Open Two Radio Switching Protocol (OTRSP)

Version	Date	Author	Change Description
0.1	6 Aug 2009	Paul Young	First draft
0.2	7 Aug 2009	Paul Young	Reduce number of tables needed for command parsing. Make events simpler. Add abort.
0.3	7 Aug 2009	Paul Young	Add reverse stereo
0.4	17 Aug 2009	Paul Young	Add serial port speed, CR command. Rewrite for greater clarity.
0.5	9 Jan 2010	Paul Young	Add description of SO2R device. Change wording for consistency (no functional changes)
0.6	10 Sept 2010	Paul Young	Add FW command and voice keyer commands.
0.7	20 June 2011	Paul Young	Add non-COM communications
0.8	27 Jan 2019	Paul Young	Multi-radio and multi-operator extensions
0.9	19 Mar 2022	Paul Young	Add TCP/IP communications

Summary

This is a protocol for communicating with a SO2R (Single Operator 2 Radio) device. It is a serial protocol and is suitable for use on a physical or virtual serial port or other communications channel. It is intended to replace the LPT port which has historically been used to control an SO2R device and to offer features which could not be supported by the LPT port.

This is an open protocol. The authors disclaim any proprietary or intellectual property rights to it. It may be implemented by anyone for any legal purpose.

SO2R Device

An SO2R device is a device which connects to two radios and allows an operator to select which radio is to be used for transmitting and receiving.

The device provides an interlocking capability – only one transmitter can transmit at a time.

It may be possible for two receivers to be used at once – The SO2R device may optionally support a stereo mode where one receiver is heard in each ear.

The transmitter which is selected is said to have "transmit focus". The receiver or receivers which are selected are said to have "receive focus".

The selection of transmit and receive focus can be controlled by the computer using the OTRSP commands described in this document. The SO2R device may optionally also have override capabilities where an operator can force a specific transmitter or receiver to have focus.

An SO2R device may switch the microphone, CW key, and PTT signals to a specific radio in order to give that radio transmit focus.

An SO2R device may optionally have a PTT switch or a connector for an external PTT pushbutton or footswitch. The SO2R device will connect this to the transmitter which has focus. Usually the computer can also set PTT.

An SO2R device may optionally have auxiliary outputs. Typically if it has them it would have one or two groups of four bit outputs. These can be controlled by the computer. They are typically used for antenna selection. Although not strictly part of SO2R operation they have historically been included in SO2R hardware.

An SO2R device may optionally have control inputs. These usually have no direct effect on the SO2R Box or the radios but a computer program can read the values and make changes based on them. The exception is the PTT switch which, if it exists, is a control but also can switch a radio from receive to transmit.

