DIY Internet Radio Receiver

The kidOYO Project

Kubernetes and Authentication

LINUX JOURNAL

THE 25TH ANNIVERSARY ISSUE

Interview with Linus Torvalds

ISSUE 297 | APRIL 2019
www.linuxjournal.com

Photo courtesy of Peter Abrams, The Faces of Open Source Project

CONTENTS | APRIL 2019 ISSUE 297

100 *DEEP DIVE:* ***Kids and Linux***

101 The Kids Take Over
by Doc Searls
As with Linux, these kids are all about making things—and then making them better. They're also up against incumbent top-down systems they will reform or defeat. Those are the only choices.

122 Linux...Do It for the Children
by Marcel Gagné
A rundown of some fun and educational Linux software for kids.

138 Thoughts from the Future of Linux
by Bryan Lunduke
What do kids want to do with Linux? And, where will the next generation take open-source computing?

IN MEMORIUM DENNIS JAMES RANKIN.

LINUX JOURNAL (ISSN 1075-3583) is published monthly by Linux Journal, LLC. Subscription-related correspondence may be sent to 9597 Jones Rd, #331, Houston, TX 77065 USA. Subscription rate is $34.50/year. Subscriptions start with the next issue.

CONTENTS

6 **The 25th Anniversary Issue**
by Bryan Lunduke

11 **25 Years Later: Interview with Linus Torvalds**
by Robert Young
Linux Journal's very first issue featured an interview between *LJ*'s first Publisher, Robert Young (who went on to found Red Hat among other things), and Linus Torvalds (author of the Linux kernel). After 25 years, we thought it'd be interesting to get the two of them together again.

22 **1994 Interview with Linus, the Author of Linux**
by Robert Young
From Issue #1, March/April 1994.

30 **Letters**

UPFRONT

34 **A Big Thanks to Our Subscribers**

38 **The Asian Penguins**
by Bryan Lunduke

43 **Patreon and *Linux Journal***

44 **FOSS Means Kids Can Have a Big Impact**
by Corbin Champion

49 **Reality 2.0: a *Linux Journal* Podcast**

50 **FOSS Project Spotlight: Drupal**
by Lizz Troudeau

55 **Plotting on Linux with KmPlot**
by Joey Bernard

63 **News Briefs**

COLUMNS

66 **Kyle Rankin's Hack and /**
What Linux Journal's Resurrection Taught Me about the FOSS Community

80 **Reuven M. Lerner's At the Forge**
Open Source Is Winning, and Now It's Time for People to Win Too

86 **Dave Taylor's Work the Shell**
Back in the Day: UNIX, Minix and Linux

92 **Zack Brown's diff -u**
What's New in Kernel Development

168 **Glyn Moody's Open Sauce**
Open Source Is Eternal

ARTICLES

146 **Kubernetes Identity Management: Authentication**

by Marc Boorshtein

You've deployed Kubernetes, but now how are you going to get it into the hands of your developers and admins securely?

158 **Build Your Own Internet Radio Receiver**

by Nick Tufillaro

Tune in to communities around the world with the push of a button.

AT YOUR SERVICE

SUBSCRIPTIONS: *Linux Journal* is available as a digital magazine, in PDF, EPUB and MOBI formats. Renewing your subscription, changing your email address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly online: https://www.linuxjournal.com/subs. Email us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, 9597 Jones Rd #331, Houston, TX 77065 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE: Your monthly download notifications will have links to the different formats and to the digital archive. To access the digital archive at any time, log in at https://www.linuxjournal.com/digital.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at https://www.linuxjournal.com/contact or mail them to *Linux Journal*, 9597 Jones Rd #331, Houston, TX 77065 USA. Letters may be edited for space and clarity.

