyvolání čísla z datového registru, RCL

Symbol **RCL**

Vyvolání tlačítkem **RCL**

Pomocí tlačítka **RCL** (Recall) lze vyvolat číslo z datového registru R0 až R9 na displej. Jako parametr instrukce se zadává číslo registru 0 až 9.

Uvedením prefixu **INV** před instrukcí **RCL** se provede obdobná funkce, ale namísto z registrů R0 až R9 se číslo načte z registrů R10 až R19. Registry R10 až R19 patří do skupiny rozšířených datových registrů se sníženou přesností 13 číslic. Mantisa se na konci doplní 2 číslicemi 0.

34 Přičtení a odečtení čísla od datového registru, SUM

Symbol **SUM**

Vyvolání tlačítkem **SUM**

Pomocí tlačítka **SUM** (Summation) lze přičíst číslo na displeji k datovému registru R0 až R9. Jako parametr instrukce se zadává číslo registru 0 až 9.


Uvedením prefixu **INV** před instrukcí **SUM** se provede opačná funkce - odečtení čísla od datového registru R0 až R9.

35 Mocnina a odmocnina, y^x





Symbol **y^x**





Vyvolání tlačítkem **y^x**

Instrukce **y^x** umocní číslo Y (první operand, v zásobníku operací) číslem X (druhý operand, číslo na displeji). Stiskne-li se nejdříve prefix **INV**, provede se inverzní operace, odmocnina. Prvním operandem Y musí být

nezáporné číslo. Jedná-li se o nejnižší úroveň výrazu, může být výpočet opakován pro jiný první operand opakovaným stiskem .

Příklad:


    [2187] ... mocnina $3^7 = 2187$


    [3] ... odmocnina $2187^{(1/7)} = 3$

36 Prodleva, Pause

Symbol 

Vyvolání tlačítka  



Příkaz  pozastaví na dobu 0.25 sekundy provádění programu a zobrazí aktuální obsah displeje (obsah registru X).

Viz též monitorování běhu programu tlačítkem  ([51 Skok, GTO](#)).

37 Vložení prázdného bajtu do programu, Ins

Symbol 

Vyvolání tlačítka  

Tlačítko  (Insert), stisknuté v režimu programování, vloží na aktuální pozici programu prázdnou instrukci . Následující data se odsunou.

Bližší informace naleznete v kapitole [Programování](#).

38 Záměna čísla s datovým registrem, Exc

Symbol 

Vyvolání tlačítka **2nd** **RCL**

Pomocí tlačítka **Exc** (Exchange) lze zaměnit číslo na displeji s obsahem datového registru R0 až R9. Jako parametr instrukce se zadává číslo registru 0 až 9.

Uvedením prefixu **INV** před instrukcí **Exc** se provede obdobná funkce, ale namísto s registry R0 až R9 se číslo zamění s registry R10 až R19. Registry R10 až R19 patří do skupiny rozšířených datových registrů se sníženou přesností 13 číslic. Z mantisy se uložením odstraní poslední 2 číslice a při načtení se mantisa na konci doplní 2 číslicemi 0.

39 Vynásobení a vydělení datového registru, Prd

Symbol **Prd**

Vyvolání tlačítkem **2nd** **SUM**

Pomocí tlačítka **Prd** (Product) lze vynásobit datový registr R0 až R9 číslem na displeji. Jako parametr instrukce se zadává číslo registru 0 až 9.

Uvedením prefixu **INV** před instrukcí **Prd** se provede opačná funkce - vydělení datového registru R0 až R9 číslem na displeji.

40 Absolutní hodnota, IxI

Symbol **IxI**

Vyvolání tlačítka **2nd** **:**

Funkce **IxI** upraví číslo na absolutní hodnotu (odstraní negativní znaménko čísla).

Je-li před stiskem **IxI** uveden prefix **INV**, provede se s číslem znaménková operace **sign**. Je-li číslo menší než 0, bude výsledkem operace číslo -1. Je-li číslo větší než 0, bude výsledkem +1. Je-li číslo 0, zůstane 0 i nadále.

41 Krok programu zpět, BST

Symbol **BST**

Vyvolání tlačítkem **BST**

Tlačítkem **BST** (Back Step) se v režimu programování sníží ukazatel adresy programu o 1.

Bližší informace naleznete v kapitole [Programování](#).

42 Režim exponentu, EE

Symbol **EE**

Vyvolání tlačítkem **EE**

Tlačítkem **EE** se zapne režim exponentu. Je-li tlačítko stisknuto během zadávání čísla, přejde se na zadávání exponentu. Současně se zapne režim zobrazení ve vědecké notaci s exponentem.

Je-li tlačítko stisknuto mimo editaci čísla, zapne se režim zobrazení ve vědecké notaci s exponentem a zahájí se editace exponentu čísla. Tato funkce je často používána k odstranění skrytých číslic čísla, protože zahájením editace se do editoru načtou pouze zobrazené číslice, ne plná přesná hodnota čísla.


Stiskem prefixu **INV** před stiskem **EE** se ukončí režim zobrazení exponentu. Jinou možností ukončení režimu zobrazení exponentu je stisk klávesy **CLR**.


Příklad:

pi **Fix** **4** **EE** **INV** **EE** **INV** **Fix** [3.1416] ... zaokrouhlí číslo na 4 místa


43 Levá závorka, (


Symbol 

Vyvolání tlačítkem 

Tlačítko  zahájí výpočet části výrazu. Závorky lze používat až do 7. úrovně.


44 Pravá závorka,)


Symbol 



Vyvolání tlačítkem 




Tlačítko  ukončí výpočet části výrazu.

45 Dělení, ÷







Symbol 

Vyvolání tlačítkem 

Tlačítko  vydělí první operand druhým operandem. Jedná-li se o nejnižší úroveň výrazu, může být výpočet opakován pro jiný první operand opakovaným stiskem .

Stiskem prefixu  před stiskem  se provede opačná funkce - operace modulo , neboli zbytek po dělení. Operace modulo vydělí první operand Y (v zásobníku) druhým operandem X (na displeji), převede výsledek na celé číslo, vynásobí jím druhý operand X a odečte od prvního operandu Y. Výsledkem je zbytek po dělení. Výsledek má stejné znaménko jako první operand.

Příklad:

      [4.4] ... $2.2 : 0.5 = 4.4$

2 **.** **2** **INV** **:** **0** **.** **5** **=** [0.2] ... $2.2 \bmod 0.5 = 0.2$

2 **.** **2** **+/-** **INV** **:** **0** **.** **5** **=** [-0.2] ... $-2.2 \bmod 0.5 = -0.2$

2 **.** **2** **INV** **:** **0** **.** **5** **+/-** **=** [0.2] ... $2.2 \bmod -0.5 = 0.2$

2 **.** **2** **+/-** **INV** **:** **0** **.** **5** **+/-** **=** [-0.2] ... $-2.2 \bmod -0.5 = -0.2$

46 Žádná operace, Nop

Symbol **Nop**

Vyvolání tlačítka **2nd** **BST**

Příkaz **Nop** (No Operation) je prázdný příkaz, neprovádějící žádnou operaci. Slouží pouze k zaplnění nevyužitého místa v programu.

47 Zrušení bajtu z programu, Del

Symbol **Del**

Vyvolání tlačítka **2nd** **EE**

Tlačítko **Del** (Delete), stisknuté v režimu programování, smaže z aktuální pozice programu bajt. Následující data se přisunou.

Bližší informace naleznete v kapitole [Programování](#).

48 Zaokrouhlení, Fix

Symbol **Fix**

Vyvolání tlačítka **2nd** **|**

Pomocí tlačítka **Fix** se zaokrouhlí číslo zobrazené na displeji na zadany počet desetinných míst. Jako parametr se zadá číslice **0** až **8**, představující počet desetinných míst za desetinnou tečkou 0 až 8.

V režimu zaokrouhlování se číslo doplní zprava nulami, až do zadaného počtu desetinných míst. Zadáním posloupnosti **INV** **Fix** nebo **Fix** **9** se zaokrouhlování vypne. V tom případě se číslo zobrazí v plné přesnosti a koncové nevýznamné nuly se odstraní.

Zaokrouhlení ovlivní pouze zobrazení čísla. Vnitřně se s číslem i nadále počítá v plné přesnosti. Je-li potřeba skryté číslice skutečně odstranit, lze to provést pomocí tlačítka **EE** (viz [42 Režim exponentu, EE](#)).

Nastavený režim zaokrouhlování ovlivňuje i způsob zobrazení velmi malých čísel. Je-li zapnuto zaokrouhlování, a není-li zapnut režim exponentu, zobrazí se na displeji nuly pro malá čísla, a to i v případě, že platné číslice přetekly za pravou hranici displeje. Není-li zapnuto zaokrouhlování, kalkulátor přejde na zobrazení s exponentem v případě, že exponent je menší než -3.

49 Celé číslo, Int

Symbol **Int**

Vyvolání tlačítka **2nd** **)**

Tlačítkem **Int** (Integer) lze z čísla odstranit číslice za desetinnou tečkou neboli oříznutí čísla na celé číslo. Funkce má stejný význam jako zaokrouhlení směrem k nule.

