QEX

A Forum for Communications Experimenters

July / August 2023
Issue No. 339 | www.arrl.org

G8AGN designs an antenna pointing system.

Get Ready for SOTA, POTA, YOTA, and 13 Colonies!

Coaxial Cable Assemblies

These low-loss cable assemblies are available in standard lengths with DX Engineering's revolutionary patented PL-259 connector. Use the online Custom Cable Builder at DXEngineering.com to build assemblies made to your exact specs. DX Engineering's coaxial cable is also available by the foot or in bulk spools. **Enter "DXE Assemblies" at DXEngineering.com. From $26.24**

DX ENGINEERING kelemen VIBROPLEX CHAMELEON ANTENNA

DXE-WA-135

Wire Antennas

If you're looking for the precise wire antenna to meet your operating needs, you'll find it at DX Engineering! Our lineup includes DX Engineering's own rugged and lightweight Multi-Band Dipole Antenna Kits that come with wire elements, ladder feedline, center-T support, and end insulators; 47 models of Kelemen Trap Dipole Antennas; Par End-Fedz® End-Fed Half-Wave Antennas; and Chameleon wire antennas, including MPAS Backpack Antenna Systems. **Enter "Wire Antenna" at DXEngineering.com for more details, manufacturers, and models.**

bhi Noise Cancellation Products HEIL HAM RADIO INRAD international radio

Headsets and Headphones

DX Engineering carries a great selection of hands-free headsets and state-of-the-art headphones from bhi, Heil, INRAD and other top brands. Don't accept anything less than clear, intelligible speech fidelity whether you're doing the speaking or listening. **Enter "Audio" at DXEngineering.com.**

Bioenno Power

Batteries, Chargers, and Solar Products

DX Engineering now carries state-of-the-art LiFePO4 (Lithium Iron Phosphate) batteries from Bioenno Power—a company with a proven track record of producing reliable, longer-lasting power solutions for portable ops. Choose from Bioenno's 12V LFP series (capacities from 3-20Ah; maximum discharges from 7-40A) and AC to DC LiFePO4 Battery Chargers. Bioenno offers customers its True Lithium Capacity Assurance policy, meaning they individually inspect and quality-check every battery before shipment. Also available are lightweight, foldable solar panels (28W to 120W) and Solar Charge Controllers. **Enter "Bioenno" at DXEngineering.com. Batteries from $49.99; Solar panels from $104.99**

ICO-LC-192

ICO-AH-705

ICOM ICO-IC-705

IC-705 HF/50/144/430 Portable Transceiver

With the features and functionality of the IC-7300, IC-7610, and IC-9700, this popular Icom QRP rig is like owning a base transceiver you can hold in one hand. It boasts SDR Direct Sampling technology for stellar transmit and receive performance; 4.3" color touchscreen; real-time spectrum scope and waterfall display; built-in Bluetooth®; wireless LAN; and full D-STAR capabilities. IC-705 accessories include backpack (ICO-LC-192) and compact automatic tuner (ICO-AH-705). **Enter "IC-705" at DXEngineering.com.**

DX ENGINEERING

Premier Telescoping Carbon Fiber Masts

When you demand superior long-lasting performance, these high-strength masts make an excellent choice for portable and temporary antenna supports. Lighter than aluminum and fiberglass and stiffer than steel of the same thickness, these masts feature short section lengths designed for low-cost shipping and easy transport. UV- and corrosion-resistant, they include adjustable lever-action clamps that secure the next smaller tube in place and allow for fast and simple raising and lowering. **Enter "DXE Carbon Fiber" at DXEngineering.com.**

DXE-TCFP-24	24' Mast, 4 Clamps, 5 Sections, Collapses to 69"	**$299.99**
DXE-TCFP-33	33' Mast, 6 Clamps, 7 Sections, Collapses to 69"	**$469.99**
DXE-TCFP-33HD	Heavy-Duty 33' Mast, 6 Clamps, 7 Sections, Collapses to 69"	**$599.99**
DXE-TCFP-49	49' Mast, 9 Clamps, 10 Sections, Collapses to 81"	**$849.99**

AlexLoop

AlexLoop HamPack Portable Magnetic Loop Antenna System

Tripod not included

PY1AHD, Alexandre Grimberg brings more than five decades of Amateur Radio experience to the new AlexLoop HamPack, the ultimate magnetic loop antenna solution for portable operating. The HamPack comes with the widely acclaimed transceiver QRP 40-10M AlexLoop antenna; reinforced, full-size backpack that accommodates the antenna, accessories, and any size QRP rig; and upgraded, easy-to-use tuner. **Enter "AlexLoop" at DXEngineering.com. $599.00**

YAESU ICOM KENWOOD ALINCO COMET VIBROPLEX samlexamerica NANUK

Visit Our Website to Get Your Copy of the New Catalog!

DX® ENGINEERING

Ordering (via phone) Country Code: +1
9 am to midnight ET, Monday-Friday
9 am to 5 pm ET, Weekends
Phone or e-mail Tech Support: 330-572-3200
9 am to 7 pm ET, Monday-Friday
9 am to 5 pm ET, Saturday
Email: DXEngineering@DXEngineering.com

Ohio Showroom Hours:
9 am to 5 pm ET, Monday-Saturday
Ohio Curbside Pickup:
9 am to 8 pm ET, Monday-Saturday
9 am to 7 pm ET, Sunday
Nevada Curbside Pickup:
9 am to 7 pm PT, Monday-Sunday

800-777-0703 | DXEngineering.com

ON ALL BANDS YouTube

Email Support 24/7/365 at DXEngineering@DXEngineering.com

Prices subject to change without notice. Please check DXEngineering.com for current pricing.

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. 339

Publisher
American Radio Relay League

Kazimierz "Kai" Siwiak, KE4PT
Editor

Lori Weinberg, KB1EIB
Assistant Editor

Ray Mack, W5IFS
Contributing Editors

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

Jodi Morin, KA1JPA
Assistant Production Supervisor

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
Fax: 860-594-0259 (24-hour direct line)
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.

July/August 2023

About the Cover

Barry Chambers, G8AGN, bases an antenna pointing system on the u-Blox C94-M8P-3 evaluation kit. Accurate alignment of microwave antennas is particularly important when using electrically large dishes or when operating QRP, especially on mm-wave bands such as 122 GHz. Real-time kinematic, differential global navigation satellite system, (RTK DGNSS) can provide extremely high-quality information about the separation and compass bearing between two global navigation satellite system (GNSS) modules, normally referred to as the Base and the Rover. This information can then be used to calibrate a rotatable protractor mounted on the operator's dish antenna tripod. It is then a simple matter to point the dish accurately in the direction of a distant station whose beam heading has been determined previously or by calculation from known latitude and longitude coordinates, or Maidenhead locators.

G8AGN designs an antenna pointing system.

In This Issue:

Index of Advertisers

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.

Kazimierz "Kai" Siwiak, KE4PT

Perspectives

A Modern Look

You may have noticed a change in the *QEX* styles that were implemented in the May/June issue. Notably, the cover sports a new fresh look and new logo. Fonts in the articles have taken on a more modern appearance, but are still serif so that variables in text can match fonts used in equations without ambiguities. We have retained the visual enhancements introduced in 2019. At that time, and by reader request, we implemented readability enhancements. All mentions such as **Figures 1** and **Table 1** are in bold typeface so that they can be easily spotted by rapidly scanning an article. Previously, references and notes were called out in barely readable superscripts. Since 2019 they are numerals set in square brackets like **[1]** and are in bold typeface. So too are equation references like **(1)** set in parentheses, and are all in bold typeface. All these readability features will be retained.

We hope that you find the new *QEX* look pleasing and find the retention of previously implemented readability enhancements satisfactory.

In This Issue:

- Alan Victor, W4AMV, characterizes voltage and current baluns.
- Anthony Le Cren, F4GOH/KF4GOH, creates a compact APRS module.
- Barry Chambers, G8AGN, bases an antenna pointing system on an evaluation kit.
- Remigiusz Lecybyl, MØLRH, and Hieronim Lecybyl, M7HBL, experiment at 30 THz.
- Michelle Thompson, W5NYV, investigates the Opulent Voice protocol.
- Eric P. Nichols, KL7AJ, in his Essay Series, discusses the Vector Network Analyzer.

Writing for *QEX*

Please continue to send in full-length *QEX* articles, or share a **Technical Note** of several hundred words in length plus a figure or two. *QEX* is edited by Kazimierz "Kai" Siwiak, KE4PT, (**ksiwiak@arrl.org**) and is published bimonthly. *QEX* is a forum for the free exchange of ideas among communications experimenters. All members can access digital editions of all four ARRL magazines: *QST*, *OTA*, *QEX*, and *NCJ* as a member benefit. The *QEX printed edition* is available at an annual subscription rate (6 issues per year) for members and non-members, see **www.arrl.org/qex**.

Would you like to write for *QEX*? 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.

Very kindest regards,
Kazimierz "Kai" Siwiak, KE4PT
QEX Editor

Alan Victor, W4AMV
8809 Ross Court, Raleigh, NC 27613 | **amvictor@alumni.ncsu.edu**

Transformer Balun Circuit Characteristics Revealed

Voltage and current baluns are investigated.

The balun is investigated as an extension of the ideal transformer. The application of scattering parameter measurements highlights several basic circuit attributes. Simplified circuits show that the transformer voltage balun and the current balun are nearly identical except for one tiny subtlety. The S parameters assist in showing the differences between these balun types and provide a method for quantifying the balun's performance. Measurements for several balun configurations are used to Illustrate the simplified circuit approach. The current mode balun is emphasized and measurements over a frequency range of 10 kHz to 1 GHz are presented.

Fundamentals

The balun is a popular circuit element and one that presents challenges in achieving good performance. This is evident as the IEEE for several years sponsored contests to participants to strut their dB's in well-designed balun structures **[1-3]**.

The balun is a mode converter **[4]**. Single ended circuits or devices with one port grounded and one port active relative to ground may need to have both ports active and referenced to an external ground **[5]**. That is to say, both ports must be floating with respect to a common ground node. These circuits are differential and they require well balanced signals as their input. Hence, the balun (bal_un) provides this mode conversion, from balanced differential to unbalanced, signal ended. This idea may be extended to BALBAL and UNUN devices to handle that group of circuits, namely balanced to balanced and unbalanced to unbalanced. The motivation is usually to achieve dc isolation between circuits.

The transformer if configured properly achieves this balun operation quite well. The ideal transformer circuit will exhibit all the properties required to achieve balun characteristics and provides an easy way to differentiate voltage balun versus current balun functionality. Linear circuit analysis is used to study the transformer operation as a balun and uncover some subtleties. These include return loss with balanced and unbalanced terminations, magnitude of the currents in the terminations and their phase values versus transformer connections and what is revealed with S parameters. Linear small signal circuit analysis is useful and the tools available provide quick verification **[6,7]**.

The transformer operation is addressed only through magnetic coupling. The high frequency effects that are attributed to transmission line transformers or TLT devices are not considered **[8]**. However, this simplified assumption is shown through experiments to be a good approach up through 1 GHz.

Simplified Circuit

A balanced to unbalanced transformation is encountered frequently in an antenna interface. The analysis of a small dipole takes the current into each thin wire as equal in magnitude and 180 degrees out of phase **[9]**. The simple ideal transformer in **Figure 1** will provide the correct interface between an unbalanced transmission line (coax) and the antenna. The ideal transformer provides uniform and 100% coupling between the primary and the secondary inductor. In reality there are a number of obstacles that prevent this ideal operation. For example, the parasitic capacitive coupling between primary start winding and secondary winding is not symmetric. This non equal C coupling contributes to imbalance. One solution is to introduce an electrostatic shield. However, in the cases considered here, this shield is not used.

The currents in the loads, RL1 and RL2 of **Figure 1** are equal and 180 degrees out of phase. Important to note, these currents are always identical and constant in value as long as the sum of the loads is a constant.

The key aspect of the transformer in **Figure 1** is the series connection of the loads in the secondary. The secondary loop current is forced to be equal in each load. Furthermore, each of these load resistors is reflected to the primary winding in series and in proportion to the inductance ratio. Hence, in this example, the input resistance seen looking into the primary is 200 Ω. It will always be 200 Ω as long as the sum of RL1 and RL2 is 200 Ω. Although the currents are equal in magnitude, they are 180 degrees out of phase due to Lenz's law.

While the currents remain balanced as the terminations vary,

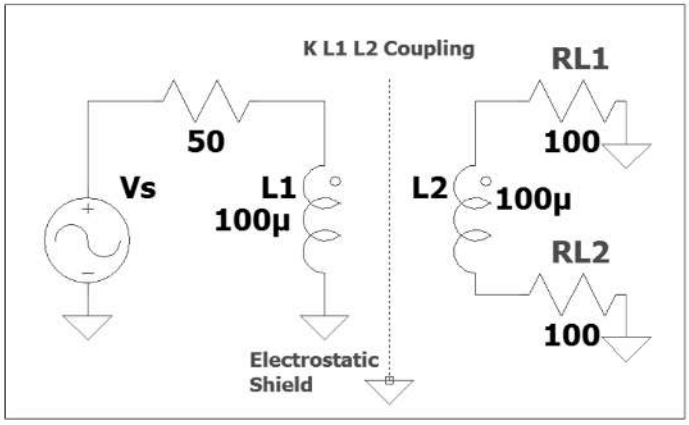


Figure 1 — An ideal transformer operating as a current balun.

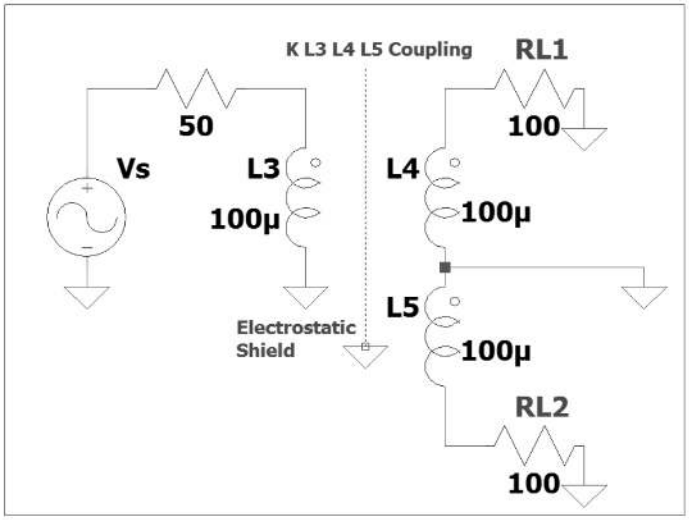


Figure 2 — An ideal transformer operating as a voltage balun.

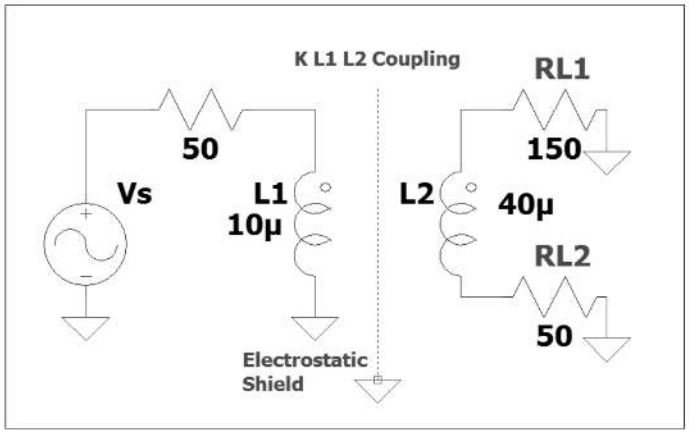


Figure 3 — A basic current balun with a 1:4 inductance ratio.

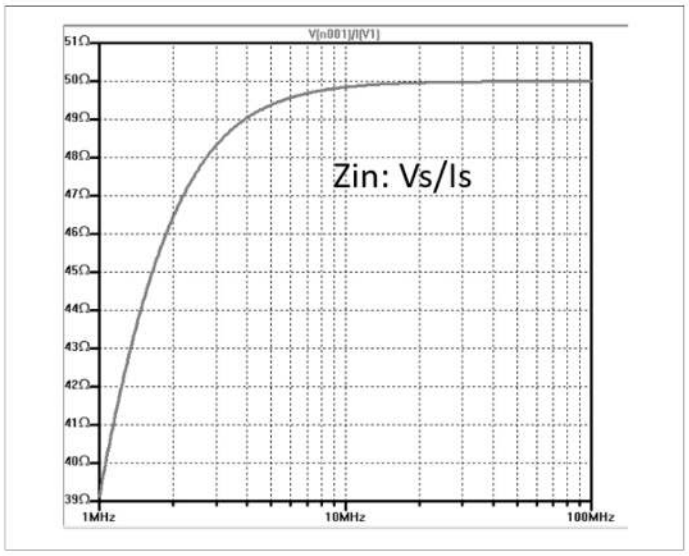


Figure 4 — The input impedance of Figure 3 as a function of frequency.

the voltages across the terminations do not. This balun is not a voltage balun. It does not preserve balanced output voltages except for one case, when RL1=RL2.

Contrast this operation to **Figure 2**. There is center tap introduced and a pair of identical secondary inductors as used in **Figure 1**. The secondary loop current is now divided while the secondary output voltages are in parallel. The output voltages will be balanced, 180 degrees out of phase, however the currents will not be balanced. Furthermore, in order to obtain the same 200 ohms, input resistance, the loads RL1 and RL2 would need to be 400 Ω. In this circuit, the loads RL1 and RL2 are reflected as parallel resistors in series with the primary winding. Hence, the input impedance will not maintain a constant value with varying terminations, even if the terminations sum to an identical constant as in **Figure 1**.

The curious aspect of these two circuits is the possibility of obtaining a current balun directly from a voltage balun simply by floating the center tap. Furthermore, the behavior of each configuration with varying load terminations provides significant insight into their performance.

In **Figure 2** a center tap is introduced and a second identical coupled inductor is added to figure 1. In this figure the input impedance is 50 Ω. As the terminations vary so do the secondary currents. However, the load voltages although they vary in value are matched and hence balanced.

A Simulation Checkpoint, A Basic Current and Voltage Balun

A 1:4 current balun is a popular configuration as it provides a convenient impedance transfer from an unbalanced 50-Ω coaxial line to a variety of antennas with balanced terminal impedances near 200 Ω. The circuit simulation is set to sweep frequency from 1 MHz to 100 MHz and the terminations must sum to 200 Ω. The inductance of the primary and secondary are chosen to be lower than desired so their effect on the input impedance is noticed at the lower frequency limits.

Figure 3 shows a basic current balun with a 1:4 ratio inductance, which will provide an impedance match from 200 Ω to 50 Ω. The primary to secondary inductance ratio must be 4 to 1. Note, the terminations are not equal. The input impedance is not termination sensitive. All that is required is their sum total 200 Ω to provide a matched condition.

In **Figure 4** the input impedance of **Figure 3** is found as a

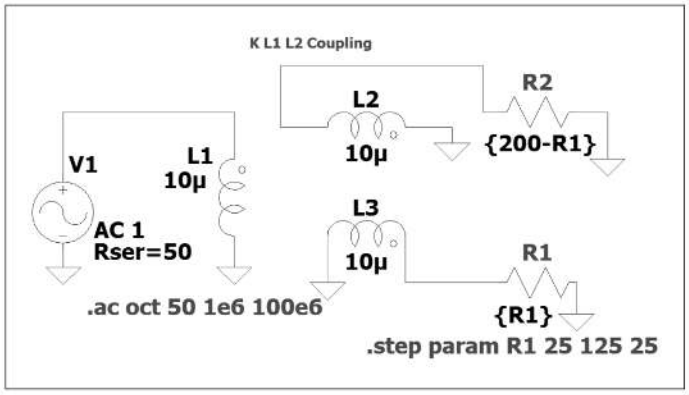


Figure 5 — The voltage balun with simulation set to step the terminations.

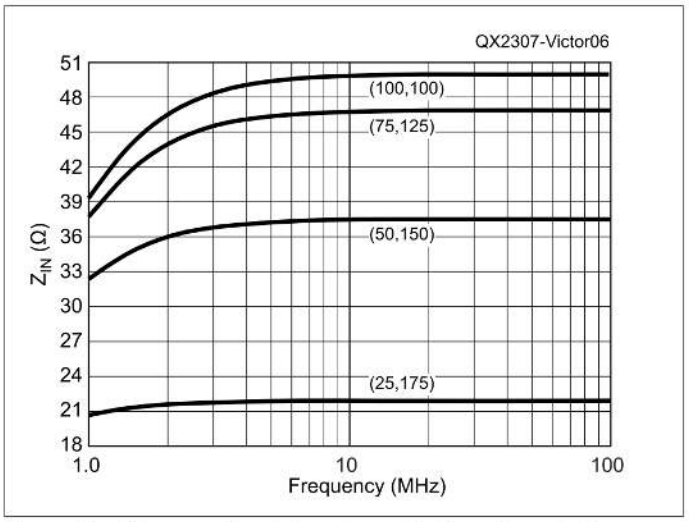


Figure 6 — The swept input impedance for the voltage balun of **Figure 5**.

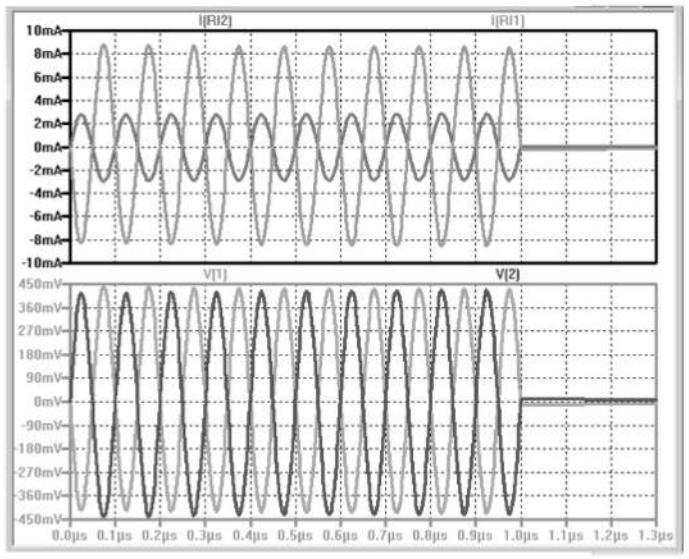


Figure 7 — The signals on the transformer voltage balun.

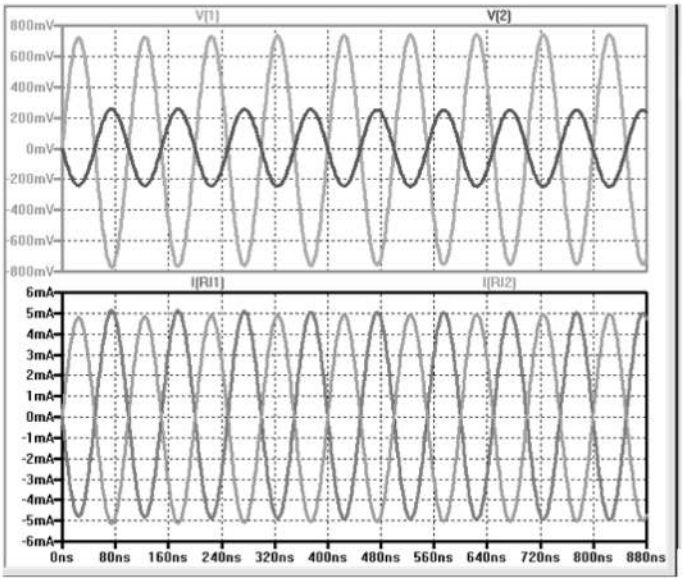


Figure 8 — The signals on the transformer current balun.

function of frequency. A 1-V source signal, Vs, is applied at the input and the source current calculated. The inductance primary to secondary ratio controls the input impedance. The terminations are swept but this transformer arrangement is termination insensitive. At low frequency, the reactance of the transformer is not sufficient.

