Quick Reference Guide

SwissMicros GmbH

Table of Contents

DM42 Quick Reference Guide

- Inspired by the HP-42S Quick Reference Guide (c) Copyright 1988 Dex Smith

Navigation

- Press ÷ to move one page backward
- Press × to move one page forward
- Press - to move one line backward
- Press + to move one line forward
- Press ▲ to select previous link
- Press ▼ to select next link
- Press ENTER to jump to selected link
- Press EXIT or F1 to exit
- Press F2 to jump to Contents
- Press F3 to open help file selection dialog

Numpad-like navigation keys

- 2/8 - single line down/up
- 3/9 - page down/up
- 4/6 - prev/next link (same as arrows)
- 5 - follow the link (same as ENTER)
- 7 - go to the Contents (same as F2)

Symbols used in this guide

- A 123 represents a soft button or anything on the LCD
- A STO represents a physical button

  ```
  +
  ```

Contents

+

Menus

DM42 calculator has far too many functions to assign a key to each one, so most of them are available in menus rather than directly on keys.
Each menu displays a row of labels at the bottom of the screen, one above each key in the top row of the keyboard.
A function within the menu is activated by pressing the key directly below it. If there are more than six functions in the menu then the ▼▲ annunciator is displayed and the ▼ and ▲ keys allow you to scroll through the menu's pages. There are two kinds of menu:

- Application Menu
- Function Menu

+

Application Menu

There are five applications built into the DM42:

- BASE Base Conversions
- MATRIX Matrix Operations
- SOLVER The Solver
- STAT Statistics
- ∫f(x) Numerical Integration

```
 +
When you enter an application and its menu opens, any other menus are
closed automatically. However, you can still open a function menu from
within the application and when you've finished with it, it will close
and reveal the application menu again. +
 +
```

Function Menu

- CATALOG All DM42 Functions
- CLEAR Functions for clearing programs and variables
- CONVERT Angle, time, and coordinate conversion functions
- CUSTOM Displays custom menu
- DISP Display Formats
- FLAGS Flags
- MODES Angles and Coordinates
- PGM.FCN Functions useful in programs
- PRINT Enable IR Printing
- PROB Combinatorics and probability functions

```
 +
Unlike application menus, a Function menu closes as soon as you have
used one of the functions that it contains, with the exception of the
[.button]#CUSTOM# menu, which remains open. Should you need to use more
than one function from a Function menu, you can instruct the DM42 to
keep the menu open by opening it twice in succession. +
 +
```

Storage

- The Stack
- Storage Registers
- Variables
- The Alpha Register

- Flags (00-99)
- Available Memory

  ```
  +
  ```

The Stack

The stack is the scratchpad that you use for performing calculations. It stores and shows you intermediate results as you work through a problem.
The four registers that make up the stack are known as X,Y,Z,T. Auxiliary register LASTx holds contents of X register used in most recent numeric function. Each of these registers can contain data of any type.

Storage Registers

Because of the ephemeral nature of the stack, data in its registers is likely to be lost when a new calculation is started.
More permanent storage can be achieved in the storage registers, of which there are 25 initially, numbered 00 to 24. This number can be altered with the SIZE function in the MODES menu.
STO *nn* STOres the contents of the X register into the storage register with the given number.
RCL *nn* ReCaLls the data from storage register *nn* into the stack.
If SIZE has been used to define more than 100 storage registers then registers numbered 100 and onwards can only be accessed using indirect addressing.
The STATistics application uses some of the storage registers for its internal housekeeping:

- R11 Σ x
- R12 Σ x^2
- R13 Σ y
- R14 Σ y^2
- R15 Σ xy
- R16 n

If ALLΣ mode is being used (as opposed to LINΣ mode) then the following additional storage registers are used:

- R17 Σ ln x
- R18 Σ $(\ln x)^2$
- R19 Σ ln y
- R20 Σ $(\ln y)^2$
- R21 Σ ln x ln y
- R22 Σ x ln y
- R23 Σ y ln x

The storage registers can be all real numbers or all complex numbers. See Variables for how to switch between the two.

Variables

Rather than referring to stored data by a simple number (the number of the storage register containing it), it is possible to create named variables. Variables allow you to associate a meaningful name with the data. Also, named variables can contain any type of data, not just real or complex numbers.
E.g. suppose you have just calculated the volume of a shape. Store the value currently in the X register representing this volume in a variable named *VOL*:
STO ENTER VOL ENTER
Variables can have names up to seven characters long.
There is one system-reserved variable, REGS. REGS is a matrix containing the storage registers. REGS can be a real matrix, in which case all of the storage registers, i.e. the elements of REGS, are real numbers, or it can be a complex matrix, in which case all of the storage registers are complex numbers.
To convert REGS (or any other real matrix) to a complex matrix:
0 ENTER COMPLEX STO + REGS
To convert REGS from a complex matrix back into a real matrix:
RCL REGS COMPLEX x<>y STO REGS

The Alpha Register

The alpha Register can store up to 44 characters of text, which equate to two lines of text on the original HP-42S.

