

CALCULATRICE PROGRAMMABLE

RPN-1250



Manuel d'utilisation

Calculatrice programmable RPN-1250
Manuel d'utilisation

Calculatrice version 4.1 © Benoit Maag

Juin 2024



HHC 2018 : Réutilisation des anciennes calculatrices TI

<https://www.youtube.com/watch?v=mxwn67G2P60>

Benoit Maag :
Réutilisation d'une TI-1250 pour en faire une calculatrice RPN-1250

Sommaire

1	Disposition du clavier	6
2	Présentation	7
3	Comment utiliser la calculatrice	8
5	La programmation	9
6	Touches et instructions	10
0..9	Chiffres          	11
1.1	Appel sous-programme	11
	 	11
	Retour sous-programme	11
	Saut vers label	12
1.2	Label (étiquette)	12
		12
	Tests	12
		12
	Recul d'un pas dans le programme	13
		13
1.3	Sauvegarde en mémoire constante	13
		13
	Chargement depuis mémoire constante	13
		13
	Avance d'un pas dans le programme	13
		13
1.4	Délai d'attente	13
		13
	Affichage registre	13
		13
	Démarrage / arrêt du programme	13
		13
2.1	Touche "shift" F	14
	Touche "shift" G	14
		14
2.2	Partie entière	14
		14
	Partie fractionnaire	15
		15
	Programmation	15
		15
2.3	Saisie en exposant de dix	15
		15
	Aucune opération	15
		15
	Changement de signe	15
		15
2.4	Nombre de décimales	16
		16
	Notation scientifique	16
		16
	Division	17
		17
3.1	Sinus	17
		17
	Arcsinus	17
		17

3.2	 	Cosinus	17
		Arccosinus	17
3.3	 	Tangente	17
		Arctangente	17
3.4	 	Enregistrement dans un registre	18
		Rappel du contenu d'un registre	19
		Multiplication	20
		Logarithme népérien	20
4.1	 	Exposant naturel	20
		Logarithme décimal	20
4.2	 	Exposant décimal	21
		Racine carrée	21
4.3		Carré d'un nombre	21
		Echange des registres X et Y	21
4.4	 	Défilement de la pile vers le bas	21
		Soustraction	21
		Inverse d'un nombre	22
5.1	 	Élévation puissance	22
		Conversion Inches vers millimètres	22
5.2	 	Conversion Millimètres vers Inches	22
		Conversion Miles vers kilomètres	22
5.3	 	Conversion Kilomètres vers miles	22
		Conversion Livres vers kilogrammes	22
5.4	 	Conversion Kilogrammes vers livres	23
		Addition	23
		Effacement de l'affichage	23
6.1	 	Effacement des données	23
		Correction de saisie	23
		PI	23
6.2	 	Conversion Degrés vers Radians	23
		Conversion Fahrenheit vers Celsius	24
6.3	 	Conversion Celsius vers Fahrenheit	24
		Séparateur décimal	24
		Last X	24
6.4	 	Validation saisie	24
			

1. Disposition du clavier

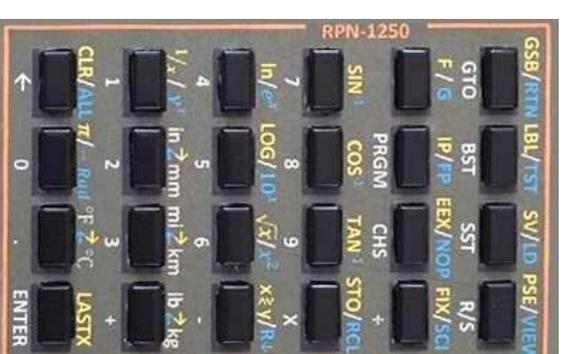
Le clavier de la **RPN-1250** est composé de 24 touches.

Les fonctions de base de chaque touche sont inscrites en blanc en dessous de la touche concernée.

Les fonctions inscrites en orange au dessus de chaque touche, à gauche, sont activées en précédant l'appui sur la touche concernée par un appui sur la touche **F**.

Les fonctions inscrites en bleu au dessus de chaque touche, à droite, sont activées en précédant l'appui sur la touche concernée par deux appuis sur la touche **F**, soit l'équivalent de **G**.

GSB RTN GTO	LBL SST BST	SV LD SST	PSE VIEW R/S
F G	IP FP PRGM	EEX NOP CHS	FIX SCI +
SIN -1 7	COS -1 8	TAN -1 9	STO RCL X
ln 4	LOG 10^x 5	√x X² 6	x^{←y} R1 -
1/x y^x 1	in → ←mm 2	mi → ←km 3	lb → ←kg +
CLR ALL ←	π →Rad 0	°F → ←°C .	LASTX Esc. ENTER



7	Exemples de programmes	25
1	Forensics	25
2	Factorielle	26
3	Fibonacci	27
4	Cercle	28
5	Stirling	29
6	Binet	30
7	PGCD	31
8	Anniversaire	32
9	Ramanujan	33
10	Trigo	34
11	Gravité	35
12	PI Day	36
13	Premier	37
14	Hilo	38
15	Fraction	39
16	Convert	40

8	Welcome to the RPN-1250 calculator © 2015 Benoit Maag	41
	Key features	42
	Run mode	43
	Program mode	44
	Stack registers / Indirect addressing	45
	Real Time Clock (RTC) <i>RPN-1250+ Only</i>	46
	Esc mode	47

2. Présentation

Utilisant une calculatrice Texas Instruments **TI-1250**, la calculatrice **RPN-1250** possède une puce Microchip flash **PIC 18F2680** de 64 Ko avec un pilote de LED **MAX7219** à 8 chiffres de 7 segments. Le logiciel est programmé en C avec l'IDE Microchip **MPLAB X**.

La calculatrice **RPN-1250** dispose de 98 pas de programmes et 20 registres en mémoire volatile.

Ces programmes et registres peuvent être sauvegardés dans trois emplacements de mémoire constante en mode programmation.

Caractéristiques :

- Pile RPN scientifique à 4 niveaux avec conversions,
 - 20 mémoires (0 à 9 et .0 à .9) avec stockage et rappel arithmétique,
 - 29 mémoires adressables en indirect.
 - Enregistrement possible de la pile et des registres dans la mémoire flash «constante» du PIC,
 - 98 pas programmables (comme le HP-29C) avec affichage alphanumérique du programme,
 - 3 zones de sauvegarde de programmes dans la mémoire flash «constante» du PIC,
 - 20 étiquettes (0 à 9 et .0 à .9),
 - sous-programmes (GSB, RTN),
 - 12 tests possibles (X=0, X<>0, X=y, X<>y, ...),
 - fonctions PAUSE et VIEW,
 - exécution pas à pas (SST),
 - conversions (in<>mm, mi<>km, lb<>kg, °F<>°C),
 - Vitesse environ 8 fois supérieure à une **HP-41C**...
- Ecart par rapport à la calculatrice HP-29C :
- Pas d'incrément ni de décrétement (ISZ, DSZ)
 - Pas de valeur absolu (ABS) [remplaçable par x<0 ? CHS]
 - Pas de conversions polaire/rectangulaire

4. Comment utiliser la calculatrice

La calculatrice **RPN-1250** est équipée d'un écran LED (alpha) numérique d'une seule ligne de 8 caractères.

L'alimentation est fournie par une pile 6LR61 de 9 Volts et la calculatrice s'allume et s'éteint grâce à l'interrupteur situé sur sa gauche :

- **ON** : position vers le haut,
- **OFF** : position vers le bas.

3 modes d'utilisation sont possibles :

• Le mode "exécution"

Le mode dans lequel la calculatrice s'utilise pour faire des calculs et des conversions est le mode d'exécution du programme chargé en mémoire volatile.