Je-li před povelom **Int** použit prefix **INV**, provede se inverzní funkce - odstranění celé části čísla a ponechání desetinné části (frac, fraction).

Příklad:

$$\mathbf{2} \mathbf{)} \mathbf{3} \mathbf{Int} [2] \dots \text{int}(2.3) = 2$$

$$\mathbf{2} \mathbf{)} \mathbf{3} \mathbf{+/-} \mathbf{Int} [-2] \dots \text{int}(-2.3) = -2$$

$$\mathbf{2} \mathbf{)} \mathbf{3} \mathbf{INV} \mathbf{Int} [0.3] \dots \text{frac}(2.3) = 0.3$$

$$\mathbf{2} \mathbf{)} \mathbf{3} \mathbf{+/-} \mathbf{INV} \mathbf{Int} [-0.3] \dots \text{frac}(-2.3) = -0.3$$

50 Stupně, Deg

Symbol **Deg**

Vyvolání tlačítka **2nd** **x**

Tlačítko **Deg** přepne výpočty goniometrických funkcí na stupně (plný úhel je 360°).

51 Skok, GTO

Symbol **GTO**

Vyvolání tlačítkem **GTO**

Tlačítko **GTO** (Go To) slouží k provedení nepodmíněného skoku v programu. Jako parametr se zadává číselný kód návěští **0** až **9**. Uvedením příkazu **Pgm** v programu před příkazem **GTO** lze provést skok na návěští v jiném programovém prostoru.

Je-li instrukce **GTO** použita v prováděcím režimu, nastaví se ukazatel programu na zvolené návěští. To lze využít k přesunu ukazatele během programování (blíže viz [Programování](#)).



Stiskem prefixu **INV** před instrukcí **GTO** lze přesunout ukazatel v programu na absolutní adresu, která se zadá jako 2-místný číselný kód 00 až 49. Tato funkce nemůže být součástí programu, ihned se provede a lze použít k přesunu ukazatele programu jak v prováděcím režimu, tak v režimu programování.

Tlačítko **GTO** má ještě jednu speciální funkci. Podržíte-li tlačítko **GTO** během chodu programu, bude na displeji problikávat v dolním řádku aktuální obsah displeje (obsah registru X) a na horním řádku aktuální adresa programu s aktuálně prováděným příkazem. Tímto monitorováním se ovšem chod programu zřetelně zpomalí.


55 Násobení, x

Symbol 


Vyvolání tlačítkem 





Tlačítko  vynásobí první operand druhým operandem. Jedná-li se o nejnižší úroveň výrazu, může být výpočet opakován pro jiný první operand opakovaným stiskem .




56 Programová smyčka, Dsz


Symbol 

Vyvolání tlačítky  

Instrukce  (Decrement and Skip if Zero) slouží k opakovanému provádění programu pomocí smyčky s čítáním průchodů v registru R0.

Funkce  spočívá v tom, že dekrementuje (sníží o 1) registr R0 a pokud dosáhl nuly, přeskočí následující příkaz a pokračuje v činnosti. Není-li dosaženo nuly (tj. smyčka není ještě ukončena), provede se příkaz následující za příkazem . Zpravidla za  následuje příkaz , který skočí na návěští začátku smyčky.

Uvede-li se před instrukcí  prefix , provede se opačný význam instrukce - následující instrukce se přeskočí v případě, že výsledek dekrementace není nula. Tato varianta se používá zpravidla na začátku smyčky. Dosáhne-li čítač smyčky nuly, provede se následující instrukce, kterou zpravidla bývá instrukce  zajišťující skok programu ven ze smyčky.

Čítání registru R0 funguje přesněji následujícím způsobem. Je-li obsah registru R0 před operací větší než 0, hodnota registru se sníží o 1. Je-li menší než 0, hodnota se zvýší o 1. Byl-li obsah registru 0, hodnota se nezmění. Pokud je výsledkem operace nula nebo operace překročila hranici nuly, provede se operace pro nulu podle zvolené funkce. Což znamená, že pokud nedosáhl výsledek dekrementace/inkrementace nulu, smyčka se opakuje (provede se následující instrukce). S prefixem  se provede skok naopak pokud výsledek operace dosáhl nulu (nebo ji

přesáhl).

Příklad - faktoriál čísla:

RST **LRN** ... aktivace programovacího módu
00 **Lbi** **1** ... návěští začátku podprogramu
01 **STO** **0** ... uložení zadaného čísla do čítacího registru R0
02 **1** ... výchozí hodnota pro počítání součinu (střadač)
03 **Lbi** **9** ... návěští začátku smyčky
04 **(** **CE** **x** **RCL** **0** **)** ... vynásobení střadače číslem z R0
09 **Dsz** **GTO** **9** ... čítání smyčky a skok na začátek smyčky
11 **INV** **SBR** ... konec podprogramu
LRN ... ukončení režimu programování
1 **2** **SBR** **1** [479001600] ... test, faktoriál 12! = 479001600

57 Nepřímé uložení čísla do registru, **STO***

Symbol **STO***

Vyvolání tlačítka **2nd** **7**

Instrukcí **STO*** se uloží obsah displeje do datového registru podobně jako u instrukce **STO** (viz [32 Uložení čísla do datového registru, STO](#)), jen namísto z parametru instrukce se číslo registru převezme z datového registru R8. Tímto způsobem lze nepřímo adresovat všechny datové registry R0 až R79. Uvede-li se před instrukcí prefix **INV**, převezme se číslo registru z datového registru R9 (alternativní index).

Registry R10 až R79 patří do skupiny rozšířených datových registrů se sníženou přesností 13 číslic. Z mantisy se uložením odstraní poslední 2 číslice.

Uvedením adresy -1 až -256 (záporné číslo) do registru R8 (resp. R9) se instrukcí **STO*** (resp. **INV STO***) odešle bajt s hodnotou 0 až 255 (= číslo

z displeje) na port 0 až 255 (= - 1 - obsah registru R8,R9) externího zařízení. Podrobněji viz [Externí zařízení a porty](#). Tuto funkci nelze použít u varianty kalkulátoru ET-57B.

58 Nepřímé vyvolání obsahu registru, RCL*

Symbol **RCL***

Vyvolání tlačítka **2nd** **8**

Instrukcí **RCL*** se vyvolá číslo z datového registru podobně jako u instrukce **RCL** (viz [33 Vyvolání čísla z datového registru, RCL](#)), jen namísto z parametru instrukce se číslo registru převezme z datového registru R8. Tímto způsobem lze nepřímo adresovat všechny datové registry R0 až R79. Uvede-li se před instrukcí prefix **INV**, převezme se číslo registru z datového registru R9 (alternativní index).

Registry R10 až R79 patří do skupiny rozšířených datových registrů se sníženou přesností 13 číslic. Mantisa se na konci doplní 2 číslicemi 0.

Uvedením adresy -1 až -256 (záporné číslo) do registru R8 (resp. R9) se instrukcí **RCL*** (resp. **INV RCL***) načte bajt s hodnotou 0 až 255 z portu 0 až 255 (= - 1 - obsah registru R8,R9) externího zařízení. Podrobněji viz [Externí zařízení a porty](#). Tuto funkci nelze použít u varianty kalkulátoru ET-57B.

59 Nepřímé přičtení a odečtení čísla z registru, SUM*

Symbol **SUM***

Vyvolání tlačítka **2nd** **9**

Instrukcí **SUM*** se přičte (nebo s **INV** odečte) číslo k datovému registru podobně jako u instrukce **SUM** (viz [34 Přičtení a odečtení čísla od datového registru, SUM](#)), jen namísto z parametru instrukce se číslo registru převezme z datového registru R8. Tímto způsobem lze nepřímo adresovat všechny datové registry R0 až R79.

Registry R10 až R79 patří do skupiny rozšířených datových registrů se

sníženou přesností 13 číslic. Z mantisy se uložením odstraní poslední 2 číslice.

60 Radiány, Rad

Symbol **Rad**

Vyvolání tlačítka **2nd** **|**

Tlačítko **Rad** přepne výpočty goniometrických funkcí na radiány (plný úhel je $2\pi = 6.283185\dots$).

61 Podprogram, SBR

Symbol **SBR**

Vyvolání tlačítkem **SBR**


Tlačítko **SBR** (Subroutine) slouží k vyvolání podprogramu. Jako parametr se zadává číselný kód návěští **0** až **9**. Uvedením příkazu **Pgm** v programu před příkazem **SBR** se zavolá podprogram z jiného programového prostoru. Je-li instrukce **SBR** použita v prováděcím režimu, podprogram se ihned spustí.



Při volání podprogramu se do zásobníku adres nejdříve uloží adresa instrukce následující za kódem instrukce **SBR**. Poté se předá řízení podprogramu. Zásobník adres má kapacitu omezenou na 7 podprogramů.





Podprogram je ukončen instrukcí **INV** **SBR**, která zajistí návrat z podprogramu. Ze zásobníku adres se převezme původní adresa za instrukcí **SBR** a předá se na tuto adresu řízení. Pokud byl spuštěn podprogram z jiného programového prostoru, řízení se navrátí zpět do původního programového prostoru. Pokud byl podprogram spuštěn z klávesnice, kalkulátor se zastaví.