Available Memory

The DM42 has about 70KB of system memory that your programs and variables can use. The storage register matrix REGS uses some of this space.
To find out how much memory your DM42 has available, open the CATALOG menu and then press and hold MEM.
If you need more memory then use the CLP (CLear Program) and CLV (CLear Variable) functions from the CLEAR menu to remove unneccesary programs and variables and thus free up additional memory.
DO NOT DELETE THE REGS VARIABLE! Remember, it holds your storage registers! If necessary you can reduce its size with the SIZE command from the MODES menu.

Data Types

- Real Numbers
- Complex Numbers

- Alpha Strings
- Matrices

```
+
```

Real Numbers

Real numbers cover integer and decimal numbers, positive or negative. The range of numbers that the DM42 can handle is approximately 10^{-6143} to 10^{6144} with a precision of 34 digits.

Complex Numbers

A complex number consists of two real numbers associated to form its real part and its imaginary part (*rectangular* coordinate mode).
Complex numbers are always stored internally in rectangular form but can be displayed directly:
0.2343 i1.4456
or in *polar* coordinate mode with a magnitude and an angle:
1.4645 □80.7937
Both of the examples here represent the same complex number.
To enter a complex number in rectangular mode:
real part ENTER *imaginary part* COMPLEX
To enter a complex number in polar mode:
Magnitude ENTER *angle* COMPLEX
The COMPLEX function either combines real numbers (or matrices) in the X and Y stack registers to form a complex number (or matrix) in X, or it performs the opposite operation by splitting the complex number (or matrix) in X into two real numbers (or matrices) in X and Y.
The role of the real number (or matrix) taken from X when combining X and Y or left in X when splitting the complex number (or matrix) in X depends on the display mode, rectangular or polar, in force when COMPLEX is executed. If *rectangular* then X is/was the imaginary part. If *polar* then X is/was the angle.
Complex numbers are displayed in polar mode normalised such that the magnitude is always positive and the absolute value of the angle does not exceed 180 degrees.

Alpha Strings

The alpha register contains strings of text up to 44 characters (2 lines on the HP-42S) in length.
Real variables and elements of real matrices (including REGS) can also contain strings of up to 6 characters in length. Basically, you can store snippets of text anywhere you can store real numbers. However, you cannot store two such snippets in a complex variable.

Matrices

Matrices can be one-dimensional or two-dimensional. A matrix can have any number of rows and columns as long as there is enough memory in the DM42 to store all of its individual elements.

See Matrix Operations for more details.

Modes

- Angles and Coordinates
- Other Modes
- Display Formats
- Printing

```
+
```

Angles and Coordinates

Press MODES
The angle specifiers DEG, RAD and GRAD tell the DM42 to use degrees, radians, and gradians respectively when displaying and interpreting angles.
RECT and POLAR tell the DM42 to display and interpret complex numbers in rectangular and polar mode, respectively.

Other Modes

Press MODES ▼
SIZE resizes the REGS matrix variable, thus defining the number of storage registers you can use.
QUIET switches off the internal buzzer and keeps the DM42 nice and.... quiet!
CPXRES allows the DM42 to return complex results when working on real numbers (e.g. square root of a negative number) while REALRES ensures that only real numbers are returned. In this case, operations that would otherwise return a complex result trigger an Invalid Data error. Operations given a complex number as an argument will return a complex number regardless of the CPXRES /REALRES setting.
KEYASN means that variables and programs assigned to the CUSTOM menu with the ASSIGN command will be shown in the CUSTOM menu. If LCLBL is selected then local alpha labels populate the CUSTOM menu.

Display Formats

Press DISP
FIX, SCI and ENG instruct the DM42 to display real numbers in fixed decimal, scientific and engineering notation, respectively, with a set number of decimals.
When ALL is selected, numbers are displayed with up to 12 digits precision, which was the native precision of the HP-42S.
Depending on where in the world you are, you will be used to using a dot as the decimal separator and a comma for grouping digits three by three before the decimal (e.g. 1,234.56) or, as in most of

Europe, a comma as the decimal separator and a dot for grouping digits before the decimal (e.g. 1.234,56). RDX. selects the dot as the decimal separator and RDX, selects the comma.