SPONSORSHIP: We take digital privacy and digital responsibility seriously. We've wiped off all old advertising from *Linux Journal* and are starting with a clean slate. Ads we feature will no longer be of the spying kind you find on most sites, generally called "adtech". The one form of advertising we have brought back is sponsorship. That's where advertisers support *Linux Journal* because they like what we do and want to reach our readers in general. At their best, ads in a publication and on a site like *Linux Journal* provide useful information as well as financial support. There is symbiosis there. For further information, email: sponsorship@linuxjournal.com or call +1-360-890-6285.

WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found online: https://www.linuxjournal.com/author.

NEWSLETTERS: Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on https://www.linuxjournal.com. Subscribe for free today: https://www.linuxjournal.com/enewsletters.

EDITOR IN CHIEF: Doc Searls, doc@linuxjournal.com

EXECUTIVE EDITOR: Jill Franklin, jill@linuxjournal.com

DEPUTY EDITOR: Bryan Lunduke, bryan@lunduke.com

TECH EDITOR: Kyle Rankin, lj@greenfly.net

ASSOCIATE EDITOR: Shawn Powers, shawn@linuxjournal.com

EDITOR AT LARGE: Petros Koutoupis, petros@linux.com

CONTRIBUTING EDITOR: Zack Brown, zacharyb@gmail.com

SENIOR COLUMNIST: Reuven Lerner, reuven@lerner.co.il

SENIOR COLUMNIST: Dave Taylor, taylor@linuxjournal.com

PUBLISHER: Carlie Fairchild, publisher@linuxjournal.com

ASSOCIATE PUBLISHER: Mark Irgang, mark@linuxjournal.com

DIRECTOR OF DIGITAL EXPERIENCE:
Katherine Druckman, webmistress@linuxjournal.com

DIRECTOR OF SALES: Danna Vedder, danna@linuxjournal.com

GRAPHIC DESIGNER: Garrick Antikajian, garrick@linuxjournal.com

ACCOUNTANT: Candy Beauchamp, acct@linuxjournal.com

Linux Journal is published by, and is a registered trade name of, Linux Journal, LLC. 4643 S. Ulster St. Ste 1120 Denver, CO 80237

SUBSCRIPTIONS
E-MAIL: subs@inuxjournal.com
URL: www.linuxjournal.com/subscribe
Mail: 9597 Jones Rd, #331, Houston, TX 77065

SPONSORSHIPS
E-MAIL: sponsorship@linuxjournal.com
Contact: Director of Sales Danna Vedder
Phone: +1-360-890-6285

Join a community with a deep appreciation for open-source philosophies, digital freedoms and privacy.

Subscribe to *Linux Journal* Digital Edition for only **$2.88 an issue.**

SUBSCRIBE TODAY!