Next, the current balun of **Figure 3** is provided with a center tap connection as shown in **Figure 5**. Now the ability to provide voltage balance is possible. However, if the terminations are stepped a significantly different result occurs. This configuration is termination sensitive. The desired 50-Ω input match occurs only for R1=R2 equal to 100 Ω. Again the primary and secondary inductances are smaller than desired and the frequency response is shown in **Figure 6**.

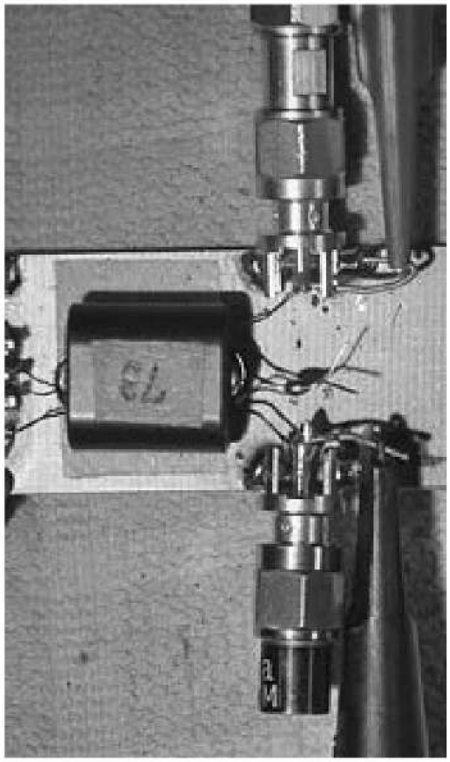
This configuration of **Figure 5** does not provide current balance and preserves an impedance match only when R1 and R2 are set to 100 Ω.

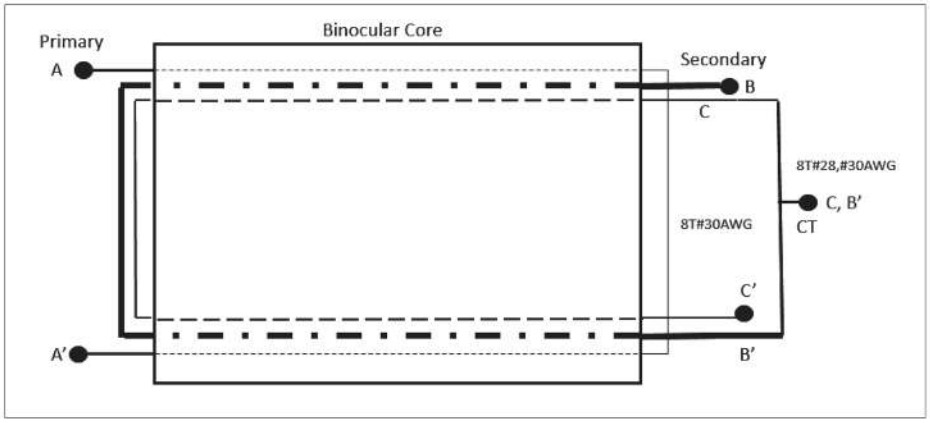
Figure 6 shows the swept input impedance for the voltage balun of **Figure 5**. The terminations are stepped. The voltage balun is termination sensitive. The roll off in low frequency input impedance is due to insufficient primary and secondary inductance.

Balun Output Voltage and Current with Swept Terminations

It is useful to try the different scenarios for the two simple transformers configured as a balun. The terminations are not equal but sum to a constant value. It is more revealing to consider unequal terminations as unequal terminations provide a reading on the performance of the transformer acting as a balun. A constant sum value is selected as this will preserve the input match and input impedance. An ac transient simulation is set up and the terminations are varied. The circuits of **Figure 3** and **Figure 5** are used. Simulation is at 10 MHz and the sum of the terminations is 200 Ω. This termination value would provide a 50 Ω input when a 1:4 impedance transformation is used.

Figure 7 shows the signals on the transformer voltage balun. Note the currents are not balanced or equal, top pane. A current divider is in place with this topology. However, the voltages are balanced and equal although they are subject to change value as





◀**Figure 9 — A transformer current balun.**

▲**Figure 10 — The transformer 1:4 balun detail.**

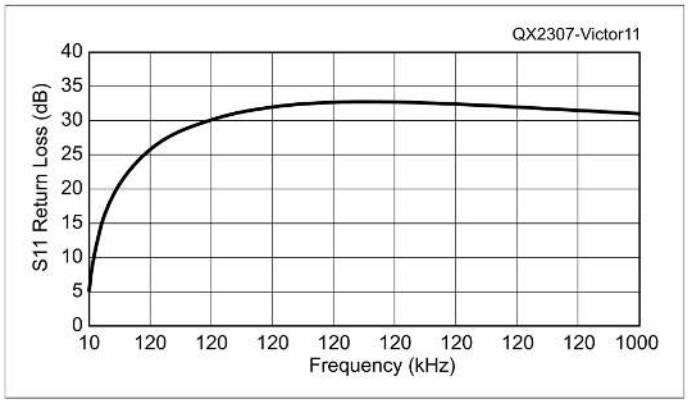


Figure 11 — The return loss of the transformer balun of Figure 9 displayed from 10 kHz to 1 MHz.

the terminations change, bottom panel. The two voltages are 180 degrees out of phase.

Figure 8 shows the signals on the transformer current balun. The currents are equal, balanced and 180 degrees phase shifted, bottom pane. The voltages are not balanced, top pane. As the terminations are varied, if their sum is constant, then the currents remain the same in value.

Investigation of how well the currents balance with variable terminations is a useful measure. It reveals the quality of the balun construction and hints to parasitic elements that are present. Evaluation of the transformer balun at low frequency is straightforward. A signal generator and oscilloscope are used, see **Figure 12** and **Figure 13**. Construction of a low frequency current balun based on the **Figure 2** schematic, but with the center tap floating is presented. The construction and measurements are discussed next.

A Low Frequency Transformer Current Balun

A two-hole core or binocular elongated toroid is used. See detail drawing of **Figure 10**. High permeability material is selected. Material 73 with a permeability of 2500 and a low frequency operation goal below 100 kHz is desired. The motivation is to keep the contribution of parasitic elements to a minimum and attempt to check the operation with the assumptions set forth so far.

Figure 9 shows a transformer current balun. Terminations of 150 and 50 Ω are used. Scope probes attached have reasonably high Z and low shunt capacitance for the frequency range of interest. The core is Fair-Rite type 2873000202. Eight turns of #30, #28 AWG enamel covered wire is used. Wires are not twisted; they are laid parallel with 1 turn defined as a single pass through both core openings.

The drawing of **Figure 10** shows the transformer 1:4 balun detail. Place primary down first. Start and finish wire on one side of core. Then place down a pair of secondary wires on the opposite side of the core. Pass all wires through both sides of the core 8 times. Secondary center tap formed as shown.

A variety of terminations are tried. Their sum is always 200 Ω. Hence, (150,50), (180,20), (100,100) and so on are paired possible terminations. Test frequencies are 100 kHz, 1 MHz, 10 MHz and as low as 10 kHz was possible. In order to first validate that the proper impedance transformation is occurring, a network analyzer is used. The nanoVNA **[10]** provides a reflection coefficient measurement as low as 10 kHz. This is a S11 measurement. More discussion on using S parameters for performance validation is presented in another section. The VNA is calibrated from 10 kHz to 1 MHz and the return loss validating the impedance transform is shown in **Figure 11**. The return loss of the transformer balun of **Figure 9** displayed from 10 kHz to 1 MHz. Terminations are 150 and 50 Ω. A return loss of 10 dB is better than a 2:1 SWR. This configuration displays a good return loss, over 20 dB, from 50 kHz to 1 MHz.

Figure 12A shows scope photos of the voltages across the terminations at 50 kHz. **Figure 12B** is at and 5 MHz. The terminations are 150 and 50 Ω. Hence, for the currents to be balanced, the voltages will be in a ratio of 3:1. Subsequent measurements show the unit is capable of operation to 5 MHz and slight degradation in current balance starts at 10 MHz.

Figure 13 shows a measurement conducted at 15 MHz. The current balance is degraded. If the reflection coefficient is mea-

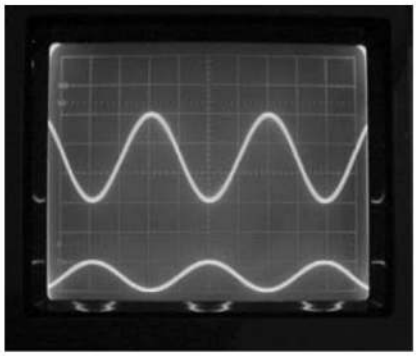


Figure 12A — Scope photos of the voltages across the terminations at 50 kHz.

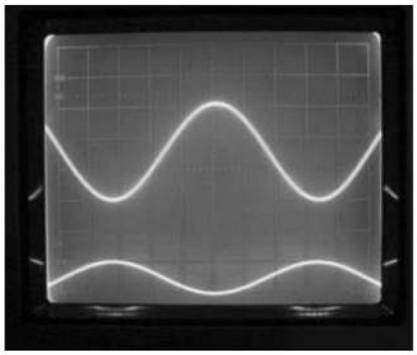


Figure 12B — Scope photos of the voltages across the terminations at 5 MHz.

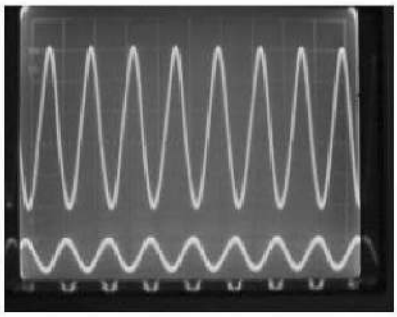


Figure 13 — Measurement conducted at 15 MHz shows the current balance is degraded.

sured, there is strong correlation to the quality of the return loss and the current balance. Deviation in good current balance begins to appear in this case around 10 MHz.

Measurements of the transformer balun continue to 20 MHz, **Figure 14**, and at that point the return loss is 10 dB, about a 2:1 SWR point. Thus this simple transformer balun provides a reasonable bandwidth that extends to over a decade in frequency, from 50 kHz to 5 MHz.

Figure 14 shows the return loss of the transformer current balun at 10 MHz is 15 dB. At 15 MHz it is 12 dB and continues to degrade with increasing frequency. As a reference, the return loss at 5 MHz is 20 dB. This would imply that a reasonable minimum return loss requirement to maintain current balance should be better than 15 dB.

Measurements of the voltages and currents directly and their balance is easy to accomplish with an oscilloscope. Finding the quality of the balance is accomplished by varying the terminations and measuring the ratio of voltages or currents at the output ports. As the frequency increases this task is more involved and not as accurate when compared to a swept frequency method. The *S* parameters provide a vehicle for obtaining the swept frequency response and as well for obtaining the quality of the balance. For the current balun, a convenient relationship between the output port's transmission parameters and the degree of balance is possible. Furthermore, this degree of balance is found as the terminations are varied without actually applying the unbalanced terminations in a measurement.

A summary of the observations noted between the transformer voltage and current balun are highlighted in **Table 1**. Some of these key observations will be revisited in more detail in the next sections.

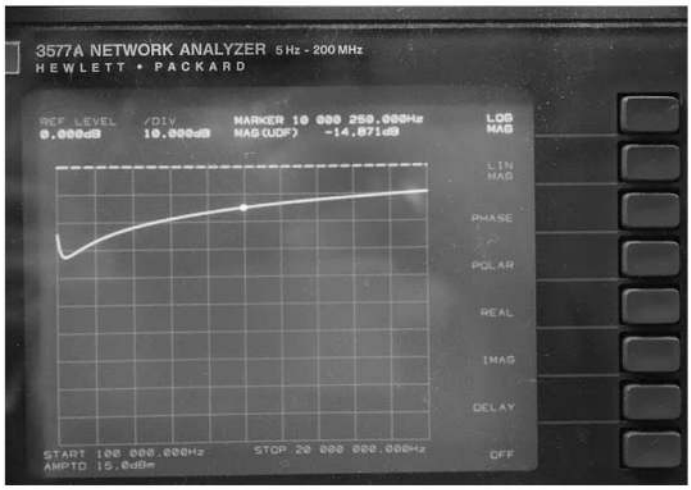


Figure 14 — The return loss of the transformer current balun.

Application of the Scattering Parameters to a Balun Measurement

A generalized transformer balun is at least a 3-port device without ground designated as one of the ports. Adding a floating ground port adds a fourth port. If a center tap port is brought out, then a 5-port device occurs. There are some configurations that have a sixth port connection, for example a center tapped primary.

The consideration of measuring such a device can be a daunting task. One approach is to use a single path full 2-port calibration and measurement, which is available on the nanoVNA. This measurement provides an accurate one-port reflection coefficient value, S_{11}, so the quality of the match is confirmed. It also pro-

Table 1 – A comparison of transformer voltage and current balun characteristics

Voltage Balun	Current Balun
Voltages are balanced and equal	Currents are balanced and equal
Zin does not remain constant with variable terminations	Zin remains constant with variable terminations
Voltages are 180 degrees out of phase	Currents are 180 degrees out of phase
Output voltage will vary with termination*	Output current will vary with termination*
Return loss varies with varying termination	Return loss is constant with varying termination
*The two output terminations vary, but their sum is constant.	

```
!Mini-Circuits Lab;
!S-parameters;
!Model:ADT4-1WT+; Unit 1;
!Run:CE1170; Date Code:1504; Temperature: 25°C;
!Fixture: 90-2-20-122/TA070; S/N 61541; Cal.Due: 07/06/2015;
!Connection: Port1 to pin1; Port2 to pin3; Port3 to pin4; port4 to pin5; Port5 to pin6; pin2-not connected.
!Tested with ENA Network Analyzer E5071C, S/N 74883, Cal.Due: 06/03/2015.

# MHZ S DB R 50
!Frequency        S11 dB  S11 DEG   S12 dB  S12 DEG   S13 dB  S13 DEG   S14 dB  S14 DEG   S15 dB  S15 DEG
!                 S21 dB  S21 DEG   S22 dB  S22 DEG   S23 dB  S23 DEG   S24 dB  S24 DEG   S25 dB  S25 DEG
!                 S31 dB  S31 DEG   S32 dB  S32 DEG   S33 dB  S33 DEG   S34 dB  S34 DEG   S35 dB  S35 DEG
!                 S41 dB  S41 DEG   S42 dB  S42 DEG   S43 dB  S43 DEG   S44 dB  S44 DEG   S45 dB  S45 DEG
!                 S51 dB  S51 DEG   S52 dB  S52 DEG   S53 dB  S53 DEG   S54 dB  S54 DEG   S55 dB  S55 DEG
    .0100     -3.848841E+001  8.826036E+001 -6.242197E-002 -8.343653E-001 -3.064975E+001 -9.708707E+001 -9.619342E+001  4.187637E+001 -9.690807E+000  1.699798E+002
              -6.580713E-002 -7.652208E-001 -3.810049E+001  8.368343E+001 -3.067833E+001  8.307143E+001 -7.105429E+001  1.603763E+002 -3.067686E+001 -9.666107E+001
              -3.115353E+001 -9.562821E+001 -3.114812E+001  8.445053E+001 -9.709048E+000  1.698945E+002 -3.585493E+000  6.841312E-002 -3.647499E+000 -4.999350E+000
              -7.594733E+001 -4.899826E+001 -7.793839E+001 -1.170406E+002 -3.583926E+000  7.856742E-002 -9.635839E+000 -1.798262E+002 -3.569535E+000  3.939349E-002
              -3.112477E+001  8.444880E+001 -3.112376E+001 -9.586889E+001 -3.643270E+000 -4.983755E+000 -3.569497E+000  7.167445E-002 -9.696358E+000  1.700264E+002
```

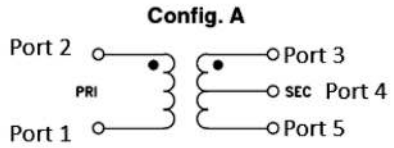


Figure 15 — 5 port S parameters for a 1:4 impedance transformer characterized in a 50-Ω system.

vides an accurate transmission measurement from the input port to a single output port. Hence all other ports need to be terminated. A little tedious but not unreasonable. Obtaining port balance requires an S_{21} and S_{31} measurement and comparing their values. There is one caveat. If for example the balun is a 1:4 impedance transformer and the measurement system characteristic Z_0 is 50 Ω then a 50-Ω termination is inappropriate to obtain true measure of balance. The port to port output impedance must be 200 Ω. The solution to this problem is to provide a resistive load match that creates a port to port termination to the transformer of 200 Ω, while the analyzer is connected with its 50-Ω load termination **[11]**. This is accomplished by using a resistor pad termination. The pad will provide proper transformer load while interfaced to the 50 Ω VNA. At the same time, the 2nd output port will be equipped with the same arrangement of resistor pad termination.

There is another method which is adopted in this work. It is somewhat more straightforward. The transformer current balun under test is simply terminated in a 50-Ω system and standard S data collected. The next step requires using the transmission data or S_{21}, S_{31}, S_{41}, …, S_{jk} data to calculate load currents. The *k*th port is the input port, the *j*th port are all outputs. The calculation of load currents simply applies the definition of the transmission *S* parameters. The calculation of transmission coefficients or S_{jk} is easy to follow with an example developed for the voltage divider and presented in the **Appendix**.

Finding Current Balance Using S Parameters and the Transmission Coefficient

The definition of the S_{21} parameter or the transmission coefficient is **[12]**

$$S_{21} = -2\sqrt{Z_{01} Z_{02}}\,\frac{I_2}{V_{s1}} \tag{1}$$

The balance between ports is desired. The ratio of transmission coefficients will directly provide the ratio of the output currents. Then (1) is used as shown in (2),

$$\frac{S_{21}}{S_{31}} = \frac{\sqrt{Z_{02}}}{\sqrt{Z_{03}}}\,\frac{I_2}{I_3} \tag{2}$$

Since Z_{01} the source, is usually 50 Ω and a constant. The other ports may be terminated in values not equal to the Z_0 of the system. The current ratios, if the system is well balanced, will be close to unity. Hence, the *S* parameters are measured in a 50-Ω system and then the *S* parameter transmission coefficient ratios are scaled by the square root of the various terminations. This provides the current ratios, hence the current balance and ideally it should be unity for a perfectly balanced transformer.

Hence, the procedure used is to choose a set of terminations that sum to the correct total termination value. For example, a 1:16 balun operating in a 50-Ω system and providing an impedance translation would use 800 Ω. Resistor terminations of 750 and 50 Ω could be used. The 1:4 transformer balun investigated here used 200 Ω. Therefore, 150 Ω and 50 Ω as a set is one possible pair. Transformation of impedance can be in the other direction as well. So if 50 Ω is the termination, then 50 Ω would be the required for the sum termination for a 4:1 impedance ratio and an appropriate pair of resistors would be required.

The load terminations used here are (100,100), (150,50), (175, 25), (180,20) Ω. Investigating this testing concept is presented next. In that section a voltage balun is configured as a current balun and the method of *S* parameter application of transmission coefficients explored in more detail.

A UHF Voltage/Current Balun and its Properties

Testing this notion of scattering parameter transmission coefficients applied to accessing current balun performance is presented. The scattering parameters of this device under test are complete and well documented for the Mini Circuits MCL ADT4-1WT transformer **[13]**. This transformer has a floating center tap. Consequently, it is perfect to apply in a current or voltage mode and investigate. The *S* parameters provided are those of a 5 port, see **Figure 15**, which shows 5 port *S* parameters for a 1:4 impedance transformer characterized in a 50-Ω system. The data is provided over a frequency range of 10 kHz to 7 GHz. Hence it is a rich set of data. The configuration A in Figure 15 is mapped with its

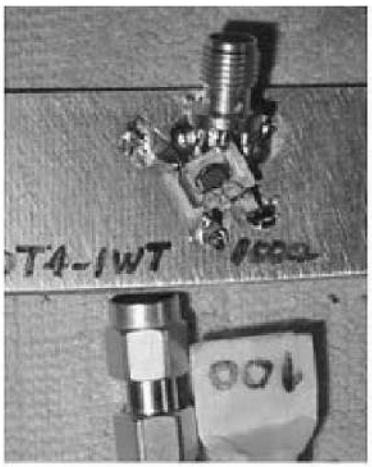


Figure 16 — Winding transformer type MCL ADT4-1WT flipped showing primary and secondary windings.

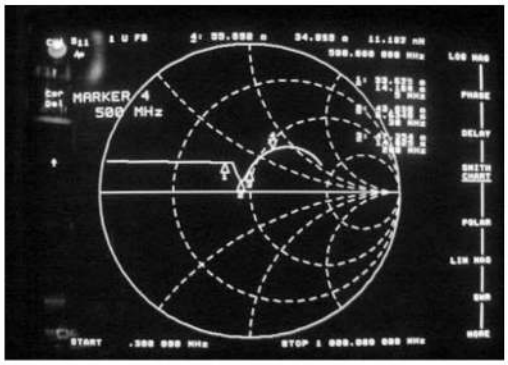


Figure 17 — Reflection coefficient, S_{11} of the transformer of **Figure 16**.

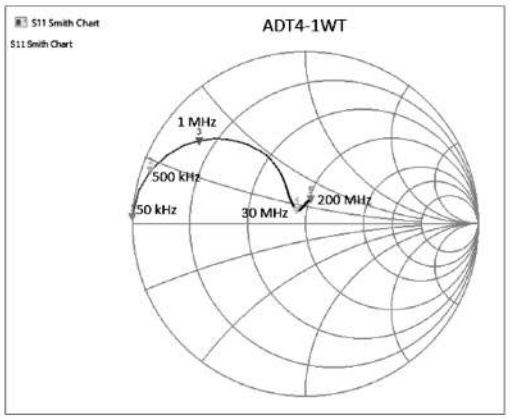


Figure 18— A low frequency sweep of input reflection coefficient, S_{11}, for the transformer.

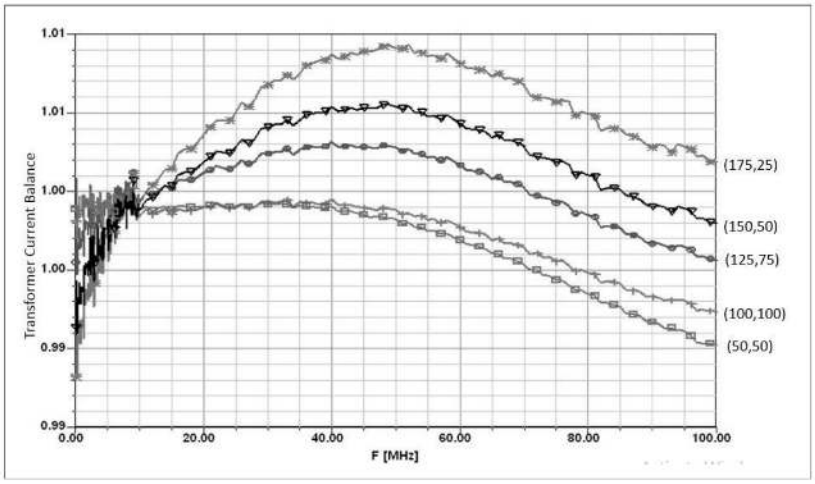


Figure 19 — The transformer is configured with center tap floating. At frequencies less than 100 MHz, current balance is excellent.