Printing

Press PRINT ▲
PRON and PROFF enable and disable printing by setting or clearing respectively flags 21 and 55.
MAN leads to items being printed manually. NORM leads to input and calculation results alone being printed automatically while TRACE sends a trace of all operations to the printer.
Additional modes are described under "Matrix Operations" and "Statistics."

Executing Functions & Programs

- Quick access to programs and functions
- Addressing modes

```
 +
Any function or program can be executed by pressing [.button]#XEQ# and
entering its name. If there is more than one program with the same name
then when you attempt to [.button]#XEQ# that program, the one closest to
the permanent [.lcd]#.END.# marker will be executed. +
Only the current program will be searched if the name that you give is
that of a local label. +
 +
```

Quick access to programs and functions

There are four ways to get at your programs and at the DM42’s functions quickly:

- You can ASSIGN up to 18 names of programs or variables to the CUSTOM menu. Pressing an item in the CUSTOM menu is the same as executing it (XEQ) if it’s a program or built in function or, if it is a variable, recalling it (RCL).
- When you press the XEQ button, the program catalog is automatially presented to you. Simply scroll through the catalog with the ▲ and ▼ keys if needed and select the program to run.
- The Run/Stop Key R/S starts running the current program at the current line or stops a running program once the instruction that it is working on has completed.
- All of the built in functions of the DM42 are available in the Function Catalog that can be opened by pressing CATALOG FCN

```
+
```

Addressing modes

Some functions require a parameter that tells them on what specific item to operate. These functions accept one or more of the following addressing modes:

- **Numeric**: these functions include RCL, SF, GTO, and others. They prompt you to enter a numerical value by displaying an underscore for each digit that they are expecting. For example, if you press SIZE, the DM42 displays
  SIZE __
  prompting you to input four digits. You can always input fewer digits and terminate the input with ENTER and the DM42 will assume that the missing digits are zeroes on the left.
- **Alpha**: some functions that accept numeric addressing also accept alpha addressing. An example of one such function is RCL, which allows you to specify the number of a storage register to recall or the (alpha) name of a variable. In many cases, the DM42 will show you a menu of alpha names that are valid for the instruction that you are typing but if not, you can always use the ALPHA menu to type it in yourself.
- **Stack**: because the stack registers can hold anything that a storage register can, they can also be used in place of a storage register. For example, to recall the contents of the T register into X (trick to roll the stack upwards instead of downwards), do the following:
  RCL . ST T
- **Indirect Addressing**: this is a technique that does not give the DM42 the parameter that it needs. Instead, it tells the DM42 *where* to find that parameter. For example, to execute the program whose name is stored in storage register 24, you would: XEQ . 24
  Stack registers can also be used with indirect addressing. For example, suppose that the name of the program that you want to execute is not in R24 but in the X register of the stack. In this case you would:
  XEQ . . ST X

  ```
  +
  ```

Programming

- Program-Entry
- Labels
- The Do-If-True Rule
- Looping

  ```
  +
  ```

Program-Entry

- PRGM enters or exits Program-entry mode
- GTO . . moves to a new program space

- GTO . nnnn moves to line number nnnn
- ← deletes the current program line
- SST moves to the next program line*
- BST moves to the previous pgm line*

```
 +
* Use [.button]#□# or [.button]#□# if no menu is displayed. +
 +
```

Labels

A program label is simply a marker used to identify a program or a routine within a program.
Global labels can be accessed from anywhere in memory (and therefore should be unique). Global labels are distinguished from local labels by quotation marks (such as LBL "SAMPLE").
Local labels can be accessed only within the current program (and should be unique within the current program). There are two types of local labels:

- Numeric (LBL 00 - LBL 99)
- Alpha (LBL A - LBL J and LBL a - LBL e)

```
+
```

The Do-If-True Rule

The do-if-true rule determines how program lines are executed when a conditional function is encountered. If the condition is "true," the line immediately following the conditional is *executed*. If the condition is "false," the line following the conditional is *skipped*.

Looping

The ISG and DSE functions control looping. Each accesses a variable or register containing a control number in the form *ccccccc.fffii*; where *ccccccc* is the current counter value, and *fff* is the final counter value, and *ii* is the increment size (default is 1). Both ISG and DSE follow a variation of the do-if-true rule: if the count is not complete, the line following the instruction is executed (usually a branch to the top of the loop).
For example, this program segment counts from 1 to 52 by threes (executing the loop 18 times) and then beeps.

```
17 1.05203
18 STO "COUNT"
19 LBL 01
…
23 ISG "COUNT"
24 GTO 01
25 BEEP
```

Using a Variable Menu

A variable menu may be displayed by the Solver or Integration applications, or by the VARMENU function within a program. Each label in the menu represents a variable. While the menu is displayed, you can:

- **Store a value into a variable:**
  Key in the value and then press the menu key.
- **Recall the contents of a variable:**
  Press RCL and then the menu key.
- **View the contents of a variable without recalling it:**
  Press (shift) and then hold the menu key down.
- **Select a variable:**
  Press the menu key without keying in a number first. This action places the variable name in the alpha register and continues execution.
  (For the Solver, this is how you select the unknown variable. For Integration, this is how you select the variable of integration.)

