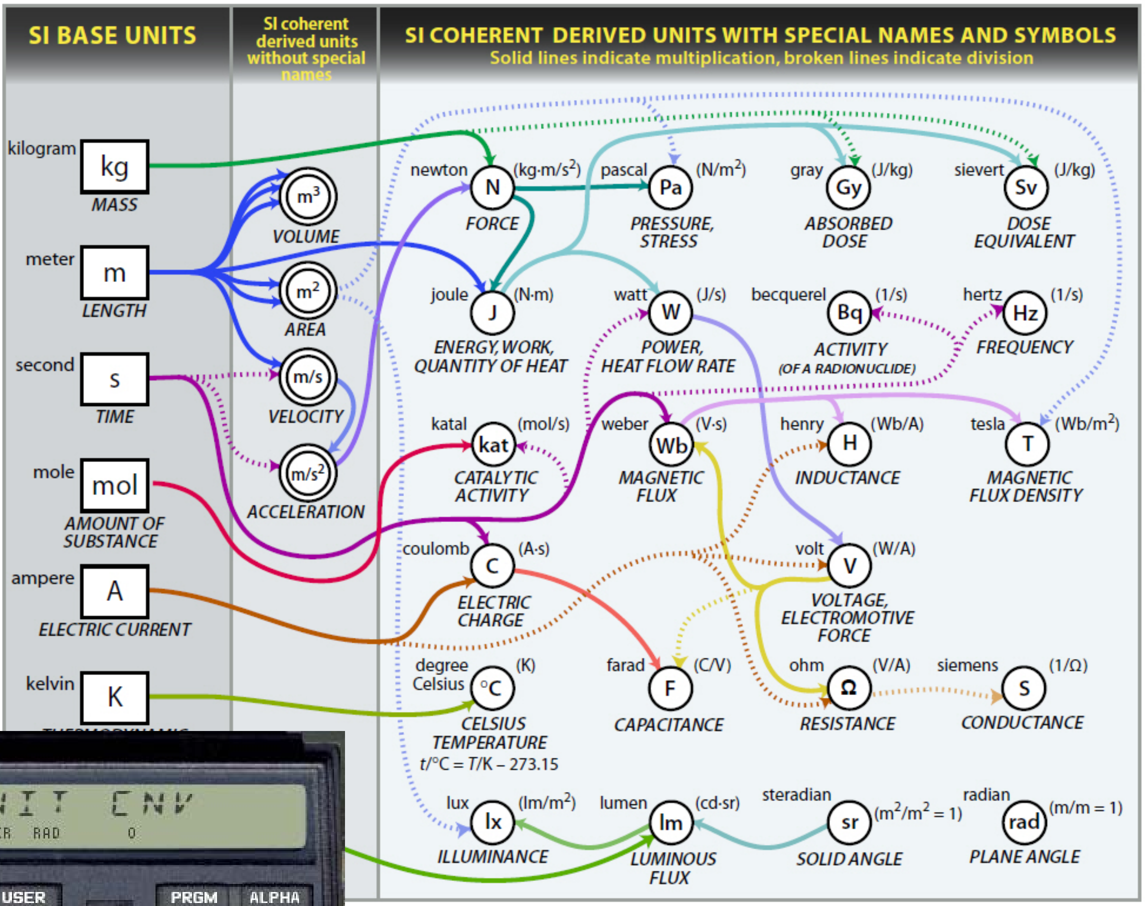
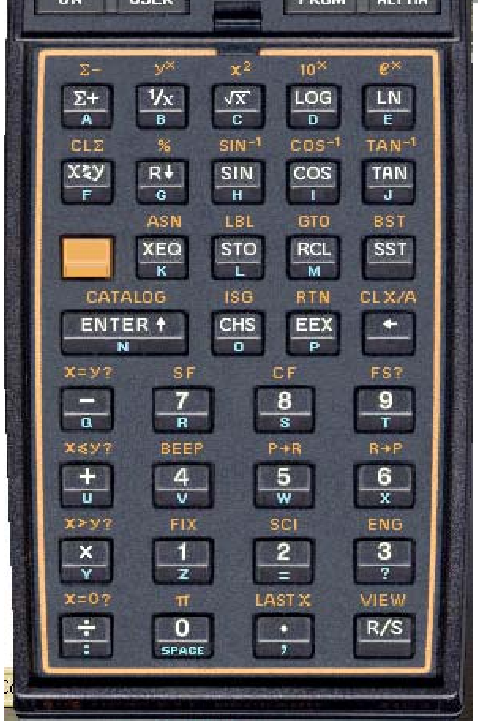
UMS-41 RELOADED
Extending the Unit Management System

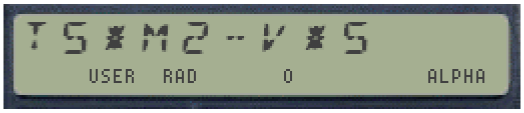

SI BASE UNITS
SI coherent derived units without special names
SI COHERENT DERIVED UNITS WITH SPECIAL NAMES AND SYMBOLS
Solid lines indicate multiplication, broken lines indicate division
kilogram kg MASS
meter m LENGTH
second s TIME
mole mol AMOUNT OF SUBSTANCE
ampere A ELECTRIC CURRENT
kelvin K THERMODYNAMIC
m3 VOLUME
m2 AREA
m/s VELOCITY
m/s2 ACCELERATION
newton N (kg·m/s2) FORCE
pascal Pa (N/m2) PRESSURE, STRESS
gray Gy (J/kg) ABSORBED DOSE
sievert Sv (J/kg) DOSE EQUIVALENT
joule J (N·m) ENERGY, WORK, QUANTITY OF HEAT
watt W (J/s) POWER, HEAT FLOW RATE
becquerel Bq (1/s) ACTIVITY (OF A RADIONUCLIDE)
hertz Hz (1/s) FREQUENCY
katal kat (mol/s) CATALYTIC ACTIVITY
weber Wb (V·s) MAGNETIC FLUX
henry H (Wb/A) INDUCTANCE
tesla T (Wb/m2) MAGNETIC FLUX DENSITY
coulomb C (A·s) ELECTRIC CHARGE
volt V (W/A) VOLTAGE, ELECTROMOTIVE FORCE
degree Celsius °C (K) CELSIUS TEMPERATURE t/°C = T/K – 273.15
farad F (C/V) CAPACITANCE
ohm Ω (V/A) RESISTANCE
siemens S (1/Ω) CONDUCTANCE
lux lx (lm/m2) ILLUMINANCE
lumen lm (cd·sr) LUMINOUS FLUX
steradian sr (m2/m2 = 1) SOLID ANGLE
radian rad (m/m = 1) PLANE ANGLE



CB*V-KW*HR
USER RAD 0 ALPHA

EV/HR-HP
USER RAD 0 ALPHA

V*A-J/MIN
USER RAD 0 ALPHA


TS*M2-V*S
USER RAD 0 ALPHA

Written and prepared by:
Ángel Martin - February 2011

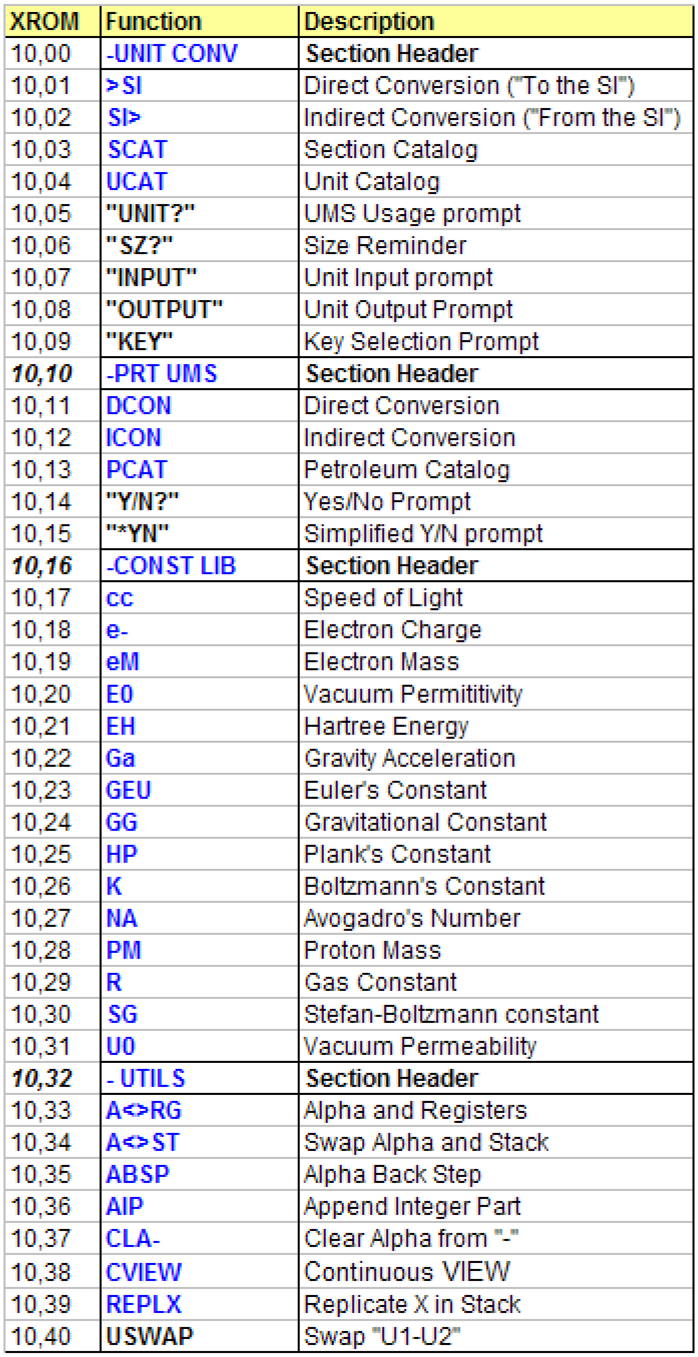
This compilation, revision A.1.2.

Screen captures taken from V41, Windows-based emulator developed by Warren Furlow. See http://www.hp41.org/

Cover picture taken from NIST Special Publication 811 – 2008 Edition.

Table of contents.

Unit Conversion ROM Function Table



XROM	Function	Description
10,00	-UNIT CONV	**Section Header**
10,01	>SI	Direct Conversion ("To the SI")
10,02	SI>	Indirect Conversion ("From the SI")
10,03	SCAT	Section Catalog
10,04	UCAT	Unit Catalog
10,05	**"UNIT?"**	UMS Usage prompt
10,06	**"SZ?"**	Size Reminder
10,07	**"INPUT"**	Unit Input prompt
10,08	**"OUTPUT"**	Unit Output Prompt
10,09	**"KEY"**	Key Selection Prompt
10,10	-PRT UMS	**Section Header**
10,11	DCON	Direct Conversion
10,12	ICON	Indirect Conversion
10,13	PCAT	Petroleum Catalog
10,14	**"Y/N?"**	Yes/No Prompt
10,15	**"*YN"**	Simplified Y/N prompt
10,16	-CONST LIB	**Section Header**
10,17	cc	Speed of Light
10,18	e-	Electron Charge
10,19	eM	Electron Mass
10,20	E0	Vacuum Permititivity
10,21	EH	Hartree Energy
10,22	Ga	Gravity Acceleration
10,23	GEU	Euler's Constant
10,24	GG	Gravitational Constant
10,25	HP	Plank's Constant
10,26	K	Boltzmann's Constant
10,27	NA	Avogadro's Number
10,28	PM	Proton Mass
10,29	R	Gas Constant
10,30	SG	Stefan-Boltzmann constant
10,31	U0	Vacuum Permeability
10,32	- UTILS	**Section Header**
10,33	A<>RG	Alpha and Registers
10,34	A<>ST	Swap Alpha and Stack
10,35	ABSP	Alpha Back Step
10,36	AIP	Append Integer Part
10,37	CLA-	Clear Alpha from "-"
10,38	CVIEW	Continuous VIEW
10,39	REPLX	Replicate X in Stack
10,40	**USWAP**	Swap "U1-U2"

1. Introduction.

The purpose of this project was the extension of the Unit Management System (UMS) implementation found in some HP-41 Application Pacs, such as Thermal & Transport Science ROM.

As impressive as the UMS was – way ahead of its time and cleverly overcoming the inherent limitations of the calculator design, it wasn't complete. - Electrical magnitudes and there units were notoriously lacking and sorely missed. Their inclusion into the UMS has been the main goal of this extension project. Other enhancements relate to cataloguing the available units, as well as easier editing of the unit string in the Alpha register.

If you're reading this you probably are acquainted with the FCON & BCON, SI- & -SI, or CON & INCON functions. With some differences in design and scope they're basically the same pair on each of the three modules that offered UMS, as follows: Machine Construction, Thermal & Transport Science, and Petroleum Fluids Packs.

The functionality improved from one module to another, with the Petroleum being the better of the three – not only offering more units to choose from, but also with improvements in the algorithms (faster searches and better error messages).

This new rendition uses components from both the Thermal and Petroleum modules, sort of combining the best from both designs. It is therefore a *theme module* (as much as also being software archeology), which purpose would be to be used with any programs requiring the unit conversion functionality.

The following improvements have been made:

1. Added 12 new magnitudes to the table, including electrical and luminal ones.
2. Added 29 more units beyond those included in the Thermal ROM table, 19 of them corresponding to electrical magnitudes. (*)
3. Added two new dedicated error messages for syntax and non-defined units' error handling.
4. Provides a *Unit Catalog* to review the unit symbol spelling - either sequentially or sorted by magnitude groups.
5. Provides Catalog *hot keys* to edit the unit string in alpha on the fly, directly from the Unit Catalog.
6. Adds some *alpha utilities* to assist with the unit string creation and handling – both under program control and in run time.

In addition to the UMS improvements, this new module *also includes a mini-library of constants*. Each one of them is accessed by an individual function, getting the constant's value in the X register and their appropriate SI units in Alpha – thus ready for any further conversion activity.

Finally, some user code routines are also included for compatibility reasons - to support your own user code programs that made use of the original ones.

(*) Note that the final unit set is not a complete superset compared to the Petroleum units, as the more specific units are not included (API, SGR, Gas Volumes, etc).

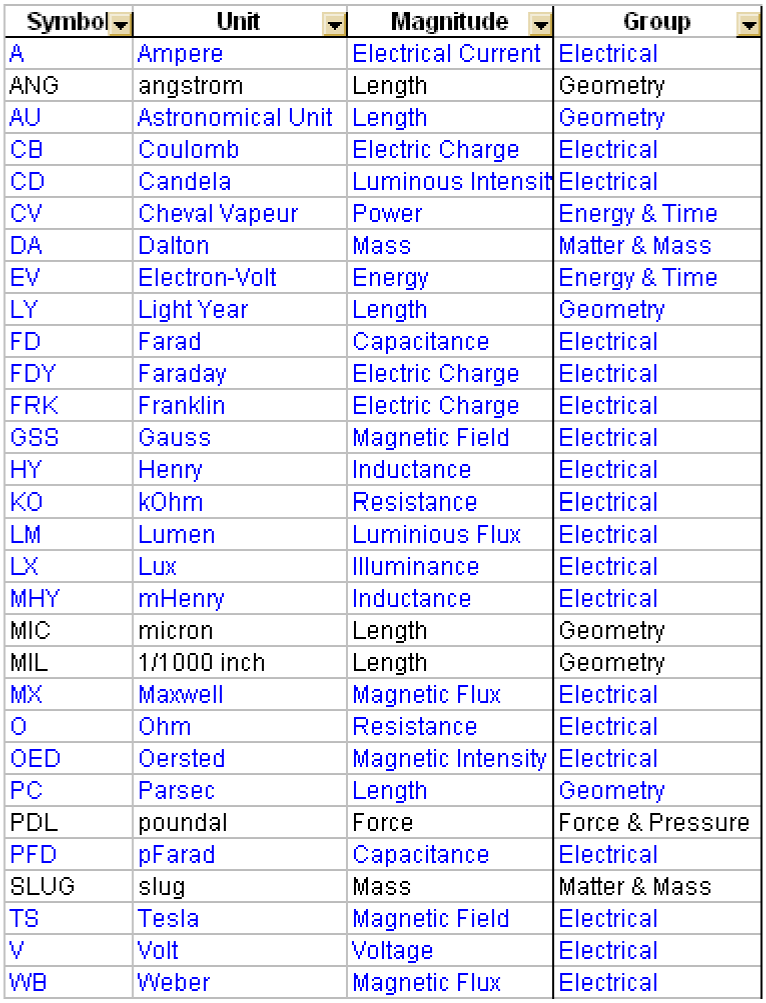
2. Extending the UMS Unit Table.