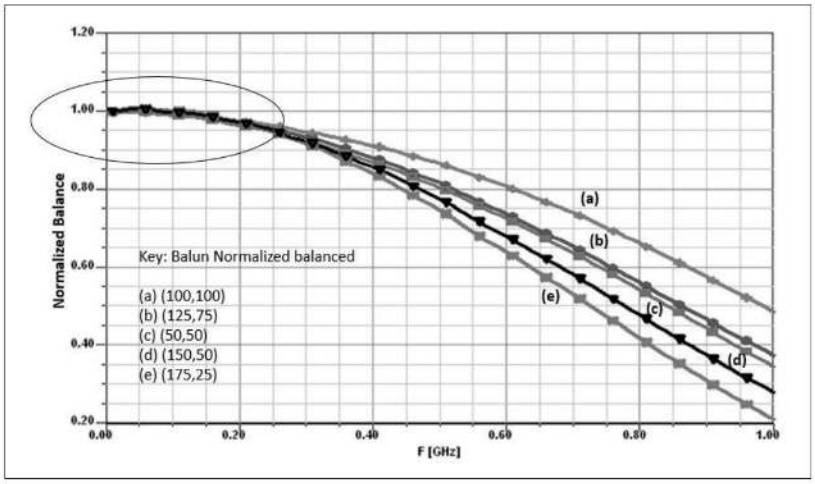


Figure 20 — Transformer configured as a current balun.

pins and corresponding ports. The format of the data set is "Touchstone" and is a standard in the industry.

The transformer under test is shown in **Figure 16** and is terminated in a pair of 100 Ω chip resistors. The reflection coefficient is shown in **Figure 17**. The frequency span is 300 kHz to 1 GHz. The display shows several noteworthy features. One, there is a shift in the reflection coefficient at lower frequency. There is a reduction in inductive reactance of the transformer at lower frequency and is similar to the results presented in **Figure 6**. The lower frequency reflection coefficient is detailed in **Figure 18** using the nanoVNA to extend below 300 kHz. At 500 MHz and above there is a combination of shunt capacitive reactance and leakage inductance, loss in coupling and hence a rotation in the reflection coefficient. The increase in series resistance is core and additive wire loss. There are probably fixture losses as well including connector losses. All of which are not de-embedded from this measurement.

Figure 17 shows the reflection coefficient, S_{11} of the transformer of **Figure 16**. The terminations are 100 Ω and calibration is a one port using short, open and 50-Ω load, 3.5 mm calibration pieces. There is a delay added to the calibration plane to align it with the SMA connector port physical length.

Figure 18 shows a low frequency sweep of input reflection coefficient, S_{11}, for the transformer. The analyzer extends the number of data points below 300 kHz providing a better view of the low frequency response.

The application of the *S* parameter data and the variation in a pair of termination resistors are used to generate a current balance graph, **Figure 19**. The 50 Ω *S* parameter ratio between port 2 and 3 is scaled by 1, 1.732, 1.29, 2.64 for pairs of terminations such as 100,100; 150,50; 125,75 Ω and so on. These ratios are current ratios and if current balance is perfect would equate to unity. This process is investigated for the center tap floating, **Figure 19** and **Figure 20**, and grounded **Figure 21**. In **Figure 19** the balanced ratio is focused at low frequency through 100 MHz. While **Figure 20** evaluates the balance through 1 GHz. There is clearly a degradation beyond 200 MHz which becomes more significant as frequency increases. A parasitic affect that shows up in the configuration of a voltage transformer operated as a current balun is evident above 400 MHz, see **Figure 20**.

Figure 19 shows the transformer is configured with center tap floating. At frequencies less than 100 MHz, current balance is excellent. **Figure 20** shows the transformer configured as a current balun. Terminations are varied and balance vs. frequency is

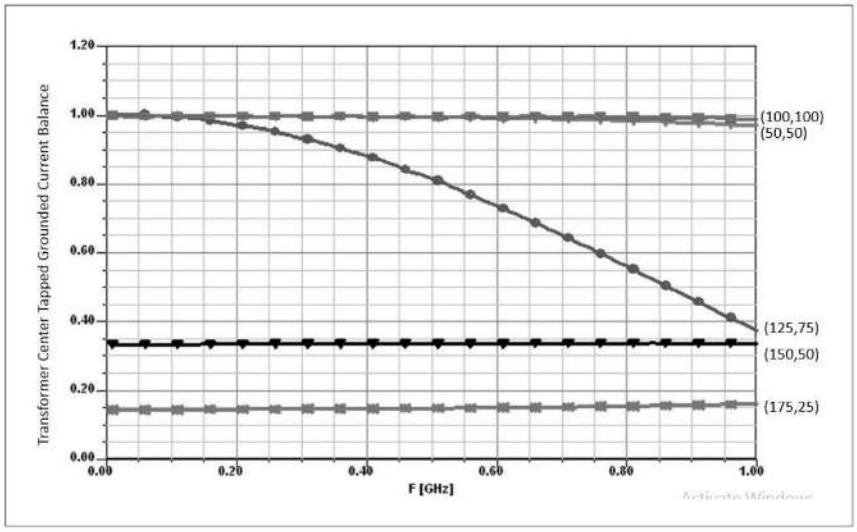


Figure 21 — Calculated current balance with various terminations, center tap grounded.

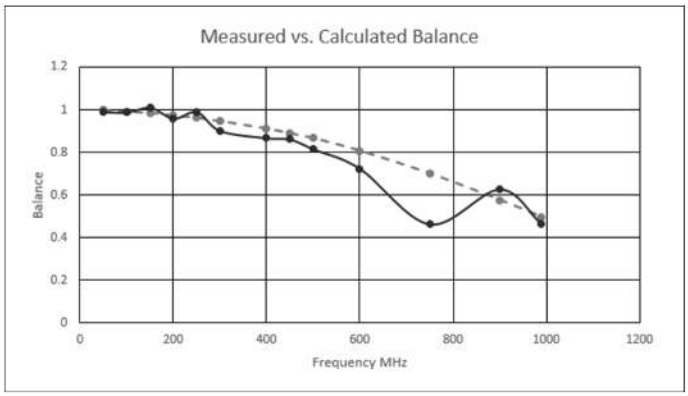


Figure 22 — Current balance from measured data (solid line) compared to calculated (broken line) with 100 Ω terminations for the MCL ADT4-1WT vs. frequency from 50 MHz to 1 GHz.

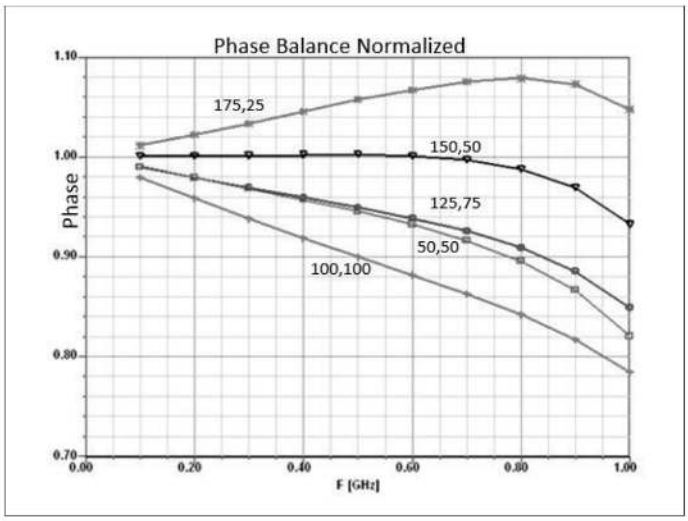


Figure 23 — Normalized phase imbalance versus terminations and frequency.

shown. Balance is significantly affected as terminations diverge from the set (100,100) Ω beyond 200 MHz. However, the utility as a current balun can be explored in lower frequency circuit applications. It is particularly interesting to compare the same calculation with the center tap grounded, see **Figure 21**. Now the current balance is significantly degraded except for the two termination cases, 50 and 100 Ω. This is clearly a voltage balun. Note that the current balance which does exist holds close to the desired unity ratio up through 1 GHz, but the termination range is limited to the pair of 100 Ω if a match is desired.

Figure 21 shows the calculated current balance with various terminations, center tap grounded. This data was obtained by post processing the *S* data tables in ANSOFT Designer=Serenade SV, which readily handles S data with multiple ports **[14]**.

As a check on the calculations, an experiment is conducted and measurements made using an active GaAs FET probe, signal generator and spectrum analyzer, see **Figure 22**. The data is gathered from each of the output ports each terminated into 100 Ω. The ratio of currents from the two output ports is plotted in **Figure 20**. The measured to calculated data agree quite well. However, around 750 MHz a resonance condition appears to exist that causes a null in the measured data on one of the output ports. The data points beyond 750 MHz clearly retraces quite close to the calculation to nearly 1 GHz.

Figure 22 shows the current balance from measured data (solid line) compared to calculated (broken line) with 100 Ω terminations for the MCL ADT4-1WT vs. frequency from 50 MHz to 1 GHz.

Phase Balance

Up to this point no mention of phase balance is discussed. The trend for phase imbalance appears to track amplitude imbalance. The phase imbalance is calculated using the *S* parameters in the same manner as the amplitude imbalance. A variety of terminations are applied and the phase of the output ports calculated versus termination and frequency. Since the output ports are 180 degrees out of phase, a normalized phase imbalance is defined as,

$$\text{phase imbalance} = \frac{ang\left(S_{31} - S_{21}\right)}{180°}$$

At low frequency where the parasitic elements contribute a small imbalance, the phase balance converges to unity. This is shown in **Figure 23**. For convenience, **Figure 20** is replotted over the frequency of 100 MHz to 1 GHz, see **Figure 24**. There are several items of interest to note. First, normalized amplitude and phase imbalance tend to converge to unity at lower frequency, below 200 MHz. Second, the terminations which minimize amplitude imbalance are different than those that minimize phase imbalance at the UHF range. While the termination pair (150,50) Ω provide a reasonably good phase balance, the pair (100,100) Ω are more suitable for amplitude balance.

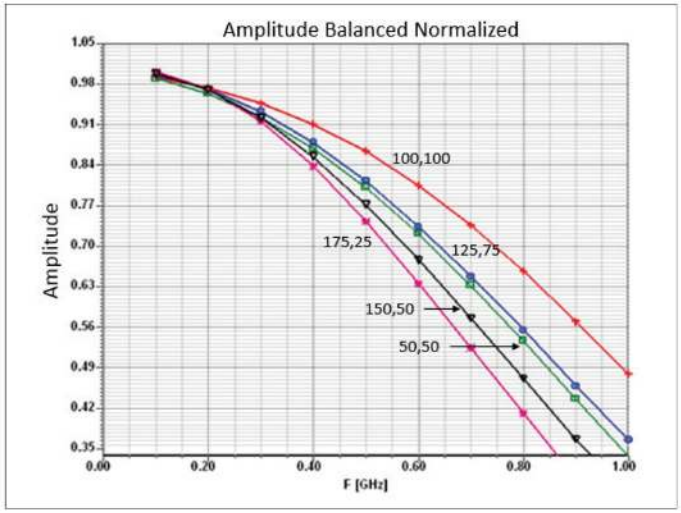


Figure 24 — Normalized amplitude imbalance versus terminations and frequency.

Conclusions

The ideal magnetic coupled transformer provides insight to operation of a voltage and current balun. When provided, a center tapped transformer will provide the features of either type of a balun. The voltage balun case occurs with the center tap grounded. The terminations are paralleled and the output voltages are forced to be equal. While with the center tap floating, terminations are in series, the secondary current is forced to be equal in both terminations. Gauging the performance with a variable set of terminations and the scattering parameters allows assessing the capability of either design. There will be a critical frequency controlled by parasitic elements that highlights the frequency range where one type balun would have improved balance performance over the other.

Appendix

It is easy to cite the voltage divider equation by inspection. However, it is not so apparent to obtain the transmission coefficient from input to output port expressed as S_{21} for the voltage divider. In the divider circuit, the divider has the same current in R_1 and R_2. KVL states that the sum of the voltage drops must equal the applied voltage. Hence $IR_1+IR_2=V_{s1}$. The output voltage is IR_2. But I is $V_{s1}/(R_1+R_2)$. Hence $V_{out}=R_2V_{s1}/(R_1+R_2)$ and the voltage divider rule results.

The S parameters add port source and load terminations to the circuit. The objective is to find the output current provided to a termination for a given source voltage with a specified series termination. The circuit now has added elements, terminations, that need addressed. This is fortunate though as the S parameters add terminations into the system and these terminations can be adjusted to suit our needs. Once the S parameters are obtained for one set of terminations they can be calculated for any other termination by scaling. The voltage divider and the modified voltage divider prepared to calculate the S_{21} transmission coefficient parameter for the same circuit are shown in **Figure A1**.With $Z_{01}=Z_{02}=Z_0$ the current I_1 is given by

$$I_1=\frac{V_{s1}}{Z_0+R_1+\dfrac{R_2Z_0}{R_2+Z_0}} \tag{A1}$$

and I_2 is obtained by applying the current divider rule as,

$$I_2=-I_1\frac{R_2}{R_2+Z_0} \tag{A2}$$

Substitution of (A1) into (A2) gives,

$$I_2=\frac{-V_{s1}R_2}{Z_0(R_2+Z_0)+R_1(R_2+Z_0)+R_2Z_0} \tag{A3}$$

Using the definition of the transmission coefficient from (1) and with $Z_{01}=Z_{02}=Z_0$, which need not be the case, the S_{21} parameter is,

$$S_{21}=\frac{2Z_0R_2}{Z_0(R_2+Z_0)+R_1(R_2+Z_0)+R_2Z_0}. \tag{A4}$$

Several features of (A4) are worth noting. The source voltage is not present. The output current expression is directly present and if the magnitude and phase of the transmission coefficient is found then so is the magnitude and phase of the current. As stated earlier, the terminations need not be equal. That is to say Z_{02} and Z_{03} in the case of a 3 port may be different. Furthermore, Z_{01} at the source might not be the same as Z_0.However, in the cases as explored here for the transformer, the source is set to Z_{01} and is 50 Ω. Now looking back at **Eqn(1)**, the ability to find the ratio of currents as $S_{kn}=-2\sqrt{Z_{01}Z_{02}}\,I_{kn}/V_{sn}$ is possible. Hence, for a 3 port there would be,

$$S_{21}=\frac{-2\sqrt{Z_{01}Z_{02}}}{V_{s1}}I_2 \tag{A5}$$

$$S_{31}=\frac{-2\sqrt{Z_{01}Z_{03}}}{V_{s1}}I_3 \tag{A6}$$

Hence forming the transmission coefficient ratio gives

$$\frac{S_{31}}{S_{21}}=\frac{\sqrt{Z_{03}}}{\sqrt{Z_{02}}}\frac{I_3}{I_2}. \tag{A7}$$

The values of S parameters are measured

(Continued on page 16.)

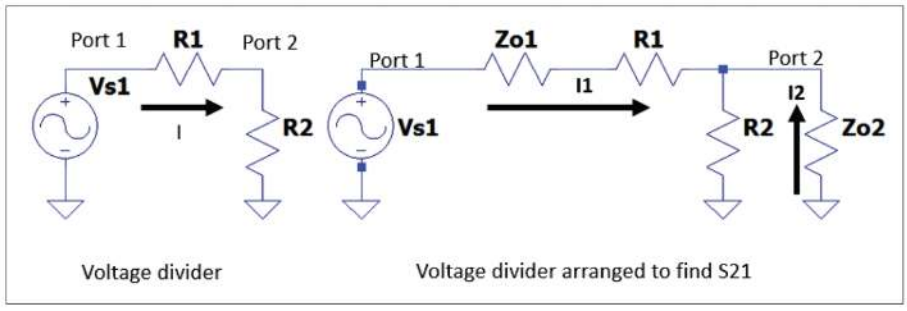


Figure A1 — The basic voltage divider and the voltage divider equipped to find the transmission coefficient, S_{21}. The voltage divider on the left is modified to incorporate the terminations Z_{01} and Z_{02} on the right.

Anthony Le Cren, F4GOH/KF4GOH
7 Rue De l'Avenir, 72220 Téloché, France | **f4goh@orange.fr**

APRS with LoRa TTGO Module

Create a compact APRS module.

APRS (Automatic Packet Reporting System) was developed by Bob Bruninga, WB4APR, as a real-time local communication system for the rapid exchange of digital data, such as geolocation, weather beacons, telemetry, and messages. A classic APRS beacon requires a 144.800 MHz FM transmitter and a TNC (Terminal Node Controller) using FSK (frequency-shift keying) modulation. In recent years there has been a proliferation of modules based on the ESP32 microcontroller. Among these modules, there is the TTGO T-Beam, which combines all the elements to create a compact APRS beacon:

- ESP32;
- LoRa™ SX1278 (a trademark of Semtech) modem delivering 20 dBm;
- GPS;
- Power manager;
- Optional OLED I²C display;
- 3.7 V battery holder.

This results in a very significant reduction in hardware volume compared to a conventional FM device. The user needs only to implement the software in the ESP32 microcontroller and to build the housing. **Figure 1** shows the hardware comparison between a classic FM beacon and LoRa.

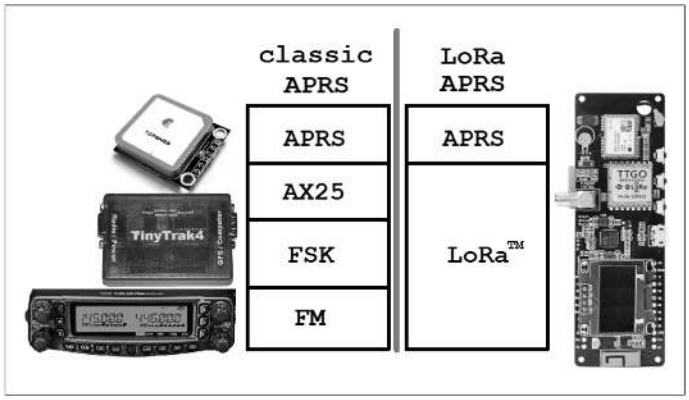


Figure 1 — Hardware comparison between a classic FM beacon and LoRa.

Peter Buchegger, OE5BPA, the initiator of the project, had the idea of using the technology dedicated to IoT (Internet of Things) to transmit APRS frames and thus be able to use existing TTGO modules **[1]**.

An APRS Transmission Link

Figure 2 shows an APRS transmission link. The beacon or mobile station (TTGO T-Beam) periodically sends its geographical position (longitude and latitude). The fixed station (Igate with a TTGO module that does not have a GPS receiver) receives the position and, via its Wi-Fi link, reports it to the APRS servers on the internet.

The user connects to **aprs.fi** with a browser and then consults the position of the mobile station being searched. An external user may not be aware of the technology used (FM or LoRa). However, it is necessary to have enough LoRa Igates to relay the information in the best conditions.

I tested the beacon around my mobile station. The reception was good within a range of 10 to 15 km. Of course the antenna

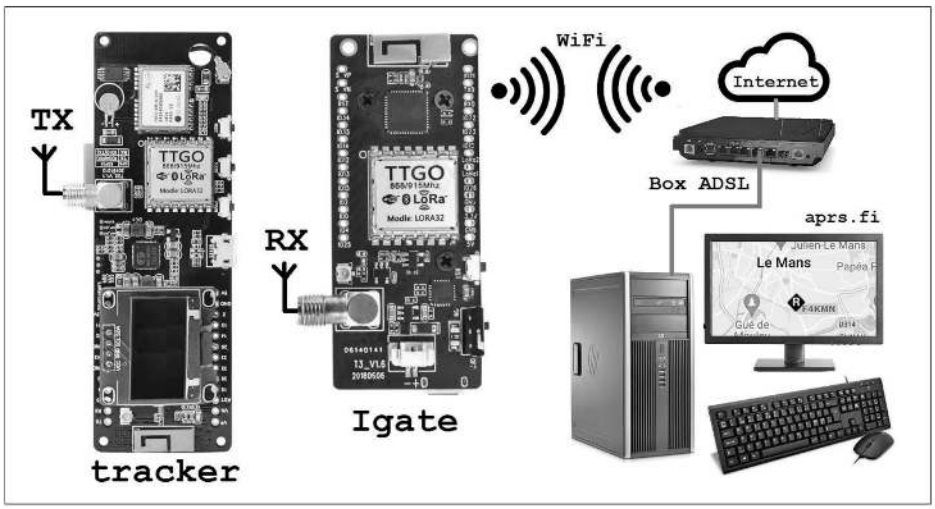


Figure 2 — APRS transmission sequence.

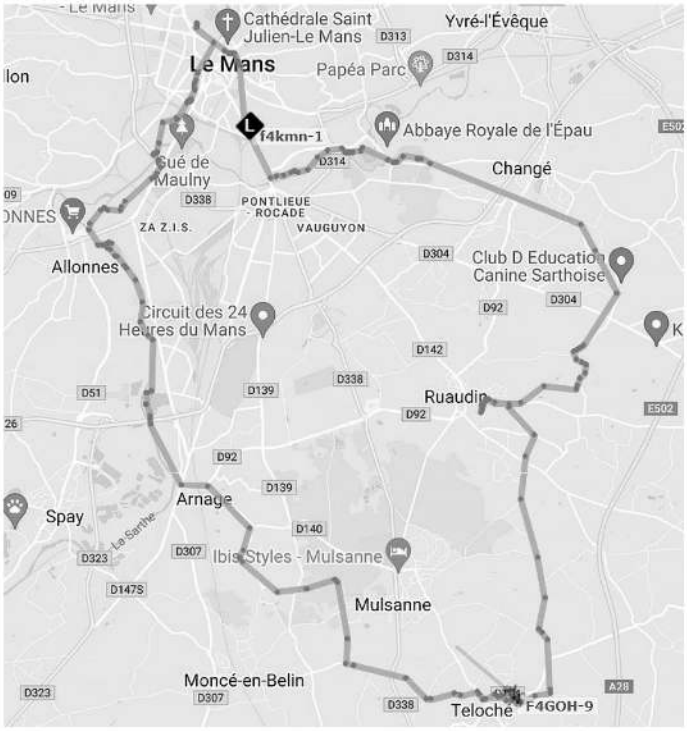


Figure 3 — Testing with an Igate around F4KMN, Radio-Club Touchard Washington High School.

