Virtual ARRL/TAPR DCC, Sept. 11-12 01
TAPR at QSO Today Virtual Ham Expo 03
TAPR Directors Election 04
Donate to TAPR 04
TAPR and COVID-19 05
multi-TICC: A Multi-Channel Timestamping Counter 06
TAPR at Hamcation 23
TAPR Wear Available 23
Experimenting with WSPR Using Raspberry Pi 24
Write Here! 46
On the Net 46
The Fine Print 47
Our Membership App 48

Virtual ARRL/TAPR DCC, Sept. 11-12

The 39th Annual ARRL and TAPR Digital Communications Conference (DCC) will be a virtual conference on September 11 and 12, using Zoom video communications and YouTube video-sharing platforms.

Registered DCC attendees participating via Zoom will be able to interact with presenters and other attendees via a chat room as well as raise a virtual hand to ask questions. To register, go to https://tinyurl.com/y5feuafn (you don't need a Zoom account to register).

Non-registered DCC attendees can watch the live stream for free on YouTube, however, non-registered DCC attendees will not be able to ask questions or chat. No registration is required for YouTube access (the YouTube URL will be announced and posted on the TAPR website preceding the DCC).

DCC registration is free for TAPR members and $30 for non-members. Members receive a 100% discount at checkout. To register, go to https://tinyurl.com/y5feuafn

Non-members who would like to join TAPR and receive the free DCC pass can simply add TAPR membership and DCC registration to their shopping carts. After checkout, they will receive the free DCC pass when their membership is processed.

Call for Papers and Speakers

Technical papers are solicited for presentation at the DCC. Papers will also be published in the Conference Proceedings. Authors do not need to present at the conference to have their papers included in the Proceedings. Submit papers to Maty Weinberg, KB1EIB (kb1eib@arrl.org) via e-mail by August 15, 2020. Papers will be published exactly as submitted and authors will retain all rights.

Conference papers will be distributed as pdf's to participants. Printed copies of the papers will be available for sale at Lulu.

Also, speakers are invited to make presentations on topics of interest without submitting papers for the Conference Proceedings.

All speakers and presenters must contact Steve Bible, N7HPR (n7hpr@tapr.org) to reserve a slot for your presentation. Indicate whether you need a 15- or 30-minute slot and if you need to present on a specific day (Friday, September 11 or Saturday, September 12). A pre-recorded presentation can be submitted in lieu of a live virtual presentation.

Paper and presentation topic areas include, but are not limited to software defined radio (SDR), digital voice, digital satellite communication, digital signal processing (DSP), HF digital modes, adapting IEEE 802.11 systems for Amateur Radio, Global Positioning System (GPS), Automatic Position Reporting System (APRS), Linux in Amateur Radio, AX.25 updates, Internet operability with Amateur Radio networks, TCP/IP networking over Amateur Radio, mesh and peer-to-peer wireless networking, emergency and homeland defense backup digital communications in Amateur Radio.

Lightning Talks

Ad hoc "lightning talks" on various topics of interest will be announced throughout the DCC. Registered attendees will be able to participate in any lightning talk that whets their appetite.

Hardware and Software Demos

Hardware and software demonstrations will be conducted during the DCC by means of Zoom's breakout room feature.

###

TAPR at QSO Today Virtual Ham Expo

TAPR will have a booth at the QSO Today Virtual Ham Expo (https://www.qsotodayhamexpo.com/), coming to your laptop, tablet, and smartphone on August 8 and 9.

Participate in this ground breaking, virtual international amateur radio expo. Packed with world renowned speakers, exhibitors, and special conference rooms built on a virtual reality platform.

Attendees have the opportunity to:

Listen to and engage with 80+ internationally recognized ham radio luminaries who have committed to lead expo sessions including TAPR Director John Ackermann, N8UR. (Here is the speaker list: https://www.qsotodayhamexpo.com/speakers.html)

Walk through a virtual exhibit hall filled with popular amateur radio suppliers. Watch new product demos, interact directly with booth staff using video, audio, or text conferencing.

Engage with fellow hams without leaving your home ham shack. And save thousands of dollars since you don't have to worry about travel, food, and lodging!

###

TAPR Directors Election

Three Director positions on the TAPR Board of Directors are now open for nomination and nominations may be submitted now.

TAPR Board members serve three-year terms and their responsibilities include:

1) Attendance at both in-person board meetings each year. [One is held at the Hamvention in May, the other at the Digital Communications Conference (DCC) in September.]

2) Regular participation in the continuous board session, which is conducted over the Internet.

3) Active engagement in TAPR's management.

To place a person in nomination, please remember that he or she must be a member of TAPR. Also, confirm that the individual is willing to have his or her name placed in nomination. By September 6, 2017, send that person's name (or your own if you wish to nominate yourself), call sign, mailing address, e-mail address, phone number(s), and a biographical sketch (250 words maximum) via contact@tapr.org or via snail mail to TAPR, 1 Glen Ave., Wolcott, CT 06716-1442..

Nominations close after the call for nominations at the TAPR Membership Meeting at the DCC on September 12, 2020, and an online election will be held from September 19, 2020 to October 2, 2020.

The three Director positions that are up for election are currently held by Steve Bible, N7HPR, Stana Horzepa, WA1LOU and Darryl Smith, VK2TDS.

###

Donate to TAPR

TAPR is now participating in the AmazonSmile program!

When you shop using the AmazonSmile program, Amazon makes a donation to TAPR equal to 0.5% of the price of your eligible AmazonSmile purchases.

AmazonSmile is the same Amazon you know. Same products, same prices, same service.

Bookmark the TAPR AmazonSmile Program link:

https://smile.amazon.com/ch/86-0455870

That link takes you to a special login portal where you enter your normal Amazon credentials and get redirected at the same Amazon home page except there will now be a notice that you are supporting TAPR.

Other ways to donate to TAPR, email contact@tapr.org:

###

TAPR and COVID-19

By Stana Horzepa, WA1LOU

Like every other aspect of our society, COVID-19 affected TAPR.

Strike One was the cancellation of the 2020 Ham Radio Science Citizen Investigation (HamSCI) Workshop in Scranton. TAPR is an integral part of the HamSCI TangerineSDR Project and some of us were on tap to attend and speak at the Workshop.

The show must go on and the brains behind the operation decided to conduct the Workshop online using Zoom and YouTube. The virtual Workshop was a complete success and TAPR did not miss a thing because of COVID-19. The Workshop was fully recorded and is available to watch on the HamSCI YouTube page (https://tinyurl.com/y4sofr77).

