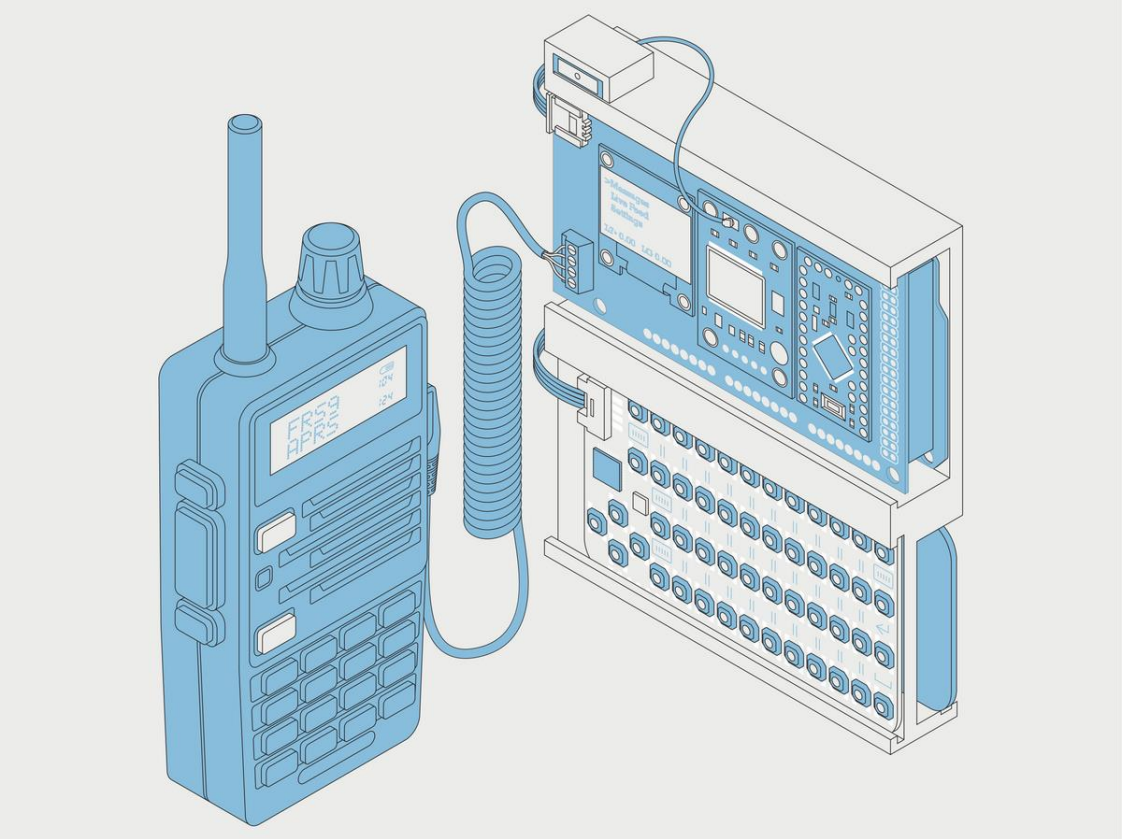
Hacking Ham Radio for Texting

An Arduino shield taps the potential of VHF handheld radios

Dale Thomas
17 Nov 2021
5 min read



The HamMessenger [right] lets you send short texts via a VHF radio without any additional equipment.
James Provost
amateur radio aprs ham radio

My first exposure to radio communication happened when I was around 5 or 6 years old. My dad was working as an airport electrician. He would bring walkie-talkies home, and my brothers and I would play with them around the yard. That's as far as my radio experience went, until a friend and I decided to get our amateur radio licenses together. This was only months before the COVID-19 lockdown, so it turned out to be the perfect time to learn to communicate using amateur radio!

However, I found that just talking over ham radio was boring for me. I started thinking about an old police scanner my dad owned and how we would sometimes hear odd sounds that sort of

sounded like a dial-up modem. And that is when the lightbulb for HamMessenger turned on. What if I could find an easy way to communicate digitally with my handheld radio?

