

P/N 35125 Rev 01

Monitor & Control a BUC / XCVR using a phone and a USB-to-Serial cable

This addendum describes how to monitor and control an AnaCom, Inc. BUC or transceiver, (XCVR,) using an Android device, such as a phone or tablet, over a Serial port connection. An ASCII terminal program will be used as the monitor and control application.

Testing for writing this document was done using a Samsung Galaxy S6. An Apple iPhone with the appropriate ASCII Terminal App and USB-to_Serial cabling ought to work as well, but that has not been tested by the author of this document.

Important note: doing this with a Protection Switch, (PS,) is also possible if not using RS-485, but the ODUs that are connected to, or behind, the PS will not be accessible. A connection to those devices would have to be done directly.

The Connection from phone to ODU

First, we need a USB-to-Serial adapter. We recommend using one that can auto start the serial FTDI driver when connected to an Android system. For example, see:

<https://www.startech.com/Cards-Adapters/Serial-Cards-Adapters/1-Port-USB-Serial-RS232-Adapter-Cable~ICUSB2321F>.

We might also need a standard USB to micro-USB adapter to plug into the Android device at the end of our cable assembly. For example, see:

https://smile.amazon.com/gp/product/B01C6032G0/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1.

Second, cabling considerations. For XCVRs, this solution is easier to get working because we can use the Serial port 6-pin connector that all XCVRs have. The 6-pin cable for a XCVR, P/N 30720, is recommended because this port is ready for use with any ASCII terminal. This port will drop automatically to 1200 baud when presented with commands at that speed.

If using the circular 18-pin connector, required on BUCs since they do not have the 6-pin port, then we use Serial Cable P/N 31336. A bit of configuration to the ODU/BUC might be desired on the bench first however, in order for this to work well because the Serial port available in the 18-pin connector, by default, does not automatically drop to 1200 baud when presented with data at that speed. The user might have to spend some time trying different bauds, such as 9600, 19200, etc.

Drawings for the Anacom, Inc. Serial cables

For cable, P/N 30720, see:

<http://anacominc.com/wp-content/uploads/2018/10/Cable-Serial-ODU-9-Pin-3072002.pdf>

For cable, P/N 31336, see:

<http://anacominc.com/wp-content/uploads/2018/10/Cable-ODU-SERIAL-TO-18-PIN-3133602.pdf>

ASCII Terminal Application

Probably, any “App” will work but we have tested, and have been using, an App called Serial USB Terminal, see at Google Play:

https://play.google.com/store/apps/details?id=de.kai_morich.serial_usb_terminal&hl=en

There are settings that need to be set in order for this connection to work:

Serial Settings:

1. Baud rate: 1200, or whatever the ODU Serial port is currently known to be using
2. Font size: this seems irrelevant, but experience has taught us to choose 10
3. Font style: select Monospace

Receive Settings:

1. Newline: CR+LF
2. Display mode: Text
3. Buffer size: unlimited

Send Settings:

1. Newline: CR
2. Edit mode: Text
3. Local echo: ON - important; more on this later

System Settings:

Keep screen on when connected: ON - useful, but not required

ODU Serial Port Configuration

There are a few Serial port settings that should be made on the ODU as well. These can be done using Supervisor.

1. Turn character echoing off. We have found that echoing every typed character back to the client device results in data loss. If connected to the same Serial port that will be used later in the ASCII terminal connection scheme envisioned here, then enter this command:

ECHO OFF

If using a different connection, then we must specify the Serial port we are configuring.

Example, to configure the 6-pin port, or the 18-pin port on a PS, RS-232 connection planned:

DTE1 ECHO OFF

Example, to configure the 18-pin port on an ODU:

DTE0 ECHO OFF

Note: in our client ASCII Terminal App, we turned on Local echo. This allows us to see the commands we are sending to the ODU.

Additionally, if planning to use the 18-pin connector on an ODU, we should enable the auto-baud feature, which allows the ODU serial port to drop to 1200 baud when receiving data that it does not recognize correctly:

DTE0 AUTOBAUD ON

This will enable us to connect easily, without having to guess the correct baud rate.

Note: DTE0 is used for configuring the 18-pin connector, and DTE1 for configuring the 6-pin Serial port connector.

Note: after changing any configuration of any parameters on an ODU, we recommend saving the changes to FLASH storage:

SAVE

We are now ready to disconnect Supervisor, connect our client device, serial cable, and ODU together, and proceed with Monitor & Control over Serial.

Note: if using a connection to the ODU/BUC 18-pin connector, sending a few CR characters might be necessary at first to get the device to respond. This is because that connector is normally intended to operate in a data packet mode and won't initially respond to commands from an ASCII terminal. After having switched, it should operate as expected.

Some Additional Suggestions...

Power

The phone will not be powered via the Serial cable connection, so leaving the screen on for a long time while running a USB Serial Terminal App is not recommended.

Display Auto-Rotation

We recommend enabling auto-rotation of the screen. It is easier to type commands in portrait mode, but some of the output displays will wrap, and rotating the phone to landscape mode to read output will be helpful.

Macros

An App, such as Serial USB Terminal for Android, allow the use of macros which are easily configured. This allows for some frequently used commands to be pre-programmed, such **LD** for the basic ASCII output display, and **LOG ALARMS *** for a listing of recent active alarms, and **SAVE**, for saving updated operating parameters to FLASH storage.

Periodic Output of the Default ASCII Display

The parameter, UTIMER, can be set in the ODU to configure the device to periodically output the default ASCII display when operating in ASCII Terminal mode. At 1200 baud, this will be every 30 seconds. This allows the user to avoid having to send a command to do this repeatedly. Example:

DTE0 UTIMER ON

Note: UTIMER should already be ON by default for the 6-pin connector on a XCVR.