Strike Two was the cancellation of Hamvention, where TAPR leaves a big footprint every May. Poof – our forum, booth and annual banquet with AMSAT were gone just like that!

Since none of us were traveling to Ohio for the Non-Hamvention, there would not be an in-person TAPR board meeting, which usually occurs on the eve of Hamvention. Following the lead of the virtual HamSCI Workshop, we decided to have our board meeting via Zoom. George Byrkit, K9TRV, set it up and it worked out great. It was the first time in (my) memory that all the board members were able to attend an "in-person" board meeting!

Strike Three was the cancellation of the ARRL/TAPR Digital Communications Conference (DCC). All our plans were torn asunder and Steve Bible, N7HPR, had the unpleasant task of reversing all the work he had done setting up the DCC in Charlotte.

TAPR's first virtual board meeting (via Zoom)

Again, the show must go on. Emboldened by the success of the virtual HamSCI Workshop, the TAPR board decided to hold a virtual DCC using the services of Zoom and YouTube. Work is underway organizing the virtual DCC and the effort is being conducted via additional board meetings via Zoom.

Healthwise the board has managed to avoid the virus and continues to conduct TAPR's business unimpeded. (Personally, I had a close call in April and had to be tested. The results were negative, but I was on pins and needles for three days waiting for the results.)

Hams are a resourceful bunch and TAPR has managed to beat this thing and carry on the work of the organization as if everything was "normal."

###

multi-TICC: A Multi-Channel Timestamping Counter Based on the TAPR TICC

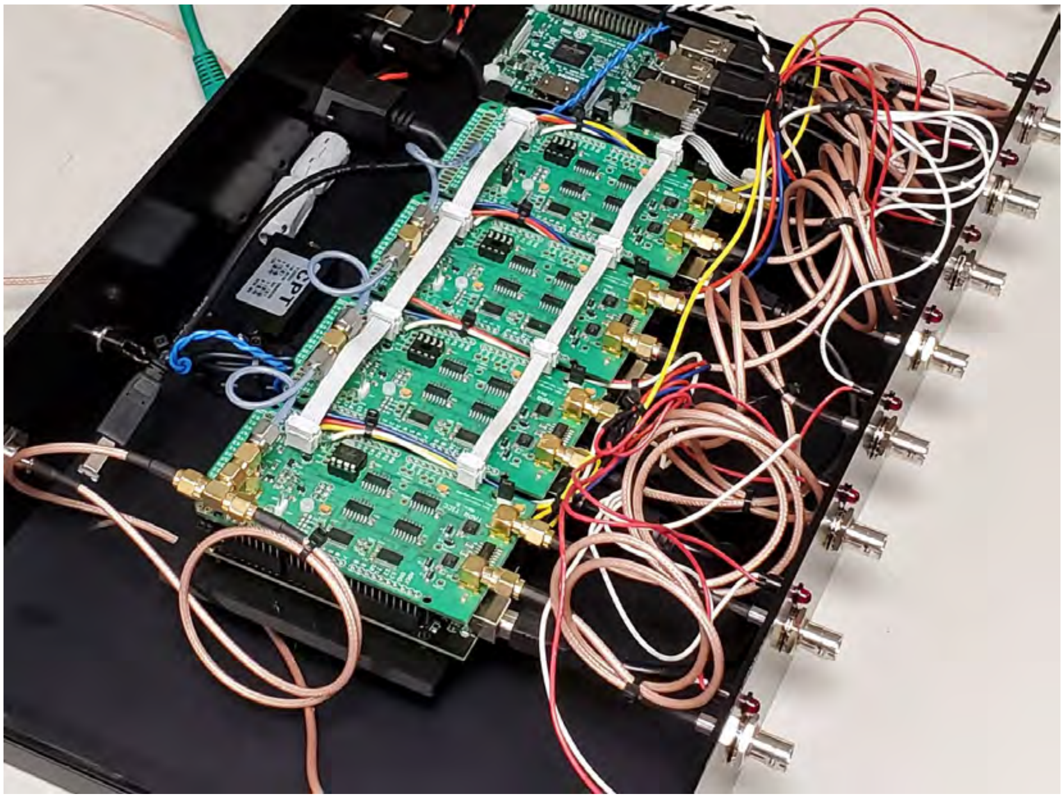
By John Ackermann, N8UR

The following 16 pages contain a pdf of N8UR's article.

###

multi-TICC: A Multi-Channel Timestamping Counter Based on the TAPR TICC

by John Ackermann N8UR[1]



The multi-TICC is an extension of the TAPR TICC[2] two-channel timestamping counter. It consists of multiple TICCs connected to each other and to a single board computer such as a Raspberry Pi. A prototype using four TICCs has been built and extensively tested. It allows timestamping up to eight input channels simultaneously, with the results made available via a network connection.

The multi-TICC can be thought of as a “timestamp appliance”: users can telnet to the unit and download live data from any or all of the channels. Input signals can be connected and disconnected at any time without impacting other channels. When a channel becomes active its timestamps may be directly compared with other channels.

1 jra@febo.com, https://febo.com
2 https://tapr.org/product/tapr-ticc/

If you are interested in assembling a multi-TICC, this application note and the files available at https://github.com/TAPR/TICC hopefully provide the information and design files you need.

The multi-TICC currently is **not** available as a TAPR product. The material cost is significant, as is the labor required for assembly and testing. We suspect that this is very much a niche device, and TAPR can't justify the financial and human resources required to stock it as a completed item. However, we know of sources potentially willing to build units to order. Please contact me if you're seriously interested in that.

A Bit About Timestamping

A **timestamping counter** uses a variable in its software program that increments in synchronization with an external **reference clock**. The variable represents the time elapsed since a fixed starting point (usually when the unit was powered on), and can be thought of as a **timescale** on which external events can be placed in time. Each event, typically the rising edge of a pulse-per-second ("**PPS**") signal output by the device under test ("**DUT**") is marked with its time of arrival (a "**timestamp**") on that timescale, and the sequence of timestamps is output via a communications port for further processing.

A timestamp is a low-level measurement and in most cases a single one is not of much value. But from a series of timestamps it is possible to derive phase, frequency, stability, and other information about an input signal compared to the reference clock. Time and frequency analysis software such as TimeLab[3] and Stable32[4] can read a sequence of timestamps and process that data to provide a wide range of information about the DUT.