Because of the specificity of the Petroleum units –more dedicated to that particular field - the basis for this extension has been the unit table from the Thermal & Transport Science Application Pack – itself a superset of the Machine construction ROM UMS. This however should not represent any drawback, since the Petroleum UMS is also included in the Unit Conversion Module – exactly as it is in the original one.

Other units also not included in the extended table are some of the standard multiples (MegaPascal, MegaNewton, MegaJoule, etc). They were deemed to be less important than the other new main units for different magnitudes which competed for the same table space.

The code largely borrows from both implementations, effectively using some sections from the Thermal Pack and others from the Petroleum module. All multiplying factors are by design to be contained within a module quad, and therefore the space limitation.

Here are the new units and their magnitudes not included in the Petroleum Unit Table (those in black are present in the Thermal Pac table, whereas those in blue are completely new:

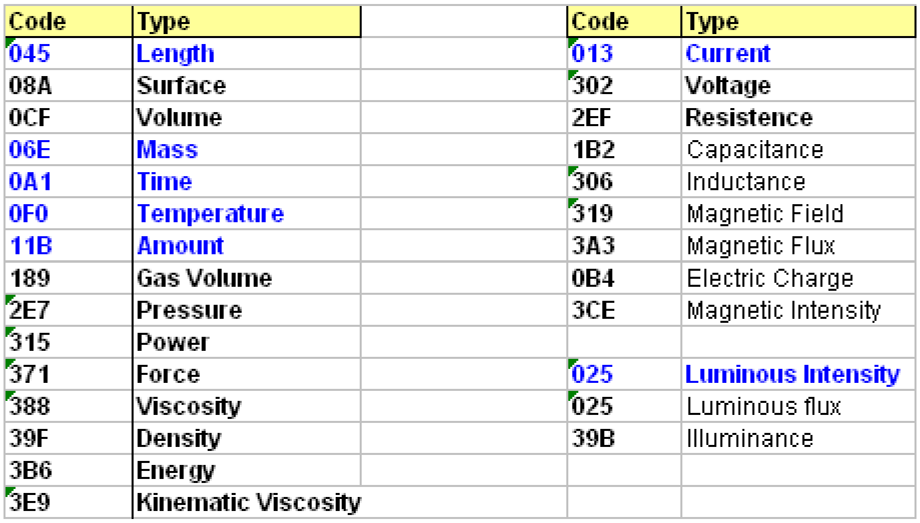


Symbol	Unit	Magnitude	Group
A	Ampere	Electrical Current	Electrical
ANG	angstrom	Length	Geometry
AU	Astronomical Unit	Length	Geometry
CB	Coulomb	Electric Charge	Electrical
CD	Candela	Luminous Intensit	Electrical
CV	Cheval Vapeur	Power	Energy & Time
DA	Dalton	Mass	Matter & Mass
EV	Electron-Volt	Energy	Energy & Time
LY	Light Year	Length	Geometry
FD	Farad	Capacitance	Electrical
FDY	Faraday	Electric Charge	Electrical
FRK	Franklin	Electric Charge	Electrical
GSS	Gauss	Magnetic Field	Electrical
HY	Henry	Inductance	Electrical
KO	kOhm	Resistance	Electrical
LM	Lumen	Luminious Flux	Electrical
LX	Lux	Illuminance	Electrical
MHY	mHenry	Inductance	Electrical
MIC	micron	Length	Geometry
MIL	1/1000 inch	Length	Geometry
MX	Maxwell	Magnetic Flux	Electrical
O	Ohm	Resistance	Electrical
OED	Oersted	Magnetic Intensity	Electrical
PC	Parsec	Length	Geometry
PDL	poundal	Force	Force & Pressure
PFD	pFarad	Capacitance	Electrical
SLUG	slug	Mass	Matter & Mass
TS	Tesla	Magnetic Field	Electrical
V	Volt	Voltage	Electrical
WB	Weber	Magnetic Flux	Electrical

At the core of the UMS is the **Unit Table**, which stores the offsets to the location addresses for the different unit **multiplying factors** within the ROM. The structure of the multiplying factor consists of an alphabetical header, a magnitude code (used for dimensionality checksums), plus the actual multiplier. The header last character has 200 Hex added to its corresponding code to signal the end of the character string for the unit symbol. These symbols are read by the Unit Catalog routines, as will be covered in the next section.

Of all the new magnitudes added, the Ampere and the Candela are primary units – used to derive all others from the same group. They have special relevance in the UMS implementation, as the magnitude codes are used in the dimensionality checks performed prior to any unit conversion.

The following table lists all codes for the Extended Unit Table. Note that the primary ones are all prime numbers – which is required to ensure the uniqueness of the compounded magnitude groupings.



Code	Type	Code	Type
045	Length	013	Current
08A	Surface	302	Voltage
0CF	Volume	2EF	Resistence
06E	Mass	1B2	Capacitance
0A1	Time	306	Inductance
0F0	Temperature	319	Magnetic Field
11B	Amount	3A3	Magnetic Flux
189	Gas Volume	0B4	Electric Charge
2E7	Pressure	3CE	Magnetic Intensity
315	Power		
371	Force	025	Luminous Intensity
388	Viscosity	025	Luminous flux
39F	Density	39B	Illuminance
3B6	Energy		
3E9	Kinematic Viscosity		

In spite of not being the main unit of the magnitude, some multiples and sub-multiples (such as pF, mH, and kOhm) are added to the new table given their practical application in real calculations.

The designers of the UMS came up with a superb implementation of their concept to build up powerful unit conversion functionality. As you can expect, some amount of Software reverse-engineering and assumption making was involved to second-guess the design and working structure of the UMS (I didn't even know about the 1983 Maryland Conference proceedings until after completing the additions – which made the process all that more enjoyable!).

Whilst all main areas of the code are well characterized there are a few that remain somewhat unclear, such as the section immediately preceding the Unit Table in the Petroleum Pac – which function is to expedite the unit search – yet I'm not clear on which criteria was used to select the "representative magnitudes" – and therefore it's not used in the Unit Conversion module.

Auxiliary Subroutines.-

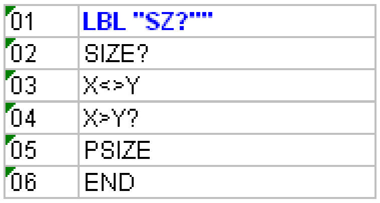
The Unit Conversion module includes the same User Code subroutines available on the Thermal Pack (plus one from the Petroleum Fluids Pac) to further integrate the UMS into your own FOCAL programs. Refer to the corresponding manuals for usage instructions and other details.

The routines are:

"UNIT?"	Used to select whether the UMS should be used for data entry and results
"SZ?"	A size error prompt (superseded by SIZE? & PSIZE combination)
"INPUT"	To select the unit of the input magnitude
"OUTPUT"	To select the unit of the output magnitude
"KEY?"	For a typical key selection choice program
"Y/N"	A general purpose Yes/No input routine
"*YN"	A poor's man version of the previous one (using user flag 00)

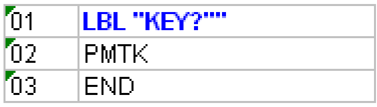
Some of these are clearly improvable using Extended functions and other Custom modules introduced chronologically after the introduction of the UMS – but nevertheless are here for completion sake, and to offer a more complete set of utilities.

Clearly using **SIZE?** & **PSIZE** combined supersedes the functionality of **"SZ?"**, and **PMTK** in the CCD Module (or the CCD OS/X derivative) is a much more powerful and complete implementation of **"KEY?"**. The following listing would be a modern-day (the irony is intended) rendition of the same routines, just for illustrative purposes:



```
01   LBL "SZ?""
02   SIZE?
03   X<>Y
04   X>Y?
05   PSIZE
06   END
```

call it with the required size in X.



```
01   LBL "KEY?""
02   PMTK
03   END
```

call it with the hot-keys in Alpha

Unfortunately the lack of space prevents from adding **PMTK** to the module, but on the other hand the 4k version CCD-OS/X is such a fundamental addition to the main 41 OS that it should always be included in the calculator configuration - so it's actually better this way to avoid function duplication.

3. Unit Catalogs.

As the unit tables get larger it becomes more challenging to remember the exact spelling of each unit symbol, resulting in frequent Invalid Conversion error conditions. A Unit Catalog is therefore almost an obvious addition to the UMS – and as such it was mentioned in the old HP documentation as a next addition, which obviously never came into being. Until now, that is.

There are three unit catalog functions in the Unit Conversion Module. The first one (**UCAT**, Unit Catalog originally *written by Peter Platzer*) lists all unit symbols sequentially, following the order in which they are stored on the Unit Table. This used to be alphabetical in the original implementation – but has been modified in this new incarnation for reasons explained in the next paragraph.

The second function **SCAT** (Section Catalog) is an extension to the previous one that adds a prompting entry to select the *magnitude section* to start the listing from. Section numbers range from 1 to 5, as follows:

1. Geometry Section (length, surface and volume units)
2. Pressure and Force Section
3. Matter and Mass section (Mass, Density, Viscosity, Temperature)
4. Energy, Power and Time Section
5. Electrical and Luminal Section

The units listing will continue until the end of the table is reached (i.e. end of section 5) regardless of where it got started. This is not perfect but good enough for the majority of circumstances where one's looking for the appropriate specific unit symbol.

The third function **PCAT** (Petroleum Catalog) is just the same as **UCAT** but applied to the Petroleum Unit Table – which as was said in previous sections, is also included in the Unit Conversion Module as a separate one for compatibility reasons. This listing will be done alphabetically here as this table has not been modified.

Note that while both **UCAT** and **PCAT** are programmable, that's not the case with **SCAT**. Note also that the need for section grouping is the reason why the new unit table is not alphabetical anymore.

Navigating the Unit Catalogs.

The catalogs can be paused and resumed at any time using R/S. Besides, you can use SST and BST to single-step the units forwards and backwards, and resuming the listing with SHIFT activated will list them in reverse order. Use the arrow key to stop the listing and return to the main OS.

- A running catalog will automatically terminate when it reaches the end of the unit table (or the beginning if running backwards)

- A single-stepped catalog will not go beyond the last unit (or first one if moving backwards) even if you keep pressing SST (or BST). To terminate it you can press R/S or the back arrow key.

- Use R/S to toggle between continuous and single-step catalog display at any time during the execution.

Catalog Hot-keys.

__With the unit catalog paused__, (i.e. in single-step mode) you can use the following keys to **directly edit the unit string in the Alpha register** in the following manner:

- **[ENTER]** – Clears Alpha and adds the displayed unit symbol to the string.
- **[SHIFT],[ENTER]** – Appends the displayed unit as *destination field* in the unit string (i.e. appends "**-**" plus the unit symbol to the text already existing in Alpha.
- **[*]** - Appends the displayed unit as multiplying unit (i.e. appends "*****" plus symbol)
- **[/]** - Appends the displayed unit as dividing unit (i.e. appends "**/**" plus symbol)

In this way it's rather simple to build complete unit string just by pressing the corresponding hot keys during the catalog listing – without needing to remember the exact spelling of the unit symbols. Obviously it's still the user's task to recognize the symbol and identify it with the corresponding unit name. Of course you can edit the Alpha register directly as always just typing the syntax in Alpha mode - if that's your preferred choice.

Note that there are no checks for the string built – so it's possible to press **[/]** multiple times, or repeat [SHIFT],[ENTER] – which obviously would not be a valid string. Note also that after every usage of a hot key you need to re-start the catalog listing again if you want to continue to build the complete unit string. A small price to pay for the convenience to occasional users (aren't we all?) to avoid syntax errors!

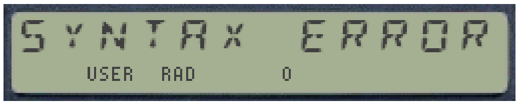
Meaningful Error Conditions.

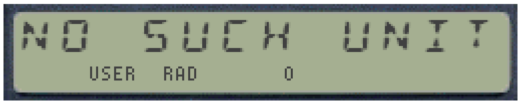
The Thermal Pac only used standard OS error messages (like ALPHA DATA and DATA ERROR) to inform the user of an error condition. It did NOT report unit spelling errors either, ignoring them even if user flag 25 was cleared! The Petroleum Pac improved on that with its dedicated error message "Invalid Conversion".

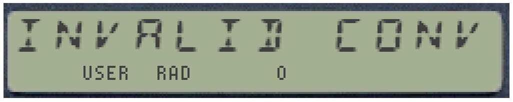
This however wasn't very informative, as it didn't indicate where exactly was the problem: either a syntax error (like typing the sigma or percent sign), an invalid unit string (non-homogeneous source and destinations), or a misspelled unit symbol all produced the same "INVALID CONV". Besides it didn't signal the ALPHA DATA condition anymore – clearly a step back here.

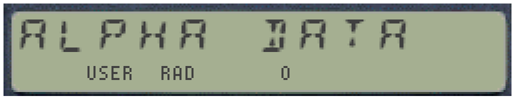
The Unit Conversion Module recovers the alpha data message and adds two new messages to the Invalid Conversion condition – offering four error-trapping cases, and so making error detection and correction a much easier task, needless to say. The messages are as follows:

ALPHA DATA – when a non-numeric input is in X
SYNTAX ERROR – when using illegal characters (like lower-case letters)
NO SUCH UNIT – when the alpha string contains a symbol not on the unit table
INVALID CONV – when the unit string is not dimensionally correct.


SYNTAX ERROR


NO SUCH UNIT


INVALID CONV


ALPHA DATA

4a. Constant Library.

What good is a Unit Conversion Module without a constant library? A pretty good one if you ask the author- but nevertheless a few universal constants are also programmed into the module for completion sake.

The implementation is a rather simple one, with an individual function in the module for each constant. A better way to implement it would have been a table-based constant catalog that also allowed individual retrieval, but not being the main purpose of the project it was good enough to illustrate the example.

The main contribution to this section is that not only the constant value is written in X, but also that *its units are returned into the alpha register* when the function is called – so you always know how they're expressed.

The constants included, their values and appropriate units in which they are expressed are listed in the following table:

Name	Description	Value	Units
cc	Speed of Light	2,9979245 E 08	M/S
e-	Electron Charge	-1,6021764 E-19	CB
eM	Electron Mass	9,1093821 E-31	KG
E0	Vacuum Permitivity	8,8541878 E-12	FD/M
EH	Hartree Energy	4,359748226 E-18	J
Ga	Free-fall Acceleration	9,80665	M/S2
GEU	Euler's Gamma Constant	0,577215665	
GG	Gravity Constant	6,6742800 E-11	N*M2/KG
HP	Planck's h constant	6,6260689 E-34	J*S
K	Boltzmann's Constant	1,3806504 E-23	J/K
NA	Avogadro's Number	6,0221417 E 23	1/MOL
PM	Proton's Mass	1,6726216 E-27	KG
R	Gas Constant	8,314472150	J/K*MOL
SG	Stefan-Boltmann	5,6705119 E-8	W/M2*K4
U0	Vacuum Permeability	1,2566370 E-06	N/A2

4b. Alpha Utilities.