Figure 1 shows an example SO2R device. The microprocessor communicates with the computer using the OTRSP protocol and it controls circuitry which switches the microphone, key, PTT, and headphones.
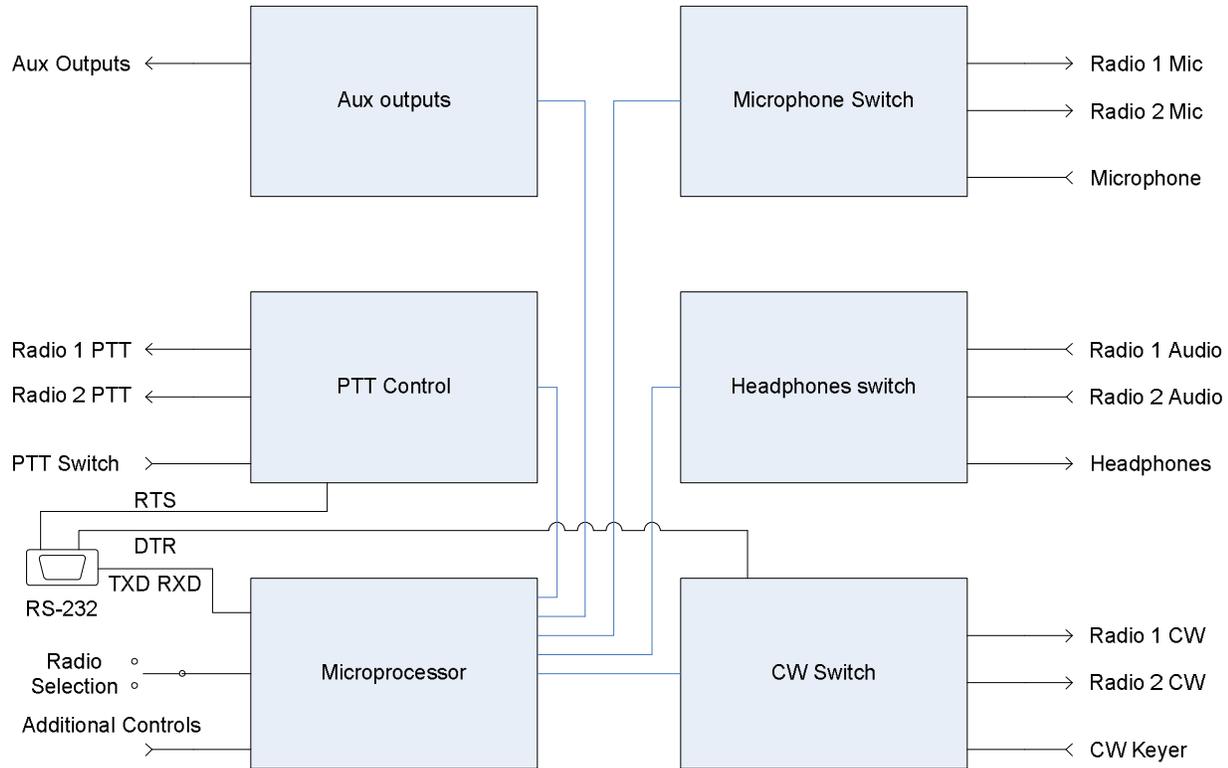
Figure 1. An SO2R device

An SO2R device may not implement all of these features. For example, it may not have any Aux outputs or additional controls, and there may be no manual override for the computer's transmit and receive focus selection.

Additional devices may be packaged with the SO2R device. For example a box may also contain a CW keyer or a sound card. These do not affect the SO2R OTRSP functions.

An SO2R device may optionally be able to control more than two radios. The OTRSP commands can only set transmit and receive focus for two radios. There is a command which specifies which two radios to use.

Serial Port Characteristics

The standard port parameters for an OTRSP SO2R device are 9600 baud, no parity, 8 bit data, one stop bit.

PTT and CW

PTT and CW keying can be controlled by the serial port lines RTS and DTR respectively. Support for PTT and CW keying are optional, as they can be done in other ways, for example through a keyer.

PTT is equivalent to LPT port pin 16. CW is equivalent to LPT port pin 17.

Note: It is possible for PTT and CW to be implemented using RTS and DTR on a serial port other than the one used for OTRSP commands.

Commands, Responses, and Events

A command is a string of characters sent from the controlling program to the SO2R device.

A response is a string of characters sent by the SO2R device to the controlling program in response to a command.

An event is a string of characters sent by the SO2R device to the controlling program in response to a state change in the SO2R device.

All characters are from the UTF-8 character set. All commands, responses, and events are terminated by a carriage return, UTF-8 character 13. The carriage return is represented in this document by the dingbat "↵".

The parts of a command are:

Query Prefix   Single character "?". If the query prefix is present the command is a query.

Keyword   One to six characters. The keyword identifies the command.

Item   Single digit, 0-9. May or may not exist, depending on the command. Identifies a particular item, such as a port or radio.

Value   One or more characters. Always present if the command is not a query.

Carriage Return   Terminates a command.

There is no delimiter between the parts of a command.

Commands which are not query commands will have a keyword, possibly an item and a value. Commands which are query commands will have the query prefix, a keyword, and possibly an item.