Build Your Own Internet Radio Receiver

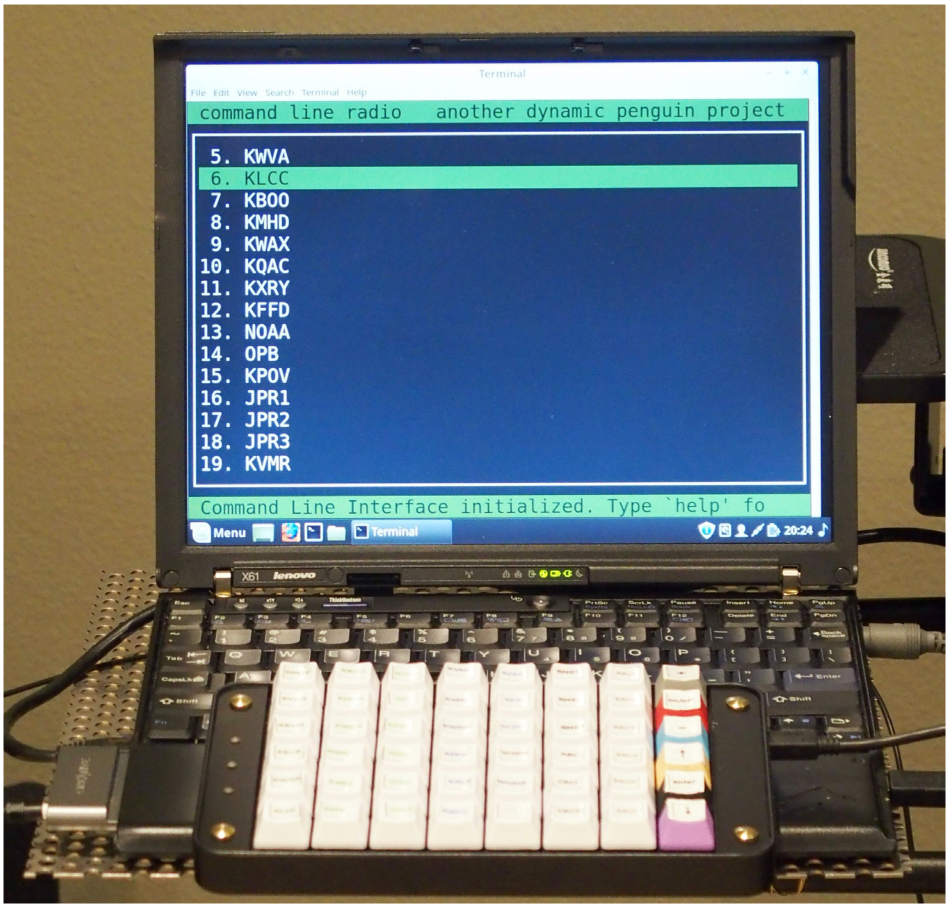
Tune in to communities around the world with the push of a button.

By Nick Tufillaro

When I get home at night, I like to tune into the world with the push of a button. I've lived in lots of different places—from Dunedin, New Zealand, to Santa Fe, New Mexico—and in each town, I've come to love a radio station (usually a community radio station) that embodies the spirit of the place. With the push of a button, I can get a bit back in sync with each of these places and also visit new communities, thanks to internet radio.

Why build your own internet radio receiver? One option, of course, is simply to use an app for a receiver. However, I've found that the most common apps don't keep their focus on the task at hand, and are increasingly distracted by offering additional social-networking services. And besides, I want to listen now. I don't want to check into my computer or phone, log in yet again, and endure the stress of recalling YAPW (Yet Another PassWord). I've also found that the current offering of internet radio boxes falls short of my expectations. Like I said, I've lived in a lot of places—more than two or four or eight. I want a lot of buttons, so I can tune in to a radio station with just one gesture. Finally, I've noticed that streams are increasingly problematic if I don't go directly to the source. Often, streams chosen through a "middle man" start with an ad or blurb that is tacked on as a preamble. Or sometimes the "middle man" might tie me to a stream of lower audio quality than the best being served up.

So, I turned to building my own internet radio receiver—one with lots of buttons that



```
command line radio   another dynamic penguin project

  5. KWVA
  6. KLCC
  7. KBOO
  8. KMHD
  9. KWAX
 10. KQAC
 11. KXRY
 12. KFFD
 13. NOAA
 14. OPB
 15. KPOV
 16. JPR1
 17. JPR2
 18. JPR3
 19. KVMR

Command Line Interface initialized. Type `help' fo
```

Figure 1. My Hardware Setup

allow me to "tune in" without being too pushy. In this article, I share my experience. In principle, it should be easy—you just need a Linux distro, a ship to sail her on and an external key pad for a rudder. In practice, it's not too hard, but there are a few obstacles along the course that I hope to help you navigate.

My recipe list included the following:

1. A used notebook with an ultra low voltage (Core 2 Duo) processor.
2. An audio interface with an optical TOSLINK.
3. pyradio: an open-source Python radio program.
4. An external keypad.

Why a notebook and not a Raspberry Pi or ship of a similar ilk? Mostly due to time—

my time in particular. It's not too hard to find a high quality notebook about ten years old for about $50, so the cost is really not that different, and I find the development platform to be much quicker.