Finally, a handful of alpha routines are included in the Unit Conversion module to facilitate the editing of the unit string under program execution. These are:

A<>RG	Swaps Alpha and memory registers	Prompts for first Register (*)
A<>ST	Swaps Alpha and Stack	In descending order M,N,O,P
ABSP	Alpha Back Space	To remove the last character
AIP	Alpha Integer Part	To append powers of units (M2, etc)
CVIEW	continuous VIEW	Regardless of Flag 21 Setting
CLA-	Clear Alpha from "-" char	Not the Same as the CCD Module
USWAP	Swaps alpha around "-" char	Interchanges "From/To" Unit info

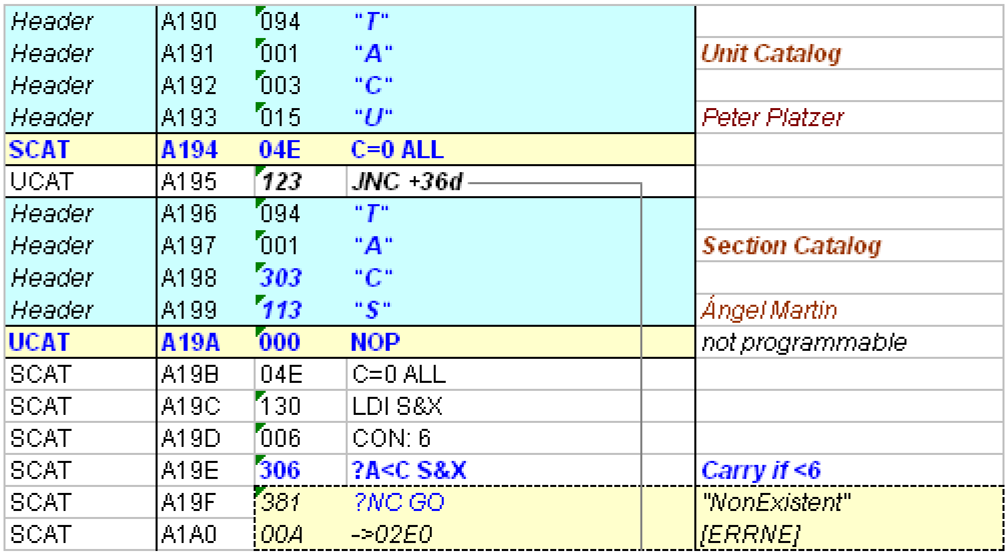
(*) Programmable: Uses next program line for REG#.

Unit Catalog MCODE Listings & Program Remarks.-

The original unit catalog routine was swiftly written by Peter Platzer - providing a proof of concept with both the run mode and the single-step mode. The author added the SHIFT handing code and the hot-key choices for editing the unit string in Alpha. Glad to see this kind of collaboration going on!

Note that both **UCAT** and **SCAT** share the same code, thus with exception of the prompting feature they are but identical. They list the units from the extended Unit table, whereas **PCAT** points at the Petroleum unit table instead – listing them alphabetically as initially laid down. It also shares the main code sections, so the same convenience features are available as well.

Notice the prompting bits in the **SCAT** function title, allowing a single digit numeric prompt (including IND and STK support). We first start by checking that the prompt argument is less than 6 – aborting if that's not the case. (Note that zero will be equated to one). **UCAT**'s simply jumps to the first section's entry point, bypassing the initial selection.



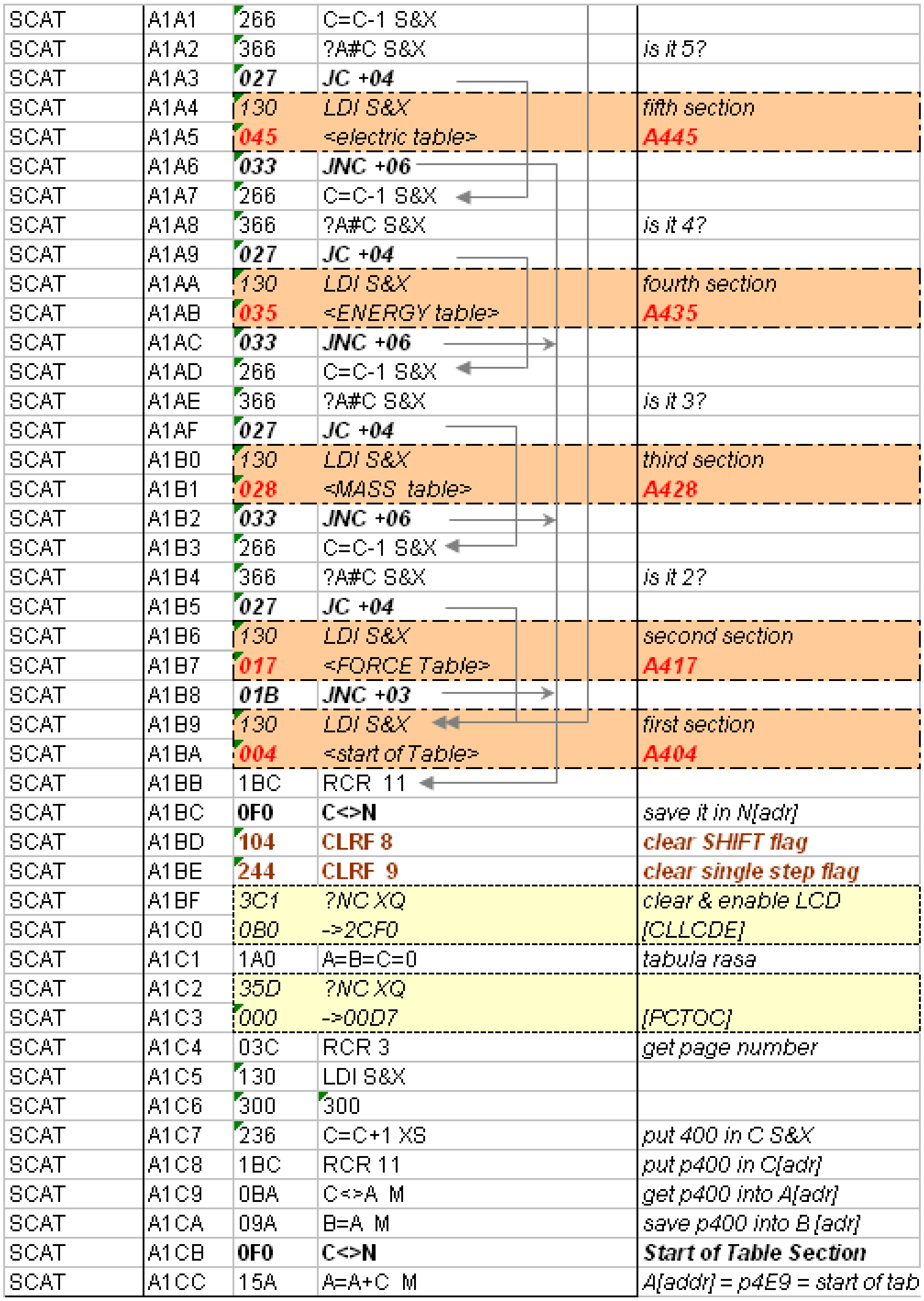
Header	A190	094	*"T"*	
Header	A191	001	*"A"*	*Unit Catalog*
Header	A192	003	*"C"*	
Header	A193	015	*"U"*	*Peter Platzer*
SCAT	**A194**	**04E**	**C=0 ALL**	
UCAT	A195	*123*	*JNC +36d*	
Header	A196	094	*"T"*	
Header	A197	001	*"A"*	*Section Catalog*
Header	A198	*303*	*"C"*	
Header	A199	*113*	*"S"*	*Ángel Martin*
UCAT	**A19A**	**000**	**NOP**	*not programmable*
SCAT	A19B	04E	C=0 ALL	
SCAT	A19C	130	LDI S&X	
SCAT	A19D	006	CON: 6	
SCAT	A19E	306	?A<C S&X	*Carry if <6*
SCAT	A19F	*381*	*?NC GO*	*"NonExistent"*
SCAT	A1A0	*00A*	*->02E0*	*[ERRNE]*

The hard-coded addresses in the next code lines refer to the current location of the Unit Table sections within the module. They'll therefore have to be changed if the table is expanded or re-arranged.

Once the section number is determined it's saved in N[adr] field for later use. Then some initialization is performed, clearing the display, CPU flags F8 and F9, and we find out which expansion port number "p" the module is plugged into using the [PCTOC] routine – this is used to store p400 Hex into both A[adr] and B[adr] as address counters for later usage in the main loop. We'll further add the section offset to A, and once we've got the start of the section address stored in A[adr] we're ready to enter the main loop, where all the action takes place

CPU Flag 9 is used to control the automated or the single-step mode of operation. When set, the other hot-keys can be accessed as well – part of the ?KEY main loop.

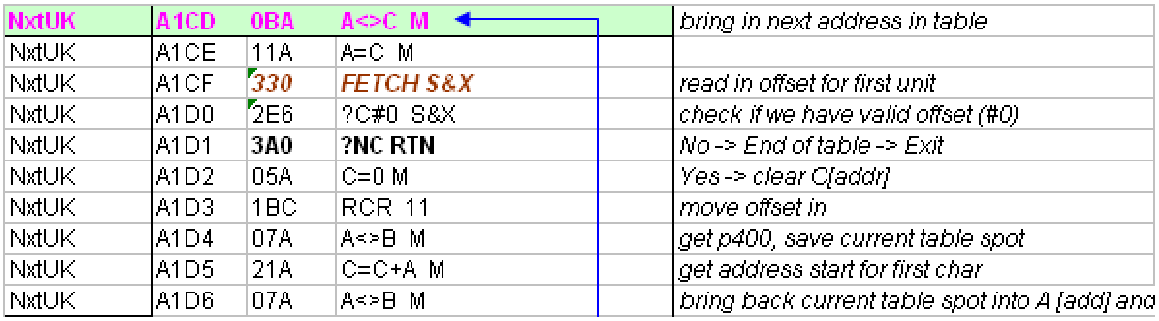
CPU Flag 8 is used to signal that SHIFT has been pressed. Note that SHIFT remains up until it's pressed a second time – a common practice used in other extended catalogs, like those from the CCD ROM.



SCAT	A1A1	266	C=C-1 S&X		
SCAT	A1A2	366	?A#C S&X		is it 5?
SCAT	A1A3	**027**	***JC +04***		
SCAT	A1A4	130	LDI S&X		fifth section
SCAT	A1A5	***045***	<electric table>		***A445***
SCAT	A1A6	**033**	***JNC +06***		
SCAT	A1A7	266	C=C-1 S&X		
SCAT	A1A8	366	?A#C S&X		is it 4?
SCAT	A1A9	**027**	***JC +04***		
SCAT	A1AA	130	LDI S&X		fourth section
SCAT	A1AB	***035***	<ENERGY table>		***A435***
SCAT	A1AC	**033**	***JNC +06***		
SCAT	A1AD	266	C=C-1 S&X		
SCAT	A1AE	366	?A#C S&X		is it 3?
SCAT	A1AF	**027**	***JC +04***		
SCAT	A1B0	130	LDI S&X		third section
SCAT	A1B1	***028***	<MASS table>		***A428***
SCAT	A1B2	**033**	***JNC +06***		
SCAT	A1B3	266	C=C-1 S&X		
SCAT	A1B4	366	?A#C S&X		is it 2?
SCAT	A1B5	**027**	***JC +04***		
SCAT	A1B6	130	LDI S&X		second section
SCAT	A1B7	***017***	<FORCE Table>		***A417***
SCAT	A1B8	**01B**	***JNC +03***		
SCAT	A1B9	130	LDI S&X		first section
SCAT	A1BA	***004***	<start of Table>		***A404***
SCAT	A1BB	1BC	RCR 11		
SCAT	A1BC	**0F0**	**C<>N**		save it in N[adr]
SCAT	A1BD	**104**	**CLRF 8**		***clear SHIFT flag***
SCAT	A1BE	**244**	**CLRF 9**		***clear single step flag***
SCAT	A1BF	3C1	?NC XQ		clear & enable LCD
SCAT	A1C0	0B0	->2CF0		[CLLCDE]
SCAT	A1C1	1A0	A=B=C=0		tabula rasa
SCAT	A1C2	35D	?NC XQ		
SCAT	A1C3	000	->00D7		[PCTOC]
SCAT	A1C4	03C	RCR 3		get page number
SCAT	A1C5	130	LDI S&X		
SCAT	A1C6	300	300		
SCAT	A1C7	236	C=C+1 XS		put 400 in C S&X
SCAT	A1C8	1BC	RCR 11		put p400 in C[adr]
SCAT	A1C9	0BA	C<>A M		get p400 into A[adr]
SCAT	A1CA	09A	B=A M		save p400 into B [adr]
SCAT	A1CB	**0F0**	**C<>N**		***Start of Table Section***
SCAT	A1CC	15A	A=A+C M		A[addr] = p4E9 = start of tab

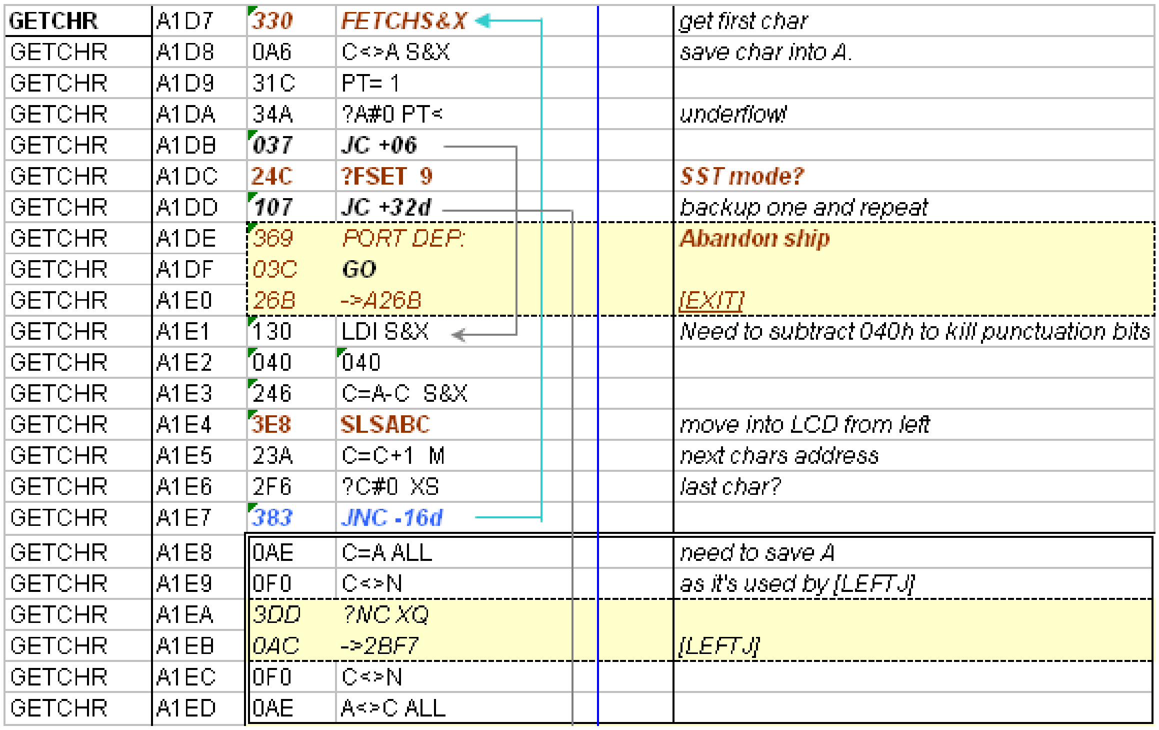
At this point B[adr] contains p400 Hex, and A[adr] the selected section start address.

