QEX

A Forum for
Communications Experimenters

March/Arpil 2025
Issue No. 349 | www.arrl.org

The Raspberry Pi Zero Board and camera that were used in the Touchard High School's balloon project to demonstrate the idea of Gas Law.

KENWOOD

Versatile and Flexible Radio Returns with a New Look and Soaring Functions That'll Thrill Amateur Radio Enthusiasts.

NEW TRIBANDER

TH-D75A

144 / 220 / 430 MHz TRIBANDER

Key Features

- **APRS® Protocol[*1] compliant**
  To exchange GPS location data and messages in real-time.
- **D-STAR[*2] with Simultaneous Reception on DV mode**
  Compatible for transferring voice and digital data over D-STAR networks.
- Reflector Terminal mode to access D-STAR Reflectors
- USB Type-C for Data Transfer and Charging
- Built-in Digipeater (a digital repeater) station to transmit received data
- Built-in GPS unit
- Easy-to-read Transflective Color TFT Display
- Call Sign Readout
- Tough & Robust – meets IP54/55 Standards
- Wide-band and multi-mode reception
- Built-in IF Filter for comfortable reception (SSB/CW)
- DSP-based Voice Processing and Reputable KENWOOD Custom Tuned Sound Quality
- Bluetooth®, microSD/SDHC Memory Card Slot for flexible link with a PC

---

*1: APRS® (The Automatic Packet Reporting System) is a registered trademark of TAPR (Tucson Amateur Packet Radio Corp.)
*2: D-STAR is a digital radio protocol developed by JARL (Japan Amateur Radio League).
All other company names, brand names and product names are registered trademarks or trade names of their respective holders.

Specifications, design, and availability of accessories may vary due to advancements in technology. Actual product colors may differ from photograph due to photography or printing conditions.
Brand or product names may be trademarks and/or registered trademarks of the respective holders.

The content of this document is based on information available at the time of its publication and may be different from the latest information.
This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

JVCKENWOOD USA Corporation

Customer Support/Distribution Customer Support:
(310) 639-4200 Fax: (310) 537-8235 **www.kenwood.com/usa**
ADS#31823

QEX

QEX (ISSN: 0886-8093) is published bimonthly in January, March, May, July, September, and November by the American Radio Relay League, 225 Main St., Newington, CT 06111-1400. Periodicals postage paid at Hartford, CT and at additional mailing offices.

POSTMASTER: Send address changes to:
QEX, 225 Main St., Newington, CT 06111-1400
Issue No. 349

Publisher
American Radio Relay League

Lori Weinberg, KB1EIB
Assistant Editor

Production Department
Becky R. Schoenfeld, W1BXY
Director of Publications and Editorial

Matt Ali
Layout & Production Specialist

David Pingree, N1NAS
Senior Technical Illustrator

Brian Washing
Technical Illustrator

Advertising Information
Janet L. Rocco, W1JLR
Business Services
860-594-0203 – Direct
800-243-7768 – ARRL
860-594-4285 – Fax

Circulation Department
Cathy Stepina
QEX Circulation

Offices
225 Main St., Newington, CT 06111-1400 USA
Telephone: 860-594-0200
Email: **qex@arrl.org**

Subscription rate for 6 print issues:

In the US: $29

US by First Class Mail: $40

International and Canada by Airmail: $35

ARRL members receive the digital edition of *QEX* as a member benefit.

In order to ensure prompt delivery, we ask that you periodically check the address information on your mailing label. If you find any inaccuracies, please contact the Circulation Department immediately. Thank you for your assistance.

March/April 2025

About the Cover

Using a Raspberry Pi Zero Board and camera to demonstrate Gas Law. The aim of this project was to verify the law of perfect gases by estimating the volume of the balloon as a function of altitude. Anthony Le Cren, F4GOH/KF4GOH, explains in “Demonstrating the Idea of Gas Law Using a Balloon, SSTV, and LoRa.”

In This Issue:

Index of Advertisers

DX Engineering: .... Cover III
ICOM America: ....Cover IV
Kenwood Communications:........ Cover II
Tucson Amateur Packet Radio:............ 10

The American Radio Relay League

The American Radio Relay League, Inc., is a noncommercial association of radio amateurs, organized for the promotion of interest in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of the radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the state of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every three years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur," ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A *bona fide* interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general correspondence should be addressed to the administrative headquarters:

ARRL
225 Main St.
Newington, CT 06111 USA
Telephone: 860-594-0200
FAX: 860-594-0259 (24-hour direct line)

Officers

President: Rick Roderick, K5UR
P.O. Box 1463, Little Rock, AR 72203

The purpose of *QEX* is to:

1) provide a medium for the exchange of ideas and information among Amateur Radio experimenters,

2) document advanced technical work in the Amateur Radio field, and

3) support efforts to advance the state of the Amateur Radio art.

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main St., Newington, CT 06111 USA. Envelopes containing manuscripts and letters for publication in *QEX* should be marked Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be submitted in word-processor format, if possible. We can redraw any figures as long as their content is clear. Photos should be glossy, color or black-and-white prints of at least the size they are to appear in *QEX* or high-resolution digital images (300 dots per inch or higher at the printed size). Further information for authors can be found on the Web at **www.arrl.org/qex** or by e-mail to **qex@arrl.org**.

Any opinions expressed in *QEX* are those of the authors, not necessarily those of the Editor or the League. While we strive to ensure all material is technically correct, authors are expected to defend their own assertions. Products mentioned are included for your information only; no endorsement is implied. Readers are cautioned to verify the availability of products before sending money to vendors.

Bill Liles, NQ6Z, and Ward Silver, N0AX

Perspectives

QEX continues to present a wide-angle snapshot of ham radio interests in this issue, ranging from analog preselectors to vector network analyzers. LeCren's article also highlights a growing aspect of amateur radio — facilitating and conducting scientific research. In this article, students verify the ideal gas law in real-life, with ham radio providing the live photographs of a balloon-borne experiment. In recent years, the "value proposition" of ham radio has become apparent to the scientific research community, particularly for geomagnetic and ionospheric phenomena. Coordinated through the **hamsci.org** community, all manner of research experiments are being conducted by amateurs with an interest in supporting science. You can, too!

The process of interviewing and qualifying the applicants for the editorship of *QEX* has begun. A new *QEX* Editor is expected to come aboard later this year.

In This Issue:

- John Stanley, K4ERO, gives examples of NVIS antennas with descriptions and gain figures.
- Anthony LeCren, F4GOH/KF4GOH, explains how Computer Science and Networks high school students programmed an HF SSTV transmitter to send images from a high-altitude baloon.
- George R. Steber, WB9LVI, describes a new version of the classic RF preselector that improves the front-end selectivity of your shortwave receiver.
- Mario Lechasseur, VE2KEC, presents an Arduino-based control unit that simplifies band and mode changes on the FT-0891.
- In his essay series, Eric Nichols, KL7AJ, discusses measuring components with a VNA.

Writing for *QEX*

QEX is a forum for the free exchange of ideas among communications experimenters. *QEX* is published bimonthly.

Please continue to send full-length *QEX* manuscripts, or share a Technical Note of several hundred words in length plus a figure or two, to **qex@arrl.org**. We pay $50 per published page for full articles and *QEX* Technical Notes. Get more information and an Author Guide at **www.arrl.org/qex-author-guide**. If you prefer postal mail, send a business-size self-addressed, stamped (US postage) envelope to: *QEX* Author Guide, c/o Maty Weinberg, ARRL, 225 Main St., Newington, CT 06111.

John Stanley, K4ERO
524 White Pine Ln., Rising Fawn, GA 30738 | **jnrstanley@gmail.com**

Extreme NVIS Antennas — How Big Can We Go?

Examples of NVIS antennas with descriptions and gain figures.

NVIS (Near Vertical Incidence Skywave) antennas are something that many are talking about these days. YouTube is full of advice and results videos. Some of these are useful and some not so much. *QST* articles and the Wikipedia NVIS article are a better source than most of the videos.

In my amateur and professional career, I have often needed an antenna that would produce a strong HF signal at a high angle. I have made contacts on 1.8 through 14.35 MHz that were in the distance range of 100 – 300 miles, consistent with skywave propagation and thus NVIS. In my professional career, I have designed a number of shortwave broadcast antennas used to cover a range out to a few hundred miles, and these called for the use of the 90, 60, 49, and 41-meter bands (3.2 to 7.4 MHz) and an NVIS antenna. I worked on and with a large NVIS array at the HIPAS (High Power Auroral Stimulation) ionospheric lab near Fairbanks, Alaska, which was used on frequencies below 7 MHz to “heat” the ionosphere. This was operated by the Plasma Physics Lab at UCLA and conducted many experiments on ionospheric modification. We also used ionosondes which used special NVIS antennas to study the results of those experiments.

I have designed, built and used NVIS antennas all the way from a simple dipole close to the ground, up to a large array of 8 dipoles with a gain of 13 dBi. Some of these “extreme” NVIS antennas may be only of passing interest to amateurs who use NVIS for shorter range net operations or just general operating, but amateurs may want to try some of the less extreme ones. I will give examples of some of these antennas along with some physical details and gain figures. I can provide more details to prospective builders.

A list of NVIS antennas I have designed, built, operated, or been involved with.

- Low dipole, the often-recommended low NVIS antenna. It “works” but not very well.
- Higher dipole, a half wave dipole at 0.2 λ above ground, optimum transmit height for NVIS. An inverted V, 0.2 λ above ground at the center is essentially the same thing.
- An end-fed half wave is identical at its fundamental, but not good for NVIS at harmonics.
- Full-wave loop, a “lazy quad,” full-wave loop either square or triangular.
- Extended double Zepp, a 1.2 wavelength doublet, center fed with ladder line.
- Two collinear dipoles, separately fed at the center of each.
- Two parallel dipoles, separately fed at the center of each.
- Double full wave delta loop, two delta loops spaced 1 λ center to center.
- “Very lazy” H, 4 dipoles, spaced end to end and side to side.
- Double lazy H, 8 dipoles, 4 dipoles in parallel with another set end to end.

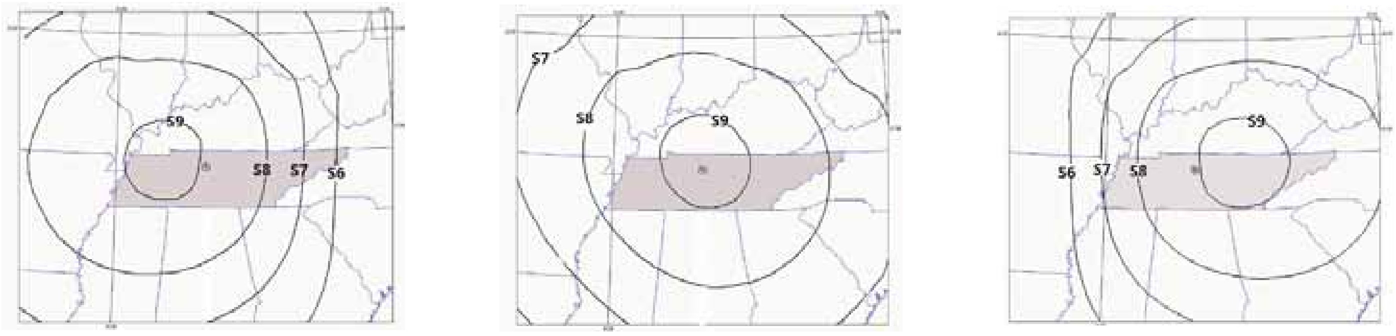


Figure 1a – Nashville antenna slewed west.

Figure 1b – Nashville no slew in antenna.

Figure 1c – Nashville antenna slewed east.

Figure 1 – Coverage footprints for a double dipole on 80 meters on a late afternoon in spring. At night the patterns would be much larger but show a similar shifting of coverage due to slewing the pattern. The greater the gain (narrower NVIS lobe), the more important slewing becomes. You don’t have to aim a flood light, but you do have to aim a spotlight. Of course, the spotlight is much brighter. The contours show relative S meter values based on 6 dB per S unit. Actual values will depend on power levels and time of day.

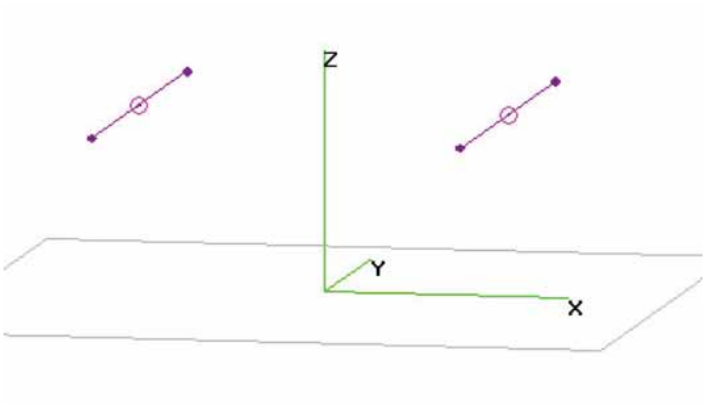


Figure 2 – Two half wave dipoles separately fed.

- HIPAS, a circular array of 8 crossed dipoles, 7 in a circle and one at the center.
- HAARP (High-frequency Active Auroral Research Project) with the same total transmitter power as HIPAS, but slightly higher ERP due to its more closely packed 48- element antenna array. HAARP can be continuously tuned between 3.1 – 9 MHz, gain is a function of frequency. It can be considered the king of NVIS antennas.

(The Jicamarca array in Chile has even more gain, but operates at around 49.9 MHz, well above the normal NVIS frequencies. The main antenna is a dual polarized antenna array that consists of 18,432 half-wavelength dipoles occupying an area of 1000 × 1000 feet. With 6 MW of transmitter power beamed directly upwards it produces reflections from the overhead atmosphere, even though only a tiny fraction of the power is reflected. Other "radio telescopes" produce even more gain, but are typically operated well above the NVIS frequencies. They can aim their very narrow beam straight up or at lower angles.)

What is "Slewing"?

More complex NVIS antennas can be "slewed." Slewing an antenna pattern means aiming it in a direction either side of the normal beam by changing the phase to two or more parts of the antenna. On HF typically slewing is limited to steering the beam up to 30 degrees. With NVIS this moves the area on the ground that is covered to one side or the other. **Figure 1** shows coverage patterns of an unslewed and a slewed antenna. A station near Nashville could shift its NIVS coverage to favor the eastern or western half of Tennessee by throwing a switch. How is this accomplished?

This example, **Figure 2**, uses an antenna consisting of two half wave dipoles spaced a half wave apart. If they are fed in phase, they produce a very round pattern. Looking at the vertical pattern from the end of the dipoles, we can see that the pattern is very clean, and the maximum signal is straight up.

If we feed one of the dipoles 90 degrees out of phase, the vertical pattern can be seen to be shifted to one side. See **Figure 3** and **Figure 4**. The coverage contours of the two cases are in Figure 1. The coverage shifts toward the side of the delayed dipole. This could be very useful in a case where the transmitter is not located in the exact center of the area that we desire to cover.

Slewing can also be useful in preventing interference from an undesired area, both in transmit and receive.

Relative Size of Various NVIS Antennas

The HAARP array is not shown. It would be too large to fit on this page. All of these are shown as viewed from above, that is a "bird's eye" view. Height above ground is a constant 0.2 wavelength unless otherwise noted. See **Figure 5**.

