

# ICS-CTRL 1X REPEATER AND SVXLINK SOFTWARE

## Additional information for setup

Not Copyrighted.  
Not Warranted.  
Use at your own risk.  
Gordon Gibby KX4Z

### INTRODUCTION

The ICS-CTRL.com 1X (single repeater) raspberry-pi based repeater controller is a very inexpensive (\$59, exclusive of the Raspberry pi) way to control a FM repeater, making the required identifications and even including the ability to make status announcements of voltages, temperatures and such, and respond to some commands. I'm early in my knowledge of this product but it appears it can potentially provide Echolink connectivity also. Since I have gotten comfortable with Raspberry Pi's (Particularly the Version 3 (model b) ) for about \$35 from Amazon, this is an attractive way to control an emergency (or any other) repeater.

However, although ICS-CTRL tries to make setup of their product duck-soup-simple, I still ran into some "gotchas" and a bit of knowlgedge of Raspberry and of linux goes a long way to get this product working. The vendor was extremely helpful, with email after email great suggestions and offers of telephone help if needed --- so I have compiled here all the information that I gathered in hopes of saving both later users and the owner the troubles that I had!

I believe that I know understand how to use a couple of soundcaard usb-dongle type devices and some simple associated circuitry to use the openrepeater.com free software to duplicate the function of the ICS-CTRL product. His product is more I2C based and I had one raspberry that just didn't work with it --- but his product does work with the right Raspberry and it is a slick product once it is working!

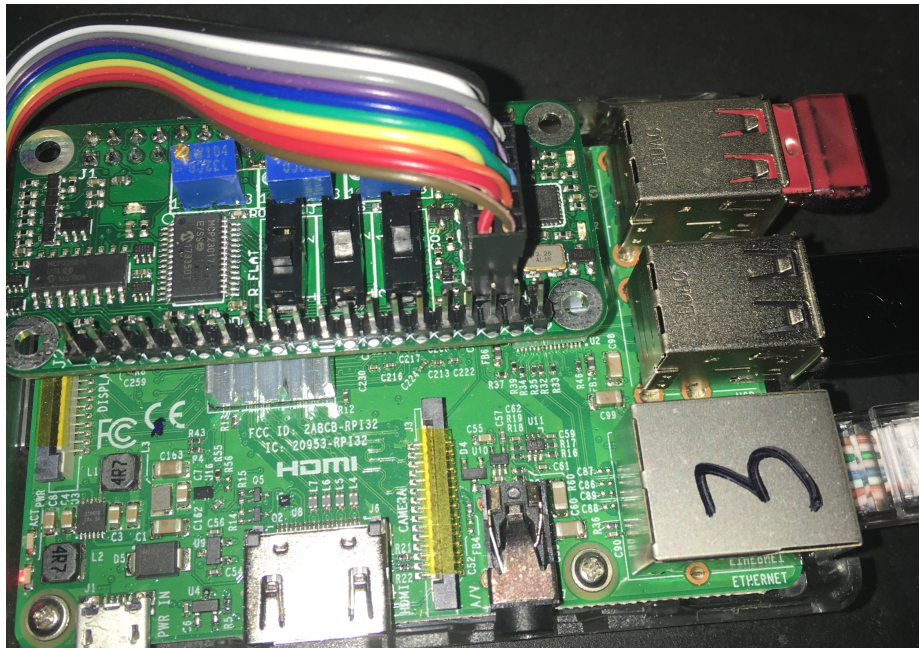
### Location of helpful software and documents:

Quickstart guide to the 1X ICS-CTRL controller	
Manual for the 1X ICS-CTRL controller	
ICS-CTRL version of svxlink software for Raspberry Pi Model 3B	<a href="http://www.ics-ctrl.com/content/pi_images/PI_REPEATER_2X_20180120_STRETCH.zip">http://www.ics-ctrl.com/content/pi_images/PI_REPEATER_2X_20180120_STRETCH.zip</a> Note user <b>pi</b> password <b>ICS-CTRL</b>