La pile et les registres utilisés étant en mémoire volatile sont réinitialisés à chaque allumage de la calculatrice mais peuvent être conservés en mémoire constante.

F SV permet la sauvegarde de la pile et des registres en mémoire constante.

G LD permet de recharger la pile et les registres depuis la mémoire constante.

L'exécution d'un programme chargé en mémoire volatile se fait - soit par positionnement en début de mémoire (pas 00) via **RTN** puis **R/S** pour lancer l'exécution (après avoir saisi éventuellement les données requises par le programme)

- soit par positionnement au label de départ du programme via **STO** suivi du label concerné (0 à 9 ou .0 à .9) puis **R/S** pour lancer l'exécution (après avoir saisi éventuellement les données requises par le programme)

- soit en lançant directement le programme via **GSB** suivi du label concerné (0 à 9 ou .0 à .9) (après avoir saisi éventuellement les données requises par le programme).

Un programme peut être aussi exécuté pas à pas pour en vérifier le bon fonctionnement.

Le lancement se fait dans ce cas en se positionnant soit par **RTN** ou par **STO** et en utilisant **SST** pour avancer pas à pas.

• Le mode "programme"

est le mode dans lequel les programmes sont saisis et peuvent être modifiés.

• Le mode "Escape"

⇒ est le mode dans lequel la luminosité peut être réglée.   

⇒ un test d'affichage aussi peut être exécuté. 

⇒ et l'ensemble des caractères alphanumériques peut être visualisé. 

5. La programmation

L'écriture d'une séquence de touches dans la mémoire programme s'appelle un programme. Le programme transforme la calculatrice en un outil puissant.

La mémoire de programme se compose de 4 zones de programme indépendantes :

- La mémoire de travail qui est la mémoire de saisie et d'exécution de programme. Cette mémoire est volatile et s'efface à l'extinction de la calculatrice.
- 3 mémoires de sauvegarde dans lesquelles la mémoire de travail peut être sauvé. Ces 3 mémoires sont constantes et conservées lors de l'extinction de la calculatrice.

Chaque zone de programme contient 98 pas de programme, donc un total de 294 pas de programme sont disponibles en réserve.

20 étiquettes (labels), numérotées de 0 à 9 et de .0 à .9, peuvent être utilisées.

Le mode de programmation est activé avec la touche **PRGM**.

Le contenu du programme est affiché en **alphanumérique** sur la ligne de l'afficheur sous la forme numéro de pas suivi de l'intitulé de l'instruction.

Espace	'	"	#	\$	%	&	'
()	*	+	0	-	.	/
C	3	0	1	1	-	.	2
0	1	2	3	4	5	6	7
8	9	:	;	<	=	>	?
B	9	-	1	L	=	7	2
@	A	B	C	D	E	F	G
P	R	b	c	d	E	F	G
H	I	J	K	L	M	N	O
h	,	J	f	L	n	n	o
P	Q	R	S	T	U	V	W
P	q	r	s	t	u	v	w
X	Y	Z	[\]	^	_
11	12	13	14	15	16	17	18

Touches et fonctions utiles à la programmation :

SST (Single Step) Incrémente le pointeur de programme de 1 ("pas suivant").

La touche **SST** peut également être utilisé en mode "exécution". Dans ce mode, après avoir appuyé sur **SST**, le code de l'instruction courante est exécuté et le pointeur de programme se positionne sur le pas suivant. (test pas à pas d'un programme)

BST (Back Step) Décrémente le pointeur de programme de 1 ("pas précédent").

↵ (Delete) Supprime l'instruction à la position courante du programme et affiche la partie suivante du programme.

no (Insert) L'insertion d'une instruction se fait automatiquement **après** l'instruction courante avec décalage des instructions suivantes et ne nécessite donc pas de touche spécifique d'insertion.

PRGM (Program) En mode "exécution" **PRGM** passe au mode programmation. En mode programmation, **PRGM** quitte le mode d'édition de programme et ramène la calculatrice en mode d'exécution.

GTO (Go To) L'instruction **GTO** est normalement utilisé pour déplacer le pointeur de programme vers l'étiquette spécifiée de 0 à 9 ou .0 à .9.

Dans le mode "programmation", l'instruction ne peut pas être utilisée pour déplacer le pointeur, car le code correspondant serait stocké dans le programme. Il faut donc quitter le mode programmation en appuyant sur **PRGM**, effectuer un saut **GTO** vers le label spécifié 0 à 9 ou .0 à .9 et revenir en mode programmation en appuyant sur **PRGM**.

RTN (Return) Ordre de retour après l'exécution d'un sous-programme appelé par **GSB**. **RTN** se place en fin de sous-programme.

Mais **RTN** peut être utilisé en mode d'exécution pour ramener le pointeur de programme à l'adresse 0.

R/S (Run/Stop) Démarre le programme ou arrête le programme (utilisé en mode exécution).

Signifie "fin de programme" en mode programmation.

6. Touches et instructions

Chaque instruction a une adresse de pas suivi d'un intitulé correspondant à une séquence d'appuis sur une ou plusieurs touches pour l'exprimer.

00...09 0..9 - Chiffres de base

Les chiffres de base sont utilisés pour entrer des chiffres dans la plage de 0 à 9. Ils sont utilisés pour entrer la mantisse d'un nombre, entrer l'exposant, un numéro de registre de mémoire ou un numéro d'étiquette.

Les numéros sont stockés dans le programme avec un code de 0 à 9.

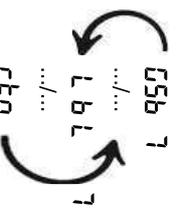
1.1 G5b - Appel sous-programme

La touche **GSB** (GoSub) est utilisé pour appeler un sous-programme en utilisant en paramètre le code numérique de 0 à 9 ou .0 à .9 du label appelé.

Si l'instruction **GSB** est utilisée en mode exécution, le sous-programme est exécuté immédiatement.

Un sous-programme se termine par l'instruction **RTN** pour assurer le retour du sous-programme vers le programme appelant.

Exemple :



rLn - Retour sous-programme

Un sous-programme appelé via **GSB** se termine par l'instruction **RTN** qui assure le retour du sous-programme juste après l'instruction **GSB** appelante.

En mode exécution **RTN** positionne le pointeur de programme sur le pas 0.



Gto - Saut vers label

GTO permet d'effectuer un saut inconditionnel dans un programme. Il a comme paramètre un code numérique de 0 à 9 ou .0 à .9 correspondant à un label (**Lbl**) du programme.

Lorsque l'instruction **GTO** est utilisée en mode exécution, le pointeur de programme est positionné sur l'étiquette correspondante.

Exemple :



1.2 Lbl - Label (étiquette)

L'instruction **Lbl** peut être utilisée pour marquer une début de séquence dans le programme comme une étiquette.

20 étiquettes peuvent être utilisées, notées de **Lbl 0** à **Lbl 9** et de **Lbl .0** à **Lbl .9**.

Le numéro de l'étiquette est spécifié comme paramètre numérique 0 à 9 ou .0 à .9 de l'instruction **Lbl**.

Vous pouvez sauter à l'endroit du programme marqué d'une étiquette en utilisant l'instruction de saut **GTO** ou l'instruction d'appel de sous-programme **GSB**.

If=0 ... If<=Y - Tests

Les instructions de test permettent de comparer le registre **X** (contenu d'affichage) soit avec la valeur zéro (0) soit avec le registre **Y**.

Si le test est satisfait, l'instruction suivant l'instruction de test est exécutée; sinon, la commande suivant le test est ignorée et l'exécution se poursuit après.

Les tests comparatifs sont :

comparaison entre **X** et zéro

  7	X=0	If=0
  8	X<>0	IfL 70
  4	X>0	If70
  3	X>=0	If7=0
  1	X<0	IfL0
  2	X<=0	IfL=0

comparaison entre **X** et **Y**

  7	X=Y	If=Y
  8	X<>Y	IfL 7Y
  4	X>Y	If7Y
  3	X>=Y	If7=Y
  1	X<Y	IfL Y
  2	X<=Y	IfL=Y



55t - Recul d'un pas dans le programme

La touche **BST** (*Back Step*) en mode programmation retourne au pas précédent.