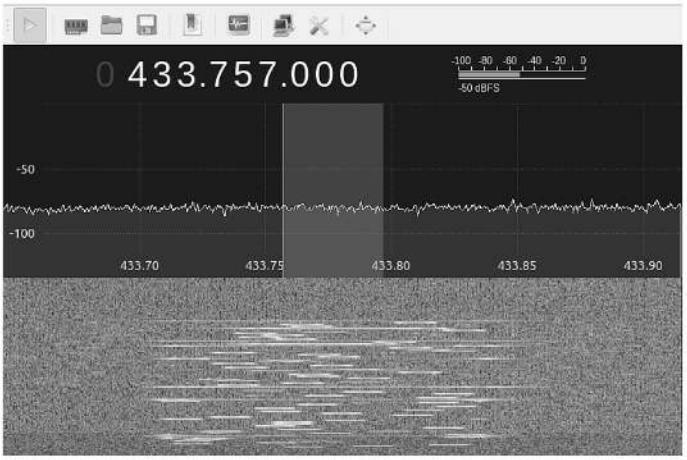


Figure 4 — The LoRa modulation shown in the *GQRX* waterfall.

was not the original one, but a 70 cm J-pole antenna running 20 dBm. **Figure 3** shows the testing around F4KMN, Radio-Club Touchard Washington High School.

LoRa Modulation

LoRa stands for Long Range. It is a technology that allows connected objects to exchange small amounts of data at low speeds. The LoRa radio is based on spread spectrum transmission. **Figure 4** shows the LoRa modulation displayed in the *GQRX* (a Linux software) waterfall. To learn more about LoRa technology, I recommend reading the excellent book written by Sylvain Montagny at the University of Savoie Mont Blanc **[2]**.

Most LoRa modems operate in the industrial, scientific and medical (ISM) radio frequency band at 868 MHz, but there are SX1278 modules using the 430 – 440 MHz UHF amateur band. You should therefore be very careful when purchasing the TTGO to ensure that you have the correct frequency range for the transmitter and receiver.

Igate Software Programming

I advise that you start by programming the TTGO Igate first and then checking whether the internet connection is correct via Wi-Fi. There are two possibilities for this:

1) Use Peter's original program on his Github site. But first you will need to install visual studio code, recompile the program and modify a *JSON* (JavaScript Object Notation) configuration file. This is not an easy thing for anyone **[1].**
2) Use a ready-to-use online programming tool. Indeed, from Peter's source code, I created my own "version" to facilitate software programming **[3]**.

After connecting the module with a USB cable to the PC, check that the USB serial COM drivers are installed. Windows 10 and Linux users will have no particular problems. Note the COM port number that is newly installed.

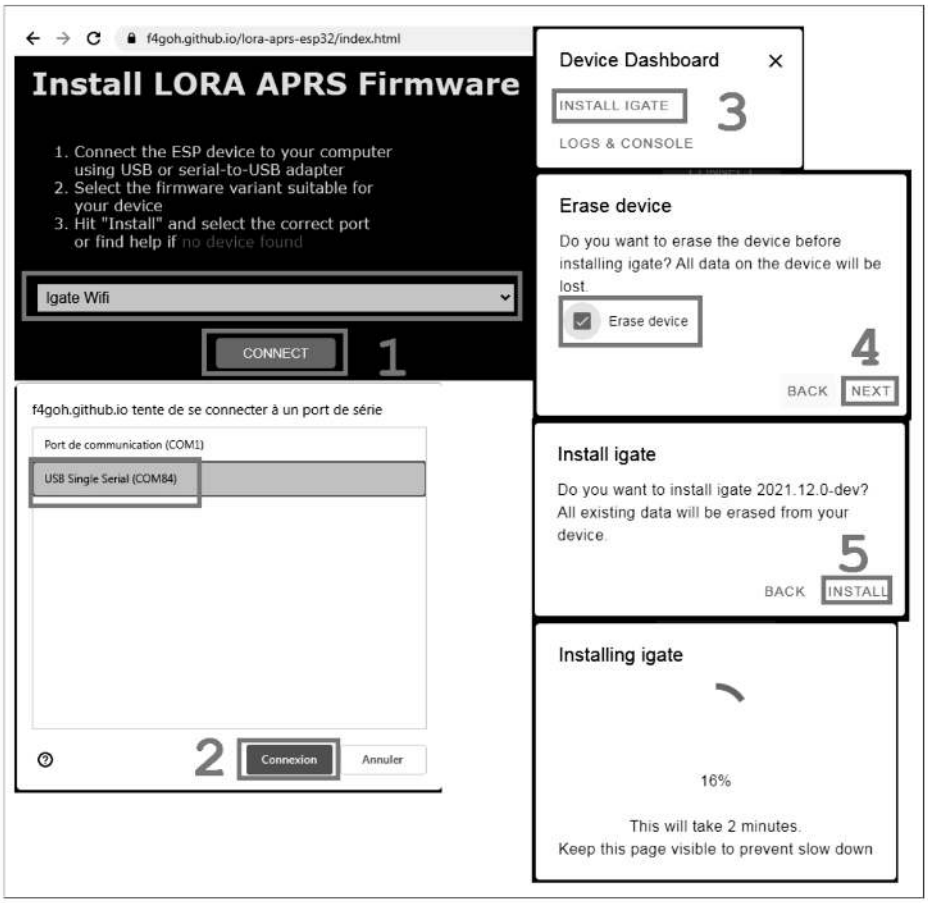


Figure 5 — The different steps to program the Igate software.

I used the Chrome browser to enter the URL **[4]**, select "Igate Wi-Fi" from the drop-down menu. Click on "connect", choose the correct COM port and follow the programming procedure as shown in the screenshots. **Figure 5** shows the different steps to program the Igate software. Once programmed, the OLED screen displays the F4KMN school logo (**Figure 6**). The Igate must now be configured using a serial terminal such as *Putty* at a rate of 115200 bps.

Once the Igate is programmed, there remains the configuration on the command line with a serial terminal. Type the command "help" (**Figure 7**) to get help on the configuration. At a minimum you will have to configure:

- Access point SSID Wi-Fi;
- Password;
- Call sign;
- Geographical position of the Igate (longitude, latitude in degrees minutes decimal);
- Activate the internet connection.

Display the results of the modifications ("show" command). The parameters are saved when exiting the menu by pressing exit. Once configured, the TTGO should connect to the APRS server. An icon indicating the location of the Igate should appear on the **aprs.fi** website.

In case of an error (**Figure 8**), GPIO0 must be connected to GND before programming the TTGO.

Figure 6 — Touchard High School Radio Club home screen.

```
>help
Available commands
Set ssid                                    : ssid mywifi
Set password                                : pass toto
Set new callsign                            : call f4goh-6
Set latitude in degrees minutes decimal     : latitude 4753.41N
Set longitude in degrees minutes decimal    : longitude 00016.61E
Set new comment                             : comment hello
Set frequency                               : freq 433775000
Enable wifi for igate (aprs.fi)             : internet 1
Enable local wifi Access Point for AprsDroid : internet 0
When disable igate, local wifi AP are enable
Set server                                  : server euro.aprs2.net
Set port                                    : port 14580
Set Digipeater enable (0 or 1)              : digi 1
Show configuration                          : show
Reset default configuration                 : raz
Exit menu                                   : exit
>
```

Figure 7 — Typing "help" displays the available commands.

Figure 8 — In case of an error GPIO0 must be connected to GND before programming the TTGO.

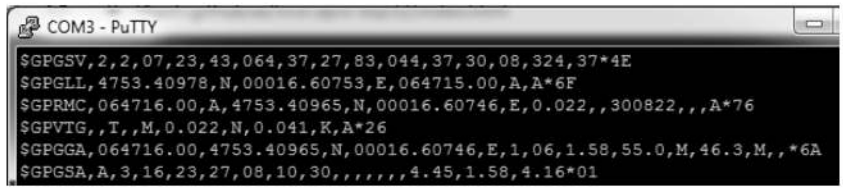


Figure 9 — The Reset GPS program displays NMEA frames on the serial port at 115200 bps.

Programming the Beacon Software

The procedure for programming the beacon is identical to that of the Igate. You will first have to initialize the GPS with the "Reset GPS" program, then reprogram the module with the "Tracker" software. The "Reset GPS" program displays NMEA frames (**Figure 9**) on the serial port at 115200 bps.

Type in the help command to get help with the configuration; a sample configuration is displayed in **Figure 10**. As a minimum, you will need to configure:

- Callsign;
- APRS symbol as an ASCII character;
- Transmission second;
- Comment.

The parameters are saved when exiting the menu by pressing exit.

The interval between two transmissions is fixed at every minute. The "seconds" parameter sets the time when to transmit the frame in the minute. This avoids collisions between several beacons in close proximity to each other. The user can also activate the sending of altitude, heading and speed information. If the measurement of voltage and current consumption is activated, the comment will not be sent.

The last parameter is the dynamic model of the GPS. If the beacon is placed in a weather balloon, it is absolutely necessary to change the model to "airborne1g" in order to guarantee a correct altitude measurement. **Figure 11** shows the dynamic platform model details. **Figure 12**

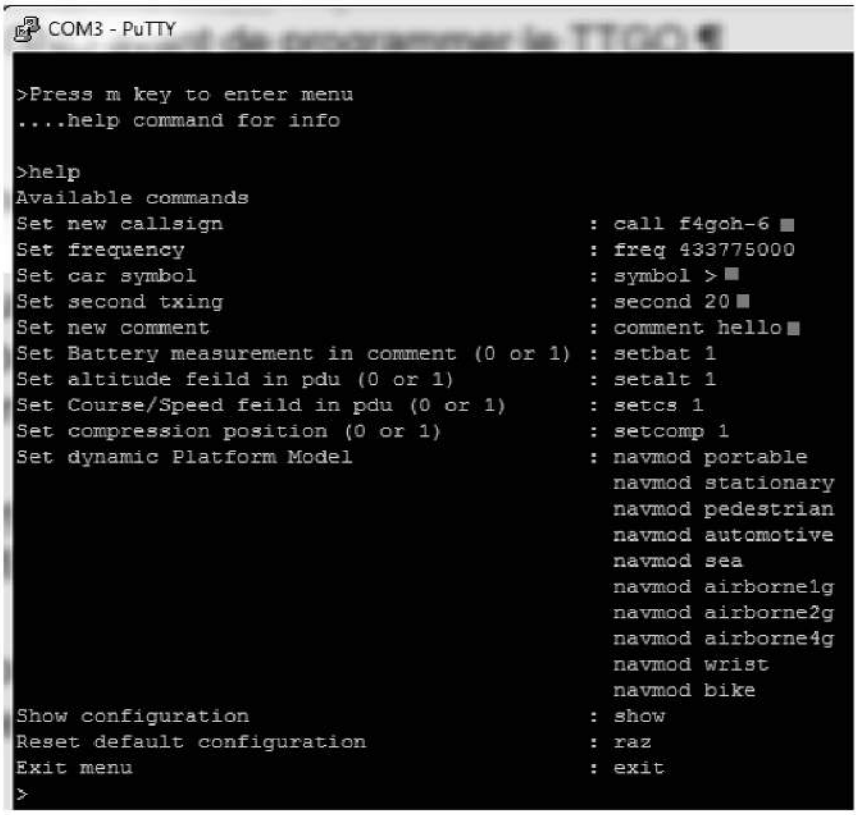


Figure 10 — At least four parameters are needed to configure the tag.

Dynamic Platform Model Details

Platform	Max Altitude [m]	MAX Horizontal Velocity [m/s]	MAX Vertical Velocity [m/s]	Sanity check type	Max Position Deviation
Portable	12000	310	50	Altitude and Velocity	Medium
Stationary	9000	10	6	Altitude and Velocity	Small
Pedestrian	9000	30	20	Altitude and Velocity	Small
Automotive	6000	100	15	Altitude and Velocity	Medium
At sea	500	25	5	Altitude and Velocity	Medium
Airborne <1g	50000	100	100	Altitude	Large
Airborne <2g	50000	250	100	Altitude	Large
Airborne <4g	50000	500	100	Altitude	Large
Wrist	9000	30	20	Altitude and Velocity	Medium
Bike	6000	100	15	Altitude and Velocity	Medium

Figure 11 — Dynamic platform model details.

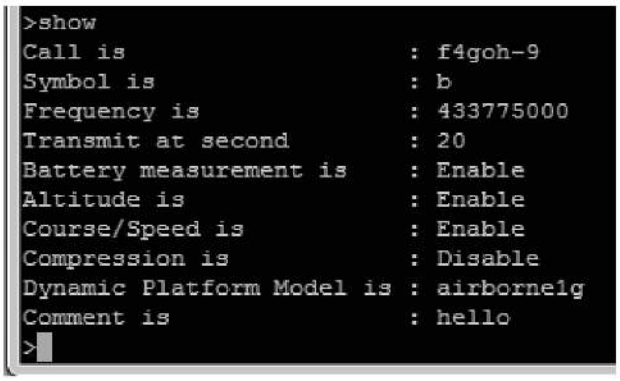


Figure 12 — An example of a weather balloon configuration.

shows an example of a weather balloon configuration.

When the beacon and the Igate are switched on, the display of the frame on the OLED screen uses several display pages in order to facilitate readability. **Figure 13** shows the display of one of the several pages on the OLED screen. Of course, you will have to visit the **aprs.fi** website to check that the position of the beacon is correct.

Using *APRSDroid* on Android

It is possible to use a TTGO module without GPS (same version as for the Igate) as a beacon with a smartphone and the *APRSDroid* software. In this case the internal GPS of the smartphone is used. In the configuration of the Igate module you must disable the internet connection. Enable local Wi-Fi access point for *APRSDroid*: internet 0.

The TTGO module then starts up as a Wi-Fi access point. With the smartphone, connect to this access point (SSID: "APRS Droid", password "totototo"). On the *APRSDroid* side, some changes in the "APRS settings" menu are necessary **[5]**:

- Connection preferences: TNC(plain text TNC2), TCP/IP
- APRS connection TCP/IP
- TCP KISS server to contact: 192.168.4.1:14580

Figure 13 — One frame of the several pages displayed on the OLED screen.

All that remains is to send a position with *APRSDroid*. The TTGO is used as a radio/smartphone gateway. This situation is very convenient for locating beacons in a mobile phone without using a PC with the built-in mapping menu with *APRSDroid*.

Conclusion

There are many projects around the TTGO. This module has a bright future among makers and do-it-yourself fans. The amateur radio side is obviously not forgotten by using the existing APRS protocol. The whole thing allows you to build a beacon very quickly and at a low cost. I rewrote the software primarily for the launch of the weather balloon at Touchard Washington High School in June 2022 **[6]**. But the application can be used in many different fields **[7]**. There will be updates of the software according to future needs. — *73 and good traffic in LoRa APRS.*

Anthony Le Cren, F4GOH/KF4GOH, has been licensed since 2010 and loves to experiment with Arduino applied to the radio. He is Professor of Computer Science at Gabriel Touchard High School, Le Mans, France. Anthony has written numerous articles and maintains a web page ***https://hamprojects.wordpress.com/*** *of amateur radio projects.*

References

[1] **https://github.com/lora-aprs**
[2] **https://www.univ-smb.fr/lorawan/wp-content/uploads/2022/01/Livre-LoRa-LoRaWAN-et-lInternet-des-Objets.pdf**
[3] **https://github.com/f4goh/lora-aprs-esp32**
[4] **https://f4goh.github.io/lora-aprs-esp32/index.html**
[5] **https://github.com/f4goh/lora-aprs-esp32/tree/main/igate**
[6] **https://www.touchard-washington.fr/2022/05/31/le-3-juin-prochain-le-lycee-touchard-washington-prend-de-la-hauteur/**
[7] **https://github.com/PhilippeSimier/Esp32/tree/master/24_Lora**

(Continued from page 11.)

with a variety of terminations not necessarily equal to Z_0. Then the current ratio is provided by (**A7**). For example, a set of 175, 25 Ω terminations are used. Their sum is the correct value for a 1:4 transformer where the input Z_{01} is 50 Ω. Then the resulting current ratio to observe balance is,

$$\frac{I_3}{I_2} = \frac{\frac{S_{31}}{S_{21}}}{2.645} \quad . \qquad \text{(A8)}$$

Acknowledgements

Invaluable inputs from Gary O'Neil, N3GO, John Marshall, KU4AF, and John Swartz, AF4ZE, during the multiple drafts. Much appreciated.

Photos by the author.
Alan Victor, W4AMV, was licensed in 1964. He operates mostly CW using an all homebrew station and enjoys design, construction and restoration of communication and test equipment. Alan worked in both the communication and semiconductor engineering fields. He received his PhD in electrical engineering from North Carolina State University and is currently involved with their mentorship program assisting new graduates in their engineering studies.

References

[1] H. Lee, W. Lee, and Y. Yang, "Design of 1:4 Ultrawideband Hybrid Transmission-Line Balun," in *IEEE Microwave Magazine*, vol. 16, no. 1, pp. 122-126, Feb. 2015, doi: 10.1109/MMM.2014.2367956.
[2] M. Kim, *et al*., "Balanced Unbalanced Wideband: A Wideband, Bifilar Transmission-Line Balun," in *IEEE Microwave Magazine*, vol. 17, no. 1, pp. 65-69, Jan. 2016, doi: 10.1109/MMM.2015.2487921.
[3] H. Oh, *et al*., "Better Balun? Done The Design of a 4:1 Wideband Balun Using a Parallel-Connected Transmission-Line Balun," in *IEEE Microwave Magazine*, vol. 18, no. 1, pp. 85-90, Jan.-Feb. 2017, doi: 10.1109/MMM.2016.2616186.
[4] S. Maas, *The RF and Microwave Circuit Design Cookbook*, Artech House, 1998, pp. 44,45, 102.
[5] F. Terman, *Electronic and Radio Engineering*, McGraw-Hill, 1955, Chapter 23-13.
[6] LTspice Simulator, **https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html**.
[7] MicroCap12 simulator, **www.spectrum-soft.com/index.shtm**.
[8] H. L. Krauss, C. W. Bostian, F. H. Raab, *Solid State Radio Engineering*, John Wiley and Sons, 1980, pp. 200, 371-379.
[9] C. A. Balanis, *Antenna Theory Analysis and Design*, John Wiley and Sons, 1997, Chapter 4.3.
[10] **https://groups.io/g/nanovna-users**.
[11] Measurement of Amplitude and Phase Balance: Center-Tapped Transformers, **https://www.minicircuits.com/appdoc/AN20-001.html**.
[12] S. S. Haykin, *Active Network Theory*, Addison-Wesley, 1970, Chapter 6, pp. 262-266.
[13] **https://www.minicircuits.com/pdfs/ADT4-1WT+.pdf**.
[14] **www.gunthard-kraus.de/Ansoft%20Designer%20SV/English%20Tutorial%20Version/index_english.html**.

Errata

■ In Keith Stammers, GØSXG, "A Graphical Method to Determine the Impedance of a Parallel Resistor and Reactance," *QEX* May/June, p. 22, the first equation should be:

$$Z = \frac{jX_pR_p}{R_p + jX_p} = \frac{jX_pR_p\left(R_p - jX_p\right)}{R_p^2 + X_p^2}.$$

Thanks to Thomas B. Fox, WB2BCD, for spotting the error.

Upcoming Conferences

2023 Central State VHF Society Conference

July 27 – 30, 2023
North Little Rock, AR
2023.csvhfs.org

Central States VHF Society 2023 Conference will be held July 27 – 30, 2023 at the Wyndham Riverfront Little Rock, North Little Rock, Arkansas.

Society of Amateur Radio Astronomers

August 20 – 23, 2023
Green Bank, West Virginia
www.radio-astronomy.org

The 2023 SARA Annual Conference will be held August 20 – 23, at the Green Bank Observatory, Green Bank, West Virginia.

Barry Chambers, G8AGN
b.chambers@sheffield.ac.uk

An Antenna Pointing System Based on the U-Blox C94-M8P-3 Evaluation Kit

This is a precise use of GPS (RTK DGNSS) for calibrating a rotatable protractor mounted on the operator's dish antenna tripod.

Accurate alignment of microwave dishes **[1]** is particularly important when using electrically large dishes or when operating QRP, especially on mm-wave bands such as 122 GHz. While researching dish alignment techniques I found an article in *DUBUS* **[2]** in which David Smith, VK3HZ, and Rex Moncur, VK7MO, discussed the use of a RTK DGNSS (real-time kinematic, differential global navigation satellite system) **[3]** setup based on a u-blox TM C94-M8P evaluation kit to align EME antennas at 10 GHz. The technique resulted in this article.

Background

RTK DGNSS can provide extremely high-quality information about the separation and compass bearing between two GNSS modules, normally referred to as the Base and the Rover. This information can then be used to calibrate a rotatable protractor mounted on the operator's dish antenna tripod (**Figure 1**). It is then a simple matter to point the dish accurately in the direction of a distant station whose beam heading has been determined previously using Google Earth or by calculation from known latitude and longitude coordinates, or Maidenhead locators.

In **Figure 2** the Rover is co-sited with the dish antenna and the Base is located some distance away, typically in the range 2 to 20 m. It is the locations of the Base and Rover GNSS antennas that is important, since their phase centers are the reference points for timing or phase measurements. The electronics of the Base and Rover modules is identical and each contains a high-grade u-blox M8P GNSS receiver and a UHF radio transceiver operating in the 433 MHz band for operations in Europe, and 915 MHz in the US. The radio link is used to send position correction data from the Base to the Rover. The distinction between a Base and a Rover lies solely in the way it has been configured in software.

In our application, the locations of the Base and Rover are interchanged from those

Figure 1 — Rotatable protractor on dish tripod.

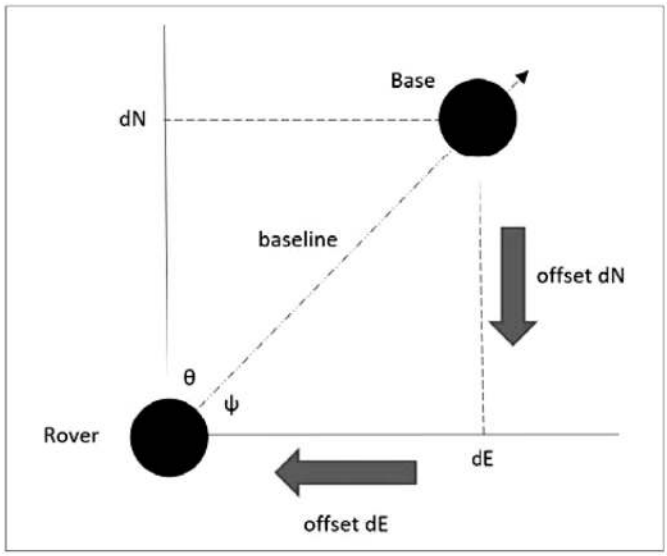


Figure 2 — Configuration of Base and Rover locations, which are at the same height. The arrows show displacements from the Base.

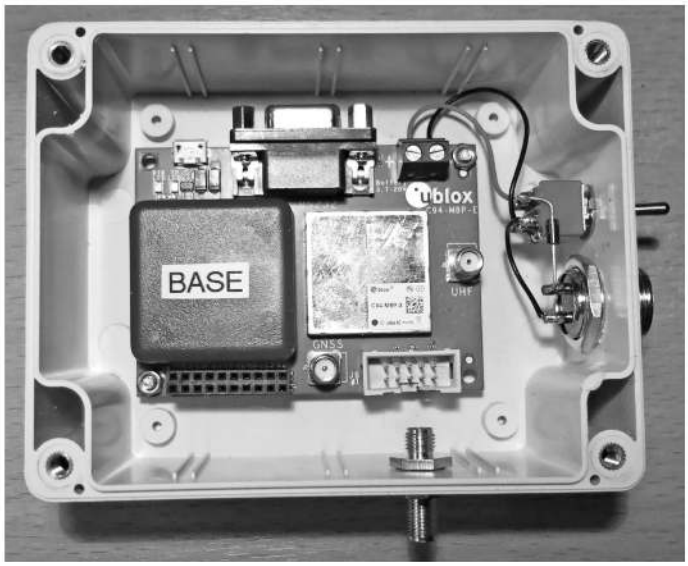


Figure 3 — The C94-M8P evaluation kit includes two of these modules.