All non-query commands have an equivalent query command, which is the keyword and possibly the item prefixed with the query prefix. The SO2R device response will be the command which would set the current value.

All command and event strings are at most 16 characters in length. All responses are less than 40 characters in length. These lengths include the return character.

If an unrecognized command which is not a query command is sent, the SO2R device will ignore it.

If an unrecognized query command is sent, the SO2R device will respond with the same command, including the question mark. For example, if the control program sent ?ABC↵ the SO2R device

would respond ?ABC↵. This can be used to find out if a command is implemented by the SO2R Device.

If a query prefix with no keyword is sent, then the response will be the query prefix. In other words, if the control program sends ?↵ the SO2R device will respond ?↵. This can be used as a "ping" command to verify that the SO2R device is present.

All commands beginning with the letter V are reserved and vendor specific. A vendor might use commands for device specific functions such as firmware update or device-specific status. These are not meant for normal SO2R device features. If a new feature is desired a vendor should propose a new command to be added to the protocol so that all vendors who support the feature will do so in the same way.

Supported Commands

This is the complete set of commands (other than the voice keyer extension) as of this version of the protocol:

Command   Item   Value

TX   n

> Set which radio has transmit focus. The values of n are 1 and 2. The ?TX command returns the actual radio which has transmit focus, which may be different than what was set by the previous TX command if the device selection has been overridden, for example by a front panel switch on the SO2R device.
>
> This is the equivalent of LPT port pins 3 and 14.

RX   n

> Set which radios have receive focus, possibly receiving in stereo. The values of n are 1, 2, 1S, 2S, 1R, and 2R.
>
> There are four choices for what radio or radios are connected to the headphones.
>
> a. Radio 1 is connected to both headphones.
>
> b. Radio 2 is connected to both headphones.
>
> c. Radio 1 is connected to the left headphone and radio 2 is connected to the right headphone. This is called stereo.
>
> d. Radio 1 is connected to the right headphone and radio 2 is connected to the left headphone. This is called reverse stereo.
>
> Not all SO2R devices support all of these modes. The value has information for an SO2R device to use a mode it supports. These are the values and what they mean:

Value	Function
1	Connect radio 1 to both headphones.
2	Connect radio 2 to both headphones.
1S	Connect radio 1 to the left headphone and radio 2 to the right headphone (stereo). If stereo is not possible connect radio 1 to both headphones.
2S	Connect radio 1 to the left headphone and radio 2 to the right headphone (stereo). If stereo is not possible connect radio 2 to both headphones.
1R	Connect radio 1 to the right headphone and radio 2 to the left headphone (reverse stereo). If reverse stereo is not possible connect radio 1 to the left headphone and radio 2 to the right headphone (stereo). If stereo is not possible connect radio 1 to both headphones.
2R	Connect radio 1 to the right headphone and radio 2 to the left headphone (reverse stereo). If reverse stereo is not possible connect radio 1 to the left headphone and radio 2 to the right headphone (stereo). If stereo is not possible connect radio 2 to both headphones.

The ?RX command returns the mode actually set for the receiving radio which may be different than what was set by the previous RX command if the SO2R device does not support the requested mode or if the device selection has been overridden, for example by a front panel switch on the SO2R device.

Receiver selection is the equivalent of LPT port pin 4. Stereo selection is the equivalent of LPT port 5.

AUX   x   n

Sets the auxiliary port. x is the port number. The values of x are 0-9. The value of n is a one to three digit decimal number. The value of n is 0-255.

If the auxiliary port does not exist the command is ignored. If the value is larger than the port supports the high order bits will be ignored.

There is a convention that auxiliary port 1 contains band or antenna information for radio 1 and auxiliary port 2 contains band or antenna information for radio 2.

The ?AUXx command returns the value actually set for a port and will not include bits which were ignored. For example, if port 1 is 4 bits, the command AUX125 will set the value to 9 and ?AUX1 will return AUX19.

