RCSL No: 31-D653

Edition: December 1981

Author: Ejvind Lynning

Title:

RC855 Standard Keyboard Driver
Reference Manual

RCSL 42-11592

A/S REGNECENTRALEN
af 1979

Keywords:

RC855, RC805, PI-1 machine, PASCAL80.

Abstract:

This manual describes the functions performed by the RC855 standard keyboard driver and specifies driver request coding and interpretation of answers.

(20 printed pages)

RCSL 42-i1592

FOREWORD

First edition: RCSL No 31-D653.

The RC855 standard keyboard driver as described in the present manual replaces the "transparent" RC850 keyboard driver (ref. [3]) for all purposes involving the RC805 keyboard which is standard on the RC855 display terminal.

Ejvind Lynning
A/S REGNECENTRALEN af 1979, December 1981

TABLE OF CONTENTS PAGE

1. GENERAL DESCRIPTION 1
   1.1 The Keyboard 1
   1.2 The Driver 2
   1.3 Special Keyboard Functions 3

2. KEYBOARD INPUT DRIVER 4
   2.1 Internal Character Code 4
      2.1.1 Control Character Codes 4
      2.1.2 Graphic Character Codes 5
      2.1.3 Attention Character Codes 5
      2.1.4 Edit Character Codes 6

3. KEYBOARD OUTPUT DRIVER 7
   3.1 Set Lamp Request 7
   3.2 Clear Lamp Request 7
   3.3 Sound Alarm Request 7

APPENDICES:

A. REFERENCES 9

B. INTERNAL CHARACTER CODE SETS 10

1. GENERAL DESCRIPTION 1.

The driver described in this manual runs on the PI-1 Machine, ref. [1], implemented on a Z80 microprocessor as part of an RC855 terminal.

1.1 The Keyboard 1.1

The RC805 keyboard is connected to the Z80 processor in the RC855 terminal via a cable and a Z80-SIO interface chip allowing transmission of eight bit character codes both ways. The keyboard has the following important features:

- when a key is pressed on the keyboard a character code is transmitted to the processor; the code identifies the position of the key on the keyboard,
- all keys cause repeated character code transmission when held down long enough,
- by transmitting appropriate code sequences to the keyboard, up to 16 keys, called release-keys, may be set to generate a character code transmission also when released,
- a number of lamps (diodes) in the keys may be switched on and off by the transmission of appropriate code sequences to the keyboard; similarly an audible alarm signal may be switched on and off.

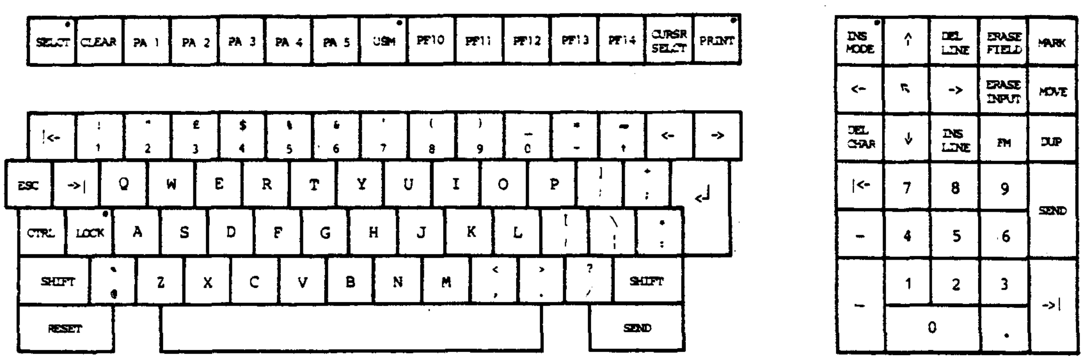

Figure 1: RC805 Keyboard, English (UK) version.

Fig. 1 shows the layout of keys and lamps on the RC805 keyboard. Notice the lamps are located in the SELCT, USM, PRINT, INS MODE, and LOCK keys.

1.2 The Driver 1.2

The RC855 standard keyboard driver performs the following functions:

- initializes the keyboard to work with the two SHIFT keys and the CTRL key as the only release keys,
- converts key position codes received from the keyboard to codes belonging to an internal character code set designed to be convenient for typical applications of the RC855,
- supports upper and lower case,
- supports CTRL + graphic character combinations (yelding graphic character code modulo 32),
- supports alpha lock, i.e. upper case lock on alphabetic characters which are present in both upper and lower case,
- allows the user to switch lamps on and off and to sound the alarm,
- supports a number of special functions invoked by CTRL + an attention character, as described in section 1.3.

Conversion is done according to conversion tables which are present in ROM. Different conversion ROMs exist for different national versions of the keyboard and the corresponding internal character code set.