A timestamping counter, like the TAPR TICC, that has more than one input channel can also be used for more complex measurements. For example, if a PPS signal from the DUT is applied to channel A, and a signal from another PPS source is applied to channel B, one can calculate the time interval between them simply by subtracting the timestamps. From that data one can also derive other useful measurements such as period and ratio. Timestamps are very useful building blocks.[5]

The two channels of the TICC can thus be used together for traditional time interval measurements, or independently to provide timestamps from two independent DUTs.

The multi-TICC Architecture

The TAPR TICC is a two-channel timestamping counter that consists of a "shield" mounted to an Arduino Mega 2560 controller. Firmware on the Arduino controls the system and provides output via its USB port. The TICC provides on-board headers that expose clock and synchronization signals, and by connecting these signals across two or more units, multiple

3 http://www.miles.io/timelab/readme.htm
4 https://ieee-uffc.org/frequency-control/frequency-control-software/stable32/
5 With the current TICC firmware, the timescale will not wrap for about 5.8 million years.

TICCs can share a single timescale. One TICC is set as the “host” in its configuration, and the others are set as “clients”. The host provides clock signals to the clients. In this way counters with four, six, eight, or even more inputs can be assembled, and one can compare measurements across all channels.

The USB output from each TICC is fed to one of the inputs of a single board computer such as a Raspberry Pi. Software on the computer reads the input streams and makes them available via an ethernet connection. The output can include data from one, many, or all of the TICC channels. Channels with no input signals are silently ignored until a signal appears.

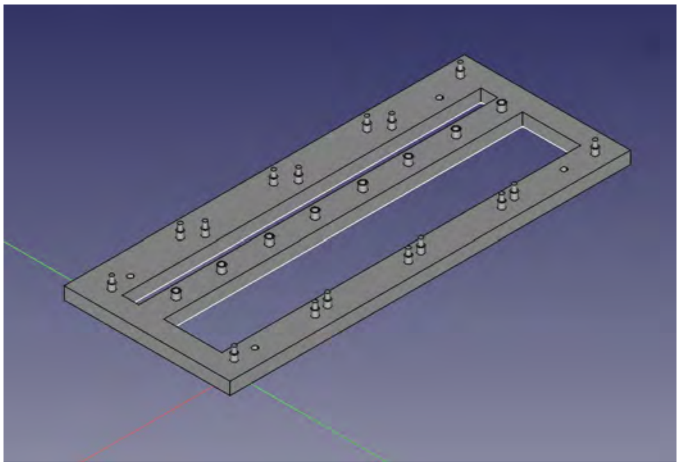
This means that the multi-TICC can be kept running across many measurement cycles, and measurements from multiple channels and multiple measurement cycles can be matched in sequence by their timestamps.[6] With the current TICC firmware, the timescale will not roll over for far longer than any of us need to worry about. Typically, only an interruption of the 10 MHz reference signal will require a reset of the multi-TICC system.

This application note describes the main aspects of assembling and using a multi-TICC: hardware setup, firmware updates, and host processor software; and also provides some test results.

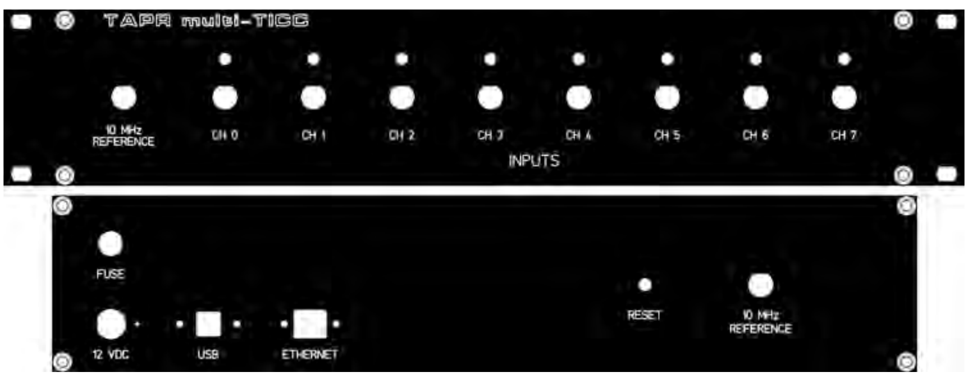
6 Enormous timestamps bring with them computational challenges. It’s often desirable to have timestamps “wrap”, or roll over like an odometer, to keep the value in range. In effect the value is truncated to two or three places. For example, when the timestamp reaches 100, the next sample might start again at 0. Most time stability analysis software tools can work with data wrapped in this way.

Mechanical Design

Four TICCs flopping around on the workbench are a bit hard to manage. To make it much easier to work with the boards, I created[7] a 3-D printed carrier that will align and secure 4 boards. That carrier can in turn be mounted to a base plate.



If you want to make your multi-TICC all pretty, I've designed a complete 2U rack enclosure using the Front Panel Express[8] design tools.



The design files for the carrier and metal enclosure are available in the multi-ticc/enclosures directory of the GitHub repository.

7 With much help from Mike Suhar, W8RKO.
8 https://www.frontpanelexpress.com

Wiring

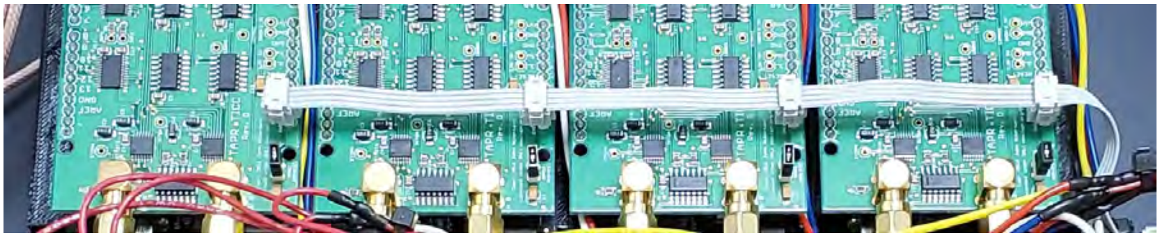
There are three points of electrical connection between the boards in a multi-TICC system.

1. A common 10 MHz clock is applied to all boards through either a passive splitter or a distribution amplifier like the TADD-1. The “TERM” termination jumper should be installed on all boards. Note that if you use a passive splitter, you may need to increase the 10 MHz drive level to compensate for the ~7 dB loss in a typical 4-way splitter.

