

ES RSU 2A

**OPERATING
SYSTEM**

ERAMCO SYSTEMS
RSU-2A Module
Quick Reference Guide

In this quick reference guide for the ES RSU-2A operating system the input for, and result of, each function is shown in a similar format:

FN (Function Name derivation)
 Input : data, format, etc.
 Output or function results, etc.

ACHR (Append ChArActer)
 Alpha : Character string
The characters in Alpha are appended to the end of the current record.
The character pointer is set to the end of the record.

AFLD (Append FieLD)
 Alpha : Character string
The characters in Alpha are appended to the end of the current record as a new field.
The character pointer is set to the end of the record.

ALEN? (Alpha LENgth ?)
 Alpha : Character string
The length of the alpha register is returned to the X-register.
If the stack lift is enabled the stack is raised and the T-register lost, if not the X-register will be overwritten.

AREC (Append a RECord)

Alpha : Character string

The characters in Alpha are appended to the end of the current file as a new record.

The record pointer is set to this record and the character pointer is set to the end of the record.

ASRM? (AScii RooM ?)

No user input required

This function returns the number of free bytes in the current ASCII file.

If the stack lift is enabled the stack is raised and the T-register lost, if not the X-register will be overwritten.

A=F? (Alpha = Field ?)

Alpha : Compare string

X-reg : Field to compare

The string in Alpha and the field given in the X-register are compared. If the X-register is zero Alpha is compared with the characters starting at the current character pointer.

If the compared field and Alpha are equal, and a program is running the next step is executed, otherwise it is skipped ("do if true").

If executed from the keyboard and the result is true "YES" is displayed, otherwise "NO".

A<>F? (Alpha <> Field ?)

Alpha : Compare string

X-reg : Field to compare

The string in Alpha and the field specified in the X-register are compared for inequality. If the X-register is zero Alpha is compared with the characters starting at the current character pointer.

See A=F? for an explanation of the result.

A<F? (Alpha < Field ?)

Alpha : Compare string

X-reg : Field to compare

The string in Alpha is tested for being less than the field that is specified in the X-register. If the X-register is zero Alpha is compared with the characters starting at the current character pointer.

See A=F? for an explanation of the result.

A<=F? (Alpha <= Field ?)

Alpha : Compare string

X-reg : Field to compare

The string in Alpha is tested for being less than or equal to the field that is specified in the X-register. If the X-register is zero Alpha is compared with the characters starting at the current character pointer.

See A=F? for an explanation of the result.

A>F? (Alpha > Field ?)

Alpha : Compare string

X-reg : Field to compare

A test is performed on Alpha and the specified field to see if Alpha is greater than the field. If the X-register is zero Alpha is compared with the characters starting at the current character pointer.

See A=F? for an explanation of the result.

A>=F? (Alpha >= Field ?)

Alpha : Compare string

X-reg : Field to compare

A test is performed on Alpha and the specified field to see if Alpha is greater than or equal to the field. If the X-register is zero Alpha is compared with the characters starting at the current character pointer.

See A=F? for an explanation of the result.

CLAF (Clear AsciiFile)

Alpha : Filename

All records in the named file are cleared. The named file becomes the "working" ASCII file.

CRAF (Create Ascii File)

Alpha : Filename

X-reg : Filelength in
registers

An ASCII file of the desired length and name is created in an RSU data block. The newly created file becomes the working ASCII file.

DCHPT (Decrement Char Pointer)

No user input required

The character pointer is decremented by one. If already positioned at the first character it remains the same.

DCHR (Delte CharActers)

X-reg : Number of characters to delete

Starting from the current character the specified number of characters is deleted.

If more characters are specified than are left in the record, all characters up to the end of the record are deleted.

DFLD (Delte FieLD)

X-reg : Field to delete

The field specified in the X-register is deleted.

The character pointer is set to the first character of the following field.

If the specified field is the last, the character pointer is set to the end of the record.

DRCPT (Decrement ReCord Pointer)

No user input required

The record pointer is decremented by one.

The character pointer is not moved unless the newly selected record is shorter than the character pointer indicates, in which case the character pointer is set to the end of the record.

