Spectral Fusion: SDR meets Wi-Fi, Bluetooth, and drones in one new Tool

By ghostop14

November 2017

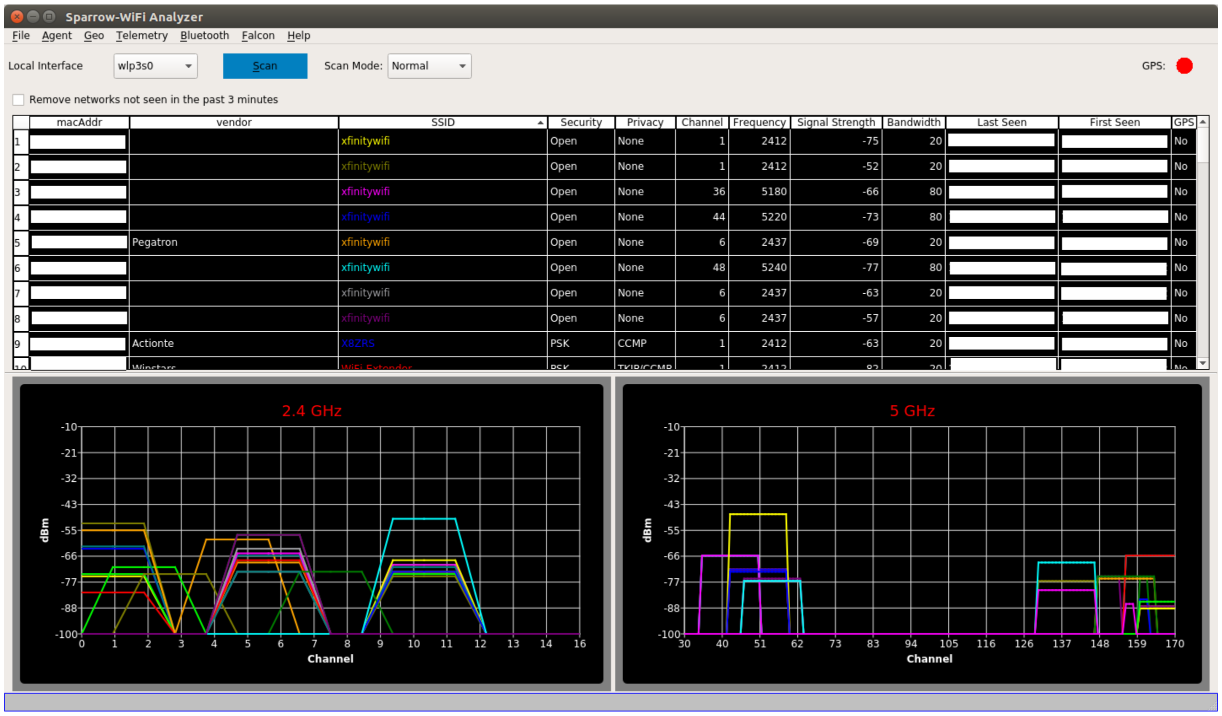
For years I've worked with traditional wireless tools and SDR, and like any other technical person we're happy with an app or tool for about 15 minutes before we wish it did something it didn't, and I'm no different. Thinking about the 2.4 and 5 GHz bands, my biggest issues with traditional wifi tools were always that apps such as inSSIDer which are great on the Windows side didn't have a nice polished Linux GUI equivalent so I'd have to run a Windows system or virtual machine to visualize the signal space. On the flip side, some of the great Linux-only capabilities didn't have a nice polished integrated UI and I'd have a lot of textual data, some of which the Windows tools didn't provide, but it was harder to visualize. Then there's the fact that wifi tools can't "see" Bluetooth (and vice versa), and SDR historically didn't have enough instantaneous bandwidth to show the whole 2.4 GHz or 5 GHz spectrum at one time. And, did I mention the tools don't integrate or talk to each other so I can't get a "single pane of glass" perspective of all the different ways to look at the same RF space simultaneously? It would be great if I could get one single view of the most common protocols and see the actual spectrum all in one place at the same time.

