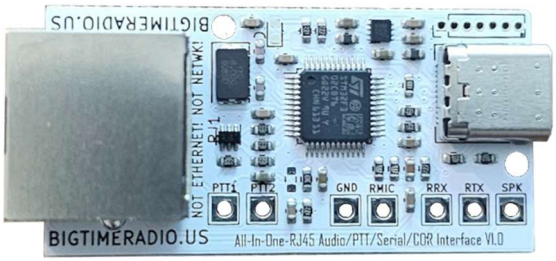
AIOR USER MANUAL

Revision 1.1 - https://bigtimeradio.us



MANUAL CHANGELOG		
Date	**Author**	**Change**
5/28/2026	Lucas Elliott	Initial Manual creation
5/28/2026	Lucas Elliott	Added Firmware 1.3 commands

Contents

Introduction

History

The AIOR is the third generation of the AIO series. The AIOC, originally designed by Simon Küppers, was the first generation, and remains the preferred option for portables to this day. The AIOB was our first adaptation of the AIOC, and primarily aimed to provide solutions for mobile HF/VHF/UHF/ETC radios, by exposing the connection points via solder pads.

The AIOR expands on the AIOB by adding an RJ45 interface. This allow quick switching between radios and easier cable integration.

Also added with the AIOR, was Big Time Radio's custom firmware for the entire suite, offering advanced config, easy flashing, and a 1200 baud packet tnc/repeater firmware, primarily designed for APRS.

Use Cases

The AIOR can be used for pretty much any digital mode. It works on Windows, Mac, Linux, and Android, and can work with FLDigi, Winlink, WSJTX, CWRig, Vara, and APRS software like PinPoint APRS, Direwolf, etc.

The AIOR can be used like a digirig for quick and reliable DTR based PTT, or it can be configured to use an internal automatic VOX circuit like the Signalink.

Additionally, the AIOR can frequently be used to program radios without having to mess with Chinese drivers.

Connecting to your radio

The AIOR is compatible with all radios that have wired PTT and have an unbalanced audio input/output. Many professional radios will have both types of IO, and many will have accessory ports such as a DB15/DB25 port on the back of the radio that the AIOR can very easily interface with using a DB15 or DB25 to RJ45 adapter.

Additionally, users might find RJ45 breakout / pin swap boards handy to make cables without soldering or crimping.

In many cases, the easiest path will be to crimp an RJ45 connector to some CAT5 with custom wiring to match up the AIOR's pinout with the radio's pinout, usually available via a service manual.

The pinout is listed on the bottom of the board for easy reference (see figure 1) and is from the perspective of the radio.

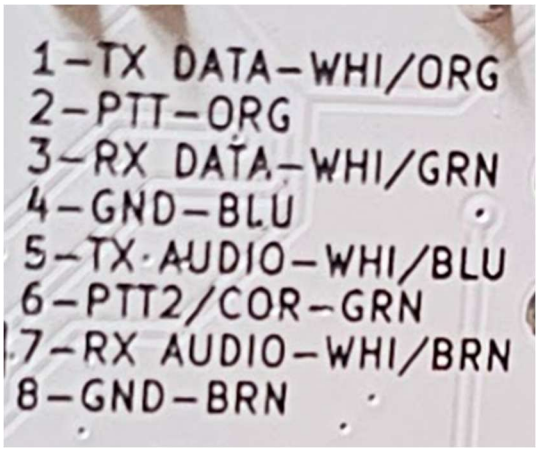


Figure 1

1 – TX (Data from the radio to the AIOR)
2 – PTT
3 – RX DATA (Data from the AIOR to the radio)
4- GND
5 – TX
6 – PTT2
7 – RX AUDIO
8 – GND

Note that the AIOR also includes a handy reference for the wire colors, assuming a standard T56B wired CAT5 cable (the standard found in most places). You'll also note that the TX audio and rx audio share a twist with ground, providing better audio signal integrity.