DREC (Delete RECORD)

No user input required

The current record is deleted. If the deleted record is the last, the record pointer is set to the preceeding record. In all other cases the record pointer is not changed.

The character pointer is always set to the first character in the newly selected record.

EDT (EDiTOR)

Alpha : Filename

This function activates the text editor.

If a working file has been specified then it will be edited without reference to Alpha. If there is no working file then the file specified in Alpha is edited. In edit mode three redefined keyboards are available to enable easy input of all HP-41 and ASCII characters.

On all three keyboards certain keys have common functions :

- <1** : This key moves the cursor one place to the left.
- 1>** : This key moves the cursor one place to the right.
- <11** : This key moves the cursor 11 places to the left.
- 11>** : This key moves the cursor 11 places to the right.

- REC↓** : The cursor is advanced one record. The cursor stays at the same character position if possible, but if the newly selected record is not long enough then the cursor is set to the end of that record.
- REC↑** : The cursor is set to the previous record. See SST for behavior of the cursor character position. If the cursor is set to record 0, the file name is displayed.
- <--** : The character to the left of the cursor is deleted.

All the preceeding functions autorepeat.

- REC** : This key deletes the current record.
- GTO** : Three prompt digits appear. Enter a valid record number to place the cursor at the first character of that record. Altenatively, to place the cursor at a selected character in the current record, press the decimal key first. If .000 is entered the cursor is set to the last character of the current record.

- INREP** : This key toggles between the replace and insert modes.
When insert mode is active the 1 annunciator is set.
- SPAS** : This key splits a record into two separate records if the cursor is within the record, but if the cursor is at the end of the record then the current record and the following record are joined together.
- +REC** : Pressing this key inserts a record after the current record.
- VPOS** : This key displays the current record and character position. Restore normal working by pressing <-- once.
- MSP** : This key toggles the normal and special keyboards. When the ALPHA annunciator is on the uppercase keyboard is active, else one of the two special keyboards is active.
- TKBD** : This key toggles special keyboards one and two. When the 0 annunciator is on the lowercase keyboard is active, else the printer keyboard is active.
- EXIT** : This key terminates an edit session.

Diagrams of the three keyboards follow. As on the HP-41 itself, non shifted characters or functions are shown on the keys, and shifted characters or functions above the keys.

If a character is displayed by the HP-41 as a boxed star then its ASCII code is substituted in the diagram, except for lower case alpha f thru z. Although these also display as boxed stars the diagrams show the actual characters.

With some of the newer models of the HP-41 lowercase characters are also displayed correctly. User flag 20 will be set when the calculator is of the new type.

On the printer keyboard characters that are recognized by the HP-41 printer show as the characters.

Uppercase keyboard

Alpha on

	"11	11"	TKBD
EXIT	<1	1>	MSP

a	b	c	d	e
A	B	C	D	E
Σ	%	\neq	<	>
F	G	H	I	J
	SPAS	INREP	GTO	REC A
	K	L	M	REC Y
¹		Δ	\$	=REC
N		O	P	←
-	7	8	9	
Q	R	S	T	
+	4	5	6	
U	V	W	X	
*	1	2	3	
Y	Z	=	?	
/	0	⁷	VPOS	
:		.	+REC	

Lowercase keyboard

Alpha off, flag 0 off

EXIT	"11 <1	11"> 1>	TKBD MSP
------	-----------	------------	-------------

Δ 17	λ 30	τ 19	\angle 124	Γ 21
a	b	c	d	e
Γ 23	$\{$ 123	$\}$ 125	$\{$	$\}$
f	g	h	i	j
	SPAS	INREP	GTO	REC λ
	k	l	m	REC γ
\dagger		\prime	$\prime\prime$	\neg REC
n	o	p	\leftarrow	
\sharp	\dagger	Γ	\sqsupset	
q	r	s	t	
$\&$	\square	μ	τ	
u	v	w	x	
$\bar{\lambda}$	$\bar{\lambda}$	$\bar{\lambda}$	$\bar{\lambda}$	
y	z	=	?	
\backslash 92	π 16	γ	VPOS	
:		.	+REC	

