Rotel RDG-1520 / RT-09
RS232 ASCII Controller Command List

Date	Version	Update Description
January 18, 2012	1.00	Original Specification
October 9, 2012	1.01	Added memory command
March 1, 2013	1.02	Added discrete opt/coax input & power status commands. Requires Main Software V1.1.7
June 4, 2014	1.10	Added special character mapping

Rotel has released updated software for RDG-1520 and RT-09. The new software includes an ASCII based RS232 protocol. The new protocol replaces the original HEX based communication protocol. This new protocol is ASCII text based to make it easier for application programmers to communicate with Rotel products.

The new protocol is effective starting with the Main software version V1.1.5.

The new protocol eliminates the Device ID and checksum requirements. The RS232 hardware does not support flow control so care needs to be take when sending and receiving data to avoid packet loss.

All commands sent to the attached Rotel device must have a terminating "!" character . Status information from the attached Rotel product with either have a terminating "!" character or a byte count for variable length text data that may include a "!" in the returned message. It is up to the sending/receiving control application to properly parse and process the packets.

Note: *The byte count only includes the text data and not the length or "," character.*

Note 2: *Do not include a carriage return or line feed after the command, only the "!" terminating character.*

Connection Settings

Baud Rate	Parity	Valid Data Bits	Stop Bit Value	Handshaking	Data Type
115200	N	8	1	None	String

Communication Protocol

Command and response messages are included on the following pages. Automatic display update information can be enabled/disabled using the "display_update_auto" and "display_update_manual".

In automatic mode each time the display changes the new display line(s) will be sent.

In manual mode the display updates must be requested each time a refresh of the display information is desired.

Section 1: Control Command List

RDG-1520 ASCII	Command Description	Unit Response
POWER & VOLUME COMMANDS		
power_on!	Power On	power=on!
power_off!	Power Off	power=standby!
power_toggle!	Power Toggle	power=on/standby!
SOURCE SELECTION COMMANDS		
iradio!	Source iRadio	source=iradio!
network!	Source Network	source=network!
aux1_coax!	Source Aux 1 Coax	source=aux1_coax!
aux1_opt!	Source Aux 1 Optical	source=aux1_opt!
fm!	Source FM	source=fm!
dab!	Source DAB	source=dab!
usb!	Source USB	source=usb!
aux1!	Source Aux 1 Coax/Opt Toggle	source=aux1_coax! / source=aux1_opt!
SOURCE CONTROL COMMANDS		
play!	Play Source	play_status=play!
stop!	Stop Source	play_status=stop!
pause!	Pause Source	play_status=pause!
track_fwd!	Track Forward / Tune Up	track=##, T##
track_back!	Track Backward / Tune Down	track=##, T##
fast_fwd!	Fast Forward / Search Forward	time=##:##:##!
fast_back!	Fast Backward / Search Backward	time=##:##:##!
random!	Random Play Mode Toggle	n/a
repeat!	Repeat Play Mode Toggle	n/a
MENU CONTROL COMMANDS		
menu!	Display the Menu	n/a
exit!	Exit Key	n/a
up!	Cursor Up	n/a
down!	Cursor Down	n/a
left!	Cursor Left	n/a
right!	Cursor Right	n/a
enter!	Enter Key	n/a
enter_long!	Long Press for Enter Key	n/a
NUMERIC KEY COMMANDS		
1!	Number Key 1	n/a
2!	Number Key 2	n/a
3!	Number Key 3	n/a
4!	Number Key 4	n/a
5!	Number Key 5	n/a
6!	Number Key 6	n/a
7!	Number Key 7	n/a

RDG-1520 ASCII	Command Description	Unit Response
8!	Number Key 8	n/a
9!	Number Key 9	n/a
0!	Number Key 0	n/a
10_plus!	Number Key 10+	n/a
FM / DAB / IRADIO PRESET COMMANDS		
memory!	Select memory for saving presets	n/a
call_iradio_preset_n	Recall iRadio Preset n (n = 01 - 30)	iradio_preset_n=##,text
call_fm_preset_n	Recall FM Preset n (n = 01 - 30)	fm_preset_n=##,text
call_dab_preset_n	Recall DAB Preset n (n = 01 - 30)	dab_preset_n=##,text
DISPLAY REFRESH COMMANDS		
display_update_auto!	Set Display Update to Auto	display_update=auto!
display_update_manual!	Set Display Update to Manual	display_update=manual!