5u - Sauvegarde en mémoire constante

La fonction **SV** (*Save*) permet la sauvegarde en mémoire constante (ou mémoire continue) des registres et de la pile en mode exécution.

En mode programmation, les programmes peuvent être sauvegardés dans 3 zones de mémoire constante au choix en précisant 1, 2 ou 3 derrière la commande **SV**.



Ld - Chargement depuis mém. constante

En mode exécution la fonction **LD** (*Load*) permet de recharger les registres et la pile sauvegardés dans la mémoire constante (ou mémoire continue).

En mode programmation, un programme peut être rechargé depuis une des 3 zones de mémoire constante au choix en précisant 1, 2 ou 3 derrière la commande **LD**.



55t - Avance d'un pas dans le programme

La touche **SST** (*Single Step*) avance d'un pas en mode de programmation.

En mode exécution, l'instruction de programme, sur laquelle le pointeur est positionné, est exécutée, ce qui permet d'exécuter le programme pas à pas à des fins de débogage.

Attention : dans ce cas de test pas à pas du programme si un sous-programme est appelé, le retour du sous-programme (**RTN**) se produit comme en mode exécution et renvoie au pas 0.

1.4 F PSE PRIJSE - Délai d'attente

La commande **Pause** arrête l'exécution du programme brièvement et affiche le contenu du registre **X** le temps de la pause.



u E - Affichage registre

Dans un programme, la commande **VIEW** affiche le contenu d'un registre sans arrêter l'exécution du programme.



r rS - Démarrage/arrêt du programme

La touche **R/S** (*Run/Stop*) peut être utilisé pour démarrer ou arrêter un programme en cours.

Au démarrage, le programme commence à s'exécuter à partir du pas de programme courant (l'adresse en cours peut être trouvée en passant au mode de programmation **PRGM**).

En mode programmation **R/S** indique l'arrêt du programme.



2.1 Touche "shift" F - Fonction alternative

La touche **F** est utilisé pour changer la signification de la touche suivante en une fonction alternative.

Après avoir appuyé sur **F**, la fonction alternative (en orange) du bouton suivant est alors exécutée.

Un deuxième appui sur **F** active la fonction alternative **G**.

Un troisième appui sur **F** annule les précédents appuis sur **F**.

L'intitulé de la touche **F** n'est pas enregistré dans le programme, c'est l'intitulé alternatif de la touche suivante qui est alors affiché.

Exemple:

s'obtient en réalité en appuyant sur et affiche **Ln**



Touche "shift" G - Fonction alternative

La touche **G** est utilisé pour changer la signification de la touche suivante en une fonction alternative.

Cette fonction **G** est une fonction alternative de la touche **F**.

Après deux appuis successifs sur **G**, la fonction alternative (en bleu) du bouton suivant est alors exécutée.

Un troisième appui sur **G** annule les précédents appuis sur **G**.

L'intitulé de la touche **G** n'est pas enregistré dans le programme, c'est l'intitulé alternatif de la touche suivante qui est alors affiché.

Exemple:

s'obtient en appuyant sur soit en réalité sur

2.2 F IP , nL - Partie entière

La touche **IP** (*Integer Part*) est utilisée pour supprimer les chiffres après la virgule décimale du nombre et pour réduire le nombre à un nombre entier.

La fonction a la même signification que l'arrondi vers zéro.

Exemple :

... partie entière de 2.3 [2]
 ... partie entière de -2.3 [-2]

G **FP** *F_rR_c* - Partie fractionnaire

La touche **FP** (*FRActionnal/Part*) est utilisée pour supprimer les chiffres avant la virgule décimale du nombre et pour réduire le nombre à un nombre fractionnaire.

Exemple :

2	.	3	FP	...	partie fractionnaire de 2.3 [.3]
2	.	3	CHS	FP	... partie fractionnaire de -2.3 [-.3]

PRGM *Pr_rG_n* - Programmation

PRGM active ou désactive le mode de programmation.

2.3 **F** **EE_X** *EE₁₁* - Saisie en exposants de dix

La fonction **EE_X** permet de saisir un nombre multiplié par une puissance de 10.

Si la touche est enfoncée lors de la saisie d'un nombre, cette saisie affiche l'exposant à 00 dans l'attente de son entrée.

Si le registre **X** (affichage) est à zéro l'appui sur **EE_X** donne 1 comme valeur de registre **X** et affiche l'exposant à 00 dans l'attente de son entrée.

L'exposant peut être négatif en utilisant la fonction **CHS**.

Dans le cas de dépassement de capacité la calculatrice affiche **ouÉrFlou**

G **NOP** *noP* - Aucune opération

La commande **Nop** (*No Operation*) est une commande "vide" qui n'effectue aucune opération. Elle sert uniquement à remplir un pas inutilisé dans le programme.

CHS *chS* - Changement de signe

La touche **CHS** change le signe du nombre sur l'affichage.

Son utilisation pendant la saisie de l'exposant d'un nombre (puissance de dix) change le signe de cet exposant.

2.4 **F** **FIX** *F₁₁* - Nombre de décimales

À l'aide de la touche **FIX**, le nombre affiché à l'écran est arrondi au nombre de décimales spécifié. Le nombre 0 à 6 est saisi en paramètre, représentant le nombre de décimales après la virgule 0 à 6.

En mode arrondi, le nombre est complété à partir de la droite par des zéros, jusqu'au nombre spécifié de décimales.

L'arrondi n'affecte que l'affichage du nombre. En interne, le nombre (registre **X**) continue à être mémoriser en entier.

Le mode d'arrondi défini affecte également la manière dont les très petits nombres sont affichés.

Si le nombre de décimales à afficher ne concerne que des décimales à zéro, le nombre est affiché en puissances de 10 négatives.

Exemples :

0	.	0	0	1	2	3	FIX	6	affiche	0.00 1230
							FIX	3		0.00 1
							FIX	2		1.23 -03
							FIX	3		0.00 123

G **SCI** *Sc₁* - Notation scientifique

À l'aide de la touche **SCI**, le nombre affiché à l'écran est affiché en puissance de dix arrondi au nombre de décimales spécifié. Le nombre 0 à 3 est saisi en paramètre, représentant le nombre de décimales après la virgule 0 à 3.

En mode arrondi, le nombre est complété à partir de la droite par des zéros, jusqu'au nombre spécifié de décimales.

L'arrondi n'affecte que l'affichage du nombre. En interne, le nombre (registre **X**) continue à être mémoriser en entier.

Exemples :

0	.	0	0	1	2	3	FIX	6	affiche	0.00 1230
							SCI	3		1.230-03
							SCI	0		1. -03
							SCI	2		1.23 -03
							FIX	3		0.00 123



÷ - Division

Le signe \div permet de diviser le premier opérande (dans la pile) par le deuxième opérande (dans le registre **X**) selon le principe de la notation polonaise inversée (RPN).

Exemple :

Division de 2.2 par 0.5

= 4.4

3.1 **Si, n - Sinus**

La fonction **sin** calcule le sinus d'un angle en radian.

L'angle doit être saisi en radian.

Si l'angle est en degré il devra d'abord être converti en radian par la fonction **Deg-Rad**.

Exemple :

$\sin(9^\circ)$

donne 0.156434

R5, n - Arcsinus

La fonction **sin-1** (Arcsinus) calcule l' arcsinus d'un angle en radian.

L'angle doit être saisi en radian.

Si l'angle est en degré il devra d'abord être converti en radian par la fonction **Deg-Rad**.