Relative Size of Various NVIS Antennas

Type of Antenna	Size for 80 Meters	Pattern and Slewing - see text for discussion.	Relative Gain
Low ½-wave dipole, end or center fed.	120 feet, 7 feet high. See birdseye 1.	Nearly round. No slewing.	–10dB*
High ½ wave dipole, straight or inverted V, end or center fed.	120 feet, 50 feet high at center. See birdseye 1.	Nearly round. No slewing.	0 dB (Reference)
Full wave loop, (lazy quad).	60 by 60 feet, 50 feet high. See birdseye 2.	Very round. No slewing.	0.6 dB
Extended double Zepp.	280 feet long, 50 feet high. See birdseye 3.	Oval, long axis broadside. No slewing,	2.5 dB
Collinear dipoles, individually fed.	280 feet long, 50 feet high. See birdseye 4.	Oval, long axis broadside. Slew along the short axis.	3 dB
Hourglass.	50 by 220 feet, 50 feet high. See birdseye 5.	Oval, along long side. No slewing.	2 dB
Two parallel dipoles.	120 by 120 feet. See birdseye 6.	Nearly round. Slew broadside to dipoles.	2.5 dB
Double delta loop.	70 by 180 feet. See birdseye 7.	Oval, long axis along line. Slew along long axis.	3 dB
Lazy H, (four dipoles).	120 by 240 feet. See birdseye 8.	Very nearly round. Slew along short axis.	5 dB
Double Lazy H, (8 dipoles).	360 by 240 feet. See birdseye 9.	Very round. Slew in any direction.	8 dB
HIPAS Array, 16 dipoles, crossed dipoles in pairs.	500-foot diameter circle. See birdseye 10.	Very round. Slew in any direction.	9 dB**
HAARP array 240 dipoles.	40 acres full of dipoles, too big to be shown.	Very round. Slew in any direction.	18 dB

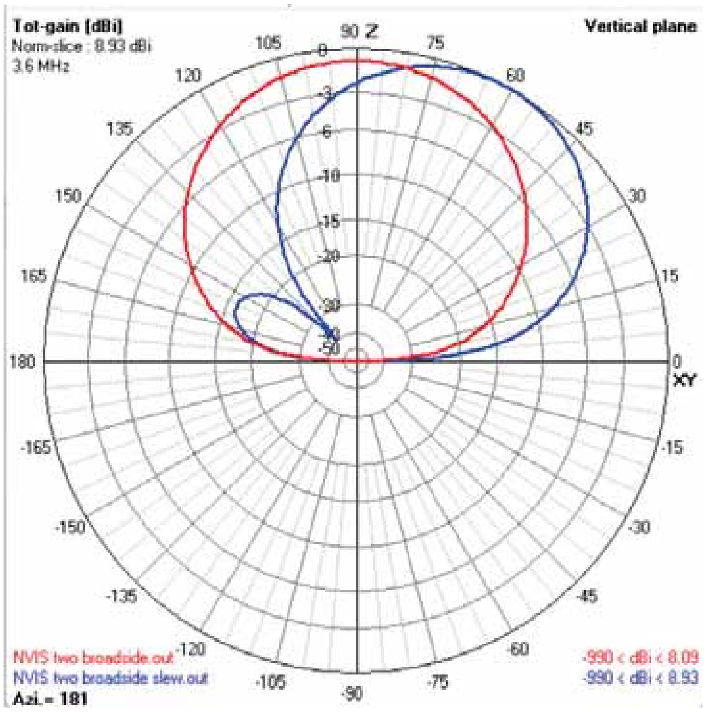


Figure 3 – Slewed and unslewed patterns.

Propagation Factors in NVIS Operation

Some presentations on NVIS underplay the most important part of NVIS operation, that is, the effect of the ionosphere. Yet NVIS is utterly dependent on the ability of the ionosphere to reflect the waves that go up at Near Vertical Incidence (the NVI in NVIS). If the high angle waves are either strongly absorbed, or simply pass though into space without reflection, *no* antenna will produce NVIS contacts. On the other hand, on the frequencies where the absorption is relatively low and the ionization sufficient to reflect the vertical incidence (high angle) waves, NVIS always occurs. You simply cannot prevent it. You can use an antenna that has very little high angle radiation, such as a vertical quarter wave, but even then, some NVIS will occur. The answer to the question "Have you ever operated using NVIS?," is "Yes, I have used frequencies below 10 MHz." So, for those who think that NVIS is *hard* to use, I want to say that *no*; rather NVIS is *hard to avoid*, unless you never use the lower bands. The amateur bands most useful for NVIS are 3.5 through 7.3 MHz. The best band will depend on time of day, sunspots, and seasonal factors. 2, 10 and 14 MHz may occasionally be useful as well.

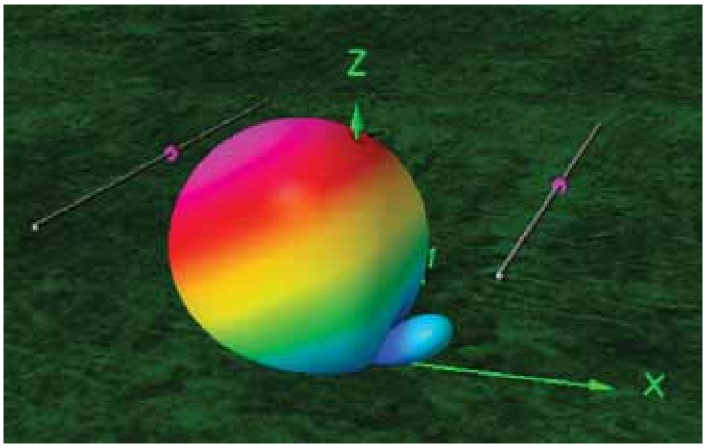


Figure 4 – A 3D view of the slewed pattern.

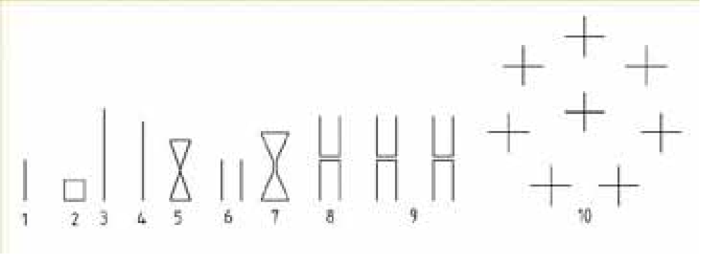


Figure 5 – Relative size of various NVIS antennas. (Birdseye view.)

*Low Dipoles

Many have advocated dipoles only a few feet above ground as the best NVIS antenna. Yet extensive studies by the Army **[1]** and a Dutch research group **[2]** have shown that the low dipole has a lot of loss compared to one at .15 to .25 wavelengths, about 40 to 65 feet for 80 meters, or 20 to 33 feet for 40 meters. An antenna used on both bands should be 30 to 40 feet up. The loss on receive is not as important as both signal and noise drop down. That contacts can be made with these very low antennas proves nothing except that NVIS signals can be strong enough so that even putting 90% of your power in the dirt will still provide communications. So, while in a hastily set up portable station, the dipole only six feet off the ground may "work," for any permanent site every effort should be made to get the antenna up to the optimum height to avoid extreme ground losses. This also prevents danger from the radiation and touching a working antenna and getting RF burns. We should characterize these low antennas not as cloud burners, but as dirt burners, and worst case people and pet burners. They are a serious compromise.

**Crossed Dipoles

You will see that the HIPAS antenna uses crossed dipoles. This allows the upgoing wave to be circularly polarized. While the experiments done at HIPAS benefited from this ability, it is not recommended for ham use. When other stations are using a dipole, transmitting using circular polarization can enhance your signal by 3 dB, but can reduce received signals by up to 20 dB. Changing the "sense" from left to right hand circular polarization will reverse these effects. Furthermore, sometimes when the X mode (extraordinary ray) predominates, a further reversal takes place. Thus, one might need to switch the sense between receive and transmit and even for different stations in a round table or net. Even then it will not be optimum for every station and time. I have used this technique on the 60-meter band and concluded that 3 dB isn't worth all the switching. But if you are adventurous, go ahead and try it.

References

[1] https://arrl-ohio.org/wp-content/uploads/2024/08/G0TJD_NVIS_Communication.pdf

[2] https://ris.utwente.nl/ws/portalfiles/portal/12174288/2014_Witvliet_NVIS_Research_in_The_Netherlands_summary_URSI_Benelux.pdf

Dr. George R. Steber, WB9LVI
11105 N. Riverland Ct., Mequon, WI 53092 | **steber@execpc.com**

A Low-Cost Tunable RF Preselector Using a Varicap Diode

A new version of the classic RF preselector that improves the front end selectivity of your short wave receiver.

A tunable RF preselector can be a useful addition to an HF receiver. It reduces adjacent channel interference by providing a bandpass function for the band of interest. Tuning to the desired band is done by adjusting a single variable capacitor. But many preselector designs require an expensive mechanical rotary capacitor for tuning. Described here is a straightforward passive design that replaces the rotary capacitor with a low-cost varactor, often called a *varicap*.

Operating homebrew ham receivers, like the NanoSSB RX described in *QEX* Nov/Dec 2021, inexpensive commercial shortwave sets like the low cost ATS20 SSB receiver, or software-defined radios (SDR) can be fun but also challenging. Often, in the interest of receiver design simplicity, the benefits of good front-end RF selectivity may have been overlooked.

For these cases adding a bandpass RF filter between the antenna and receiver as shown in **Figure 1** can help a lot. Filtering signals from the antenna provides for rejection of strong out-of-band signals which otherwise might overload the input. This filter can also prove to be highly beneficial in situations where there is a lot of noise or interference; for example from a local computer.

If you're only interested in a single band, say 30 meters, a fixed RF filter for that band is easily constructed. But if you like to hop around on the bands, when listening to shortwave, a tunable preselector may be the way to go. Back in 2018, I presented a design in *QEX* (Jan/Feb 2018, pp: 11-14) for a very selective tunable RF preamplifier. It was enormously popular as attested to by the volume of email received over time. That design used a varicap tuning element and a two stage buffer amplifier.

Over the years many have wondered if it were achievable to design a passive (without amplifier), tunable preselector. As this article shows, it is possible to build a passive varicap tuned preselector that has many of the characteristics of the amplifier version in the Jan/Feb 2018 *QEX* article. Downsides of this design are that it is not as selective as the amplifier version and has an insertion loss of a few dB. For many cases, like the NanoSSB or ATS20 receiver, where there is adequate RF amplification, these are not important factors. These radios will perform better with a preselector by rejecting extraneous signals that would cause the AGC to reduce the desired signal. A big upside for this design is that it requires few parts and is easy to build, even without a PCB.

I'll show you a design that covers the frequency range of about 2.5 MHz to 12.0 MHz. But other ranges are possible, and I'll show you how to change component values to cover the band of frequencies of your own interest. I'll also show you an *LTspice* (**www.linear.com/designtools/software**) simulation and spectrum analyzer view that proved useful in verifying the operation. A single 9V battery is used for operation. As noted above, this design uses a variable capacitance diode, a varicap. This tunable preselector should find use in many interference situations.

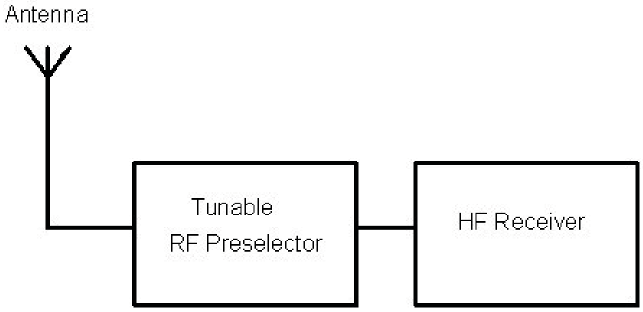


Figure 1 – A tunable RF preselector is placed between the antenna and the receiver.

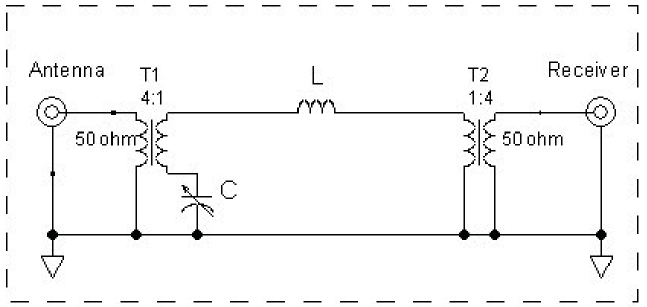


Figure 2 – Schematic of the tunable preselector.

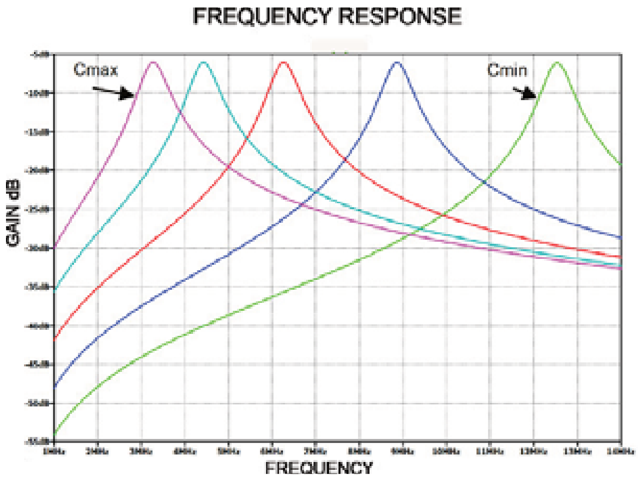


Figure 3 – Five frequency response curves of the tunable preselector as capacitor C is varied.

So, if you would like to learn more about this passive, varicap-controlled preselector and find out how it was designed, read on. Building your own version might be just the ticket for you.

RF Preselector Introduction

A classic tunable circuit that is often used to provide RF selectivity is shown in **Figure 2**. It is not the same as an *antenna tuner*, which is used to match the impedance of your antenna/feedline to your receiver or transmitter. The tuner shown here is basically an adjustable *bandpass filter*. It uses capacitor C and inductor L as a resonant circuit. As C is varied the center frequency or peak of the filter is changed. Wideband transformers T1 and T2 are used for impedance transformation which creates a sharper filter. Shown here are 4:1 transformers. But 9:1 or even higher ratio ones may be used to increase the sharpness of the bandpass response, but at the cost of more insertion loss.

The peak in the RF response is moved over the frequency range by adjusting the variable capacitor C. Response curves for L=6.4 µH and various values of C (30 pF to 365 pF) are shown in **Figure 3**. They were taken from an *LTspice* simulation and have been found to closely follow those seen with a spectrum analyzer on the actual circuit.

Using a mechanical variable capacitor would normally be the way to go, but rotating capacitors are becoming harder to find. This is because they are a precision piece of hardware and expensive to manufacture. Moreover, they have been replaced by digital circuitry in many modern designs. Nonetheless they can still be found on surplus or auction sites, but the price is often quite high.

Fortunately, there is another way to do the tuning. Use a varicap! In the next section we'll talk about the varicap and how it can be used in this application.

Using A Varicap

A varicap is essentially a *voltage-controlled* capacitor. They have been around since the 1960s and are commonly used in voltage-controlled oscillators, parametric amplifiers and frequency multipliers. It is a type of solid-state diode.

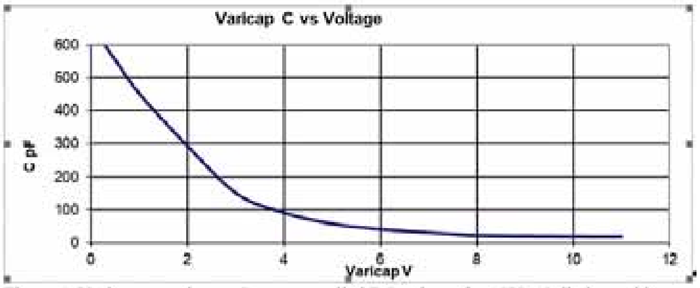


Figure 4 – Varicap capacitance versus applied dc voltage for the 1SV149 diode. Data was taken from manufacturer's data sheet.

Here's how it works. When a diode is operated in a reverse biased state, very little current flows in the device. The effect of applying the reverse bias voltage is to control the thickness of the depletion zone and therefore its p-n junction capacitance. The greater the applied voltage, the greater is the depletion zone and the smaller the capacitance. Most diodes exhibit this characteristic to some extent but varicaps are manufactured to exploit this effect and increase the capacitance over a larger range.