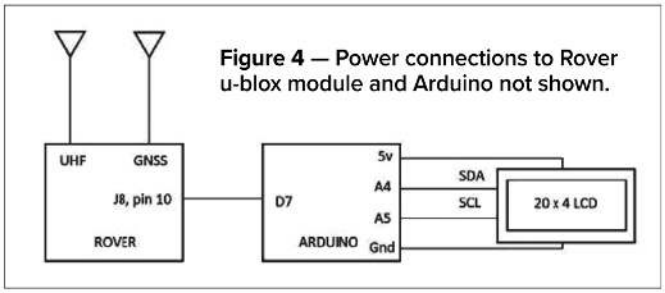


Figure 4 — Power connections to Rover u-blox module and Arduino not shown.

Figure 5 — Rover LCD display.

of a normal RTK positioning system and, in theory at least, both the Base and Rover are not fixed and so the separation and angle between them can vary with time; this is known as the "moving baseline" mode. This is done for two reasons. First, the Rover needs to be at the location of our dish antenna since this is where we will need to display to the operator the information about the angle between the Base and Rover. Second, it is only when the system is configured to be in moving baseline mode that we have information from the Rover about where it is in relation to the Base. This information is expressed as the two quantities *dN* and *dE* in Cartesian coordinates (**Figure 2**).

Angle θ, given by

$$\theta = 90 - \psi = 90 - \text{atan}\left(-dN/-dE\right),$$

is the orientation of the Rover-Base baseline with respect to true north. We are interested only in the relative location of the Rover with respect to that of the Base. Hence the Base location to an accuracy of about a meter will be adequate. The system should then be able to determine the length of the baseline to within a few mm and the baseline orientation to within about 0.1°. Once the latter has been determined, the dish antenna, with its associated sighting scope, is then rotated to face the Base and the protractor rotated independently so its pointer indicates the value θ. The protractor is then locked into position on the tripod base. This completes the protractor calibration and now any dish pointing angle can be set accurately by simply rotating the dish so that the pointer indicates the desired angle.

Hardware Requirements

Figure 3 shows a C94-M8P module. When used as a Base, there are only four connections to the outside world. Two of these are via SMA sockets into which are plugged the GNSS and UHF radio antennas. These are supplied as a part of the evaluation kit. The other two connections, J3 on the module upper right, are for the power supply. The hardware can operate on any voltage between 3.7 and 20 V. I use a small 12 V gel battery since the current required is modest.

When used as a Rover, as well as the above connections, an additional one is required to carry data from the Rover, via pin 10 of J8 (on the lower left of the module), to a serial interface input on an Arduino Nano or Uno as shown in **Figure 4**. This serial data is transmitted at 19200 bps. The Arduino is used to parse the message data and to display the results on a 20 by 4 LCD display via a I2C interface.

Note that the data passed from the Rover to the Arduino is not in the form of NEMA messages, but in a proprietary u-blox message format called UBX.

Display and Arduino

I opted to use a 20 by 4 I2C LCD display and an the Arduino Nano. A typical LCD screen of data is shown in **Figure 5**.

The top line of the display shows the Maidenhead locator and the standard deviation of one minute's values of the derived Rover to Base azimuth angle θ. The second line shows the current value of the azimuth angle in degrees. The third line shows the current derived value of the Rover to Base baseline length in m. The fourth line displays information about the Rover system status. FIX (in upper case) shows that the Rover has a valid GNSS fix. DIF (in upper case) shows that differential position corrections are being applied, REL (in upper case) shows that the relative position components (*dN* and *dE*) are valid and CAR (in upper case) shows that the GNSS receiver is operating using a carrier phase range solution with fixed ambiguities. The number (55) indicates that all the status flags described above are "true." When the Rover is first powered up, this number starts at 0 and increases in steps up to 55 as the various status flags become true.

A copy of my Arduino sketch can be obtained on request.

u-blox Board Configuration

The two u-blox modules included in the C94-M8P kit are identical but need to be configured differently, so it is wise to label them as Base and Rover before proceeding further. A block diagram of the modules is shown in **Figure 6**.

Configuration is carried out by connecting a u-blox module to a PC via the USB

interface J2 (on the upper left of the module shown in **Figure 3**). The PC should be running the u-blox configuration software "u-center", which can be downloaded from the u-box URL **https://www.u-blox.com/en/product/c94-m8p**.

Once a u-blox module is connected to u-center, it will act as a conventional GNSS receiver and transmit the usual NEMA messages. These can be viewed using the u-center "View" pull-down menu options. This data can also be viewed as lists/plots of visible satellites, position data, etc. This step confirms that the u-blox module is working in its factory-default configuration. The module can be restored to the factory-default condition by pressing the "restore" button.

As seen in **Figure 6** the actual u-blox M8P GNSS receiver is connected to the UHF radio transceiver and to the outside world via a UART. This will be referred to in the configuration documentation as UART1. This is also accessible via the multi-pin connector J8. Pin 10 of this connector provides us with a means of extracting 19200 bps-rate serial UBX messages from the Rover GNSS receiver, which are subsequently processed by the Arduino. This facility is not used in a conventional RTK application and so will require additional configuration steps to be taken in the Rover procedure which is described below. These are not discussed in detail in either the u-blox documentation or in the *DUBUS* article **[2]**.

At this stage download the C94-M8P Setup Guide **[4,5]** and the C94-M8P User Guide **[6]** from the u-blox URL. Section 4.3, Moving Baseline RTK Configuration, of the user guide is the relevant part. It may be helpful to view the helpful YouTube video **[7]**. The video shows the setup process for both the Base and Rover boards, as described pictorially in the quick Setup Guide from the u-blox URL. The steps shown in the Setup Guide do not quite correspond to those we require, but this is clarified in the paragraphs below.

Base Configuration

The Base should be configured as discussed in the Quick Setup Guide **[5]** but with the following modifications. In the u-center tool bar select "View", then "Messages View". Place your mouse over NEMA and right-click. Then click on "Disable Child Messages". This will stop the GNSS receiver from outputting NEMA messages.

Now go to Slide 7 in the quick setup guide. This step is not needed so just check that the "Mode" box is set to "Disabled" (it should be, as this is the factory-set default).

Turn now to Slide 9 and fill in the boxes as shown. Then press the "Send" button to send these commands to the GNSS receiver in the module.

Fill in the boxes as shown in Slide 10 but replace the F5-05 RTCM3.3 1005 message with F5-05 RTCM3.3 4072.0 (the guide mentions messages as RTCM3.2 rather than 3.3 but this is okay). Note that each time you fill in the CFG_MSG window with new data in the boxes, you must press "Send" before moving onto the next data set.

This completes the Base configuration procedure but before disconnecting the module from the PC, go to UBX-CFG-CFG and "save current configuration".

The procedure outlined above is also explained in more detail in Section 4.3 in **[6]** User Guide.

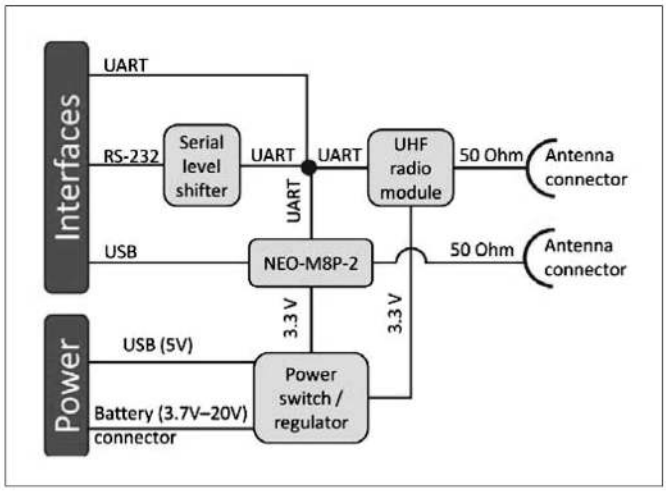
Rover Configuration

Repeat step 1 of the Base configuration for the Rover module (turn off NEMA messages). Move the mouse down to UBX and right click. Then click on "Enable Child Messages". This will command the GNSS receiver to only output u-blox proprietary UBX messages. A list of these may be seen by clicking on the + sign to the left of UBX.

Turn to Slide 12 in the quick setup guide and fill in the boxes as shown, but for "Protocol out" select UBX. Press "Send".

Select UBX-CFG-MSG. In the message dropdown list find "01-3C NAV-RELPOSNED." Tick the box next to UART1 and change the 1 in the adjacent box to 5. This commands the Rover GNSS receiver to output a RELPOSNED message via UART1 (and pin 10 on J8) every 5 seconds. It is this information which the Arduino will process to provide the Rover-Base offsets and hence the baseline length and azimuth angle. Press "Send".

Select UBX-CFG-MSG. In the message dropdown box find "01-02 NAV-POSLLH". Tick the box next to UART1 and change the 1 in the adjacent box to 10. This commands the Rover GNSS receiver to output a POSLLH message via UART1 (and pin 10 on J8) every 10 seconds. It is this information which the Arduino will process to provide



▲**Figure 6** — Block diagram of the C94-M8P Base and Rover modules.

▶**Figure 7** — 10 m baseline configuration.

the Rover Maidenhead locator. Press "Send".

Finally select UBX-CFG-CFG and "save current configuration". It is now safe to power down the Rover module and the configuration process for both u-blox modules is now complete.

The interested reader will find detailed information about the UBX NAV-RELPOSNED and UBX NAV-POSLLH message formats in **[8]**.

Using the C94-M8P System in the Field

To achieve optimum results, both GNSS antennas need to be mounted on ground planes. The ones supplied are just about adequate but ideally the Base antenna should be mounted on a larger one. Both antennas should have a clear view of the sky so as to access as many GPS and GLONASS satellites as possible.

Initial testing of the complete system was carried out "in the field" over nominal baseline lengths of 2 m, 10 m and 20 m and with both GNSS antennas having a very clear view of the sky down to the horizon. The baseline lengths were measured out over fairly level ground using a 30m tape and the tripods supporting the Base and Rover GNSS antennas positioned accordingly; **Figure 7** shows the 10-m baseline test configuration. The Rover electronics were housed in my car as shown in **Figure 8**. The Rover GNSS antenna is on the tripod nearest to the car and the baseline length could be changed by moving the Base GNSS antenna nearer or further away. The baseline orientation (Rover to Base) was pointing roughly East.

Each test was carried out over a 20-minute period after the system was first locked up and showing a status value of 55, indicating that a carrier phase range solution with fixed ambiguities had been obtained. Every minute the following data were recorded: time, status flag number (always 55), indicated azimuth angle in degrees, indicated baseline length in m and the one-minute standard deviation of the indicated azimuth angle values.

Figure 9 shows the derived baseline azimuth angle variation with time for the three baseline lengths. The actual mean value of the derived baseline azimuth angle for the different baseline lengths is of no interest here since in moving the Base GNSS antenna tripod when altering the nominal baseline length between tests, some variation in tripod positioning transverse to the baseline direction was inevitable; this effect will become less important as the length of the baseline is increased.

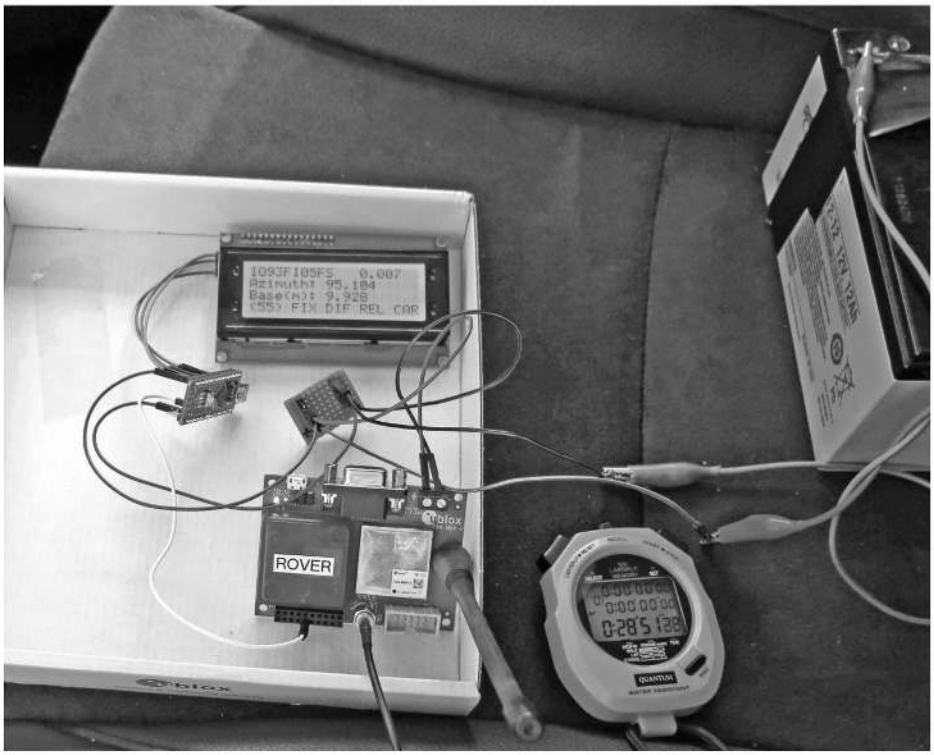


Figure 8 — Rover electronics package in the car (not yet properly boxed up).

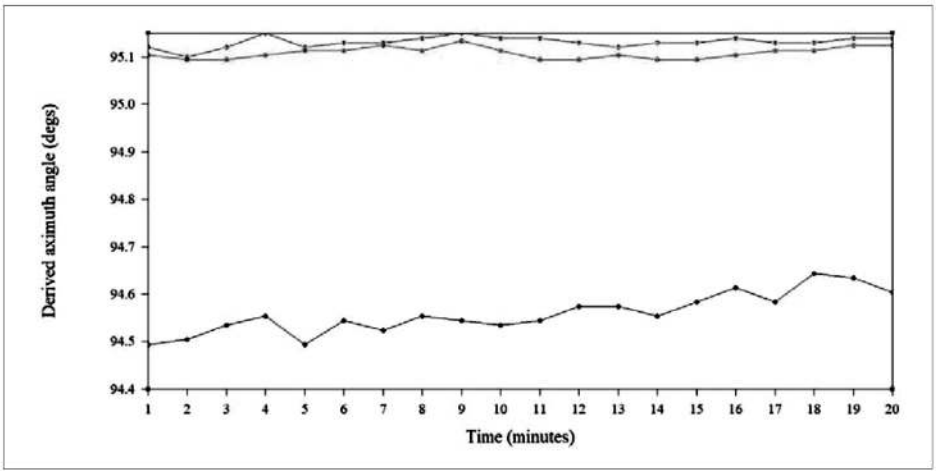


Figure 9 — Variation of derived baseline azimuth angle with time; Bottom (black) curve = 2 m, Mid (red) curve = 10 m, Top (blue) curve = 20 m.

What is of interest is the peak-peak variation of the azimuth angle with time. As might be expected, this becomes larger in the case of the shortest baseline, but even then, the variation is less than ±0.1° and for the 20 m baseline the variation is approximately ±0.03. Looking at the azimuth graph for the 2 m baseline, the azimuth angle seems to increase slightly with time, unlike the 10 and 20 m cases. It is suggested that this might be due to the tripod holding the Base antenna tilting slightly. The ground where the Base tripod was situated for the 2 m test was uneven with long "tufty" grass. I did not "dig-in" the tripod feet and as the tripod had almost no weight on it, it is possible that over 20 minutes the grass "relaxed" and tilted the tripod very slightly in a direction transverse to the baseline. Only a small amount of tilt would be required to see the slight change in azimuth angle and the effect would be most evident for a 2 m baseline. When positioning the

Base tripod for the 10 and 20 m tests, the ground was flatter with shorter grass and so the tripod was unlikely to move over time.

Although the data is not shown here, the derived baseline lengths (2.098, 9.938 and 19.890 m, respectively) showed no variation over the 20-minute test periods. When considering the 1-minute azimuth standard deviation results, as might be expected, the largest values were found for the shortest baseline, but even then, the majority of the values were around 0.02 with the occasional value reaching 0.033, which is very acceptable.

A final comment to make about these results is that the complete set of tests took place over a period of almost two hours and so the GPS and GLONASS constellations as seen by the GNSS antennas would have changed considerably in that time; nevertheless, this appears to have had minimal effects on the observed results.

The next phase of the testing program involved using a short rotatable baseline, as shown in **Figure 10**. It was made from a length of timber 1.6 m long, with a central pivot point and large steel washers were placed symmetrically at 25 cm intervals. The latter were used to position the GNSS antennas and their associated circular ground planes. Initial tests were made with baseline lengths of 1.5 m and 50 cm. The test site was my back garden, which is not ideal since it is only about 20 m long and bounded by the house at one end (roof line maybe 10 m) and trees at the other end (up to at least 15 m). Nevertheless, I was able to get a Rover 55 status after about 15 minutes and then it was good for about 95% of the test duration.

With this arrangement of rotating baseline, both the Base and Rover positions move as the timber spar is rotated. The new angle and baseline length were shown at the next display update but did take a few seconds to settle down. During and after a rotation the Rover status remained at 55 but as expected, the standard deviation value initially became very high (several 10s if the baseline angle of rotation was large) before settling down over several minutes . The indicated baseline angle immediately after a rotation was within a degree or so of its final value and the indicated baseline was within 1 cm of its nominal value. I tried to keep both GNSS antenna orientations the same with respect to the baseline axis so as to keep the antenna phase centre positions in roughly the same place.

By this stage in the testing program, both

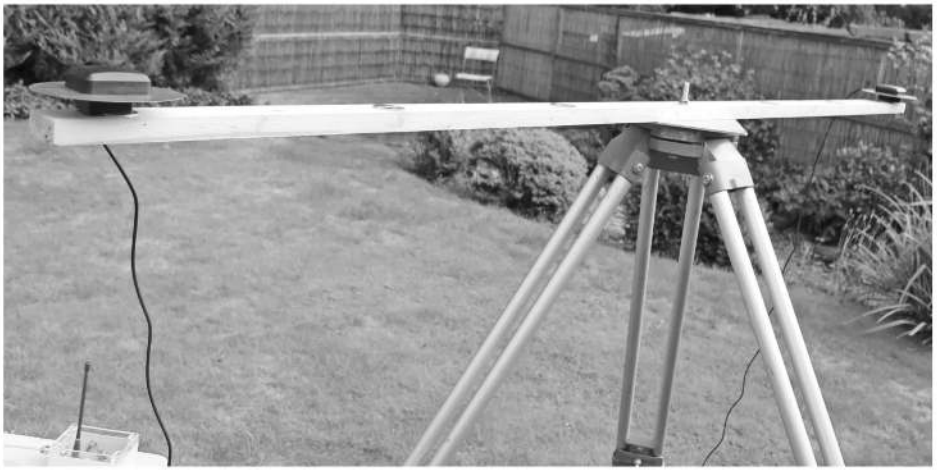
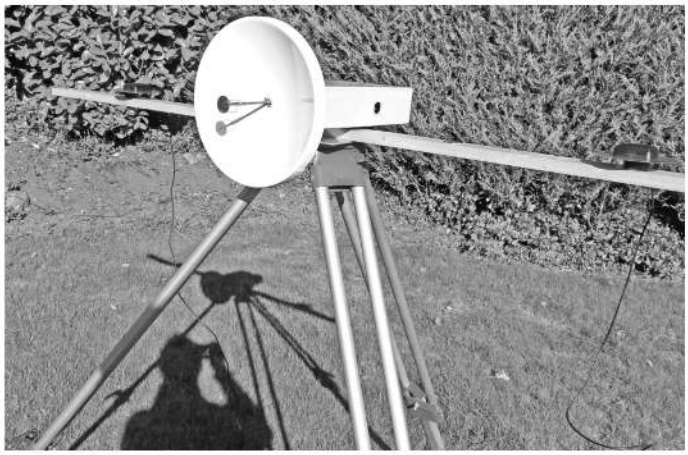


Figure 10 — Prototype of rotatable baseline.

Figure 11 — Rover module and LCD display.

Figure 12 — Base module.



▲**Figure 13** — Rotating baseline parallel to dish antenna aperture.

▶**Figure 14** — One possible geometry for a GNSS pointer.

Rover and Base modules had been boxed up properly, as shown in **Figures 11** and **12**. The next test was made using the rotating baseline to see how accurate the GNSS indicated baseline heading was in comparison to optical sighting using a mobile phone base station antenna mast as the target. The latter was approximately 10 km away from the turntable. Google Earth was used to estimate the separation between the turntable baseline and the antenna mast and the latter's orientation. The baseline was then rotated to face the distant mast and aligned by sighting down the line joining the two GNSS antennas. This introduces a small angular error as it is not possible to know exactly where the GNSS antenna phase centers are within their plastic enclosures. Nevertheless, it was found that for a baseline length of 1.5 m, the GNSS indicated baseline heading was within 0.5° of that obtained using Google Earth. Using a baseline length of 0.5 m, the error slightly larger but less than 1°.

In practice, the rotating baseline approach can be used in two ways. Firstly, the baseline can be established across the dish antenna aperture. Then the antenna heading is that of the baseline ±90°. The choice of sign depends on whether the Base or the Rover GNSS antenna is located on the right-hand side of the microwave rig. This arrangement is shown in **Figure 13**.

Alternatively, the baseline can be established along the dish antenna bore sight direction, in which case the "GNSS pointer" gives the dish heading directly. One possible implementation of this case is shown in **Figure 14**, although others are possible. In **Figure 14**, the Base GNSS antenna is the one closest to the rig. If the rig were mounted behind the GNSS pointer, then the Base GNSS antenna would be the one furthest away from the rig.

Conclusions

From the work which has been carried out to date, the concept of a GNSS pointer seems to offer the potential for high accuracy dish antenna alignment when operating portable. The equipment is simple to set up and operate and can be situated close to large metallic objects such as cars. It does require, however, a fairly unobstructed view of the sky if a rapid RTK carrier phase range solution with fixed ambiguities is to be obtained.

Since the rotating baseline concept discussed above uses short baseline lengths, it might be possible to dispense with the radio link between the Base and Rover units and to send data via their hard-wired serial interfaces. Such a scheme could lead to the development of simpler and less expensive Base and Rover units but the development of these is left to others.

Finally, it should be possible to make the baseline arrangement shown in **Figure 14** tiltable in elevation. Then use could be made of the *dD* component for the difference in height between the Rover and Base modules to calculate the dish elevation angle and to display it on the LCD.

I am grateful to David Smith, VK3HZ, for his useful advice and suggestions concerning further testing while this project was being undertaken.