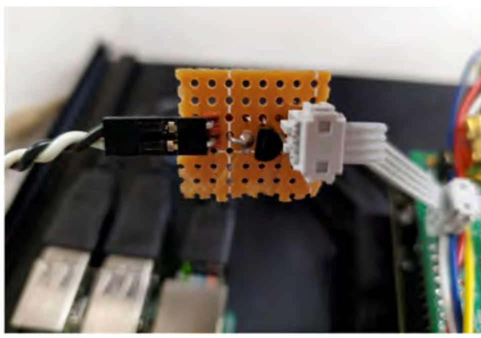
IMPORTANT NOTE: I originally used daisy-chained connectors (short SMA-SMA cables and SMA tee adapters) to pass one 10 MHz signal cable to the four TICCs, but that seemed to result in odd behavior, with boards showing differing noise levels; in the worst case, noise levels on one board would noticeably rise, or one board would show occasional phase jumps of several hundred picoseconds. There’s more discussion of this issue in the “Performance” section below, but testing has shown that feeding each board separately from a splitter or distribution amplifier, and setting the termination jumper on each board, avoids the problem.

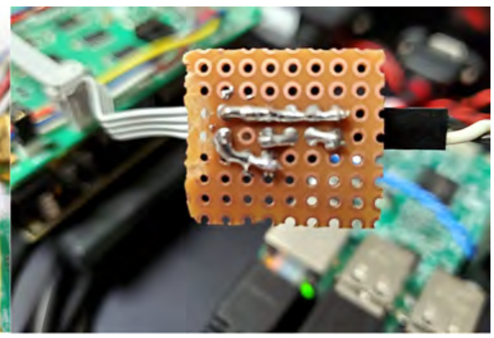
2. As shown below (but ignore the daisy-chained coax lines!), the three pins of the JP2 header are connected in a daisy-chain fashion across all the boards. The easiest way to make this cable is to use 0.1 inch spaced IDC ribbon cable connectors with 6 positions – just make sure you plug the same row of the connector onto each JP2! If you use the 4-board frame described above, the nominal spacing between JP2 headers is 2.25 inches.

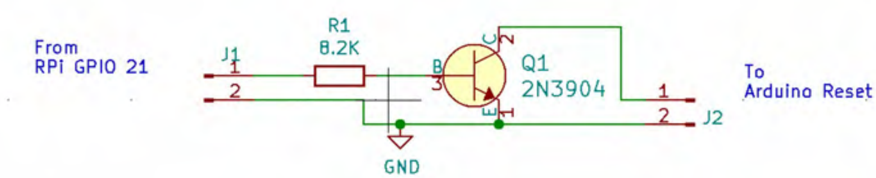
3. As shown below, the RESET lines of all the boards are also connected in parallel, so that they can be reset simultaneously. To do that, solder a 2 pin header in the holes marked “GNDB” and “RESET” located on the right front of each board, and make a daisy-chain connector tying them together. Again, it’s easiest to do this with a 4 pin ribbon cable. Allow extra length at one end if you want to allow the Raspberry Pi to provide a reset function as discussed in step 4 below. One note: with the reset lines in parallel, a powered-down TICC will load the reset line of the others, with the result that they will not start. Make sure all the TICCs are powered up when the reset lines are daisy-chained.



4. If a Raspberry Pi is used as the host computer, one of the Pi's GPIO pins can be used to cause the TICCs to reset via software command. The problem is that the Arduino processor board uses 5V logic and the Raspberry Pi is 3.3V. The very simple level translator shown below can be used to interface between the two boards. It's easy to assemble this on a small piece of perf board as shown below.







5. Install shorting blocks on the "DISABLE AUTO-RESET" header (JP1) on each board. If this is not done, each time a connection is made to the Arduino serial port, that TICC will reset, and that will cause synchronization problems. With this jumper installed, you will not be able to upload new firmware without removing the jumper or performing magic. The multi-ticc_updater.py program discussed below does the appropriate magic and you can use it to update firmware even if the DISABLE AUTO-RESET header is shorted. If you don't use that tool, remember to remove the jumper block before programming.

6. On "client" boards, remove IC10 (the 12F675 PIC). This is because the coarse clock signal generated by the PIC on the master board is routed via JP2 to the client boards.

7. If you are building a front panel for the unit, each TICC has three connection points at A11, A12, and A13 for external LEDs. The LEDs should have a 270 to 1k resistor in series with the anode lead, and the cathode grounded. A11 is active when the system detects that

the required clock signals are present. A12 and A13 track the on-board LED1 and LED2, which indicate presence of signals on chA and chB respectively.

In the daisy-chain multi-TICC configuration, it makes sense to use the A11 signal on the last board of the chain as an overall “status OK” indicator since if that board is clocking properly, it is highly likely that the others are as well.

With these changes, your set of boards is ready to emerge as a multi-TICC!

Firmware Configuration

The firmware shipped on recent TICC units[9] has basic multi-TICC capability included. However, a newer version (20200412.1, available in the TICC GitHub repository) adds helpful features for multi-TICC use:

1. You can assign channel IDs other than A and B via the [I] ("channel name") configuration item.

2. You can use configuration item [J] ("PROP_DELAY") to set an offset value, or propagation delay, in picoseconds for each channel which will be added to the values reported from that channel. This allows you to compensate for the length of the interconnect cables and other sources of board-to-board delay. (PROP_DELAY serves the same purpose as the existing FUDGE0 variable, but I've been requested to make two delay settings available. The PROP_DELAY and FUDGE0 settings are additive.)

3. A bug fix enables an external LED to show that the board is being clocked. If the LED is attached to the last board in the string, it will give you some assurance that the board interconnections are correct. The LED can be attached to the A11 pin on the right rear area of the board. (External channel activity LEDs can be connected to the A12 and A13 pins as well.)

To enable multi-TICC operation, use configuration item [G] to set the master board (typically the first in the chain) as [H]ost, and to set the other boards as [C]lient.

Optionally, use the [I] item to change each channel ID to any single printable ASCII character. Since the master board by default has channels A and B, it's sensible to set the second board to "C D", the third to "E F" and so on.

Finally, if desired use the PROP_DELAY or FUDGE0 items to set the offset for each channel in picoseconds.

Note: the Arduino serial monitor program is a bit funky when it comes to data input; you'll probably have better luck if you use a "real" serial terminal program to set configuration parameters.