Alpha lock mode may be set permanently, by means of the configuration parameter KBL. Alternatively the mode is set by pressing the LOCK key, and reset by pressing the key again. Setting alpha lock mode by means of the LOCK key causes the lamp in the key to be lit. All this is handled by the driver. It is not possible for the user program to know when the LOCK key is pressed.

The standard keyboard driver does not allow user defined release keys. An application which requires non-standard release keys should use the transparent keyboard driver which provides direct access to all keyboard codes cf. ref. [3].

In fact the standard keyboard driver consists of two drivers, an input and an output driver, each having its own request semaphore. All request coding for the keyboard drivers is done in accordance with the conventions spelled out in ref. [2], using the u1 and u3 fields. Data buffers are not needed for any request type.

The driver is coded in Z80 assembler language and must be included in a PASCAL system as part of its basic system.

1.3 Special Keyboard Functions 1.3

A number of special keyboard functions may be invoked by CTRL + attention character combinations.

CTRL + CLEAR causes a soft reset of the RC855.

Corresponding to each of the keys PA1-5 and PF10-14, a flag is defined. CTRL + a flag key causes the flag to be set. A flag is reset by CTRL + SHIFT + the flag key.

The PA1 flag, when set, indicates that the display is to be used for a non-standard function, typically a field engineering function, rather than by the currently running application program. Accordingly, the screen is cleared whenever CTRL + PA1 is typed.

CTRL + PRINT will cause a hard copy of the current display image to be printed. This function only works provided the PA1 flag is set, the picture has not been scrolled since CTRL + PA1 was typed, and the printer driver is running and has been initiated.

2. KEYBOARD INPUT DRIVER 2.

The keyboard input driver allows only one type of request, i.e. single character input. Thus no encoding of the requests is necessary.

answer: u3 = internal character code of typed character

There is no timeout handling in the driver, and character input request messages are not removed from the keyboard input request semaphore until the requested character has actually been typed.

2.1 Internal Character Code 2.1

The characters typed in via the RC805 keyboard fall in four categories:

- control characters
- graphic characters
- attention characters
- edit characters

The graphic characters have internal codes that may vary with the national version of the keyboard and the associated conversion ROM, whereas the other three categories always have the same codes. The character codes assigned to the different categories fall in distinct intervals as described in the following subsections.

2.1.1 Control Character Codes 2.1.1

There is only one genuine control character key on the keyboard, viz. ESC (code 27). However, all control character codes, i.e. codes in the range 0-31 may be keyed in as CTRL combinations. A graphic character typed while the CTRL key is held down yields the internal character code of the graphic character module 32.

2.1.2 Graphic Character Codes 2.1.2

Internal codes for graphic characters are assigned according to the appropriate national RC-version of ISO's 7-bit coded character set, ref. [4]. Six different versions of the code set exist, corresponding to the six different versions of the keyboard. The code sets may be found in appendix B.

2.1.3 Attention Character Codes 2.1.3

Attention characters are typically entered when the terminal operator has finished entering data and/or editing the display and wishes to invoke processing of the data. The internal codes associated with the attention characters are all in the interval 129-166, as listed below:

character	code	character	code	character	code
PF1	130	PA1	142	SEND	129
PF2	131	PA2	143	CLEAR	145
PF3	132	PA3	144	USM	152
PF4	133	PA4	150	PRINT	153
PF5	134	PA5	151	SELCT	154
PF6	135	PA6	162	RESET	155
PF7	136	PA7	163	CURSR	
PF8	137	PA8	164	SELCT	149
PF9	138	PA9	165		
PF10	139	PA10	166		
PF11	140				
PF12	141				
PF13	147				
PF14	148				

PF10-PF14, SEND, CLEAR, USM, PRINT and CURSR SELCT yield the same codes whether or not a SHIFT key is held down when the key is pressed. SELCT and RESET can only be entered in combination with a SHIFT key.

PF1-PF9 are entered as SHIFT + a digit in the numeric key bank. PAi+5, where i=1,2,3,4,5 is entered as SHIFT+PAi.

CTRL combinations with attention keys are reserved for very special purposes, viz. soft reset and paflags.

2.1.4 Edit Character Codes 2.1.4

The characters intended to provide display editing functions all have codes in the interval 200-217. None of the edit keys are affected by SHIFT. A few edit characters may be generated by more than one key on the keyboard (tab, back tab, cursor left, cursor right). An edit key pressed while CTRL is held down is ignored by the driver.

The codes generated by the edit keys are shown in the following table:

key	code
INS MODE	200
INS LINE	201
DEL LINE	202
ERASE FIELD	203
DEL CHAR	204
ERASE INPUT	205
↑ (cursor up)	206
<-(cursor left)	207
->(cursor right)	208
↓ (cursor down)	209
↖(cursor home)	210
->| (tab)	211
|<- (back tab)	212
MARK	213
MOVE	214
<┘ (newline)	215
FM	216
DUP	217

3. KEYBOARD OUTPUT DRIVER 3.

The keyboard output driver allows three types of request:

- set a lamp
- clear a lamp
- sound the alarm.

