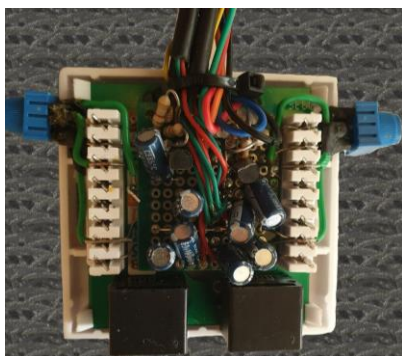




connection is of course best, but on Öland we have good experience of using mobile internet. SvXLink itself does not need a public IP address. But if you want to be able to connect to *EchoLink*, this is required. You may also need a public IP address for remote administration – where *ssh* terminal login is usually used. If you have the possibility to connect the repeater via *the amprnet*, it is of course preferable. It should also be possible to connect to the *amprnet* with a VPN and obtain a public IP address that way.

### Selection of interface to use

You can of course do as we first did on Öland, i.e., we let our local electronics expert SM7GMD manufacture an interface in all its simplicity.



SM7GMDs interface, model "KISS".

However, if you want to minimize the risk of problems, one should consider buying a ready-made interface.

For this article, I have chosen to describe the card that SA2BLV has developed. This card not only replaces the USB audio device and necessary electronics, such as transistors or opto-couplers – but it can also be used to power the pie from repeater's 12V.

SA2BLV's card also has ready-made connectivity to the Motorola MTR2000, as well as an adapter to connect the card to an Ericsson F800. In these cases, the card is placed inside the repeater itself, in its free slot. The pie is mounted under the card.

If you want to adapt to other radio hardware, there are also 2 RJ45 sockets on the card.

In the current version, the card supports 2 transmitters/receivers. These can be used as repeater and/or simplex nodes.

On the card there is also some other useful things, such as an ADC. It can be used to read analogue signals, such as an RSSI output from the receiver or other voltages. There is also a socket, where you can connect different sensors, like *Dallas one-wire*, for example to measure temperature. There are also some 5V GPIO pins that you can control.

### Preparation of the computer

It may be tempting to buy an SD card with the OS preinstalled, but it is actually not that difficult to do this yourself. You need some kind of SD card reader. It is usually found in most laptops, for example. Otherwise, you can buy a reader that connects to the computer with USB.

Nowadays, the installation is quite simple. One just enters Raspberry Pi's website and there you choose to install *Raspberry Pi Imager*. When running this tool, you can choose which image you want to put on the card. After that, the card is formatted and then the operating system is installed. If you want to be able to access the pie with *ssh* right from the start, it is recommended to create the file "ssh" in the card's boot partition.

Now you take the card out of your reader and put it in the single-board computer and then you boot it up.

For a beginner, it may be tempting to having the desktop (GUI) installed. There comes a "guides", where you get to fill in the necessary things such as computer

name. But all this can also be done via *ssh*. The corresponding settings can be made in the *raspi-config* tool. There you can also start by expanding the file system, putting computer names, entering wireless network, etc. You must also enter a (new) password for the user *pi*.

I myself now never install the desktop, because I mean that it burdens the computer completely unnecessarily, and that then there will be many more updates. An SD card has a limited lifespan and therefore you want to avoid unnecessarily numerous writes against the card.