In particular, I used the site ThinkWiki to research the Linux support of Thinkpads. On eBay, I found that the least expensive units often were sold without HDD—which is just fine with me, since I wanted a small SSD to keep the computer (whose main tasks is audio) quiet. I settled on a Thinkpad X61, but any notebook from that era will have more than enough oomph, and generally much more than any low-cost single-board computer option.

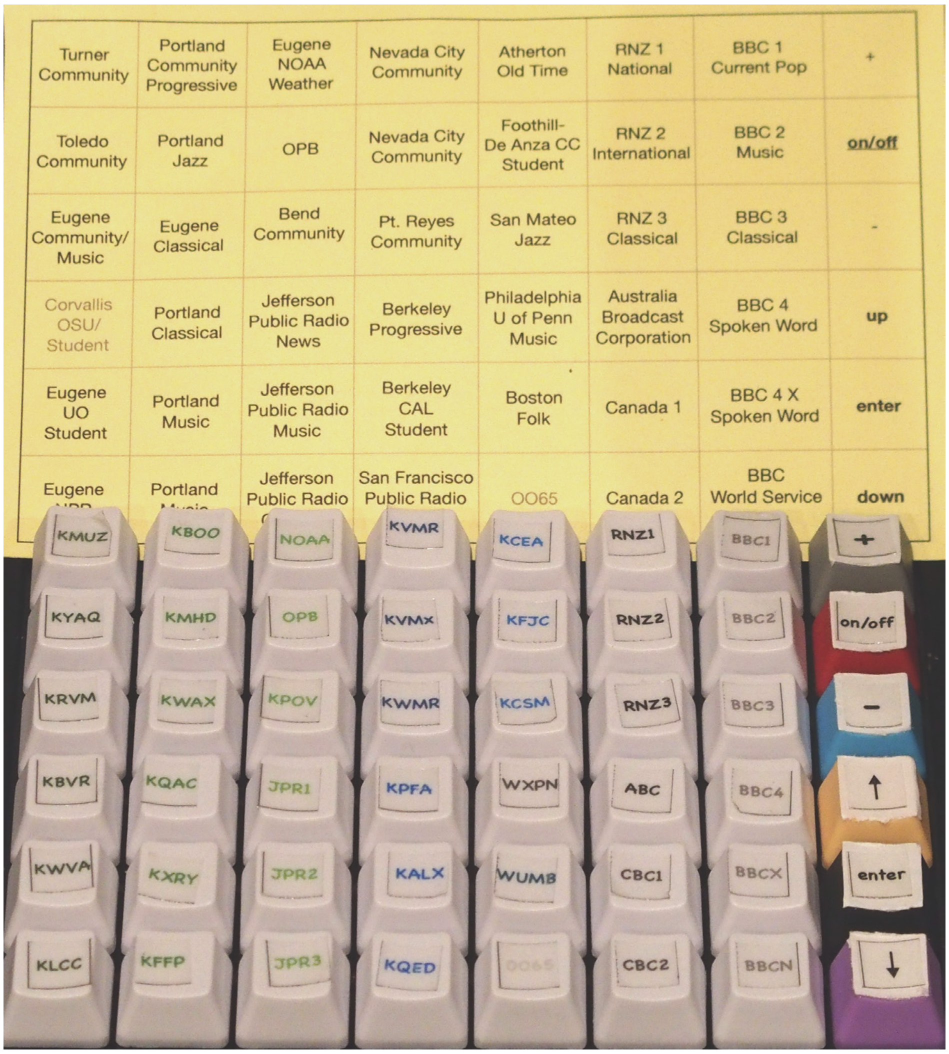
I wanted an optical audio link, a TOSLINK, and again, ThinkWiki is an excellent resource for looking into issues like driver support. I went with a used Soundblaster Audigy Cardbus sound card (because the system also doubles as an audio server for my FLAC recordings), which was a bit more pricey, but to save a few bucks, you can pick up a USB to TOSLINK converter on eBay for ~$10. My fondness for TOSLINK is due to its inherent electrical isolation that minimizes the chance of any audio hums from ground loops. And heck, I just think communicating by light is cool.