At startup time the driver initializes the keyboard with all lamps and the alarm off.

Requests to the keyboard output driver are processed in the order they arrive at the request semaphore.

An illegal request is answered with u2=4. All other requests are answered with u2=0. Request coding is described in the following three sections.

3.1 Set Lamp Request 3.1

u1 = 0 + 3x4 = 12
u3 = lamp number (0: SELCT, 1: INS MODE, 6: PRINT, 7: USM).

The lamp in the LOCK key is used by the driver to indicate alpha lock mode.

3.2 Clear Lamp Request 3.2

u1 = 0 + 4x4 = 16
u3 = lamp number, as for 'Set Lamp' request.

3.3 Sound Alarm Request 3.3

u1 = 0 + 5x4 = 20
u3 = duration of alarm in units of 20 msecs.

A. REFERENCES A.

[1] RCSL No 31-D615:
The PI-1 Machine, Reference Manual

[2] RCSL No 31-D617:
PASCAL80 Driver Conventions

[3] RCSL No 31-D614:
RC850 Keyboard Driver, Reference Manual

[4] ISO-646:
7-bit Coded Character Set for Information Processing
Interchange

B. INTERNAL CHARACTER CODE SETS B.

Bits 3210	Bits 654 / Hex 1 / Hex 2	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000	0			SP	0	ü	P	ä	p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			#	3	C	S	c	s
0100	4			$	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			'	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	Æ	k	æ
1100	C			,	<	L	Ø	l	ø
1101	D			-	=	M	Å	m	å
1110	E			.	>	N	↑	n	ö
1111	F			/	?	O	_	o	

Figure 2: Standard Danish language version.

Bits 3210	Bits 654 / Hex 1 / Hex 2	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000	0			SP	0	Ö	P	↑	p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			§	3	C	S	c	s
0100	4			$	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			'	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			¤	:	J	Z	j	z
1011	B			+	;	K	Æ	k	æ
1100	C			,	<	L	Ø	l	ø
1101	D			-	=	M	Å	m	å
1110	E			.	>	N	Ü	n	ü
1111	F			/	?	O	_	o	

Figure 3: Danish language public sector version.

Bits 654		000	001	010	011	100	101	110	111
Bits 3210	Hex 1 / Hex 2	0	1	2	3	4	5	6	7
0000	0			SP	0	É	P	é	p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			#	3	C	S	c	s
0100	4			¤	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			'	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	Ä	k	ä
1100	C			,	<	L	Ö	l	ö
1101	D			-	=	M	Å	m	å
1110	E			.	>	N	Ü	n	ü
1111	F			/	?	O	_	o	

Figure 4: Swedish language version.

Bits 654		000	001	010	011	100	101	110	111
Bits 3210	Hex 1 / Hex 2	0	1	2	3	4	5	6	7
0000	0			SP	0	§	P	`	p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			#	3	C	S	c	s
0100	4			¤	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			'	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	Ä	k	ä
1100	C			,	<	L	Ö	l	ö
1101	D			-	=	M	Ü	m	ü
1110	E			.	>	N	↑	n	ß
1111	F			/	?	O	_	o	

Figure 5: German language version.

	Bits 654	000	001	010	011	100	101	110	111
Bits 3210	Hex 1 / Hex 2	0	1	2	3	4	5	6	7
0000	0			SP	0	@	P	`	p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			£	3	C	S	c	s
0100	4			$	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			'	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	[	k	{
1100	C			,	<	L	\	l	¦
1101	D			-	=	M	]	m	}
1110	E			.	>	N	↑	n	~
1111	F			/	?	O	_	o	

Figure 6: English language (UK) version.

	Bits 654	000	001	010	011	100	101	110	111
Bits 3210	Hex 1 / Hex 2	0	1	2	3	4	5	6	7
0000	0			SP	0	@	P	`	p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			#	3	C	S	c	s
0100	4			$	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			'	7	G	W	g	w
1000	8			(	8	H	X	h	x
1001	9			)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	[	k	{
1100	C			,	<	L	\	l	¦
1101	D			-	=	M	]	m	}
1110	E			.	>	N	↑	n	~
1111	F			/	?	O	_	o	

Figure 7: English language (US) version.

RETURN LETTER

Title: RC855 Standard Keyboard Driver Reference Manual RCSL No.: 31-D653

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

____

____

____

____

Do you find errors in this manual? If so, specify by page.

____

____

____

____

How can this manual be improved?

____

____

____

____

Other comments?

____

____

____

____

____

Name: ______________________ Title: ______________________

Company: ______________________________________

Address: ______________________________________

Date: __________

Thank you

42-i 1288

Fold here

Do not tear - Fold here and staple

Affix
postage
here

REGNECENTRALEN
af 1979

rmation Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark