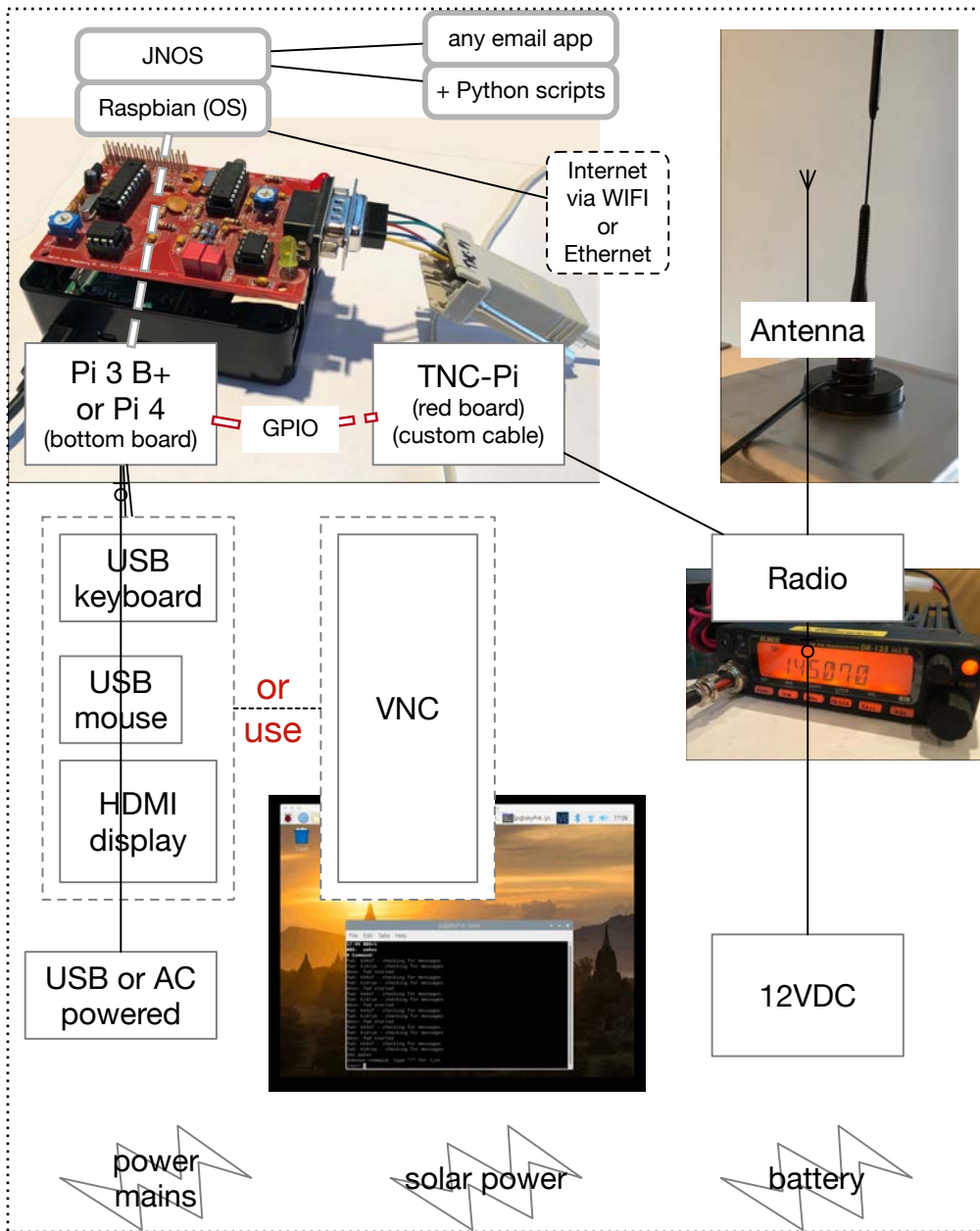


A JNOS-based packet radio BBS with Internet email connection

In early 2020, I constructed a system capable of sending, receiving, and forwarding (relaying) packet radio text messages to and from amateur radio packet stations, and with an ability to relay these messages to and from standard Internet email addresses. My particular configuration has certain portability improvements over earlier email gateways.



You could think of this as a “radio-based email server” and other radio stations as “client” stations in the sense of client-server. Each of those also requires radio+computer+software (though they could also be a clone of this system) to operate.



The JNOS “BBS” is a central place where the messages are temporarily stored and routed. Although it can be used in a “BBS” fashion, it also interoperates with computer software such as Outpost to process messages that for all intents are “radio email.” Outpost keeps its own repository of sent/received messages on the client station’s own computer. Although the BBS does have a very high capacity, in general it is not a place where messages are stored for the long term. Outpost is not required on the JNOS system.