Now enter the era of the Internet-of-Things, new SDR receivers, and even drones and my old wifi tools seem to have been left a bit behind. Why do I say that? I can't "see" all of the chatter from wireless networks, Bluetooth, ZigBee, NEST devices, remotes, etc. scattered all over my wireless bands in one view. Sure, I can run 3 or 4 tools independently to find the signals and try to see what they are, but it becomes tough to get a single integrated perspective. Especially when I can't see my RF spectrum overlaid on top of the wifi SSID's and Bluetooth advertisements to sort out what may be related to a a signal I know about and what may be something else. Ultimately, it means that I can't clearly explain why I have poor wifi connections in one area versus another even though I may not have overlapping channels (I know, use 5 GHz and sparrow-wifi supports that too). The reason for this is simple; current tools don't have true spectral awareness based on the most common possibilities in one integrated solution.

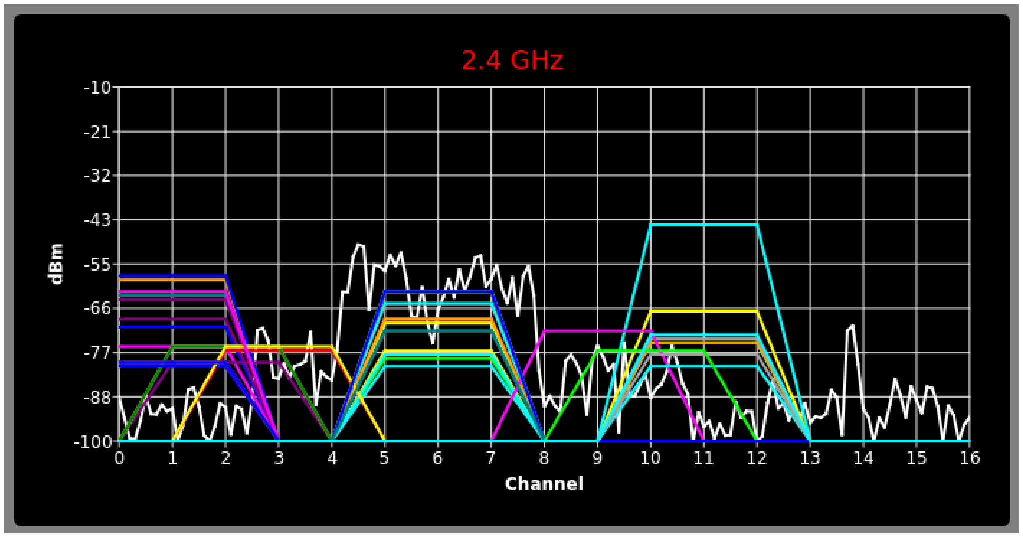
Now, let's ask even harder questions. What if I want to step up my wifi "wardriving" and start "warflying"? Or, what if I need a mobile platform that can be sent into an area on a rover? Can I bring the same spectral awareness in a small enough platform to fly for example as an under-350-gram payload complete with power, wifi, spectral scans, and even pull GPS for anything we see? And, can I interact with it remotely for real-time visibility or have it work autonomously? Okay, now you're just asking a lot.

These were all goals of a new tool I just released called "Sparrow-wifi" which is now available on GitHub (https://github.com/ghostop14/sparrow-wifi.git). Sparrow-wifi has been purpose-built from the ground up to be the next generation 2.4 GHz and 5 GHz spectral awareness and visualization tool. At its most basic, it provides a more comprehensive GUI-based replacement for tools like inSSIDer and linssid and runs specifically on Linux. In its most comprehensive use cases, Sparrow-wifi integrates wifi, software-defined radio (HackRF), advanced Bluetooth tools (traditional and Ubertooth), GPS via gpsd, and drone/rover operations using a lightweight remote agent and GPS using the Mavlink protocol in one solution.