Firstly we'll FETCH the word placed at A[adr] and establish the end- or beginning of table condition. The End-of-Table condition is clearly signaled when word 000 is reached. However its beginning doesn't have an immediately recognizable mark. This is important when running (or single-stepping) backwards to stop the listing. The code interrogates the *magnitude code* word, looking for 200 Hex as the indication for the table initial entry.



NxtUK	A1CD	0BA	A<>C M	bring in next address in table
NxtUK	A1CE	11A	A=C M	
NxtUK	A1CF	330	FETCH S&X	read in offset for first unit
NxtUK	A1D0	2E6	?C#0 S&X	check if we have valid offset (#0)
NxtUK	A1D1	3A0	?NC RTN	No -> End of table -> Exit
NxtUK	A1D2	05A	C=0 M	Yes -> clear C[addr]
NxtUK	A1D3	1BC	RCR 11	move offset in
NxtUK	A1D4	07A	A<>B M	get p400, save current table spot
NxtUK	A1D5	21A	C=C+A M	get address start for first char
NxtUK	A1D6	07A	A<>B M	bring back current table spot into A [add] and

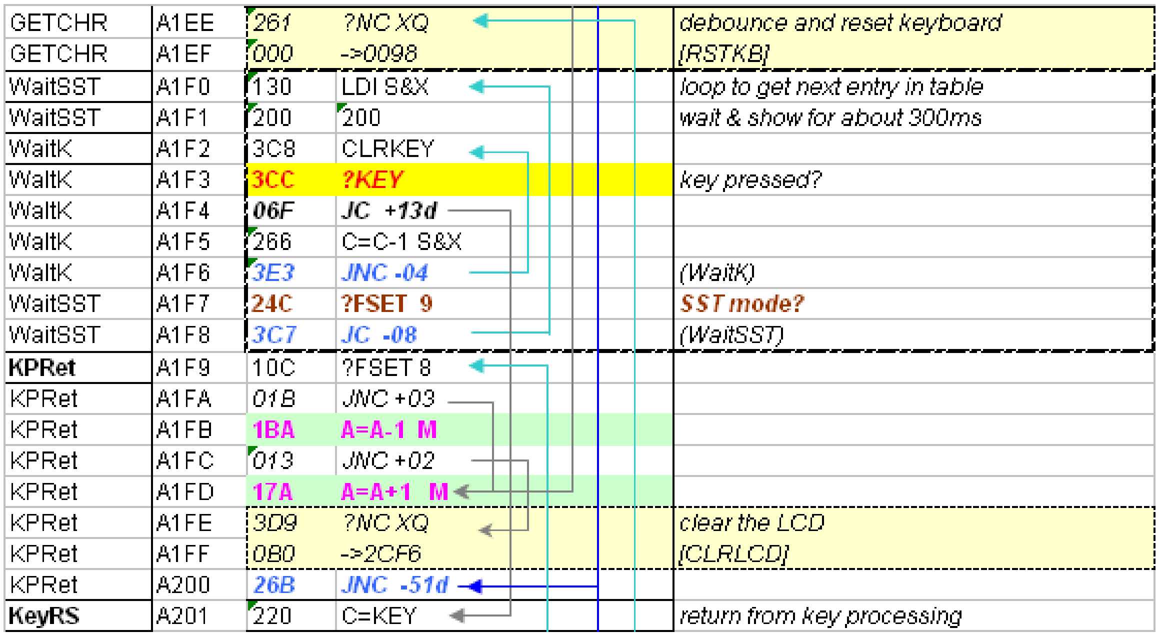
If the word fetched was neither the last or first one we're to read the unit symbol characters sequentially, until we find the last one - which *has a 200 Hex offset to its char number*. Because the char code in Alpha has 40 Hex added to the display char code, so we'll have to adjust it accordingly. Note that checking for char validity, adjusting the code and adding it to the display is all done within the same loop – and that CPU flag U9 is checked for single-step mode as well (the behavior will be different: terminate in run mode, hold-up in single-step mode).



GETCHR	A1D7	330	FETCHS&X	get first char
GETCHR	A1D8	0A6	C<>A S&X	save char into A.
GETCHR	A1D9	31C	PT= 1	
GETCHR	A1DA	34A	?A#0 PT<	underflow!
GETCHR	A1DB	037	JC +06	
GETCHR	A1DC	24C	?FSET 9	SST mode?
GETCHR	A1DD	107	JC +32d	backup one and repeat
GETCHR	A1DE	369	PORT DEP:	Abandon ship
GETCHR	A1DF	03C	GO	
GETCHR	A1E0	26B	->A26B	[EXIT]
GETCHR	A1E1	130	LDI S&X	Need to subtract 040h to kill punctuation bits
GETCHR	A1E2	040	040	
GETCHR	A1E3	246	C=A-C S&X	
GETCHR	A1E4	3E8	SLSABC	move into LCD from left
GETCHR	A1E5	23A	C=C+1 M	next chars address
GETCHR	A1E6	2F6	?C#0 XS	last char?
GETCHR	A1E7	383	JNC -16d	
GETCHR	A1E8	0AE	C=A ALL	need to save A
GETCHR	A1E9	0F0	C<>N	as it's used by [LEFTJ]
GETCHR	A1EA	3DD	?NC XQ	
GETCHR	A1EB	0AC	->2BF7	[LEFTJ]
GETCHR	A1EC	0F0	C<>N	
GETCHR	A1ED	0AE	A<>C ALL	

With the complete unit symbol placed on the display we proceed to left-justify it using [LEFTJ]. Note how we needed to preserve A during the left-justification step, since it's used by [LEFTJ] but also holds the address of the current unit fetched from the unit table. We use N to save it temporarily and restore things back after the justification.

We now move right into the ?KEY loop where we'll wait for a key to be pressed if in single-step mode, or move on to the next unit (after a small delay pause to see the unit name) if in run mode - as determined by the status of flag 9.

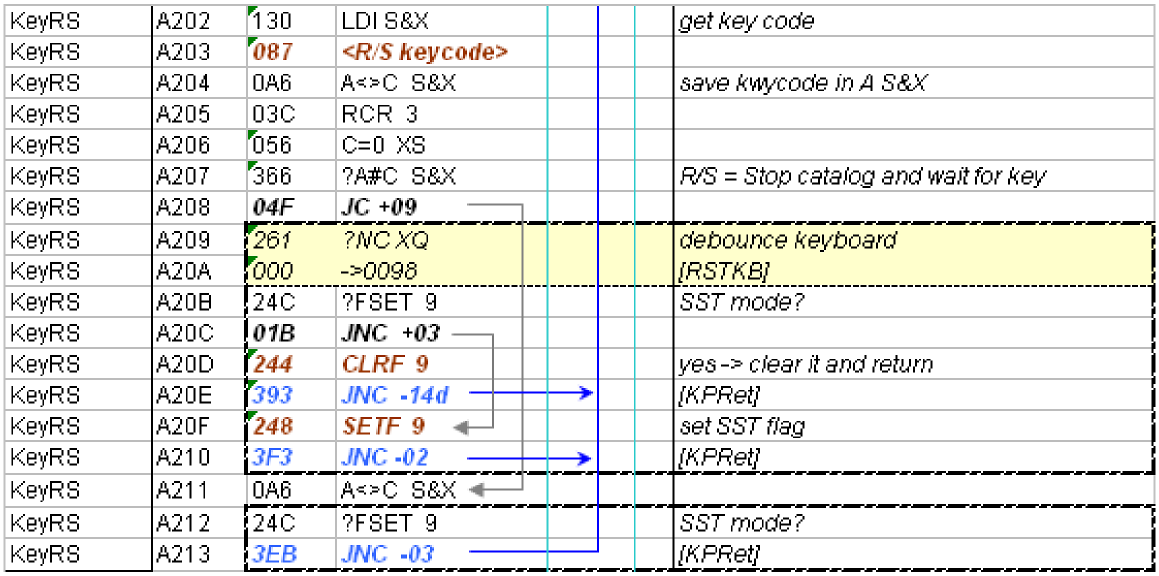


GETCHR	A1EE	261	?NC XQ	debounce and reset keyboard
GETCHR	A1EF	000	->0098	[RSTKB]
WaitSST	A1F0	130	LDI S&X	loop to get next entry in table
WaitSST	A1F1	200	200	wait & show for about 300ms
WaitK	A1F2	3C8	CLRKEY	
WaltK	A1F3	3CC	?KEY	key pressed?
WaltK	A1F4	06F	JC +13d	
WaltK	A1F5	266	C=C-1 S&X	
WaltK	A1F6	3E3	JNC -04	(WaitK)
WaitSST	A1F7	24C	?FSET 9	SST mode?
WaitSST	A1F8	3C7	JC -08	(WaitSST)
KPRet	A1F9	10C	?FSET 8	
KPRet	A1FA	01B	JNC +03	
KPRet	A1FB	1BA	A=A-1 M	
KPRet	A1FC	013	JNC +02	
KPRet	A1FD	17A	A=A+1 M	
KPRet	A1FE	3D9	?NC XQ	clear the LCD
KPRet	A1FF	0B0	->2CF6	[CLRLCD]
KPRet	A200	26B	JNC -51d	
KeyRS	A201	220	C=KEY	return from key processing

If a key is pressed it will be detected by ?KEY while the CPU is busy doing nothing but counting in the delay loop. Then the execution will jump to line A201 to decode the pressed key and determine the corresponding action, as follows:

- If R/S is pressed, toggle flag 9 status and get into single-step mode if F9 was clear – or revert to run mode if F9 was set
- If any other key is pressed, do nothing in run mode or process the corresponding action if in single-step mode.

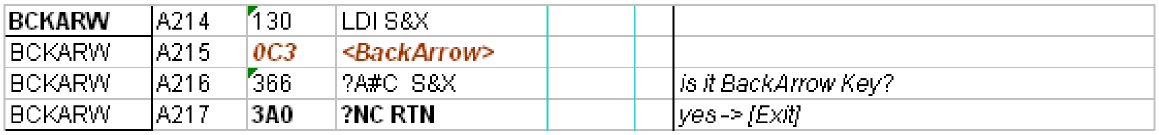
When no keys are pressed the execution will continue, going on to display the following (or previous) unit from the unit table: we'll check flag 8 to see whether we're going forwards or backwards to either add or subtract one to/from the previous address in A[adr]. If in run mode the display is cleared (but not disabled) after the delay loop is complete and we loop back to the beginning of the process.



KeyRS	A202	130	LDI S&X	get key code
KeyRS	A203	087	<R/S keycode>	
KeyRS	A204	0A6	A<>C S&X	save kwycode in A S&X
KeyRS	A205	03C	RCR 3	
KeyRS	A206	056	C=0 XS	
KeyRS	A207	366	?A#C S&X	R/S = Stop catalog and wait for key
KeyRS	A208	04F	JC +09	
KeyRS	A209	261	?NC XQ	debounce keyboard
KeyRS	A20A	000	->0098	[RSTKB]
KeyRS	A20B	24C	?FSET 9	SST mode?
KeyRS	A20C	01B	JNC +03	
KeyRS	A20D	244	CLRF 9	yes -> clear it and return
KeyRS	A20E	393	JNC -14d	[KPRet]
KeyRS	A20F	248	SETF 9	set SST flag
KeyRS	A210	3F3	JNC -02	[KPRet]
KeyRS	A211	0A6	A<>C S&X	
KeyRS	A212	24C	?FSET 9	SST mode?
KeyRS	A213	3EB	JNC -03	[KPRet]

If in single-step mode we continue comparing the key pressed with the other six single-cased ones: BackArrow, ENTER, SST, SHIFT, *, and /. Each case will have independent code to handle the actions.

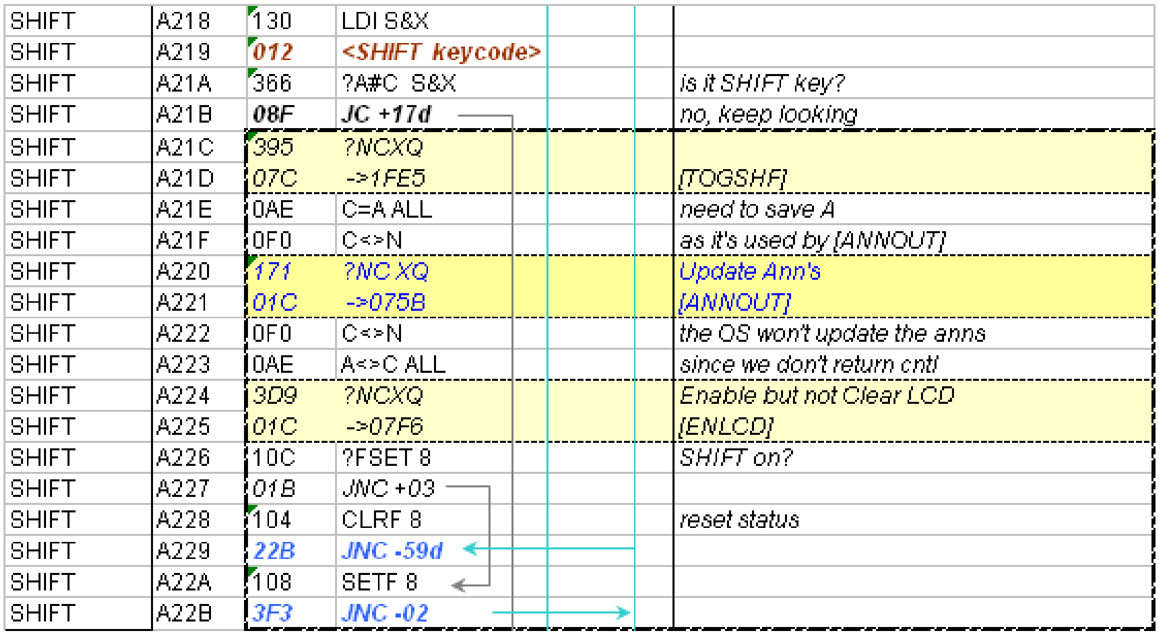
For the Back Arrow case, it's quite trivial: we simply terminate the execution with ?NC RTN, which seems to work quite well – even restoring the SHIFT annunciator if it was active – and this is rather surprising as it doesn't occur anywhere else in the code!



BCKARW	A214	130	LDI S&X	
BCKARW	A215	*0C3*	*<BackArrow>*	
BCKARW	A216	366	?A#C S&X	*is it BackArrow Key?*
BCKARW	A217	**3A0**	**?NC RTN**	*yes -> [Exit]*

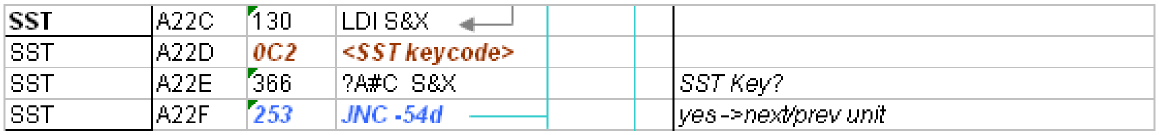
For the SHIFT case, we'll toggle the status of F8 to switch back and forth, and return to the main ?KEY loop. We also need to toggle the SHIFT annunciator to provide visual feedback to the user – and this must be done by our code because the ?KEY loop does not yield CPU control back to the OS – and thus it isn't done in the [TOGSHF] routine. We use [ANNOUT] for this.

Notice also that [TOGSHF] returns with the display not enabled and thus must be re-enabled before transferring the execution back to the main ?KEY loop.



