ROTEL®

Rotel X430
RS232 / IP ASCII Controller Command List

Date	Version	Update Description
04/21/2026	1.00	Original Specification

The X430 supports an ASCII based RS232 and IP protocol. 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.

Example Command: power_on!

Note: *Do not include any spaces in the command, and do not include a carriage return or line feed after the command, only the "!" terminating character.*

Status information from the attached Rotel product will have a terminating "$" character. It is up to the sending/receiving control application to properly parse and process the packets.

Connection Settings

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

IP Control Settings

The X430 will accept and respond to IP control commands if the product is connected to a local network and has a valid IP address.

Commands will be accepted via TCP port 9596, and the unit will send responses back via the same port. The command and response format is identical to the serial commands.

Communication Protocol

Command and response messages are included on the following pages. Automatic status update information can be enabled/disabled using the "rs232_update_on" and "rs232_update_off" commands.

With RS232 update set to ON, any status changes to the unit will be transmitted via RS232. With RS232 update set to OFF, the unit will not send any feedback unless polled by the controller.

Section 1: Control Command List

X430 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$
vol_up!	Volume Up	volume=##$
vol_dwn!	Volume Down	volume=##$
vol_nn!	Set Volume to level n (00 - 96)	volume=##$
mute!	Mute Toggle	mute=on/off$
mute_on!	Mute On	mute=on$
mute_off!	Mute Off	mute=off$
SOURCE SELECTION COMMANDS		
cd!	Source CD	source=cd$
coax1!	Source Coax 1	source=coax1$
coax2!	Source Coax 2	source=coax2$
coax3!	Source Coax 3	source=coax3$
opt1!	Source Optical 1	source=opt1$
opt2!	Source Optical 2	source=opt2$
opt3!	Source Optical 3	source=opt3$
aux!	Source Aux	source=aux$
tuner!	Source Tuner	source=tuner$
phono!	Source Phono	source=phono$
bluetooth!	Source Bluetooth	source=bluetooth$
bal_xlr!	Source XLR	source=bal_xlr$
pcusb!	Source PC-USB	source=pcusb$
SOURCE CONTROL COMMANDS		
play!	Play Source	n/a
stop!	Stop Source	n/a
pause!	Pause Source	n/a
trkf!	Track Forward/Tune Up	n/a
trkb!	Track Backward/Tune Down	n/a
TONE CONTROL COMMANDS		
bypass_on!	Tone Bypass On	bypass=on$
bypass_off!	Tone Bypass Off	bypass=off$
bass_up!	Bass Up	bass=000/+##/-##$
bass_down!	Bass Down	bass=000/+##/-##$
bass_-nn!	Set Bass to -n (n = 01-10)	bass=-##$
bass_000!	Set Bass to 0	bass=000$
bass_+nn!	Set Bass to +n (n = 01-10)	bass=+##$
treble_up!	Treble Up	treble=000/+##/-##$

X430 ASCII	Command Description	Unit Response
treble_down!	Treble Down	treble=000/+##/-##$
treble_-nn!	Set Treble to -n (n = 01-10)	treble=-##$
treble_000!	Set Treble to 0	treble=000$
treble_+nn!	Set Treble to +n (n = 01-10)	treble=+##$
BALANCE CONTROL COMMANDS		
balance_r!	Balance Right	balance=000/l##/r##$
balance_l!	Balance Left	balance=000/l##/r##$
balance_lnn!	Set Balance to left n (n = 01-15)	balance=l##$
balance_000!	Set Balance to 0	balance=000$
balance_rnn!	Set Balance to right n (n = 01-15)	balance=r##$
SPEAKER COMMANDS		
speaker_a!	Toggle Speaker A Output	speaker=a/a_b/off$
speaker_b!	Toggle Speaker B Output	speaker=b/a_b/off$
RS232 FEEDBACK COMMANDS		
rs232_update_on!	Set RS232 Update to Auto (On)	update_mode=auto$
rs232_update_off!	Set RS232 Update to Manual (Off)	update_mode=manual$

Section 2: Feedback Request Command List

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

Command:	source?
Description:	Request current source
Return String(s):	source=cd$ / source=coax1$ / source=coax2$ / source=coax3$ source=opt1$ / source=opt2$ / source=opt3$ / source=tuner$ / source=phono$ / source=aux$ / source=pc_usb$ / source=bal_xlr$ / source=bluetooth$
Return Description:	Current source
Example:	source=usb$

Command:	name?
Description:	Request current source name
Return String(s):	name=custom_input_name$
Return Description:	Current source's name – default or custom name
Example:	name=custom_input_name$

Command:	volume?
Description:	Request current volume value
Return String(s):	volume=##$
Return Description:	2 digit current volume level
Example:	volume=40$

Command:	mute?
Description:	Request current mute status
Return String(s):	mute=on$ / mute=off$
Return Description:	Current mute status
Example:	mute=off$

Command:	bypass?
Description:	Request current tone bypass state
Return String(s):	bypass=on$ / bypass=off$
Return Description:	Current tone bypass state
Example:	bypass=off$

Command:	bass?
Description:	Request current bass level
Return String(s):	bass=###$ (+01-10, -01-10, 000)
Return Description:	Current tone control bass level
Example:	bass=+02$

Command:	treble?
Description:	Request current treble level
Return String(s):	treble=###$ (+01-10, -01-10, 000)
Return Description:	Current tone control treble level
Example:	treble=-01$

Command:	balance?
Description:	Request current balance setting
Return String(s):	balance=###$ (l01-15, r01-15, 000)
Return Description:	Current balance setting
Example:	balance=r03$

Command:	speaker?
Description:	Request current active speaker outputs
Return String(s):	speaker=a$ / speaker=b$ / speaker=a_b$ / speaker=off$
Return Description:	Current active speaker outputs
Example:	speaker=a$

Command:	freq?

Description:	Request current frequency for digital source input
Return String(s):	freq=off$ / freq=32$ / freq=44.1$ / freq=48$ / freq=88.2$ / freq=96$ / freq=176.4$ / freq=192$ / freq=384$ / freq=2.8m / freq=not_support
Return Description:	Current frequency for digital source input
Example:	freq=48$

Command:	pcusb?
Description:	Request current PC-USB class
Return String(s):	pcusb_class=1$ / pcusb_class=2$
Return Description:	Current PC-USB class
Example:	pcusb_class=2$

Command:	version?
Description:	Request the current software version
Return String:	version=#.##$
Return Description:	Rotel current software version
Example:	version=1.53$

Command:	pc_version?
Description:	Request the PC-USB software version
Return String:	pc_version=#.##$
Return Description:	Rotel PC-USB software version
Example:	pc_version=1.13$

Command:	ip?
Description:	Request the IP address of the product
Return String:	ipaddress=###.###.###.###$
Return Description:	Current IP address
Example:	ipaddress =192.168.100.8$

Command:	mac?
Description:	Request the MAC address of the product
Return String:	mac=############$
Return Description:	MAC address (uppercase characters)
Example:	mac=0CEFAF90125E$

Command:	model?
Description:	Request the model number
Return String:	model=text$
Return Description:	Rotel model number
Example:	model=X430$

Command:	discover?

Description:	Request the device to identify itself on the network
Return String:	discover=ip=###.###.###.### port=#### mac=############$
Return Description:	Device's IP address, port number and MAC address
Example:	discover=ip=192.168.100.25 port=9596 mac=0cefaf90125e$