65 Odečtení, -

Symbol 

Vyvolání tlačítkem 



Tlačítko  odečte druhý operand od prvního operandu. Jedná-li se o nejnižší úroveň výrazu, může být výpočet opakován pro jiný první operand opakovaným stiskem .

Uvedením prefixu  před instrukcí  se vypne ladicí mód (zobrazení mantisy čísla). Ladicí mód byl aktivován instrukcí  .

66 Rovno, x=t

Symbol 



Vyvolání tlačítky  


Instrukce  umožňuje porovnat registr X (obsah displeje) s pomocným registrem T (nastaveným tlačítkem , odpovídá registru R7). Jsou-li registry shodné, provede se následující instrukce. V opačném případě program následující instrukci přeskočí.

67 Nepřímá inkrementace/dekrementace registru, Inc*

Symbol 

Vyvolání tlačítky  

Instrukce  inkrementuje/dekrementuje obsah datového registru podobně jako instrukce  (viz [77 Inkrementace a dekrementace registru, Inc](#)), jen namísto z parametru instrukce se číslo registru převezme z datového registru R8.

Uvedením prefixu  před instrukcí se namísto inkrementace (zvýšení o 1) provede dekrementace (snížení o 1).

68 Nepřímá záměna čísla s datovým registrem, Exc*

Symbol **Exc***

Vyvolání tlačítka **2nd** **5**

Instrukcí **Exc*** lze zaměnit číslo na displeji s obsahem datového registru podobně jako instrukce **Exc** (viz [38 Záměna čísla s datovým registrem, Exc](#)), jen namísto z parametru instrukce se číslo registru převeze z datového registru R8.

Tímto způsobem lze nepřímo adresovat všechny datové registry R0 až R79. Uvede-li se před instrukcí prefix **INV**, převeze se číslo registru z datového registru R9 (alternativní index).

Registry R10 až R79 patří do skupiny rozšířených datových registrů se sníženou přesností 13 číslic. Mantisa se při uložení zkrátí o 2 poslední číslice a při načtení se na konci doplní 2 číslicemi 0.

69 Nepřímé násobení a dělení registru, Prd*

Symbol **Prd***

Vyvolání tlačítka **2nd** **6**

Instrukcí **Prd*** lze vynásobit či vydělit (s prefixem **INV**) obsah datového registru číslem na displeji stejně jako instrukce **Prd** (viz [39 Vynásobení a vydělení datového registru, Prd](#)), jen namísto z parametru instrukce se číslo registru převeze z datového registru R8. Tímto způsobem lze nepřímo adresovat všechny datové registry R0 až R79.

70 Grady, Grad

Symbol **Grad**

Vyvolání tlačítka **2nd** **+**


Tlačítko **Grad** přepne výpočty goniometrických funkcí na grady (plný úhel

je 400).

71 Reset, RST


Symbol 



Vyvolání tlačítkem 




Tlačítko  slouží k resetování ukazatele programu, tj. nastavení ukazatele na 0. Zvolený programový prostor zůstane nezměněn.

75 Přičtení, +

Symbol 

Vyvolání tlačítkem 




Tlačítko  přičte druhý operand k prvnímu operandu. Jedná-li se o nejnižší úroveň výrazu, může být výpočet opakován pro jiný první operand opakovaným stiskem .

Uvedením prefixu  před instrukcí  se zapne ladicí mód. V ladicím módu je na prvním řádku displeje namísto přepínačů zobrazen obsah mantisy zobrazeného čísla. Deaktivace ladicího módu je možná stiskem .

76 Větší nebo rovno, x>=t

Symbol 

Vyvolání tlačítky  

Instrukce  umožňuje porovnat registr X (obsah displeje) s pomocným registrem T (nastaveným tlačítkem , odpovídá registru R7). Je-li registr X větší nebo roven registru T, provede se příkaz následující za instrukcí . V opačném případě se následující příkaz přeskočí.

Je-li před kódem **X>=t** zadán prefix **INV**, provede se inverzní funkce - provedení následujícího příkazu v případě, že registr X je menší než registr T.

77 Inkrementace a dekrementace registru, Inc

Symbol **Inc**

Vyvolání tlačítka **2nd 1**

Instrukce **Inc** inkrementuje (zvýší o 1) obsah datového registru R0 až R9, jehož číslo **0** až **9** je uvedeno jako parametr instrukce. Uvede-li se před instrukcí **Inc** prefix **INV**, provede se opačná operace - dekrementace registru (snížení o 1).

78 Výběr programového prostoru, Pgm

Symbol **Pgm**

Vyvolání tlačítka **2nd 2**

Tlačítkem **Pgm** lze vybrat programový prostor 0 až 9. Číslo programového prostoru se zvolí číslem **0** až **9**, zadaným jako parametr instrukce **Pgm**. V kalkulátoru je k dispozici celkem 10 nezávislých programových prostorů, označených číslem 0 až 9. V každém programovém prostoru je k dispozici 50 programových kroků, tedy celkem 500 programových kroků.

Při přepnutí programového prostoru v prováděcím režimu (z klávesnice) se současně resetuje ukazatel programu na začátek nově zvoleného programového prostoru.

Uvedením příkazu **Pgm** v programu se nepřepne programový prostor trvale, ale pouze dočasně pro jednu následující instrukci **GTO** nebo **SBR**. Tak lze volat podprogramy či provádět skoky mezi programovými prostory. Instrukce **GTO** a **SBR** nemusí bezprostředně následovat za instrukcí **Pgm**.

79 Generátor náhodného čísla, Rand

Symbol **Rand**

Vyvolání tlačítka **2nd** **3**

Funkce **Rand** vypočte náhodné číslo v rozsahu 0 až 1, včetně hodnoty 0, ale vyjma hodnoty 1. K výpočtu náhodného čísla se používá LCG generátor (Linear Congruential Generator, Lineární kongruentní generátor) a vzorec $\text{Rand} = (\text{Seed} = (\text{Seed} * 214013 + 2531011) \bmod 4294967296) / 4294967296$.

Seed generátoru má rozsah 32 bitů. Vygenerované číslo je převedeno na float vydělením hodnotou 2^{32} . To zajistí, že výsledné náhodné číslo je v rozsahu 0 až 1, včetně nuly, ale vyjma hodnoty 1. Omezený počet číslic výchozího čísla (9 a půl číslice, rozsah čísla 0 až 4294967295) zajišťuje takovou granularitu údaje, že ani po vynásobení nedojde k přetečení na hraniční koncovou hodnotu 1.

Generátor náhodného čísla při každém použití počítá neustále dál a to zajistí, že vygenerované sekvence čísel se neopakují (přesněji - interval opakování je velmi velký, 2^{32} čísel). Při každém zapnutí kalkulátoru je seed generátoru načteno z EEPROM a uložena nová hodnota. To zajistí, že se generované sekvence neopakují po zapnutí kalkulátoru.

Příklad programu pro hod kostkou:

RST **LRN** ... aktivace programovacího módu

00 **Lbl** **1** ... návěští začátku podprogramu

01 **[** **6** **x** **Rand** **+** **1** **]** ... výpočet $6 \times \text{rand} + 1 = \text{číslo } 1..6.9999$

08 **Int** ... celočíselná část čísla

09 **INV** **SBR** ... konec podprogramu

LRN ... deaktivace programovacího módu

SBR **1** ... hod kostkou, vygeneruje náhodné číslo 1 až 6

Pokračování příkladu - statistický test hodu kostkou:

LRN ... znovu aktivace programovacího módu

10 **Lbl** **2** ... návěští začátku podprogramu

11 **SBR** **1** ... vygenerování náhodného čísla 1..6

12 **STO** **8** ... uložení čísla do indexového registru R8

13 **Inc*** ... inkrementace registru s indexem R8

14 **GTO** **2** ... pokračování testu

LRN ... deaktivace programovacího módu

INV **C.t** ... vynulování všech datových registrů

SBR **2** ... start testu

R/S ... po nějaké chvíli zastavení programu

RCL n ... kontrolní zobrazení registrů 0 až 7. Při správné funkci obsahují registry R1 až R6 zhruba podobné číslo (podobný počet případů), registry R0 a R7 musí obsahovat 0.

V testu lze pokračovat novým spuštěním **SBR** **2**. Po 10 minutách testu obsahují registry R0 až R7 čísla (např.) 0, 2762, 2727, 2834, 2701, 2671, 2763, 0, což odpovídá předpokladu. Rychlost generování je 27 hodů kostkou za sekundu.


80 Rozptyl, Var

Symbol **Var**


Vyvolání tlačítka **2nd** **=**

Instrukce **Var** (Variance) vypočte rozptyl hodnot 'x' a 'y' zadaných pomocí statistické funkce **Stat**. Po stisku **Var** se na displeji zobrazí rozptyl z hodnot 'y'. Zadáním prefixu **INV** před instrukcí **Var** se vypočte rozptyl z hodnot 'x'.



Rozptyl je počítán vzorcem $\text{var} = \text{sum}(y^2)/N - (\text{sum}(y)/N)^2$.

Odmocněním rozptylu funkcí  se vypočte směrodatná odchylka 's'.

81 Start a stop programu, R/S

Symbol 


Vyvolání tlačítkem 



Tlačítkem  lze spustit nebo zastavit probíhající program. Při spuštění se program začne provádět od aktuálně nastaveného ukazatele programu (aktuální adresu lze zjistit přepnutím do programovacího módu ).