The varicap used in this project (1SV149) has a very large capacitance variation and was designed to replace the tuning capacitor in AM radios. It has high Q (200 min.), a small package, high capacitance ratio, and low voltage operation. And it can be found on the internet for under a dollar! The varicap capacitance variation for the 1SV149 with respect to applied dc voltage is shown in **Figure 4**. Notice that above about 9V, there is little variation of capacitance with voltage.

Substituting a varicap for a mechanical variable capacitor requires a bit of planning. We need to know the capacitance range covered and the tuning voltage required. The tuning curves shown in Figure 3 covered a capacitance range of 30 pF to 365 pF. So, we would like to find a varicap that covers that range with a

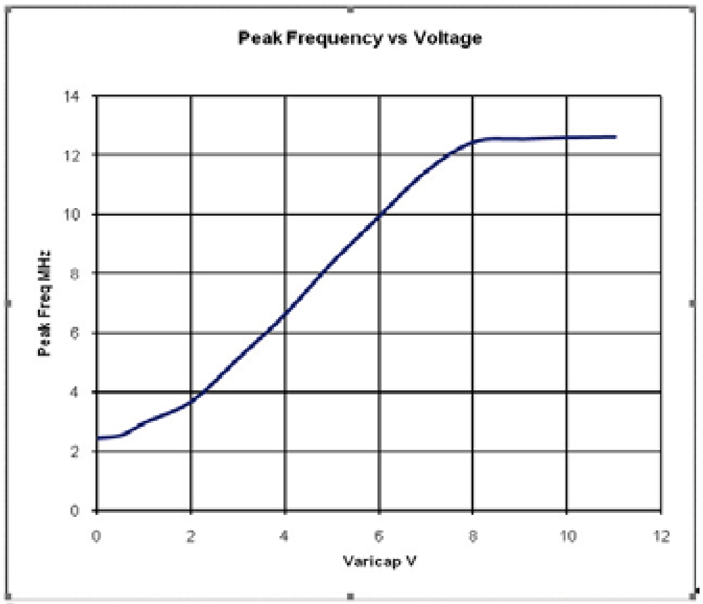


Figure 5 – Variation of peak frequency with varicap voltage. Data is taken for the 1SV149 used in this project. The frequency range is about 2.5 MHz to 12.5 MHz.

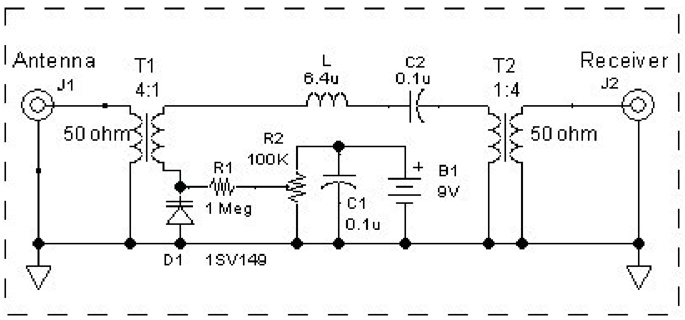


Figure 6 – Final schematic for the tunable preselector with varicap D1 and inductor L. C1 is for RF bypass. C2 blocks dc current.

reasonable voltage bias range. The 1SV149 varicap covers 25 pF to 500 pF with a voltage range of less than 10 volts.

When the 1SV149 varicap is substituted for the mechanical capacitor using an appropriate circuit, the *tuning law* shown in **Figure 5** is obtained. As is seen, the varicap easily covers the same range as the mechanical capacitor and more – in this case 2.5 MHz to 12.5 MHz.

Tunable RF Preselector Design Notes

This tunable RF preselector can be built for around $25 or less, depending on your parts inventory, and does not require extraordinary skill with RF circuits. It makes a nice weekend project once the material is on hand. It is unique in some aspects, like manual tuning the peak frequency. However, it will certainly not compete with an expensive transceiver. But it does produce good results in many situations. It made my low cost ATS20 SSB receiver, mentioned earlier, perform significantly better.

Figure 6 shows the complete schematic of the tunable preselector. It looks very much like that of Figure 2 and operates the same way. This circuit was designed using the *LTspice* simulator before construction started. This will be discussed in more detail later on. After the breadboard circuit was built, it was analyzed using a spectrum analyzer with tracking generator. Very close agreement was found between the *LTspice* simulation and the hardware.

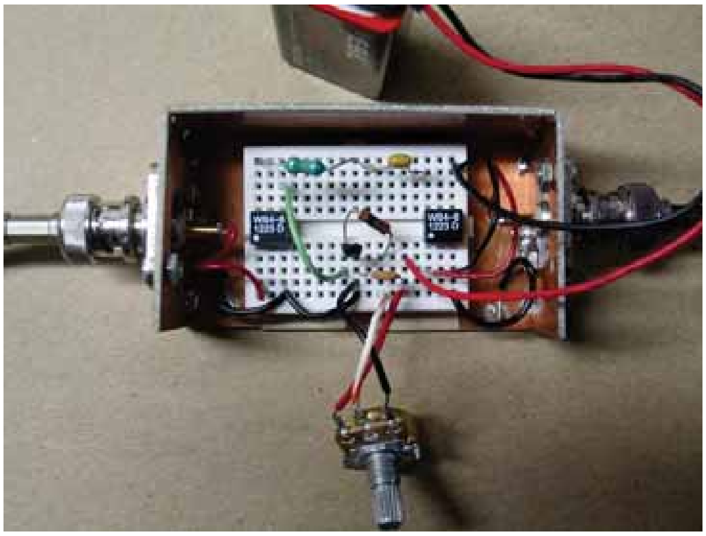


Figure 7 – The tunable preselector breadboard.

Table 1 – Preselector components

Qty	*Label*	*Value*	*Description*
2	C1, C2	0.1 µF	ceramic capacitor
1	L	6.8 µH	Mouser 77F6R8K, inductor
1	R1	1 Megohm	1/4w 5% resistor
1	R2	100 K ohm	Potentiometer
1	D1	1SV149	Varicap
1	T1, T2	4:1 RF transformer	Coilcraft WB4-6L
2	J1, J2	Connector	BNC connector

Here's how it works. The inductor L and varicap D1 form a series resonant circuit between the antenna and the receiver. Note that the varicap is essentially a capacitor, but a variable one. So, the resonant frequency is determined by its value. C2 is provided to block the dc from the varicap circuit. If C2 is made much higher than capacitance of varicap then the varicap capacitance dominates since they are in series. C1 is used to bypass any RF that might find its way into the circuit. R2 is adjusted to tune for the peak frequency desired. Try to obtain a R2 that rotates over a large angle (320 degrees) as this will provide finer control over your tuning. R1 feeds the dc voltage from R2 and the 9V battery to D1 and is large value to avoid loading the LC circuit.

The 9V battery is an EBL 9V lithium-ion rechargeable battery with Micro USB port, model 6F22. Yes, I said the battery has a USB port! It charges from a standard 5V USB port but provides 9 volts. It is rated at 5400 mWh. This is a lot of energy and provides many days of operation since the circuit draws very little current. Regular 9V carbon batteries will also work.

It is assumed that we are using a nominal 50 Ω antenna and 50 Ω input to the receiver. The resonant LC circuit alone would work here but the bandpass would be broad. To sharpen the bandpass, the resistance in the LC loop is lowered. This is done using T1 and T2, shown here as 4:1 RF transformers (Coilcraft WB4-6L). An even sharper bandpass can be obtained using a higher ratio, like 9:1. The transformers introduce some insertion loss, typically 0.5 dB each.

BNC connectors were used for J1 (antenna) and J2 (output) as they matched my equipment. Other connector types may be used. Table 1 lists the necessary components.

Figure 7 shows my first breadboard. It was constructed inside a homebrew case made from copper PC boards. With a sharp eye you can see the small varactor in the center and the RF transformers on the left and right sides of the solder-less board. The battery is at the top and potentiometer at the bottom. This circuit board worked fine but may not be the best approach for RF circuitry. Shown in **Figure 9** is the output of the circuit from my Advantest R3261spectrum analyzer with tracking generator. Two peaks are shown corresponding to the extents of the varicap tuning range.

Various techniques may be employed to build this preselector. Use your own ingenuity! Some good construction methods may be found in previous *QEX* articles or one of the ARRL *Handbooks*.

Figure 8 shows the actual measured performance. The two peaks correspond to the extents of the varicap tuning range.

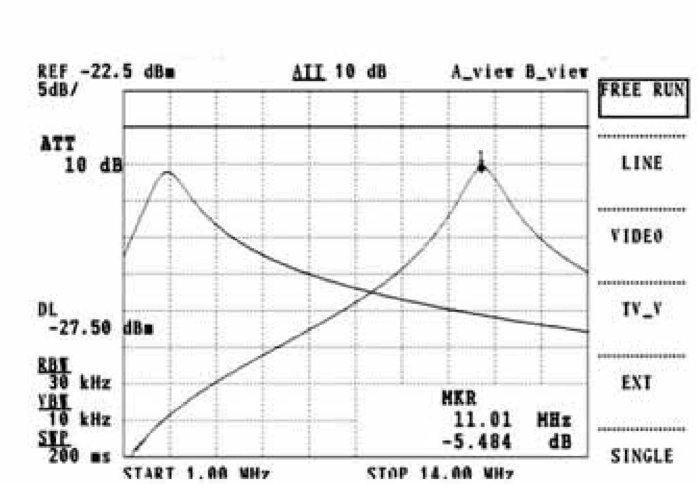


Figure 8 – Photo of Advantest R3261 spectrum analyzer with tracking generator. Peaks at extreme of tuning are shown.

Tunable Preselector Range Modifications

The tuning range *measured* for the circuit given here was roughly 2.5 MHz to 12.0 MHz. This range can easily be changed. It is mainly dependent on the inductance of the inductor L and varicap C. But it is difficult to predict the range exactly. This is because inductors may be constructed slightly differently, the stray capacitance of the circuit is unknown and the varicaps have a large tolerance.

First, let's take a look at the problem *theoretically*. The peak frequency (f) occurs at resonance; $f=1/(2\pi\sqrt{(LC)})$.

Calculating the tuning range for L= 6.4 µH (manufacturer value is 6.8 µH) and two values of C (500 pF and 30 pF) yields a range of 2.8 MHz to 11.5 MHz. This is different from the measured values – most likely due to inductor values being slightly different, stray capacitance and the varicap.

So, for the tuner scale to be accurately calibrated, you need to measure the response after building it. A suitable RF generator would prove useful in this case. Note that for general operating, this need not be done. With AGC off, tuning by just listening for the noise peak in the audio, works well. With that caveat, here are some ranges that were obtained for my circuit.

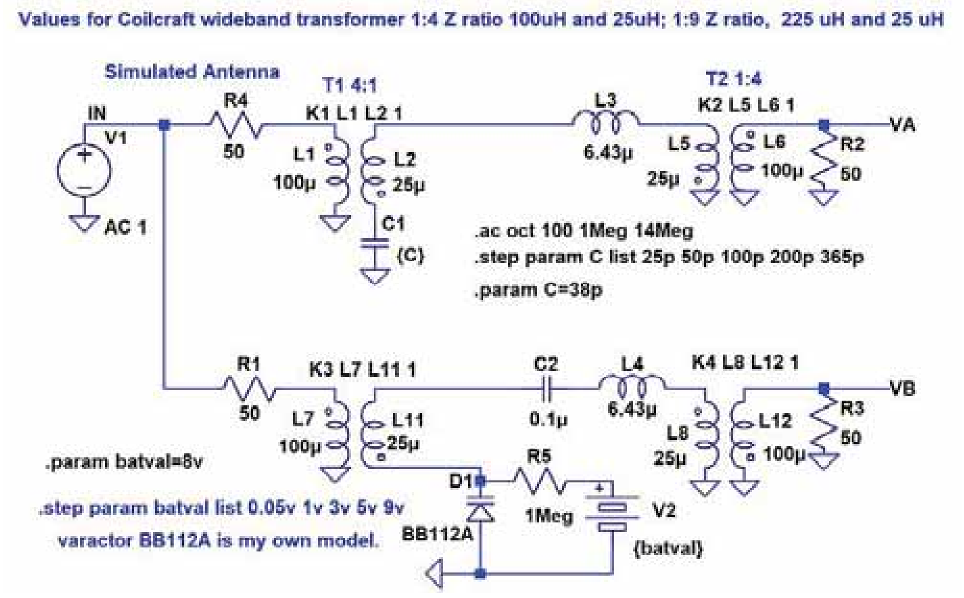
Choosing L= 2 µH covers 5 MHz to 20 MHz. Going lower with L=20 µH covers 1.6 MHz to 7.9 MHz. You can play with inductor values to get many ranges. You may want to consider putting in a range selector switch to cover the ranges you want.

LTspice Simulation

My *LTspice* simulation is shown in Figure 9. It is actually two simulations. The top part is the circuit without the varicap and the lower part includes the varicap. They can be run by enabling the 'step' parameter. Enable the 'step param C list' for the top simulation and enable 'step param batval list' for the varicap simulation. A good model for the varicap was not found in the literature, so my own model was developed. It is shown below:

```
.MODEL BB112A D ( IS=1e-14
RS=0.0011 CJO=6.5e-10 VJ=0.75 TT=5n
M=1.15
 BV=100 XTI=3 FC=0.5 IBV=0.0001
mfg=Philips_GRS type=varactor)
```

This model, BB112A, needs to be added to the diode library. It is not perfect but does emulate the working of a varicap diode. You can obtain a plot of this varicap by using the 'capometer' in *LTspice*. If you encounter debugging problems with *LTspice*, you may wish to visit the *LTspice* user group at **LTspice@groups.io**. For those interested, my *LTspice* 'asc file' will be available at **www.arrl.org/QEXfiles**.



```
.ac oct 100 1Meg 14Meg
.step param C list 25p 50p 100p 200p 365p
.param C=38p
.param batval=8v
.step param batval list 0.05v 1v 3v 5v 9v
```

Figure 9 – *LTspice* models of two passive preselectors. Top is the mechanical version. Lower is the varicap version.

Conclusion

The RF tunable preselector can prove helpful in a variety of interference situations. Even though its design is simple, the varicap works well and is a good substitute for the mechanical variable capacitor. One nice benefit of using the varicap is that the variable resistor R2, controlling the dc voltage, has a larger angle of rotation (320 degrees) than the mechanical capacitor (180 degrees) and therefore has finer control. The varicap also has a larger range. But it is nonlinear as shown by the plots.

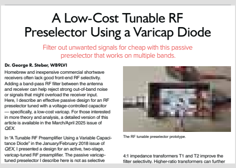
Using the preselector is straightforward. Simply adjust the front panel-tuning resistor R2 until the received signal is the loudest, usually with the AGC off. If you are using decoding software such as

WSJT-X FT9 or Fldigi that have a spectrum analyzer display you will see a pronounced increase in the signal as you approach the peak. Tune slowly. Remember to turn AGC back on.

It was enjoyable constructing and using this preselector. Hopefully you will experience that too. Building projects like this are fun, but they can also be educational. Consider demonstrating this application of a varicap in a teaching situation such as a presentation at your local ham club or possibly as an outreach tool at your local high school science class.

When you get your RF preselector working, please let me know about it. With your radio interference reduced, you may now be able to copy those weak signals from the far distant regions of the world.

Figures and captions provided by the author.



A Low-Cost Tunable RF
Preselector Using a Varicap Diode

Filter out unwanted signals for cheap with this passive preselector that works on multiple bands.

Dr. George R. Steber, WB9LVI

Homebrew and inexpensive commercial shortwave receivers often lack good front-end RF selectivity. Adding a band-pass RF filter between the antenna and receiver can help reject strong out-of-band noise or signals that might overload the receiver input. Here, I describe an effective passive design for an RF preselector tuned with a voltage-controlled capacitor — specifically, a low-cost varicap. For those interested in more theory and analysis, a detailed version of this article is available in the March/April 2025 issue of *QEX*.

In "A Tunable RF Preamplifier Using a Variable Capacitance Diode" in the January/February 2018 issue of *QEX*, I presented a design for an active, two-stage, varicap-tuned RF preamplifier. The passive varicap-tuned preselector I describe here is not as selective

The RF tunable preselector prototype.