Barry Chambers, G8AGN, is a retired Professor of Electromagnetic Engineering at the University of Sheffield in the UK where he headed the Antennas Research Group. His research interests included microwave antennas and propagation, free-space microwave measurements and smart microwave materials and structures. He was elected Fellow of the IEEE for his pioneering contributions in microwaves. Barry was licensed in 1965 and made his first microwave contact in 1968. He has worked on all the microwave bands from 1.3 GHz to 134 GHz as well as 30 THz, IR, visible and UV nanowave bands. Over the years, Barry has served as a member of the RSGB Microwave and Propagation Studies Committees and the UK Microwave Group committee. He has written articles for DUBUS, *the RSGB* RadCom *Magazine and* QEX *and is a frequent contributor to the UK Microwave Group newsletter* Scatterpoint.

References

[1] **https://sites.google.com/a/bsaontarget.org/ontarget/mirrors?overridemobile=true**

[2] D. Smith, VK3HZ, and R. Moncur, VK7MO, "Differential GPS Azimuth Reference for Microwave Portable Operations," *DUBUS* 3/2019, pp. 9 – 15.

[3] How RTK works - YouTube; **https://www.youtube.com/watch?app=desktop&v=ieearzWTCZw**.

[4] **www.u-blox.com/en/product/u-center**

[5] **https://content.u-blox.com/sites/default/files/C94-M8P-Appboard-Setup_QuickStart_(UBX-16009722).pdf**

[6] **https://content.u-blox.com/sites/default/files/C94-M8P-AppBoard_User-Guide_(UBX-15031066).pdf**

[7] Neo-M8P RTK setup and demo; **https://www.youtube.com/watch?v=n8PUyOtiGKo**.

[8] **https://www.u-blox.com/docs/UBX-13003221**

Hieronim Lecybyl, M7HBL
Oriel College, Oriel Square, Oxford, OX14EW, UK
hieronim.lecybyl@oriel.ox.ac.uk

Remigiusz Lecybyl, MØLRH
M0LRH@lecybyl.com

30 THz Experiment Over 100 m Distance

Ham radio expands into the 30 THz band with these experiments.

Ham radio exploration of the 30 THz band has started recently. In November 2020, Andrew J. Anderson, VK3CV/WQ1S, and Karl Harbeck, VK3LN, made contact in the 30 THz band. A two-way contact over 60 meters reported 5 by 5 signals received each way. This is believed to be the first ever communication in a previously unexplored 30 THz band **[1]**. However, currently there are significant technical limitations that make difficult the generation and detection of signals at 30 THz. The band is in its infancy, and its present state and development may be compared to that of shortwave radio over 100 years ago. Despite the technical difficulties, the 30 THz band creates enormous potential for amateur experimentation.

This work aims to popularize the 30 THz band by experimental confirmation that ham radio communication in the 30 THz band is possible at a distance of over 100 meters. This article consists of five parts. In Part I, experimental aspects of the 30 THz band are reviewed. This is followed by precisely describing the transmitter (Part II) and receiver (Part III). In Part IV, details of the outdoor experiment are provided. Off-line data analysis is described in Part V. The article finishes (Part VI) by discussing the rationale that leads us to believe that a distance over 10 km is likely achievable in the 30 THz band for ham radio amateurs.

The following innovations for the 30 THz ham-radio band were used in the presented design:

- Phase shift modulation
- Low power (5 – 10 W) transmission (previous experiments in the 30 THz band use power exceeding 200 W);
- Locked-in signal detection using a 1 pps GPS signal as a reference;
- Dual use of the same astronomical telescope as a "dish" for ham radio and as a scope for alignment;
- Video camera for receiver alignment;
- Jupyter notebook for offline analysis of 30 THz signal.

Part I: 30 THz Band

In the UK the 30 THz band is not regulated, and experiments do not require any permission or license. Nevertheless, the authors contacted OFCOM (UK's government-approved regulator for telecommunication service). OFCOM representative confirmed that in the UK, frequencies above 3 THz are not within the scope of the Wireless Telegraphy Act 2006, and a license under that Act is therefore not needed to use equipment that operates only on those frequencies.

Two excellent articles were published previously about the 30 THz band in *QEX* by Andrew VK3CV, **[2]** and Barry G8AGN, **[3]**. Barry G8AGN, also set up a 30 THz discussion forum, providing an excellent information source **[4]**. The 30 THz band theoretical aspects were reviewed in the online resources above, so only the practical aspects will be discussed here. The authors of this article benefited from attending the 30 THz presentation and equipment demonstration by Chris, GØFDZ, at Cray Valley Radio Society **[5]**.

The 30 THz band is located between the infrared and terahertz regions in the electromagnetic spectrum; its primary applications include the non-contact measurement of temperature, thermal cameras, astronomy, CO_2 lasers, and LWIR spectroscopy. Exploring the current applications of 30 THz provides information about materials' relevant properties and directions on where to buy components appropriate for this band. 30 THz ham radio transmitters currently use a very large bandwidth in the range of a few THz due to the use of black body emitters. According to Planck's Law, any object with a temperature above 0 Kelvin emits wideband electromagnetic radiation with a peak frequency determined by its temperature. For 30 THz, this temperature should be around 200 °C (392 °F). The simplest emitter is a ceramic resistive heater painted with black ink and heated to 200 °C. CO_2 lasers produce 30 THz (10.6 μm wavelength) narrowband radiation. CO_2 lasers must not be used for ham radio communication due to extensive health, fire, and aviation-related risks. CO_2 lasers are Class 4 with emitted power from twenty to a few thousand watts; the laser beam is invisible — any attempt to use such lasers outdoors without special precautions and permissions may be a criminal offence. Modulation is restricted to basic on-off mechanical modulators, which technically could achieve a modulation frequency in the range of kilohertz, but unfortunately, affordable detectors, due to their time constant, can not operate above 100 Hz (usually below 10 Hz).

The most common detectors for 30 THz radiation are pyroelectric crystals and thermopiles. Other available detectors include the Golay cell, bolometers and microbolometers, but there is no evidence that they have ever been used in ham radio experiments. One of the difficulties of building equipment for the 30 THz band is that most materials display unusual properties in this band. Glass, which is transparent for optical and microwave bands, is opaque around 30 THz. In contrast, germanium, which is opaque for optical and microwave bands, in 30 THz is transparent. Increasing the gain and narrowing beam of the transmitter and receiver can be achieved by the use of parabolic mirrors and lenses. Mirrors must either be wholly metallic, or the front of the mirror must consist of a metallic layer (first-surface, front surface mirror). Astronomical telescope mirrors are often covered by a thin protection layer, which may affect their work in 30 THz. The usual glass mirror works poorly as the front surface is glass, which absorbs 30 THz radiation. Though possibly an alternative to mirrors, lenses tend use rather exotic materials (still available for amateurs), materials such as germanium, zinc selenide, gallium arsenide and hard polyethylene (HDPE). HDPE Fresnel lenses are cheap (a 20 mm diameter lens costs below US$1) and safe for use. Other lens materials are rather expensive ($10 – $50+ for a 20 mm lens), and there is a risk of toxicity. It is important to recognize the difference between HDPE Fresnel lenses and common visible light Fresnel lenses, which are used as magnification lenses for reading, DIY, etc. HDPE Fresnel lenses are milky-matte for visible light, whereas common Fresnel lenses are glass clear (made from PMMA, which unfortunately is opaque for 30 THz).

Figure 1 — Heater assembly with paddle mechanical modulator.

Part II: Transmitter

The transmitter consists of a heater assembly, mechanical modulator with the driver, GPS module and Arduino microcontroller. A 30 THz electromagnetic signal is produced by a black body-type radiator made from a metal ceramic heater (MCH, 12 V, 580 C, 12 by 18 mm). The heater is covered by black china ink and installed inside the mirror from a GU10 halogen spot lamp . When powered to between 5 and 7 V (power: 5 – 10 W), it heats to between 200 and 300 °C, which is optimal for 30 THz. Halogen GU10 spot lamp mirror proved to be not very effective, likely because it is covered by protection layer of SiO_2 and practically acted as decoration. The heater mirror is a potential area for significant improvement to this design. The mechanical modulator is made from a 2.5-inch laptop hard drive (HDD) with a metal paddle attached to the arm of the hard drive (**Figure 1**). The HDD actuator is controlled by powering the actuator coil with ±5 V. The whole transmitter is controlled by an Arduino microcontroller (**Figure 2**), which generates a 2 Hz carrier signal synchronized by a 1 pps signal from GPS. The signal is phase modulated with QRSS-6 Morse code. A dot is a 6-second signal with a 180° phase shift compared to a GPS 1 pps signal; a dash is an 18-second signal with 180° phase shift against the GPS 1 pps signal; spaces are coded as with no phase shift against the GPS 1 pps signal. Phase shift modulation and phase-locked detection allow for decoding weak signals with amplitude much smaller than noise — similar techniques are used in PSK31 and FT8. A simple message containing only two Morse letters, "CQ" is continuously transmitted. The 2 Hz carrier with phase shift modulation is generated by an Arduino-controlled mechanical modulator (HDD actuator with paddle) via a half bridge drive SN754410. As mentioned above, the 2 Hz carrier signal is synchronized with the 1pps signal from the GPS module. Usually, a GPS module is used to achieve high-frequency accuracy, but this is not the case in this design. Here the almost perfect phase synchronization of the 1 pps signal of any two GPS modules is used. There is an identical GPS module in the receiver that produces the 1 pps signal almost perfectly synchronized with the transmitter used for phase-locked detection. The GPS module consists of NEO-6M GPS modules, an OLED display and a second Arduino module running a basic example program from Arduino GPS library. The OLED display shows exact location, time and basic GPS diagnostic data. There are two functions of the GPS module in the transmitter: produce a 1 pps signal for synchronization and show the exact location of the transmitter. The transmitter is powered by two type 18650 3.7 V Li-ion batteries. The transmitter is mounted on a tripod (**Figure 3**).

Part III: Receiver

The receiver consists of an astronomical telescope, half-blinded PIR sensor, basic analogue filter, 24-bit ADC, GPS module, RTC module, SD card module and Arduino. The Celestron FirstScope is an affordable, very basic, entry-level telescope with surprisingly good image quality. The telescope focuses a 30 THz signal on the PIR sensor to achieve a high gain, likely above 30 dB. A basic Passive Infrared Sensor/PIR (NICERA RE03R129) is used. For movement detection, such PIR sensors consist of two pyroelectric crystals and a FET. To use them as 30 THz detectors, one of the crystals simply needs to be blinded (**Figure 4**). The PIR sensor is mounted in a 3D printed "telescope eyepiece" in such a way that the unobscured half of the sensor is in the center. The signal from the half-blinded PIR sensor is fil-

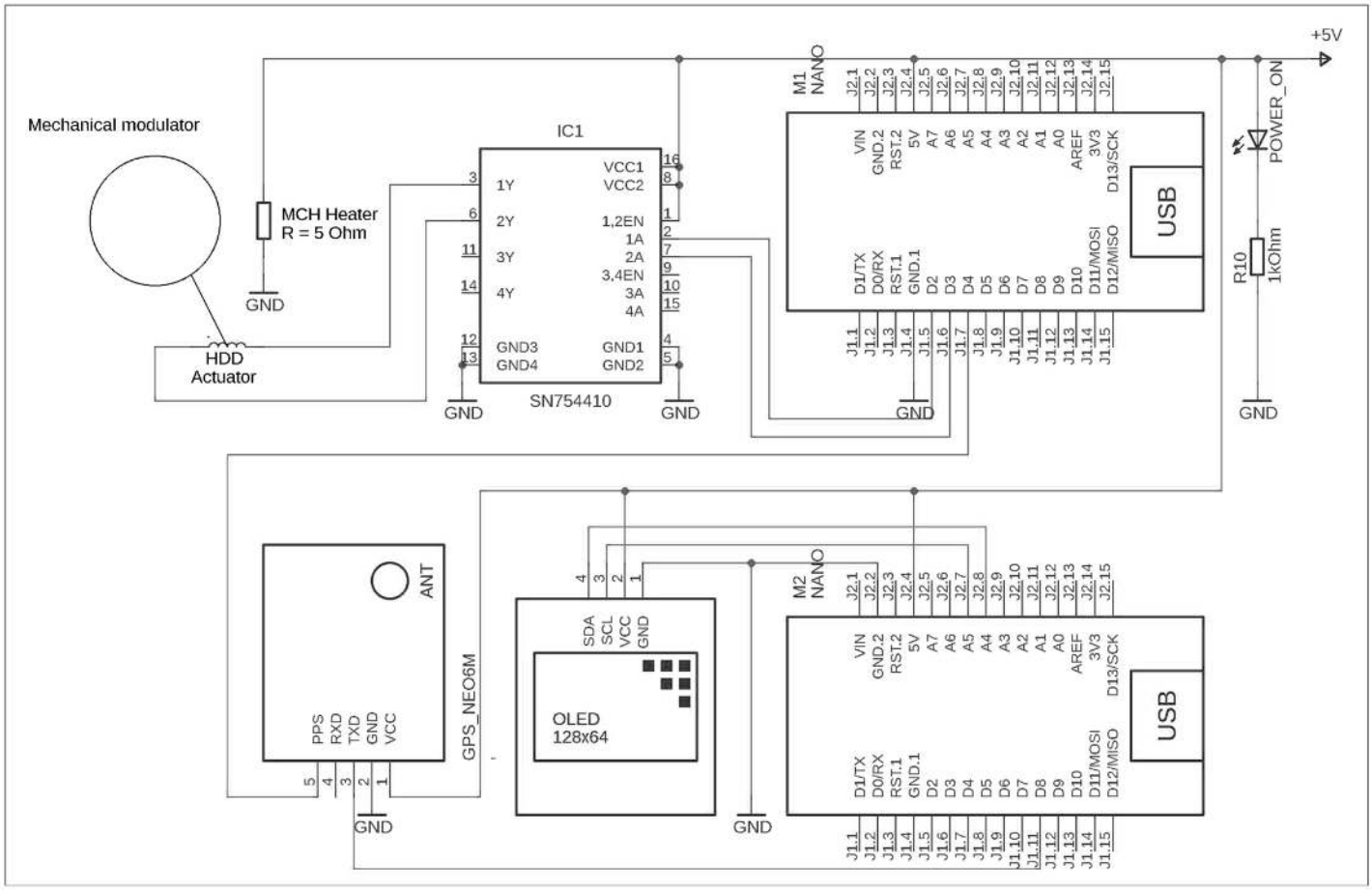


Figure 2 — Schematic diagram of the transmitter.

Figure 3 — Transmitter operating outdoors.

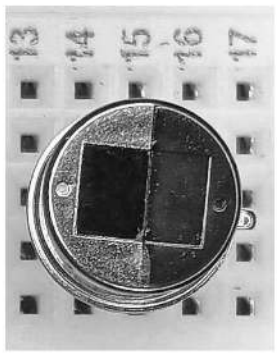


Figure 4 — Passive infrared sensor with one pyroelectric crystal blinded using black nail varnish.

tered and sampled by a 24-bit ADC (**Figure 5**). The aim of the basic analogue high pass filter (R2, C2) is to remove the dc component and slow noise related to ambient temperature fluctuations, wind, and receiver heating from the Sun. An Adafruit NAU7802 24-bit ADC module is used to digitize the signal. The NAU7802 is designed as a load cell ADC, and a custom Arduino library was written to optimize its use for 30 THz receivers. The digital signal, together with the 1 pps GPS signal, is logged on the SD card. Additionally, each file contains at the beginning a date stamp from RTC. The GPS modules in the receiver and transmitter are identical. There are two functions of the GPS module in the receiver: to produce a 1 pps signal for lock-in detection and to show the exact location of the receiver. The receiver is controlled by an Arduino microcontroller that also continuously sends raw data from the ADC to a laptop PC, which works as a raw sig-

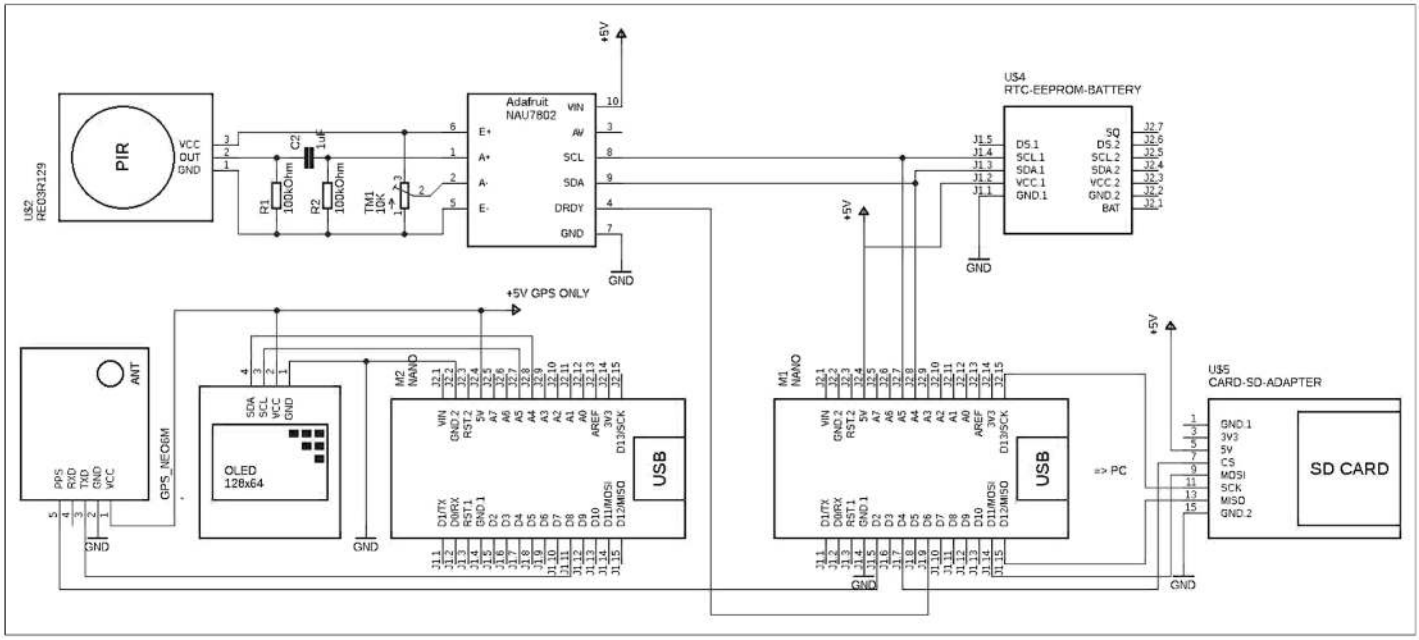


Figure 5 — Schematic diagram of receiver.

Figure 6 — Geodesic Tripod with receiver.

nal display during the experiment (Arduino plotter). A laptop powers the receiver (except the GPS module). A separate 5 V battery powers the GPS module to reduce interference. The receiver is mounted on a geodesic tripod (**Figure 6**) to allow for precision receiver alignment.

Part IV: The Experiment in the Field

There is a short video on *YouTube* describing this experiment **[6]**. The transmitter was directed manually towards the receiver. The transmitter with a mirror from the GU10 spot lamp exhibits poor beamwidth (the beam is probably >20 degrees wide), but in contrast, the receiver using a telescope as the "dish" has a very narrow beam (less than one degree) and requires a special technique for alignment, described below. The distance between the transmitter and receiver was over 100 m, measured using a GPSmap 62ST as 109 meters. The Celestron FirstScope was initially used as a scope to locate the transmitter (at magnification 15×), and for rough alignment using a high magnification (75×). In the next stage of alignment, the optical eyepiece was gently replaced with a 5MP video camera connected to a laptop computer as display for precision alignment (**Figure 7**). Before the experiments, the telescope was calibrated with the camera for alignment at a distance of 10 m. In the final step, the video camera was replaced with a 3D printed "PIR eyepiece" and the telescope was used as a high gain "dish" antenna for 30 THz; the signal was monitored using a laptop display (Arduino plotter) as described in the receiver section.

The signal was immediately visible on the screen (Arduino plotter), and the final adjustments were made using a geodesic tripod leveling screw. The signal was then logged for further analysis. It is reasonably easy to incorporate locked-in phase shift demodulation into Arduino software; however, at this stage of 30 THz band development, the signal recording allowed further multiple analyses and may facilitate future educational activities. Additionally, an exact position, from the built-in GPS modules in the receiver and transmitter, was

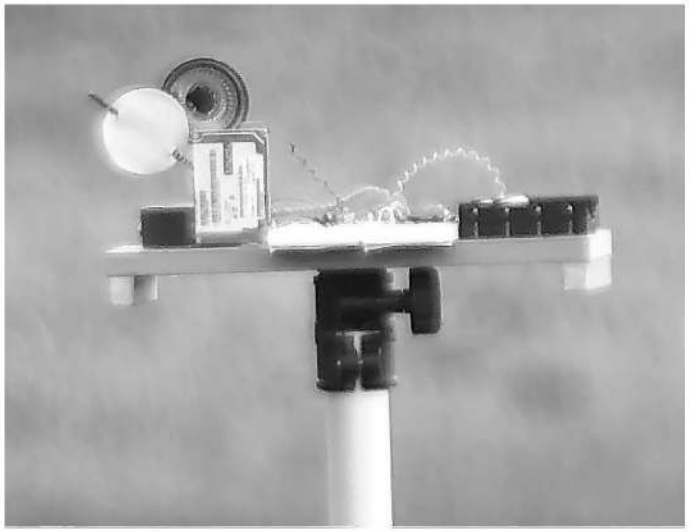


Figure 7 — The transmitter observed through Celestron FirstScope with a video camera at a distance of 109 m.

noted and later plotted on Google Maps, confirming the 109 m distance. The error of GPS distance measurement is estimated at ±2 m.

Part V: Data Analysis

The data analysis was performed using the SciPy Python library in Jupyter notebook. **Figure 8A** shows a sample of the raw recorded signal. The same signal but after the digital Butterworth band-pass filter (3rd order, 2 Hz center) is presented on **Figure 8B**. A phase shift is visible in the middle of graph. Using the logged timing from GPS, a 2 Hz sinusoidal signal was generated for phase locked detection. A 175 ms phase correction was used to address delays caused by PIR inertia, analogue HPF and digital BPF. For phase-locked detection, the filtered recorded signal was multiplied by a phase-locked 2 Hz sinusoidal signal (**Figure 8C**). Finally, a digital 0.2 Hz low-pass filter was used to decode a simple message containing the two Morse letters, "CQ" (**Figure 8D**).

