

Sierra Radio Systems Station Controller

USB-C Console Port Reference Manual



sierraradio.net

Version 1

Overview

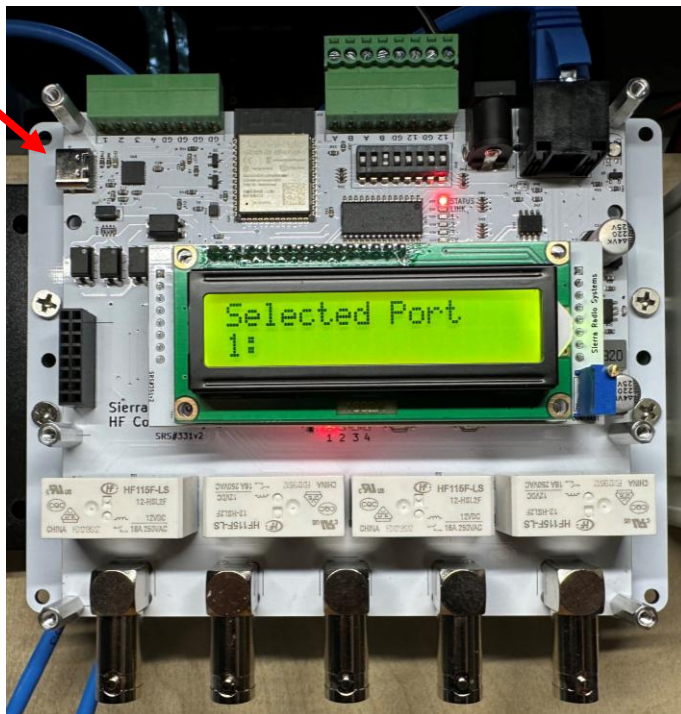
All SRS Station Console Modules feature a USB-C port. This port can be used for three purposes:

- Configure operating parameters for that specific module
- Download new versions of firmware
- Connect to a computer for using the module directly rather than over the DCN.

This manual provides information about how the console port works and an introduction to the DCN command structure. The information in this manual generally applies to all control modules.

USB-C Connection

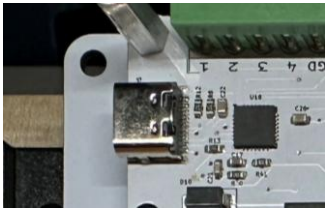
USB-C
Console Port



The picture above shows the location of the USB-C console port on the CX-1 coax relay module.

USB-C Console Port

The USB port is not normally used in typical operation of the system. It's primary purpose is to configure the operating parameters of the coax relay board and flash new versions of firmware. The console port that allows you to communicate with the board from a dumb terminal program like Putty. You can see the various steps that get executed as the module boots up, send it configuration commands, and other uses. Set your dumb terminal for 9600 baud, 8N1 and check out what the board is doing on boot or configure parameters.



If you are using a Windows computer we recommend the free dumb terminal program called Putty. You can get putty here:

For Putty go to <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

If you are using a linux computer, the popular terminal program is called minicom.

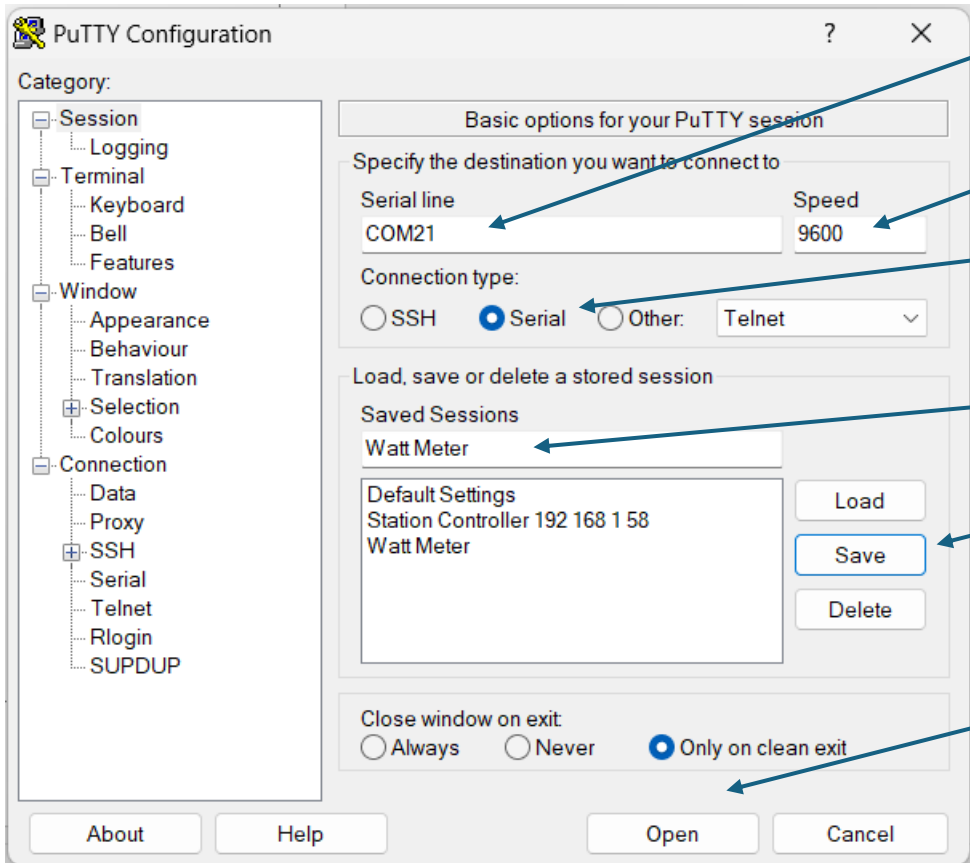
The USB port is not normally used in typical operation of the system. It's primary purpose is to configure the operating

When you plug your USB-C cable into the watt meter module and PC, a new serial port will be added to your system. On Windows you can determine the COM port number by looking at the "Computer Management" utility. At the Windows search bar, type "computer" and you should see Windows pop up an option to select Computer Management. Launch this program. Click on "Device Manager" then click on "Ports". All the available serial ports will be shown. The watt meter will present itself as a SiLabs serial device. Take note of the COM port number. We will need that for the Putty terminal program.