Screenshots convey a thousand words, so here's a quick view of the main screen in basic wifi mode, then a 2.4 GHz sample (5 GHz is also covered by Sparrow-wifi) with a real-time spectral overlay. Depending on your hardware, spectral overlays can come from either a HackRF and the new hackrf_sweep firmware (for 2.4 and 5 GHz bands), or an Ubertooth (2.4 GHz only – not and quite as good resolution as the HackRF):


Sparrow-WiFi Analyzer
File Agent Geo Telemetry Bluetooth Falcon Help
Local Interface
wlp3s0
Scan
Scan Mode:
Normal
GPS:
Remove networks not seen in the past 3 minutes
2.4 GHz
5 GHz
dBm
Channel

	macAddr	vendor	SSID	Security	Privacy	Channel	Frequency	Signal Strength	Bandwidth	Last Seen	First Seen	GPS
1	[redacted]		xfinitywifi	Open	None	1	2412	-75	20	[redacted]	[redacted]	No
2	[redacted]		xfinitywifi	Open	None	1	2412	-52	20	[redacted]	[redacted]	No
3	[redacted]		xfinitywifi	Open	None	36	5180	-66	80	[redacted]	[redacted]	No
4	[redacted]		xfinitywifi	Open	None	44	5220	-73	80	[redacted]	[redacted]	No
5	[redacted]	Pegatron	xfinitywifi	Open	None	6	2437	-69	20	[redacted]	[redacted]	No
6	[redacted]		xfinitywifi	Open	None	48	5240	-77	80	[redacted]	[redacted]	No
7	[redacted]		xfinitywifi	Open	None	6	2437	-63	20	[redacted]	[redacted]	No
8	[redacted]		xfinitywifi	Open	None	6	2437	-57	20	[redacted]	[redacted]	No
9	[redacted]	Actionte	X8ZRS	PSK	CCMP	1	2412	-63	20	[redacted]	[redacted]	No
10	[redacted]	Winstars	WiFi-Extender	PSK	TKIP/CCMP	1	2412	[illegible]	20	[redacted]	[redacted]	No

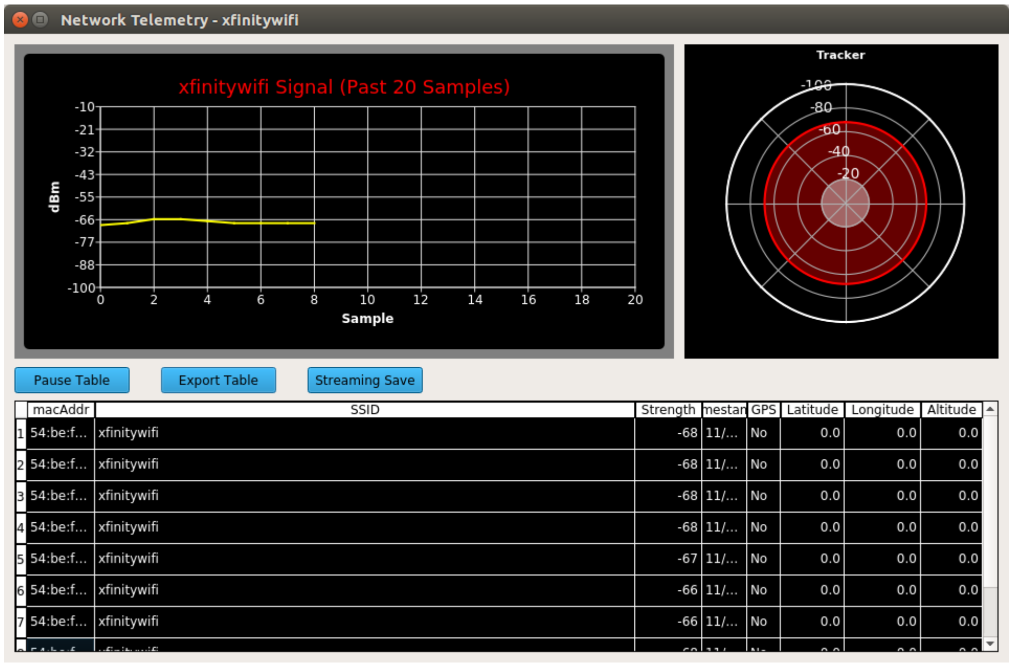


It's tough to get a screenshot of a fast-moving signal but running live you can actually see Bluetooth devices doing frequency hopping all over your wifi spectrum in real-time.