Using SA2BLV's card, you also need to activate the kernel driver

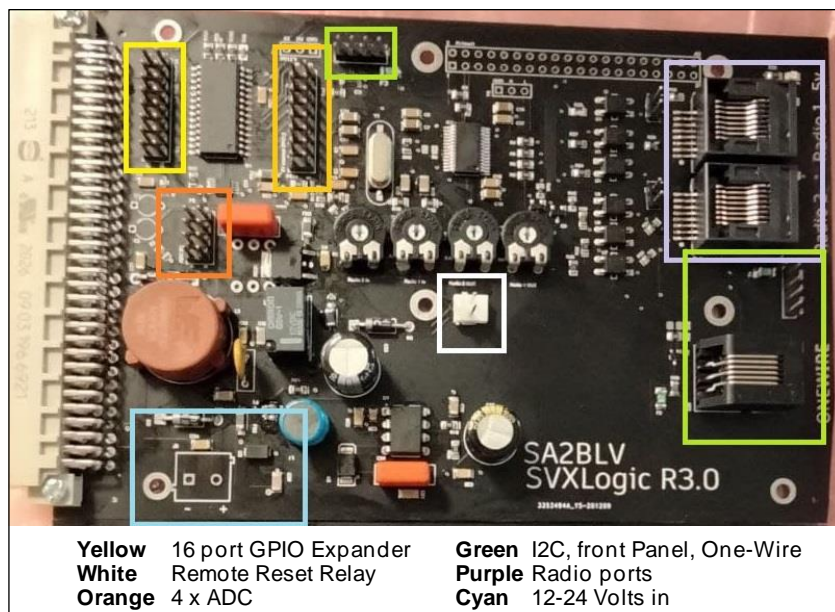
```
dtparam=i2c_arm=on
dtparam=i2s=on
dtparam=spi=on
dtparam=audio=on
dtoverlay=audioinjector-wm8731-audio
dtoverlay=w1-gpio
```

for the card's built-in entities by editing/adding the following lines to */Boot/config.txt*.

### Installation of SvXLink

There are prebuilt binaries for SvXLink, but they can be "ancient". Therefore, I recommend building SvXLink from source code, which is located at [Github](#). Then you always get the latest version. It's not as hard as it sounds, actually.

There is an [instruction](#), which step-by-step describes exactly what commands you need to give



to download the source codes and to compile them. It also describes how to get SvxBLink to start as a service at boot.

A similar procedure is used to keep SvxBLink up to date on an ongoing basis, when SM0SVX issues new versions. You then just do a *git pull* in the source code folder and then you perform the same *cmake* and *make* commands as in the case of new installation.

I myself have chosen to create *bash* scripts that perform the installation and updates SvxBLink.

### SvxBLink Configuration

SvxBLink comes with a standard configuration and a set of script files. In the */etc/svxblink* folder, the installation has placed some configuration files that now must be edited.

*gpio.conf* contains configuration for which GPIO pins you need to use. If you build your own interface, you have more or less full freedom to choose GPIO pins to use. If you use SA2BLV's card, however, it is predetermined which pins are used. The following configuration will then need to be in the file:

```
GPIO_IN_HIGH="gpio17 gpio27 gpio22"
GPIO_OUT_HIGH="gpio24 gpio23"
```

In more recent versions of SvxBLink, this access method is deprecated and replaced by *GPIO\_D*, which accesses the pins directly from the config file.

*svxblink.conf* is SvxBLink's main configuration file. Of course, there is no room for describing every parameter here. You get a detailed description of everything you can do in this file by giving the command *man svxblink.conf*.

You need to configure, among other things:

- Call sign
- Audio device(s)
- Login to the reflector
- Default Talk Group
- Monitored talk groups
- PTT
- Squelch
- Possibly, CTCSS

### Modular construction

SvxBLink is very flexible and can be adapted to different requirements. As a minimum, you need either a [RepeaterLogic] or a [SimplexLogic]. The latter requires only one radio and therefore no cavity filters are needed.

To the logic, you also need to tie a transmitter and a receiver. These configuration blocks specify, among other things, which audio device to use, and how PTT and Squelch should be signalled.

The sound levels are set using the *alsamixer* command in the operating system. These settings should be made carefully, so that there are no irregularities in the network from different connected nodes. To help, the *devcal* tool is available.

When using SA2BLV's card, you need to keep in mind that it has one audio device, but with two channels (stereo – left/right).

It is important to configure how PTT is to be signalled to the radio. Typically, GPIO is used for this, but you can also use a serial port or another HID-compatible device.

It is, of course, possible to use the radio's squelch output. However, if you have the option, you should use a receiver with a "discriminator output", which is not affected either by the squelch nor the volume control of the radio. In such cases, you need to configure some other type of squelch, such as CTCSS or why not let the logic measure signal/noise ratio!