Printer keyboard

Alpha off, flag 0 on

EXIT	«11 1	11» 1	TKBD MSP
-------------	----------	----------	---------------------------

å	ä	ö	ü	œ
A	B	C	D	E
À	Ä	Ö	Ü	Æ
F	≠	£	⋮	σ
	SPAS	INREP	GTO	REC A
	CR	FF	LF	REC Y
ESC		BEL	BS	-REC
†		-	E	←
α	β	λ	τ	
-	7	8	9	
Γ	δ	Θ	Φ	
+	4	5	6	
μ	Ω	χ	̄	
*	1	2	3	
←	→	⋮	VPOS	
/	0	⋮	+REC	

FINDPS (**FIND** **Pos**ition)

Alpha : Target string

X-reg : Field to search for

Y-reg : Record step size

The working ASCII file is searched for an occurrence of the string in Alpha. When the target string is found the record pointer is set to that record.

When a program is running the next program step is executed if the target string is found, otherwise not ("do if true").

If the function is executed from the keyboard the message "YES" will be displayed if the target string is found, otherwise "NO".

The search is controlled by the X and Y-registers.

The X-register determines the starting position in each record to be compared.

If the number specified in the X-register is zero, a search will be made in each record from the position of the character pointer when FINDPS is executed.

If the X-register contains a non-zero value the search is performed starting at the first character of that field.

The sign of the X-register determines search behavior when a record has fewer fields or characters than specified. If the X-register is negative the search is continued when such a record is encountered, whereas if it is positive the search is stopped and the number of the record concerned returned to the X-register, with the record pointers left unaltered.

The Y-register controls the record step-size and direction of search. Negative numbers specify a backwards search and positive numbers a forwards search.

FLEN? (Field LEngth ?)

X-reg : Field specifier

The length of the field specified in the X-register is returned to the X-register.

If the stack lift is enabled the stack is raised and the T-register lost, if not the X-register is overwritten.

GCHR (Get CHaRacters)

X-reg : Number of characters

This function clears Alpha and transfers the specified number of characters to it, starting at the current character and ending after the specified number of characters has been transferred, the end of the record is reached, or Alpha is filled.

If the end of the record is reached during transfer then user flag 17 is cleared, else it is set.

The character pointer is set to the first character after the last transferred character. If the end of the record is reached during the transfer the pointers are set to the first character of the next record.

GFLD (Get FieLD)

X-reg : Field to read

This function clears Alpha and transfers to it the field specified in the X-register.

Transfer starts at the first character of the field and ends when the end of the field is reached or Alpha is filled.

If the end of the field is reached during transfer then user flag 17 is cleared, else it is set.

For a description of pointer behavior see GCHR.

GREC (Get RECord)

No user input required

This function clears Alpha and transfers the current record to it. Transfer starts from the current character pointer and ends when Alpha is filled or the end of the record is reached.

If the end of the record is reached during transfer then user flag 17 is cleared, else it is set. For a description of pointer behavior see GCHR.

ICHPT (Increment CHar PoinTer)

No user input required

The character pointer is incremented by one.

ICHR (Insert CHaRacters)

Alpha : String to insert

This function inserts the contents of Alpha to the left of the cursor. The cursor remains unchanged.

IFLD (Insert FieLD)

Alpha : String to insert

X-reg : Field number

The string in Alpha is inserted as a new field to the left of the field specified in the X-register. The character pointer is set after the delimiter of the just inserted field.

IRCPT (Increment ReCord PoinTer)

No user input required

The record pointer is advanced to the next record.

The character pointer is not moved unless the newly selected record is shorter than the position it points to, in which case it is set to the end of the record.

IREC (Insert RECord)

Alpha : Character string

The characters in Alpha are inserted as a new record before the current record.

The record pointer is set to this record with the character pointer set to the end of the record.

MCHPT (Move CHaracter PoinTer)

X-reg : Step size

The character pointer is moved by the number of characters specified in the X-register.

The sign of the X-register determines whether the pointer is decremented or incremented. Negative values decrement the pointer and positive values increment it.

If an attempt is made to position the pointer before the first character it is set to the first character.

MRCPT (Move Record Pointer)

X-reg : Step size

The record pointer is moved by the number of records specified in the X-register.

The sign of the X-register determines whether the pointer is decremented or incremented. Negative values decrement the pointer and positive values increment it.