I started learning about the many different types of digital communication modes that people use with ham radio and I came across APRS (Automatic Packet Reporting System). APRS is a store-and-forward radio network protocol developed over 25 years ago by U.S. Navy researcher Robert Bruninga and was originally designed to track tactical information in real time. APRS operates on a frequency within the VHF 2-meter band and is popular for applications like location transponders or weather stations. You can view APRS activity in your area at www.aprs.fi right now.

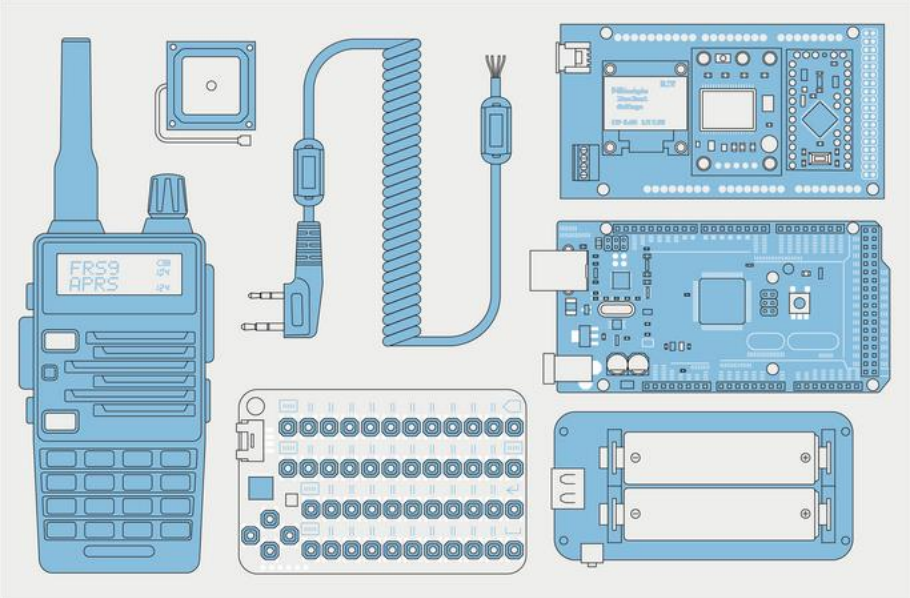
APRS supports sending text messages, and if you're in range of an Internet-connected gateway node you can even exchange SMS texts with cellphones and send one-line emails. Sending texts traditionally meant using a PC hooked up to a so-called terminal node controller (TNC) packet radio modem, which is in turn connected to a radio (signals are transmitted as audio tones, just like old dial-up modems). More recently, TNC modems that interface with smartphones have been created. And these are awesome projects! But at its core, HamMessenger was created in the shadow of my simple childhood experiences. I wanted a portable device I could connect to my handheld radio that was completely self-contained, with a keyboard, screen, and GPS receiver all built in.

First, I would need to nail down the hardware and software I was going to use. I found MicroAPRS, which is an open-source and Arduino-compatible firmware package for DIY packet radio modems. With MicroAPRS you can quicky implement a full-featured APRS modem with the ability to automatically switch the radio between receiving and transmitting.

I wanted a portable device that was completely self-contained, with a keyboard, screen, and GPS receiver all built in.

This was perfect. I could now focus on the rest of the HamMessenger. I thought about building it around a Raspberry Pi. It would have been cool, but a Pi is overkill. It would need a lot of power, and there's a risk of corrupting the filesystem if you don't do a controlled shutdown, a problem if the battery dies.

I decided on a dual Arduino approach. An Arduino Pro Mini (US $10) would act as the modem, running MicroAPRS and communicating with the rest of the system via a serial connection. An Arduino Mega 2560 ($40) would be the central controller, tying together the modem, keyboard, display, and GPS. Rechargeable batteries with a battery-management board would provide the power.