SHIFT	A218	130	LDI S&X	
SHIFT	A219	*012*	*<SHIFT keycode>*	
SHIFT	A21A	366	?A#C S&X	*is it SHIFT key?*
SHIFT	A21B	*08F*	*JC +17d*	*no, keep looking*
SHIFT	A21C	*395*	*?NCXQ*	
SHIFT	A21D	*07C*	*->1FE5*	*[TOGSHF]*
SHIFT	A21E	0AE	C=A ALL	*need to save A*
SHIFT	A21F	0F0	C<>N	*as it's used by [ANNOUT]*
SHIFT	A220	*171*	*?NC XQ*	*Update Ann's*
SHIFT	A221	*01C*	*->075B*	*[ANNOUT]*
SHIFT	A222	0F0	C<>N	*the OS won't update the anns*
SHIFT	A223	0AE	A<>C ALL	*since we don't return cntl*
SHIFT	A224	*3D9*	*?NCXQ*	*Enable but not Clear LCD*
SHIFT	A225	*01C*	*->07F6*	*[ENLCD]*
SHIFT	A226	10C	?FSET 8	*SHIFT on?*
SHIFT	A227	*01B*	*JNC +03*	
SHIFT	A228	104	CLRF 8	*reset status*
SHIFT	A229	*22B*	*JNC -59d*	
SHIFT	A22A	108	SETF 8	
SHIFT	A22B	*3F3*	*JNC -02*	

For the SST case we'll simply transfer the execution to the section where one was added or subtracted to the current table address in A[adr] – depending on the status of F8.

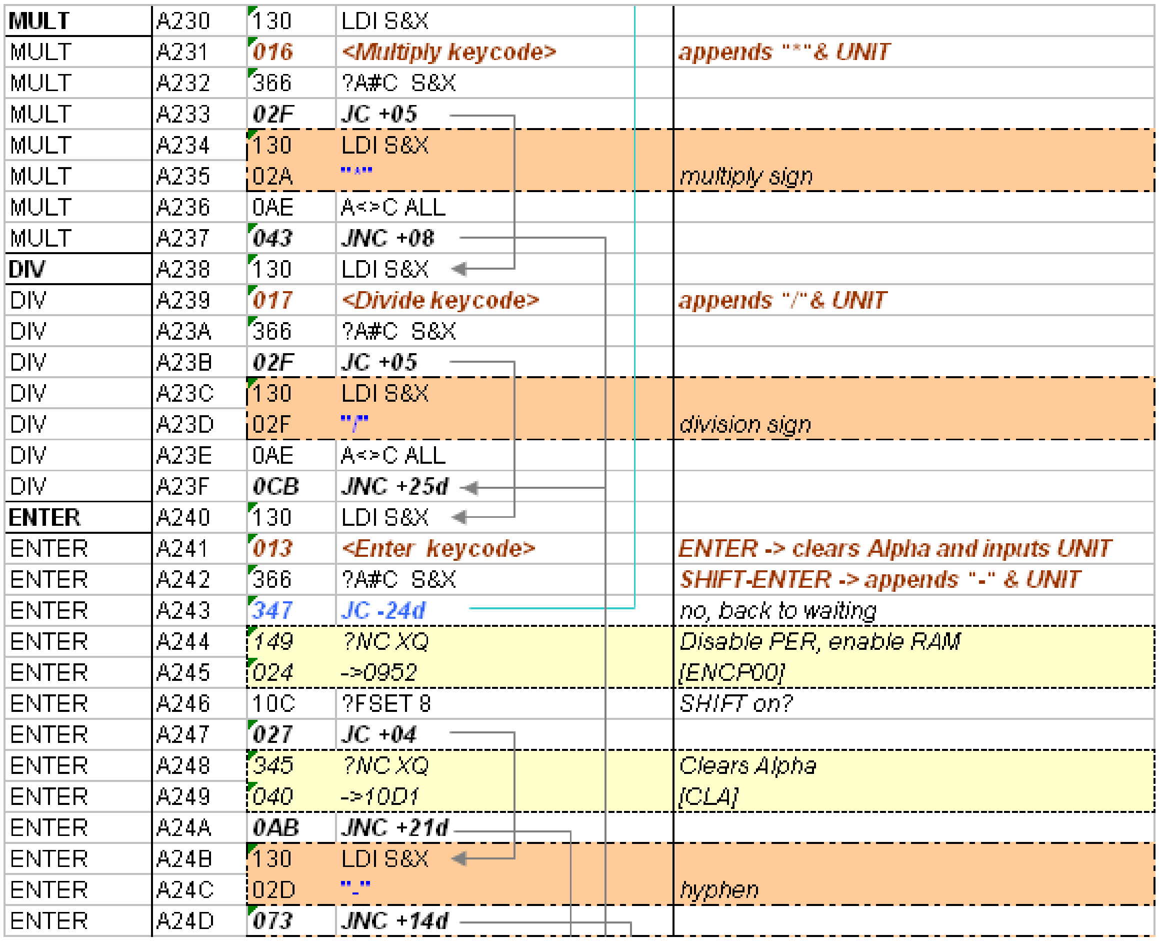


SST	A22C	130	LDI S&X	
SST	A22D	*0C2*	*<SST keycode>*	
SST	A22E	366	?A#C S&X	*SST Key?*
SST	A22F	*253*	*JNC -54d*	*yes->next/prev unit*

So far we've dealt with the navigation keys, it's time to see about the hot keys next.

For the "*" and "/" cases we'll need to append the corresponding character to the Alpha register, followed by the current content of the display – which is holding the unit symbol. This is independent from the status of F8 – the SHIFT indicator.

The ENTER case is either a subset of the above (when SHIFT is not active), or a similar case only appending the "-" character instead (when SHIFT is up). So logically all code to handle the hot-keys will be quire similar and shared, as follows:

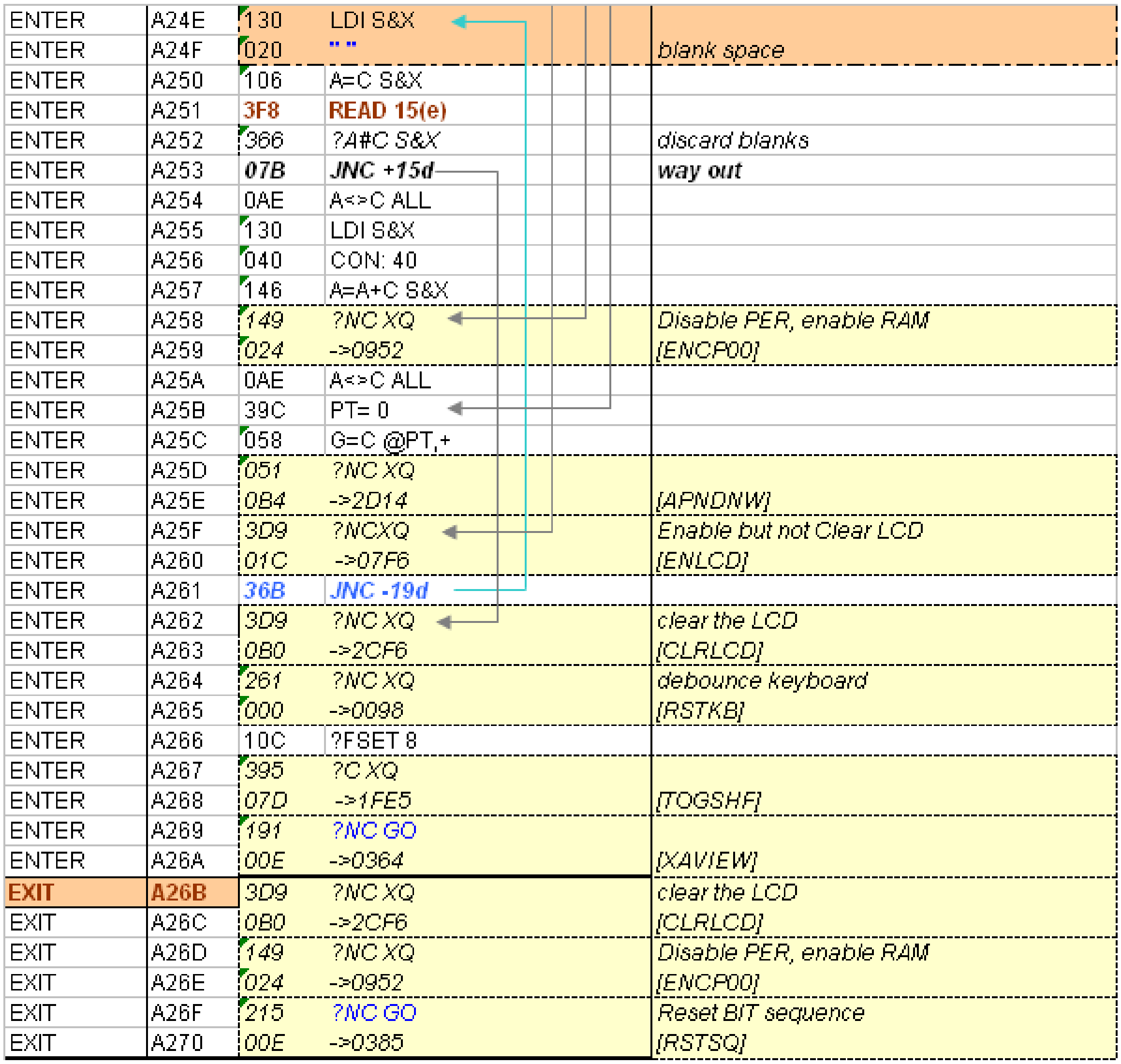


MULT	A230	130	LDI S&X	
MULT	A231	***016***	***<Multiply keycode>***	***appends "*"& UNIT***
MULT	A232	366	?A#C S&X	
MULT	A233	***02F***	***JC +05***	
MULT	A234	130	LDI S&X	
MULT	A235	02A	**"*"**	*multiply sign*
MULT	A236	0AE	A<>C ALL	
MULT	A237	***043***	***JNC +08***	
DIV	A238	130	LDI S&X	
DIV	A239	***017***	***<Divide keycode>***	***appends "/"& UNIT***
DIV	A23A	366	?A#C S&X	
DIV	A23B	***02F***	***JC +05***	
DIV	A23C	130	LDI S&X	
DIV	A23D	02F	**"/"**	*division sign*
DIV	A23E	0AE	A<>C ALL	
DIV	A23F	***0CB***	***JNC +25d***	
ENTER	A240	130	LDI S&X	
ENTER	A241	***013***	***<Enter keycode>***	***ENTER -> clears Alpha and inputs UNIT***
ENTER	A242	366	?A#C S&X	***SHIFT-ENTER -> appends "-" & UNIT***
ENTER	A243	***347***	***JC -24d***	*no, back to waiting*
ENTER	A244	*149*	*?NC XQ*	*Disable PER, enable RAM*
ENTER	A245	*024*	*->0952*	*[ENCP00]*
ENTER	A246	10C	?FSET 8	*SHIFT on?*
ENTER	A247	***027***	***JC +04***	
ENTER	A248	*345*	*?NC XQ*	*Clears Alpha*
ENTER	A249	*040*	*->10D1*	*[CLA]*
ENTER	A24A	***0AB***	***JNC +21d***	
ENTER	A24B	130	LDI S&X	
ENTER	A24C	02D	**"-"**	*hyphen*
ENTER	A24D	***073***	***JNC +14d***	

Notice that we cleared Alpha when unshifted-ENTER is pressed, to begin building the unit string – and that this required enabling chip-0 to access the alpha registers M,N,O, and P.

We're now ready to transfer the display contents to the Alpha register, but remembering that it was left-justified we'll have to ignore all those blank space characters located at the right side of the unit symbol text.

Once all non-blank characters are read from the display and appended into Alpha (which will require enabling & disabling the display for each char) we'll exit the routine but not before tidying things up, like clearing the display and the SHIFT annunciator if it was on, resetting the keyboard status bits using [RSTKB], and prompting the contents of the Alpha register using [XAWIEW] (which would hold a partial or complete unit string by now) to provide the final visual feedback to the user.

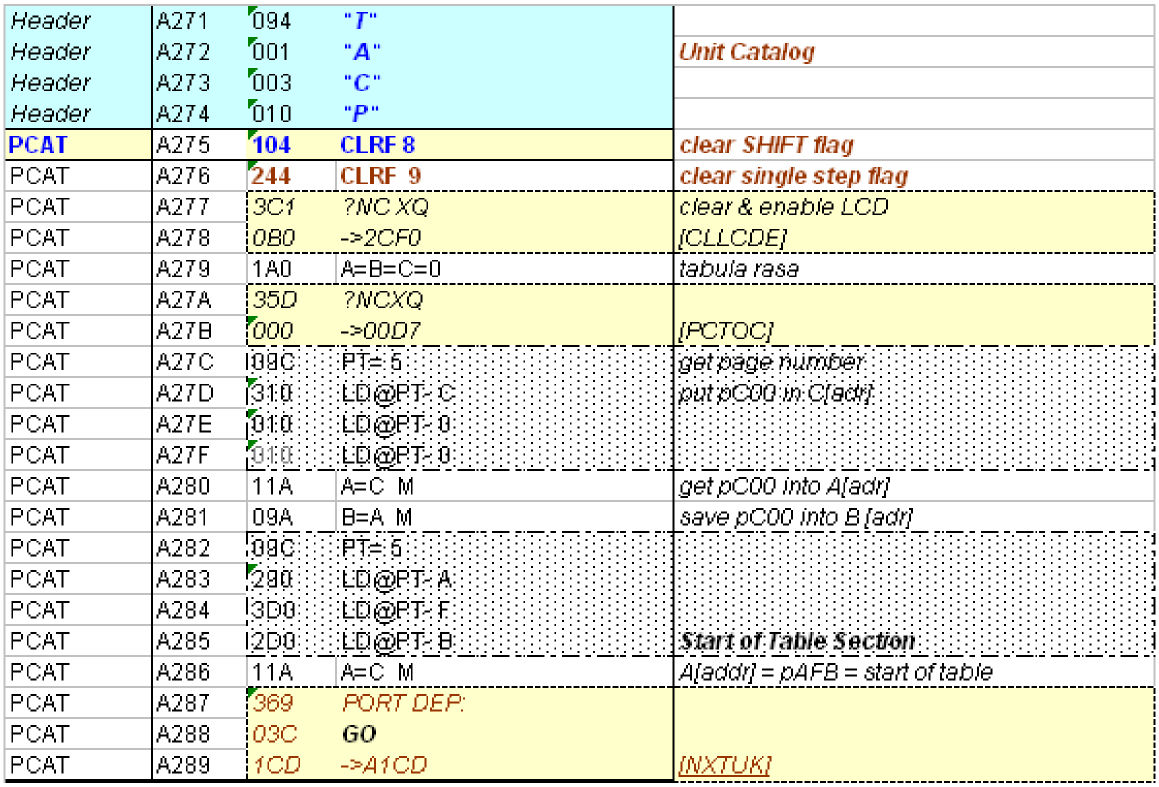


ENTER	A24E	130	LDI S&X	
ENTER	A24F	020	" "	blank space
ENTER	A250	106	A=C S&X	
ENTER	A251	3F8	READ 15(e)	
ENTER	A252	366	?A#C S&X	discard blanks
ENTER	A253	07B	JNC +15d	way out
ENTER	A254	0AE	A<>C ALL	
ENTER	A255	130	LDI S&X	
ENTER	A256	040	CON: 40	
ENTER	A257	146	A=A+C S&X	
ENTER	A258	149	?NC XQ	Disable PER, enable RAM
ENTER	A259	024	->0952	[ENCP00]
ENTER	A25A	0AE	A<>C ALL	
ENTER	A25B	39C	PT= 0	
ENTER	A25C	058	G=C @PT,+	
ENTER	A25D	051	?NC XQ	
ENTER	A25E	0B4	->2D14	[APNDNW]
ENTER	A25F	3D9	?NCXQ	Enable but not Clear LCD
ENTER	A260	01C	->07F6	[ENLCD]
ENTER	A261	36B	JNC -19d	
ENTER	A262	3D9	?NC XQ	clear the LCD
ENTER	A263	0B0	->2CF6	[CLRLCD]
ENTER	A264	261	?NC XQ	debounce keyboard
ENTER	A265	000	->0098	[RSTKB]
ENTER	A266	10C	?FSET 8	
ENTER	A267	395	?C XQ	
ENTER	A268	07D	->1FE5	[TOGSHF]
ENTER	A269	191	?NC GO	
ENTER	A26A	00E	->0364	[XAVIEW]
EXIT	A26B	3D9	?NC XQ	clear the LCD
EXIT	A26C	0B0	->2CF6	[CLRLCD]
EXIT	A26D	149	?NC XQ	Disable PER, enable RAM
EXIT	A26E	024	->0952	[ENCP00]
EXIT	A26F	215	?NC GO	Reset BIT sequence
EXIT	A270	00E	->0385	[RSTSQ]