Physically the system consists of

Raspberry Pi 3 or 4 computer	Quad CPU; 2GB RAM; 32GB main storage (SD)	\$45 to \$80
TNC-Pi or TNC-X	Interface between computer and radio. Cabling varies by radio model and can be home-brew or purchased.	\$150 for TNC-X plus \$30 pre-wired cable or \$15 for connectors and self-wiring
During install only: physical keyboard, mouse, HDMI monitor	During operation use VNC as a “virtual” keyboard, mouse, and screen.	\$0 to about \$150 for HDMI monitor
Internet via Wi-Fi	Ethernet cable also an option	\$0
One external email account (gmail, etc.)		\$0
2 meter transceiver	Alinco DR-135; 5/20/50 W	under \$200
Antenna	Mag-mount, ducky, etc. will suffice; coax cable;	\$20 to \$30 (example Ed Fong J-Pole)
Power	110VAC to USB for the Pi; 110VAC to 12VDC for the transceiver; conventional power supply works, but solar and battery desired.	~\$100
TOTAL		~ \$600 without solar, battery, display, keyboard, mouse

How the Internet mail connector works

I wrote small Python scripts that run on my Raspberry Pi (also known as RasPi) system and interface with JNOS on one side, and with an external email server (such as gmail) on the outside:

INBOUND: The script checks the remote email server using secure IMAP, brings in any incoming traffic, delivers it to JNOS addressed to the proper account, then archives the incoming email in case later analysis is desired (after-action). The message may be relayed to another BBS if required, or will be delivered to the recipient at next connection.

OUTBOUND: JNOS rules route any outbound (Internet) email to a special spool area. The script checks and when messages are waiting it reads them and uses an SMTP connection to

the remote email (gmail, etc.) to send those to the public Internet. The Reply-To address is configured so that a regular email user's "reply" comes back into our system automatically.

There are other existing solutions to take packet messages, via JNOS or PiGate and route them to and from Internet email. But, all of the solutions I know of require external system configuration, such as gaining an AMPR IP address, registering and maintaining a Winmail account, or configuring a SENDMAIL or EXIM gateway. They may also require DNS entries in the case of a gateway setup. My goal was to accomplish this with only the requirement of an external email account. Most of us do not have, and do not want to keep up our skills as system administrators. And establishing an email account is easy.

What remains to be done

Power supplies for transceivers are bulky. Need either a more compact 120VAC supply and/or a 12V supply that is solar and battery based.

And I want to package this all so it can be stored out of sight in a garage somewhere, with a good solar-panel placement, long-lasting battery power, and a high antenna.

Sending and Receiving

In practice, radio operators use software such as Outpost to connect from their own packet radio station to the base BBS at AA6AX-1 (or KK6SF-1 or KI6RYE-1 etc.). Any stations connecting to any of these BBS's on the air can send and receive messages to and from any operator on that same BBS without additional relaying.

Example (local to the BBS):

To: KI6RYE
From: AA6AX

This kind of message, sent on the air to AA6AX-1, just goes directly to the recipient's account on AA6AX-1. It then stays in the JNOS mailbox until the receiving station's Outpost (or operator manually) deletes it. Long-term storage of messages is on the sending or receiving packet stations' software.

Example (from this BBS to another BBS):

To: KI6RYE@KK6SF
From: AA6AX@AA6AX

Messages entered through AA6AX-1 may be addressed to users at other packet BBS stations, such as KI6RYE and KK6SF. (An explicit forwarding rules must be set up in JNOS.) The JNOS software forwards these on the air to the destination BBS, from which they can be picked up by radio operators who connect to the destination BBS. This would be useful when a station has access only to a more local BBS and not the originating BBS. In the example, mail entered thru AA6AX-1 would be forwarded to KI6RYE who connects to the KK6SF BBS system.

And finally, messages may be addressed to Internet email addresses:

Example message to the Internet:

To: user@example.com
From: aa6ax@AA6AX

This message originates with the packet radio station, is sent on the air to AA6AX-1, which determines that it needs to be routed to the Internet, and connects with a remote service (like gmail or other — but must be specified) for delivery. This requires that the Internet connection from AA6AX be operational at the time.

Inbound messages to the gateway address (<foo@example.com> let's say, but that's not the real address, contact me if you want to know the real address) from outside systems (again, gmail, etc.) come in via Internet, are retrieved by our software, and inserted into the JNOS mailbox, or routed to one of the remote BBS systems. The target JNOS user is indicated within quotes in front of the gateway email address.

Example message incoming:

To: "aa6ax@kk6sf" <foo@example.com>
From: user@example.com

This message from user@example.com goes to user aa6ax on the BBS at kk6sf and must be picked up on the air from KK6SF-1.