4:1 impedance transformers T1 and T2 improve the filter selectivity. Higher-ratio transformers can further

A companion article also by George Steber, WB9LVI, appears in the April 2025 issue of *QST*.

George R. Steber, Ph.D., is Emeritus Professor of Electrical Engineering and Computer Science at the University of Wisconsin-Milwaukee. He is now retired, having served over 35 years. George, has an advanced class license, is a life member of ARRL and IEEE and is a professional engineer. He has also worked for NASA and the USAF. His last article for ARRL was "NanoWSPR — An Ultra-Low-Cost Multiband Transmitter" in the January 2022 issue of QST. *George also penned an article on the amazing story behind "The Discovery of Radio Waves" in the January/February 2019 issue of* Nuts and Volts Magazine. *In the article "Dark Energy and the Expanding Universe" in the March/April 2019 Issue of* Nuts and Volts *he showed experimentally how expanding space can have constant energy density. And in an atypical article he described how to use an ordinary PIN diode as a radiation detector in "Searching for Gold With X-Rays" in the July 2022 issue of* Circuit Cellar.
George currently has a book, "Invisible Visitors From Outer Space — The Story of Cosmic Rays," on Amazon books. It is written for beginners and veterans interested in science. It discusses many current cosmic ray projects on Earth as well as the historical events leading to their discovery. The new field of cosmic neutrino astronomy is also discussed.
George still lectures occasionally on science and engineering topics at the University. He is currently involved in various projects. When not dodging protons, pions and muons, he enjoys WSPR/JT9 amateur radio, astronomy, and jazz. You may reach him at **steber@execpc.com** *with "Preselector" in subject line and email mode set to text.*

Mario Lechasseur, VE2KEC
917 rue Arthur-Buies, Ste. Julie, QC, Canada | **lecm26000@gmail.com**

Yaesu FT-891 Control Box

An Arduino-based control unit that simplifies band and mode changes on the FT-891 transceiver.

A newcomer, in 2023 I joined the ham radio community and shortly after that I acquired my first rig, the FT-891. At first, I was using it exclusively for FT8. Using *WSJT-X* for FT8 does pretty much everything for you: band changes, mode, settings, etc. But when I started to do some listening on 80 and 40 meters, changing mode and band became an issue. The FT-891 has no dedicated controls for band change and mode change. To change bands for example, it is necessary to push a button quickly and then rotate the main tuning dial until you land on the desired band. You must also act quickly because the radio will simply select the band the cursor is on, which may not be the one you want.

Installing rig control software, *Commander V15.7*, on my PC made a big difference. I could enter the frequency using the numerical keypad of the PC. Band and mode changing was now done by moving the mouse. It's when I became tired of having to use the mouse that the idea of designing a control box came to me.

I felt that I would be better off with a control box having a 1 meter-long cable than continuously using a mouse on a pad. So, I downloaded the FT-891 CAT (Computer Aided Transceiver) operation reference book and ran some tests with terminal emulation software. After having sent many control commands to the rig, I felt confident that I could design my own FT-891 control box as shown in **Figure 1**.

The FT-891 is not a base transceiver, but the control box I designed does an honest job of providing base-style functions. It has a keypad for direct frequency input, a dedicated band button, and a dedicated mode button. It also has a QMB (Quick Memory Bank) recall and save button. Three rotary encoders were included: one to adjust the RF power, a second one to adjust the DNR level (Digital Noise Reduction), and a third one to adjust the IF Width.

This little box will not solve all the issues the experienced ham operators have with the FT-891. For example, it does not save all your settings when changing bands, nor will it transform the FT-891 into a mobile-style rig. But it will alleviate some of the steps for basic operations.

Figure 1 — This Yaesu FT-891 control box has a 4×4 keypad for direct frequency input, a dedicated band button and a dedicated mode button.

Methodology

The heart of this project is the Arduino UNO R3 platform. The control box and the FT-891 are connected to a PC through serial COM ports. The Arduino is powered directly through the USB port of the PC, so no external power supply is needed. No circuitry or special cables are required, eliminating the chance of damaging the FT-891 during project development. To link the control box and the FT-891, software needs to be installed on your PC.

Along with the FT-891, here are the pieces of hardware you will need :

- 1 × PC running Windows 10
- 1 × Arduino UNO R3 board
- 2 × 1-meter Type A to Type B USB cables
- 1 × 4×4 keypad (0-9, *, #)
- 3 × Rotary encoders (Use a KY-040 or equivalent)
- 25 × male to female breadboard jumper wires

Arduino is an open-source electronics platform well known by hobbyists. The Arduino board will be used to decode a keypad and rotary encoders to communicate with the CAT interface of the FT-891.

A CAT interface provides a connection between a PC and a radio, usually through a USB COM port. CAT software programs send control commands to the rig to change bands, frequency etc.

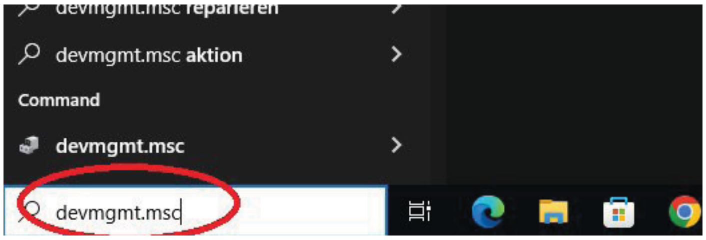


Figure 2 — Here is how to get to the **Windows 10 Device Manager** on your PC by typing in **Devmgmt.msc** in the Search window.

A control command is simply an ASCII character string sent to a rig through a USB COM port. All commands, whether to change frequency, change bands, or change mode, etc. are generated by the Arduino.

Arduino Development Software Environment

The program or code is what makes Arduino do the job. The language Arduino understands is a variant of the C++ programming language. The source code file **FT-891-control_box200.ino** can be downloaded from the ARRL *QEX* web page **(www.arrl.org/QEXfiles)**. However, uploading the executable code to your Arduino board requires the installation of Arduino development software. (This is known as an *Integrated Development Environment* or *IDE*).

Arduino IDE Installation

Connect a USB cable between your Arduino board and PC. Windows 10 normally finds the proper driver by itself for your board. If properly installed, Windows 10 will assign a COM port number to your Arduino board. (The Arduino board is in fact the FT-891 control box. From now on both terms will refer to the same piece of hardware.)

On your PC, open the Windows *Device Manager* and open “Ports COM and LPT” to verify that a new Arduino COM port has been created. Take note of the COM port number that was assigned to the Arduino board. You may also get to that display by typing “Devmgmt.msc” in the Search bar (**Figure 2**) .

Go to the official Arduino site at **arduino.cc/software** and select the proper IDE version for your specific version of the Arduino. Download the IDE and run the installer.

Once in the IDE environment, from the main menu select:

- Tools > Board > select Arduino UNO
- Tools > Port > select the COM port matching your Arduino board, it should be displayed.
- Manage Libraries > press Enter and wait until the libraries are updated.
- In the Library Search window, type in “keypad” and press Enter. Wait for the search to display the keypad library and select Install.

At this point, Windows 10 has installed the Arduino driver and assigned a free COM port to it. The IDE libraries on your PC were updated and one extra library was added.

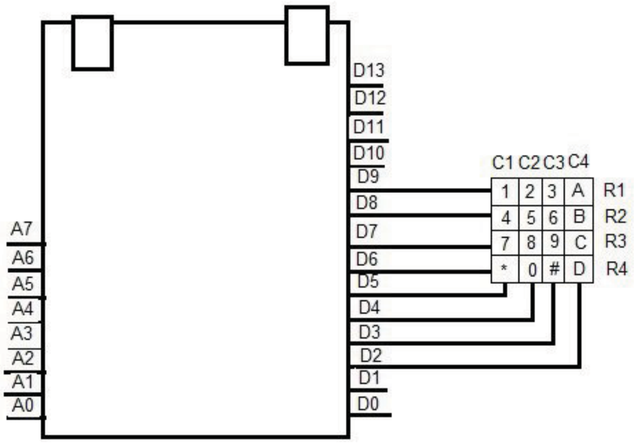


Figure 3 — Here are the column and row connections to the Arduino UNO board.

Keypad Connections and Testing

Despite what you might have read online, wiring the keypad to the Arduino is not that simple. You’ll need to know which pins correspond to the rows and columns. In my experience pinouts vary from one manufacturer and model to another. The keypad I use may be very different from the one you have. That’s why I have not included my keypad’s specific pinout in **Figure 3**.

If you happen to have the keypad datasheet, great. If not here is what I suggest: You can use a multimeter to determine which are the rows and columns of your keypad. Or you can try the following method. The Arduino IDE has a built-in serial monitor that can verify what’s coming out of the Arduino serial port. Here are the steps to upload the source code for **keypad_test.ino** and find the pinout of your keypad. (**keypad_test.ino** can be downloaded from **www.arrl.org/QEXfiles**.)

- Download the file **keypad_test.ino** into a folder on your PC.
- Run Arduino IDE.
- Go to the main menu File > Open > select the control box source file **keypad_test**.
- Go to Sketch > Verify/Compile
- If no errors were found, go to Sketch > Upload.
- Go to menu Tools > Serial Monitor to reset the Arduino and start the test program.
- Set the baud rate of the serial monitor to 9600 baud.

Figure 4 — Here is what you should expect on your screen when entering a 5-digit frequency like **14074** and pressing the ‘**#**’ Enter key.

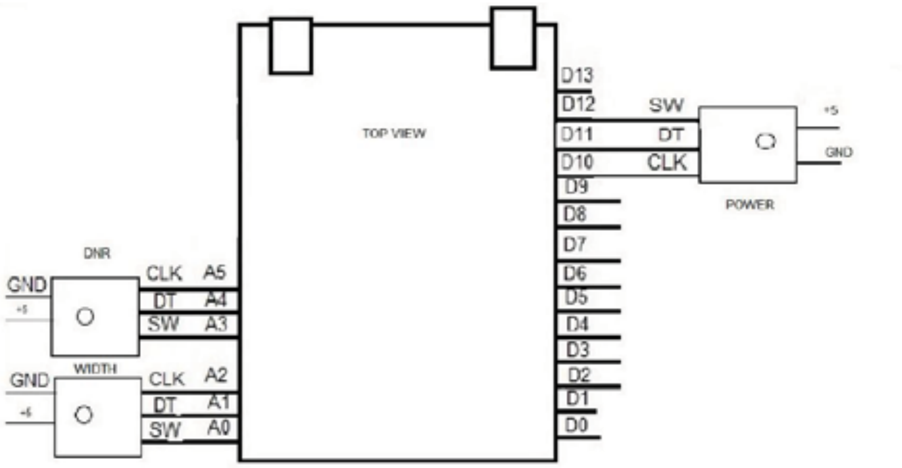


Figure 5 — Here are the Rotary Switch Connections to the Arduino Uno board.

Temporarily connect the rows and columns of your keypad to the Arduino data pins as follows:

Pin 9 Row R1	Pin 5 Column C1
Pin 8 Row R2	Pin 4 Column C2
Pin 7 Row R3	Pin 3 Column C3
Pin 6 Row R4	Pin 2 Column C4

With **keypad_test** running on your Arduino, if you short data pins D5 and D9 for instance, you'll get a '1' on your screen and so on. This is the first step to finding all the other connections. Finding all the connections is essential for this project to function properly. By experimenting with your keypad, you can identify each of the four rows and columns, then make permanent connections to the Arduino board.

Programming the Arduino Uno with the Control Box Software

The next step is to load the **FT-891_control_box200.ino** program file, compile it, and upload the executable code to your board.

- Download the control box source file in a folder on your PC.
- Run Arduino IDE.
- Go to main menu File > Open > select the control box source file (**FT-891_control_box200**).
- Go to Sketch > Verify/Compile.
- If no errors were found, go to Sketch > Upload.
- Go to menu > Tools > Serial Monitor to reset the Arduino and start the test program.
- Set the baud rate of this serial monitor to 9600 baud.

Here is what you should expect the serial monitor to display on your screen when entering a 4-digit frequency like 3750 and pressing the '#' key to enter the numeric value: "FA003750000;" When entering a 5-digit frequency like 14074 and pressing the '#' key you should see: "FA014074000;" as in **Figure 4**.

At this point, the keypad is properly connected and *FT-891_control_ box200* is running on your Arduino. You have entered a 4-digit and a 5-digit frequency and obtained the expected results on your screen.

Rotary Encoder Connections and Testing

The connections of the three rotary encoders (**Figure 5**) are as follows:

WIDTH	DNR	POWER
Rotary Encoder 3	Rotary Encoder 2	Rotary Encoder 1
CLK A2-D16	CLK A5-D19	CLK D10
DT A1-D15	DT A4-D18	DT D11
SW A0-D14	SW A3-D17	SW D12
+5	+5	+5
GND	GND	GND

For sake of clarity the +5 and GND pins of the rotary switches are not shown on the wiring diagram. All +5 rotary encoder pins shall be tied to the (only) +5 pin on the Arduino board. All GND rotary encoder pins may be tied to any GND pin on the Arduino board. See **Figure 6** for a photo of rotary encoder 2 connected to the Arduino board.

Now that the rotary encoders are connected to your Arduino board, test all the connections with the *Rotary_encoder_and_switch_test* utility program which can be downloaded from the **www.arrl.org/QEXfiles**.

Programming the Arduino Uno with the Rotary Encoder Test Program

Start the Arduino IDE and load **Rotary_encoder_and_switch_test.ino**. Verify/Compile *Rotary_encoder_and_switch_test* and upload it to your Arduino board. Go to menu Tools > Serial Monitor to reset Arduino and start the test program. Set the baud rate of this serial monitor to 9600 baud.

For this test you'll need to rotate rotary encoder 1 clockwise (CW) and counterclockwise (CCW).

When rotating the knob CW you should get this message on your screen "Encoder 1 CW".

When rotating the knob CCW you should get this message on your screen "Encoder 1 CCW".

When pressing the rotary encoder 1 switch you should get this message on your screen "Encoder 1 pressed".

Run the same tests for rotary encoders 2 and 3. If you have obtained the expected results, reload the control program

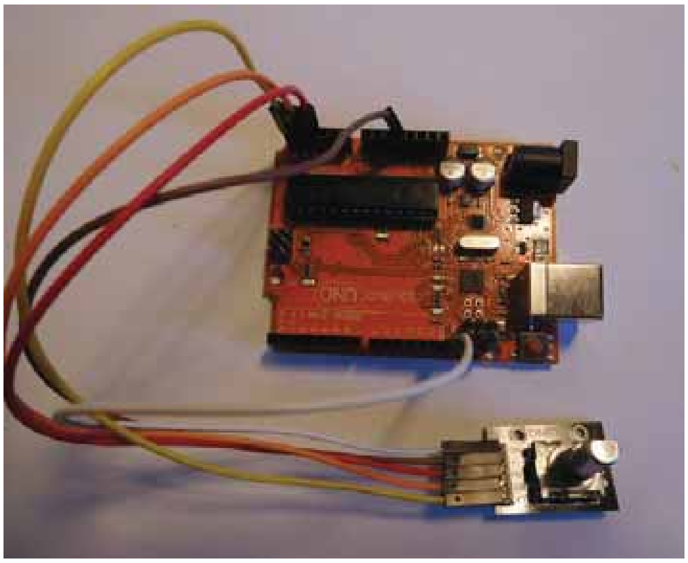


Figure 6 — Here is the Arduino connected to Rotary Encoder 2 using male to female breadboard jumpers

FT-891_control_box200. At this point, the keypad and the encoders are correctly wired to the Arduino.

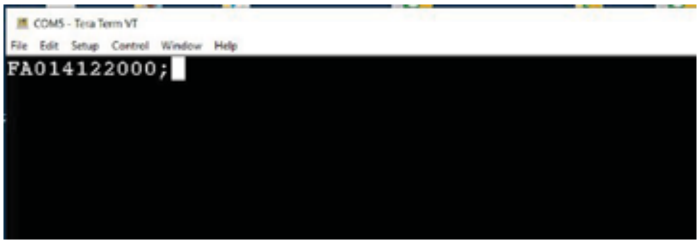
Installing the FT-891 USB Driver

Caution: If you already have installed a USB driver for the FT-891, skip this section.