You can select and use any function menu without exiting from the variable menu.

The Solver

The Solver is a root finder that allows you to solve for an unknown variable in an expression, given values for all the other variables. Expressions are written as programs. There are three parts to a Solver program:

- The program must begin with **a global label.**
- Immediately following the global label, **menu variables** are declared with MVAR instructions.
- Finally, the body of the program should evaluate the expression. Recall the variables as they are needed and calculate *f(x)* (where *f(x)* = 0 for your expression of one or many variables).

```
 +
After entering the program, these are the steps for using the Solver: +
```

- Press SOLVER
- Select a Solver pgm from the menu.
- Use the variable menu to store a value into each of the known variables. Optional: store one or two guesses into the unknown variable to direct the Solver to a solution.
- Solve for the unknown variable by pressing the corresponding menu key.

```
 +
*A Simple Example:* For the expression _A + B = C_, rewrite the
expression as _A + B - C_ = 0. The Solver program looks like this: +
 +
[.lcd]#01 LBL "SIMPLE"# +
[.lcd]#02 MVAR "A"# +
[.lcd]#03 MVAR "B"# +
[.lcd]#04 MVAR "C"# +
[.lcd]#05 RCL "A"# +
[.lcd]#06 RCL+ "B"# +
[.lcd]#07 RCL- "C"# +
[.lcd]#08 END# +
 +
Hint: create the variables before entering the program. After entering
the program, you can use it to solve for any variable, given a value for
each of the others. For example, find _A_ when _B_ = 12 and _C_ =
log(_B_). +
 +
Select the program: [.button]#SOLVER# [.lcd]#SIMPL# +
Store _B_: 12 [.lcd]#B# +
Store _C_: [.button]#LOG# [.lcd]#C# +
Solve for _A_: [.lcd]#A# +
 +
```

Numerical Integration

The Numeric Integration application allows you to calculate an approximation of a definite integral. The integrand, *f(x)*, is written as a program similar to a Solver program (see The Solver). That is, the program must use a global label, declare the menu variables, and evaluate *f(x)*.

After entering the integrand program, here are the steps for using the Integration application:

1. Press ∫f(x)
2. Select an integrand program from the menu.
3. Use the variable menu to store a value into each of the variables that should remain constant.
4. Select the variable of integration by pressing the corresponding menu key.
5. Store the lower limit (*LLIM*), the upper limit (*ULIM*), and the accuracy factor (*ACC*).
6. Press ∫ to calculate the integral. The approximation for the integral is returned to the X-register and the uncertainty of computation is returned to the Y-register.

```
+
```

Matrix Operations

To create a new $m \times n$ matrix, enter the dimensions:
m ENTER *n* (for m rows and n columns) and then press:

MATRIX NEW for a matrix in the X-register.
Or MATRIX ▼ DIM ENTER *name* ENTER for a matrix in a variable. If the matrix already exists, the DIM function redimensions it.

Edit the matrix in the X-register:
MATRIX EDIT

Edit a named matrix:
MATRIX ▼ EDITN name

When a matrix is being edited it is said to be indexed. (To index a named matrix without editing it, use the INDEX function.) Whenever there's an indexed matrix, two pointers are used to indicate the row and column of the current element: *I* and *J*, respectively.

Wrap and Grow Modes. If the index pointers are positioned to the last (lower-right) element in a matrix and you move to the right one position:

- The pointers wrap around to the first element of the matrix (Wrap mode).
- Or, the matrix grows by one complete row and the pointers move to the new row (Grow mode).

Wrap mode is automatically selected whenever you enter or exit the matrix editor. (The WRAP and GROW functions are in the second row of the editor menu.)

Matrix Arithmetic. Most arithmetic and other operations work for matrices just as for individual numbers. Anytime a matrix is used in a mathematical operation with a complex number, the result is a complex matrix. Therefore, you can make any matrix complex by adding 0 + *i*0 to it:
0 ENTER COMPLEX # + or 0 [.button]#ENTER# [.button]#COMPLEX# [.button]#STOM# [.button]# name

To solve a system of simultaneous linear equations represented by the matrix equation AX = B:

1. Press MATRIX SIMQ.
2. Key in the number of unknowns. The calculator automatically creates or redimensions the matrix variables *MATA, MATB,* and *MATX.*
3. Optional: If your equations involve complex numbers, make *MATA* and/or *MATB* complex (as shown at the top of this page).
4. Press MATA , fill the matrix and press EXIT
5. Press MATB , fill the matrix and press EXIT
6. Press MATX to calculate the solution matrix. Use the matrix editor keys to view the results.