Program po zastavení nemusí být vždy schopen pokračovat v běhu - může dojít ke ztrátě návratové adresy z podprogramu nebo může zůstat přepnutý jiný programový prostor.

83 Desetinná tečka, .

Symbol 


Vyvolání tlačítkem 

Tlačítko  je oddělovač celočíselných číslic mantisy a desetinných číslic mantisy. Je-li tlačítko  stisknuto během editace exponentu čísla, přejde editace zpět k zadávání mantisy čísla.

84 Změna znaménka, +/-

Symbol 


Vyvolání tlačítkem 


Tlačítko  změní znaménko čísla na displeji. Je-li stisknuto během zadávání exponentu čísla, změní znaménko exponentu.


85 Provedení výpočtu, =

Symbol 





Vyvolání tlačítkem 



Tlačítko  slouží k uzavření otevřených aritmetických operací a provedení výpočtu.





Opakovaným stiskem tlačítka  lze opakovat poslední prováděnou operaci nejnižší úrovně. Prvním operandem je číslo na displeji, druhým operandem je číslo zadané během operace jako druhý operand (nebo druhý výsledek mezivýpočtu).




Upozornění: Některé programy původní TI-57 s opakováním operací nepočítají. Použijí klávesu  vícekrát a tím může vzniknout chybný výpočet. Při importu programů může být nutné tuto skutečnost zkontrolovat a ošetřit.

Příklad:

    [30] ... $5 \times 6 = 30$

  [42] ... $7 \times 6 = 42$







    [3] ... $9 / 3 = 3$

   [4] ... $12 / 3 = 4$

86 Návěští, Lbl

Symbol 

Vyvolání tlačítky  

Instrukcí  lze označit místo v programu jako návěští. V každém programovém prostoru lze použít 10 návěstí, označených  až . Číslo návěští je uvedeno jako číselný parametr  až  instrukce .

Na místo programu označené návěští lze skákat pomocí instrukce skoku

GTO nebo instrukcí podprogramu **SBR**. S využitím instrukce pro volbu programového prostoru **Pgm** lze provádět skoky a volání podprogramů i mezi programovými prostory.

87 Faktoriál, x!

Symbol **x!**

Vyvolání tlačítka **2nd 0**

Tlačítko **x!** slouží k výpočtu faktoriálu. Faktoriál je číslo, které vznikne vynásobením hodnot 1, 2, 3, ... až po zadané vstupní číslo x. Vstupní číslo je celé číslo v rozsahu 1 až 69.

Příklad:

6 x! [720] ... faktoriál čísla 6! = $1*2*3*4*5*6 = 720$

88 Statistika, Stat

Symbol **Stat**

Vyvolání tlačítka **2nd !**

Instrukce **Stat** slouží k zadávání dat při provádění statistických výpočtů (průměr, rozptyl). Instrukce používá datové registry R0 až R5 k ukládání mezivýpočtů. Před použitím je vhodné registry nejdříve vynulovat příkazem **INV C.t**, který vynuluje všechny datové registry R0 až R79.

Použití datových registrů:

R0 ... počet položek N

R1 ... součet y

R2 ... součet y^2

R3 ... součet x

R4 ... součet x^2

R5 ... součet $x*y$

R7 ... registr T

Při vkládání statistických dat po dvojicích (x, y) se nejdříve zapíše hodnota 'x' a stiskem **x<>t** se přeneseme do registru T (registr R7). Poté se zapíše hodnota 'y' a stiskem **Stat** se obě hodnoty 'x' a 'y' uloží. Na displeji (v registru X) se objeví počet dosud vložených položek 'n'. Obsah registru T (hodnota 'x') se instrukcí **Stat** zvýší o 1. To z důvodu, že pokud se hodnoty 'x' mají lišit o 1, není potřeba je vkládat, stačí vložit počáteční hodnotu 'x' do registru T a poté zapisovat už jen hodnoty 'y'. Nejsou-li potřeba vyhodnocovat dvojice hodnot (x, y), postačí zadávat pouze hodnotu 'y'.

Uvede-li se před instrukcí **Stat** prefix **INV**, vložená hodnota se naopak odečte. Tak lze opravit chybně vloženou hodnotu - vloží se hodnota 'x' chybného údaje, stiskne se **x<>t**, vloží se hodnota 'y' chybného údaje a stiskem **INV Stat** se chybné údaje odečtou. Obsah registru T (s hodnotou 'x') se sníží o 1. Poté lze pokračovat novým správným údajem. Nejsou-li potřeba vyhodnocovat dvojice hodnot (x, y), postačí zadávat pouze hodnotu 'y' chybného údaje.

Příklad:

INV C.t ... vynulování datových registrů

9 6 Stat [1] ... 1. údaj 96

8 1 Stat [2] ... 2. údaj 81

9 7 Stat [3] ... 3. údaj je chybný

9 7 INV Stat [2] ... zrušení 3. údaje

8 7 Stat [3] ... 3. opravný údaj 87

7 0 Stat [4] ... 4. údaj 70

9 3 Stat [5] ... 5. údaj 93

7 7 Stat [6] ... 6. údaj 77

Mean [84] ... průměr zadaných hodnot = 84

Var [81.333...] ... rozptyl = 81.333...

Vx [9.0185...] ... směrodatná odchylka = 9.0185...

RCL 1 [504] ... součet všech hodnot = 504

89 Průměr, Mean

Symbol **Mean**

Vyvolání tlačítka **2nd** **+/-**

Instrukce **Mean** vypočte průměr z hodnot 'x' a 'y' zadaných pomocí statistické funkce **Stat**. Po stisku **Mean** se na displeji zobrazí průměr z hodnot 'y'. Zadáním prefixu **INV** před instrukcí **Mean** se vypočte průměr z hodnot 'x'.

15. Příklady programů

1. Hod kostkou

Program využívá interní generátor náhody.

Použití:

SBR **1** ... vygeneruje náhodné celé číslo 1 až 6 (**R/S** = další číslo)

Program:

00: 86 1	Lbl 1	návěští začátku podprogramu
01: 43	(výpočet (6 x rand + 1), náhodné číslo 1..6.9999
02: 06	6	
03: 55	x	
04: 79	Rand	náhodné číslo 0 (včetně) až 1 (vyjma)
05: 75	+	
06: 01	1	
07: 44)	
08: 49	Int	celočíselná část čísla
09: -61	INV SBR	konec podprogramu
10: 51 1	GTO 1	další číslo s R/S

2. Hod kostkou 2

Program využívá ke generování náhodného čísla standardní vybavení kalkulátoru TI-57, bez rozšíření ET-57.

Použití:

x **STO** 0 ... inicializace generátoru náhody na číslo Seed = x

SBR 1 ... vygeneruje náhodné celé číslo 1 až 6 (**R/S** = další číslo)

Registry:

R0 ... střadač generátoru náhody (seed)

Program:

00: 86 1	Lbl 1	návěští začátku podprogramu
01: 43	(výpočet seed = frac((seed + pi)^8)
02: 43	(
03: 33 0	RCL 0	střadač generátoru (seed)
04: 75	+	
05: 30	pi	
06: 44)	
07: 35	y^x	
08: 8	8	
09: 44)	
10: 43	(
11: -49	INV Int	
12: 32 0	STO 0	uložení nového střadače (seed)
13: 55	x	výpočet int(seed x 6 + 1)
14: 06	6	
15: 75	+	
16: 01	1	
17: 44)	
18: 49	Int	
19: -61	INV SBR	
20: 51 1	GTO 1	další číslo s R/S

3. Světelný LED had

Program ovládá externí zařízení - efektový rám ERAM100. Zápisem bajtu na port 0 se ovládá 8 LED. Zapsaný bajt je kombinací bitů rozsvěčujících jednotlivé LED, jako součet bitových vah:

bit 0 ... váha 1 ... LED 1
bit 1 ... váha 2 ... LED 2
bit 2 ... váha 4 ... LED 3
bit 3 ... váha 8 ... LED 4
bit 4 ... váha 16 ... LED 5
bit 5 ... váha 32 ... LED 6
bit 6 ... váha 64 ... LED 7
bit 7 ... váha 128 ... LED 8

Použití:

RST **R/S** ... spustí hada, po pásu rotují 3 svítící LED

Registry:

R5 ... střadač patternu hada

R8 ... index, -1 je index portu 0 (výstup na LED)

Program:

00: 07	7	bitový pattern vzhledu hada (rozsvítí LED 1,2,3)
01: 32 5	STO 5	uloží střadač patternu
02: 01	1	
03: 84	+/-	
04: 32 8	STO 8	nastaví index R8 na -1, to je adresa portu 0
05: 02	2	
06: 05	5	
07: 06	6	
08: 22	x<>t	do T připraví 256 pro test přetečení patternu
09: 86 9	Lbl 9	návěští začátku cyklu
10: 33 5	RCL 5	vyvolá střadač patternu
11: 55	x	
12: 02	2	
13: 85	=	(pattern x 2) odpovídá posunu o 1 bit vlevo
14: -76	INV x>=t	je pattern < 256 ?
15: 51 8	GTO 8	není přetečení patternu, přeskočení na Lbl 8

16: 65	-	
17: 02	2	
18: 05	5	
19: 05	5	
20: 85	=	rotace: - 256 (odstraní bit 7) + 1 (přidá bit 0)
21: 86 8	Lbl 8	
22: 32 5	STO 5	uložení nového patternu
23: 57	STO*	uložením na adresu -1 se bajt odešle na port 0
24: 36	Pause	malá prodleva 0.25 sekundy
25: 51 9	GTO 9	opakování smyčky

4. Vyčíslení polynomu

Program vypočte hodnotu polynomu $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ řádu n pro dané x , jsou-li zadány koeficienty a_0 až a_n .