The other big piece of hardware is the keypad. To prototype, I just grabbed a wireless numeric pad with 22 keys, but for the final project, I spent a little more for a dedicated 48 key pad (Figure 2). The wireless keypad, of course, has the advantage that it can act as a remote control I can carry around the room when switching stations.

After getting all the pieces together, the next step is to install your favorite distro. I went with Linux Mint, but I'll probably try elementary for the next iteration.

The main piece of code is pyradio, which is a Python-based internet radio turner. The install is simple with snap:

```
$ sudo apt install snapd
$ sudo snap install pyradio
```



Turner Community	Portland Community Progressive	Eugene NOAA Weather	Nevada City Community	Atherton Old Time	RNZ 1 National	BBC 1 Current Pop	+
Toledo Community	Portland Jazz	OPB	Nevada City Community	Foothill-De Anza CC Student	RNZ 2 International	BBC 2 Music	on/off
Eugene Community/ Music	Eugene Classical	Bend Community	Pt. Reyes Community	San Mateo Jazz	RNZ 3 Classical	BBC 3 Classical	-
Corvallis OSU/ Student	Portland Classical	Jefferson Public Radio News	Berkeley Progressive	Philadelphia U of Penn Music	Australia Broadcast Corporation	BBC 4 Spoken Word	up
Eugene UO Student	Portland Music	Jefferson Public Radio Music	Berkeley CAL Student	Boston Folk	Canada 1	BBC 4 X Spoken Word	enter
Eugene	Portland	Jefferson Public Radio	San Francisco Public Radio	OO65	Canada 2	BBC World Service	down

Figure 2. Dedicated 48 Key Pad

You'll also need a media player, such as VLC or MPlayer.

I always need to look for where stuff gets dropped, for that I use:

```
$ cd /
$ sudo find . -name pyradio
```

In this case, I found the executable at /snap/bin/pyradio.

Like Music on Console (MOC), pyradio is a curses-based player. I find myself reverting to curses interfaces these days for a few reasons: nostalgia, simplicity of programming and an attempt to shake free of the ever-more clogged browser control interfaces that once held the promise of a universal portal, but have since become bogged down with push "services"—that is, advertising.

If you have not done any previous curses programming, check out the recent example provided by Jim Hall of using the ncurses library in *Linux Journal*. Take a look at the pyradio GitHub repository if you run into any installation issues. You also can build pyradio from source after cloning the repository with the commands:

```
$ python setup.py build
$ python setup.py install
```

You don't really need to know much, if any, Python beyond the two simple commands above to get running from source. Also, depending on your setup, you may need to use `sudo` with the commands above.

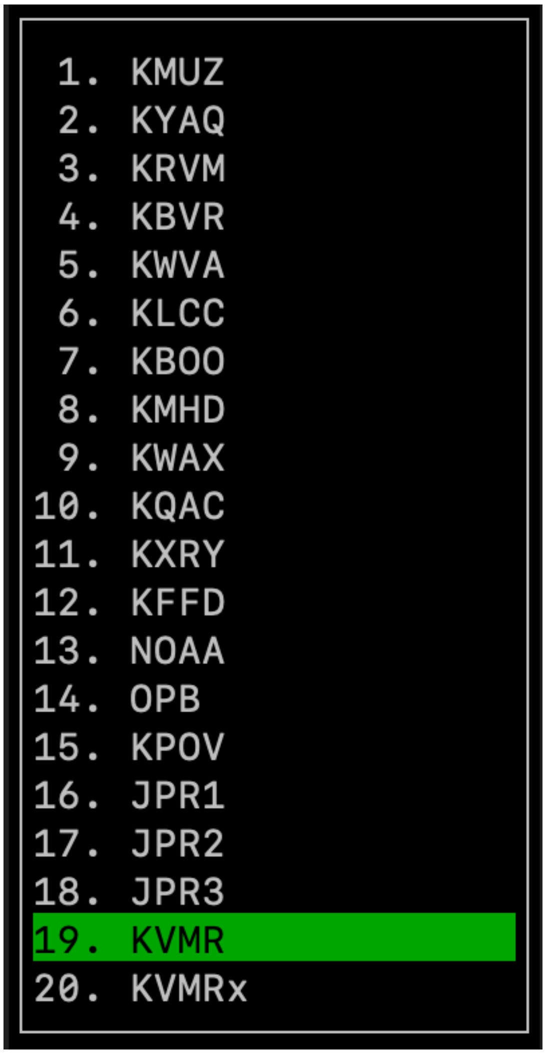
If all goes well, and after adding /snap/bin to your path, issuing the command:

```
$ pyradio
```