   ```
   +
   ```

Statistics

Statistical data is accumulated into 6 or 13 sequential storage registers. Initially, the first summation register is R11. Use the ΣREG function to change the location of the first summation register. ΣREG does not move the data in the registers.

Options to set the summation mode:

- STAT ▼ ALLΣ to use all 13 coefficients
- STAT ▼ LINΣ to use only the first six coefficients (which allows only linear curve fitting)

Clear the summation registers:
CLEAR CLΣ

Options to accumulate data:

- For each *x-y* data pair: *y-value* ENTER *x-value* Σ+
- For each single-point data value: *x-value* Σ+
- For *x-y* data pairs stored in a two-column matrix (*x-values* in column 1; *y-values* in column 2): Place the matrix in the X-register and then press Σ+

```
 +
*To undo mistakes:* +
Put the incorrect data in the stack (try [.button]#LASTx# ). +
Press [.button]#Σ-# and continue accumulating data. +
 +
*To select a curve model for forecasting:* +
Press [.button]#STAT# [.lcd]#CFIT# [.lcd]#MODL# +
and then one of the following: +
```

- LINF *linear model: y = mx + b*
- LOGF *logarithmic model: y = m ln(x) + b*
- EXPF *exponential model: ln(y) = mx + ln(b)*
- PWRF *power model: ln(y) = m ln(x) + ln(b)*
- BEST selects the model that returns the best correlation coefficient

```
+
```

Base Conversions

Real numbers are displayed according to the current base mode (Hexadecimal, Decimal, Octal, or Binary). You can change the base mode using the BASE menu or by manually executing HEXM, DECM, OCTM, or BINM. Decimal mode is automatically selected when you exit from the BASE menu.

Press and hold SHOW to display:

- A hexadecimal, decimal, or octal number in full-precision decimal form.
- Or, all 36 bits of a binary number.

```
 +
When the BASE menu is displayed, the following keys are temporarily
redefined with these integer functions: +
```

± BASE±	36-bit 2’s complement.
÷ BASE÷	36-bit integer divide.
× BASE×	36-bit integer multiply.
- BASE-	36-bit integer subtract.
[.button]## BASE	36-bit integer add.

```
 +
Bits are numbered from right to left beginning with 0. Bit 35 (the most
significant bit) is the sign bit. Negative numbers are represented in
2□s complement form. Nondecimal numbers longer than 36 bits are
displayed as [.lcd]#<Too Big># +
 +
```

DM42 Functions

ABS		Absolute value	
ACOS		Arc cosine	
ACOSH		Arc hyperbolic cosine	
ADV		Advance paper	
AGRAPH	Alpha graphics		
AIP		Alpha integer part	
ALENG		Alpha length	
ALL		All display format	
ALLΣ		ALLΣ mode (13 sum regs)	
AND		Logical AND	
AOFF		Alpha off	
AON		Alpha on	
ARCL		Alpha recall	

AROT		Alpha rotate	
ASHF		Alpha shift	
ASIN		Arc sine	
ASINH		Arc hyperbolic sine	
ASSIGN	Assign CUSTOM menu key		
ASTO		Alpha store	
ATAN		Arc tangent	
ATANH	Arc hyperbolic tangent		
ATOX		Alpha to X	
AVIEW		Alpha view	
BASE+		Base add	
BASE-		Base subtract	
BASE×		Base multiply	
BASE÷		Base divide	
BASE+/-	Base change sign (2’s compl.)		
BEEP		Beep	
BEST		Best fit model	
BINM		Binary mode	
BIT?		Bit test (x-th bit of y)	
BST		Back step	
CF			Clear flag
CLA		Clear alpha register	
CLALL		Clear all memory	
CLD		Clear display	
CLKEYS	Clear CUSTOM menu keys		
CLLCD		Clear LCD	
CLMENU	Clear programmable menu		
CLP		Clear program	
CLRG		Clear registers	
CLST		Clear stack	
CLV		Clear variable	