If a port does not exist the response to a query for that port would be that the command does not exist. For example, ?AUX0 would return ?AUX0 if there is no port 0.

Auxiliary port 1 is equivalent to LPT port pins 2, 7, 8, 9.

BAND   x   n

Tells the SO2R device the band information for a radio. x is the radio number. The values of x are 1 or 2. The value of n is the frequency in MHz of the bottom of the band, for example 3.5 or 144.

If the band is not set ?BAND⏎ would return BAND0⏎.

MODE   x   n

Tells the SO2R device the mode information for a radio. x is the radio number. The values of x are 1 or 2. The value of n is C for CW, U for USB, L for LSB, R for RTTY, F for FM, A for AM, X for other.

If the mode is not set ?MODE⏎ would return MODE0⏎.

MAP   x   n

Tells the SO2R device to map a physical radio to one of the two radios specified in the TX and RX commands. X is the radio number. The values of x are 1 or 2. The value of n is a one to three digit number corresponding to the physical radio.

This command is intended for SO2R devices which support more than two radios. It is not needed for an SO2R device which supports only two radios.

If this command is not implemented it is assumed that radio 1 specified in the TX and RX commands is always physical radio 1 and radio 2 is always physical radio 2.

The controlling program should not issue this command unless it is prepared to deal with an SO2R device having more than two radios connected.

NAME   n

Sets the name of the SO2R device. The value of n is the name to set the SO2R device to.

This command is usually ignored. The ?NAME command is more useful as it returns the name of the SO2R device.

The maximum length of the name is 32 bytes. It is valid to use characters from the multi-byte UTF-8 character set but be aware that some programs might not handle these characters correctly.

FW   n

Sets the firmware version of the SO2R device. The value of n is the firmware version to set the SO2R device to. The format of the firmware version is x.y.z where x is the major

version, y is the minor version, and z is the product or patch version. x, y, and z are one to three digit numbers with optional leading zeros.

This command is usually ignored. The ?FW command is more useful as it returns the firmware version of the SO2R device.

CR   x   n

Sets the value of a control. x is the control. The values of x are 0-9. The value of n is the value to set the control to.

This command is usually ignored, as controls usually cannot be written to. The ?CR command is more useful because it returns the current value of the control.

Controls can be items such as switches, which will return values 0 if open and 1 if closed, or a more complex control which can return a value between 0 and 255.

CR0 is reserved for a PTT switch which is known to have the side effect of activating the transmitter which has transmit focus.

ERX   n

Enable RX events. The value of n is 1 to enable the events and 0 to disable the events. See the Unsolicited Events section for more details.

ETX   n

Enable TX events. The value of n is 1 to enable the events and 0 to disable the events. See the Unsolicited Events section for more details.

ECR   x   n

Enables events for a control. The values of x are 0-9. The value of n is 1 to enable the events and 0 to disable the events. See the Unsolicited Events section for more details.

Unsolicited Events

If unsolicited events are enabled the SO2R device may send unsolicited strings to the controlling program. All unsolicited events are disabled by default. The event will start with the $ character, followed by the event name, item number if used, value, and a carriage return.

Enabling an event which has a value causes it to immediately send an event string with the current value of the event.

The events currently defined are:

Event   Item   Value

RX   n

     The receiver selection has changed. The values of n are 1, 1S, 1R, 2, 2R, and 2S. The
     change could be in response to a command or to a change in the SO2R device, such as a
     user throwing a switch to set the receiver.

TX   n

     The transmit focus has changed. The values of n are 1 or 2. The change could be in
     response to a command or to a change in the SO2R device such as a user throwing a switch
     to set the transmitter.

AB

     The user has done something bad, such as trying to change the transmit focus while PTT is
     set. The controlling program should abort any transmission if possible.

CR   x   n

     A control has been changed. x is the control index. n is the value, which can range from 0
     to 255, expressed as a one to three digit integer. Switches normally have values of 0 for
     open and 1 for closed.

Timing and Synchronization

The SO2R device operates as a synchronous device. Commands are executed in order. A
command completes before the next command is started.

An SO2R device is expected to be able to accept characters while it is sending characters. It is not
necessary for the control program to wait for the result of a query before sending another
command.

There is a limit to how fast commands can be sent to an SO2R device. Normally at least five
commands can be sent at a time. However the SO2R device may have a buffer as small as 64
characters so if long commands are sent it may lose characters.

Also the SO2R device may have a limited output buffer size. If many queries are sent this buffer
could be filled. Since most queries are almost the same length as their responses this should not be
a problem in normal use. Note that the ?NAME command may send a longer response than other
commands.

The SO2R device must not send events frequently enough that they will overrun a normal
communications buffer.

An SO2R device is expected to respond to a query within 250 ms.

Command examples

TX1↵

    Future transmissions will be from radio 1.

RX2S↵

    Receive stereo if possible, otherwise from radio 2.

AUX17↵

    Set the value of the auxiliary port 1 pins to 7.

?AUX9↵

    If there is an aux 9 port then the SO2R device would respond AUX9n↵ where n is the current value on the port. If there is no aux 9 port the so2r device would respond ?AUX9↵.

MODE1L↵

    Radio 1 has been set to lower sideband.

?NAME↵

    The SO2R device responds NAME followed by the name of the device followed by a carriage return. If the device was named "SO2R Device" it would respond with NAMESO2R↵

?FW↵

    The SO2R device responds FW followed by the firmware version of the device followed by a carriage return. If the major version was 1, the minor was 2, and the patch was 34 it could respond with FW1.2.34↵

?XXXX↵

    XXXX is not a known command. Reply ?XXXX↵.

?CR0↵

    Returns the value of the CR0 control. The device replies with the current value. This is normally a PTT switch. If the switch was closed it would respond CR01↵. If the switch was open it would respond CR00↵. If there is no PTT switch it would respond ?CR0↵.

ECR21↵

Enable event CR2. The device replies with an event that contains the current value of CR2. If CR2 were a footswitch which was not pressed it would reply $CR20↵. If the switch were later pressed it would send $CR21↵. If the SO2R device does not have a CR2 control there will be no response.

?ECR2↵

Returns ECR21↵ if the CR2 event is enabled or ECR20↵ if not. If the SO2R device does not have a CR2 control it will respond ?ECR2↵.

EAB1↵

Turns on the abort event. There is no response. If the SO2R device wants transmission to cease it will send $AB↵.

TX1↵
?TX↵

Future transmissions will be from radio 1. The device will respond TX1↵. It will not respond until the transmit focus has been set to radio 1.

TX1↵
RX1↵
?TX↵
?RX↵

Future transmissions will be from radio 1. Radio 1 will be the receiving radio. The device will respond TX1↵RX1↵ after radio 1 has transmit and receive focus.

Minimum Device Implementation

An SO2R Device must implement the following:

The TX and RX commands and the ?TX and ?RX commands.

The AUX command and ?AUX command if it has any Auxiliary ports.

The ?NAME command.

For ?<command> where the command or item is not recognized, return the the command unmodified.

It should be capable of operation on a port set to 9600 baud, no parity, 8 data bits, 1 stop bit.

Minimum Control Program Implementation

A control program could replace LPT port SO2R control with this protocol by implementing the following:

The TX1, TX2, RX1, RX2, RX1S, RX2S, RX1R, and RX2R commands.

RX1R and RX2R can be treated as RX1S and RX2S.

RTS as a PTT control if appropriate.

The AUX1 command if the control program supports sending band or antenna information.

Note that the SO2R device will never send any characters unless a command is sent that causes a response or an event is enabled.

A minimum implementation can send commands and never needs to read anything.

If the control program sets the speed of the serial line it should be capable of setting it to 9600 baud.

Control program enhancements

Send ?NAME⏎ after connecting to the SO2R device and display or log the result.

If the control program can provide information for two radios use the AUX2 command to put out this information for the second radio.

