

Jan.28, 2022

**ARL2300LOCAL: Linux OS receiver control software for AOR AR5700D, and AR2300 / AR5001D / AR6000 receivers equipped with the IQ5001 option.**

Following instructions are based on version 4.2.9. ". Many instructions have changed since 4.0.10, please read this manual again thoroughly.

Improvements over the previous version 4.0.10:

- I/Q recording feature addition
- Detached and resizable spectrum display
- Hovering over signal peak does display the frequency

**Tested and confirmed to work on:**

Raspberry Pi O.S on RPI4/RPI400

Linux MINT 20 on Intel Celeron PC

Ubuntu 20 on Intel I5 PC.

It is likely to work on any recent version of DEBIAN, UBUNTU and RASPBIAN.

**Confirmed not to work on:**

Raspberry Pi O.S on Raspberry Pi zero,1/2/3 series. Captured I/Q stream data amount via USB is too high for the weaker Raspberry Pi models to handle.

**Required Linux knowledge:** Medium to advanced

You must be comfortable with Linux operating systems, know how to install software packages, manage folder permissions and compile data.

**<Supplied files>**

- \* ARL2300LOCAL\_installation\_instructions.pdf --- This file
- \* ARL2300LOCALv4.2.9.jar--- The software executable based on JAVA.
- \* ARL2300\_client\_software\_guide.pdf --- Software user guide (originally for the ARL2300 Ethernet Controller client software, but instructions are almost identical except for login)
- \* 98-iqfk.rules --- File for udev
- \* Makefile --- Used to compile ariq\_rcv.cc
- \* ariq\_rcv.cc --- The source code of the recording program
- \* iqwr2.sh --- Firmware download executable file
- \* fx2fw.hex --- Firmware file

**<Features>**

- \* Local receiver control via USB, memory management, scan, search, basic spectrum display, audio recording to SD. Received audio via receiver audio outputs only. Simultaneous multi-receiver control on the same PC!
- \* I/Q recording (playback is via GQRX or Gnuradio only). Please note that SD card WRITE speeds are too slow, therefore captured I/Q data must be saved to an external USB memory or drive, connected via USB 3.0. Captures to SD cards will result in broken I/Q data.
- \* The software can be used and shared freely, however the source code is not public.
- \* The software is supplied "as is" without any obligation of support.

### <Preparing for receiver control>

1.

Install following packages with "sudo apt install":

openjdk-11-jre librtx-java

cycfx2prog          Cypress EZ-USB FX2 (LP) programmer

usbutils            Linux USB utilities

libusb-1.0-0        Userspace USB programming library

libusb-1.0-0-dev   Required to compile files

clang                Compiler

2.

Extract the downloaded file "arl2300localv4.2.9.tar.gz". Let's assume it is located in the "Downloads" directory.

```
$ cd Downloads
```

```
$ tar xf arl2300localv4.2.9.tar.gz
```

This will create the "arl2300localv.4.2.9" in the Downloads directory, with all the required files inside.

3.

Copy 98-iqfk.rules to /etc/udev/rules.d/ as follows:

```
$ cd
```

```
$ cd Downloads/arl2300localv4.2.9
```

```
$ sudo cp 98-iqfk.rules /etc/udev/rules.d
```

Then execute the following command in terminal:

```
$ sudo udevadm control -R
```

4.

ariq\_rcv.cc needs to be compiled for the particular PC hardware/operating system you are running the control software on.

Go to the directory where you extracted the supplied file. If it's into "Downloads", that would be:

```
$ cd
```

```
$ cd Downloads/arl2300localv4.2.9
```

Execute the following commands in terminal:

```
$ make clean
```

```
$ make
```

The resulting "ariq\_rcv" is now ready to use.

5.

Grant the execute permission to the file "iqwr2.sh":

```
$ cd
```

```
$ cd Downloads/arl2300localv4.2.9
```

```
$ chmod 0755 iqwr2.sh
```

## 6.

Set the login user name in vigr:

```
$ sudo vigr
```

(You can find out your login username by opening a command window. It will indicate for example "pi@raspberrypi". In this case, "pi" is the user name.)