CLX		Clear X-register	
CLΣ		Clear summation registers	
COMB		Combinations	
COMPLEX	Complex		
CORR		Correlation	
COS		Cosine	
COSH		Hyperbolic cosine	
CPXRES	Complex-result enable		
CPX?		Complex test	
CROSS	Cross product		
CUSTOM	CUSTOM menu		
DECM		Decimal mode	
DEG		Degrees mode	
DEL		Delete program lines	
DELAY	Printer delay time		
DELR		Delete matrix row	
DET		Determinant	
DIM		Dimension matrix	
DIM?		Dimensions of matrix in X	
DOT		Dot product	
DSE		Decrement, skip if ≤ zero	
EDIT		Edit matrix in X-register	
EDITN		Edit named matrix	
END		End of a program	
ENG		Engineering display format	
ENTER	Enter		
EXITALL	Exit all menus		
EXPF		Curve-fitting model: exp.	
E□X		Natural exponential	

E□X-1		Nat. exp values near zero	
FCSTX	Forecast x given y (stat)		
FCSTY	Forecast y given x (stat)		
FIX		Fixed dec. display format	
FRNM		Frobenius norm of X matrix	
FP			Return fractional part
GAMMA		Gamma function	
GETKEY	Put next key number to X-reg		
GETM		Get matrix	
GRAD		Grads angular mode	
GROW		Grow mode (matrix edition)	
GOTO		Go to label	
HEXM		Hexadecimal mode	
HMS+		Add x and y using H.MMSSss	
HMS-		Subtract x from y H.MMSSss	
I+			Inc. row pointer (matrix)
I-			Dec. row pointer (matrix)
INDEX		Index matrix	
INPUT		User input (prog. only)	
INSR		Insert row (matrix edition)	
INTEG		Integrate	
INVRT	Invert matrix in X-register		
IP			Return integer part
ISG		Increment, skip if greater	
J+			Inc. column pointer (matrix)

J-			Dec. column pointer (matrix)
KEYASN	Key-assignments mode		
KEYG		On key, go to	
KEYX		On key, execute	
LASTX	Last x		
LBL		Label	
LCLBL		Local label mode	
LINF		Linear fit model	
LINE		Linear mode (six sum regs)	
LIST		List program lines	
LN			Natural logarithm
LN1+X		Nat log for values near zero	
LOG		Common logarithm	
LOGF		Logarithmic fit	
MAN		Manual printing	
MAT?		Matrix test	
MEAN		Arithmetic mean (average)	
MENU		Programmable MENU	
MOD		Modulo	
MVAR		Menu variable	
N!			Factorial
NEWMAT	New matrix		
NORM		Normal printing	
NOT		Logical NOT	
OCTM		Octal mode	
OFF		Off	
OLD		Old element value	
ON			Continuous on
OR			Logical OR
PERM		Permutations	
PGMINT	Program to integrate		

PGMSLV	Program to solve		
PI			pi
PIXEL		Pixel on	
POLAR	Polar mode		
POSA		Position in alpha	
PRA		Print alpha	
PRLCD	Print LCD		
PROFF	Printing off		
PROMPT	Prompt		
PRON		Printing on	
PR?		Print program	
PRSTK	Print stack		
PRUSR	Print user vars and labels		
PRV		Print variable	
PRX		Print X-register	
PR2		Print summation registers	
PSE		Pause	
PUTM		Put matrix	
PWRF		Power fit	
QUIET		Quiet mode	
RAD		Radians mode	
RAN		Random number	
RCL		Recall	
RCL+		Recall add	
RCL-		Recall subtract	
RCL×		Recall multiply	
RCL÷		Recall divide	
RCLEL	Recall element		
RCLIJ		Recall IJ pointers	
RDX,		Radix comma	
RDX.		Radix period	
REALRES	Real-results only		
REAL?		Real test	

RECT		Rectangular mode	
RND		Round	
RNRM		Row norm	
ROTXY	Rotate y by x bits		
RSUM		Row sum	
RTN		Return	
R<>R		Row swap row	
R↑			Roll up
R↓			Roll down
SCI		Scientific notation	
SDEV		Standard deviation	
SEED		Seed (for RAN)	
SF			Set flag
SIGN		Sign	
SIN		Sine	
SINH		Hyperbolic sine	
SIZE		Size of REGS	
SLOPE		Slope	
SOLVE		Solve for variable	
SQRT		Square root	
SST		Single step	
STO		Store	
STO+		Store add	
STO-		Store subtract	
STO×		Store multiply	
STO÷		Store divide	
STOEL	Store element		
STOIJ		Store IJ pointers	
STOP		Stop program	
STR?		String test	
SUM		Recall Σx and Σy	
TAN		Tangent	
TANH		Hyperbolic tangent	
TONE		Tone (0-9)	