If the SO2R device is known to be very slow it might be prudent to send a TX command followed by a ?TX command and to wait for the result before transmitting.

Send the BAND and MODE commands. The SO2R device will ignore them if it has not implemented them.

Enable the abort event and stop transmitting if an abort event is received. This can help protect a good station from a bad operator.

Enable events for receive and transmit selection and set transmit and receive focus if they are different than the current selections.

Enable events for a couple controls and allow them to be mapped to useful functions, for example allowing a footswitch control to change focus.

 Use the MAP command to handle more than two radios.

Voice Keyer Extension

OTRSP can be used to control a voice keyer. This uses additional commands and events. The commands and events will not be recognized by an SO2R box which does not implement a voice keyer.

The voice keyer is not strictly part of an SO2R device but there are several SO2R devices with integrated voice keyers. The voice keyer logically is between the microphone and the microphone switch.

The voice keyer is expected to assert PTT when it plays a message and to drop PTT when it finishes or aborts the message.

Supported Voice Keyer Commands

Command	Item	Value
KR		n

Starts recording a message. n is the message number, which can range from 1 to 255, expressed as a one to three digit integer. Not all voice keyers support all message numbers. The ?KR command returns the message number of the message currently being recorded or 0 if no message is being recorded.

KS		

Stops recording. The command has no effect if a message is not being recorded. The ?KS command returns KS.

KP		n

Starts playing a message. N is the message number, which can range from 1 to 255. The behavior of the KP command when the voice keyer is already playing a message is not defined. The ?KP command returns the message number of the message currently playing or 0 if no message is being played.

KA		

Aborts any message which is being played. The command has no effect if a message is not being played. The ?KA command returns KA.

EKR		n

Enable voice keyer record events. The value of n is 1 to enable the events and 0 to disable the events. See the Unsolicited Voice Keyer Events section for more details.

EKP		n

Enable voice keyer play events. The value of n is 1 to enable the events and 0 to disable the events. See the Unsolicited Voice Keyer Events section for more details.

Unsolicited Voice Keyer Events

The voice keyer can send unsolicited events if they are enabled.

The events currently defined are:

Event   Item   Value

KR   n

 The voice keyer has started or stopped recording. The value of n is the message number or 0 if the voice keyer has stopped recording.

KP   n

 The keyer has started or stopped playing. The value of n is the message number or 0 if the keyer has stopped playing.

Voice Keyer command examples

KR1⏎

 Start recording message 1.

KS⏎

 Finish recording the message being recorded.

KP1⏎

 Start playing message 1.

KA⏎

 Abort the currently playing message.

?KR⏎

 Returns the number of the message currently being recorded. If message 1 is being recorded the device will respond KR1⏎. If no message is being recorded the device will respond KR0⏎. If the SO2R device does not have a voice keyer it will respond ?KR⏎.

EKP1⏎

 Enable voice keyer playback events. The device replies with an event that contains the current message being played, or 0 if no message is being played. If message 2 is being played it would reply $KP2⏎. If the SO2R device does not have a voice keyer there will be no response.

Communications over Other Channels

The OTRSP protocol was originally designed to be used with serial ports with RTS and DTR capabilities. It can be used with other interfaces such as named pipes and TCP/IP ports. Other

interfaces do not provide modem signals, therefore an additional command is needed to control PTT. This command is only supported when RTS is not available.

Command	Item	Value
X		n

> Turns PTT on or off. PTT is turned on if the n is 1, and off if it is 0.

Command examples

X1⏎

> Turn on PTT.

?X⏎

> Returns X1⏎ if PTT has been turned on using the X command and X0⏎ otherwise. If the SO2R device is not supporting this command it will return ?X⏎.

Some communications mechanisms are message oriented. For example, Windows named pipes are usually used in message mode. There is no requirement that commands have any correspondence with messages. It is recommended that an SO2R device using such a mechanism accept messages of at least 40 characters and that it send one character per message. For TCP/IP messages sent and received can be any reasonable length.