Launch Putty, configure the serial port parameters and "open" a connection to the serial port.

- You must select "Serial" as your connection type.
- Enter COMxx on the host name field where xx is the port number you noted in the Computer Management utility. For example "com21".
- Set the baud rate to 9600
- Enter a name for this connection in the "Saved Sessions" field. This can be any text like Watt Meter.
- Then click on the "open" button and you will be connected to the watt meter.

Configuring the PuTTY session

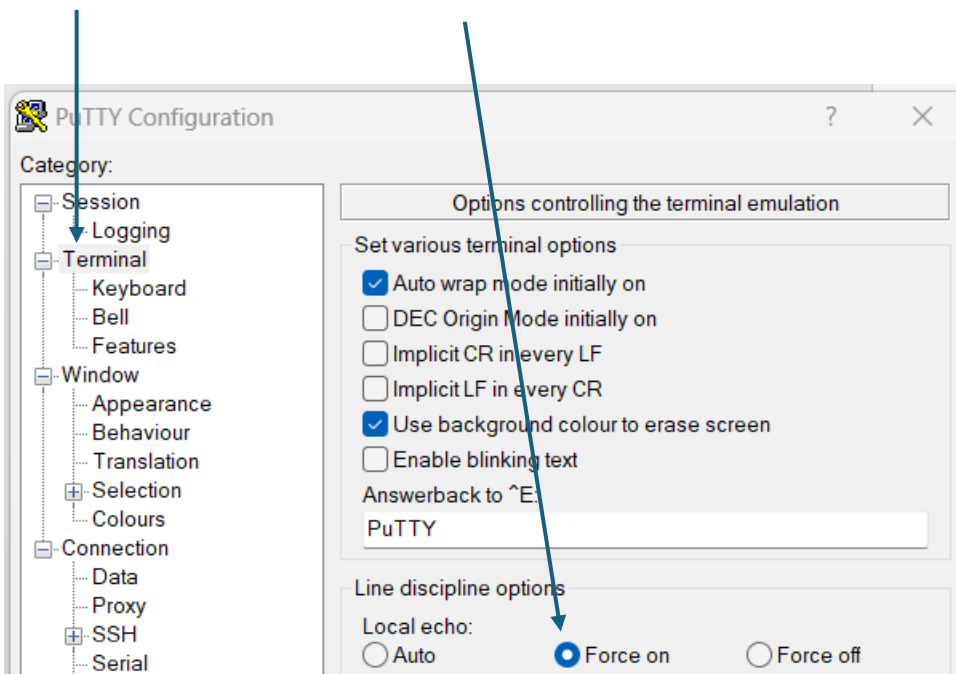


The screenshot shows the PuTTY Configuration dialog box with the following settings and annotations:

- Category:** Session
- Basic options for your PuTTY session:**
 - Specify the destination you want to connect to:
 - Serial line: COM21 (Annotation: Enter the COM port number.)
 - Speed: 9600 (Annotation: Enter 9600)
 - Connection type: Serial (Annotation: Select Serial)
- Load, save or delete a stored session:**
 - Saved Sessions: Watt Meter (Annotation: Enter connection name)
 - Buttons: Load, Save (Annotation: Click save), Delete
- Close window on exit:** Only on clean exit (Annotation: Click open)

Buttons at the bottom: About, Help, Open, Cancel.

Select the “Terminal” menu. Turn on local echo.



The screenshot shows the PuTTY Configuration dialog box with the following settings and annotations:

- Category:** Terminal
- Options controlling the terminal emulation:**
 - Set various terminal options:
 - Auto wrap mode initially on
 - DEC Origin Mode initially on
 - Implicit CR in every LF
 - Implicit LF in every CR
 - Use background colour to erase screen
 - Enable blinking text
 - Answerback to ^E: PuTTY
 - Line discipline options:
 - Local echo: Force on (Annotation: Turn on local echo.)
 - Auto
 - Force off

Annotations: A blue arrow points to the "Terminal" category in the left sidebar, and another blue arrow points to the "Force on" radio button under "Local echo".

Using the Console Port

Now that you have established a connection with the console port, you can enter commands and get information about the state of the control module. The commands you use are the same as the commands that the Node Red dashboard uses to communicate with the module.

The DCN supports two message (or “packet”) formats. A normal “addressed” message and a “broadcast” message.

The addressed format is used to direct commands to a specific device and determine where commands are coming from. The broadcast format is decoded by all devices and executed if that device supports the command in the message.

The broadcast message format is very simple. The format is // followed by the command or “payload”

For example to execute the command on the GPIO module to turn relay 2 on would be RY2,1

To broadcast this message to all devices you would send

//RY2,1

You would rarely want to send a broadcast message to all devices unless you are telling them all to reboot or turn off all relays. To direct a command to a specific device, you would use the addressed message format.

The addressed message format from the main control computer to a GPIO module address 4 to turn on relay 2 you would send:

/0004:RY2,1:XX

/	Start of message indicator
00	From address
04	To address
:	Delimiter
RY2,1	Command payload
:	Delimiter
XX	CRC check value. XX is placeholder when this mode is off.

Using the Console Port