Written entirely in Python3 and leveraging all those great Linux tools under the covers, Sparrow-wifi has been designed for the following scenarios:

- ✓ Basic wifi SSID identification.
- ✓ Wifi source hunt - Switch from normal to hunt mode to get multiple samples per second and use the telemetry windows to track a wifi source.
- ✓ 2.4 GHz and 5 GHz spectrum view - Overlay spectrums from Ubertooth (2.4 GHz) or HackRF (2.4 GHz and 5 GHz) in real time on top of the wifi spectrum (invaluable in poor connectivity troubleshooting when overlapping wifi doesn't seem to be the cause).
- ✓ Bluetooth identification - LE advertisement listening with standard Bluetooth, full promiscuous mode in LE and classic Bluetooth with Ubertooth.
- ✓ Bluetooth source hunt - Track LE advertisement sources or iBeacons with the telemetry window.
- ✓ iBeacon advertisement - Advertise your own iBeacons.
- ✓ Remote operations - An agent is included that provides all of the GUI functionality via a remote agent the GUI can talk to.
- ✓ Drone/Rover operations - The agent can be run on systems such as a Raspberry Pi and flown on a drone (it’s made several flights on a Solo 3DR), or attached to a rover in either GUI-controlled or autonomous scan/record modes. And yes, the spectrum output works over this connection as well.
- ✓ The remote agent is HTTP JSON-based so it can be integrated with other applications
- ✓ Import/Export - Ability to import and export to/from CSV and JSON for easy integration and revisualization. You can also just run 'iw dev <interface> scan' and save it to a file and import that as well.
- ✓ Produce Google maps when GPS coordinates are available for both discovered SSID's / Bluetooth devices or to plot the wifi telemetry over time.

There are a number of task-specific features for each of these goals to make your life easier. For instance, as soon as you survey a wireless band, you will inevitably see a signal you can’t identify and want to track it down. To make this wifi or Bluetooth hunt easier, sparrow-wifi has two features to look at: hunt mode and the telemetry view. Normally it takes time (measured in seconds) to do a full wifi radio scan of all 2.4 GHz and 5 GHz channels. After all, the radio has to change frequencies, listen, and then move on to the next channel like any frequency sweep. However, if I have an idea from a broad scan what channel I’m looking for, can we lock in to that channel and fast-sweep it and get what looks like one of those “cool” sci-fi bug trackers? The answer is yes, switch to hunt mode and open a telemetry window (right click on the SSID in the table or a Bluetooth device on the Bluetooth screen), and now you can track with a few extra bells and whistles:



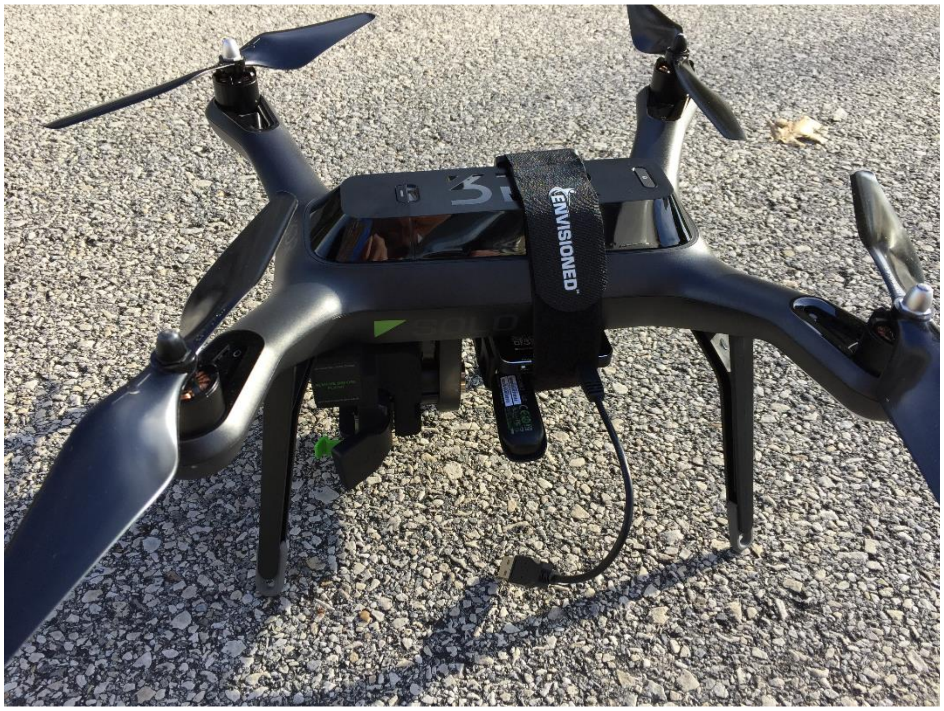
	macAddr	SSID	Strength	mestan	GPS	Latitude	Longitude	Altitude
1	54:be:f...	xfinitywifi	-68	11/...	No	0.0	0.0	0.0
2	54:be:f...	xfinitywifi	-68	11/...	No	0.0	0.0	0.0
3	54:be:f...	xfinitywifi	-68	11/...	No	0.0	0.0	0.0
4	54:be:f...	xfinitywifi	-68	11/...	No	0.0	0.0	0.0
5	54:be:f...	xfinitywifi	-67	11/...	No	0.0	0.0	0.0
6	54:be:f...	xfinitywifi	-66	11/...	No	0.0	0.0	0.0
7	54:be:f...	xfinitywifi	-66	11/...	No	0.0	0.0	0.0

A tracker-like radar gives you a new perspective of received signal strength along with a time history to get an idea when you're getting close. You can save all time-based samples to a file, and if you have GPS capabilities through gpsd or Mavlink you can later plot the tracked signal strength on a Google map from the Geo menu on the main screen.

Now to add to the spectral awareness challenge, how can we take this to the air or put it on a rover? To make this scenario work, something new was required: a lightweight remote agent that was capable of running on a small system like a Raspberry Pi (see readme in GitHub for all the details) and included all of the underlying capabilities of the GUI to be accessed remotely. In keeping with current technology trends, the agent is an HTTP server that responds to requests for each of the core functions by replying with JSON that can not only be used in the sparrow-wifi GUI but really any application (future projects?). For those security-conscious folks you can also apply IP-based restrictions on who can connect to the agent.   I then had to make sure that all of the GUI functionality was available in the agent and that the GUI could talk to it rather than any local hardware in a way that looked transparent to the user. The result was complete remote sensing capabilities. Now we can fly!

Weight is always an issue as the drones can only carry so much payload. In particular, the 3DR Solo I have can carry an additional 350 grams of payload while the camera is attached. So the first questions become what do we really need and what can we replace? Well there's a power source, the Pi itself, and the dual-band Wifi adapter to start. What if we wanted GPS, could we forego the external GPS receiver and gpsd to save on weight? These drones do have an onboard GPS and support the Mavlink protocol. In particular, there's already a python library called dronekit that wraps up a number of the vehicle communications issues including pulling GPS. So there's our GPS source…. no need for an external receiver. Sparrow-wifi just needs to be able to speak Mavlink (check, that's in the agent).

What about operating autonomously? Could the agent start as soon as the drone's GPS was synchronized and ready to fly and start recording, then use the GUI to pick up the recording files afterwards and revisualize them in the GUI and on a Google map? After some remote agent commandline parameter additions and some configuration screens in the UI, the answer is yes to all of these questions. The proof is always in execution, so below is a quick picture of the agent flying on a Solo 3DR drone (the hanging cord is just power before it was connected) with a Cisco AE1000 dual-band wifi adapter.



Hopefully this gives you a quick overview of what Sparrow-wifi is all about and where it's going. While there's a lot of functionality already in Sparrow-wifi, there's still a lot of work to be done. I already have some areas where I'd like to work on performance, and some new features and enhancements to add. In addition, I'm sure there will be the inevitable bug fixes. So keep an eye out for regular updates to the GitHub repository. In the meantime, enjoy the new tool!