Prior to running the driver installation software, unplug any USB cable connected to the FT-891. Don't connect your FT-891 to your PC before installing the FT-891 USB driver. If you do so, WIndows 10 could install a different driver than the one required.

Install the USB driver for the Yaesu FT-891 with the following steps:

- Search for "yaesu ft-891" and navigate to the Yaesu web page specifically for FT-891. Select File and at the bottom of the screen there will be a file link titled "FT-891/ SCU-17 USB Driver (Virtual COM Port Driver)". Download this file and unzip it to a suitable directory on your PC.
- Double-click on the appropriate filename for your specific operating system. e.g. Either "CP210xVCPInstaller_x64" or "P210xVCPInstaller_x32" and proceed with the driver installation.
- Turn on the FT-891 and connect the USB cable to your PC. If the driver installation was successful, your PC will recognize the new hardware and automatically finalize the driver installation.
- Go to the Windows 10 Device Manager under "Ports COM and LPT" and verify that two new COM ports have been created with the following names:
  Silicon Labs Dual CP210x USB to UART
  Bridge:Enhanced COM Port (COM 5)*
  Silicon Labs Dual CP210x USB to UART Bridge:
  Standard COM Port (COM 6)*
  *The exact COM port numbers are determined during driver installation and are specific to your PC. Your COM port numbers will probably differ from COM 5 and COM 6. The COM port that is labeled "(Enhanced Port)" will be the one you use to send control commands to the FT-891 connected to your PC. Take note of this COM port number for later.



```
FA014122000;
```

Figure 7 — In *TeraTerm*, here is what you should expect on your screen when typing in "FA014122000;" and then pressing the Enter key

- On the FT-891 press the "Function F" button for at least one second to activate the menu mode. Once in the menu mode select:
  05-06 and set it to: 9600bps
  05-07 and set it to: 3000ms
  05-08 and set it to: disable

At this point, the FT-891 is ready to receive ASCII control commands either from the Arduino or any terminal emulation software through its (Enhanced) USB port which is shown as COM 5 here.

Installing the Terminal Emulation Software

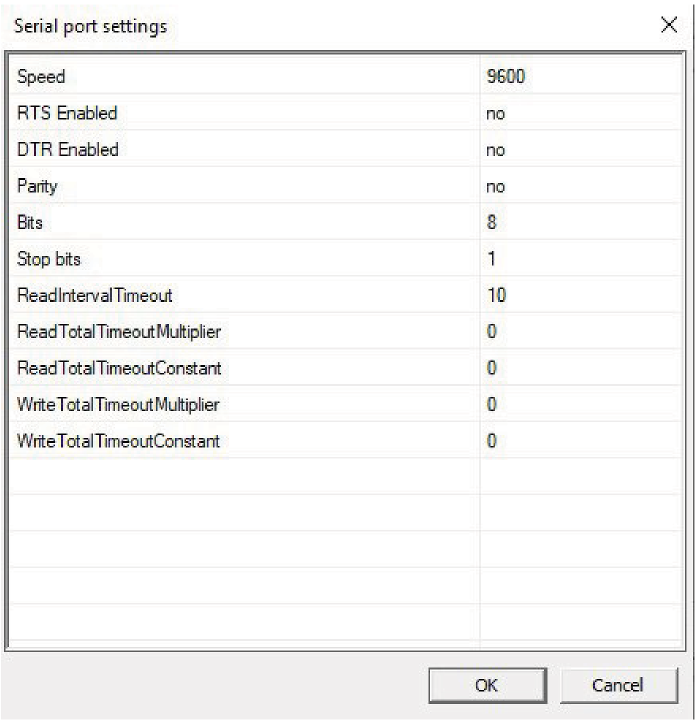
To test the CAT system functionality of the FT-891 you'll need terminal emulation software that sends control commands to the FT-891 through a PC. Many free and open source software packages are available online. I personally use a free version of *TeraTerm* (**teratermproject.github.io/index-en.html** or **sourceforge.net/projects/tera-term**) but other software can perform this task. Note: Although the following steps apply to *TeraTerm*, you'll need to make equivalent choices as well on the terminal emulation software of your choosing.

Download *TerraTerm* emulation software, unzip it, and install it. Go to:

- Setup, Terminal under Terminal ID choose " VT-100" press "OK".
- Setup, Terminal, Local Echo – check to enable.
- Setup, Port choose " Com 5 " (use the Enhanced port number that you wrote down previously).
- Setup, Data choose "8 bit".
- Setup, Parity choose "none".
- Setup, Stop Bits, choose "1".
- Setup, press "New Setting".
- Setup, Save Setup, choose TERATERM.INI and press Enter.

Testing Communication Between the FT-891 and Terminal Emulation Software

- Connect a USB cable from FT-891 to your PC.
- Turn on the FT-981.
- Run TerraTerm using the above settings.



Speed	9600
RTS Enabled	no
DTR Enabled	no
Parity	no
Bits	8
Stop bits	1
ReadIntervalTimeout	10
ReadTotalTimeoutMultiplier	0
ReadTotalTimeoutConstant	0
WriteTotalTimeoutMultiplier	0
WriteTotalTimeoutConstant	0

Figure 8 — Here is an example of VSPE serial COM port settings. These settings shall be performed for both ports. (Rig enhanced port and Arduino/Control Box)

The communication between the FT-891 and the terminal emulation software will only be successful if they both share the same serial communication protocol. e.g. port number and baud rate.

A control command is composed of an alphabetical command, optional parameters, and a semicolon as the terminator. Note: Type in only what is between the two double quotation marks! Use capital letters. Don’t take more than 3 seconds to enter any command.

In *TerraTerm*, type in “FA;” and then the Enter key. The radio should respond by sending back the actual dial frequency.

In *TerraTerm*, type in “BS03;” and then the Enter key. The radio shall respond by going to the 40-meter band.

In *TerraTerm*, type in “BS04;” and then the Enter key. The radio shall respond by going to the 10-meter band.

In *TerraTerm*, type in “FA014122000;” and then the Enter key. That command will set VFO A to 14122000 Hz as shown in **Figure 7**.

If these commands were not successful, review the FT-891 driver installation procedure as well as the terminal emulation software installation.

At this point, the control box (Arduino UNO) and the FT-891 are ready to be linked with one another.

Linking the FT-891 to the Control Box

The control box and the FT-891 are connected to a PC through their respective serial ports. Linking the FT-891 and the control box is done by means of software that is installed on the PC. I have experimented with two different software packages to link the FT-891 to the control box: *VSPE* (Virtual Serial Port Emulator by Eterlogic, **eterlogic.com/products.vspe.html**) and *hub4com* (**sourceforge.net/projects/com0com/files/hub4com/2.1.0.0**). I personally use *VSPE*. The 64-bit version is affordable and easy to use. Note that on an old PC, *VSPE* may be much slower to react than *hub4com*, especially with the width rotary encoder. (A description of the process to use *hub4com* is available **www.arrl.org/QEXfiles** file **Using hub4com for FT-891 Control**.)

- Connect the FT-891 to the PC (turn on the FT-891).
- Connect the Control Box to the PC.
- Make sure the **FT-891_control_box200.ino** program was uploaded to the Arduino.

Download the *VSPE* software and unzip it. Here’s an example of the required VSPE settings to link a Control Box connected on COM 19 to an FT-891 connected on COM 5:

- Run **SetupVSPE.msi** from the downloaded VSPE software package.
- From the main menu select Device > Create > Device Type > choose Serial Redirector, then press Next.
- Set serial port 1 to COM 5 with settings of Speed 9600, RTS enabled-NO, DTR enabled-NO, and select OK.
- Set serial port 2 to COM 19 with settings of Sp eed 9600, RTS enabled-NO, DTR enabled-NO, and select OK.

Press Finish. If successful, a message in green should appear at the bottom of the screen. All these settings can be saved in a configuration file and recalled automatically when *VSPE* starts. (**Figure 8**)

- From the main menu, select File > Open > select the control box source file (**FT-891_control_box200**)
- Go to Sketch > Verify/Compile
- If no errors were found, go to Sketch > Upload

At this point, the Control Box is connected to the FT-891 and the Arduino was uploaded with the FT-891 Control Box program

Testing the Control Box

Enter a 4-digit frequency and then press the ‘#’ key, the FT-891 VFO A will be set to that frequency. Enter a 5-digit frequency and then press the ‘#’ key, FT-891 VFO A will be set to that frequency.

Press the ‘A’ key on the keypad to change band (up) . The rig should respond by going through all possible bands in a round-robin fashion as the ‘A’ key is pressed (160m, 80m, 40m, 30m, 20m, 17m, 15m,12m, 10m, 6m).

Press the ‘B’ key to change band (down). The rig should respond by going through all possible bands in a round-robin fashion as the ‘B’ key is pressed. (6m, 10m, 12m, 15m,17m, 20m, 30m,40m, 80m,160m).

Press the ‘C’ key to change mode. The rig should respond by going through all possible modes in a round-robin fashion as the ‘C’ key is pressed. (LSB, USB, CWU, CWL, AM, FM, R-Ll, R-U,D-U, D-L).

Press the ‘D’ key to change mode. The rig should respond by going through all possible modes in a round-robin fashion as the ‘D’ key is pressed. (USB, LSB, D-L, D-U, R-U, R-L, FM, AM, CWL, CWU).

Press the ‘*’ key to toggle through the QMB memories in a round-robin fashion.

Press the rotary encoder 1 (Power) switch to save the current VFO-A frequency to one of the 5 QMB memories.

Rotate the rotary encoder 1 (Power) to increase or decrease power output.

Rotate rotary encoder 2 (DNR) to select one of the 15 Digital Noise Reduction (DNR) algorithms. If the DNR is OFF, turning the control in either direction will turn it on.

Press the rotary encoder 2 (DNR) switch to turn DNR ON or OFF.

Rotate rotary encoder 3 (WIDTH) to vary the IF WIDTH. Turning the control in either direction will turn on the WIDTH operation.

Press the rotary encoder 3 (WIDTH) switch to turn IF WIDTH operation OFF.

Control Box Program Limitations

The program only accepts 4- or 5-digit frequencies. Therefore, the frequency entered will always be in kilohertz (kHz). The allowed frequency range is from 1800 kHz to 54000 kHz. If you enter anything outside this range, the program will not respond.

Since there is no Clear Entry (CE) key, if you enter a wrong value the only option is pressing the "#" key and re-entering the value.

This control box software will not work with *Omnirig*.

The Arduino Uno generates harmonics from 14074-14077 kHz, which falls exactly on the FT8 band. You may need to shield the Arduino and add ferrite EMI suppression cores (Type 31) to all control and power lines.

Conclusion

I would like this article to pique the curiosity of FT-891 owners and encourage them to try this project and experiment with Arduino. I think there is enough information in this document to guide either a novice or an experienced Arduino user.

As for the Arduino code, I am not an Arduino language expert, but this version of *FT-891_control_box200.ino* is stable. When programming, I try to avoid complicated loops. In my view, it can help with debugging and readability. I am not afraid of redundancy when coding. This alone can offend C language experts. Therefore, there is no doubt in my mind that my code will be optimized.

Supplementary Information

Changing the Control Box Baud Rate

If for some reason you wish to change the baud rate of the control box, look up the following instruction in the ino file: Serial.print(9600); Change it to 4800, 19200 or 38400 bps. Then run the Arduino IDE.

- From the main menu File > Open > select the control box source file (**FT-891_control_box200.ino**)
- Go to Sketch > Verify/Compile
- If no errors were found go to Sketch > Upload

Source Code Files

All required source files can be downloaded from the ARRL *QEX* files web page (**www.arrl.org/qexfiles**):

keypad_test.ino
Rotary_encoder_and_switch_test.ino
FT-891_control_box200.ino

Mario Lechasseur, VE2KEC, is currently retired from the Department of National Defence of Canada (DND) where he spent 36 years. As a civilian, he held different positions ranging from electronic technician, technical writer, team leader, and finally electronic workshop manager where the main task was the maintenance and installation of communications systems, mainly in US armored vehicles and Leopard German tanks, and shielded cable manufacturing for various armored vehicles.

Mario has a technical background as an Electronic Technician, being a Telecommunications graduate (1978). During his career at DND, he was trained and worked on various HF, VHF and UHF radios and subassemblies.

He first got acquainted with 8-bit microprocessors in 1984 with Motorola 6809, Intel 8080 and 6502. In 1993 he completed a university certificate in microprocessors and had the opportunity to program in assembly language in different processor architectures like Motorola 68008, DEC PDP-11 and VAX. During the same time, he did a lot of assembly language programming on IBM PC clones.

Although he has designed microcontrollers projects before, when he was introduced to Arduino Uno around 2011 it definitely was a game changer. Mario never designed any electronic projects with microcontrollers after that.

In 2022, long after his retirement from public service, Mario developed an interest in FT8, and received his Canadian ham radio license in 2023. He's been operating a Yaesu FT-891 since December 2023.

Anthony Le Cren, F4GOH/KF4GOH
47 rue de l'avenir, 72220 Teloche, France | **kf4goh@sfr.fr**

Demonstrating the Idea of Gas Law Using a Balloon, SSTV and LoRa

Computer Science and Networks high-school students program an HF SSTV transmitter to send images.

Introduction

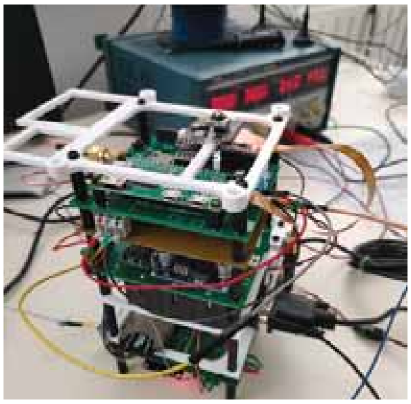


Figure 1 – The SSTV and LoRa transmitter

Each year, a new sounding balloon is launched at the Touchard High school in Le Mans (France). This is an opportunity for 3 BTS Computer Science and Networks students to improve last year's electronic boards. As part of their end-of-year project, the students had to program an HF SSTV transmitter to send images. **[Figure 1]** The board was on board the sounding balloon flight on 31 May 2024 [1]. **[Figure 2]** The transmitter's upward-facing camera regularly sent images of the progress of the balloon's envelope before it burst. The electronic board also features an Ra02 LoRa modem for ground telemetry.

The main aim of this project was to verify the law of perfect gases by estimating the volume of the balloon as a function of altitude. To do this, photographs of the balloon at different altitudes were taken. Another goal was to experiment with LoRa radio frequency transmission, both during ascent and descent. (video on YouTube **[2]**).

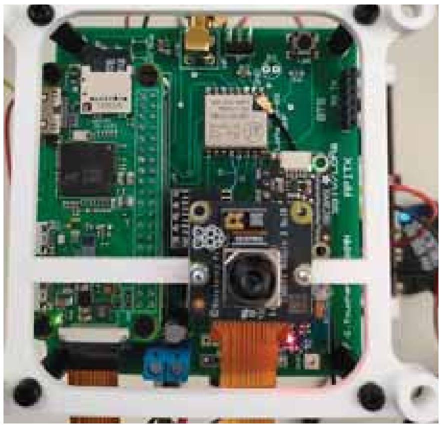


Figure 2 – The Raspberry zero board and its camera

Predicting the trajectory of the balloon probe

The **predict.sondehub.org** website is the online tool used for trajectory prediction. It predicts the trajectory of the stratospheric sounding balloon based on weather conditions and launch data. This includes the direction and distance the balloon will travel before reaching its maximum height and descending. Predicting where the balloon will land helps us organize the recovery of the balloon and its payload. This is crucial for our project, as the data recorded by the on-board instruments needs to be recovered.