Using “regular email apps” instead of Outpost

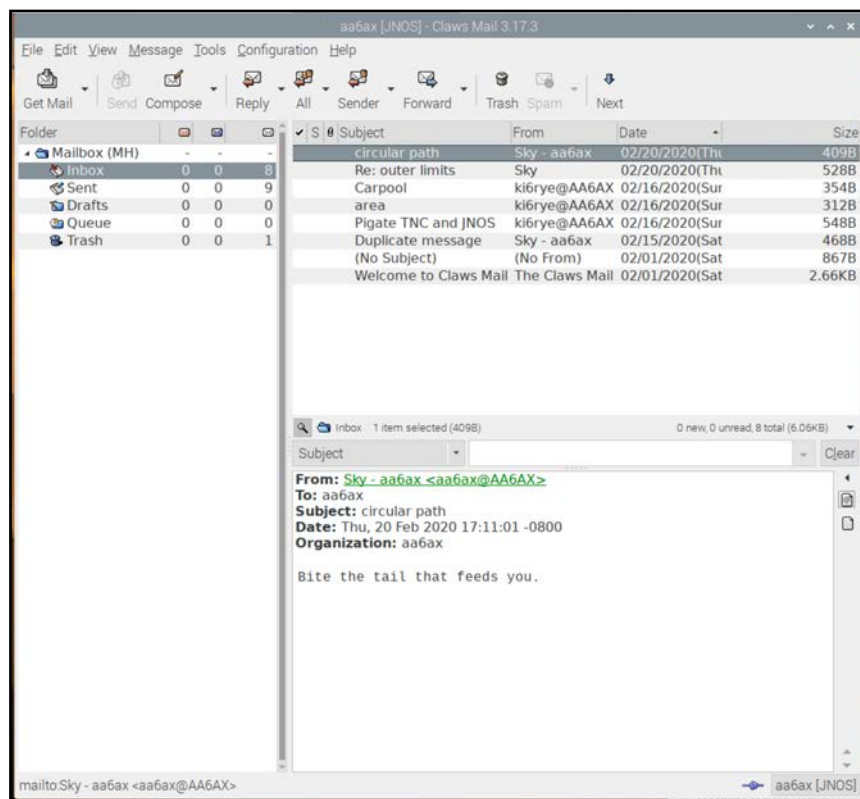
I’d like this system to be self-contained. To achieve this, I needed to be able to send and receive both packet and Internet email from the RasPi desktop. An email client app like **Claws Mail**, **Thunderbird**, or anything else you’re familiar with, would work.

For packet operation, most people use the Outpost app, and it does everything I want, but it requires Windows (or the WINE app), which I want to avoid. So Outpost is not an option for me on this project.

To enable *outbound* mail from a modern email app requires adding ESMTP (Extended SMTP) to the existing JNOS SMTP server. I have made a simple code change and recompiled JNOS to allow a basic ESMTP interaction. And so that works, so with this addition any email app can now send mail directly using a JNOS user account.

To enable *inbound* mail to an app required adding a simple POP3 server on the RasPi. One that can read the JNOS internal formats and then provide local POP3 interaction turns out to be enough. You can think of this as a program that knows the JNOS formats on one side and knows how to talk (POP3) to an email app on the other side. So that’s what I’ve written. The easy codebase to start from was **pypopper** and I was able to code up a Python script mailpopper.py that now interfaces between JNOS and POP3.

Here, for example, is what the Claws Mail interface looks like. You just write up an email and send it out (via packet or BBS), and you just click **Get Mail** and your JNOS incoming messages appear in the list and the message window. No BBS command line or clunky interfaces are required any more. Easy huh?



TNC — two options

Initially I constructed and tested the system using a TNC-Pi (from Coastal Chipworks) which plugs onto the GPIO pins that stick out the top of the RasPi3 or RasPi4 boards. That's the brown/red board you see in the top photo.

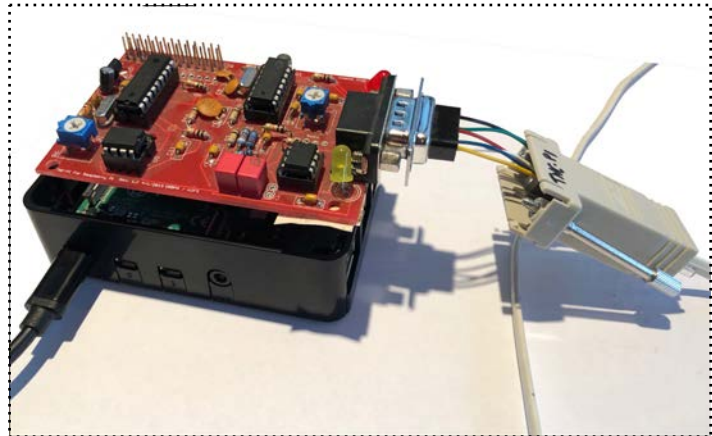
This functioned beautifully, but left the pins and the board exposed on the top. Although one can purchase a custom case, that was a hassle.

Ultimately I switched to a TNC-X. Coastal Chipworks has licensed this design to MFJ and it is available as the MFJ-1270. Although the TNC is twice the size of the RasPi enclosure, it is a USB device so it plugs in externally and allows the protective RasPi case to be closed. In the lower photo I left the RasPi open so you can see the computer.

There is a 5-pin DIN connector on the back of the TNC that has to be wired to the radio. On the radio end it's equivalent to the DB9 cable I have previously shown.

The TNC-X does seem to be very sensitive to RF, and it is important to position it far from the radio's antenna. In my case I found a separation of about 10 feet was required.

The system could also use a Kantronics TNC in "KISS" mode. This might require a serial adapter (there are USB-to-serial adapters in the \$15 range). Although I know it would work, I have not tested it with my configuration.



Routing diagram with email routing examples

The system operator and users (on AA6AX BBS) now just use a regular email app to write and to receive messages, whether they're from packet radio or from the Internet.