Part VI: Future Perspectives

This work presents the transmission of a simple slow CW signal at a distance of 109 m, which at the time of this writing may be the longest distance in 30 THz achieved so far; but it is still a very short distance. The value of the presented work is not in the distance achieved but in showing future perspectives and areas in which progress may be easily achieved:

- The signal, after analysis, was at least 10 dB above noise, so even with an unmodified design longer distances can be achieved;
- The low power allows for portable use, but in order to achieve longer distances, higher power may be used. A portable camping gas heater could be an interesting alternative to an electric heater;
- Better mirror in the transmitter would achieve an additional 20 – 30 dB signal gain;
- The receiver uses a telescope with a 76 mm mirror diameter, but telescopes with bigger mirrors are easily available;
- The receiver also uses the cheapest available PIR detector. Other detectors (spec-

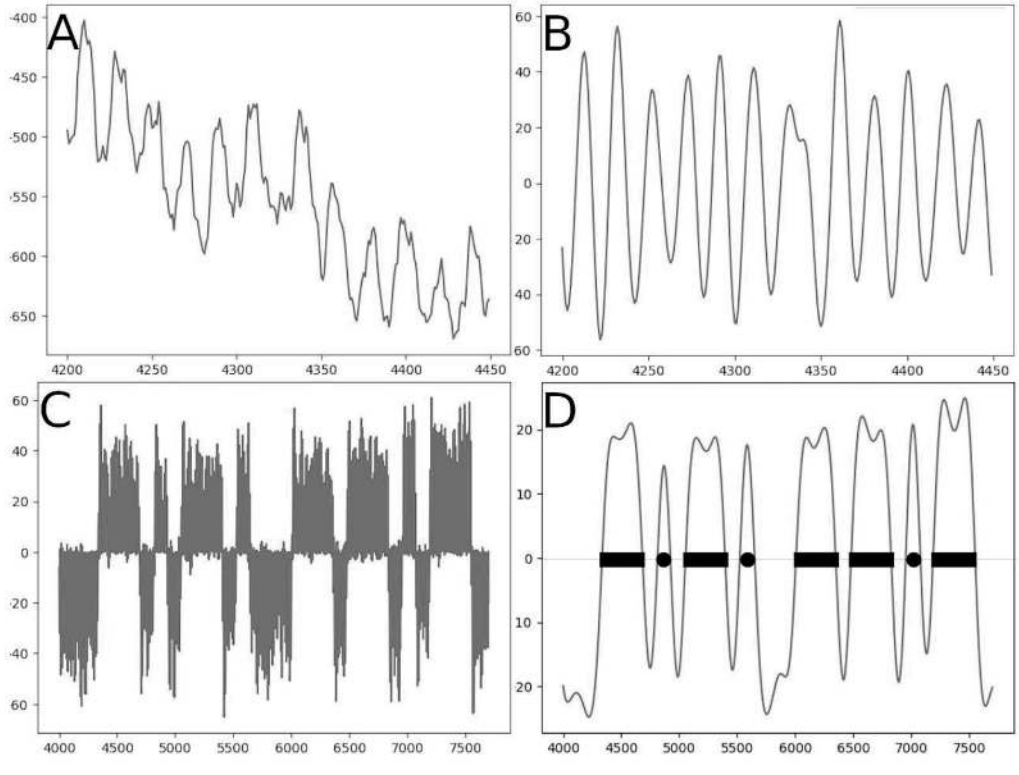


Figure 8 — Analysis of data at a 109 m distance: (A) sample of the raw recorded signal; (B) signal after digital Butterworth band-pass filter; (C) phase locked detection, signal from (B) multiplied by 2 Hz signal; (D) decoded "CQ" message.

troscopy pyroelectric detectors, thermopiles, FLIR cameras, bolometers, microbolometers, etc.) may provide better sensitivity — this is an obvious area for experimentation.

With the above described easily available improvements, a distance between 1 and 10 km sounds easy to achieve, approaching 100 km would be a real challenge.

There are also technologies not yet affordable for ham radio amateurs, which may revolutionize the 30 THz band soon. I will mention here only two. The first is a Tunable Quantum Cascade Laser, which is able to generate a tunable narrowband signal in the 30 THz band. The latter is the already commercially available, (but rather expensive) very fast (100 MHz bandwidth), HgCdTe (Mercury Cadmium Telluride) detectors for 30 THz. With technological progress, the 30 THz band may exhibit great potential in the near future.

Should the reader want the Arduino software (receiver, transmitter, GPS modules, NAU7802 library for 30 THz), completed Jupyter notebook and samples of raw data recorded at distance 109 m, please get in touch with me at: **m0lrh@lecybyl.com**.

See **www.arrl.org/QEXfiles** for a companion document with additional information.

Acknowledgements

Authors acknowledge and thank Izabela, M7IBL, and Tymek, M7TBL, for proof reading and for ongoing support during experimental work.

Hieronim Lecybyl, M7HBL, was first licensed as a radio operator in 2020. He is a member of the Radio Society of Great Britain and the Cray Valley Radio Society. Currently, he is studying Mathematics and Computer Science.

Remigiusz Lecybyl, MØLRH, has been interested in electronics for approximately 40 years and was first licensed as a radio operator in 2020. He is a member of several organizations, including the Radio Society of Great Britain, Cray Valley Radio Society, GQRP Club, UK Microwave Group, and FISTS CW Club. His radio interests are focused on frequencies above 100 GHz, and he is dedicated to exploring this area of the radio spectrum.

References

[1] **https://ghz-europe.com/new-30 THz-band-used-for-experimental-amateur-radio-communications/**

[2] A. J. Anderson, VK3CV/WQ1S, "Bridging the Terahertz Gap at 30 THz," *QEX* Nov./Dec. 2021, pp. 8 – 12.

[3] B. Chambers, G8AGN, "30 THz – It's Radio, But Not As You Know It;" *QEX* Mar./Apr. 2022, pp. 3 – 8.

[4] **https://groups.io/g/Amateur-30THz**

[5] C. Whitmarsh, GØFDZ, "30 THz Radio"; Cray Valley Radio Society Hybrid Meeting, London, 17 March 2022.

[6] **https://www.youtube.com/watch?v=Coo5u3XPHcs**

Michelle Thompson, W5NYV
5379 Carmel Knolls, San Diego, CA 92130 | **w5nyv@arrl.net**

Opulent Voice

Error correction and interleaving together can reverse some of the damage done by noise and interference.

Opulent Voice is a modern open source amateur radio voice and data protocol suitable for 222 MHz and above. Transmitted voice is of excellent quality, with bitrates starting at a default of 16 kbps and can be configured up to a maximum available bit rate of 500 kbps. Data is transmitted without having to use a separate packet mode.

Opulent Voice uses modern digital communications techniques including randomization through scrambling, error correction enhancement through interleaving, and more. The transmitted signal is divided up into frames. There is a Preamble, Frame Headers, Sync Words, and Payloads. There are two types of Forward Error Correction.

A Preamble is a pre-defined burst of signal that allows an Opulent Voice transmission to be quickly recognized by a receiver. It's sent once, at the beginning of a transmission. Frame Headers contain vital information about the link such as transmitter identification and authentication values. Sync Words are pre-defined bursts of signal that help keep frame timing. They help ensure that we are looking at the correct part of the signal at all times. Forward Error Correction inserts additional bits in the transmission that make it possible to correct errors at the receiver.

At the transmitter, we do the following in order. We encode our data with additional error correcting bits, apply interleaving, scramble the interleaved bits to make our data appear more random, insert a Sync Word to mark frame boundaries, and finally modulate and transmit it over the air. **Figure 1** is a drawing that summarizes the process of creating a Payload Frame and inserting a Sync Word at the transmitter.

Each frame in Opulent Voice is 40 ms long and was formed using minimum frequency shift keying modulation with four tones. Think of this as four distinct frequency tones being sent one after another. It's like sitting at a piano and playing four particular notes. Each note is heard across the room by someone with perfect pitch, who then writes down the notes on a piece of paper. Since there are four tones, we have four different values of information sent per piano note. Four values translates into two bits of information per tone. 40 ms frames of Opulent Voice sent at 18,700 FSK tones per second means we are getting 37,400 bits per second. Minimum Frequency Shift Keying means that we set the frequency difference between the tones to half the symbol rate. The waveforms that represent 00, 01, 10, and 11 differ by 9350 Hz. This makes our signal spectrally efficient.

As the signal travels between transmitter and receiver, it picks up noise and interference of many types. Some noise comes from the circuits we're using. Some noise comes from the natural environment. Some noise comes from other radio signals. The distance that the signal has to travel affects the received power. All of these deleterious effects combine. The result is that our received signal will be noisy and damaged. What can we do at the receiver to take advantage of all the work the transmitter did? Error correction gives digital communications a critical advantage over analog communications. Error correction can reverse damage done by noise and interference. How does an Opulent Voice receiver use the extra bits that were sent to detect and correct errors?

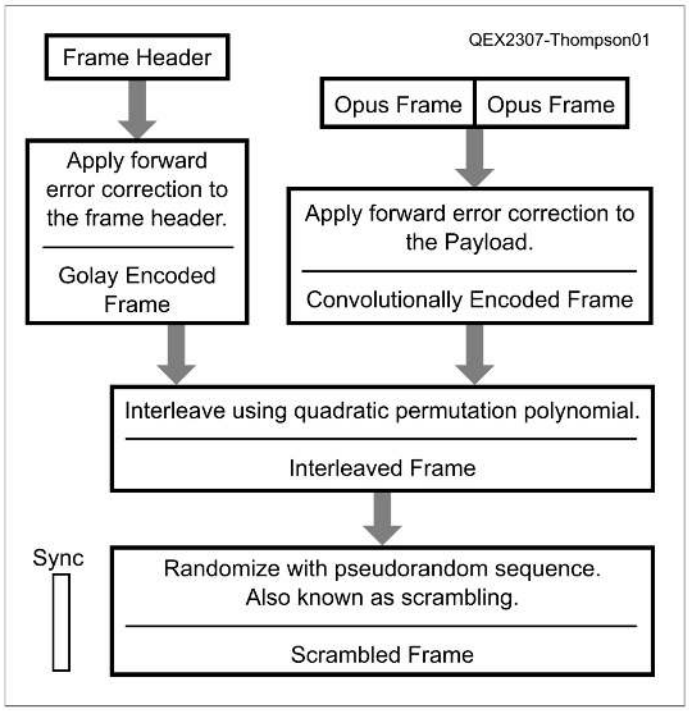


Figure 1 — Creating a Payload Frame and inserting a Sync Word at the transmitter.

There are some measurements and some calculations that we need to make in order to use and understand error correction. The math is simple, but the concepts are quite powerful.

First, we need to know our Signal to Noise Ratio (SNR). SNR is the signal power divided by the noise power. We use SNR to measure how tough it is out there for our signal compared to the noise. We usually see SNR expressed in dB, but in this calculation we are going to keep SNR as a linear ratio. Second, we need to know our bandwidth in Hz. Third, we calculate our Channel Capacity:

$$Capacity = Bandwidth \times \log_2(1+\mathrm{SNR}) \text{ in bits per second.}$$

What does this calculated value of Channel Capacity tell us? At any bit rate below our Channel Capacity, we know for sure that an error control code can be designed that can bring our error rate down to something arbitrarily small. For any particular SNR and bandwidth, we know for a fact we can control our error rate if we stay below the Channel Capacity.

There's a catch. This theorem doesn't tell us how to design this error control code. It just tells us that one exists. It's up to us to figure out how to construct a particular error correcting code. Figuring out better error correcting codes is where a lot of energy is spent in digital communications engineering. The results over the past few decades have been nothing short of amazing, with a variety of error correcting codes that operate very near the Channel Capacity and are also relatively easy to implement.

Opulent Voice transmissions can be thought of as something like a freight train. There's an engine up front pulling a lot of train cars. Each train car, or frame of data, is full of bits. We take the information that we want to send, which might be digitally encoded voice, and pack these bits into the train cars. When we pack them in for their journey, we prepare the bits in two ways. First, we take out any unnecessary redundancy in the digitized voice. This is called source coding. Think of this as re-packing the boxes we want to send as freight to eliminate empty space in order to make more efficient use of the train car. Opulent Voice uses an open source voice coding and decoding system called Opus. With Opus, our voice frames are dense and valuable cargo.

Once we've gotten rid of any extra bits that don't help voice quality, by using Opus, we then pack the bits into the train cars by adding carefully chosen additional bits for error correction, by using our channel codes. Golay Codes and Convolutional Codes are used for channel coding.

Source coding and channel coding work together. The goal is to first get rid of bits that aren't strictly necessary to express our information. We may also optimize the bits we have left, with techniques like equalization or filtering. We then add in bits that are going to help make our signal more resilient for the journey over the air. These extra bits are like having extra players on the field in a ball game, who are all working together to get the ball down the field without it being lost to the other team. Forming an effective team and picking the right strategy to get past the opposing players is a very similar process to what happens in forward error correction.

The default version of Opulent Voice, with a 16 kbps version of the Opus voice codec, operates at 18,700 symbols per second (37,400 bits per second), and sends 1496 bits in every 40 ms frame.

Golay Code is used for the Frame Headers, and a Convolutional Code is used for the Payload frames. The Payload frames contain our data. If it's voice, then we are sending Opus frames. If it's data, then we use Consistent Overhead Byte Stuffing (COBS) to frame arbitrary data for our Payload. Regardless of the cargo, whether Opus or other digital data, our train cars all end up being the same length as they roll down the track.

We are now at the receiver, waiting for an Opulent Voice transmission. How do we find our signal? We are looking for the engine pulling the train of frames.

The first frame produced by a transmitted Opulent Voice signal is called the Preamble. It functions like the bright light on the front of a train engine, cutting through the darkness and warning us a train is on the way. The Opulent Voice Preamble is a series of the highest and lowest tones from the set we're using, alternating back and forth. This produces a very recognizable signal in the frequency domain. Once we have identified and received the Preamble, we discard it. Our Preamble has no error correction.

The rest of the frames are sent using combinations of all four tones. Through coding and other techniques, the four tones create a spectrum that looks much more random than the Preamble.

This is an important difference. The Preamble is chosen to be very distinctive, with very little uncertainty. It has very little information in it. These characteristics make it easy for the receiver to find. The rest of the frames have to pack in a lot of information. They have a much richer spectrum, and look more like a lump of noise. We use the Preamble to find the start of the transmission.

Once we've identified the start of the transmission, we wait for the end of the Preamble and start looking for the start of the first full frame. We know we have 40 ms frames, but where do those frames start if the rest of the transmission just looks like an undifferentiated bunch of noise? The Preamble gave us some indication, but it's not the only thing the transmitter did to help us at the receiver. The start of every frame coming after the Preamble begins with a distinctive Sync Word. Think of this as a smaller light or reflective patch on the front of each train car. The transmitter constructed each frame to begin with a fixed pattern of 16 bits. We know that pattern, so when we see it we know exactly where the data frame starts, every time. When we receive a frame, we confirm the first 16 bits match our expected pattern. We can then set those bits aside and examine the rest of the frame with confidence that we know when and where we are in the transmission.

After removing the 16 bits of Sync pulse from the 1496 total bits in the frame, we are looking at 1480 bits of data. These 1480 bits were scrambled up in a particular way to make the signal look more random. This is an important characteristic for digital communications. A random signal doesn't have long stretches of zeros or ones, but our structured voice and data signals certainly can have long stretches of zeros or ones. We don't want long stretches of zeros or ones. More transitions in the data pattern helps our receiver synchronize to the proper bit timing. Long stretches of zeros and ones don't have transitions on the bit boundaries. So, we use scrambling to change the statistics of the signal to something we want without destroying or changing any of the underlying information. We do this by using a pseudorandom scrambling sequence to transform the bits in our frame so that we're sure we have lots of bit transitions to work with.

Pseudorandom sequences have a starting point and the sequence repeats after some number of bits. They aren't random, but they have statistically similar characteristics to random numbers. We exploit the characteristics of pseudorandom sequences to get large benefits for a very small cost.

Specifically, in any pseudorandom sequence, 1/2 of the runs (of zeros and ones) have length 1, 1/4 have length 2, 1/8 have length 3, 1/16 have length 4, and so on. The number of runs of zeros is equal

to the number of runs of ones. And, a pseudorandom sequence has an equal number of zeros and ones. It might be off by one if the sequence has an odd number of digits.

We need to unscramble our received bits in order to get any further. We know the exact pattern the transmitter used, so we reverse that pattern and restore the original bit values. How did the transmitter scramble up the data to make it look random? The transmitter will XOR (exclusive OR operation) the entire frame with a fixed known sequence of pseudorandom bits the same length as the frame. **Table 1** shows some of the XOR operation done at the transmitter. **Table 2** shows the XOR operation done at the receiver.

Applying XOR operation twice, using the same pseudorandom sequence, gives us our original data bit back. So, at the receiver, we take a frame, XOR it with our pseudorandom sequence, and we get back what the transmitter had before it scrambled the data. So, now we've got unscrambled data. The next layer down is interleaved data.

Interleaving re-orders the bits. It's different from scrambling, where the bits were flipped in place to make them appear more random. Interleaving moves bits around in a particular way. We want bits to spread out as far apart from each other as possible. Why interleave? Well, it has to do with being clever about sudden crashes of noise blasting our signal. A steady level of noise will cause some amount of damage to our transmission. But, every so often, we will get a large amount of noise that might appear for a long enough period of time to completely wipe out a sequence of bits. The rest of the time, the signal is in decent shape with an occasional single-bit error, but with these larger bursts we miss out on a chunk of data, and many bits might be missing in a row.

Why is this a problem? We can only correct so many bits at a time. The background error rate from typical channel conditions may be well within our ability to correct, but these sudden bursts of noise are a problem because our error correction algorithm will fail. Wouldn't it be a great thing to be able to distribute these burst errors throughout the frame? Can we somehow turn the burst errors into a slightly higher average error rate? We can and this is called interleaving. And, this is something our Opulent Voice transmitter did for us. That means it is something we have to un-do at the receiver before we can get to the layer with the error correction.

Here's an example of how interleaving works with a famous poem called Ozymandias by Percy Bysshe Shelley. Here's the data we sent:

"My name is Ozymandias, king of kings:
Look on my works, ye Mighty, and despair!"

Nothing beside remains. Round the decay
Of that colossal wreck, boundless and bare
The lone and level sands stretch far away."

Here's our transmission with a significant burst error marked with **X**.

"My name is Ozymandias, king of kings:
XXXXXXXXXXXXXye Mighty, and despair!"

Nothing beside remains. Round the decay
Of that colossal wreck, boundless and bare
The lone and level sands stretch far away.

Well, we aren't going to be able to tell what was under those errors. What if we interleave the poem before sending it? We know we're going to get this type of burst damage, so we mix up the letters in a predictable way and then send it over the air.

soyba hh y a bk oii a h skn !.l bpu" sia nhynri
nn wcia y,tc, rsegnrynyf'e aeso d,rioamodoe
Ms,gameaan .fwMs ah oeskvtacnNdokL
or:hndsdkcmieaOa nsl odadusl yetef elTnlelmtn
dtogatir rRiseewet dnd gesOzes

We get a burst error. Errors are marked with **X**.

soyba hh y a bk oii a h skn !.l bpu" sia nhynri
nn wcia y,tc, rsegnrynyf'e aeso d,rioamodoe
XXXXXXXXXXXXXwMs ah oeskvtacnNdokL
or:hndsdkcmieaOa nsl odadusl yetef elTnlelmtn
dtogatir rRiseewet dnd gesOzes

We deinterleave the poem at the receiver.

"My name i**X** Ozymandias**X** king of kings:
Look on my works, ye **X**ighty, and despair!"

Nothin**X** beside re**X**ains. Round the d**X**cay
Of th**X**t colossal wreck, boundless**X**and bare
The lone a**X**d level s**X**nds stretch **XX**r away**X**

When we receive the poem, we can see it has some missing letters. But, we now have a much higher probability of being able to sort out all the damage because the errors are distributed throughout the poem instead of all in a row. We can recover most if not all of the damaged words if the words are missing a letter or maybe two. We can't recover an entire missing phrase, which is what would have happened if the original poem was sent over the air with all the letters in order.

This is how interleaving and error correcting codes work together. The error correcting codes are like your brain, fixing the

Table 1 – Input 1 XOR with Input 2 = Result for transmission

Input 1: Data bit	Input 2: Pseudorandom bit	Result: Transmitted bit
0	0	0
0	1	1
1	0	1
1	1	0

Table 2 – Receiving bits

Input 1: Received bit	Input 2: Pseudorandom bit	Result: Data bit
0	0	0
1	1	0
1	0	1
0	1	1

words you read that have occasional errors. The interleaving makes sure that those errors are scattered around rather than concentrated in one place. Interleaving is a common technique in digital signal processing and is required to get all of the functionality out of certain types of error correcting codes. Since we are using one of those types of codes, a convolutional code, we need to interleave.

We do have to know exactly where each original bit came from. For Opulent Voice, we have a formula that told us how to interleave, and this same formula will tell us how to get everyone back in line in the correct position. Without knowing this, nothing else will work. Once we re-order things, then we have all the bits in the right sequential order and can use our error correcting code.

There are several ways to interleave, but Opulent Voice uses a quadratic permutation polynomial interleaver. It is defined by a simple equation. x is the original position of the bit. y is the interleaved position:

$$177x + 130x^2 = y$$

We are interleaving 1480 bits of data, so if the result is larger than 1480 or a multiple of 1480, we take the remainder. In other words, the equation is done with modulo 1480.

Let's look at the first few results in **Table 3**. The bit at position 0 stays where it is. The bit at position 1 is moved to position 547. The bit at position 2 is moved to position 354. The bit at position 3 is moved to position 901. The bit at position 4 is moved to position 708. The bit at position 5 is moved to position 1255.

Let's say we get a burst of noise that wiped out 6 bits in a row of interleaved data. At the receiver, we reversed the interleaving. Those 6 bits in a row of damaged interleaved data are all sent back to their original positions. They are, by mathematical design, as far away from each other as is possible. The quadratic permutation polynomial is selected specifically for how far apart it scatters contiguous bits.

Now we've got deinterleaved data. Our frame starts to show some structure. There are two sub-frames. The first is the Frame Header, and the second and larger of these two sub-frames is our Payload data. In most cases, the Payload will be Opus voice.

The Frame Header is stuck on to the front of each Payload. The information in the Header allows a listener to join a transmission in progress. For a narrowband signal, this can be a lot of overhead. There are techniques to mitigate the overhead, like splitting up the information in the Frame Header and distributing it round-robin style. For higher bit rate modes like Opulent Voice, the relatively small number of bits in the Header are not a burden, so they are sent every time in full.

How do we decode these frames? Well, we read the Opulent Voice specification document and we know we have to deal with 8 sections of 24 encoded bits for 192 bits total. We used a 12 to 24 bit Golay Encoder at the transmitter. We took our twelve 8-bit bytes of Frame Header data, organized it into eight 12-bit groups of bits, and multiplied each of these 12-bit sections by the Golay Code generator matrix. Each of these multiplications resulted in 12 parity bits. For each 12-bit portion of the original data, we attach the corresponding 12-bit parity result to it, and this creates eight 24-bit codewords. These eight 24-bit codewords are what we then interleaved, scrambled, and then sent out over the air. Codes that send the original data plus some parity bits are called systematic codes.

Table 3 – Interleaving

x	y
0	0
1	547
2	354
3	901
4	708
5	1255

When we receive this 24 bit codeword, we unscramble and deinterleave. Then, we correct errors by multiplying the received codeword with another special matrix. This one is called the parity check matrix. It's the partner of the generator matrix that created the parity bits at the transmitter. The result from this multiplication is called the syndrome. If the syndrome is zero, then there were no errors in the codeword. We drop the parity bits and what is left is the original data. If the syndrome has ones in it, then we take that result and go to a lookup table that we've made based on the protocol specification. This table maps the result of the multiplication to a list of which bit positions in the codeword have errors. Whatever bits have errors, we flip them. Now we have a corrected codeword. We drop the parity bits and what is left is corrected original data. This process is called syndrome decoding.