SIGLEV needs to be calibrated and this is done with the tool *siglevdetcal*. It still needs to be done,

as the value is used on the SvxB Portal as well.

SvxBLink supports *deemphasis* and *preemphasis*, which is used if you connect "further into" the radio, which is preferable – because it provides significantly better audio quality, compared to using, for example, a microphone input.

### ReflectorLogic

In order to connect to the national reflector, you also need a [ReflectorLogic]. It specifies where you want to connect and your assigned account settings.

This configuration block also specifies which talk groups to monitor on the reflector, as well as which talk group the repeater should activate as default on start-up.

<b>240</b>	Sweden
<b>2402</b>	SM2
<b>24020</b>	SK2SSA bulletin
24021	"Kalix line" (Norrbotten)
24022	Västerbotten
<b>2403</b>	SM3
24031	Gävleborg
24033	Sundsvall
24034	Örnsköldsvik
<b>2404</b>	SM4
<b>2405</b>	SM5
240515	Eskilstuna
<b>2406</b>	SM6
24061	Falkenberg
24062	Lysekil
<b>2407</b>	SM7
<b>24070</b>	SK7SSA bulletin
24078	Öland

A complete talk group list can be found at the [SvxB Portal](#).

Usually all nodes monitors (as a minimum) talk group 240, its own district talk group and a local



SA2BLV's card, mounted in an Ericsson F800.

talk group that should be default on the repeater. You can also monitor adjacent districts or repeaters.

Personally, I prefer to choose the talk group with DTMF, but SvxBLink supports and encourages subtone activation (CTCSS). A subtone per talk group is then defined, to be activated when starting the repeater. There you should follow the standard that exists, so that the same subtone always activates the same talk group, regardless of which node you run over. The [Svx Portal](#) provides a list of CTCSS mappings in use.

### node\_info.json

In order for the [Svx Portal](#) to display the correct information, this file needs to be edited to reflect the local conditions. On the [Svx Portal](#) there is even a generator, so you don't have to edit on your own. Json can be tricky for the unfamiliar.

The [Svx Portal](#) can show received signal quality and signal strength, which talk group each node is on right now, and many other interesting things. The portal also has a map and the json file indicates where the repeater is located. The receiver ID must match the respective letters specified in *svxlink.conf*.

### Links

In order for *RepeaterLogic* and *ReflectorLogic* to "talk" to each other, you also need a "link". It determines which logics should be interconnected. Such a link can also bridge multiple repeater logics or a repeater logic with a simplex logic (which can be used as an analogue link to a nearby repeater, for example).

This link also specifies which DTMF command to use to activate the link and any subcommands, such as talk group selection. At SvxBLink, we normally use **9** as a command to enable talk group based links.

<b>91</b>	<i>tg#</i>	Activates talk group
<b>92</b>	<i>#</i>	Manual QSY
<b>93</b>	<i>#</i>	Follow QSY
<b>94</b>	<i>tg#</i>	Monitor extra talk group

### Modules

SvxBLink also has several modules, which can be attached to a *RepeaterLogic* or a *SimplexLogic*.

*Parrot*; sending back what is received.

*MetarInfo*; weather report for specified airports is read out verbally.

*EchoLink*; a client for the EchoLink network.

*Frn*: a client for the Free Radio Network.

*SelCallEnc*; used to send various selective calls, such as CCIR.

*DtmfRepeater*; used to transmit DTMF on a repeater or link.

*PropagationMonitor*; used to receive warnings about various events that affect radio propagation.

*TclVoiceMail*; a mailbox function for voicemail.

*Trx*; used to connect to and remotely control a transceiver or receiver, such as an RTL-SDR.

Of course, you decide for yourself which modules you need and would like to use. All module configuration files can be found in the *svxlink.d* subfolder.

### Other features

If you want to use multiple receivers for your logic, there is a so-called *Voter* – who chooses the receiver that receives the signal best at the moment.