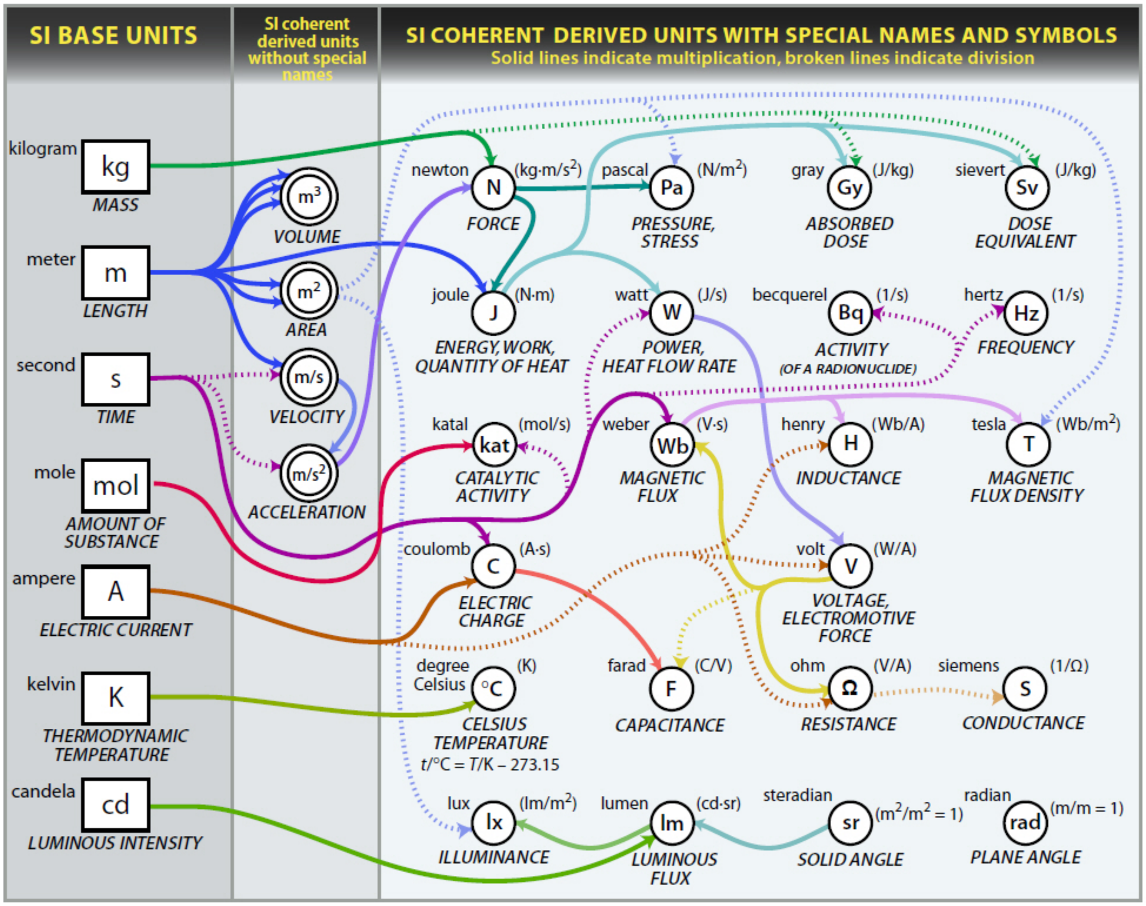
The [EXIT] subroutine is executed when reaching the beginning of the unit table running backwards in run mode. Strangely enough the SHIFT indicator would not clear unless the [RSTSQ] routine is run, so it couldn't be handled by the normal ending of the ENTER case above it (lines A262 to A26A). – which conversely, would not work unless [TOGSHF] is called!

So a bit of mystery– or rather not fully understood behavior - to end things… any reader comments to explain this better would be appreciated ☺

The following MCODE listing corresponds to the PCAT routine, which uses the same main code only pointing at a different Unit Table within the module. There's nothing much to point out or remark, apart from a straight transferring to the appropriate section within UCAT – i.e. good old code reuse!

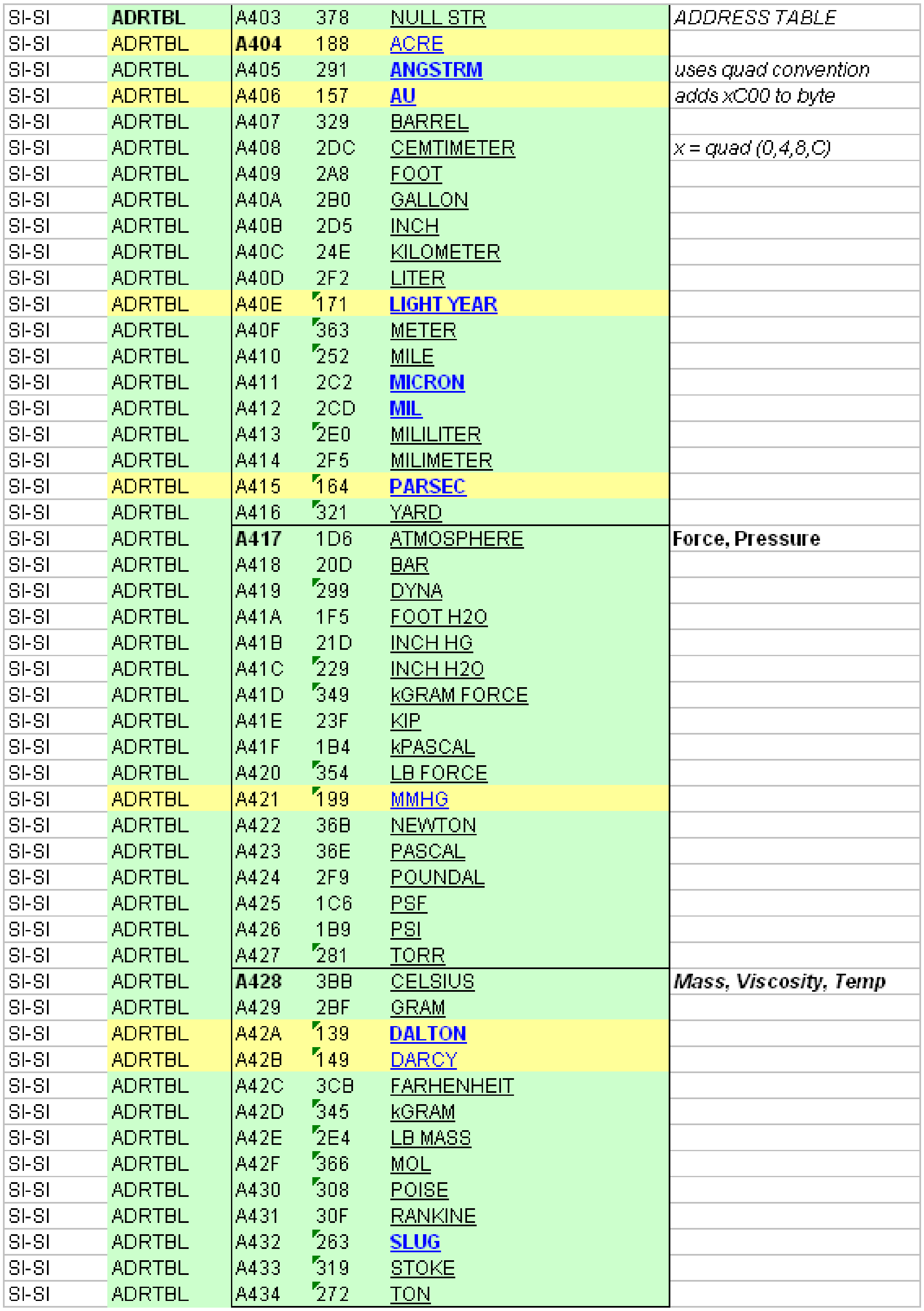


Header	A271	094	"T"	
Header	A272	001	"A"	Unit Catalog
Header	A273	003	"C"	
Header	A274	010	"P"	
PCAT	A275	104	CLRF 8	clear SHIFT flag
PCAT	A276	244	CLRF 9	clear single step flag
PCAT	A277	3C1	?NC XQ	clear & enable LCD
PCAT	A278	0B0	->2CF0	[CLLCDE]
PCAT	A279	1A0	A=B=C=0	tabula rasa
PCAT	A27A	35D	?NCXQ	
PCAT	A27B	000	->00D7	[PCTOC]
PCAT	A27C	09C	PT= 5	get page number
PCAT	A27D	310	LD@PT- C	put pC00 in C[adr]
PCAT	A27E	010	LD@PT- 0	
PCAT	A27F	010	LD@PT- 0	
PCAT	A280	11A	A=C M	get pC00 into A[adr]
PCAT	A281	09A	B=A M	save pC00 into B [adr]
PCAT	A282	09C	PT= 5	
PCAT	A283	290	LD@PT- A	
PCAT	A284	3D0	LD@PT- F	
PCAT	A285	2D0	LD@PT- B	Start of Table Section
PCAT	A286	11A	A=C M	A[addr] = pAFB = start of table
PCAT	A287	369	PORT DEP:	
PCAT	A288	03C	GO	
PCAT	A289	1CD	->A1CD	[NXTUK]

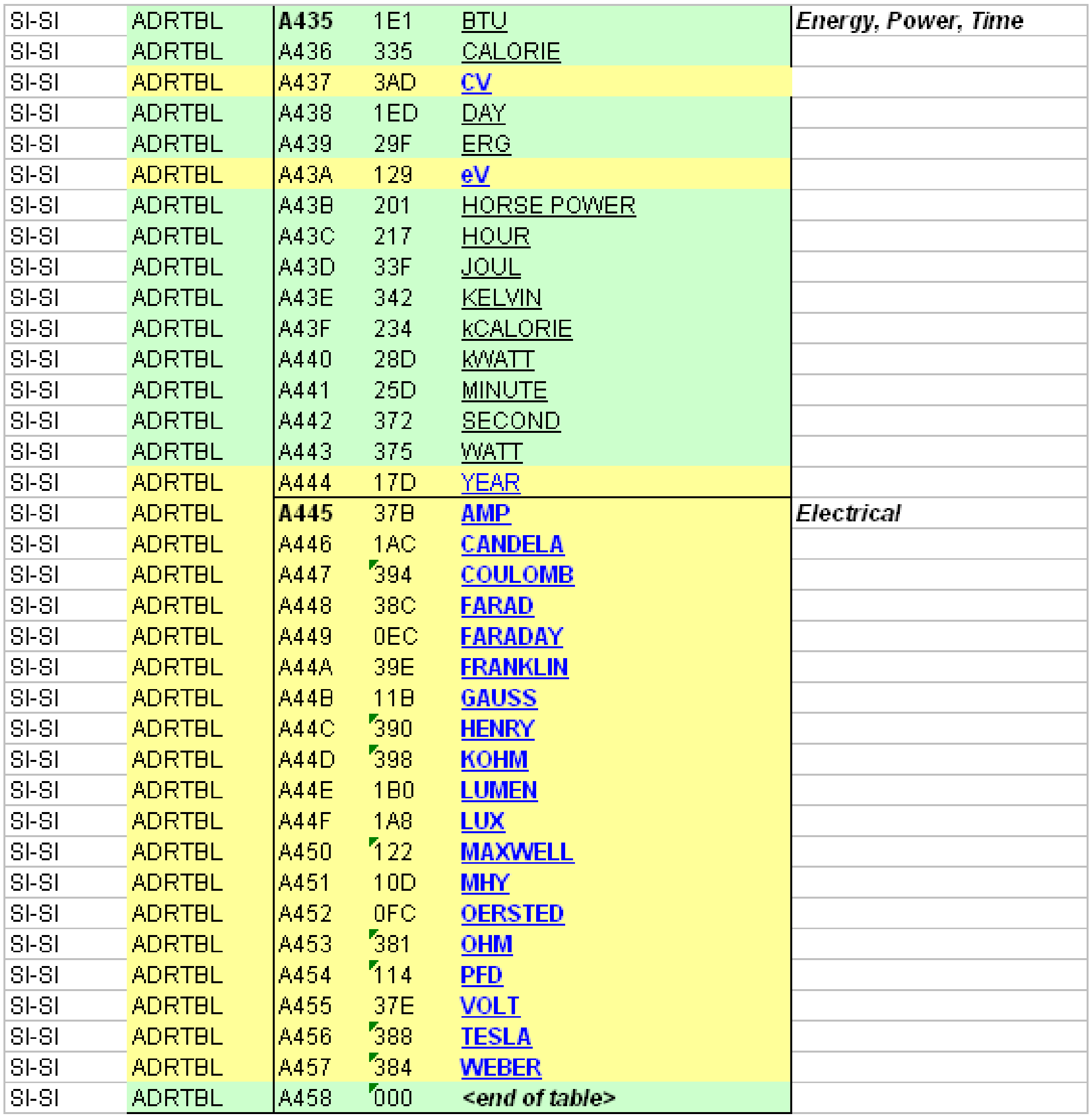


Appendix: The Extended Unit Table.-

Below you can see the new look of the Unit Table, with the units sorted by their magnitude sections. New ones are in blue on yellow background.