3.2 **co5 - Cosinus**

La fonction **cos** calcule le cosinus d'un angle en radian.

L'angle doit être saisi en radian.

Si l'angle est en degré il devra d'abord être converti en radian par la fonction **Deg-Rad**.

Rco5 - Arccosinus

La fonction **cos-1** (Arccosinus) calcule l' arccosinus d'un angle en radian.

L'angle doit être saisi en radian.

Si l'angle est en degré il devra d'abord être converti en radian par la fonction **Deg-Rad**.

3.3 **LRn - Tangente**

La fonction **tan** calcule la tangente d'un angle en radian.

L'angle doit être saisi en radian.

Si l'angle est en degré il devra d'abord être converti en radian par la fonction **Deg-Rad**.

RLRn - Arctangente

La fonction **tan-1** (Arctangente) calcule l' arctangente d'un angle en radian.

L'angle doit être saisi en radian.

Si l'angle est en degré il devra d'abord être converti en radian par la fonction **Deg-Rad**.

3.4 **Sto - Enregistrement dans un registre**

STO (Store) permet de stocker le nombre affiché dans le registre de données **0** à **9** ou **.0** à **.9**.

Le numéro de registre de 0 à 9 ou .0 à .9 est entré comme paramètre d'instruction.

STO (Store) est aussi utilisable de manière "arithmétique" en ajoutant un opérateur avant le numéro de registre (0 à 9 ou .0 à .9) entré comme paramètre.

Exemple :

ajoute 2 au contenu du registre 7
 multiplie le contenu du registre .1 par 4

STO (Store) permet de modifier le contenu d'un registre de la pile **X**, **Y**, **Z** ou **T**.

Attention : l'introduction d'un nombre avant le STO décale la pile !

Les opérateurs arithmétiques peuvent aussi être utilisés.

<input type="text" value="STO"/>	<input type="text" value="STO"/>
<input type="text" value="STO"/>	<input type="text" value="BST"/>
<input type="text" value="STO"/>	<input type="text" value="SST"/>
<input type="text" value="STO"/>	<input type="text" value="R/S"/>

affecte le registre **X** de la pile.
affecte le registre **Y** de la pile.
affecte le registre **Z** de la pile.
affecte le registre **T** de la pile.

En plus des fonctions de stockage direct dans les registres 0 à 9, .0 à .9 et pile, **STO** (Store) peut aussi stocker des données de manière indirecte dans des registres de 1 à 29, le registre 0 étant utilisé comme index.

Ces 29 registres sont les registres "standards" ou "primaires" pour les 15 premiers qui correspondent donc aux registres de R0 à R-5, et des registres indirects pour les 14 suivants utilisables uniquement en adressage indirect.

La touche pour signifier "index" est la touche

Exemple :

stocke 7 comme valeur d'index dans le registre 0,
 puis stocke la valeur 24 dans le registre indirect (R7)
 multiplie par dix le contenu du registre dont l'index est stocké dans le registre 0.

Registres primaires	Registres indirects
R.1	R(16)
R.2	R(17)
R.3	R(18)
R.4	R(19)
R.5	R(20)
R.6	R(21)
R.7	R(22)
R.8	R(23)
R.9	R(24)
R.0	R(25)
R.1	R(26)
R.2	R(27)
R.3	R(28)
R.4	R(29)
R.5	15

G **RCL**

rcl - Rappel du contenu d'un registre

RCL (Recall) est utilisé pour rappeler un nombre du registre de données **0 à 9** ou **.0** à **.9** vers l'affichage.

Le numéro de registre de 0 à 9 ou .0 à .9 est entré comme paramètre d'instruction.

RCL (Recall) est aussi utilisable de manière "arithmétique" en ajoutant un opérateur avant le numéro de registre (0 à 9 ou .0 à .9) entré comme paramètre.

Exemple :

2 **RCL** **+** **7** rappelle le contenu du registre 7 et ajoute 2
4 **RCL** **×** **1** rappelle le contenu du registre .1 et multiplie par 4

Attention : cette opération arithmétique n'affecte pas le contenu du registre mais seulement la valeur affichée.

RCL (Recall) permet de rappeler le contenu d'un registre de la pile **X**, **Y**, **Z** ou **T**.

Attention : le nombre affiché décale la pile !

Les opérateurs arithmétiques peuvent aussi être utilisés.

RCL **STO**
RCL **BST**
RCL **SST**
RCL **R/S**

rappelle le registre **X** de la pile.
rappelle le registre **Y** de la pile.
rappelle le registre **Z** de la pile.
rappelle le registre **T** de la pile.

En plus des fonctions de rappel direct depuis les registres 0 à 9, .0 à .9 et depuis la pile, **RCL (Recall)** peut aussi rappeler des données de manière indirecte depuis des registres de 1 à 29, le registre 0 étant utilisé comme index.

Ces 29 registres sont les registres "standards" ou "primaires" pour les 15 premiers qui correspondent donc aux registres de R0 à R.5, et des registres indirects pour les 14 suivants utilisables uniquement en adressage indirect.

La touche pour signifier "Index" est la touche **INDEX**.

Exemple :

7 **STO** **0** stocke 7 comme valeur d'index dans le registre 0,
RCL **INDEX** puis rappelle la valeur contenue dans le registre Indirect (R7)
1 **0** **RCL** **×** **INDEX** rappelle le contenu du registre dont l'index est stocké dans le registre 0 et multiplie l'affichage par 10.

Registres primaires	Registres indirects
R.1 1	R(16) 16
R.2 2	R(17) 17
R.3 3	R(18) 18
R.4 4	R(19) 19
R.5 5	R(20) 20
R.6 6	R(21) 21
R.7 7	R(22) 22
R.8 8	R(23) 23
R.9 9	R(24) 24
R.0 10	R(25) 25
R.1 11	R(26) 26
R.2 12	R(27) 27
R.3 13	R(28) 28
R.4 14	R(29) 29
R.5 15	

×

9 - Multiplication

Le signe **X** permet de multiplier le premier opérande (dans la pile) par le deuxième opérande (dans le registre **X**) selon le principe de la notation polonaise Inversé (RPN).

Exemple :

Multiplication de 2.2 par 0.5
2 **.** **2** **ENTER** **0** **.** **5** **×** **↓** **1.1**

4.1 **F** **ln**

Ln - Logarithme népérien

In calcule le logarithme népérien du nombre affiché.

Ce logarithme naturel utilise la constante d'Euler comme base avec la valeur 2.718281828459 pour être calculé.

L'argument de la fonction **ln** doit être un nombre positif non nul.

Dans le cas de zéro ou de nombre négatif, l'affichage affichera la valeur **dt Error** (*data error*), comme indication d'erreur.

Exemple :

5 **ln** calcule le logarithme népérien de 5 soit **1.609438**

G **ex**

EIP - Exposant naturel

L'exposant naturel, est calculée à partir de la constante d'Euler (valeur 2.718281828459) élevée à la puissance X.

Exemple :

5 **ex** calcule l'exposant naturel de 5 soit **148.4131**

4.2 **F** **LOG**

Log - Logarithme décimal

LOG calcule le logarithme décimal du nombre affiché.

L'argument de la fonction **LOG** doit être un nombre positif non nul.

Dans le cas de zéro ou de nombre négatif, l'affichage affichera la valeur **dt Error** (*data error*), comme indication d'erreur.

Exemple :

5 **LOG** calcule le logarithme décimal de 5 soit **0.698970**

G **10^x** **RloU** - Exposant décimal

10^x calcule l'exposant décimal du nombre **X** affiché, soit 10 élevée à la puissance **X**.

Exemple :

5 **10^x** calcule l'exposant décimal de 5 = **100000**.

4.3 **F** **√x** **Sqr-t** - Racine carrée

√x (**SQR**) permet de calculer la racine carrée d'un nombre. Le nombre ne doit pas être négatif.