If you want to control two repeaters in parallel, you can use *MultiTx*.

Of course, you might as well have two *RepeaterLogics* or *SimplexLogics* linked directly or via the reflector.

If you want an extra receiver on another QTH, you can use a *RemoteTrx*. This Rx can be linked with analogue radio, but an IP network is of course the best. Note that it is the main node that connects to the receiver, which must therefore have a reachable (public) IP address.

### TCL scripts

We mentioned earlier that there is a folder structure for the scripts that control different tones and voice messages from the repeater. Normally you do not have to edit these, but you can do a lot of interesting things in these files!

What you have to keep in mind, if you want to edit these, is to create a subfolder *local* under the folder */usr/share/svxbLink/events.d* – where you put copies of the files you want to edit. The original files should not be changed because they are overwritten when updating SvxBLink.

### Connection to the Network

There are a few simple [rules and conditions](#) that must be met, including:

- Most importantly, audio and signal levels are calibrated.
- Configuration of talk groups and CTCSS should follow the standard.
- The file *node\_info.json* is updated.

When you are ready to connect to the Swedish National network, you send an email to [reflector-sm-join@svxbLink.org](mailto:reflector-sm-join@svxbLink.org).

### Svx Portal

We need to mention *The SvxB Portal* again, which is an excellent tool to gain a better understanding of how the system works.

The portal is operated and developed by SA2BLV, Peter.

There you can see in real time how the system behaves in different situations. But it is also a valuable source of information about, for example, which talk groups exist and how they are used and monitored by the respective nodes.

If you need to ask questions and seek information in general, there is of course also a forum – or rather several.

- [International Software Forum](#)
- [Swedish Sysop Forum](#)
- [Swedish User Forum](#)

### Summary

This is written in the hope that several repeater owners will see benefits in updating their repeater to SvxBLink, instead of switching to DMR, for example – or in the worst case, closing down their repeater due to lack of interest or poor activity.

In the case of a computer-based logic, there may be thresholds to overcome – both for the sysop and the users. The purpose of this article is that those who are

interested in migrating to SvxBLink, but who at the same time do not feel so familiar with Linux and single-board computers, should still dare to take the step. Personally, I knew almost nothing about Linux or repeater logics when I started working on this in the summer of 2019.

SvxBLink does not require anything from the user, who can continue to use their old FM radio. I always conclude by pointing out that we have to be very careful not to scare off users. Of course, anyone who just wants to start the repeater as usual and run local QSOs must be able to continue to do so.

This is precisely the beauty of SvxBLink, especially when compared to other modes – such as DMR, D-Star and C4FM. And on SvxBLink, the audio is analogue, and sounds just as good as it does

at a local QSO on any FM repeater!

My experience from the island of Öland is that not many people activate talk groups with DTMF. But with monitored talk groups and a default talk group, you can come very far – without being able to transmit DTMF or CTCSS.

On the island of Öland, for example, all 4 repeaters are connected through a default talk group, so users don't have to do anything special. And to answer a call to e.g., talk group 240, all you have to do is pressing your PTT and answer.

### Sources of Information

[SvxBLink's website](#) of course contains a lot of information, as well as [SvxBLink's Github](#).

[The SvxB Portal](#) also has some information pages, which you can study. Each sysop is prompted to

create an account on the Portal, in order to update the information about their own repeaters.

On [SK7RFL.se](#) there is my "Repeater School" as well as some [slideshows](#), which you can study or use as starting material – for example for a lecture. There you can also study how we used both talk groups and analogue links, to bridge the repeaters together on the Island of Öland.

It is worth noting that we at Öland have also further developed SvxBLink. Some of the commands and functions described on our website are not SvxBLink standard.

It is not "forbidden" to create your own solutions, but it is good if they are backward compatible. Otherwise, those who visit the area may have problems. It is also desirable to document all deviations.

**Repeater Flitiga Lisa – SK7RFL**  
 SvxBLink 24078 – Echolink SK7RFL-R – DMR 240721 – YSF SE-SK7RFL – D-star DCS010X