SI-SI	**ADRTBL**	A403	378	NULL STR	*ADDRESS TABLE*
SI-SI	ADRTBL	**A404**	188	ACRE	
SI-SI	ADRTBL	A405	291	**ANGSTRM**	*uses quad convention*
SI-SI	ADRTBL	A406	157	**AU**	*adds xC00 to byte*
SI-SI	ADRTBL	A407	329	BARREL	
SI-SI	ADRTBL	A408	2DC	CEMTIMETER	*x = quad (0,4,8,C)*
SI-SI	ADRTBL	A409	2A8	FOOT	
SI-SI	ADRTBL	A40A	2B0	GALLON	
SI-SI	ADRTBL	A40B	2D5	INCH	
SI-SI	ADRTBL	A40C	24E	KILOMETER	
SI-SI	ADRTBL	A40D	2F2	LITER	
SI-SI	ADRTBL	A40E	171	**LIGHT YEAR**	
SI-SI	ADRTBL	A40F	363	METER	
SI-SI	ADRTBL	A410	252	MILE	
SI-SI	ADRTBL	A411	2C2	**MICRON**	
SI-SI	ADRTBL	A412	2CD	**MIL**	
SI-SI	ADRTBL	A413	2E0	MILILITER	
SI-SI	ADRTBL	A414	2F5	MILIMETER	
SI-SI	ADRTBL	A415	164	**PARSEC**	
SI-SI	ADRTBL	A416	321	YARD	
SI-SI	ADRTBL	**A417**	1D6	ATMOSPHERE	**Force, Pressure**
SI-SI	ADRTBL	A418	20D	BAR	
SI-SI	ADRTBL	A419	299	DYNA	
SI-SI	ADRTBL	A41A	1F5	FOOT H2O	
SI-SI	ADRTBL	A41B	21D	INCH HG	
SI-SI	ADRTBL	A41C	229	INCH H2O	
SI-SI	ADRTBL	A41D	349	kGRAM FORCE	
SI-SI	ADRTBL	A41E	23F	KIP	
SI-SI	ADRTBL	A41F	1B4	kPASCAL	
SI-SI	ADRTBL	A420	354	LB FORCE	
SI-SI	ADRTBL	A421	199	MMHG	
SI-SI	ADRTBL	A422	36B	NEWTON	
SI-SI	ADRTBL	A423	36E	PASCAL	
SI-SI	ADRTBL	A424	2F9	POUNDAL	
SI-SI	ADRTBL	A425	1C6	PSF	
SI-SI	ADRTBL	A426	1B9	PSI	
SI-SI	ADRTBL	A427	281	TORR	
SI-SI	ADRTBL	**A428**	3BB	CELSIUS	***Mass, Viscosity, Temp***
SI-SI	ADRTBL	A429	2BF	GRAM	
SI-SI	ADRTBL	A42A	139	**DALTON**	
SI-SI	ADRTBL	A42B	149	DARCY	
SI-SI	ADRTBL	A42C	3CB	FARHENHEIT	
SI-SI	ADRTBL	A42D	345	kGRAM	
SI-SI	ADRTBL	A42E	2E4	LB MASS	
SI-SI	ADRTBL	A42F	366	MOL	
SI-SI	ADRTBL	A430	308	POISE	
SI-SI	ADRTBL	A431	30F	RANKINE	
SI-SI	ADRTBL	A432	263	**SLUG**	
SI-SI	ADRTBL	A433	319	STOKE	
SI-SI	ADRTBL	A434	272	TON	



SI-SI	ADRTBL	**A435**	1E1	BTU	***Energy, Power, Time***
SI-SI	ADRTBL	A436	335	CALORIE	
SI-SI	ADRTBL	A437	3AD	CV	
SI-SI	ADRTBL	A438	1ED	DAY	
SI-SI	ADRTBL	A439	29F	ERG	
SI-SI	ADRTBL	A43A	129	eV	
SI-SI	ADRTBL	A43B	201	HORSE POWER	
SI-SI	ADRTBL	A43C	217	HOUR	
SI-SI	ADRTBL	A43D	33F	JOUL	
SI-SI	ADRTBL	A43E	342	KELVIN	
SI-SI	ADRTBL	A43F	234	kCALORIE	
SI-SI	ADRTBL	A440	28D	kWATT	
SI-SI	ADRTBL	A441	25D	MINUTE	
SI-SI	ADRTBL	A442	372	SECOND	
SI-SI	ADRTBL	A443	375	WATT	
SI-SI	ADRTBL	A444	17D	YEAR	
SI-SI	ADRTBL	**A445**	37B	**AMP**	***Electrical***
SI-SI	ADRTBL	A446	1AC	**CANDELA**	
SI-SI	ADRTBL	A447	394	**COULOMB**	
SI-SI	ADRTBL	A448	38C	**FARAD**	
SI-SI	ADRTBL	A449	0EC	**FARADAY**	
SI-SI	ADRTBL	A44A	39E	**FRANKLIN**	
SI-SI	ADRTBL	A44B	11B	**GAUSS**	
SI-SI	ADRTBL	A44C	390	**HENRY**	
SI-SI	ADRTBL	A44D	398	**KOHM**	
SI-SI	ADRTBL	A44E	1B0	**LUMEN**	
SI-SI	ADRTBL	A44F	1A8	**LUX**	
SI-SI	ADRTBL	A450	122	**MAXWELL**	
SI-SI	ADRTBL	A451	10D	**MHY**	
SI-SI	ADRTBL	A452	0FC	**OERSTED**	
SI-SI	ADRTBL	A453	381	**OHM**	
SI-SI	ADRTBL	A454	114	**PFD**	
SI-SI	ADRTBL	A455	37E	**VOLT**	
SI-SI	ADRTBL	A456	388	**TESLA**	
SI-SI	ADRTBL	A457	384	**WEBER**	
SI-SI	ADRTBL	A458	000	***<end of table>***	

The addresses for the corresponding multiplying factors are actually formed by adding 4000 Hex to the word values from the table. This means all of them must be contained within the second quad (from 4000 to 7FFF).

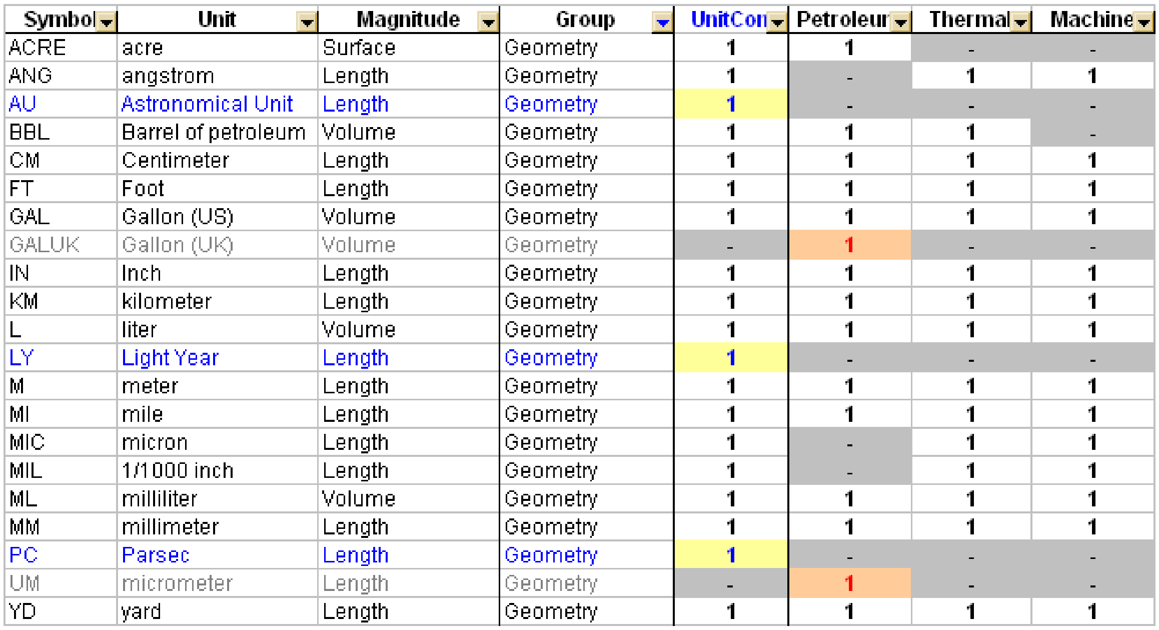
There are about 12 available places to extend the table with more units, but further re-positioning of the main code would be required to ensure that the corresponding multiplying factors are located within Quad 2.

The unit symbol for Mol has been changed to MOL (like it's done on the Petroleum Pac) for obvious consistency reasons. The SLUG and POISE have however been kept the same despite their rather lengthy text.

Appendix.- Module Unit Table Comparisons

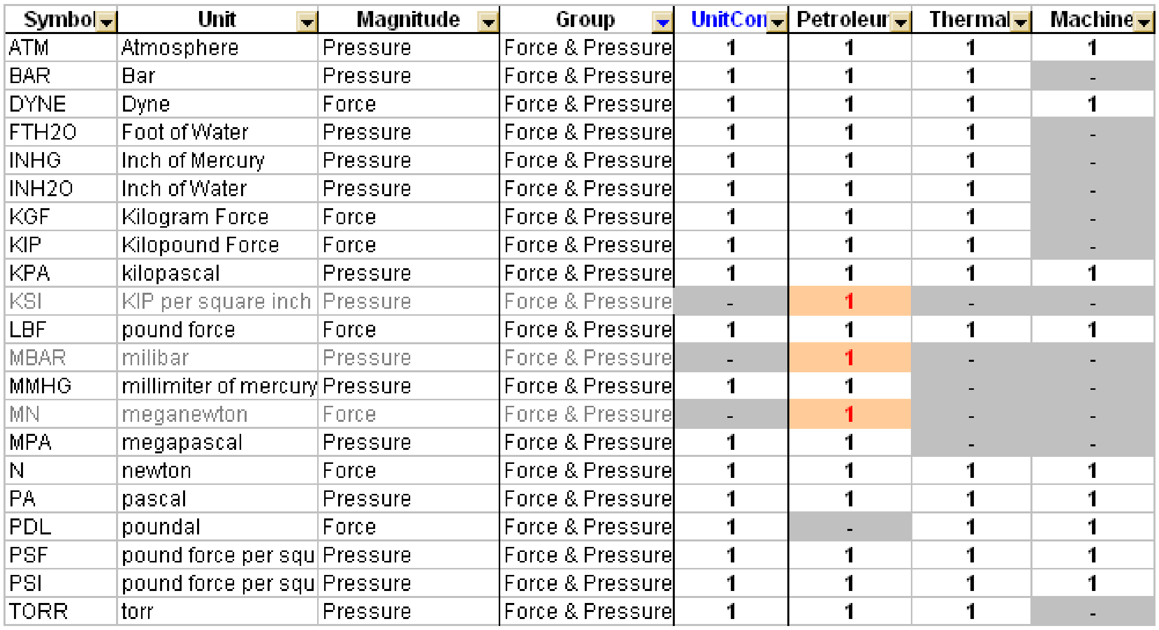
The following five sections correspond to the **UCAT** prompting function argument:

1. Geometry Units.



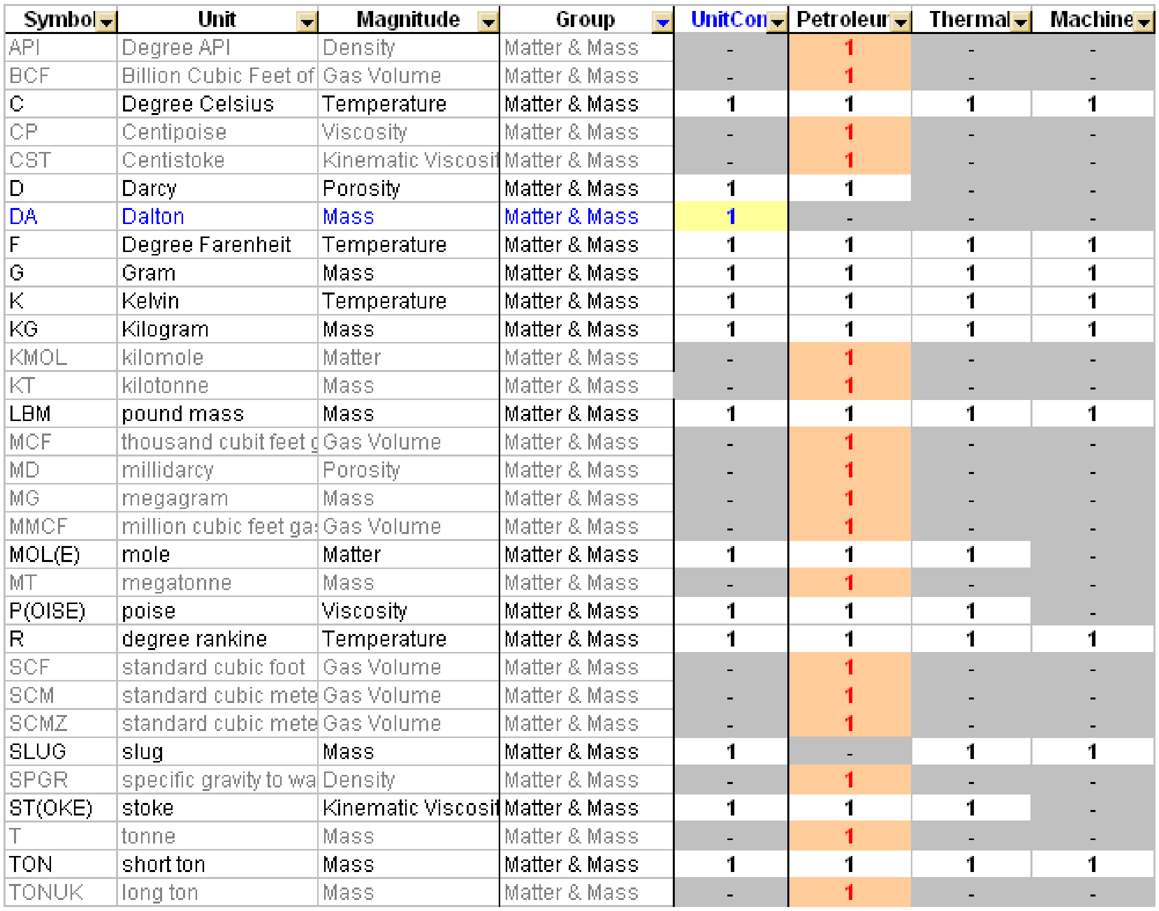
Symbol	Unit	Magnitude	Group	UnitCon	Petroleur	Thermal	Machine
ACRE	acre	Surface	Geometry	1	1	-	-
ANG	angstrom	Length	Geometry	1	-	1	1
AU	Astronomical Unit	Length	Geometry	1	-	-	-
BBL	Barrel of petroleum	Volume	Geometry	1	1	1	-
CM	Centimeter	Length	Geometry	1	1	1	1
FT	Foot	Length	Geometry	1	1	1	1
GAL	Gallon (US)	Volume	Geometry	1	1	1	1
GALUK	Gallon (UK)	Volume	Geometry	-	1	-	-
IN	Inch	Length	Geometry	1	1	1	1
KM	kilometer	Length	Geometry	1	1	1	1
L	liter	Volume	Geometry	1	1	1	1
LY	Light Year	Length	Geometry	1	-	-	-
M	meter	Length	Geometry	1	1	1	1
MI	mile	Length	Geometry	1	1	1	1
MIC	micron	Length	Geometry	1	-	1	1
MIL	1/1000 inch	Length	Geometry	1	-	1	1
ML	milliliter	Volume	Geometry	1	1	1	1
MM	millimeter	Length	Geometry	1	1	1	1
PC	Parsec	Length	Geometry	1	-	-	-
UM	micrometer	Length	Geometry	-	1	-	-
YD	yard	Length	Geometry	1	1	1	1