The HamMessenger is compatible with most handheld VHF radios [left] by using an adapter cable [top, middle] that connects to a printed circuit board with a display, GPS receiver, and Arduino Pro acting as a modem [top right]. The PCB plugs into an Arduino Mega [middle right], a GPS antenna [top left], a mini keyboard [bottom middle], and batteries [bottom right].James Provost

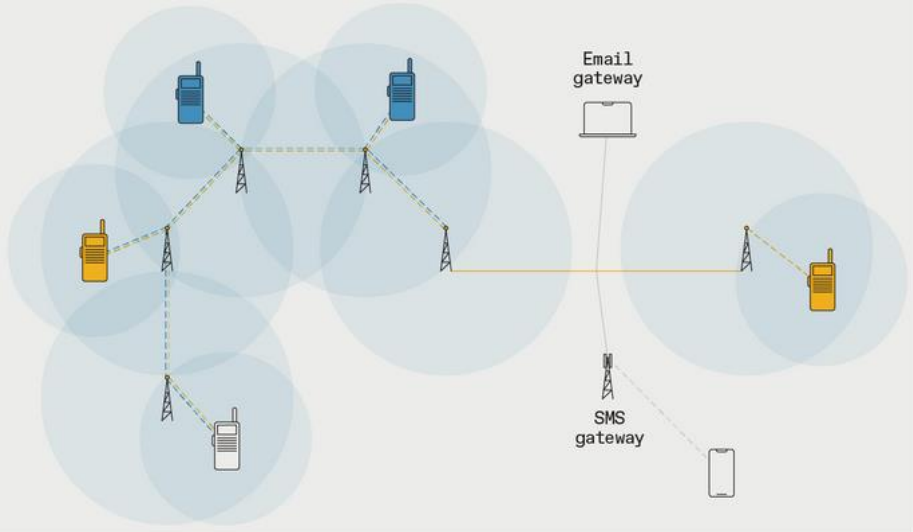
The GPS provides the location data that is integrated into most APRS transmissions. I chose a $10 NEO 6M-based GPS receiver that is popular with hobbyists for things such as DIY drones. Like my modem, the NEO has a serial interface.

In my initial design, the human input setup was very simple, with just three buttons. One button let me step through displayed menus and modify parameters, one button selected a submenu or set a parameter, and the last button let me cancel a parameter entry or navigate to a previous menu.

Ultimately, because of the difficulty of using the buttons to enter text messages, I replaced them with a mini CardKB QWERTY keyboard ($8.50). However, the limits of the three-button system forced me to simplify the HamMessenger's user interface as much as possible, something I am very thankful for now, as it means the HamMessenger is easy to operate with just a basic knowledge of APRS.

For the display, I chose an OLED screen for its power efficiency. The only drawback for hobbyist OLEDs is their small size. The 0.96-inch displays are the most common, but I was able to find a $9 1.3-inch display that communicates via an I2Cserial bus.

The final modular component I needed for the HamMessenger was some nonvolatile storage for received messages. I decided on a micro-SD card reader because they natively speak the SPI Interface protocol.



The Automatic Packet Reporting System relies on a backbone of digital repeaters, or digipeaters, that repeatedly retransmit messages sent by handheld and other radios. Other digipeaters that pick up the signal in turn will retransmit the message up to a specified number of hops. Some digipeaters are connected to the Internet, which allows the user to send messages to distant digipeaters or relay them as cellphone SMS messages or emails.James Provost

All of these feed into the Arduino Mega. The Mega was chosen for the central controller as it doesn't need a lot of power, yet has enough resources to handle all the different module connections—two serial, two SPI, and one I2C connection. (And then I added a third serial port so you can control the HamMessenger with a PC or other device using an ASCII-based API.)

I designed a shield (a printed circuit board that accommodates the modules and some supporting circuitry that simply plugs into the top of the Mega), using Autodesk's Eagle, and then used the shield design files to help create a 3D-printed enclosure in Fusion 360 (full details are available on the HamMessenger GitHub page).

Currently, the HamMessenger is still in a prototype stage, but it works well. I have a HamMessenger installed in my truck that doubles as a location beacon. It will never replace a cellphone for most people, of course, but those in places without coverage might find it useful. Still, it was primarily created as a way to promote electronics and alternative uses of amateur radio, and if you want an easy way to learn and blend these hobbies, then I think the HamMessenger is a great way to do that.

This article appears in the December 2021 print issue as "Phone-Free Texting."