Dans le cas d'un nombre négatif, l'affichage affichera la valeur **dE Error** (data error), comme indication d'erreur.

G **X²** **Sqr-** Carré d'un nombre

La fonction **X²** calcule le carré d'un nombre, ou le multiple d'un nombre par lui-même.

4.4 **F** **x↔y** **SwRP** - Echange des registres X et Y

Avec la touche **x↔y**, il est possible de permuter les registres **X** et **Y**.

Le registre **X** est le registre de travail et également le contenu de l'affichage.

Le registre **Y** est le registre précédent du registre **X** dans la pile RPN.

G **R1** **rdown** - Défilement de la pile vers le bas

Avec la touche **R down**, il est possible de faire défiler les registres de la pile en les permutant en cascade.

Exemple :

Pile	R1	R1	R1
T	1	4	3
Z	2	1	4
Y	3	2	1
X	4	3	2



= **-** - Soustraction

Le signe **-** permet de soustraire le deuxième opérande (dans le registre **X**) du premier opérande (dans la pile) selon le principe de la notation polonaise inversée (RPN).

Exemple :

Soustraction de 0.5 de 2.2

2 **.** **2** **ENTER** **0** **.** **5** **=** **1.7**

5.1 **F** **1/x** **f/y1** - Inverse d'un nombre

La fonction **1/x** permet de calculer l'inverse d'un nombre.

Si le nombre est zéro, l'affichage affichera la valeur **dE Error** (data error), comme indication d'erreur.

G **y^x** **r** - Puissance

L'instruction **y^x** élève le premier opérande **Y** (dans la pile) à la puissance exprimée par le deuxième opérande **X** (affichage dans le registre **X**).

Exemple :

3 **ENTER** **7** **y^x** élévation de 3 à la puissance 7 soit **2187**

5.2 **F** **in→** **, n-ññ** - Conversion Inches-Millimètres

La fonction **in→mm** permet de convertir des pouces (*Inches*) en millimètres. 1 " = 25.4 mm

Le pouce est une unité de longueur utilisée dans le système d'unités de mesure anglo-saxonnes, représentant 1/12 d'un pied.

G **mm←** **ññ-, n** - Conversion Millimètres-Inches

La fonction **mm←in** permet de convertir des millimètres en pouces (*Inches*). 1 mm = 0.039370 "

Le millimètre est une unité de longueur utilisée dans le système métrique, équivalent à un millièème de mètre.

5.3 **F** **mi→** **ñi-tñ** - Conversion Miles-Kilomètres

La fonction **mi→km** permet de convertir des miles en kilomètres.

1 mi = 1.60934 km

Le mile est une unité de longueur utilisée dans le système d'unités de mesure anglo-saxonnes, équivalent à 5 280 pieds ou à 1 760 yards.

G **km←** **tñ-ñi** - Conversion Kilomètres-Miles

La fonction **km←mi** permet de convertir des kilomètres en miles.

1 km = 0.62137 mi

Le kilomètre est une unité de longueur utilisée dans le système métrique, équivalent à 1000 mètres.

5.4 **F** **lb→** **lb-tU** - Conversion Livres-Kilogrammes

La fonction **lb→kg** permet de convertir des livres en kilogrammes.

1 lb = 0.45359 Kg

La livre est une unité de poids utilisée dans le système d'unités de mesure anglo-saxonnes, équivalent à 16 Onces (OZ).

7. Exemples de programmes

1. Forensics

Test classique de calculatrice pour tester la précision du calcul.

Cet algorithme "forensics" (*médecine légale*) inventé par Mike Sebastian pour fournir rapidement une comparaison de la précision des calculatrices scientifiques applique le calcul suivant :

$$\arcsin(\arccos(\arctan(\tan(\cos(\sin(9))))))$$

ou

$$9 \sin \cos \tan \operatorname{atan} \operatorname{acos} \operatorname{asin}$$

Utilisation :



... lance le calcul

Programme :

1	Lbl 0	étiquette de début de programme
2	F,11 9	fixe les décimales au maximum (6 en réalité)
3	9	
4	DEG-rR	convertit en radian l'angle exprimé en degré
5	S, n	sinus
6	cos	cosinus
7	LRn	tangente
8	RLRn	arctangente
9	Rcos	arccosinus
10	RS, n	arcsinus
11	1	
12	8	
13	0	> convertit l'angle obtenu en degré pour comparaison
14	9	
15	P, 1	
16	r	
17	rtn	

Résultat : **8.999996 1**

2. Factorielle

Calcul de la factorielle d'un nombre

Utilisation :

n nombre pour lequel il faut calculer la factorielle



... lance le calcul de factorielle

Programme :

1	Lbl 1	étiquette de début de programme
2	F,11 0	fixe les décimales à 0
3	Sto 1	stocke le nombre n dans le registre 1
4	1	stocke la valeur 1
5	Sto 2	dans le registre 2
6	Lbl 9	étiquette pour boucle itérative
7	rcl 1	rang du calcul
8	Sto 2	qui multiplie le résultat
9	1	1
10	St- 1	décrémente le rang du calcul
11	rcl 1	rappel de la valeur du rang
12	IL 0	si différent de zéro
13	Gto 9	alors retour au début de boucle
14	rcl 2	rappel résultat
15	rtn	fin

Résultat : **n !**

3. Fibonacci

Calcule un nombre de Fibonacci de rang n

Utilisation :

n rang pour lequel il faut rechercher le nombre de Fibonacci

GSB **2** ... lance la recherche

Programme :

1	Lbl	2	étiquette de début de programme
2	Sto	.0	stocke le nombre n dans le registre .0
3	I		stocke la valeur 1
4	Sto	.1	dans le registre .1
5	0		
6	Sto	.2	étiquette pour boucle itérative
7	Lbl	7	
8	rcl	.0	1
9	I		décrémente le rang du calcul
10	-		rappel de la valeur du rang
11	Sto	.0	si égal de zéro
12	If=0		alors nombre trouvé donc fin
13	Gto	8	
14	rcl	.1	
15	rcl	.2	
16	I		
17	Sto	.3	> calcul du nombre
18	rcl	.1	
19	Sto	.2	
20	rcl	.3	
21	Sto	.1	itération suivante
22	Gto	7	
23	Lbl	8	
24	rcl	.3	rappel du résultat
25	F.II	0	fixe les décimales à 0
26	rP5		fin du programme

4. Cercle

Calcule le périmètre et la surface d'un cercle à partir du rayon

Utilisation :

r rayon du cercle

GSB **5** calcule et affiche le périmètre

R/S calcule et affiche la surface

Programme :

1	Lbl	5	étiquette de début de programme
2	F.II	2	fixe les décimales à 2
3	Sto	0	stocke le rayon r dans le registre .0
4	2		
5	0		> calcule le périmètre $r \times 2 \times \pi$
6	P1		
7	0		
8	rP5		rappelle le rayon r du registre .0
9	rcl	.0	
10	Sqr		
11	P1		> calcule la surface $r^2 \times \pi$
12	0		
13	rP5		
14	Gto	5	

5. Stirling

La formule de Stirling (James Stirling, mathématicien écossais, né en mai 1692 à Garden près de Stirling et mort le 5 décembre 1770 à Edimbourg) permet d'approcher la factorielle d'un nombre.

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

Cette formule améliorée donnera une meilleure approche :

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \frac{1}{12n}\right)$$

Lorsque la fonction i (factorielle) n'existe pas sur une calculatrice, ce calcul de factorielle d'un nombre est habituellement fait sur les calculatrices programmables grâce à une boucle itérative.

Ce genre de calcul peut être très peu coûteux en termes de nombre de pas mais excessif en temps pour des grands nombres.