2. Force and Pressure Units`



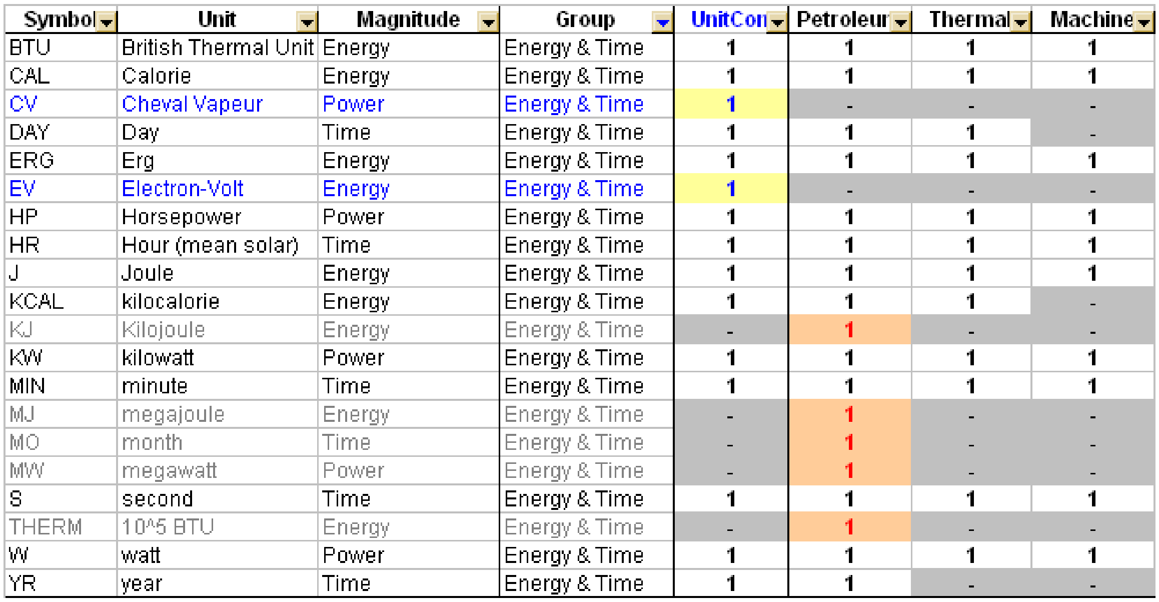
Symbol	Unit	Magnitude	Group	UnitCon	Petroleur	Thermal	Machine
ATM	Atmosphere	Pressure	Force & Pressure	1	1	1	1
BAR	Bar	Pressure	Force & Pressure	1	1	1	-
DYNE	Dyne	Force	Force & Pressure	1	1	1	1
FTH2O	Foot of Water	Pressure	Force & Pressure	1	1	1	-
INHG	Inch of Mercury	Pressure	Force & Pressure	1	1	1	-
INH2O	Inch of Water	Pressure	Force & Pressure	1	1	1	-
KGF	Kilogram Force	Force	Force & Pressure	1	1	1	-
KIP	Kilopound Force	Force	Force & Pressure	1	1	1	-
KPA	kilopascal	Pressure	Force & Pressure	1	1	1	1
KSI	KIP per square inch	Pressure	Force & Pressure	-	1	-	-
LBF	pound force	Force	Force & Pressure	1	1	1	1
MBAR	milibar	Pressure	Force & Pressure	-	1	-	-
MMHG	millimiter of mercury	Pressure	Force & Pressure	1	1	-	-
MN	meganewton	Force	Force & Pressure	-	1	-	-
MPA	megapascal	Pressure	Force & Pressure	1	1	-	-
N	newton	Force	Force & Pressure	1	1	1	1
PA	pascal	Pressure	Force & Pressure	1	1	1	1
PDL	poundal	Force	Force & Pressure	1	-	1	1
PSF	pound force per squ	Pressure	Force & Pressure	1	1	1	1
PSI	pound force per squ	Pressure	Force & Pressure	1	1	1	1
TORR	torr	Pressure	Force & Pressure	1	1	1	-

3. Matter & Mass Units.



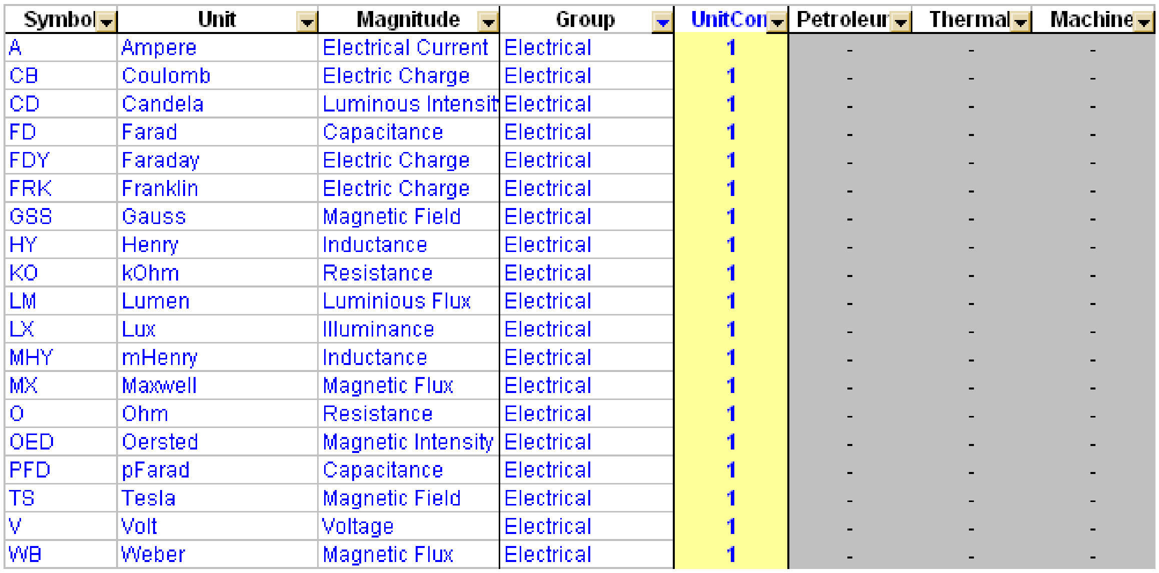
Symbol	Unit	Magnitude	Group	UnitCon	Petroleur	Thermal	Machine
API	Degree API	Density	Matter & Mass	-	1	-	-
BCF	Billion Cubic Feet of	Gas Volume	Matter & Mass	-	1	-	-
C	Degree Celsius	Temperature	Matter & Mass	1	1	1	1
CP	Centipoise	Viscosity	Matter & Mass	-	1	-	-
CST	Centistoke	Kinematic Viscosit	Matter & Mass	-	1	-	-
D	Darcy	Porosity	Matter & Mass	1	1	-	-
DA	Dalton	Mass	Matter & Mass	1	-	-	-
F	Degree Farenheit	Temperature	Matter & Mass	1	1	1	1
G	Gram	Mass	Matter & Mass	1	1	1	1
K	Kelvin	Temperature	Matter & Mass	1	1	1	1
KG	Kilogram	Mass	Matter & Mass	1	1	1	1
KMOL	kilomole	Matter	Matter & Mass	-	1	-	-
KT	kilotonne	Mass	Matter & Mass	-	1	-	-
LBM	pound mass	Mass	Matter & Mass	1	1	1	1
MCF	thousand cubit feet g	Gas Volume	Matter & Mass	-	1	-	-
MD	millidarcy	Porosity	Matter & Mass	-	1	-	-
MG	megagram	Mass	Matter & Mass	-	1	-	-
MMCF	million cubic feet ga	Gas Volume	Matter & Mass	-	1	-	-
MOL(E)	mole	Matter	Matter & Mass	1	1	1	-
MT	megatonne	Mass	Matter & Mass	-	1	-	-
P(OISE)	poise	Viscosity	Matter & Mass	1	1	1	-
R	degree rankine	Temperature	Matter & Mass	1	1	1	1
SCF	standard cubic foot	Gas Volume	Matter & Mass	-	1	-	-
SCM	standard cubic mete	Gas Volume	Matter & Mass	-	1	-	-
SCMZ	standard cubic mete	Gas Volume	Matter & Mass	-	1	-	-
SLUG	slug	Mass	Matter & Mass	1	-	1	1
SPGR	specific gravity to wa	Density	Matter & Mass	-	1	-	-
ST(OKE)	stoke	Kinematic Viscosit	Matter & Mass	1	1	1	-
T	tonne	Mass	Matter & Mass	-	1	-	-
TON	short ton	Mass	Matter & Mass	1	1	1	1
TONUK	long ton	Mass	Matter & Mass	-	1	-	-

4. Energy, Power & Time Units



Symbol	Unit	Magnitude	Group	UnitCon	Petroleur	Thermal	Machine
BTU	British Thermal Unit	Energy	Energy & Time	1	1	1	1
CAL	Calorie	Energy	Energy & Time	1	1	1	1
CV	Cheval Vapeur	Power	Energy & Time	1	-	-	-
DAY	Day	Time	Energy & Time	1	1	1	-
ERG	Erg	Energy	Energy & Time	1	1	1	1
EV	Electron-Volt	Energy	Energy & Time	1	-	-	-
HP	Horsepower	Power	Energy & Time	1	1	1	1
HR	Hour (mean solar)	Time	Energy & Time	1	1	1	1
J	Joule	Energy	Energy & Time	1	1	1	1
KCAL	kilocalorie	Energy	Energy & Time	1	1	1	-
KJ	Kilojoule	Energy	Energy & Time	-	1	-	-
KW	kilowatt	Power	Energy & Time	1	1	1	1
MIN	minute	Time	Energy & Time	1	1	1	1
MJ	megajoule	Energy	Energy & Time	-	1	-	-
MO	month	Time	Energy & Time	-	1	-	-
MW	megawatt	Power	Energy & Time	-	1	-	-
S	second	Time	Energy & Time	1	1	1	1
THERM	10^5 BTU	Energy	Energy & Time	-	1	-	-
W	watt	Power	Energy & Time	1	1	1	1
YR	year	Time	Energy & Time	1	1	-	-

5. Electrical & Luminance Units



Symbol	Unit	Magnitude	Group	UnitCon	Petroleur	Thermal	Machine
A	Ampere	Electrical Current	Electrical	1	-	-	-
CB	Coulomb	Electric Charge	Electrical	1	-	-	-
CD	Candela	Luminous Intensit	Electrical	1	-	-	-
FD	Farad	Capacitance	Electrical	1	-	-	-
FDY	Faraday	Electric Charge	Electrical	1	-	-	-
FRK	Franklin	Electric Charge	Electrical	1	-	-	-
GSS	Gauss	Magnetic Field	Electrical	1	-	-	-
HY	Henry	Inductance	Electrical	1	-	-	-
KO	kOhm	Resistance	Electrical	1	-	-	-
LM	Lumen	Luminious Flux	Electrical	1	-	-	-
LX	Lux	Illuminance	Electrical	1	-	-	-
MHY	mHenry	Inductance	Electrical	1	-	-	-
MX	Maxwell	Magnetic Flux	Electrical	1	-	-	-
O	Ohm	Resistance	Electrical	1	-	-	-
OED	Oersted	Magnetic Intensity	Electrical	1	-	-	-
PFD	pFarad	Capacitance	Electrical	1	-	-	-
TS	Tesla	Magnetic Field	Electrical	1	-	-	-
V	Volt	Voltage	Electrical	1	-	-	-
WB	Weber	Magnetic Flux	Electrical	1	-	-	-

Conclusion.

The Unit Conversion Module offers a superset of units and magnitudes when compared to either the Thermal Transport or the Machine Construction Pacs. It does not, however, have all units found in the Petroleum fluids pac – which were too specific to the trade or multiples of other existing units.

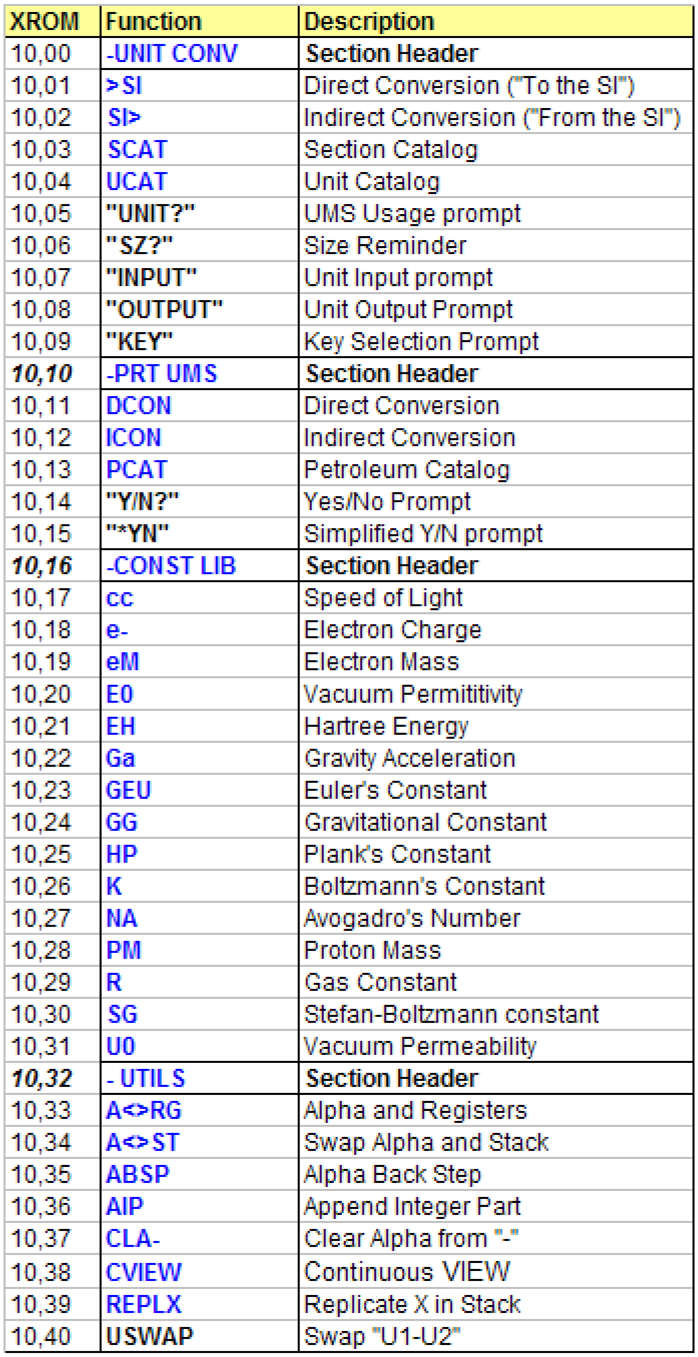
Those units unique to the Unit Conversion module have yellow background, whilst nits unique to the Petroleum Pac have orange background.

So life is about choices and here too you'd need to choose which unit table to use, as follows:

- Use **SI>** and **>SI** for the new extended Unit Table,
- Use **DCON** and **ICON** for the Petroleum Unit Table.

Both are included in the Unit Conversion Module, so the choice should be an easy one, and without any trade-offs to make.

Unit Conversion Module – Function Table



XROM	Function	Description
10,00	-UNIT CONV	**Section Header**
10,01	>SI	Direct Conversion ("To the SI")
10,02	SI>	Indirect Conversion ("From the SI")
10,03	SCAT	Section Catalog
10,04	UCAT	Unit Catalog
10,05	**"UNIT?"**	UMS Usage prompt
10,06	**"SZ?"**	Size Reminder
10,07	**"INPUT"**	Unit Input prompt
10,08	**"OUTPUT"**	Unit Output Prompt
10,09	**"KEY"**	Key Selection Prompt
10,10	-PRT UMS	**Section Header**
10,11	DCON	Direct Conversion
10,12	ICON	Indirect Conversion
10,13	PCAT	Petroleum Catalog
10,14	**"Y/N?"**	Yes/No Prompt
10,15	**"*YN"**	Simplified Y/N prompt
10,16	-CONST LIB	**Section Header**
10,17	cc	Speed of Light
10,18	e-	Electron Charge
10,19	eM	Electron Mass
10,20	E0	Vacuum Permititivity
10,21	EH	Hartree Energy
10,22	Ga	Gravity Acceleration
10,23	GEU	Euler's Constant
10,24	GG	Gravitational Constant
10,25	HP	Plank's Constant
10,26	K	Boltzmann's Constant
10,27	NA	Avogadro's Number
10,28	PM	Proton Mass
10,29	R	Gas Constant
10,30	SG	Stefan-Boltzmann constant
10,31	U0	Vacuum Permeability
10,32	- UTILS	**Section Header**
10,33	A<>RG	Alpha and Registers
10,34	A<>ST	Swap Alpha and Stack
10,35	ABSP	Alpha Back Step
10,36	AIP	Append Integer Part
10,37	CLA-	Clear Alpha from "-"
10,38	CVIEW	Continuous VIEW
10,39	REPLX	Replicate X in Stack
10,40	**USWAP**	Swap "U1-U2"