We can correct damage wherever it occurs across the 24-bit codeword, within reason. The limit for this particular type of code is that it can correct 3 errors. It can detect up to 7. If we have a situation where we've detected 1, 2, or 3 errors, then we can correct them on our own based solely on the results of our "secret decoder ring" matrix math we use at the receiver. If we have 4, 5, 6, or 7 errors, we know we have them, but we cannot correct them on our own. 8 or more errors and some interesting things might happen. We will be pulled off course so far that our codeword resolves to something completely different than what was sent. The resulting data stream will be very damaged. There's not much we can do about this, but the level of noise or interference that would cause half or more of lost bits in a codeword are extremely high.

Now that we have these very helpful bits of information about our link decoded, we can turn our attention to the payload itself, which contains Opus voice frames. The payload is convolutionally encoded data. We know that the deinterleaved bits were convolutionally encoded at rate one-half.

When we talk about the rate of a code, we are talking about a ratio between how many bits go into an encoder (the number on the top) and how many bits come out of an encoder (the number on the bottom). A rate 1/2 means that one bit of our original information was turned into two bits sent over the air. For a rate 1/2 encoder, every bit of source data is converted into two parity bits. Notice that the Golay Code is also a rate 1/2 code. 12 bits went in and 24 came out. Not all error correction codes are rate 1/2. There are a variety. Rates won't be less than zero or greater than one.

With a convolutional code, what we send over the air is the parity bits. No original data is sent. At the receiver, we figure out what the original data was by analyzing the stream of coded parity bits. Each sequence of coded parity bits stands for a unique original data stream. We infer the original data from the path the parity bits trace through a particular graph.

For a convolutional encoder, it's all about this path through a graph. The graph is called a trellis diagram, and it's used for both encoding and decoding.

How can we think of this? Is there something in everyday life that might be similar? Yes, there is. **Figure 2** shows a peg game. We drop a disc or ball in at the top, and it hits pegs on the way down. At each peg, it goes either left or right. We can think of this as going in either the 0 or 1 direction. At the bottom, we get a score or payout depending on where it fell. Each path that our game token took from top to bottom is unique. When we play the game over and over, we can see the patterns that we take

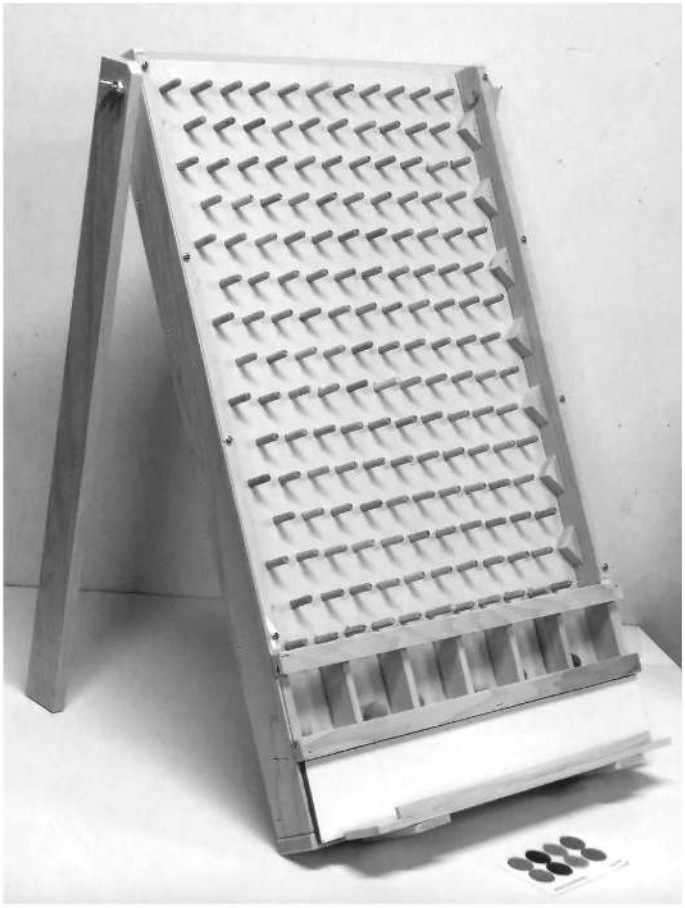


Figure 2 — A peg game.

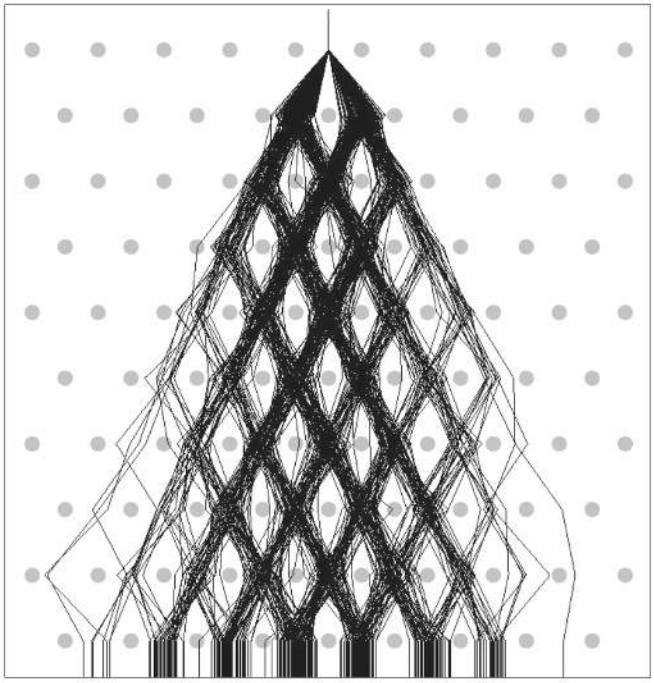


Figure 3 — Paths make something that looks a lot like a trellis.

falling through the pegs. When we do this over and over, and record them, the paths make something that looks a lot like a trellis (**Figure 3**). We can think of gravity the way we think of time. Gravity pulls our token down to the ground through the field of pegs. Time pulls our received parity bits through the states of a trellis diagram (**Figure 4**).

An enduring truth of engineering is that you don't get something for nothing. If you want to magically correct errors, lessen the impact of large bursts of errors by distributing them over time, and then deliver clean and clear results at the receiver as if nothing had happened, then there is a price to be paid. That price is complexity.

When we use convolutional error correction, which does the function that our brain does in correcting the missing letters in the poem *Ozymandias*, then there are things that we have to give up. We are giving up simplicity and flexibility for the ability to repair the data that we received. The benefit that we are getting is that we can repair this data on our own, without requiring it to be re-sent. At least, up to a point. There are limits to how many bits per received codeword we can correct, but with error correction, we can clean up badly damaged signals. An equivalent signal sent without error correction would be useless.

A convolutional encoder is matched to its convolutional decoder. The signal we've created in Opulent Voice can only be undone by the matching Opulent Voice decoder at the receiver. Going back to the comparison to a peg game, our receiver recreates the path the ball took at the transmitter in the peg game. We can do this because the receiver knows what particular transmitted values each peg "stood for" as that peg was hit on the way down. The original information sequence that we sent was defined as "which direction do we choose go at each peg." The path labels between pegs represents the unique parity code. These parity codes are what was sent over the air. Those two bits sent over the air represent the one bit that got us to that path. So, it is the path itself through the field of pegs that stands for a particular sequence of data. The way that we construct this path allows us to lose some bits here and there over the air and still be able to see the most likely path. The path is like a sentence in English. If we lose a letter here and there, we can fill it in. There's only so many missing letters that make sense. We can only paper over so many missing segments of the path through the trellis.

Another metaphor for this process is if we mapped the paths a robotic vacuum cleaner takes while cleaning a room. Let's say we record the paths with time-lapse photography. Even if the robot goes under a coffee table, and we lose part of the path, we can successfully guess which path it took by looking at the trails it made and extrapolating which paths connect together under the table.

The Opulent Voice specification tells us exactly what we need to know to do convolutional decoding in a compact and standard way. We express the particular encoder we are using in the form of a polynomial. A diagram for the trellis defined by the polynomial in the specification can be found at **https://www.openresearch.institute/wp-content/uploads/2022/09/opv-trellis-diagram-segment.png**.

Here is how we encoded the data, with reference to the image at the URL. We start at dark blue node 0 and we take a bit of our input data. We take the green path if the input data was a 1, or a red path if the data input was a 0. Each node has two pairs of numbers separated by a slash. These tell us what parity bits we transmit. We write down the stream of parity bits we create by moving through the trellis with our input data guiding the way. Look at node 15.

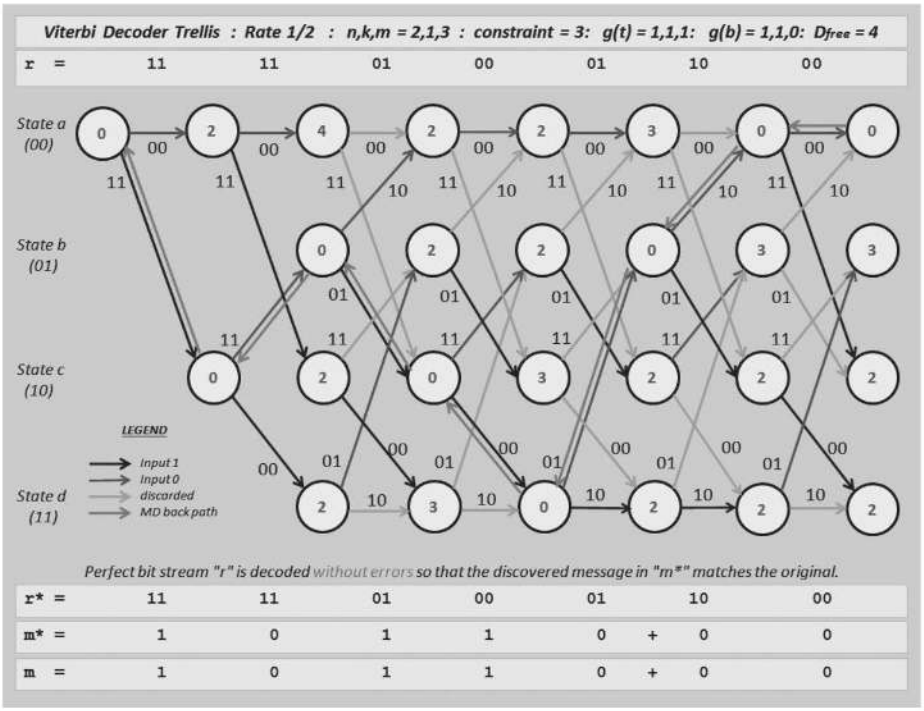


Figure 4 — Time pulls our received parity bits through the states of a trellis diagram.

The top pair of numbers is 0/3. This tells us that if we are at this node, and the input data was 0, we emit 11 (3 in binary). We then take the red path to 15 and select the next data bit. Let's say that data bit is 1. We emit 00 (0 in binary) and then take the green path, which moves us to node 7.

To receive, we use the same trellis diagram. We feel our way forward through the graph, comparing the received set of parity bits to the ones that the path said we'd see. If they match closely, then we'd be wise to pick that path. In the end, the path that most closely matches what we have to work with is picked as the winner. We use this path to define what we think was the most likely original data sequence.

Figures 4, 5, and 6 at the following URL: **https://en.wikibooks.org/wiki/A_Basic_Convolutional_Coding_Example** are examples of decoding by hand using a trellis diagram for a four-state system. Each vertical line of circles is a step in time. The green arrows are where we explore forward, comparing the received parity bits to what was generated by that path when the trellis diagram was used to encode. For each path forward, we mark down how many errors we've accumulated along the way. When we reach a stopping point, we then turn around and look at all the paths we explored, and pick the one that had the best metrics. Usually, this is the path with the lowest number of bit errors.

When we decode we must periodically stop and look back at our path and make a call on what the sequence was up to that point. The choice of how far to go, before looking back and making a decision about a sequence of data, is an important one. The longer we go, the more time it takes to process that section of the bit stream. Too short, and we might end up with not enough space around the coffee table to clearly see which Roomba path was the right one. The determination of how long to let each batch of convolutional decoding run is an important design decision for these types of codes.

This process of convolutional decoding is like going through a house of mirrors at a carnival. Everyone enters the house of mirrors the same way, through a single entry point. As you go forward, you explore the potential paths. Paths that don't work out are paths that you remember as "bad." You do eventually reach the end. Your knowledge of the directions you took through that house of mirrors can be thought of as the original data stream. To decode the next batch of received bits, you enter another house of mirrors with different glass walls and different mirror locations. The original data stream sets the glass and mirrors in a particular pattern. As long as there are not too many errors, then you will have a path through the house of mirrors. You will get the original data back. If you have too many errors in your received parity stream, then you may be completely cut off from the exit. There's no obvious path forward and you can't recreate the data stream.

Digital forward error correction, and digital signals in general, make for a different radio experience from analog communications. In analog, a relatively simple receiver circuit can recover an audio signal from a wide variety of analog transmitters. In order to be able to receive digital signals that have been damaged by noise, our transmitter and receiver are much less flexible. If we have the wrong decoder, then we may get nothing at all from the headphones. A different set of polynomial representations for a convolutional encoder is like having a completely different house of mirrors to work through.

We now have the Frame Header and the two Opus frames that we put into the data Payload. The next step is to decode these two Opus frames and recover the voice signals. We use the decoding functions in the Opus protocol standard, and we get audio waveforms.

That is the story of forward error correction in the Opulent Voice receiver. The system uses two types of forward error correction, a convolutional encoder and a Golay code, to protect our data from noise and interference it encounters over the air. This is a solid protocol done in a common-sense manner that uses modern error correction to achieve very high quality 222 MHz and above voice and data communications. Source code for a C++ implementation of Opulent Voice modulator and demodulator can be found at **https://github.com/phase4ground/opv-cxx-demod**.

If you would like to see more projects like Opulent Voice succeed in the amateur radio community, please join **https://openresearch.institute** at the "Getting Started" menu option. There is no cost. You do not have to be an expert to join, you just have to be willing to become more of one along the way. Everything ORI does is open source and education is a central part of the mission.

Michelle D. Thompson, W5NYV, enjoys thinking and doing — not necessarily in that order! Book learning includes BSEET, BSCET, math minor, MSEE Information Theory. Actual doing includes engineering at Qualcomm, engineering at Optimized Tomfoolery, Amateur Extra-class license, AMSAT Phase 4 Ground Lead, DEFCON, IEEE, Burning Man, and community symphony.

Eric P. Nichols, KL7AJ
AlasKit Educational and Scientific Resources, 3763 Lyle Ave., North Pole, AK 99705 | **kl7aj72@gmail.com**

Self-Paced Essays – #18 Vector Network Analyzer

The VNA is a Very Nice Apparatus.

I was first exposed to the Vector Network Analyzer (VNA) when I began working at HIPAS Observatory in the Fall of 1994. This massive Hewlett-Packard instrument weighed in at about 45 pounds and about $1000 per pound. It greatly simplified a lot of the tasks that we had at the facility, especially when it came to tuning up a variety of antennas.

Before working at HIPAS, I worked with a number of AM broadcast antenna facilities for a couple of decades, where the instrument of choice was a combination of: a General Radio impedance bridge, an RF signal generator, and an HF receiver (in our case an R-390, which itself weighed about twice what the HP VNA did). So we had already made some progress by 1994.

Now we have the NanoVNA, which is no bigger than a smart phone, and does about 99% of what the Hewlett-Packard did at about 0.1% of the cost. Some things have gotten a lot better, even with significant inflation.

The NanoVNA — and its various clones — are so good and so cheap that there's no excuse for any ham, or other RF person, not to have one. Besides the mere convenience of the instrument, the learning one derives from its use is of inestimable value. The only downside is that there is no single authoritative operating manual for the VNA, so one must glean most of it from the Web, which really isn't a bad source. Regardless of your particular "flavor" of VNA, most of the important functions are common to them all.

Three Points, One Port

You can learn a lot about transmission line theory just in the standard calibration process of the VNA. The following discussion addresses only the more common single-port calibration. We will discuss the two-port measurements in a subsequent essay. The process is commonly called the "three-point calibration" the purpose of which is to "calibrate out" the transmission line between your VNA and the device under test (DUT). The nice thing about the three-point procedure is that it doesn't matter how long, or how lousy, the transmission line is. When you perform the calibration at the far end, the VNA will accurately display the impedance of any DUT at the far end. This is one point you can completely miss if you go strictly by the "free floating" instructions out there in cyberspace, which tell you to perform the calibration right at the output terminals of the VNA. If you're using a very short test lead, this will usually do, but it is not the proper way to perform the calibration. You will definitely want to get a nice selection of RF adapters for your VNA, especially some SMA to BNC adapters, so you can conveniently perform the far-end OPEN, SHORT, and LOAD tests. The test loads supplied with the NanoVNA all have SMA connectors.

When you perform the three-point calibration, your VNA then knows what the characteristic impedance, length, and loss of the line are. There's actually a lot of rather sophisticated number crunching that goes on in the VNA that allows these factors to be determined by the three different terminations. We might go into some of this later on. The nice thing about the calibration procedure is, once you've performed it, you don't have to know any of this. All you need to know is that your VNA will now tell you the truth about what's at the far end of the cable. By the way, this was extremely important when network analyzers were the size and weight of refrigerators and you needed to find the impedance of an antenna at the top of a tower. Nowadays, you can just carry a NanoVNA up the tower with you.

With all this in mind, what does a VNA tell you that an old-school impedance (RX) bridge doesn't? Technically, not much. Except that it's a whole lot more convenient. First of all, it performs an automatic frequency sweep of the impedance in question. Second, and perhaps more importantly, it instantly plots the result on a convenient Smith Chart.

Now, if you're an old timer like some of us, you may be shocked and astounded with disbelief to see 'convenient' and 'Smith Chart' in the same sentence, or even in the same paragraph. But not to worry. We will soon prove that the Smith Chart is your friend!

Why Be Normal?

Before we get started, please print out half a dozen copies of a good Smith Chart. Here is my favorite link for printable Smith Charts: **https://www.acs.psu.edu/drussell/Demos/SWR/SmithChart.pdf**.

Now, these are the full-fledged, fully

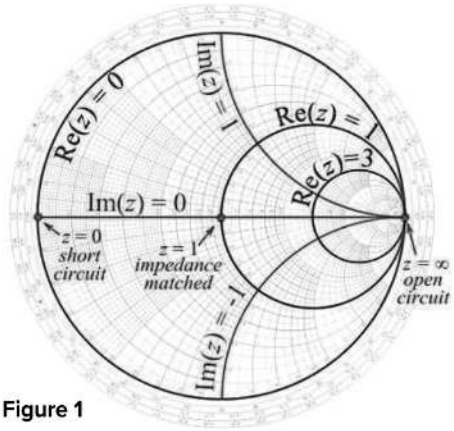


Figure 1

daunting Smith Charts. **Figure 1** shows an undaunting version. I sometimes refer to this as the "fat purple crayon" (FPC) version because I draw it with an actual fat purple crayon — or dry erase marker — in my brick-and-mortar electronics classes.

There are two general forms of Smith Chart, the non-normalized, universal version (as in **Figure 1**), and the normalized version, which is what your VNA displays. The non-normalized version works with any impedance transmission line, while the normalized version is geared around a fixed impedance, normally 50 Ω transmission line, as in the case of your NanoVNA.

I normally (pun intended) taught the non-normalized version first, since it's more general, but since the advent of the NanoVNA, I start out with the 50 Ω normalized version. Nowadays, this is the style most folks first encounter with Mr. Smith. Once we have that nailed down, we will show you how to normalize any Smith Chart with just a couple of extra steps.

The Same but Different

Like the familiar complex impedance graph we explored many essays back, the Smith Chart has both a real and imaginary axis. Like the standard complex impedance graph, up is inductive reactance (+*j*), and down is capacitive reactance (–*j*). The horizontal axis is the real (resistive) axis. No surprises here.

One readily apparent thing that is different is that, while the normal impedance plot has zero ohms (both resistive and reactive) at the origin, the Smith Chart has zero resistance at the far left, and infinite resistance at the far right. The reactance is expressed by a family of arcs, all converging at infinity at the far right, which represent the loci of all possible reactance values. Remember from your ancient geometry classes that a locus is a set of all points that satisfy some condition, a very important concept as we explore the Smith Chart.

Any complex impedance we plot on the Smith Chart will fall on the intersection of two loci: a resistance circle, and a reactance arc.

The center of the chart (origin) is the characteristic impedance of any transmission line we're using. We assume that the transmission line is ideal, exhibiting a pure real value, while in reality any transmission line has some minute value of reactance.

Taking Laps

Now, we get to the salient point of distributed versus lumped constant behavior. While the impedance at any given frequency is plotted at a single point of intersection between resistance and reactance, we find that this impedance will change depending on where it is located; in this case, where it is located along a transmission line.

To show this, we now need to look at a fully daunting Smith Chart, which we trust you have printed out. If you take a look at the far left of the chart, just above 0 Ω resistance, you see the nomenclature "wavelengths toward generator." This means exactly what it says. For any complex impedance at the end of a transmission line, we can find out what the impedance is at the input of a transmission line — or at any point in between. Now, we need to know that one complete lap around the Smith Chart is one-half wavelength at whatever frequency we're working with. We will find that any half wave or multiple thereof of transmission line will repeat the impedance at the starting point — most commonly the "load" end. Similarly, if we move a quarter wave along the line, we find our impedance is diametrically opposite the starting point.

Handy Special Case

The quarter wave transmission line is a particularly interesting case, and an extremely useful device. If a load impedance is a pure resistance, the input impedance Z_S of a quarter-wave section is the geometric mean of the characteristic impedance Z_0 and the load impedance Z_L:

$$Z_0^2 = Z_S \cdot Z_L$$

This is one of those formulas you'll want engraved on the inside of your eyelids.

Let's say you have a quarter-wave 50-Ω transmission line with a 150-Ω resistor at the end. What will the impedance be at the input? We rearrange to get Z_S, so,

$$Z_s = Z_0^2/Z_L$$

which gives us 2500/150 or 16.6 Ω. Again, this works only for resistive loads, which is actually quite common. In the next issue, we'll actually work this out with the Smith Chart. In the meantime, please inspect the daunting Smith Chart for a while, and see what great insights you acquire. — *73, Eric*

KENWOOD

Prepare Yourself for What's Coming

Gather information, coordinate efforts, be effective when it counts. TH-D75A gives you the communication technology you need in the palm of your hand.

TH-D75A

144 / 220 / 430 MHz TRIBANDER

Experience the TH-D75A up close and personal by simply scanning this code with your smartphone's camera.

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 WB4APR (Bob Bruninga) in the USA.
*2: D-STAR is a digital radio protocol developed by JARL (Japan Amateur Radio League).

JVCKENWOOD USA Corporation
Communications Sector Headquarters
1440 Corporate Drive | Irving, TX 75038

www.kenwood.com/usa

ADS#11923

Aim Higher

Enter the world of SHF

IC-905

VHF/UHF/SHF All-Mode Transceiver

*Optional

CX-10G

10GHz Transverter

IC-7300

HF / 6M SDR Transceiver

IC-9700

2M / 70CM / 23CM SDR Transceiver

IC-7610

HF / 6M SDR Transceiver

For the love of **ham radio.**

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

ICOM®