Použití:

RST **R/S** ... zahájení zadávání koeficientů, pokračování: **R/S**

SBR **1** ... vyhodnocení polynomu pro dané ' x ', pokračování: nové ' x ' **R/S**

Příklad, polynom $P(x) = 2 - 3x + x^2$:

RST **R/S** ... zahájení zadávání koeficientů

2 **R/S** ... zadání koeficientu $a_0 = 2$

3 **+/-** **R/S** ... zadání koeficientu $a_1 = -3$

1 **R/S** ... zadání koeficientu $a_2 = 1$

2 **SBR** **1** [0] ... výpočet hodnoty polynomu $P(2) = 0$

1 **+/-** **R/S** [6] ... výpočet hodnoty polynomu $P(-1) = 6$

1 **5** **R/S** [182] ... výpočet hodnoty polynomu $P(15) = 182$

Registry:

R0 ... čítač položek Dsz

R6 ... počet koeficientů N (= řád polynomu $n + 1$)

R8 ... indexový registr

R9 ... vstup x

R10 a další ... koeficienty polynomu a_0, a_1, \dots, a_n (N koeficientů)

Program:

- obsluha zadání koeficientů

00: 15	CLR	vynulování
01: 32 6	STO 6	nulování počtu koeficientů N
02: 09	9	index prvního koeficientu $10 - 1 = 9$
03: 32 8	STO 8	nastavení indexového registru

04: 86 9	Lbl 9	návěští smyčky pro zadání koeficientů
05: 81	R/S	stop programu, čekání na další koeficient
06: 77 8	Inc 8	inkrementace indexového registru R8
07: 57	STO*	uložení koeficientu
08: 77 6	Inc 6	inkrementace počtu koeficientů N
09: 51 9	GTO 9	pokračování dalším koeficientem
- obsluha vyhodnocení polynomu pro dané 'x'		
10: 86 8	Lbl 8	návěští opakování výpočtu
11: 81	R/S	zobrazení výsledku a čekání na další 'x'
12: 86 1	Lbl 1	návěští začátku funkce vyčíslení polynomu
13: 32 9	STO 9	úschova zadaného 'x'
14: 43	(
15: 33 6	RCL 6	počet koeficientů N
16: 32 0	STO 0	příprava čítače Dsz
17: 75	+	
18: 09	9	$N + 9 =$ index posledního koeficientu an
19: 44)	
20: 32 8	STO 8	nastavení indexového registru R8
21: 58	RCL*	načtení posledního koeficientu an
22: 86 7	Lbl 7	návěští začátku smyčky
23: -56	INV Dsz	dekrementace R0 a přeskočení instrukce když > 0
24: 51 8	GTO 8	konec smyčky pokud R0 dosáhne 0
25: -77 8	INV Inc 8	dekrementace indexového registru R8
26: 43	(
27: 14	CE	zde je nyní střadač výsledku
28: 55	x	
29: 33 9	RCL 9	střadač * zadané 'x'
30: 75	+	
31: 58	RCL*	přičtení dalšího koeficientu ai (z indexu R8)
32: 44)	
33: 51 7	GTO 7	pokračování ve smyčce

5. Komplexní čísla

Aritmetika komplexních čísel. V registru T (registr R7) je vždy reálná část čísla, v registru X (displej) je imaginární část. Při zadání čísla zadejte reálnou část, stiskněte $x \leftrightarrow t$ a zadejte imaginární část. Při čtení výsledku po stisku $x \leftrightarrow t$ přečtete reálnou část, po druhém stisku $x \leftrightarrow t$ imaginární část.

U funkcí se 2 operandy zadejte první operand (X) pomocí **SBR** 1. Poté zadejte druhý operand (Y) a zavolejte funkci. Výsledek se stane současně novým prvním operandem (X).

U funkcí s 1 operandem zadejte operand (Y) a zavolejte funkci. Původní první operand (X) zůstane neovlivněn.

Program obsadí 2 programové prostory, **Pgm** 0 a **Pgm** 1.

Poznámka: Funkce $\ln(Y)$ a $\exp(Y)$ přepnou úhlovou míru na radiány.

Použití:

Programový prostor **Pgm** 0:

SBR 1 ... zadání prvního operandu X

SBR 2 ... přičtení druhého operandu $X+Y$

SBR 3 ... odečtení druhého operandu $X-Y$

SBR 4 ... násobení druhým operandem $X*Y$

SBR 5 ... dělení druhým operandem X/Y

SBR 6 ... negace $-Y$

*Poznámka: Funkce **SBR** 5 volá z **Pgm** 1 funkci **SBR** 1.*

Programový prostor **Pgm** 1:

SBR 1 ... převrácená hodnota $1/Y$

SBR 2 ... druhá mocnina Y^2

SBR 3 ... druhá odmocnina \sqrt{Y}

SBR 4 ... přirozený logaritmus $\ln(Y)$

SBR 5 ... přirozený exponent $\exp(Y)$

Příklad, výpočet $\exp(((2+3i) - (1-i)) / (4+5i))$:

Pgm 0 ... volba programového prostoru 0

2 **x<>t** 3 **SBR** 1 ... zadání prvního operandu $(2+3i)$

1 **x<>t** 1 **+/-** **SBR** 3 ... odečtení druhého operandu $(1-i)$

x<>t [1] **x<>t** [4] ... mezivýsledek $= (1+4i)$

4 **x<>t** 5 **SBR** 5 ... vydělení operandem $(4+5i)$

x<>t [0.5853...] **x<>t** [0.2682...] ... mezivýsledek $= (0.5853...+0.2682...i)$

Pgm 1 ... volba programového prostoru 1

SBR 5 ... přirozený exponent $\exp(Y)$

x<>t [1.7314...] **x<>t** [0.4760...] ... výsledek $(1.7314... + 0.4760..i)$

Registry:

R1 ... 'a', reálná část prvního operandu X

R2 ... 'b', imaginární část prvního operandu X

R3 ... 'c', reálná část druhého operandu Y

R4 ... 'd', imaginární část druhého operandu Y

R7 ... registr T

Program:

Pgm 0 ... volba programového prostoru 0

- obsluha odečtení druhého operandu X-Y

00: 86 3 Lbl 3 návěští funkce X-Y

01: 61 6 SBR 6 volání funkce negace -Y

- obsluha přičtení druhého operandu X+Y

$X + Y = (a + c) + (b + d)*i$

02: 86 2 Lbl 2 návěští funkce X+Y

03: 34 2	SUM 2	přičtení imaginární části Y k X
04: 33 2	RCL 2	vyvolání imaginární části X
05: 22	x<>t	záměna částí
06: 34 1	SUM 1	přičtení reálné části Y k X
07: 33 1	RCL 1	vyvolání reálné části X
08: 22	x<>t	záměna částí

- obsluha zadání prvního operandu X

09: 86 1	Lbl 1	návěští funkce X
10: 32 2	STO 2	uložení imaginární části X
11: 22	x<>t	záměna částí
12: 32 1	STO 1	uložení reálné části X
13: 22	x<>t	záměna částí
14: -61	INV SBR	

- obsluha dělení druhým operandem X/Y

15: 86 5	Lbl 5	návěští funkce X/Y
16: 78 1	Pgm 1	volba programového prostoru 1
17: 61 1	SBR 1	volání funkce převrácená hodnota 1/Y

- obsluha násobení druhým operandem X*Y

$$X * Y = (a*c - b*d) + (a*d + b*c)*i$$

18: 86 4	Lbl 4	návěští funkce X*Y
19: 32 4	STO 4	úschova imaginární části Y (=d)
20: 22	x<>t	záměna částí
21: 43	(
22: 32 3	STO 3	úschova reálné části Y (=c)
23: 55	x	
24: 33 1	RCL 1	reálná část X (=a)
25: 65	-	
26: 33 2	RCL 2	imaginární část X (=b)
27: 55	x	
28: 33 4	RCL 4	imaginární část Y (=d)
29: 44)	- zde je vypočteno (a*c - b*d), reálná část
30: 22	x<>t	záměna částí
31: 43	(
32: 33 1	RCL 1	reálná část X (=a)
33: 55	x	
34: 33 4	RCL 4	imaginární část Y (=d)
35: 75	+	
36: 33 2	RCL 2	imaginární část X (=b)
37: 55	x	
38: 33 3	RCL 3	reálná část Y (=c)

39: 44)
 40: 51 1 GTO 1 - zde je vypočteno ($a*d + b*c$), imaginární část
 uložení výsledku do X