TRACE	Trace printing		
TRANS	Transpose matrix		
UVEC		Unit vector	
VARMENU	Variable menu		
VIEW		View	
WMEAN	Weighted mean		
WRAP		Wrap mode	
X<>		x exchange	
X<>Y		x exchange y	
XEQ		Execute	
XOR		Exclusive OR	
XTOA		X to alpha	
X^2			Square, x^2
YINT		Y-intercept	
Y^X			Power, y^x
1/X		Reciprocal	
10^X		Common exponential, 10x	
+			Add
-			Subtract
×			Multiply
÷			Divide
+/-		Change sign	
Σ+			Summation plus
Σ-			Summation minus
ΣREG		Set loc. of 1. sum reg	
ΣREG?		Recall loc. of 1. sum reg	
→DEC		To decimal	
→DEG		To degrees	
→HMS		To hours-minutes-seconds	
→HR		To decimal hours	
→OCT		To octal	
→POL		To polar	
→RAD		To radians	

→REC		To rectangular	
←			Index pointers left
□			Index pointers up
□			Index pointers down
→			Index pointers right
%			Percent
%CH		Percent change	
ADATE	Append date from X to alpha		
ATIME		Append time from X to alpha	
AT24		Append 24h time from X to a.	
CL12		Set 12h time format AM/PM	
CL24		Set 24h time format	
DATE		Put date to X register	
DATE+	Add dates in X and Y regs		
DDAYS	Delta days of X and Y regs		
DMY		Set date format to DMY	
DOW		Day of Week of date in X reg	
MDY		Set date format to MDY	
TIME		Put time to X register	

```
 +
 +
Note: If you execute an link:#HP-41_Functions[HP-41 Function], it is
automatically converted into the corresponding DM42 function. +
 +
```

Test Functions

If true, execute the next program line. If false, skip the next program line.

X<0?	less than zero
X<Y?	less than y

X≤0?	less than or equal to zero
X≤Y?	less than or equal to y
X=0?	equal to zero
X=Y?	equal to y
X≠0?	not equal to zero
X≠Y?	not equal to y
X>0?	greater than zero
X>Y?	greater than y
X≥0?	greater than or equal to zero
X≥Y?	greater than or equal to y

```
+
+
```

HP41 Functions

CHS		+/-	
DEG		→DEG	
D-R		→RAD	
ENTER□	ENTER		
FACT		N!	
FRC		FP	
HMS		→HMS	
HR			→HR
INT		IP	
OCT		→OCT	
P-R		→REC	
RDN		R□	
R-D		→DEG	
R-P		→POL	
ST+		STO+	
ST-		STO-	
ST*		STO×	
ST/		STO÷	
X⇐0?		X≤0?	

X⇐Y?		X≤Y?	
*			×
/			÷

```
+
```

Using the ALPHA Menu

To type an alpha string into the alpha register:

1. Press ALPHA to select the ALPHA menu.
2. Optional: press ENTER to turn on the cursor (in Program-entry mode, inserts the I- symbol).
3. Type the string using the characters shown below. Use shift to type lowercase letters.
4. Press EXIT or ENTER

    ```
    +
    +
    ```

Flags

- 00-10 User Flags
- 11 Auto Execute
- 12 Print Double-wide
- 13 Print Lowercase
- 15-16 Print Mode
- 19-20 General Use
- 21 Printer Enable
- 22 Numeric Input
- 23 Alpha Input
- 24 Ignore Range Errors
- 25 Ignore Next Error
- 26 Beeper Enable
- 27 CUSTOM Menu
- 28 Radix Mark Period
- 29 Digit Separators
- 30 Stack Lift Disable
- 31 DMY Date Format

- 34-35 AGRAPH Control
- 36-39 Number of Digits
- 40-41 Display Format
- 42 Grads Mode
- 43 Radians Mode
- 44 Continuous On
- 45 Solving
- 46 Integrating
- 47 Variable Menu
- 48 Alpha Mode
- 49 Low Battery Power
- 50 Message
- 51 Two-Line Message
- 52 Program-Entry Mode
- 53 INPUT
- 55 Printer Existence
- 56 Linear Model
- 57 Logarithmic Model
- 58 Exponential Model
- 59 Power Model
- 60 AllΣ Mode
- 61 Log Model Invalid
- 62 Exp Model Invalid
- 63 Pwr Model Invalid
- 64 Shift State
- 65 Matrix Editor In Use
- 66 Grow Mode
- 67 YMD Date Format
- 68-71 Base Mode
- 72 Local-Label Mode
- 73 Polar Mode
- 74 Real-Result Only
- 75 MENU
- 76 Edge Wrap
- 77 End Wrap

- 78 Base Signed
- 79 Base Wrap
- 81-99 User Flags

Flags 0-10 are indicated by annunciators at the top of the display.
Flags 36-80 are read-only and cannot be altered with SF, CF, FS?C, or FC?C.