9 version 20170309.1

Data Interface and Host Processor

If you have 4 TICCs configured in master/slave mode, you also have 4 USB ports carrying serial data. It would be nice to consolidate those data streams, and maybe even make them available via Ethernet. A Raspberry Pi is an excellent tool to provide this function – its four USB ports are a perfect match for a four-board multi-TICC configuration. I've written a simple Python TCP server program to serve as a data aggregator spitting all the TICC data out over a telnet connection, as well as some other tools. All are available under the multi-ticc/rpi_files directory of the TICC GitHub repository.

In order to keep this document reasonably short, and also because the code is still being revised, I won't go into a lot of details, but here are the basics on the main program as well as some other programs that make managing the system easier. At some point when things have had some time to be debugged, I will create a ready-to-run SD card image so all you'll need to do is plug that in.

Described in a text file located in the rpi_files directory is a description of software prerequisites that need to be installed to use the software, as well as some other setup information that describes how to persistently assign serial port names (dev/ttyTICC0, dev/ttyTICC1, dev/dev/ttyTICC2, and dev/ttyTICC3) to the four devices. Those names are used in all the programs described here, and the assumption below is that the system contains four TICC units. All the programs are installed in the /home/pi/ directory on the Raspberry Pi.

`multi-ticc_server.py` – if run with no options, this program will look for four TICC units connected to the Raspberry Pi USB ports. It outputs several streams of data on different TCP ports, by default:

port 9190: Outputs multiplexed data from all active TICC channels, unsorted
port 9191: Outputs multiplexed data from all active TICC channels, sorted by timestamp
port 9192: Outputs data from chA
port 9193: Outputs data from chB
port 9194: Outputs data from chC
port 9195: Outputs data from chD
port 9196: Outputs data from chE
port 9197: Outputs data from chF
port 9198: Outputs data from chG
port 9199: Outputs data from chH

There are many tools you can use to connect to the server from another machine and access the data. The easiest is the telnet command:

```
telnet server.ip.address 9190
```

To make it easier to log the data to a file, you can use the Linux netcat program:

```
nc server.ip.address 9192 > datafile.dat
```

`multi-ticc_server.py` can be left running on a console and will show startup and connection status. Unfortunately, at this point the program can only support one client connection per data stream, but I'm hoping to add multi-connection support.

Here are some other utility programs that make managing the multi-TICC easier:

`multi-ticc_reset.py` will momentarily send Raspberry Pi GPIO pin 21 high, which if the level translator circuit described above is installed, will cause the TICCs to reset. Run this after starting the Raspberry Pi and before attempting to connect to the TICCs; on startup the TICCs normally freeze and kicking them with this program will cause a clean restart with all boards synchronized. NOTE: multi-ticc_server.py will perform a reset when it starts, so you should not ordinarily need to use this program.

`multi-ticc_updater.py` will sequentially update all connected TICCs with a new firmware file in hex format. Supply the file path and name as a command-line argument. (The program assumes all the TICCs have port names set to `/dev/ttyTICCx` as described above.)

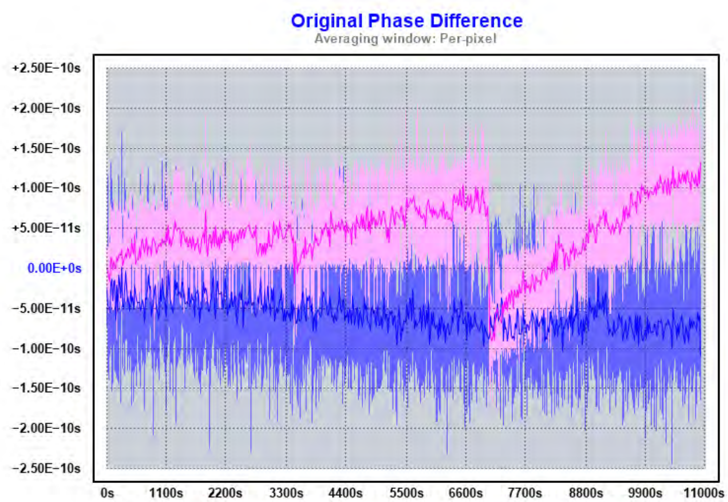
`miniterm.sh` is a shell script that will launch a simple terminal program allowing you to communicate with one TICC. Just give the TICC serial port name as a command line argument, such as `./miniterm.sh /dev/ttyTICC0`.

`multi-ticc_server.service` is a file that can be placed in `/lib/systemd/system/` to cause the server to automatically start at runtime, and also to restart if it fails. After putting it in the directory, run `sudo sysctl enable multi-ticc_server`.

multi-TICC Performance

Performance tests on the multi-TICC indicate that very little if any performance is lost compared to a stand-alone TICC, and I've collected over 5 million samples on the prototype unit without any apparent glitches. However, those tests have also shown that when you are trying to match 4 devices and 8 ports to picosecond levels, there are a lot of factors that make life very interesting.

The main thing learned during testing is that using a simple daisy-chain (short coax jumper cables with "tee" connectors on each TICC reference input) for the 10 MHz clock feed to the boards results in strange behavior. Either one or more of the boards shows higher or lower jitter than the others, or alternatively phase glitches can be introduced on one or more channels. Here are two examples from early testing.



Origin	Slope (sec/sec)
−3.49E−11	−4.09E−15
+2.36E−11	+3.71E−15

Trace	Notes	Input Freq	Sample Interval	Acquired
chA (Unsaved)	Noise Floor via HP 5071A	1 Hz	1 s	10910 pts
chG (Unsaved)	Noise Floor via HP 5071A	1 Hz	1 s	10910 pts

Tau	Sigma(Tau)
1s	1.26E−10
2s	6.26E−11
4s	3.07E−11
8s	1.58E−11
10s	1.27E−11
20s	6.24E−12
40s	3.13E−12
80s	1.63E−12
100s	1.29E−12
200s	6.42E−13
400s	3.35E−13

Trace	Notes	Input Freq	Sample Interval	Acquired
chA (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts
chB (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts
chC (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts
chD (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts
chE (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts
chF (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts
chG (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts
chH (Unsaved)	Residual Noise	1 Hz	1 s	3160 pts