- obsluha negace -Y

41: 86 6 Lbl 6 návěští funkce -Y
 42: 84 +/- negace imaginární části
 43: 22 x<>t záměna částí
 44: 84 +/- negace reálné části
 45: 22 x<>t záměna částí
 46: -61 INV SBR

Pgm 1 ... volba programového prostoru 1

- obsluha převrácené hodnoty 1/Y

$$1/Y = (c - d*i)/(c^2 + d^2)$$

00: 86 1 Lbl 1 návěští funkce 1/Y
 01: 43 (
 02: 84 +/- negace imaginární části Y (=d)
 03: 45 :
 04: 43 (
 05: 23 x^2 d^2
 06: 75 +
 07: 22 x<>t
 08: 32 3 STO 3 úschova reálné části Y (=c)
 09: 23 x^2 c^2
 10: 44)
 11: -39 3 INV Prd 3 vydělení $c/(c^2 + d^2)$
 12: 44)
 13: 22 x<>t výměna částí
 14: 33 3 RCL 3 vyvolání nové reálné části
 15: 22 x<>t výměna částí
 16: -61 INV SBR

- obsluha druhá mocnina Y^2

17: 86 2 Lbl 2 návěští funkce Y^2
 18: 43 (
 19: -27 INV P->R konverze na polární souřadnice
 20: 55 x úhel se bude násobit *2
 21: 22 x<>t záměna částí
 22: 23 x^2 mocnina radiusu
 23: 86 9 Lbl 9 interní návěští

24: 22	x<>t	záměna částí
25: 02	2	
26: 44)	zde je vypočítán úhel * 2 (resp. úhel / 2)
27: 27	P->R	konverze zpět na kartézské souřadnice
28: -61	INV SBR	

- obsluha druhá odmocnina \sqrt{Y}

29: 86 2	Lbl 3	návěští funkce \sqrt{Y}
30: 43	(
31: -27	INV P->R	konverze na polární souřadnice
32: 45	:	úhel se bude dělit /2
33: 22	x<>t	záměna částí
34: 24	Vx	odmocnina radiusu
35: 51 9	GTO 9	dokončení funkce

- obsluha přirozený logaritmus $\ln(Y)$

36: 86 4	Lbl 4	návěští funkce $\ln(Y)$
37: 60	Rad	přepnutí na radiány
38: -27	INV P->R	konverze na polární souřadnice
39: 22	x<>t	záměna částí
40: 13	Inx	logaritmus radiusu
41: 22	x<>t	záměna částí
42: -61	INV SBR	

- obsluha přirozený exponent $\exp(Y)$

43: 86 5	Lbl 5	návěští funkce $\exp(Y)$
44: 22	x<>t	záměna částí
45: -13	INV Inx	exponent $\exp(x)$
46: 22	x<>t	záměna částí
47: 60	Rad	přepnutí na radiány
48: 27	P->R	konverze na kartézské souřadnice
49: -61	INV SBR	

6. Ramanujanova aproximace faktoriálu x!

Výpočet faktoriálu (včetně desetinných čísel) pomocí Ramanujanovy aproximace. Dosažená přesnost: Hodnoty kolem 1 přesnost 3 číslice, hodnoty kolem 69 (maximum) přesnost 10 číslic.

Vzorec: $x! = \sqrt{\pi} * (x/e)^x * (((8*x + 4)*x + 1)^x + 1/30)^{(1/6)}$.

Použití:

SBR **1** ... výpočet faktoriálu pro dané 'x' (další 'x' pokračovat s **R/S**)

Příklady:

1 **SBR** **1** [1.00028..] ... 1! = 1.00028..., správná hodnota má být 1.0

1 **2** **R/S** [1.101987...] ... 1.2! = 1.101987..., správně 1.101802...

1 **0** **R/S** [3628800.3116] ... 10! = 3628800.3116, správně 3628800

6 **9** **R/S** [1.7112245244+98] ... 69! = 1.7112245244+98,

správná hodnota 1.7112245243+98

Registry:

R1 ... zadaná hodnota x

Program:

00: 86 1	Lbl 1	návěští programu
01: 43	(
02: 43	(
03: 32 1	STO 1	úschova zadané hodnoty x
04: 45	:	počítá se x/e
05: 01	1	
06: -13	INV ln x	výpočet konstanty e
07: 44)	- zde je vypočteno x/e
08: 35	y^x	
09: 33 1	RCL 1	- zde se počítá (x/e)^x
10: 55	x	
11: 30	pi	

12: 24	Vx	výpočet konstanty sqrt(pi)
13: 55	x	
14: 43	(
15: 43	(
16: 43	(
17: 08	8	
18: 55	x	
19: 33 1	RCL 1	výpočet $8 \cdot x$
20: 75	+	
21: 04	4	
22: 44)	- zde je vypočteno $(8 \cdot x + 4)$
23: 55	x	
24: 33 1	RCL 1	
25: 75	+	
26: 01	1	
27: 44)	- zde je vypočteno $((8 \cdot x + 4) \cdot x + 1)$
28: 55	x	
29: 33 1	RCL 1	
30: 75	+	
31: 03	3	
32: 00	0	
33: 25	1/x	
34: 44)	- zde je vypočteno $\dots \cdot x + 1/30$
35: -35	INV y^x	odmocnina
36: 06	6	
37: 44)	
38: -61	INV SBR	
39: 51 1	GTO 1	opakování výpočtu pro další x

7. Stirlingova aproximace faktoriálu $\ln(x!)$

Výpočet faktoriálu (včetně desetinných čísel) pomocí Stirlingovy aproximace. Dosažená přesnost: Hodnoty kolem 1 přesnost 3 číslice, hodnoty kolem 69 (maximum $x!$) přesnost 10 číslic, hodnoty kolem 200 přesnost 15 číslic. Stirlingova aproximace se používá především k výpočtu logaritmu faktoriálu velmi velkých čísel, kdy dosahuje velké přesnosti.

Vzorec: $\ln(x!) = x \cdot \ln(x) + \ln(\sqrt{2 \cdot \pi}) - x + \ln(\sqrt{x + 1/6 + 1/72/x - 31/6480/x^2})$

Použití:

SBR 1 ... výpočet logaritmu faktoriálu $\ln(x!)$ (další 'x' s **R/S**)

SBR 2 ... výpočet faktoriálu $x!$ (další 'x' s **R/S**)

Příklady:

1 SBR 2 [0.99990...] ... $1! = 0.99990...$, správná hodnota má být 1.0

1 2 R/S [1.10177...] ... $1.2! = 1.10177...$, správně 1.101802...

1 0 R/S [3628800.1322] ... $10! = 3628800.1322$, správně 3628800

6 9 R/S [1.7112245243+98] ... $69! = 1.7112245243+98$,
správná hodnota 1.7112245243+98

1 0 0 0 SBR 1 [5912.1281785] ... $\ln(1000!) = 5912.1281785$,
správná hodnota 5912.1281785

Registry:

R1 ... zadaná hodnota x

Program:

- obsluha funkce $\ln(x!)$

00: 81 1	Lbl 1	návěští začátku funkce $\ln(x!)$
01: 43	(
02: 32 1	STO 1	úschova zadané hodnoty x
03: 55	x	

04: 13	lnx	výpočet $x \cdot \ln(x)$
05: 75	+	
06: 43	(
07: 02	2	
08: 55	x	
09: 30	pi	
10: 44)	
11: 24	\sqrt{x}	
12: 13	lnx	- zde je vypočteno $\ln(\sqrt{2 \cdot \pi})$
13: 65	-	
14: 33 1	RCL 1	zadané x
15: 75	+	
16: 43	(
17: 33 1	RCL 1	zadané x
18: 75	+	
19: 06	6	
20: 25	$1/x$	člen + $1/6$
21: 75	+	
22: 07	7	
23: 02	2	
24: 25	$1/x$	
25: 45	:	
26: 33 1	RCL 1	člen + $1/72/x$
27: 65	-	
28: 03	3	
29: 01	1	
30: 45	:	
31: 06	6	
32: 04	4	
33: 08	8	
34: 00	0	
35: 45	:	
36: 33 1	RCL 1	
37: 23	x^2	člen - $31/6480/x^2$
38: 44)	
39: 24	\sqrt{x}	odmocnina řady
40: 13	lnx	
41: 44)	
42: -61	INV SBR	
43: 51 1	GTO 1	opakování výpočtu
- obsluha funkce x!		
44: 86 2	Lbl 2	
45: 61 1	SBR 1	volání funkce $\ln(x!)$

46: -13	INV Inx	exponent, převod na x!
47: -61	INV SBR	
48: 51 2	GTO 2	opakování výpočtu

8. Vyhledání průchodu funkce nulou

Program vyhledá průchody uživatelské funkce nulou. Program obsadí programové prostory **Pgm 0** a **Pgm 1**. Uživatelská funkce je zadána v programovém prostoru **Pgm 2** a označena návěštím **Lbl 0**. Funkce může přechodně používat registr R7 (T registr) a registry R10 a výše.

Funkce nejdříve vyhledá interval s krokem $dx = (x_{\max} - x_{\min})/10$, ve kterém dojde ke změně znaménka y. Poté upřesní místo průchodu nulou dělením intervalu až do odchylky $\epsilon = dx/100000$. Velikost kroku dx lze nastavit na adrese 22. Při velké hodnotě kroku dx se může některý průchod nulou přeskočit, malá hodnota dx zpomaluje vyhledávání. Velikost odchylky ϵ lze nastavit na adrese 27. Hodnota ϵ ovlivňuje dosažitelnou přesnost vyhledání průchodu nulou za cenu zpomalení vyhledávání.

Použití:

SBR 0 ... výpočet hodnoty uživatelské funkce y pro dané x

SBR 1 ... zadání dolního limitu x_{\min}

SBR 2 ... zadání horního limitu x_{\max}

SBR 3 ... vyhledání prvního průchodu nulou

SBR 4 ... vyhledání dalšího průchodu nulou

Příklad, průchod nulou funkce $f(x) = 4 \cdot \sin(x) + 1 - x$:

Program funkce je uveden v následujícím výpisu, v prostoru **Pgm 2**.

Pgm 0 ... volba programového prostoru 0

3 +/- SBR 1 ... zadání dolního limitu $x_{\min} = -3$

3 SBR 2 ... zadání horního limitu $x_{\max} = 3$

SBR 3 [-2.2100...] ... vyhledání prvního průchodu nulou = -2.2100...

SBR 4 [-0.3421...] ... vyhledání druhého průchodu nulou = -0.3421...

SBR 4 [2.7020...] ... vyhledání třetího průchodu nulou = 2.7020...

SBR4 [9.999+99] ... vyhledání dalšího průchodu, nenalezen

Registry:

R0 ... hodnota y testované funkce

R1 ... dolní limit xmin

R2 ... horní limit xmax

R3 ... delta interval dx

R4 ... x začátku intervalu

R5 ... x konce intervalu

R6 ... aktuální x

R7 ... registr T, přechodně k dispozici pro **SBR**0

R8 ... odchylka eps (minimální dx)

R9 ... aktuální dolní limit xmin2

Program:

Pgm0 ... volba programového prostoru 0

- výpočet hodnoty uživatelské funkce

00: 86 0 Lbl 0 návěští uživatelské funkce

01: 78 2 Pgm 2 volba programového prostoru 2

02: 61 0 SBR 0 zavolání uživatelské funkce

03: -61 INV SBR

- obsluha zadání dolního limitu xmin

04: 86 1 Lbl 1 návěští funkce zadání dolního limitu xmin

05: 32 1 STO 1 uložení dolního limitu xmin

06: -61 INV SBR

- obsluha zadání horního limitu xmax

07: 86 2 Lbl 2 návěští funkce zadání horního limitu xmax

08: 32 2 STO 2 uložení horního limitu xmax

09: -61 INV SBR

- indikace chyby, další průchod nenalezen

10: 86 9 Lbl 9 návěští indikace chyby

11: 15 CLR

12: 25 1/x indikace chyby

13: 86 8 Lbl 8 návěští pro návrat z Pgm 1

14: -61 INV SBR

- obsluha hledání první nuly

15: 86 3	Lbl 3	návěští funkce hledání první nuly
16: 33 2	RCL 2	horní limit xmax
17: 65	-	
18: 33 1	RCL 1	dolní limit xmin
19: 32 9	STO 9	aktuální dolní limit xmin2
20: 85	=	- zde je vypočten rozdíl (xmax - xmin)
21: 45	:	
22: 01	1	
23: 00	0	
24: 85	=	- zde je vypočten dx = interval/9
25: 32 3	STO 3	uložení delta intervalu dx
26: 45	:	
27: 05	5	
28: -18	INV log	konstanta 100000
29: 85	=	- zde je vypočtena max. odchylka eps
30: 32 8	STO 8	uložení odchylky eps

- obsluha hledání další nuly

31: 86 3	Lbl 4	návěští funkce hledání další nuly
32: 33 2	RCL 2	horní limit xmax
33: 22	x<>t	xmax do T
34: 33 9	RCL 9	aktuální dolní limit xmin2
35: 76	x>=t	dosažen horní limit?
36: 51 9	GTO 9	další průchod nenalezen
37: 32 4	STO 4	uložení x začátek intervalu
38: 75	+	
39: 33 3	RCL 3	delta interval dx
40: 85	=	
41: 32 9	STO 9	nový aktuální dolní limit xmin2
42: 32 5	STO 5	x konce intervalu
43: 61 0	SBR 0	výpočet y na konci intervalu
44: 32 0	STO 0	úschova hodnoty y na konci intervalu
45: 33 4	RCL 4	x na začátku intervalu
46: 32 6	STO 6	uložení nového aktuálního x
47: 61 0	SBR 0	výpočet y v aktuálním x
48: 78 1	Pgm 1	volba programového prostoru 1
49: 51 9	GTO 9	pokračování funkce v prostoru 1

Pgm 1 ... volba programového prostoru 1

- pokračování obsluhy hledání další nuly

00: 86 9	Lbl 9	návěští pokračování obsluhy hledání nuly
01: 19	C.t	nulování T
02: 66	x=t	je hodnota $y = 0$?
03: 51 8	GTO 8	nalezen průchod nulou
04: 55	x	
05: 38 0	Exc 0	aktuální_y * počáteční_y
06: 85	=	
07: -76	INV x>=t	je-li násobek < 0 , nalezen průchod 0
08: 51 7	GTO 7	je-li nalezen průchod 0, upřesnění
09: 78 0	Pgm 0	přepnutí na programový prostor 0
10: 51 4	GTO 4	pokračování dalším intervalem
- smyčka upřesnění nuly		
11: 86 7	Lbl 7	návěští obsluhy upřesnění nuly
12: 33 4	RCL 4	x na začátku intervalu
13: 75	+	
14: 33 5	RCL 5	x na konci intervalu
15: 85	=	
16: 45	:	
17: 02	2	
18: 85	=	- zde je vypočten střed intervalu x
19: 32 6	STO 6	nové aktuální x
20: 33 8	RCL 8	odchylka eps
21: 22	x<>t	uložení odchylky eps do T
22: 33 5	RCL 5	x na konci intervalu
23: 65	-	
24: 33 4	RCL 4	x na začátku intervalu
25: 85	=	- zde je vypočtena délka intervalu
26: -76	INV x>=t	je interval $< \text{epsilon}$?
27: 51 6	GTO 6	interval je $< \text{epsilon}$, nula nalezena OK
28: 33 6	RCL 6	aktuální x (střed intervalu)
29: 78 2	Pgm 2	volba programového prostoru 2
30: 61 0	SBR 0	výpočet y v aktuálním x
31: 55	x	
32: 33 7	RCL 0	porovnání s předešlým y
33: 85	=	
34: 19	C.t	nulování T
35: 66	x=t	některý z y je 0 ?
36: 51 6	GTO 6	nalezena nula
37: 76	x>=t	je-li součin > 0 , není změna znaménka
38: 51 5	GTO 5	není změna znaménka, posun k vyššímu x
39: 33 6	RCL 6	aktuální x (střed intervalu)
40: 32 5	STO 5	posun dolů, střed bude novým koncem
41: 51 7	GTO 7	pokračování smyčky upřesnění nuly

42: 86 5	Lbl 5	návěští pro posun k vyššímu x
43: 33 6	RCL 6	aktuální x (střed intervalu)
44: 32 4	STO 4	posun nahoru, střed bude novým začátkem
45: 51 7	GTO 7	pokračování smyčky upřesnění nuly
46: 86 6	Lbl 6	návěští výsledku OK
47: 33 6	RCL 6	vyvolání výsledku
48: 78 0	Pgm 0	volba programového prostoru 0
49: 51 8	GTO 8	návrat z funkce

Pgm 2 ... volba programového prostoru 2

- výpočet hodnoty uživatelské funkce, příklad $4 \cdot \sin(x) + 1 - x$

00: 86 0	Lbl 0	návěští uživatelské funkce
01: 43	(
02: 32 7	STO 7	úschova hodnoty x
03: 60	Rad	přepnutí na radiány
04: 28	sin	
05: 55	x	
06: 04	4	
07: 75	+	
08: 01	1	
09: 65	-	
10: 33 7	RCL 7	
11: 44)	
12: -61	INV SBR	

9. Simpsonův integrál funkce

Program počítá numerický integrál uživatelské funkce Simpsonovou aproximací. Program je uložen v programovém prostoru **Pgm 0**, uživatelská funkce je vytvořena v programovém prostoru **Pgm 1** pod návěští **Lbl 0**. Zadaný počet kroků n musí být sudé číslo.

Použití:

- 1) **CLR RST** ... vynulování operací a reset ukazatele programu
- 2) x_{\min} **R/S** ... zadání dolní hranice
- 3) x_{\max} **R/S** ... zadání horní hranice
- 4) n **R/S** [y] ... zadání počtu kroků (sudé číslo!) a výpočet

Příklad, integrál funkce $f(x) = 1/(\cos(x) + 2)$ v rozsahu 0 až $\pi/2$:

Program funkce je uveden v následujícím výpisu v programovém prostoru **Pgm 1**, pod návěští **Lbl 0**.