<p>ICS-CTRL version of svxlink software for Raspberry Pi Model 3B+ (introduced March 2018)</p>	<p><a href="http://www.ics-ctrl.com/content/pi_images/PI_REPEATER_2X_20180425_STRETCH.zip">http://www.ics-ctrl.com/content/pi_images/PI_REPEATER_2X_20180425_STRETCH.zip</a></p> <p>Note user <b>pi</b> password <b>ICS-CTRL</b></p>
<p>Open Repeater version of software (note that it <i>does not appear to be able to control the I2C-bus-based ICS-CTRL controller</i> -- but it will work with USB-based sound cards)</p>	<p>See: <a href="https://openrepeater.com/downloads">https://openrepeater.com/downloads</a></p> <p>Note: the open repeater group use the user root and the password is usually either OpenRepeater or openrepeater -- but it seems to be different from different access types. See this page: <a href="https://openrepeater.com/getting-started#5b80c3c50c3d5">https://openrepeater.com/getting-started#5b80c3c50c3d5</a></p>
<p>Manual for the configuration file <i>/etc/svxlink/svxlink.conf</i></p>	<p><a href="http://www.svxlink.org/doc/man/man5/svxlink.conf.5.html">http://www.svxlink.org/doc/man/man5/svxlink.conf.5.html</a></p>
<p>Video on hooking up cables</p>	<p><a href="https://youtu.be/uqAOG9JomGs">https://youtu.be/uqAOG9JomGs</a></p>
<p>Video on setting up input level</p>	<p><a href="https://youtu.be/aSA_0l625Fw">https://youtu.be/aSA_0l625Fw</a></p>
<p>Video on setting up output level</p>	<p><a href="https://youtu.be/nGCcUR05pZU">https://youtu.be/nGCcUR05pZU</a></p>
<p></p>	<p></p>

# HARDWARE

1. Be very careful with the 1X (1X means able to control ONE repeater input receiver-output transmitter system) ICS-CTRL on the Raspberry Pi Model 3B that you don't let the tiny ICS-CTRL circuitboard press down and short out against the conductive heatsinks below. A bit of black electrical insulation tape might be in order here or a simple standoff system.
2. To properly insert the ribbon cable from the male radio pins on the ICS-CTRL board to the DB-9 female connector, you must know where pin 1 is on the male pins, and you must know that the BROWN wire of the ribbon cable is the #1 wire. See photo below:



3. Until you are more familiar with the ICS-CTRL board, you might want to **take a fine tipped marker and mark the IN, OUT and GAIN potentiometers**. The GAIN potentiometer controls the amplification of the output amplifier (going to the transmitter microphone input) and works in conjunction with the (linear) OUT potentiometer, which is in a normal volume control configuration. This gives TREMENDOUS potential output variability in signal level!

The function of each potentiometers

Potentiometer (10 turn precision pot)	Clockwise Rotation Will
INPUT	Increase the sensitivity of the board to the signal from the receiver
OUTPUT	Increase the amplitude of the signal from the

	board to the transmitter
GAIN	DECREASE the gain (down to 1:1) of the output amplifier.

4. The ICS-CTRL configuration of svxlink comes set for COR/COS sensing of receiver input signal, relying on a signal from the receiver that it has detected a valid signal. This works well because even with the repeater user isn't yet speaking, the FM signal is constantly strong, constantly sending any CTCSS tones, and the receiver is easily able to detect the presence of the signal. Most uhf/vhf fm radios (other than handhelds) include some sort of a signal detection output. The ICS-CTRL board is able (by switch selection) to handle either active-high (high when signal detected) or active low. Handheld radios such as the Baofeng UV-5R don't come with such a signal. There are some methods to create one from them, but for your initial testing (if you are using one of these) you may wish to simply have two wires from ground and the COS input (pin 7 of the DB-9) and short them to simulate this signal.