Par contre, la formule de Stirling donne une approximation du résultat très rapidement mais coûte quelques pas de programmes. (Voir programme factorielle page 24)

Utilisation :

n nombre pour lequel il faut calculer la factorielle



... lance le calcul de factorielle

Programme :

```
1 Lbl Y
2 StO .0
3 2
4 0
5 P1
6 0
7 SqrL
8 StO .1
9 rCL .0
10 1
11 EHP
12 P
13 rCL .0
```

```
14 n
15 rCL .1
16 0
17 rCL .0
18 HP1
19 1
20 2
21 P
22 1
23 1
24 0
25 F11 .0
26 rP5
```

6. Binet

La formule de Binet (Jacques Philippe Marie Binet, mathématicien et astronome français, né à Rennes le 2 février 1786 et mort à Paris le 12 mai 1856) fournit le n-ième terme de la suite de Fibonacci.

$$F_n = \frac{1}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1 - \sqrt{5}}{2} \right)^n$$

Le calcul du n-ième terme de la suite de Fibonacci est habituellement fait sur les calculatrices programmables grâce à une boucle jusqu'à n. (Voir programme Fibonacci page 25)

Ce genre de calcul peut être très peu coûteux en termes de nombre de pas mais excessif en temps pour des valeurs de n élevées.

Par contre, la formule de Binet donne le résultat très rapidement mais coûte de nombreux pas de programmes.

Utilisation :

n rang pour lequel il faut rechercher le nombre de Fibonacci



... lance le calcul

Programme :

```
1 Lbl 3
2 StO 1
3 5
4 SqrL
5 1
6 1
7 2
8 P
9 rCL 1
10 n
11 5
12 SqrL
13 P
14 StO .1
15 5
16 SqrL
```

```
17 1
18 -
19 2
20 P
21 rCL 1
22 n
23 5
24 SqrL
25 P
26 StO .2
27 rCL .1
28 rCL .2
29 -
30 F11 .0
31 rP5
```

7. PGCD

Un des petits programmes classiques de la programmation des calculatrices...

Beaucoup de programmeurs ont débuté avec ces petits programmes dont l'utilité était de s'initier au langage de la calculatrice nouvellement acquise.

Utilisation :

n1 premier nombre

ENTRÉE

n2 deuxième nombre

GSB

0

calcul et affiche le PGCD

Programme :

```

1 Lbl 6
2 Sto 2
3 rDown
4 rDown
5 Sto 1
6 Lbl 7
7 rcl 1
8 rcl 2
9 Sto 3
10 r
11 nT
12 rcl 2
    
```

```

13 9
14 ch5
15 rcl 1
16 4
17 Sto 2
18 rcl 3
19 Sto 1
20 rcl 2
21 HLT
22 Sto 7
23 rcl 1
24 rP5
    
```

8. Anniversaire

Le paradoxe des anniversaires calcule le pourcentage de chance de trouver 2 personnes ayant la même date d'anniversaire (pas forcément nées la même année) dans un groupe de n personnes.

$$p(n) = 1 - \frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \dots \cdot \frac{365 - n + 1}{365}$$

Pour simplifier, la formule retenue suppose que toutes les années sont non bissextiles. *Considérer les années bissextiles changerait peu les résultats des calculs, mais rendrait les programmes plus compliqués.*

Utilisation :

n nombre de personnes

GSB

0

... lance le calcul du pourcentage

Programme :

```

1 Lbl 0
2 Sto 4
3 3
4 6
5 5
6 nPI
7 Sto 1
8 1
9 Sto 2
10 Sto 3
11 Lbl 1
12 1
13 St- 4
14 rcl 4
15 HLT
16 Sto 2
    
```

```

17 rcl 1
18 St- 2
19 rcl 2
20 St- 3
21 Sto 1
22 Lbl 2
23 F.H
24 1
25 rcl 3
26 -
27 1
28 0
29 0
30 0
31 rP5
    
```

9. Ramanujan

La formule de Ramanujan permet de calculer la factorielle d'un nombre n.

$$n! \sim \sqrt{\pi} \left(\frac{n}{e}\right)^n \sqrt{8n^3 + 4n^2 + n + \frac{1}{30}}$$

(Srinivasa Ramanujan, mathématicien indien, né le 22 décembre 1887 à Erode et mort le 26 avril 1920 à Kumbakonam)

Utilisation :

n nombre pour lequel il faut calculer la factorielle

GSB **0** ... lance le calcul de factorielle

Programme :

```

1 Lbl 0
2 Scl0 .0
3 1
4 E
5 rcl .0
6 rcl .0
7 y
8 P,
9 SqrL
10 0
11 8
12 rcl .0
13 0
14 y
15 4
16 rcl .0

17 0
18 1
19 4
20 rcl .0
21 0
22 3
23 0
24 1PI1
25 4
26 5
27 1PI1
28 y
29 0
30 F,11 0
31 rrs

```

10. Trigo

Calcul des sinus, cosinus et tangente d'un angle en degré. N'utilise pas les fonctions SIN, COS, TAN, PI de la calculatrice. Les résultats sont stockés dans les registres .1 .2 et .3

Utilisation :

n angle en degré

GSB **9** ... lance le calcul

Programme :

```

1 Lbl 9
2 3
3 5
4 5
5 Enter 1
6 1
7 1
8 3
9 rcl .0
10 0
11 1
12 8
13 0
14 rcl .0
15 Scl0 .0
16 Enter
17 0
18 2
19 0
20 rcl .0
21 1

22 4
23 1PI1
24 1
25 0
26 0
27 1
28 -
29 rcl .0
30 3
31 rcl .0
32 0
33 Scl0 .1
34 rcl .0
35 Enter
36 0
37 3
38 0
39 rcl .0
40 1
41 4
42 1PI1

43 5
44 0
45 3
46 -
47 rcl .0
48 rcl .0
49 0
50 y
51 ch5
52 rcl .0
53 0
54 1
55 4
56 Scl0 .2
57 rcl .1
58 SqrP
59 rcl .0
60 Scl0 .3
61 clr
62 rrs

```

12. PI Day

Différentes approches de PI...

Utilisation :

ou ou ou ou ... calcul de PI
 ... écart avec la fonction PI

Programme :

1	LbL 1	34	.	67	59-E
2	7	35	9	68	59-E
3	EntEr	36	9	69	59-E
4	5	37	1	70	59-E
5	EntEr	38	EntEr	71	59-E
6	5	39	7	72	1
7	59-E	40	rP	73	4
8	rP	41	rP5	74	59-E
9	59-E	42	LbL 5	75	59-E
10	rP	43	8	76	59-E
11	59-E	44	59-E	77	59-E
12	rP5	45	59-E	78	59-E
13	LbL 2	46	59-E	79	59-E
14	1	47	59-E	80	59-E
15	.	48	59-E	81	59-E
16	8	49	59-E	82	59-E
17	59-E	50	59-E	83	59-E
18	LbL 11	51	59-E	84	59-E
19	rP	52	59-E	85	59-E
20	rP5	53	59-E	86	5
21	LbL 3	54	59-E	87	8
22	3	55	59-E	88	59-E
23	5	56	59-E	89	59-E
24	5	57	59-E	90	59-E
25	EntEr	58	59-E	91	59-E
26	1	59	59-E	92	59-E
27	1	60	59-E	93	4
28	3	61	59-E	94	4
29	rP	62	59-E	95	rP5
30	rP5	63	59-E	96	LbL 9
31	LbL 4	64	59-E	97	P
32	2	65	59-E	98	-
33	1	66	59-E		

11. Gravité

Calcul du temps de chute, en secondes, selon la hauteur

Utilisation :

h hauteur en mètres
 ... lance le calcul

Programme :

1	LbL 3
2	2
3	9
4	9
5	.
6	8
7	rP
8	59-E
9	rLn

13. Premier

Recherche du nombre premier le plus proche du nombre n saisi.