In the vigr window, find the "dialout" line and make sure your login username is set. For example: **dialout:x:20:pi** ("pi" would be the login user name)

Save changes with CTRL+O and ENTER.

Close vigr with CTRL+X.

Set the login user name in vigr -s:

```
$ sudo vigr -s
```

In the vigr -s window, find the "dialout" line and make sure your login username is set. For example: **dialout:\*::pi** ("pi" would be the login user name)

Save changes with CTRL+O and ENTER.

Close vigr -s with CTRL+X.

## 7.

Connect the receiver to the PC via two USB cables. One for receiver control and one for the I/Q stream capture.

Prefer USB2.0 ports as 3.0 sometimes might create problems.

Disconnect other USB peripherals (mouse & keyboard can stay connected) to avoid USB number allocation complications.

Now you can switch on the receiver.

## 8.

Check if the USB connection for receiver control is correctly detected:

```
$ ls -l /dev | egrep ttyUSB0
```

If the reply is terminated by "ttyUSB0" as below, then it is detected correctly.

```
crw-rw---- 1 root dialout 188, <date & time stamp> ttyUSB0
```

### <Start the receiver control software>

## 9.

Go to the directory where you extracted all the supplied files. If it's into "Downloads", that would be:

```
$ cd
```

```
$ cd Downloads/arl2300localv4.2.9
```

Execute the program:

```
$ java -jar ARL2300LOCALv4.2.9.jar
```

(Some operating systems such as Linux MINT might require sudo)

Wait a few seconds until the program is launched.

If you can see "/dev/ttyUSB0" in the PORT section top left of the program window, then everything is OK and you can click CONNECT.

(Leave the command window open during use.)

For simultaneous multi-receiver control, launch another session of ARL2300LOCALv4.2.9.jar. in the PORT section top left of the program window, there should be a drop-down menu where you can select the USB port corresponding to the other receiver(s).

10.

**<Known limitations>**

- The spectrum display is basic as it is based on serial data, not on the I/Q stream.
- The GSSI filtering function for the TETRA mode on AR5700D is not supported.
- Audio recording to receiver SD card only.
- I/Q playback in GQRX: The displayed center frequency is zero, which is incorrect. However, you can change it manually via the digits.
- I/Q playback in GQRX: The receive frequency cannot be changed via the digits, only by dragging the vertical bar on the spectrum with your mouse.

11.

**<Q&A>**

Q.

What is the "device string" to use in GQRX?

A.

Device= Other

file=(path to your iqdatI.Q.bin file),rate=1.125e6,repeat=true,throttle=true

Q.

How much data is the AOR I/Q file?

A.

The sampling rate is 1.125MSps and the data length per sample is 64 bits. The transfer rate is 9 Mbytes / sec (72 Mbit / sec). In other words, the I and Q files will be 270MB per minute each, and the combined I/Q file 540MB per minute.

Q.

Can the I/Q data be saved on a Raspberry Pi?

A.

Yes. However, only on the RPI400 and RPI 4B as the I/Q data is too big for weaker Raspberry Pi models to handle. Also, as SD card WRITE speeds are too slow, I/Q data must be saved to an external USB memory or drive, connected via USB 3.0.

Q.

I recorded I/Q from H.F signals, but when played back in GQRX or GNU Radio, there seem to be no signal. Why is that?

A.

Remember that as described in the AR-IQ3 software manual, when using the I/Q stream and receiving frequencies below 25MHz, the HF antenna should be physically connected to the antenna socket #1, even though it's usually #2 when the receiver is used "stand-alone". This is due to the fact that the receiver's internal antenna wiring is different when used "stand-alone" vs controlled by software. It also means that for pure I/Q capture you won't be able to hear via the receiver what you are receiving. A workaround is to connect an antenna to both socket #1 and socket#2, during the capture.