Weighing Climbing Speed

The balloon rises into the atmosphere thanks to Archimedes' thrust. The gondola's mass is 1610 g, i.e. a weight of 15.8 N, thus respecting the 1800 g limit required by the CNES specifications. The balloon is filled with helium with an initial volume of V ≈ 5.0 m³, i.e. two 15 L bottles of helium at a pressure of 180 bar. On the ground:

- Archimedean thrust = $1.225\ \text{kg/m}^3 \times 5\ \text{m}^3 \times 9.81\ \text{m/s}^2 \approx 60.0$ N

At 20 km altitude (air density approx. 0.0889 kg/m³ balloon volume 65 m^3):

- Archimedean thrust = $0.0889\ \text{kg/m}^3 \times 65\ \text{m}^3 \times 9.81\ \text{m/s}^2 \approx 56.7$ N
  - Balloon (envelope + Helium): 2.2 kg 21.0 N
  - Parachute: 0.350 kg 3.4 N
  - Radar reflector: 0.250 g 2.4 N

Motion is rectilinear and uniform; the principle of inertia applies.
$F_r = P_a - P = 60 - (15.8 + 21 + 3.4 + 2.4) = 17.4$ N
A resisting force in the air Fr = 17.4 N allows an ascent speed of 5 m/s.
Lift-off speed = 4.26 m/s Lift-off speed at 4000 m = 4.64 m/s.
Rate of climb at 20,000 m = 5.73 m/s Rate of climb at 30,000 m = 6.77 m/s.

Tracking the Balloon Probe

Trajectory

The update rate for tracking is once a minute on the 70cm band (LoRa). **[Figure 3]**

Flight time:

- Departure 13h 13’ 50”
- Arrival 15h 58’ 15”
- Duration 2h 44’ 25”

Figure 3

Balloon Burst

As the balloon rises, the pressure of the surrounding air decreases. As a result, the helium occupies an ever-increasing volume, causing the balloon to expand until it bursts (14:57:50 CET – 33050 m).

As the center of gravity is located in the upper part of the gondola (opposite side to the camera), when the balloon bursts, the gondola tilts to the side, allowing the GoPro camera to film the remains of the balloon envelope. **[Figure 4]** The balloon bursts at an altitude of 33,050 m.

The center of gravity is located in the upper part of the gondola (opposite the camera), so when the balloon bursts, the gondola tilts to one side, allowing the gopro camera to film the remains of the balloon envelope. The recorded burst altitude is 33,050 m.

Acceleration measured on the Z axis of the sensor. **[Figure 5]**

At this point, the flight chain descends back to earth. The parachute, initially inserted as a “fir tree” between the balloon envelope and the radar reflector, now acts as a brake.

We observe an almost constant ascent speed, oscillating between 5 m/s and 6 m/s, as well as a descent speed under the parachute which corresponds to that of the air’s density. **[Figure 6]**

First phase. This phase corresponds to the ascent of the stratospheric balloon at constant speed.

Acceleration is vertical, and corresponds to the acceleration due to gravity 1g.

14:57:50 balloon bursts, gondola falls, sensor is weightless, 0 g.

Figure 4

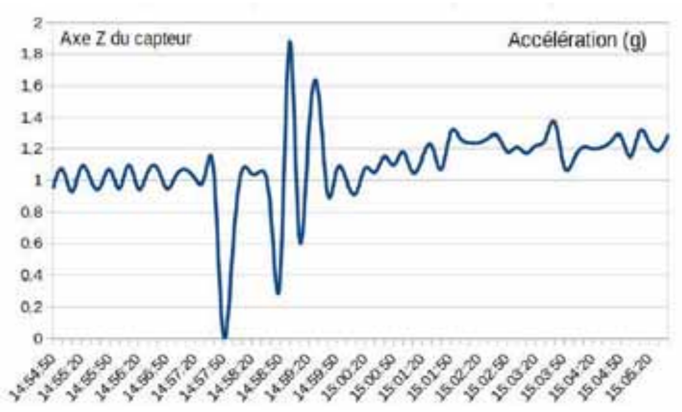


Figure 5

Second phase. For two minutes, the gondola performs aerial pirouettes before stabilizing, tossed between inverted positions, before finally righting itself under the action of the parachute. This is why the acceleration oscillates around 1g.

Conclusion. It should be possible to detect the bursting of the balloon by detecting a decrease in acceleration along the z axis of the sensor.

Sensor calibration. It will be necessary to calibrate the sensor to minimize offsets and measurement errors. Calibration involves collecting data when the sensor is stationary and calculating average offsets for each axis.

The SSTV Transmission and Reception System

The upward-facing camera regularly sends the progression of the balloon envelope’s magnification before bursting, then that of the parachute during descent.

SSTV (Slow Scan Television) is a method of transmitting still images via radio waves, used mainly by amateur radio operators. Unlike traditional television that transmits images at a rate of several dozen per second, SSTV transmits a single image. The images are converted into audio signals, sent in SSB modulation via the radio frequency 29,000 MHz, with a transmission power of 0.7 W. They are then converted back into images on ground reception. SSTV requires a reduced bandwidth of around 3 kHz.

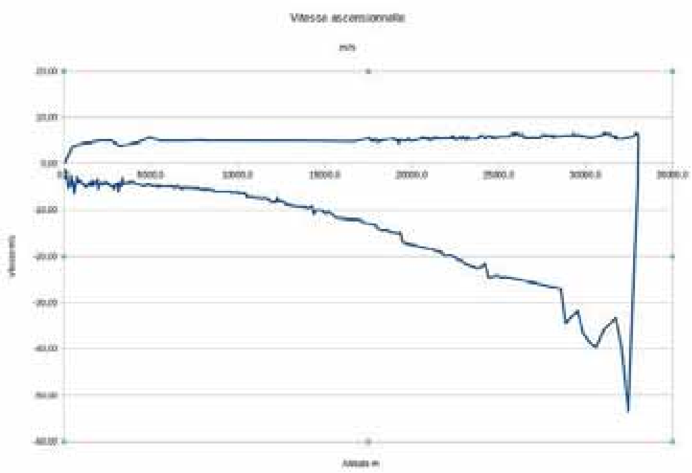


Figure 6

There are several SSTV transmission modes, each with its own characteristics (image definition and transmission speed). The most commonly used modes are Martin, Scottie and Robot. The Martin1 SSTV mode is used because it provides a good balance between transmission speed and image resolution. It operates on an SSB frequency of 29 MHz.

The SSTV image is received by a Yaesu FT897-D. Once the image has been reconstructed by the *YONIQ* software (MMSSTV was used previously), it is saved on our web server so the whole school can see the balloon photos in real time.

Here's a summary of the key points of the Martin1 format:

- Resolution: 320x240 pixels
- Lines: 240 + 16 gray lines
- Time per line for one component: 146.432 ms
- Line synchronization 5 ms
- Frame synchronization 30 ms
- Color sequence: RGB
- Synchronization frequency: 1200 Hz (line and frame)
- Black/white frequency: 1500 Hz for black, 2300 Hz for white
- Total image time:

$$T=((240+16)\times 146{,}432\times 3)+256*5+30=113769{,}776\,ms$$

- 114 seconds (1 minute and 54 seconds)

These characteristics enable the transmission of images of reasonable quality on bandwidth-limited radio channels.

Image information is transmitted line by line. Color is transmitted in three successive scans, the first for red, the second for green and the last for blue, according to the RGB (Red Green Blue) composition principle.

Line synchronization signal (1200 Hz):

- Duration: 4.862 ms
- Used at the start of each line to indicate synchronization.
- Color tones for pixels:
  Red, Green, Blue (RGB): Each line is scanned for each color component.
- Each color level is coded by a frequency that varies between 1500 Hz (level 0) and 2300 Hz (level 255).
- Frame synchronization signal (1200 Hz):
  Used to indicate synchronization of the entire image, sent after each complete frame. Photos are sent at a frequency of one every five minutes.

This very simple structural scheme is based around a Raspberry Pi Zero. The SSB SSTV signal is generated directly by the Raspberry Pi on GPIO4.

Thirty-three images were sent during the flight; and some SSTV images were received. **[Figures 7, 8, and 9]**

Figure 7

Figure 8

The photos are of satisfactory quality, with the call sign of the school's radio club and the time of shooting superimposed. This enables radio amateurs who receive them to send them back to us by e-mail. (The radio club's e-mail address is available on **QRZ.com**.) (*We thank Sébastien Barbier, F4IRT, for his report.*)

Please note that when recording a JPG image on the Raspberry Pi's mSD card during SSTV transmission, a very brief interruption occurs, causing one or more unwanted lines on the received image.

In addition to the low-resolution photos sent, high-resolution photos are captured every minute and stored on the Raspberry Pi's SD card.

- 15 minutes after take-off apparent balloon diameter = (50 / 70) * 1.08 = 0.80 m
- Photo taken just before bursting. apparent balloon diameter = (146 / 70) * 1.08 = 2.24 m

The pressure value superimposed on the display is incorrect. The problem is software-related, and it is corrected now in the last version.

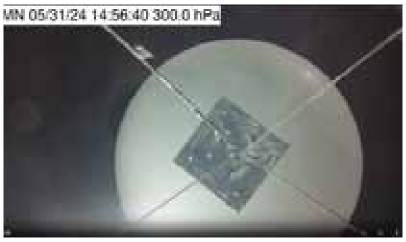


Figure 9

The Hardware

The image captured by the Raspberry Pi CAM v3 camera module will be recorded every minute on the SD card and transmitted in HF on the 10-m band, every 8 minutes in SSTV Martin1.

The SSTV image is received by a Yaesu FT897-D or a simple RTL-SDR receiver. Once the image has been reconstructed by the *YONIQ* software, it is saved on our web server so the whole school can see the images in real time. **[Figure 10]**

The camera is aimed at the sky, the objective being to measure changes in the diameter of the envelope of the balloon containing

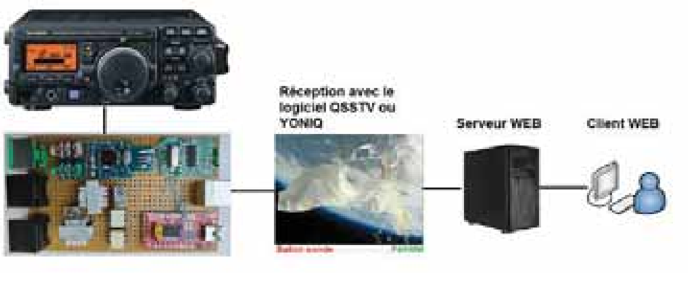


Figure 10 – The ground reception system.

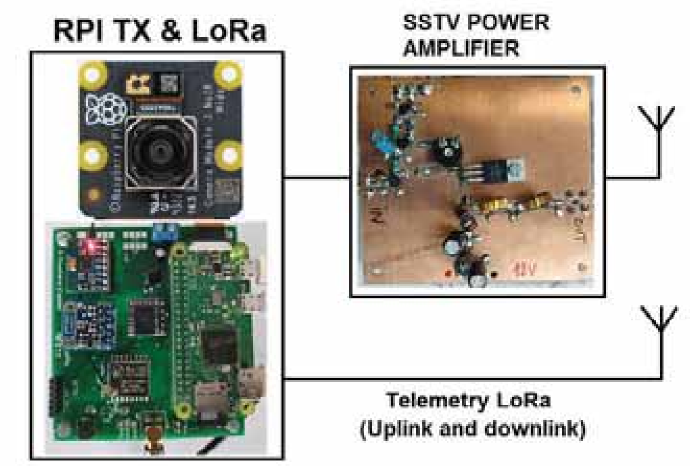


Figure 11 – SSTV and LoRa transmitter diagram

the helium. The diameter increases with altitude as atmospheric pressure decreases and the gas contained in the balloon expands.

This experiment will allow us to demonstrate the accuracy of the law of perfect gases: PV = nRT.**[3]**

LoRa Telemetry

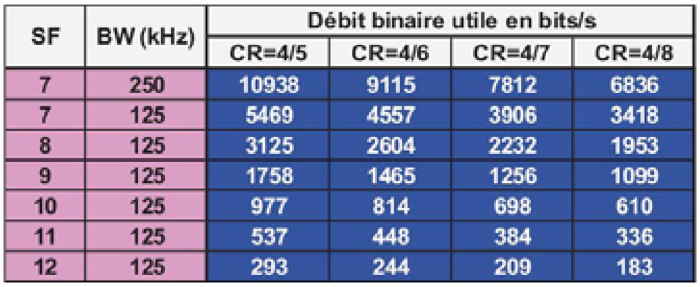
LoRa telemetry (Long Range), enables the transmission of telemetry data such as temperature, pressure, humidity and vertical acceleration, via radio waves in the 70 cm band. **[Figure 11]**

This technology is essential for transmitting data from various sensors on board a balloon or other equipment. Its main advantage lies in its ability to provide reliable and efficient connectivity with the ground station, making it easier to track and monitor the balloon in real time during its mission.

Radio parameters used:

- Center frequency: 433.775 MHz (in the 70 cm band from 430 to 440MHz)
- SpreadingFactor SF: 12
- CodingRate CR: 4/8

The useful bit rate is therefore 183 bits/s, or 22.9 characters per second. **[Figure 12]**



SF	BW (kHz)	Débit binaire utile en bits/s			
		CR=4/5	CR=4/6	CR=4/7	CR=4/8
7	250	10938	9115	7812	6836
7	125	5469	4557	3906	3418
8	125	3125	2604	2232	1953
9	125	1758	1465	1256	1099
10	125	977	814	698	610
11	125	537	448	384	336
12	125	293	244	209	183

Figure 12

LoRa modulation as seen in the GQRX waterfall. **[Figure 13]**

During the flight, one frame was transmitted every minute. The frames were picked up and relayed by the following relays.

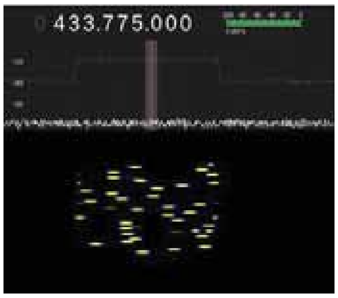


Figure 13

Figure 14

As can be seen from the map, the frames transmitted in LoRa modulation were received within a radius of 700 km.

LoRa telemetry enables the transmission of telemetry data such as temperature, pressure, humidity and vertical acceleration. The sensors used in this context are the BME280, LM75 and MPU6050, which are useful for collecting environmental and movement data.

For telemetry, the balloon is identified on the ground as a simple weather station (WX), which means it can be easily integrated into the APRS network. **[Figure 14]**

The sensors used in this context, such as the BME280, LM75 and MPU6050, will be useful for collecting environmental and motion data.

For ground data reception, we chose to use a LoRa TTGO equipped with an ESP32, combined with the APRS (Automatic Packet Reporting System) protocol. This solution offers great flexibility and compatibility with existing networks. The balloon is identified on the ground as a simple weather station (WX), making it easy to integrate into the APRS network. What's more, thanks to this configuration, the telemetry data can be viewed in real time on platforms such as aprs.fi, giving us continuous, accurate monitoring of the balloon's flight. On the APRS.fi map, there will be two call signs: **[Figures 15 and 16]**

F4KMN-13 · centrer · zoomer · info

2024-04-18 17:03:59 - 2024-04-18 17:06:00

alt 79 m

27 26 0 12.5 97 +0.1

[APLT00 via WIDE1-1,WIDE2-2,qAO,F4KMN]

démarrer poursuite

Figure 15 – The tracking trace.

f4kmn-9 · centrer · zoomer · info

2024-04-10 14:06:27 - 2024-04-10 14:30:59

Météo APRS/CWOP 2024-04-18 17:47:34: Graphiques météo

Température **25.0°C** Humidité **26%** Pression **1018.2 mbar**

[APLT00 via WIDE1-1,qAS,f4kmn-6]

en poursuite · arrêter poursuite

Figure 16 – The “flying” weather station.

- Telemetry seen as a classic weather station
- The symbol of the balloon providing tracking during the flight

Figure 17 is an example of an APRS frame received on **aprs.fi** and sent by the balloon.