Utilisation :

n
 ... lance la recherche

Programme :

1	LbL	0	24	Sto	.1
2	F,11	0	25	rcl	.1
3	1	1	26	rcl	.3
4	1	1	27	SURP	
5	SqrL		28	IL=Y	
6	Sto	.4	29	Oto	3
7	1	1	30	rcl	.2
8	Sto	.1	31	Sto	.0
9	3	3	32	1	
10	Sto	.2	33	Sto	.1
11	LbL	1	34	Oto	1
12	2	2	35	LbL	3
13	Sto	.2	36	rcl	.2
14	rcl	.2	37	rcl	.1
15	SqrL		38	1	
16	Sto	.3	39	FrAc	
17	rcl	.4	40	IL=0	
18	IL=Y		41	Oto	2
19	Oto	2	42	1	
20	rcl	.0	43	Sto	.1
21	rP5		44	Oto	1
22	LbL	2	45	rP5	
23	2	2			

14. Hilo

Jeu du HILLO : il faut deviner un nombre...

Si le nombre proposé est inférieur au nombre recherché affichage de -1
 Si le nombre proposé est supérieur au nombre recherché affichage de 1
 Si trouvé affichage de 88888 puis affichage du nombre de coups joués.

Utilisation :

xx.xxxx nombre "semence" ...
 ... lance le jeu

n (entre 0 et 1000) ... à répéter jusqu'à fin de jeu

Programme :

1	LbL	8	24	LbL	3
2	SqrL		25	rP5	
3	FrAc		26	Sto	5
4	3	3	27	1	
5	RLoG		28	Sto	5
6	9	9	29	rcl	5
7	1,nt		30	rcl	1
8	Sto	1	31	IL=Y	
9	0	0	32	Oto	1
10	Sto	5	33	IL=Y	
11	F,11	0	34	Oto	2
12	8	8	35	rcl	4
13	8	8	36	Oto	3
14	8	8	37	LbL	1
15	8	8	38	rcl	2
16	8	8	39	PSE	
17	Sto	2	40	PSE	
18	1	1	41	PSE	
19	chs		42	rcl	5
20	Sto	3	43	rP5	
21	1	1	44	LbL	2
22	Sto	4	45	rcl	3
23	clr		46	Oto	3

15. Fraction

Retourne un nombre à 2 décimales sous forme de fraction.
(Exemple : 12.48 ... 312/25)

Utilisation :

nn.nn nombre à transformer en fraction (Exemple : 12.48)

GSB ... lance le calcul

... puis affiche le numérateur (Exemple : 312)

R/S ... affiche le dénominateur (Exemple : 25)

Programme :

```

1 Lbl 0
2 F.11 2
3 1
4 0
5 0
6 StO 1
7 StO .1
8 0
9 ,nt
10 StO .2
11 StO 2
12 Lbl 7
13 rcl 1
14 rcl 2
15 StO 3
16 r
17 ,nt
18 rcl 2
19 0
20 ch5
21 rcl 1
22 1
23 StO 2
24 rcl 3
25 StO 1
26 rcl 2
27 HLT
28 StO 7
29 rcl .2
30 rcl 1
31 r
32 rP5
33 rcl .1
34 rcl 1
35 r
36 rP5

```

16. Convert

Conversion décimal vers binaire ou binaire vers décimal.

Utilisation :

GSB pour décimal vers binaire affiche 10.2

ou

GSB pour binaire vers décimal affiche 2.10

nombre **GSB** ... lance la conversion

Programme :

```

1 Lbl 0
2 F.11 0
3 StO .0
4 StO .4
5 rcl .1
6 rcl .2
7 1
8 StO .3
9 Lbl 1
10 rcl .4
11 rcl .2
12 r
13 ,nt
14 StO .4
15 HLT
16 StO 2
17 rcl .3
18 0
19 StO .0
20 rcl .1
21 StO .3
22 StO 1
23 Lbl 2
24 rcl .0
25 rP5
26 Lbl 8
27 1
28 0
29 StO .1
30 2
31 StO .2
32 1
33 0
34 r
35 1
36 F.11 1
37 rP5
38 Lbl 8
39 2
40 StO .1
41 1
42 0
43 StO .2
44 2
45 .
46 1
47 F.11 2
48 rP5

```

WELCOME TO THE RPN-1250 CALCULATOR



Benoit Maag © 2015

KEY FEATURES

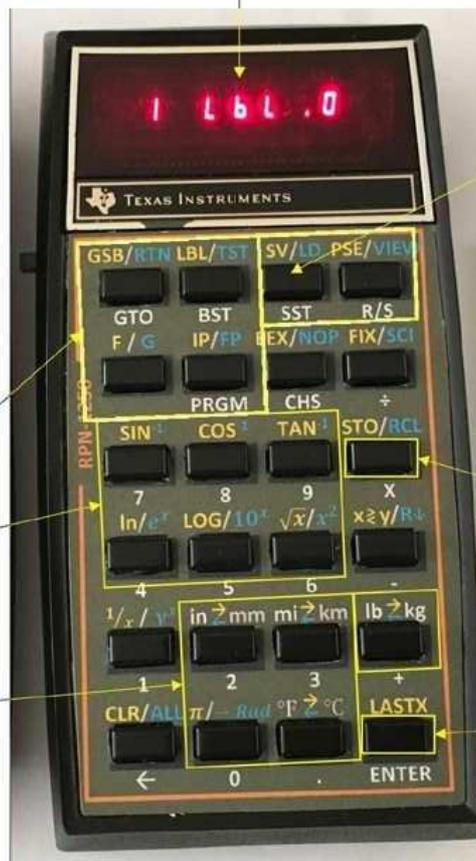
Based on Microchip PIC 18F2680 (64k FLASH) at 8 MHz or PIC 18F26K80 (64k FLASH) at 16 MHz
 MAX 7219 Display driver
 8-digit 7-segment LED display
 RTC (RPN-1250+)
 Original TI-1250 enclosure
 9V standard battery

Programmable – 98 steps
 Save 3 programs to continuous memory
 6 levels of subroutine
 All 12 tests
 Indirect addressing

Scientific Functions

Basic Conversions

Alphanumeric Display (brightness adjustable)



In RUN mode:
 SV saves all memories and stack into continuous memory
 LD restores all memories and stack from continuous memory

20 memories: 0~9 and .0 ~ .9 + 10 additional through indirect addressing
 Store and Recall Arithmetic

RPN

RUN MODE

GTO 0~9 or .0~.9 goes to corresponding label
 GSB 0~9 or .0~.9 execute program from corresponding label
 RTN in run mode goes to program step 0

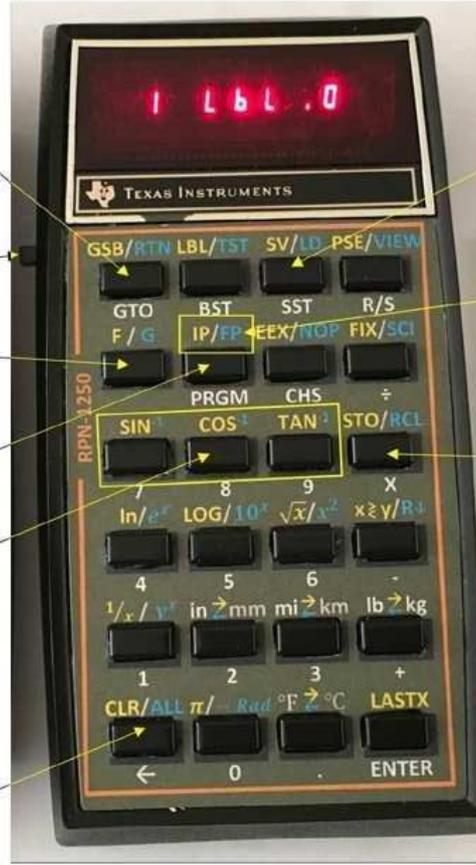
OFF/ON Switch

SHIFT KEY:
 Press once for F
 Press twice for G
 Press a third time to go back to unshifted