Virtual variables

The concept of "Virtual Variables" is used to manage some of the new features specific to the DM42 calculator. These are variables that do not (and cannot, for that matter) exist in the DM42’s variable table, but are used to read or change state variables specific to DM42.

DevID	(read-only) Hardware device ID
GrMod	(read-write) Graphic Modes • 0 = HP-42S resolution 131x16 • 1 = reserved for future use • 2 = DM42 half resolution 200x120 • 3 = DM42 full resolution 400x240
ResX	(read-only) X-resolution for current graphics mode set in GrMod
ResY	(read-only) Y-resolution for current graphics mode set in GrMod
Vbat	(read-only) battery voltage

```
+
+
```

F-Buttons

The top row consists of six unprinted buttons referenced as F-Buttons (F1-F6) or soft-menu buttons.

- **F1** Displays this Help
- **F2** Toggle the system soft menu visibility
- **F3** (not assigned)
- **F4** Increase buzzer volume
- **Shift-F4** Decrease buzzer volume
- **F5** Toggle stack alignment (Left, Right, Right Underlined)
- **Shift-F5** Toggle between sans and HP font

- **F6** Increase font size
- **Shift-F6** Decrease font size

```
 +
Note that font changes using F5/F6 affect displaying of calculator stack
and font in program mode and are stored independently for both stack and
program mode. +
 +
```

Screenshot

Pressing SHIFT and DISP simultaneously makes a screenshot and saves a BMP graphics file in the folder /SCREENS.
The file name is made up by the current date and time.

Setup Menu

- File
- Calculator State
- Print_To_File
- Settings
- System
- About

```
+
+
```

Setup Menu → File

Save and Load DM42 Programs to FAT Disk as *.raw files.

- Load Program
  Select RAW file and load in DM42 memory
- Save Program
  Select one or more DM42 programs from memory and save as one RAW file
- Activate USB Disk
  Prepare DM42 for connection with computer for file exchange
- Show Disk Info
  Display FAT Disk information

```
+
```

Setup Menu → Calculator State

The complete calculator state can be saved and loaded as one file (*.f42).

- Load State
  Load previously saved calculator state
- Save State
  Save current calculator state as a file
- Load Clean State
  Load default settings

```
+
```

Setup Menu → Print To File

- Graphics Print
  Selecting this menu item enables printing to graphics file.
- Text Print
  Selecting this menu item enables printing to text file.
- Graphics in Text
  Selecting this menu item enables to see graphics in text file.
- Don’t print to IR
  Selecting this menu item disables default printing to IR printer.

Setup Menu → Settings

- Set Time
  Set time and select 12h/24h format
- Set Date
  Set date and select MDY/DMY format
- Status Bar
  Display **State Filename**
  Display **Day of Week**
  Display **Date**
  Select a **Date Separator** of . - /
  Display **Month Shortcut** as 3 letter abrv.
  Display **Time**
  Display **Power Voltage in Header**
- Stack Font Sizes
  Allows to set font size ofsets for registers X, Y, Z, T, L and A

- Beep Mute
  Mute beep in all cases.
- Slow Auto-repeat
  This option makes auto-repeat of pressed and holded buttons slower.
- Stack Layout
  These are the options:
  XYZTL
  XYZTA
  XYZL
  XYZA
  XY
  LXYZT
- Dynamic Stack Extension
  This option enables or disables the 'Big Stack' extension of Free42

  ```
  +
  ```

Setup Menu → System

Warning: Certain actions in this menu can **destroy all stored data** of the calculator.
Please run 'Save Calculator State' in 'File' menu first to preserve data over potential RESET.

- Flashing firmware from FAT
  Starts flashing of new firmware if firmware file is copied to root of FAT filesystem.
- Bootloader
  Connect USB to enter bootloader mode. Note that reset is the only way to exit bootloader mode.
- Program Info
  Displays info about currently loaded program
- Reset to DMCP menu
  Resets calculator and enters DMCP menu
- Format FAT Disk
  This will format FAT filesystem. All data will be erased!
- FAT Disk Media Test
  All data stored in the FAT disk will be destroyed by this low-level media test!
- Power OFF mode
  Allows to set power OFF mode. Only for diagnostics.
- Self Test
  Set of tests for KBD, LCD, IR, BEEP and Diagnostics

  ```
  +
  +
  ```

About

- **DM Calculator Platform**
  © 2014-2021, SwissMicros GmbH
- **Free42**
  © 2004-2021, Thomas Okken
- **Intel Decimal Floating-Point Lib**
  © 2007-2018, Intel Corp.
- **DM42 Quick Reference Guide**
  based on HP-42S Quick Reference Guide (c) Copyright 1988 Dex Smith

  ```
   +
  link:#Top[Top]
  ```