Multi-Radio and Multi-Operator Extensions

The OTRSP protocol handles two radios at a time and provide for a single stereo output. This optional extension allow control of additional radios and outputs. Up to 36 radios and 36 stereo outputs can be supported.

Although radios may use the numbers $0 – 9$, for interoperability with logging program implementations which do not use extended commands, the first radio should be radio 1 and the second should be radio 2.

Extensions to supported commands

These are the commands which have extended values. Their use is backwards compatible with non-extended implementations:

Command	Item	Value
TX		n

> Set which radio has transmit focus. The value of n can be any numeric digit or upper case alphabetic character ($0 – 9$ or $A – Z$). The ?TX command returns the actual radio which has transmit focus, which may be different than what was set by the previous TX command if

the device selection has been overridden, for example by a front panel switch on the SO2R device, or if the specified TX device does not exist.

MAP   x   n

Tells the SO2R device to map a physical radio to one of the radios specified in the TX, RX, and PH commands. The item is the radio number. The value of x can be any numeric digit or upper case alphabetic character (0 – 9 or A – Z). The value of n is a one to three digit number corresponding to the physical radio.

This command is intended for SO2R devices which support more than two radios. It is not needed for an SO2R device which supports only two radios.

If this command is not implemented it is assumed that radio 1 specified in the TX, RX, and PH commands is always physical radio 1, radio 2 is always physical radio 2 etc.

Supported Multi-Radio and Multi-Operator commands

These are the commands which have extended values. Their use is backwards compatible with non-extended implementations:

Command   Item   Value

PH   x   n

Set which radios are connected to a pair of headphones. The item is the headphones. The value of x can be any numeric digit or upper case alphabetic character (0 – 9 or A – Z). The value of n is a list of which radio outputs are connected to the headphones.

Each radio has two outputs, left and right. These outputs could be connected to the left or right headphones channel.

The string contains the radios for the left headphones channel, a dot, and the radios for the right channel. A radio's left or right channel is specified by using the left or right brackets before the radio number. For example [1 is radio one left channel and]2 is radio 2 right channel. A bracket effects all radios specified after it until the next bracket is seen. For example [12 specifies the left channels of radio 1 and radio 2. If a bracket is not specified before the first radio a left bracket is assumed.

If the device is not capable of connecting multiple radio channels to a headphone channel the first specified radio channel will be used.

If the specified radio channel cannot be connected to the specified device the result is device-dependent. It may connect a different channel or make no connection.

EPH   x   n

Enables PH events. The item is the headphones. The value of x can be any numeric digit or upper case alphabetic character (0 – 9 or A – Z). The value of n is 1 to enable the events and 0 to disable the events. See the Unsolicited Events section for more details.

Command examples

PH11.2↵

Headphones 1 will listen to radio 1 left channel in the left ear and radio 2 left channel in the right ear.

PH11.]1↵

Headphones 1 will listen to radio 1 left channel in the left ear and radio 1 right channel in the right ear.

PH113.2]3↵

Headphones 1 will listen to radio 1 left channel and radio 3 left channel in the left ear and radio 2 left channel and radio 3 right channel in the right ear.

EPH21↵

Enable event for headphones 2. The device replies with an event that contains the current value of headphones 2. If the SO2R device does not have headphones 2 there will be no response.

Compatibility

If radios other than 1 and 2 are used for TX the ?TX command and TX events will show transmitters other than 1 or 2. This could be a problem if a logging program which does not implement multi-radio extensions runs after a logging program that does.

The RX command always operates on the same channels as PH1. Headphones 1 can be changed using either command.

The ?RX query and RX events may be incapable of showing the results of a PH command. If the PH command is used then the ?RX command may show information which is completely incorrect. PH commands may not trigger RX events. If a PH command connects receivers to headphones 1 in a way that matches an RX command then the result of an RX query should be correct. If the RX command is used then the ?PH1 command must correctly show the selected radio channels.