will bring up a screen like that shown in Figure 3.

You drive pyradio with a few keyboard-based commands like the following:

- Up/Down/j/k/PgUp/PgDown — change station selection.
- g — jump to first station.
- <n>G — jump to nth/last station.
- Enter/Right/l — play selected station.
- r — select and play a random station.
- Space/Left/h — stop/start playing selected station.
- -/+ or ,/. — change volume.
- m — mute.



```
 1. KMUZ
 2. KYAQ
 3. KRVM
 4. KBVR
 5. KWVA
 6. KLCC
 7. KBOO
 8. KMHD
 9. KWAX
10. KQAC
11. KXRY
12. KFFD
13. NOAA
14. OPB
15. KPOV
16. JPR1
17. JPR2
18. JPR3
19. KVMR
20. KVMRx
```

Figure 3. pyradio Screenshot

- v — save volume (not applicable for vlc).
- o s R — open/save/reload playlist.
- DEL,x — delete selected station.
- ? — show keys help.
- Esc/q — quit.

Some of those commands will change after you do the keypad mapping.

Next, you'll want to add your own station list to the mix. For that, I search for the file stations.csv with the command:

```
$ sudo find . -name stations.csv
```

And see that snap put the file at:

```
$ /home/[user_id]/snap/pyradio/145/.config/pyradio/stations.csv
```

Open stations.csv with an editor and replace the default stations there with your own selection. For instance, some of my entries look like this:

```
KMUZ,   http://70.38.12.44:8010/
KVMR,   http://live.kvmr.org:8000/aac96.m3u
RNZ1,   http://radionz-ice.streamguys.com:80/national.mp3 etc ...
```

The syntax is as straightforward as it looks. The field separator is a comma, and the first field is any text you want, presumably describing the station. I just use the call sign. The second field is a link to the stream. And this is where you face the first obstacle. Finding all the streams you want can be a bit tedious, particularly if you want to go directly to the source and not a secondary link from an aggregator website. Also, once upon a time, there were just a few encoding formats (remember .ram?), but now there are a multitude of formats and proprietary services. So identifying a good URL for the stream can be a bit of a challenge.

I start by going directly to the station's website, and if you are lucky, it will provide the URL for the given stream. If not, you need to do a bit of hunting. Using the Google Chrome browser, pull up the page View→Developer→Developer Tools. On the left part of the screen is the web page and on the right are a few windows for Developers. Click the menu labeled Network, and then start the audio stream. Under the "Network" window, step through the column labeled "Name". You should see the "Request URL" appear on the right, and you want to take notice of any link that could lead to the audio stream. It will be the one with a lot of packets bouncing to and fro. Copy the URL (and the IP number at "Request IP"), and then test it out by pasting the URL or IP:PORT number into the address box in the a browser. The URL might cause the start of the audio stream, or it might lead to a file that contains information—like a Play LiSt File (.pls file)—used to identify the stream.