The character pointer is not moved unless the new record is shorter than the position it indicates in which case the character pointer is set to the end of the record.

NAMAF (NAME AsciiFile)

No user input required

The name of the working ASCII file is returned to Alpha.

NAMDF (NAME DataFile)

No user input required

The name of the working datafile is returned to Alpha.

NFLD? (Number of FieLDs ?)

No user input required

This function returns the number of fields in the current record.

If the stack lift is enabled the stack is raised and the T-register lost, if not the X-register is overwritten.

NREC? (Number of RECs ?)

No user input required

This function returns the number of records in the working ASCII file. For remarks on stacklift see NFLD?

NUMREC (get NUMber in RECord)

No user input required

The current record is searched for a number. The search starts at the current character and continues to the end of the record.

The character pointer is set to the first character after the number.

If the stack lift is enabled the stack is raised and the T-register lost, if not the X-register is overwritten.

A number starts with either a digit or a minus sign. The mantissa is up to 10 digits long. If there are more than 10 digits in the mantissa the surplus digits are ignored. A period or comma following the mantissa is treated according to the setting of flag 28: if flag 28 is set a decimal point separates the integer and fractional parts of the number; if flag 28 is clear a comma separates the integer and fractional parts.

After the mantissa an exponent may begin with an "E", followed by an optional minus sign and one or two digits. If there are more than 2 digits in an exponent the surplus digits are ignored.

A number is terminated whenever a character is encountered that is no more part of the number.

If the end of the record is reached during search then user flag 17 is cleared, else it is set.

OUTCHR (OUTput CHaracters)

X-reg : Number of characters

Y-reg : Column width

Flag 17 : Terminate with CR/LF

This function sends the specified number of characters to the selected device on the HP-IL loop. Transmission starts with the current character and ends when the specified number of characters has been sent or the end of the record is reached.

To facilitate columnar formatting the Y-register controls the length of the output string. If this length is greater than the number of characters to be sent the output will be padded with blanks: Trailing blanks if the Y-register is negative and leading blanks if it is positive.

If flag 17 is clear, the sequence is terminated with a CR/LF, else not.

The character pointer is set after the last character sent. If the end of the record is reached before the specified number of characters are sent, the record pointers are set to the first character of the next record.

OUTFLD (OUTput FieLD)

X-reg : Field to send

Y-reg : Column width

Flag 17 : Terminate with CR/LF

The field specified in the X-register is sent to the selected device on the HP-IL loop.

Transmission starts with the first character of the field and ends when the next field or the end of the record is reached.

For a description of the use of the Y-register and flag 17 see OUTCHR.

The character pointer is set to the first character of the next field if possible, else the pointers are set to the first character of the next record.

OUTREC (OUTput RECord)

Y-reg : Column width

Flag 17 : Terminate with CR/LF

The current record is sent to the selected device on the HP-IL loop.

Transmission starts at the current position of the character pointer and ends when the end of the record is reached.

For a description of the use of the Y-register and flag 17 see OUTCHR.

The pointers are set to the first character in the next record.

RCHPT (Recall CHaracter Pointer)

No user input required

The value of the character pointer is recalled to the X-register.

If the stack lift is enabled the stack is raised and the T-register lost, if not the X-register is overwritten.

RCHR (Recall CHaracters)

X-reg : Number of characters

This function appends the specified number of characters to Alpha.

The function stops when the specified number of characters is appended to Alpha, the end of the record is reached, or Alpha is full.

If the end of the record is reached during transfer then user flag 17 is cleared, else it is set. The character pointer is set after the last transferred character. If the end of the record is reached during transfer the pointers are set to the first character of the next record.

RESAF (REsize AsciiFile)

Alpha : Filename

X-reg : New filesize

The named ASCII file will be resized as specified by the X-register.

When resized, data storage blocks are packed to leave as much space as possible.

The "working file" becomes unspecified.

RFLD (Recall FieLD)

X-reg : Field to append

Flag 17 : Continue at pointer

The field specified in the X-register is appended to Alpha. Transfer starts at the first character of the field except when flag 17 is set in which case it starts at the current character pointer position.

Transfer stops when the whole field is read back, the end of the record is reached, or Alpha is filled.