Section 2: Feedback Request Command List

Command:	get_display!
Description:	Request the entire display to be sent
Return String:	display=###,text
Return Description:	Current display data; must include 3 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	display=080, Sample Text

Command:	get_display1!
Description:	Request display line #1 to be sent
Return String:	display1=##,text
Return Description:	Current display line 1, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	display1=20, Sample Text

Command:	get_display2!
Description:	Request display line #2 to be sent
Return String:	display2=##,text
Return Description:	Current display line 2, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	display2=20, Sample Text

Command:	get_display3!
Description:	Request display line #3 to be sent
Return String:	display3=##,text
Return Description:	Current display line 3, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	display3=20, Sample Text

Command:	get_display4!
Description:	Request display line #4 to be sent
Return String:	display4=##,text
Return Description:	Current display line 4, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	display4=20, Sample Text

Command:	get_product_type!
Description:	Request the product type
Return String:	product_type=##,text
Return Description:	Rotel product type name, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	product_type=08,RCX-1500

Command:	get_product_version!
Description:	Request the main CPU software version
Return String:	product_version=##,text
Return Description:	Rotel main CPU software version, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	product_version=13,v1.1.2-110316

Command:	get_display_size!
Description:	Request display size
Return String:	display_size=##,##!
Return Description:	Columns and rows on current display
Example:	display_size=20,04!

Command:	get_display_update!
Description:	Request display update
Return String(s):	display_update=auto! / display_update=manual!
Return Description:	Status of if the display refresh is automatic or manual
Example:	display_update=auto!

Command:	get_current_power!
Description:	Request current power status
Return String(s):	power=on! / power=standby!
Return Description:	Current power status
Example:	power=on!

Command:	get_current_source!
Description:	Request current source
Return String(s):	source=cd! / source=iradio! / source=network! / source=aux1! / source=aux2! / source=usb! / source=fm! / source=dab!
Return Description:	Current source
Example:	source=cd!

Command:	get_current_preset!
Description:	Request current preset
Return String(s):	preset_iradio=##! / preset_fm=##! / preset_dab=##!
Return Description:	Current preset station 2 digit length
Example:	preset_iradio=13!

Command:	get_iradio_preset_n!
Description:	Request saved iRadio station info for present n (n = 1 – 30)
Return String(s):	iradio_preset_n=##,text
Return Description:	Saved iRadio preset number, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	iradio_preset_02=##,text

Command:	get_allpreset_iradio!
Description:	Request all saved iRadio station info [1..30]
Return String(s):	iradio_preset_n=##,text
Return Description:	Saved iRadio preset number, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	iradio_allpreset_01=##,text .. iradio_allpreset_30=##,text

Command:	get_fm_preset_n!
Description:	Request saved FM station info for present n (n = 1 – 30)
Return String(s):	fm_preset_n=##,text
Return Description:	Saved FM preset number, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	fm_preset_04=##,text

Command:	get_allpreset_fm!
Description:	Request all saved FM station info [1..30]
Return String(s):	fm_preset_n=##,text
Return Description:	Saved FM preset number, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	fm_allpreset_01=##,text .. fm_allpreset_30=##,text

Command:	get_dab_preset_n!
Description:	Request saved DAB station info for present n (n = 1 – 30)
Return String(s):	dab_preset_n=##,text
Return Description:	Saved DAB preset number, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	dab_preset_01=##,text

Command:	get_allpreset_dab!
Description:	Request all saved DAB station info [1..30]
Return String(s):	dab_preset_n=##,text
Return Description:	Saved DAB preset number, must include 2 digit length of text string at beginning followed by "," and text string (no terminating character)
Example:	dab_allpreset_01=##,text .. dab_allpreset_30=##,text

Command:	get_play_status!
Description:	Request play status
Return String(s):	play_status=play! / play_status=stop! / play_status=pause!
Return Description:	Source play status
Example:	play_status=play!

Section 3: Special Character Mapping

Certain characters on the RDG-1520 display may be represented by a combination of 2-3 hex bytes in the feedback string provided by the unit. Refer to the chart below for a mapping of the different characters.

Symbol	Hex Value	Symbol	Hex Value	Symbol	Hex Value
A	EE 82 85	D	EE 82 8A	■	EE 82 99
C	EE 82 84	◖	EE 82 8B	□	EE 82 9A
F	EE 82 92	‖	EE 82 81	(	EE 82 88
G	EE 82 87	►	EE 82 82	→	EE 82 95
I	EE 82 8E	■	EE 82 83	←	EE 82 96
L	EE 82 89	◄	EE 82 94	*	EE 82 90
M	EE 82 93	►	EE 82 97	■	EE 82 91
R	EE 82 8C	◄	EE 82 98	Z	EE 82 8D
S	EE 82 8F	T	EE 82 80	END	EE 80 80 EE 80 81 EE 80 82