Pgm 0 ... volba programového prostoru 0

CLR RST ... vynulování operací a reset ukazatele

0 R/S ... zadání dolní hranice

π : 2 = R/S ... zadání horní hranice

2 0 R/S [0.6046..] ... zadání počtu kroků (sudé číslo!) a výpočet

výsledek pro 2 kroky = 0.604998902985045 (přesnost 3 číslice)

výsledek pro 4 kroky = 0.604619708564478 (přesnost 4 číslice)

výsledek pro 10 kroků = 0.604600226682395 (přesnost 6 číslic)

výsledek pro 20 kroků = 0.604599814753009 (přesnost 7 číslic)

výsledek pro 100 kroků = 0.604599788120369 (přesnost 10 číslic)

výsledek pro 200 kroků = 0.604599788080709 (přesnost 11 číslic)

výsledek pro 1000 kroků = 0.604599788078063 (přesnost 13 číslic)

výsledek pro 2000 kroků = 0.604599788078057 (přesnost 13 číslic)

referenční výsledek = 0.604599788078073

Registry:

R0 ... počet kroků n , čítač smyčky

R1 ... dolní limit x_{\min}

R2 ... přírůstek kroku dx

R3 ... střadač výsledku integrálu y

Program:

Pgm 0 ... volba programového prostoru 0

00: 32 1	STO 1	uložení dolního limitu x_{\min}
01: 81	R/S	stop programu, čekání na horní limit x_{\max}
02: 65	-	
03: 33 1	RCL 1	dolní limit x_{\min}
04: 85	=	- zde je vypočten interval $(x_{\max} - x_{\min})$
05: 32 2	STO 2	uložení intervalu do dx
06: 81	R/S	stop programu, čekání na počet kroků
07: 32 0	STO 0	uložení čítače kroků n
08: -39 2	INV Prd 2	interval / n , výpočet přírůstku kroku dx
09: 61 9	SBR 9	výpočet aktuálního x a y
10: 32 3	STO 3	inicializace střadače výsledku integrálu I
11: 86 8	Lbl 8	návěští začátku smyčky výpočtu
12: -77 0	INV Inc 0	dekrementace čítače smyčky R0
13: 61 9	SBR 9	výpočet aktuálního x a y
14: 55	x	
15: 04	4	
16: 85	=	- vypočten $y * 4$
17: 34 3	SUM 3	přičtení ke střadači výsledku y
18: -56	INV Dsz	dekrementace čítače R0, přeskocení když > 0
19: 51 7	GTO 7	skok na konec smyčky, když R0 = 0
20: 61 9	SBR 9	výpočet aktuálního x a y
21: 55	x	
22: 02	2	
23: 85	=	- vypočten $y * 2$
24: 34 3	SUM 3	přičtení ke střadači výsledku y
25: 51 8	GTO 8	další krok smyčky
26: 86 7	Lbl 7	návěští konce smyčky
27: 61 9	SBR 9	výpočet aktuálního x a y
28: 34 3	SUM 3	přičtení ke střadači výsledku y
29: 33 2	RCL 2	přírůstek dx
30: 45	:	
31: 03	3	- počítáme přírůstek $dx / 3 *$ střadač y

32: 55	x	
33: 33 3	RCL 3	střadač výsledku y
34: 85	=	
35: 81	R/S	stop, zobrazení výsledku
36: 86 7	Lbl 9	návěští výpočtu aktuálního x a y
37: 33 1	RCL 1	dolní limit xmin
38: 75	+	
39: 33 0	RCL 0	čítač kroků
40: 55	x	
41: 33 2	RCL 2	přírutek kroku dx
42: 85	=	- vypočtena aktuální souřadnice x
43: 78 1	Pgm 1	volba programového prostoru 1
44: 51 0	GTO 0	skok na uživatelskou funkci

Pgm 1 ... volba programového prostoru 1

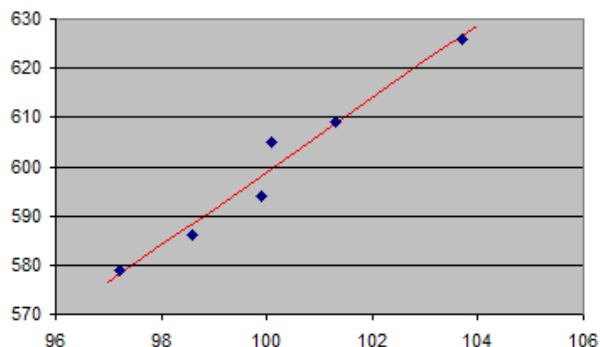
- příklad uživatelské funkce $f(x) = 1/(\cos(x) + 2)$

00: 86 0	Lbl 0	návěští začátku uživatelské funkce
01: 60	Rad	
02: 29	cos	
03: 75	+	
04: 02	2	
05: 85	=	
06: 25	1/x	
07: -61	INV SBR	

10. Lineární regresní přímka

Program počítá koeficienty aproximační lineární regresní přímky metodou nejmenších čtverců. Dvojice hodnot (X,Y) je zadána pomocí statistické funkce **Stat**. Regresní přímka má tvar $y = m \cdot x + b$.

Koeficient 'm', tedy strmost přímky, počítá podle vzorce $m = (\text{sum}(x \cdot y) - \text{sum}(x) \cdot \text{sum}(y) / N) / (\text{sum}(x^2) - \text{sum}(x)^2 / N)$. Koeficient 'b', tedy posun přímky ve směru Y, počítá podle vzorce $b = (\text{sum}(y) - m \cdot \text{sum}(x)) / N$.



Použití:

- 1) Zadejte dvojice (X,Y) pomocí funkce **Stat** (viz [88 Statistika, Stat](#))
- 2) **CLR** **RST** ... resetování ukazatele programu
- 3) **R/S** [m] ... výpočet koeficientu 'm'
- 4) **R/S** [b] ... výpočet koeficientu 'b'

Následující funkce lze volat až po výpočtu 'm' a 'b'.

x **SBR** **1** [y] ... výpočet y pro zadanou hodnotu x

y **SBR** **2** [x] ... výpočet x pro zadanou hodnotu y

Příklad:

CLR INV C.t ... vymazání operací a registrů

1 0 1 . 3 x<>t 6 0 9 Stat ... bod 1 (101.3, 609)

1 0 3 . 7 x<>t 6 2 6 Stat ... bod 2 (103.7, 626)

9 8 . 6 x<>t 5 8 6 Stat ... bod 3 (98.6, 586)

9 9 . 9 x<>t 5 9 4 Stat ... bod 4 (99.9, 594)

9 7 . 2 x<>t 5 7 9 Stat ... bod 5 (97.2, 579)

1 0 0 . 1 x<>t 6 0 5 Stat ... bod 6 (100.1, 605)

RST ... resetování ukazatele programu

R/S [7.4734...] ... výpočet koeficientu $m = 7.4734325186$

R/S [-148.506...] ... výpočet koeficientu $b = -148.5063762$

9 7 SBR 1 [576.4166...] ... pro $X = 97$ je $Y = 576.41657811$

1 0 4 SBR 1 [628.7306...] ... pro $X = 104$ je $Y = 628.73060574$

5 8 0 SBR 2 [97.4794...] ... pro $Y = 580$ je $X = 97.479488091$

6 3 0 SBR 2 [104.170...] ... pro $Y = 630$ je $X = 104.16985425$

Registry:

R0 ... počet položek N

R1 ... součet y

R2 ... součet y^2

R3 ... součet x

R4 ... součet x^2

R5 ... součet $x*y$

R7 ... registr T

R8 ... vypočtený koeficient 'm'

R9 ... vypočtený koeficient 'b'

Program:

- obsluha výpočtu 'm'

00: 33 5 RCL 5 součet $x*y$

01: 65 -

02: 33 3	RCL 3	součet x
03: 55	x	
04: 33 1	RCL 1	součet y
05: 45	:	
06: 33 0	RCL 0	počet položek N
07: 85	=	- mezivýsledek (sum(x*y) - sum(x)*sum(y)/N)
08: 45	:	
09: 43	(
10: 33 4	RCL 4	součet x^2
11: 65	-	
12: 33 3	RCL 3	součet x
13: 23	x^2	
14: 45	:	
15: 33 0	RCL 0	počet položek N
16: 85	=	
17: 32 8	STO 8	uložení koeficientu m
18: 81	R/S	zastavení programu, zobrazení m

- obsluha výpočtu 'b'

19: 33 1	RCL 1	součet y
20: 65	-	
21: 33 8	RCL 8	koeficient m
22: 55	x	
23: 33 3	RCL 3	součet x
24: 85	=	
25: 45	:	
26: 33 0	RCL 0	počet položek N
27: 85	=	
28: 32 9	STO 9	uložení koeficientu b
29: 81	R/S	zastavení programu, zobrazení b

- obsluha výpočtu y z x

30: 86 1	Lbl 1	návěští funkce výpočtu y z x
31: 43	(
32: 14	CE	
33: 55	x	
34: 33 8	RCL 8	koeficient m
35: 75	+	
36: 33 9	RCL 9	koeficient b
37: 44)	
38: -61	INV SBR	

- obsluha výpočtu x z y

39: 86 2	Lbl 2
----------	-------

40: 43	(
41: 43	(
42: 14	CE	
43: 65	-	
44: 33 9	RCL 9	koeficient b
45: 44)	
46: 45	:	
47: 33 8	RCL 8	koeficient m
48: 44)	
49: -61	INV SBR	