5. DB-9 CONNECTOR PINOUT

- 1- CTCSS ENCODE OUTPUT**
- 2- CTCSS INPUT FROM RX RADIO**
- 3- PUSH TO TALK (OPEN COLLECTOR) (OUTPUT)**
- 4- AUDIO OUTPUT (To Transmitter mic input)**
- 5- AUDIO INPUT (From Receiver Audio output)**
- 6- GROUND**
- 7- COS INPUT FROM RX RADIO**
- 8- GROUND**
- 9- GROUND**

# SOFTWARE

1. The ICS-CTRL software includes the `i2cdetect` program, which is incredibly helpful in detecting that your board's sound card has been detected. See this page for more indepth information: <https://linux.die.net/man/8/i2cdetect>

Carefully note that the "2" is inbetween the i and the c.

## `i2cdetect 1`

will cause your system (after a warning message) to scan the first range and you should see TWO "UU" entries in the resulting table, indicating that your board's soundcard has been detected. (1X Model).

If you don't, something is awry and you should ask for help. Note that this program can confuse things, so reboot after using. (`sudo shutdown now`)

2. LOCALIZATION: If you are accessing your ICS-CTRL board through the SSH protocol, you may not have any keyboard-layout issues. However, if you are using a keyboard & monitor connected directly to the Raspberry Pi, you may have some characters/keys not properly working until you use `raspi-config` to set them --- and because of the vast array of choices, this can be problematic. The character set you may wish in the Unites States is: `en_US.UTF-8`

3. Editing the `svxlink.conf` file. This is located in `/etc/svxlink/` There are a multitude of text editors that can be employed. You will likely need to use `sudo vi` or `sudo nano` or similar to be able to edit the file (which is read-only for user pi). Be very careful obviously when editing this file. As noted in the Table at the beginning of this information, the terse manual explaining all the options is found at: <http://www.svxlink.org/doc/man/man5/svxlink.conf.5.html>

4. Identification: In the `svxlink.conf` configuration file, the times for "short" and "long" identification are given in MINUTES. You may wish to shorten them temporarily in order to listen more carefully. My callsign was not properly sent with the "/R" suffix until I added some SPACES after the callsign and forced their usage by placing quotation marks around the callsign in the configuration file. "NF4RC " worked for me.

5. Receiver squelch issues: This is in the [Rx1] section of the ini-type configuration file `svxlink.conf` There a a multitude of choices for receiver activtion -- the ICS-CTRL software comes configured for

```
SQL_DET=GPIO
```

expecting the COS signal from the receiver. Other choices are VOX, CTCSS, SERIAL, and several others (see the svxlink.conf manual cited above). The VOX choice can be made to work marginally with uncommenting the FILTER depth and THRESHOLD options below it.

6. AUDIO LEVELS & CONNECTIONS. The ICS-CTRL system uses the well-known alsamixer to select the proper signals and set their gains. You should probably run

```
sudo alsamixer
```

Then hit F5 to "show all settings" and be certain that CAPTURE is set to be "lin-in"  
Then Set all sound levels to 60% and exit using the ESCAPE key.

I prefer to them store these settings using

```
sudo alsactl store
```

and you may wish to check them with sudo alsamixer if there are questions.

7. Getting the software to run -- it is supposed to be started as a background service in the bootup sequence. In my case I couldn'd find it running (using top, or

```
top -d 0.25 | grep svxlink
```

The executable is located at

```
/usr/bin/svxlink
```

You can execute it manually by just typing that, but we want it to run forevermore....

Using my favorite editor I created the following file /home/pi/svxlinkscript:

```
# find svxlink process identifier
/usr/bin/pgrep svxlink
# if not running
if [ $? -ne 0 ];
then /usr/bin/svxlink
fi
```

You can test that this file will start up svxlink by using

```
bash /home/pi/svlinksript
```

Then using the command **crontab -e** (and selecting the nano editor, which is self-explanatory) I appended two lines to the crontab file, and let it save it as it pleased:

```
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 */12 * * * sudo /sbin/shutdown -r now >/dev/null
*/2 * * * * bash /home/pi/svlinksript >/dev/null
```

The first line causes the raspberry to reboot every 12 hours and the second line causes it to check for svxlink every two minutes and if not running, it uses the script above to start it up.