If the end of the field is reached during transfer then user flag 17 is cleared, else it is set.

The character pointer is set after the last transferred character. If the end of the record is reached during transfer the pointers are set to the first character of the next record.

RLEN? (Record LENgth ?)

No user input required

The length of the current record is recalled to the X-register.

For remarks on stacklift see RCHPT.

RRCPT (Recall ReCord PoinTer)

No user input required

The value of the record pointer is recalled to the X-register.

For remarks on stacklift see RCHPT.

RREC (Recall RECord)

No user input required

This function appends the contents of the current record to Alpha.

Transfer starts at the position of the character pointer. Transfer is stopped when either the end of the record is reached or Alpha is filled. If the end of the record is reached during transfer then user flag 17 is cleared, else it is set. The character pointer is set after the last character transferred. If the end of the record is reached during transfer the pointers are set to the first character of the next record.

RSTAF (ReSeT AsciiFile)

Alpha : Filename

The named file will become the working ASCII file. Both record and character pointers are set to zero.

SCHPT (Set CHaracter Pointer)

X-reg : New char pointer

The character pointer is set to the position specified in the X-register. The record pointer remains unchanged.

SFPT (Set Field Pointer)

X-reg : Field position

The character pointer is set to the first character of the specified field in the X-register. The record pointer remains unchanged.

SRCPT (Set ReCord Pointer)

X-reg : New record pointer

The record pointer is set to the record specified in the X-register. The character pointer is not moved unless the new record is shorter than the character position pointed to, in which case the character pointer is set to the end of the record.

ST<>DF (STack <> DAtaFile)

No user input required

Exchanges the stack, in T-Z-Y-X-LastX order, with DF-registers, starting at the DF-pointer. The DF-pointer is not moved.

SWPREC (SWaP RECord)

X-reg : Number of record to swap

The current record and the record specified in the X-register are swapped. The record pointer is not changed. The character pointer is set to the first character of the record.

SZ?AF (SIze ? ASciiFile)

No user input required

Returns the size of the working ASCII file to the X-register. For remarks on stacklift see RCHPT.

X<>CHR (X <> CHaRacter)

X-reg : ASCII value

The ASCII character represented by the value in the X-register is swapped with the character at the current character pointer position.

?EOF (**? End Of File**)

No user input required

If the pointers are at the end of the file, and a program is running the next step is executed, else it is skipped.

If executed from the keyboard and the result is true "YES" is displayed, otherwise "NO".

ERROR MESSAGES

ALPHA DATA Alpha data was input when numerical data was expected.

DATA ERROR Illegal input data has been entered.

DUP FL NAME The specified file-name already exists.

END OF FL An attempt is made to use DF-registers that lay beyond the end of the file.

END OF REC An attempt has been made to set the character pointer beyond the end of the record.

FILE FULL There is no room in the file to append or insert more characters.

FL TYP ERR An attempt has been made to use an incorrect file-type for the required function.

NAME ERROR	The specified name is too long.
NO FIELD	The specified field does not exist.
NO FL NAME	The Alpha register does not contain an RSU filename.
NO HPIL	No HP-IL module is plugged in.
NONEXISTENT	The ES RSU-2A system is not present or function parameters greater than 999 have been inputted.
NO RECORD	An attempt has been made to set the record pointer to a nonexistent record or to operate on record zero.
NO ROOM	None of the RSU blocks have enough free registers to perform the function
NO RSU PAGE	No RSU "block" is initialized, or the RSU "blocks" are switched off.
NOT A FILE	The name in the alpha register is not that of an RSU file.
NO WORKFL	There is no working ASCII file specified.

**PACKING,
TRY AGAIN** There is not enough
room in main memory
to hold the working
file information.

REC TOO LONG An attempt has been
made to lengthen the
record beyond 255
characters.

RUN-TIME MESSAGES

FILE FULL There is no room in
the file to accept
more characters.

NO NEXT REC An attempt has been
made to join two
records when the
record pointer is at
the last record.

REC TOO LONG An attempt has been
made to insert
characters into the
current record when
it already has 255
characters.

PROGRAMMABILITY

All functions are programmable.

ERAMCO SYSTEMS

W. van Alcmade str. 54
1785 LS Den Helder
The Netherlands