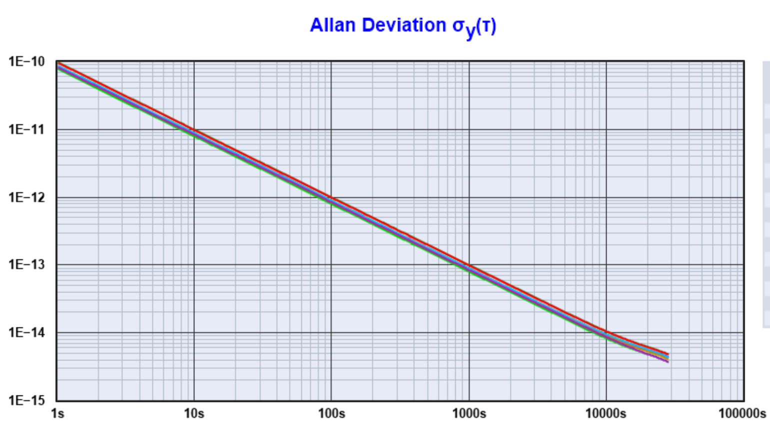
It is highly likely that the combination of multiple coax cables, “tee” connectors, and mismatch of the unterminated TICC clock input circuit caused reflections that translated into jitter at one or more of the 10 MHz reference inputs. Driving each TICC via its own cable from a passive splitter or distribution amplifier (remembering to install the termination jumper on each TICC, which ensures a good impedance match) appears to avoid the problem.

All the results shown below were taken with this clock configuration. I suspect further testing to determine optimal 10 MHz input signal levels will be worthwhile and might produce slightly better results.

The final test configuration used an HP 5071A Cesium standard with high-performance tube to provide both 10 MHz and 1 PPS signals. The 10 MHz output from the 5071A went to a TADD-1 distribution amplifier, four of whose output channels were fed to one TICC 10 MHz reference input. The TADD-1 signal output was about 10 dBm..

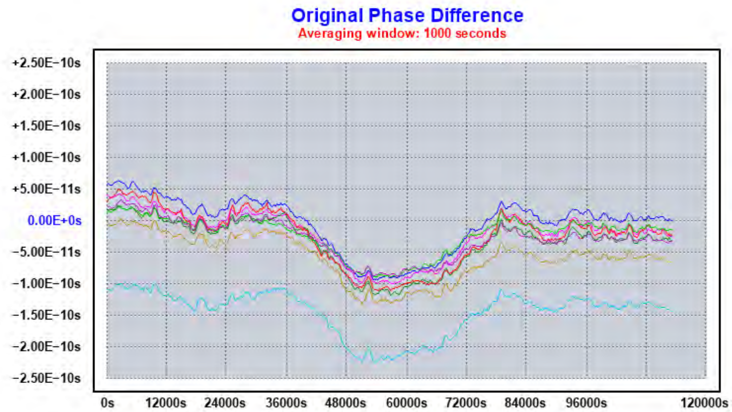
The 5071A PPS signal was fed into a 3dB attenuator and then to the multi-TICC inputs via a daisy-chain of cables using BNC tee connectors. At the end of the chain was a 50 ohm termination. The jumpers between channels each added about 2.4 nanoseconds delay. (While this daisy-chain could also result in reflection-induced jitter, none was noted in testing.)

With all that said, here are results from a test that collected over 125,000 samples from each of the 8 multi-TICC channels, starting with plots showing the ADEV of each channel and the phase record (ignore the apparent offset of channel E in the phase record; that’s a plotting anomaly).



Tau	Sigma(Tau)
1s	8.40E−11
2s	4.20E−11
4s	2.10E−11
8s	1.05E−11
10s	8.41E−12
20s	4.23E−12
40s	2.12E−12
80s	1.06E−12
100s	8.44E−13
200s	4.24E−13
400s	2.12E−13
800s	1.07E−13
1000s	8.48E−14
2000s	4.28E−14
4000s	2.13E−14

Trace	Notes	Input Freq	Sample Interval	Acquired
chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chB	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chC	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chD	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chE	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chF	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chG	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chH	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts

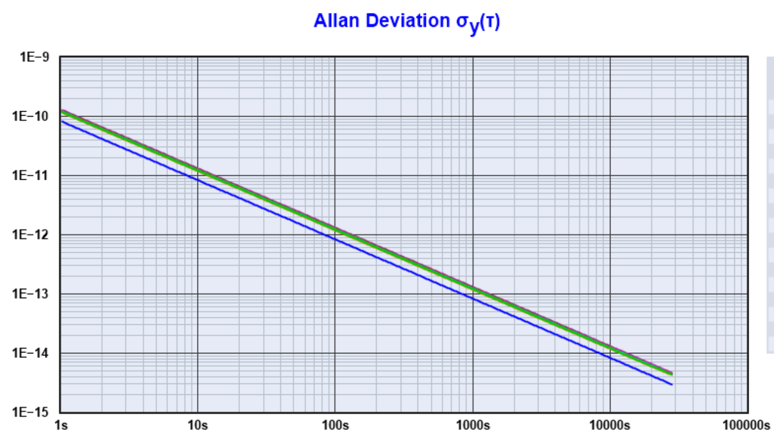


Origin	Slope (sec/sec)
+1.46E-11	-3.54E-16
-1.30E-12	-3.45E-16
-1.33E-11	-3.71E-16
-2.20E-13	-4.37E-16
-1.34E-10	-1.98E-16
-3.03E-11	-4.90E-16
-1.38E-11	-1.47E-16
-4.76E-12	-4.23E-16

Trace	Notes	Input Freq	Sample Interval	Acquired
chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chB	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chC	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chD	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chE	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chF	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chG	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chH	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts

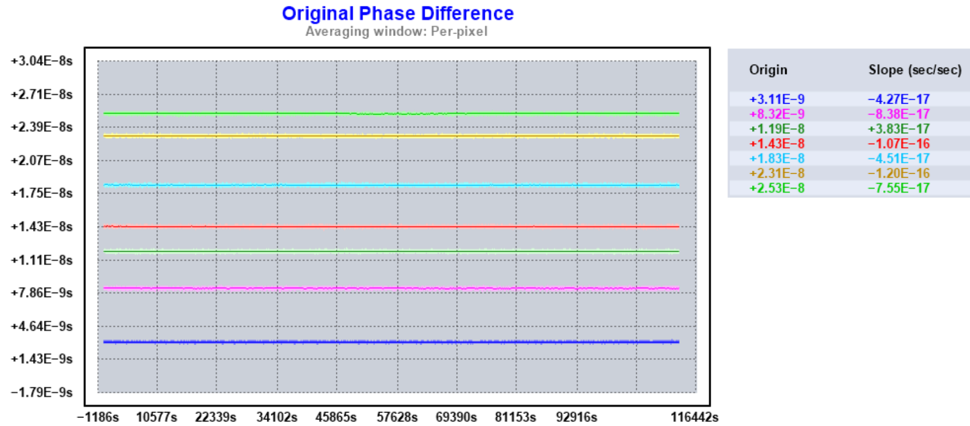
The ADEV readings are tightly grouped at just under $1x10^{-10}$, slightly worse than typical single-TICC performance around $8x10^{-11}$. Note the dip in the middle of the phase chart. My best guess is that this was caused by overnight temperature changes. Though it looks significant here, remember that the change is only about 100 picoseconds over about 40,000 seconds!