For a specific example, consider the KMUZ (a community radio station in Turner, Oregon). I first go to KMUZ's home page at the URL, KMUZ.org. I note the "Listen Live" button on the home page, but before running the stream, I open the "Network" window in "Developers Tools". When that window is open, I click the "Listen Live" button, and search through the names in Requested URLs and see http://sc7.shoutcaststreaming.us:8010/ with IP number and port, 70.38.12.44:8010.

Pasting either of those identifiers into the URL box of the browser, I find the stream is from (a proprietary service) Shoutcast, which provides a Play LiSt file (.pls). I then open the playlist file with an editor (.pls are ascii files) to confirm that the IP/Port is the stream for listening to KMUZ.

Note two things. First, there are a lot formats/protocols in use to create a stream. You might find an MP3 (.mp3) file during your hunting, a multimedia playlist file (.m3u), an advanced audio encoding (.aac) or just a vanilla URL. So, getting a link to the stream you want requires some hunting and pecking. Second, if there is a preamble to the stream, you can usually avoid that by waiting for the stream to pass to the live broadcast, and then grab the live stream. That way, you won't need to listen to the preamble the next time you start the station. Your choice of audio player (VLC, MPlayer or similar) needs to be able to decode whatever formats you end up with for your particular group of radio stations.

The other place you might run into difficulties is mapping the keys on the keypad. That, of course, depends on the specific keypad you use. If you are lucky, the keypad is documented. If not, or to double-check the map, use a program to capture the keycodes as you press each key. Here is a Python program to find the keycodes:

```
from msvcrt import getch
while True:
   print(ord(getch()))
```

The other small piece of coding you need to do is point each keypress to a station. Locate the radio.py program in the same directory as the stations.csv. Edit the Python script so that each keypress causes the desired action. For instance, the streams in the station.csv are indexed by pyradio from 1 to N. If the first station in the list is KMUZ, and the keycode for the key you want to use is "h", then add or modify the radio.py script to include the snippet:

```
if char == ord('h'):
   self.setStation(1)
   self.playSelection()
   self.refreshBody()
   self.setupAndDrawScreen()
```

The functions/methods you will use are clearly labeled, such as the `playSelection` method above. So you really don't need any detailed knowledge of Python to make these changes. Make sure though that any changes do not conflict with other assignments of the keycode within the script. Functions, such as "mute", can be reassigned with the snippet:

```
if char == ord('m'):
   self.player.mute()
   return
```

Whatever changes you make though, try to keep the program usable from the notebook keyboard, so you still can do basic operations without the external keypad.

And that's just about it. Every good program, however, should have one kludge so as not to offend the gods. I wanted the pyradio program to run automatically after booting, and for that, I put a ghost in the machine. There are more natural ways to run pyradio at boot, but I like a rather spooky way using a shell script at login with `xdotool`:

```
sleep 0.2
xdotool mousemove 100 100 click 1
xdotool type "pyradio"
xdotool key KP_Enter
```

`xdotool` lets you script keyboard and mouse inputs to run as if you were actually typing from the keyboard. It comes in quite handy for `curses` programs.

Finally, I would be remiss if I didn't recommend a good radio show. My favorite at the moment is Matinee Idle on Radio New Zealand National, which plays a few times a year during holidays. It's like College Radio for the over 50 set. ■

Nick Tufillaro started programming on a vt52 in the terminal ward at Reed College in Portland, Oregon. These days, he monitors water quality around the globe using the science of ocean color and remote sensing. See aquahue.net, dynamicpenguin.com and http://ceoas.oregonstate.edu/profile/tufillaro for more info about the author.

Resources

- ThinkWiki
- MPlayer
- VLC
- Music on Console
- "Getting Started with ncurses" by Jim Hall
- pyradio
- Matinee Idle on Radio New Zealand National

Send comments or feedback via http://www.linuxjournal.com/contact or email ljeditor@linuxjournal.com.