Press once for program entry mode
 Press again to go back to Run mode

TRIG functions : angle in RAD only

CLR: clears X Register
 ALL: clears stack and all memories



In RUN mode:
 SV saves all memories and stack into continuous memory
 LD restores all memories and stack from continuous memory

IP: Integral Part
 FP: Fractional Part

20 memories: 0~9 and .0 ~ .9
 Store and Recall Arithmetic

PROGRAM MODE

20 labels 0~9 and .0~.9

OFF/ON Switch

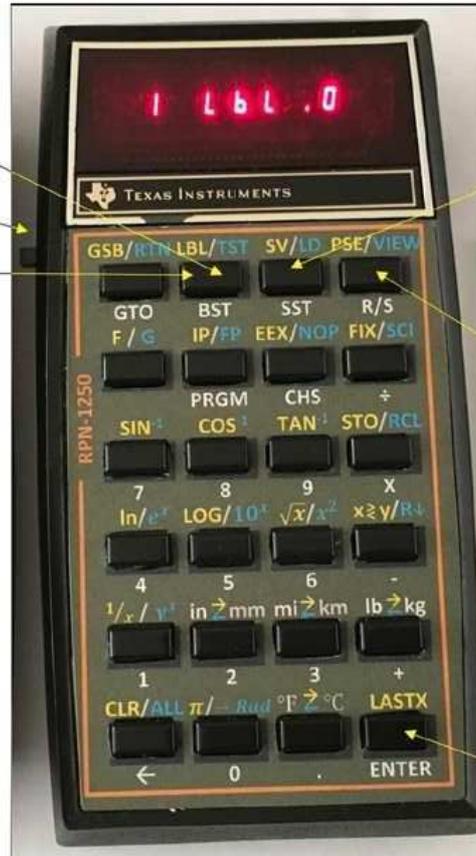
TST: all 12 tests

After TST, press below key for test vs 0:

SIN ⁻¹	COS ⁻¹
X =	X <>
07	08
ln/e ^x	LOG/10 ^x
X >	X >=
04	05
1/x / y ^x	in \rightarrow mm
X <	X <=
01	02

After TST, press . (dot) and then below key for test vs Y:

SIN ⁻¹	COS ⁻¹
X = Y	X <>
7	Y 8
ln/e ^x	LOG/10 ^x
X > Y	X >=
4	Y 5
1/x / y ^x	in \rightarrow mm
X < Y	X <=
1	Y 2



In Program mode:
 SV followed by 1, 2 or 3 saves program into continuous memory (you can save up to 3 programs)
 LD followed by 1, 2 or 3 restores program 1, 2 or 3 from continuous memory

Run Program
 VIEW 0~9 or .0~.9 to view a register without stopping program

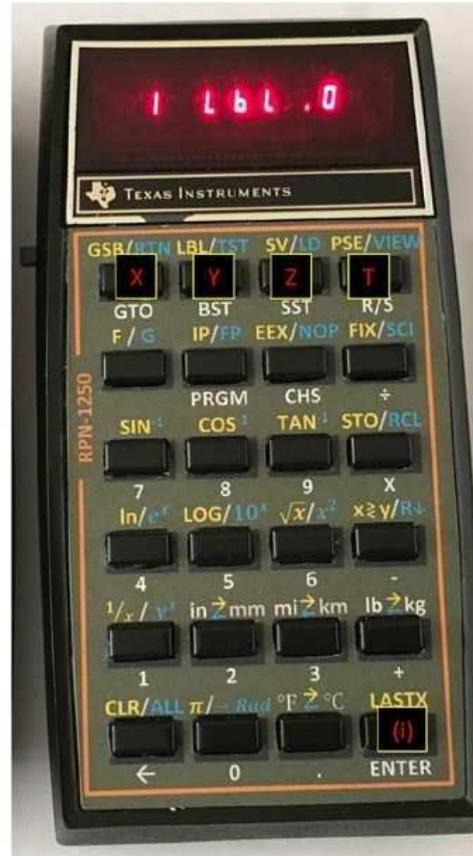
Stops a running Program

STACK REGISTERS INDIRECT ADDRESSING

Store and Recall functions (including store and recall arithmetic) and VIEW can operate on stack registers, which are linked to the top row keys

As in the HP-29C the indirect addressing register is R0

Indirect addressing (i) works with Store, Recall (including store and recall arithmetic) and VIEW functions only (No fix (i), or GSB (i)) and is called with the [ENTER] key



REAL TIME CLOCK (RTC) RPN-1250+ Only

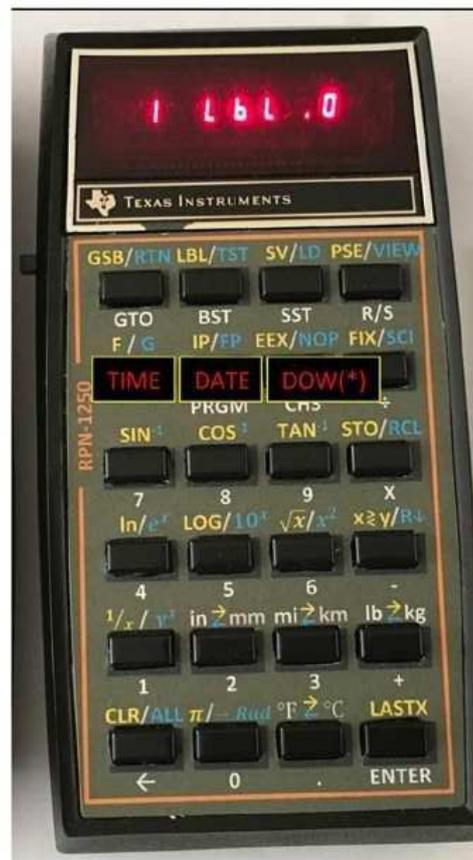
Time, Date and Day Of Week are addressed as memory registers using the [PREFIX] [PRGM] and [CHS] Keys

Time format is hh.mmss
(24 hour)

Date format is mm.ddyy

Day of Week is 1~7 for
Mon-Sun

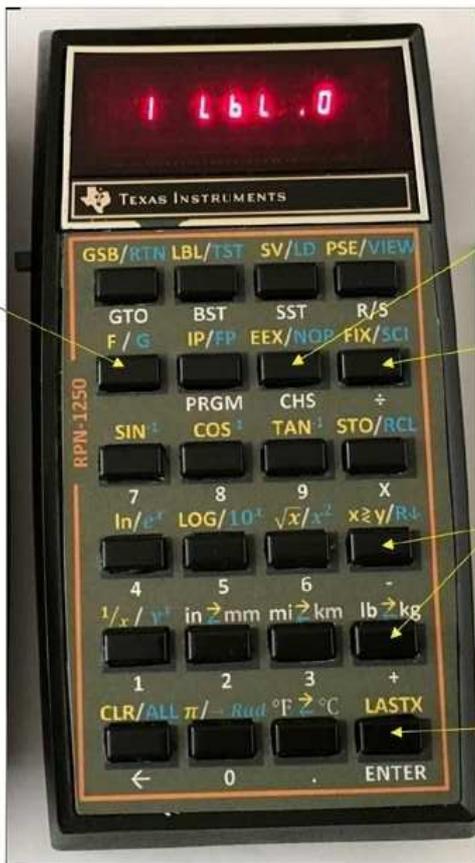
RTC functions are
programmable



(*) DAY OF WEEK

ESC MODE

Displays Firmware Version



Shows all alphanumeric characters

Display test (lits up all display segments)

- to lower display brightness
+ to increase display brightness

G prefix + ENTER: to enter 'Esc mode'
Press ENTER to leave 'Esc Mode'