By referencing the manual or service manual on your radio, you can determine how to adapt the wires to fit whatever connectors are on your radio. On some rigs, you might need to break out one of the ground wires and the RX audio to plug into the rig's speaker jack if it does not expose RX audio anywhere else. You may find it easier to solder these onto the solder pads that remain from the AIOB design and use the RJ45 for the rest of the connections (figure 2)

Figure 2

Flashing Firmware

Flashing on Windows

Windows requires the installation of the libusb drivers STM32 chips use. If you've ever worked with RTL-SDR dongles, it's a similar process.

If you are flashing a **newly fabbed board** for the first time (the board currently has no firmware), short the jumper using the BTR-JP-1 jumper pin, and you will need to run Zadig (or a similar libusb driver installer) on the "STM32 Bootloader" device to give it the driver, then you will be able to flash the firmware while the jumper is installed. Then, you'll need to run Zadig a **second** time to install the driver for the AIOR itself (since it shows up with a different ID then a generic STM32 bootloader).

If this board has already been flashed, you will only need to install Zadig once.

Once the driver is ready to go, it's as simple as going to https://flash.bigtimeradio.us and clicking "flash."

Flashing on Linux

On Linux kernel based systems, no libusb driver installation should be necessary for most distros. Instead, you can navigate right to https://flash.bigtimeradio.us and flash there.

If you prefer to flash locally, you can simply install dfu-util using your distro's package manager, such as ***sudo apt install dfu-util*** on Debian/Ubuntu.

Some Motes on Firmware

The AIOR remains backwards compatible with Simon's firmware for hardware 1.0, though most users will likely find more functionality in Big Time Radio's firmware.

There are two base firmware revisions, namely the standard revision, and the KISS TNC revision. The KISS TNC revision acts as (you guessed it) a KISS TNC and can also function as a standalone digipeater and APRS messenger.

Note that for the AIOR, you should keep the hardware version to 1.0 in the settings.

Configuring the AIOR

Unlike the classic firmware, the Big Time Radio firmware supports configuring the settings on the AIOR directly using a serial interface, or using the web flasher https://flash.bigtimeradio.us (see figure 3). REMEMBER: **YOU WILL NEED TO INSTALL LIBUSB DRIVERS USING ZADIG ON WINDOWS.**

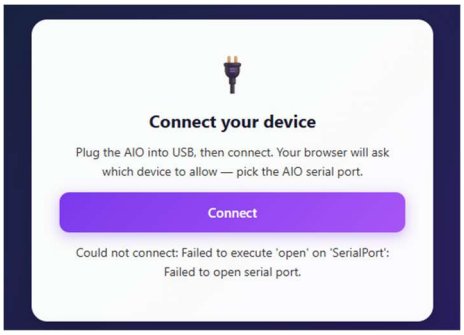


Figure 3

Serial Config Settings

To access the settings, you can simply connect using a standard serial tool like Putty or MobaXterm with the following settings:

Data bits: 8
Stop Bits: 1
Parity: None
Flow Control: **None**

Then, entering the menu depends on the firmware revision you are using.

Menu Access for Standard Firmware

The standard firmware uses the serial connection for USB audio streams, so it typically won't take commands. As such, the following procedure is used to access its menu:

1. Connect using Putty
2. Wait 1 second
3. Enter "+++" and press Enter (see figure 3)

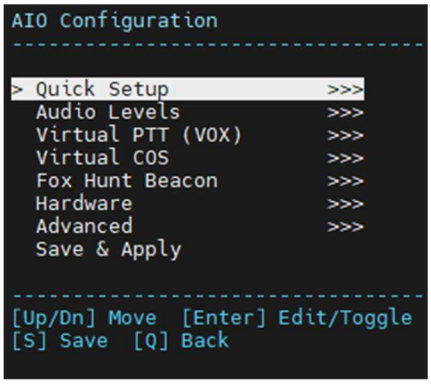


Figure 4

Menu Access for KISS TNC Firmware

The KISS interface uses binary based serial commands to speak the KISS language, so it can accept a simpler procedure:

1. Connect using putty
2. Wait 1 second
3. Enter “menu” and press Enter (see figure 4)

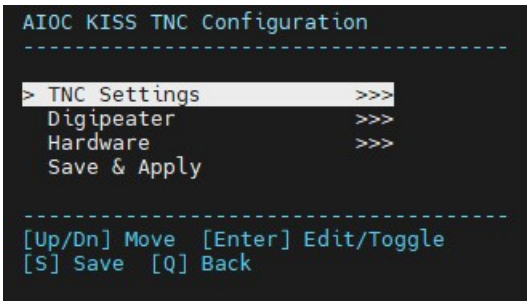


Figure 5

Direct Config Commands

You can also send direct set commands over serial using “---” before the command. This is useful for a quick fix, and is also useful for scripting.

Standard Firmware Commands:

Mode presets

- ---MODE STANDARD — CM108 PTT + serial DTR/RTS PTT
- ---MODE AUTO — Automatic / VOX (virtual PTT)
- ---MODE ALLSTAR-HCOS — AllStarLink, hardware COS
- ---MODE ALLSTAR-VCOS — AllStarLink, virtual (audio) COS
- ---MODE ALLSTAR — alias for ALLSTAR-VCOS
- ---MODE FOX — Fox-hunt beacon mode

Identity

- ---HW v1.0|v1.2 — set hardware revision

Audio / PTT tuning

- ---SET RXGAIN <0-4> — 0=1x, 1=2x, 2=4x, 3=8x, 4=16x (v1.2 hardware only)
- ---SET TXBOOST <0|1> — 0=mic level, 1=line level (v1.2 hardware only)
- ---SET VPTT <0-65535> — VOX (virtual PTT) threshold

- ---SET VPTTTIME <ms> — VOX hang/timeout in ms
- ---SET VCOS <0-65535> — COS (virtual COS) threshold
- ---SET VCOSTIME <ms> — COS timeout in ms

System

- ---STATUS — report current configuration (see below)
- ---SAVE — save settings to flash
- ---DFU — reboot into the bootloader for a firmware update
- ---EXIT — exit command mode

---STATUS reply:

OK FW=<ver> VAR=<AIOC|AIOB|AIOR> HW=<0|1> VID=<hex> PID=<hex> RXG=<0-4> TXB=<0|1> VPTT=<n> VPTTT=<ms> VCOS=<n> VCOST=<ms> FOX=<interval>

TNC Firmware Commands

Identity

- ---HW v1.0|v1.2 — set hardware revision

APRS / digipeater

- ---CALL <CALL-SSID> — set station callsign, e.g. ---CALL W1ABC-7
- ---DIGI ON|OFF — enable/disable digipeating (WIDE1-1 + WIDE2-n)
- ---POS <lat> <lon> — set beacon position in APRS format, e.g. ---POS 4245.53N 07151.43W
- Tuning
- ---SET TXDELAY <1-255> — TX delay in ×10 ms units (30 = 300 ms)
- ---SET BEACON <seconds> — position-beacon interval; 0 = off

System / diagnostics

- ---STATUS — report current configuration (see below)
- ---SAVE — save settings to flash
- ---DFU — reboot into the bootloader for a firmware update
- ---TX — transmit a test APRS packet
- ---DIAG — dump modem/decoder diagnostics

---STATUS reply:

OK FW=<ver> VAR=<AIOC|AIOB|AIOR> CALL=<CALL-SSID> DIGI=<0|1> W1=<0|1> W2=<0|1> TXDLY=<n> BCN=<seconds> HW=<0|1>

Accessing APRS Messenger in TNC Firmware

Type “Messages” in the cli and it will open the messaging menu. This is in beta and untested so far.

Troubleshooting

1. No RX or TX Audio
    a. Ensure the hardware version matches. At the time of writing, all AIOB/AIOR board should be set to hardware version 1.0 in the config. AIOC will depend on the hardware version written on the back of the board
2. Can’t flash the board on Windows
    a. Run Zadig again and select the current device. Major changes in settings such as USB ID changes, as well as different variations will require installing the driver for a new ID.