There are two separating characters (on a green background):

- the ‘:’ separates the packet header from the actual data transmitted (payload)
- the ‘>’ separates the source call sign f4kmn-9 and the destination call sign APLT00 in the header

APLT00 is not a specific destination code. Rather, it is a convention specific to the LoRa-APRS protocol, which stands for Automatic Packet LoRa Transmission in the standard APRS system.

The LoRa-APRS protocol uses a different frame structure which does not support destination codes in the same way.

The value APLT00 is used by LoRa-APRS devices as the default destination code when sending APRS packets. This has no particular meaning to other APRS stations and is only used to identify the packet as coming from a LoRa-APRS device. It can be interpreted as a broadcast address.

The WIDE1-1 Path

A WIDEn-N path has two integers, n and N. WIDE1-1 would have n over 1 and N over 1. The first integer n means “I would like this packet to be repeated by as many digipeater hops” (once in the case of our WIDE1-1 example).

The second integer means “there are so many jumps left before the repetition stops.” When it becomes 0, it will no longer be repeated and the “has been repeated” bit will be set (an “*” will appear).

- Digipeaters that have heard the frame add their call signs to the frame header.
- F4kmn-6 is the call sign that heard the frame and retransmitted it.

Figure 18 – Plotting the pressure curve over time.

“qAS” indicates that the frame was injected into the APRS network by an IGate (Internet Gateway), which is a connection point between the radio networks and the Internet.

The Weather Frame

`_05311416c...s...g...t062h50b10234 1,0045`

The first character indicates the graphic symbol on the map:

- _ Represents a weather station.
- MDHM: month day hour minutes
- 05 May
- 31 day in the month
- 14 hours
- 16 minutes
- c...s...g... no information for the wind (direction, speed, gust) the values are replaced by dots.
- t062 temperature is in ° Fahrenheit 062 = 16.7 °C
- h50 - : Relative humidity (50%).
- b10234 Barometric pressure in tenths hPa (1023.4 hPa). **[Figure 18]**

Precipitation over 24 hours and since midnight are optional (letters p and P) and are not shown here.

As it is possible to add a comment in an APRS weather frame, this is used to transmit the acceleration measured on the gondola.

Uplink

One of the objectives is to change the time intervals between two SSTV transmissions. To achieve this, a LoRa uplink has been programmed, with the default time interval between two transmissions being 8 minutes. If the battery runs down quickly, the ground station will decide to change the transmission interval every 16 minutes.

The TTGO module, incorporating an ESP32 and a Ra-02 LoRa modem, will establish uplink communication with the balloon.

- Context of LoRa reception
- Dipole receiving antenna tuned to 433 MHz

RSSI, the received signal strength indicator, shows the power of a radio signal received by the antenna. It is a value relative to the mW, expressed in decibels.

```
f4kmn-9>APLT00,WIDE1-1,qAS,f4kmn-6:_06081832c...s...g...t073h45b10054 1.158936
```

Figure 17 – An APRS frame sent by the balloon.

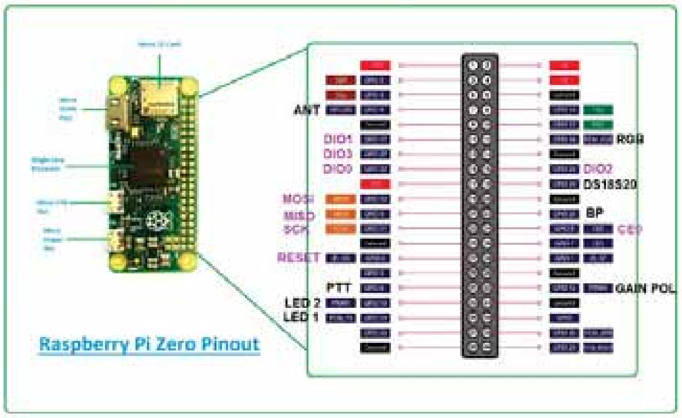


Figure 19 – GPIOs used on RPI Zero.

At 2.19 PM, the balloon was at an altitude of 19,000 m and 50 km from the school.

Receiver sensitivity: The Ra-02 LoRa receiver used is designed to be very sensitive, with typical sensitivity thresholds around –130 dBm for the best conditions (low data rates and narrow bandwidths). Reception at –108 dBm is well above this threshold, indicating a good reception margin.

- rx -------------------------------> done
- RSSI: –108 dBm
- buffer 14h19 – TEST
- Conclusion: All the messages sent have been received.

The Transmitter Schematic

The very simple structural diagram is based around a Raspberry PI zero. The SSB SSTV signal is generated directly by the Raspberry Pi on GPIO4.

- The SPI link manages the RA2 modem.
- All the BME280, LM75 and MPU6050 sensors are used via the I²C bus.
- The BP1 pushbutton triggers a clean shutdown of the Raspberry Pi before the board is powered down.
- Connector J9 is dedicated to controlling the power amplifier. Only the PTT output is used. Software gain control is not currently supported. **[Figure 19]**

The Software

A useful thing about the Raspberry Pi is its Linux operating system. In fact, three programs are run at start-up:

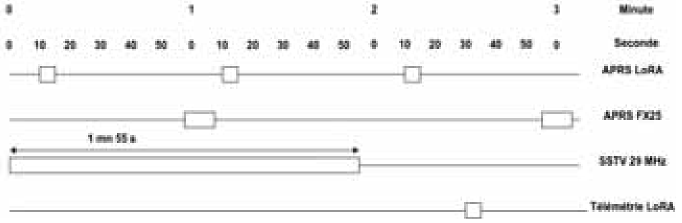


Figure 20 – Transmission intervals between different transmitters.

- SSTV management program
- LoRa transmission and reception program
- Program for reading sensors and telemetry in APRS format.

The three programs communicate with each other using local inter-process communication software techniques. This facilitates transmission intervals by minimizing overlap between transmissions.**[4] [Figure 20]**

Despite this, if a JPG image is stored in the RPI's SD card during the SSTV transmission, there is a very brief interruption, hence the unwanted line in the ground reception.

Programming was carried out using the NetBeans IDE and RPITX designed by F5OEO **[5] [6]**.

As the LoRa TTGO is in the tracking vehicle, esp32 uses the FreeRTOS library, enabling real-time multitasking.
[Figure 21]

Conclusion

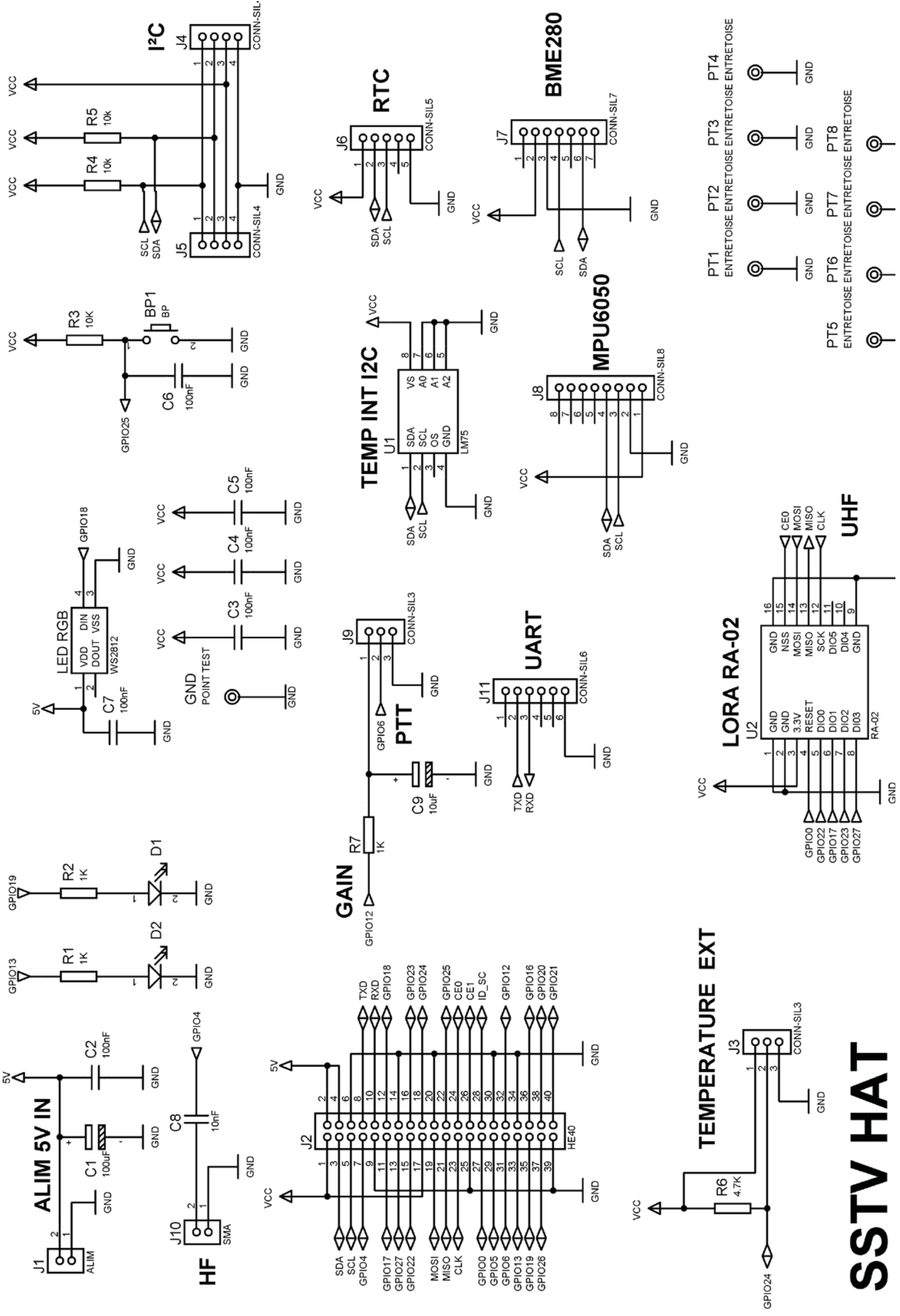
The SSTV probe balloon project, using a Raspberry Pi and the LoRa modem, was an incredibly rewarding experience for the BTS computing and networks students. The project has enabled them to put into practice the knowledge they have acquired throughout their course, as well as developing new skills in programming and electronic development.

I hope that this project will inspire other students and developers. I want to thank Mr. Simier, F4JRE; Mr. Martin; Mr. Cruchet, and Mr. Bernard, all teachers at Lycée Touchard, for their help throughout the project. We also thank Sébastien Barbier, F4IRT.

References

[1] https://youtu.be/X4jhOELSesA?si=iW_uR4ZEk_Rt_OdZ
[2] https://youtu.be/p3vWaG2_YjE?si=tti7X3qW2UJ9tjoQ
[3] https://en.wikipedia.org/wiki/Ideal_gas_law
[4] https://github.com/PhilippeSimier/Ballon_stratospherique_SSTV_2024
[5] https://github.com/F5OEO/rpitx
[6] https://publications.r-e-f.org/raspberry/documents.htm

Figure 21 – The transmitter schematic.



Appendix: Verification of the Law of Perfect Gases

See **[Figure A]**

D - Actual balloon diameter in meters

d - Apparent balloon diameter in meters

l - reflector center length / gondola top 2.7 m

L - balloon center length / gondola top 7.47 + R

Calculation of actual diameter D for apparent diameter d = 2.24 m

$\frac{D}{2,24}=\frac{7,47+R}{2,7}$ avec $D=2\times R$

$$2\times R=\frac{7,47+R}{2,7}\times 2,24$$

$$R=\frac{6,12}{1,18}=5,18m$$

Calcul du Volume

$$V=\frac{4}{3}\times pi\times R^3=584m^3$$

Calculation of theoretical pressure

For the amount of helium in the balloon, whatever the altitude, we can write:

$$\frac{P\times V}{T}=nR$$

P: pressure hPa V: volume m^3 T: temperature in Kelvin

The ideal gas law is PV = nRT. If there is a closed volume, then nR is a constant and the ideal gas law can be rewritten as PV/T = nR which is a constant for any set of measured P, V and T.

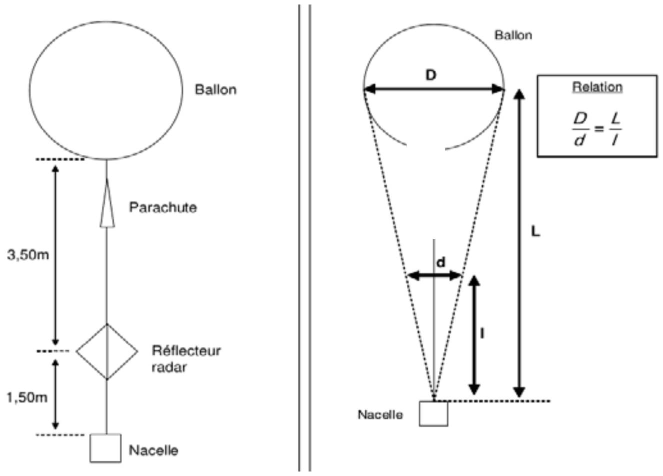


Figure A

Lycée
TOUCHARD
WASHINGTON

Students in the computer room.

At ground level we have the following values:

D_{ground} = 2.14 m

P_{ground} = 1006 hPa

V_{ground} = 5.13 m^3

T_{ground} = 273 + 21 = 294 K

At burst altitude we have the following values:

P_{theory} = to be calculated

V_{burst} = 584 m^3 as calculated above.

T_{burst} = 273 - 5 = 268 K

Since nR is a constant, we can calculate P_{burst} with the following relationship:

$P_{theory} * V_{burst} / T_{burst} = nR = P_{ground} * V_{ground} / T_{ground}$

Eliminate nR

Substituting in the values:

P_{theory} * 584 m^3 / 268 K = 1006 hPa * 5.13 m^3 / 294 K

P_{theory} * 2.179 = 17.55

P_{theory} = 8.05 hPa

$P_{measured}$ = 10.08 hPa

Conclusion

Calculation of the theoretical pressure at 33,000 m is roughly equal to the pressure measured by the on-board sensor.

To sum up, we can say that helium behaves very much like a perfect gas under the usual conditions of temperature and pressure in flight, due to its weak interatomic interactions and small atomic volume.

We can also conclude that the BME280 pressure transducer used works correctly at this low pressure, despite the lack of information in the manufacturer's documentation.

A sensor measures the temperature inside the nacelle.

Figure B – Thermal behavior of the gondola.

The aim is to maintain a positive temperature inside the gondola during all phases of flight.

To achieve this, the gondola has been entirely covered with a gold-colored blanket to conserve internal heat. The cover has two distinct metal faces: one gold, the other silver. Each face has a specific function: the silver side reflects 90% of the infrared radiation, while the gold side absorbs 50% of the incident heat. By orienting the gold side outward, the gondola is protected from the cold, retaining its internal heat. **[Figure B]**

The internal temperature dropped to –3.2°C during the descent to an altitude of 7,000 meters, once again underlining the influence of the cold drop. This icy cold underscores the extreme conditions to which the electronic boards in the gondola were exposed.

Cooling and heating of the gondola: heat transfer analysis

Analysis of the graph highlights the thermal behaviour of the gondola during flight, marked by initial cooling followed by heating.

Cooling by conduction during ascent

During ascent, the gondola undergoes continuous cooling up to an altitude of 22,000 meters. This phenomenon is explained by thermal conduction, the mechanism by which heat escapes from the gondola into the colder ambient air. The drop in temperature of the ambient air with altitude favors this heat transfer, leading to the gradual cooling of the gondola.