In some situations the multi-TICC is used in time interval rather than timestamp mode. In this case, channel A will typically be the START input and channels B through H the STOP inputs. The time interval is (chX – chA). The following plots show the result of that operation.



Tau	Sigma(Tau)
1s	1.22E-10
2s	6.10E-11
4s	3.06E-11
8s	1.52E-11
10s	1.22E-11
20s	6.15E-12
40s	3.06E-12
80s	1.53E-12
100s	1.22E-12
200s	6.12E-13
400s	3.06E-13
800s	1.53E-13
1000s	1.22E-13
2000s	6.14E-14
4000s	3.05E-14
8000s	1.52E-14
10000s	1.22E-14

Trace	Notes	Input Freq	Sample Interval	Acquired
chB - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chC - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chD - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chE - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chF - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chG - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chH - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts



Trace	Notes	Input Freq	Sample Interval	Acquired
chB - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chC - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chD - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chE - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chF - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chG - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts
chH - chA	Noise Floor via HP 5071A	1 Hz	1 s	112997 pts

You can see that the phase change in the middle of the run is no longer visible. That is because it was common mode to each channel (i.e. a change in the phase of the HP 5071A 10 MHz vs. PPS outputs that affected both the measurement channel and channel A equally, so subtracting channel A removes the phase change). You can also see the increasing phase delay of each channel as the cable length (and delay) between it and channel A grows.

Using Tom Van Baak's command line ADEV and statistics tools, I extracted the following measurements from the run:

	ADEV	Picoseconds				
	@ 1 SEC	MEAN	SDEV	MIN	MAX	RANGE
chA	8.38e-11	149045	65.22	148827	149289	462
chB	8.64e-11	152151	64.50	151892	152438	546
chC	1.00e-10	157357	71.75	157056	157658	602
chD	9.72e-11	160956	72.76	160670	161244	574
chE	9.03e-11	163360	65.22	163121	163624	503
chF	8.56e-11	167386	63.05	167072	167632	560
chG	7.89e-11	172147	54.65	171917	172381	464
chH	8.35e-11	174307	57.62	174042	174525	483

I attempted to work out the channel-to-channel delay from these measurements:

Mean Phase Difference (picoseconds)

	Raw Values		**Minus Cable Delay**	
	chA	**Prev CH**	**chA**	**Prev CH**
chB	3106	3106	706	706
chC	8312	5206	3502	2806
chD	11911	4599	4711	2199
chE	14315	2404	4715	4
chF	18341	4026	6341	1626
chG	23102	4761	8702	2361
chH	25262	2160	8462	-240

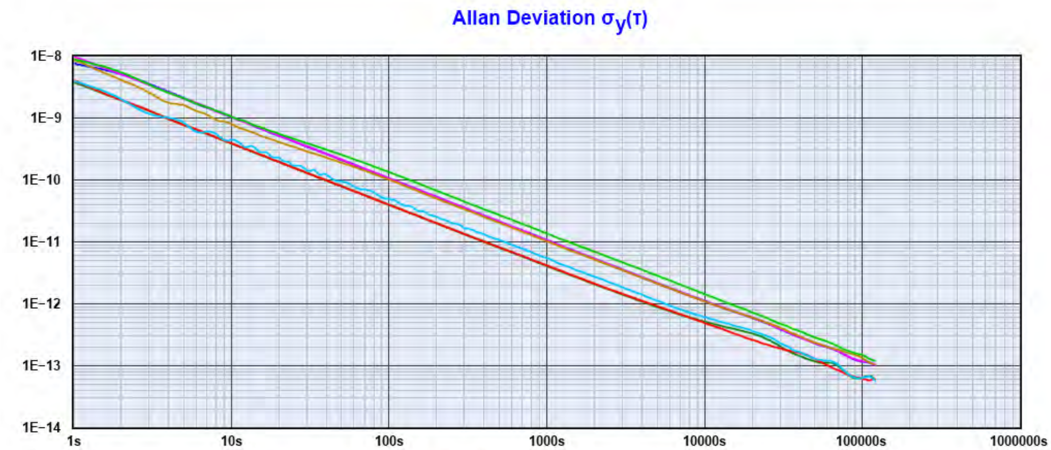
N = 112,970

Calculations from TVB ADEV1 and STAT

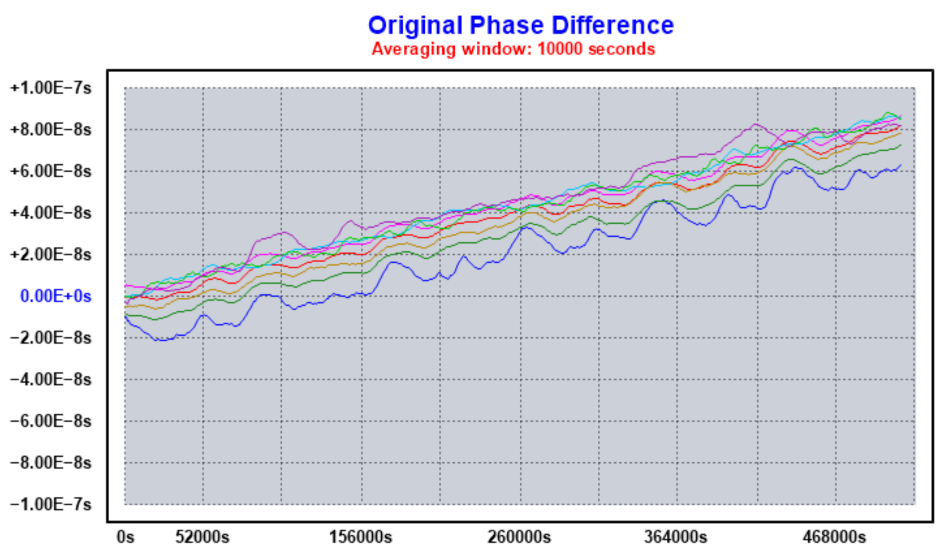
In the table above, the "`chA`" columns show the subtraction of channel A from each succeeding channel. The "`Prev CH`" columns show the subtraction of the prior channel (e.g., the "chC" row shows (chC – chB). The "`Raw Values`" are simple subtraction of the mean values reported above. The "`Minus Cable Delay`" values subtract the delay of each of the jumper cables between inputs, which were 50 cm pieces of RG-316D with a measured delay of about 2.4 nanoseconds. In an ideal world, the `Prev Ch Minus Cable Delay` values should be the same for each channel. It's easy to see that's not the case here.