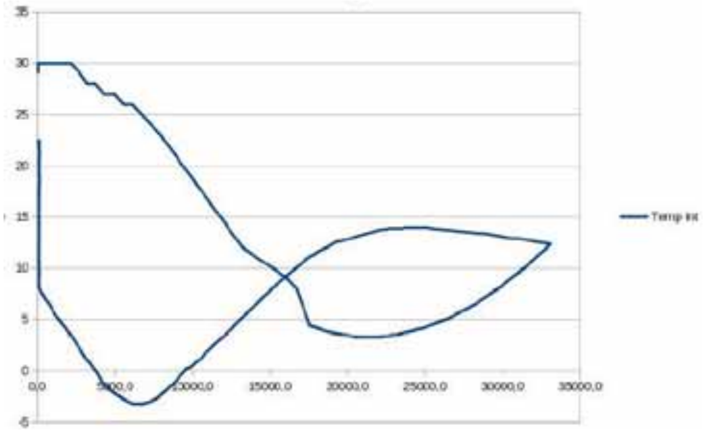


Figure C

Heating by infrared radiation

Above 22,000 meters, the gondola begins to warm up. This change in trend is attributable to infrared radiation, a mode of heat transfer that does not require direct contact between the gondola and its environment.

In the upper atmospheric layers, the concentration of gas decreases, reducing thermal conduction. On the other hand, the gondola absorbs solar radiation, contributing to its warming.

Minimum temperature on descent

The minimum temperature of the gondola is reached on descent, at around 7,000 meters altitude. This observation can be explained by the combination of the two heat transfer mechanisms.

Minimum temperature on descent

The minimum nacelle temperature is reached during descent, at around 7,000 meters altitude. This observation can be explained by the combination of the two heat transfer mechanisms.

Descent involves a reduction in altitude and therefore an increase in the ambient air temperature. Thermal conduction then promotes heating of the gondola. However, this warming is counterbalanced by the infrared radiation emitted by the nacelle, which continues to escape into space, helping to maintain a relatively low temperature. **[Figure C]**

Eric P. Nichols, KL7AJ
138 Shenandoah Dr., Fairbanks, AK 99712 | **kl7aj@arrl.net**

Self-Paced Essays — #27 Overkill

Measuring components with a VNA.

It's been a while since we last visited the amazing Vector Network Analyzer (VNA), the most "civilian" form being the very popular NanoVNA. At one time the VNA would have been considered extreme overkill for any task a typical amateur might encounter. However, VNAs have gotten so good and so inexpensive of late, there's really no excuse for any technically minded ham not to have one. The VNA really is a universal piece of test equipment that could replace a whole bench full of test gear of just a decade or two ago. This essay will measure some simple components with a VNA to show you a few things that traditional test equipment might not reveal.

Touchstone

Before we dive in, I'd like to talk a little about the Touchstone file format. Touchstone files are built from the data acquired with a VNA in a form that's easily plotted on a vector or X-Y display. The Touchstone format is highly standardized and interchangeable between many types of software.

One of the beauties of the Touchstone data is that it contains both the In-phase (I) and Quadrature (Q) data for each port, each in its own column. The data values can be either Magnitude and Phase (polar format) or Real and Imaginary (Cartesian format). Both of these are useful but can be more convenient for certain purposes. For swept-frequency measurements, there is an additional column of frequency data.

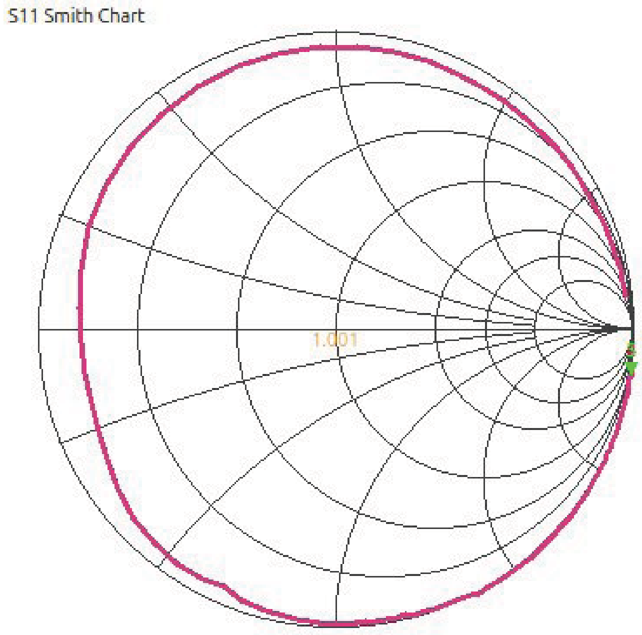


Figure 1 – A 100 pF air-variable capacitor, swept from 3 MHz to 50 MHz.

NanoVNA Saver

A good way to access the Touchstone files is through a freeware program called *NanoVNA Saver* (**nanovna.com/?page_id=90**) It is a relatively small program available for Windows and Linux. Aside from reading the Touchstone files, it saves the graphic displays of the data for export. I used that feature to generate the figures below. It also provides calibration and automation functions, which I find handier all the time.

A Capacitor That's Not

We've mentioned a few times earlier in this series that one must always study the ideal before exploring the real. A capacitor *should* be just a capacitance at any frequency, for example, but a wide-range frequency sweep will show you that there is much more than just pure capacitance in a real-world physical capacitor.

Figure 1 shows a single-port (S 1,1) sweep of a 100 pF air-variable capacitor from 3 MHz to 50 MHz, displayed on a Smith chart. The data starts at the lowest frequency which is the green dot just below the right-hand horizontal axis along the perimeter of the chart. As expected, here the impedance is an almost infinite capacitive reactance.

Moving in a clockwise direction with increasing frequency, after about 8 o'clock, the trace moves away from the perimeter, indicating the impedance has added some resistance. Continuing a little further, the trace crosses the horizontal axis: this is a series resonance where reactance is zero.

Any point above the horizontal axis corresponds to inductive reactance. In this region, the capacitor is not acting like a capacitor at all! The trace for a pure capacitance will never cross the horizontal axis to the upper half of the chart. but will simply be an arc along the lower perimeter of the chart from infinity to 0 Ω.

An Inductor That's Not

Well, let's see how an inductor fares. In this case, **Figure 2** is the Smith chart trace for a 20 µH air-core inductor with fairly wide turns spacing, swept from 300 kHz to 20 MHz. Beginning at the left-hand edge of the horizontal axis, this looks a lot more like a pure reactance; we stay cleanly above the horizontal axis, in the

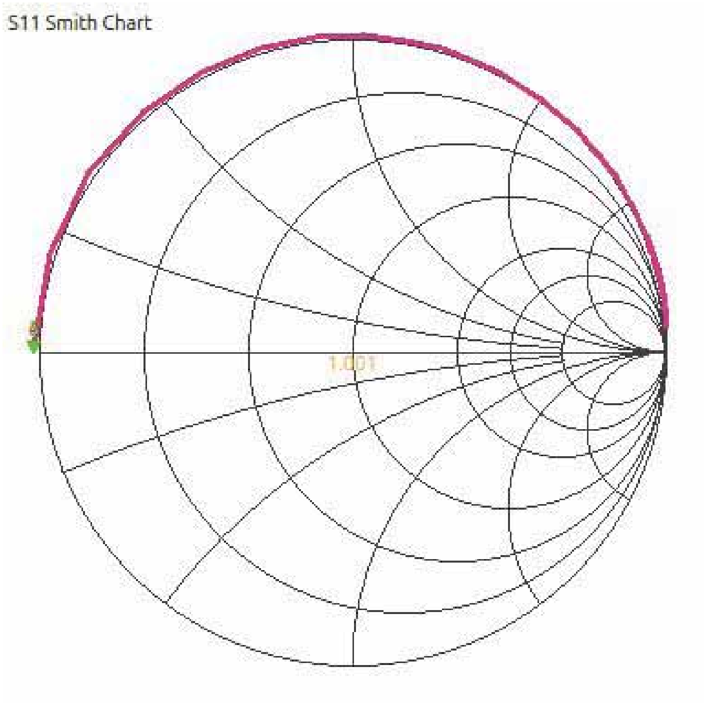


Figure 2 – A 20 µH air-core inductor, swept from 300 kHz to 20 MHz.

inductive reactance region, throughout the sweep's frequency range.

Lest you get the impression that somehow inductors are inherently "cleaner" than capacitors, **Figure 3** shows a 10 mH, scramble-wound inductor swept over the same frequency range. Yikes! We have *multiple* resonances, both series and parallel, and a lot of resistance as well.

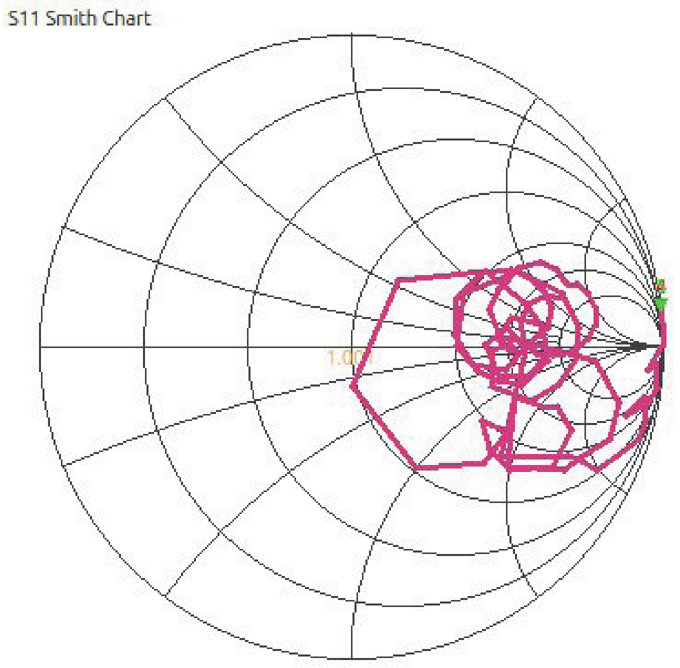


Figure 3 – A 10 mH inductor, swept from 300 kHz to 20 MHz.

Series Versus Parallel

It is simple to distinguish between series and parallel resonances. With increasing frequency, if you cross the horizontal axis *upward* (from capacitive toward inductive), it's a series resonance. If you cross the horizontal axis *downward* (from inductive toward capacitive), it's a parallel resonance.

Older texts refer to parallel resonance as *anti-resonance*, which actually has a special significance when dealing with linear circuits like wire antennas. A center-fed dipole antenna is never an actual parallel circuit, but it exhibits anti-resonant characteristics at even harmonics. The increasingly popular end-fed half-wave (EFHW) antenna exhibits anti-resonances on all harmonics, including the fundamental. This is why EFHW antennas can be very touchy, as the anti-resonances are much narrower. (We will explore this in depth in a future essay).

Touching Touchstone Again

Here is part of the Touchstone file for our capacitor sweep. (The actual file is about a thousand lines long; this is only a sample.) The important thing is the header. Here are the parameters:

: This simply shows this IS the header
HZ: Representing default units of Hz
S: This tells us that the columns are Scatter (S) parameters. Impedance (Z) parameters are available, too.
RI: Values are Real and Imaginary (default). This can also be Mag/Phase
R+(value): This is the normalized impedance (default 50 Ω)

And that's about it. Because this is a single-port measurement, we only have two columns of sampled data, the I and Q values. These are floating point values, so if you load Touchstone files into a spreadsheet, be sure to use the floating-point cell format.

```
# HZ S RI R 50
3000000 0.9909985165431382 -0.12832893684429195
3470000 0.9887472778189667 -0.149274397691316
3940000 0.9855569844176474 -0.17000288782332693
4410000 0.9824072692514252 -0.19089412854071278
4880000 0.9784494861701987 -0.21217355795448492
5350000 0.9741258481266285 -0.23302647081568842
5820000 0.9692509190524822 -0.25371618410953795
6290000 0.9638927186883427 -0.2752251336609704
6760000 0.9571927796774888 -0.29651903591464235
7230000 0.9513886595972583 -0.31677493478294827
7700000 0.9439359716977733 -0.33779769797427533
8170000 0.9367180687681502 -0.358351725269076
8640000 0.9288427996450602 -0.37891789272287485
9110000 0.9201735290133214 -0.39964983094274964
9580000 0.9112679007404662 -0.42025612790081174
10050000 0.9019400731610189 -0.4413269884111751
10520000 0.8915551516891611 -0.4617767898188182
10990000 0.8814125565281417 -0.48234940701382606
```

And that's about it. In our next essay, we will dive into the less-used (at least by hams) two-port measurements, something not normally available with common antenna analyzers, which only make single-port measurements. *— Until then! 73, Eric, KL7AJ*

Upcoming Conferences

SCaLE 22x

March 6 – 9, 2025

Pasadena, California

www.socallinuxexpo.org/scale/22x

The 22nd Annual Southern California Linux Expo, SCaLE 22x, will take place Thursday, March 6 through Sunday, March 9, 2025, at the Pasadena Convention Center in Pasadena, California.

HamSCI 2025 Community Workshop

March 14 – 15, 2025

Newark, New Jersey

https://hamsci.org

HamSCI, the Ham Radio Science Citizen Investigation, is pleased to announce that its 2025 community workshop will be held Friday, March 14 through Saturday, March 15, 2025, at the New Jersey Institute of Technology in Newark, New Jersey. This annual HamSCI meeting will provide an opportunity for both the amateur radio and professional research communities to collaborate and share ideas. See website for details.

2025 SARA Western Conference

March 14 – 15, 2025

Socorro, New Mexico

www.radio-astronomy.org

The 2025 SARA Western Conference will be held at the Pete V. Domenici Science Operations Center in Socorro, New Mexico, and the Very Large Array near Magdalena, New Mexico, Friday, March 14 through Sunday, March 16, 2025.

Southeastern VHF Society Conference

April 4 – 5, 2025

Clarksville, Tennessee

http://svhfs.org/wp/2025-conference

The Southeastern VHF Society will hold its 2025 Annual Conference on Friday, April 4 through Saturday, April 5 at the Quality Inn in Clarksville, Tennessee.

Aurora’25 Conference

June 7, 2025

Plymouth, Minnesota

www.nlrs.club

Aurora, the largest annual gathering of weak-signal VHFers in the Upper Midwest, will be held Saturday, June 7, 2025, at the West Medicine Lake Community Club, 1705 Forestview Lane North, Plymouth, Minnesota.

2025 Central States VHF Conference

July 24 – 27, 2025

Lincoln, Nebraska

https://2025.csvhfs.org

The 2025 Central States VHF Conference will be held Thursday, July 24 through Sunday, July 27, 2025, at the Lincoln North Hotel and Conference Center in Lincoln, Nebraska.

Reach New Heights!

Top Towers Available

at DX Engineering.

Make DX Engineering your source for the most trusted towers in amateur radio. Find products from American Tower Co. (Amerite), Rohn, TBX, and Universal at DXEngineering.com, plus big advantages: knowledgeable advice, expert packaging, and fast shipping that rises above the competition.

Also choose from DX Engineering masts, tower accessory shelves, and advanced thrust bearings; Phillystran guy lines; Carlson heavy-duty base plates; and much more!

Make the Change to a More Satisfying Ham Radio Purchasing Experience

- Easy ordering by phone or web
- Products from over 170 leading manufacturers
- Friendly customer service from hams with a combined 1,000+ years in amateur radio
- Fastest shipping in the industry
- Responsive and ongoing technical support
- Not 100% happy? We make it right!

Order by Phone
800-777-0703 Country Code: +1
9 am to midnight ET, Monday-Friday
9 am to 5 pm ET, Weekends

Order Online
www.DXEngineering.com
Most orders over $99 ship free
Request a Free Catalog

Tech Support
330-572-3200
DXEngineering@DXEngineering.com
9 am to 7 pm ET, Monday-Friday
9 am to 5 pm ET, Saturday

DX 25 ENGINEERING™

PREFORMED LINE PRODUCTS
ROHN™ Products LLC
American Tower Company
TBX Tower

Come See What Icom Has To Offer In The New Year

IC-7760
HF / 50MHz

IC-PW2
HF / 6M 1 kW Amplifier

IC-9700
2M / 70CM / 23CM

IC-705
HF / 6M / 2M / 70CM

IC-7300
HF / 6M

IC-7610
HF / 6M

For the love of **ham radio.**

www.icomamerica.com/amateur
insidesales@icomamerica.com

ICOM®