One would expect the two channels on each board (e.g., chA and chB) to have only a small delay because both are fed from the same clock signals within the board. The delay between boards would be greater because there is an increasing delay in the propagation of the 10 MHZ and 10 kHz clock signals from board to board. A third consideration is how close to simultaneously the boards boot up. Finally, the phase of the 16.67 MHz CPU clock (which is free-running) may affect the initial starting condition. More experiments may determine what causes the range of delays seen here, and how to minimize the range.

To finish things up, the following page shows an example of "real world" data capture from a multi-TICC with the PPS from eight GPS units connected to channels A through H:



Trace	Notes	Sample Interval	Acquired	Instrument
NEO-M8P	vs HP 5071A	1 s	510244 pts	multi-TICC
NEO-M8T	vs HP 5071A	1 s	510244 pts	multi-TICC
ZED-F9P	vs HP 5071A	1 s	510244 pts	multi-TICC
ZED-F9T	vs HP 5071A	1 s	510244 pts	multi-TICC
LEA-M8F	vs HP 5071A	1 s	510244 pts	multi-TICC
NEO-M9N	vs HP 5071A	1 s	510244 pts	multi-TICC
NEO-M8N	vs HP 5071A	1 s	510244 pts	multi-TICC



Origin	Slope (sec/sec)
−2.09E−8	+1.66E−13
+2.38E−9	+1.61E−13
−1.14E−8	+1.61E−13
−2.15E−9	+1.61E−13
+2.83E−9	+1.61E−13
−6.82E−9	+1.66E−13
+2.09E−9	+1.62E−13
+5.23E−9	+1.62E−13

Trace	Notes	Sample Interval	Acquired	Instrument
CNS-II	vs HP 5071A	1 s	510244 pts	multi-TICC
NEO-M8P	vs HP 5071A	1 s	510244 pts	multi-TICC
NEO-M8T	vs HP 5071A	1 s	510244 pts	multi-TICC
ZED-F9P	vs HP 5071A	1 s	510244 pts	multi-TICC
ZED-F9T	vs HP 5071A	1 s	510244 pts	multi-TICC
LEA-M8F	vs HP 5071A	1 s	510244 pts	multi-TICC
NEO-M8N	vs HP 5071A	1 s	510244 pts	multi-TICC
NEO-M9N	vs HP 5071A	1 s	510244 pts	multi-TICC

TAPR at Hamcation

For the second year in a row, TAPR had a booth at Hamcation in Orlando, Florida. Former TAPR Director Mel Whitten, K0PFX, was in attendance and took the following photos of the action at our booth.

###

TAPR Wear Available

Personalized Land's End clothing with the TAPR logo and your name and call sign are now available from the TAPR Store at

http://business.landsend.com/store/tapr/

Select from the Men's or Women's catalog. (To make shopping easier, there are "TAPR Recommended Shirts" in the Men's catalog including two styles of polo shirts, each available with or without pockets.)

The logo is available in three colors -- red, blue, and white. The name/call sign monogram thread will match the logo color. (We recommend that you use the white logo with dark colored shirts.)

Prices are very reasonable, for example, after adding the logo and monogram, a mesh pocket shirt is $39.85 plus shipping and sales tax where applicable. Processing time is 5-7 days.

###

Experimenting with WSPR using Raspberry Pi 2017 2019 2020

By Ian Saturley, M0IJS

The following 21 pages contain a pdf of M0IJS's presentation.

###.

Rugby Amateur Transmitting Society [RATS]

Experimenting with WSPR using Raspberry Pi

2017 2019 2020

Ian M0IJS

WSPR & Raspberry Pi

- WSPR "Whisper"

  Weak Signal Propagation Reporter

  https://en.wikipedia.org/wiki/WSPR_(amateur_radio_software)

- Raspberry Pi

  A very convenient computing platform

A summary overview showing how to get running with WSPR using a Raspberry Pi

Requirements:

- Raspberry Pi. We're using model B Version3
- SD card with Raspbian image
- TAPR / WSPR board

  https://www.tapr.org/kits_20M-wspr-pi.html
- Software project

  https://github.com/JamesP6000/WsprryPi/

Bill of Materials – May 2017

RS Part	Description	Cost
896-8660	Raspberry Pi Model B	32.99
909-8132	Official Raspberry Pi 3 B Case	5.50
909-8126	Raspberry Pi official PSU	6.50
Amazon	SanDisk 16GB micro SD	6.50
TAPR	20M WSPR shield	25.00
	Total	£76.49

Rugby Amateur Transmitting Society [RATS]

Prepare the system

Obtain the latest Raspbian image from raspberrypi.org. Write this to an SD card and place a blank file in the root called 'ssh'. This will enable headless operation using SSH. Since November 2016 SSHd has been disabled by default. On boot, the platform looks for the file 'ssh' and starts the SSH server if found.

https://downloads.raspberrypi.org/raspbian_latest

RASPBIAN JESSIE WITH PIXEL
Image with PIXEL desktop based on Debian Jessie
Version:April 2017
Release date:2017-04-10
Kernel version:4.4

Either login remotely or connect a keyboard and screen locally.

Update the system & Install project

Get upto date:

sudo apt-get update && sudo apt-get dist-upgrade

An indication of requirements:
16 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 90.0 MB of archives.
After this operation, 4,517 kB of additional disk space will be used.

The project is hosted on github and we 'clone' the files from this as follows:

cd ~
git clone https://github.com/JamesP6000/WsprryPi.git

Now make the program:

cd WsprryPi
make

Update the system & Install project

Get upto date:
sudo apt-get update && sudo apt-get dist-upgrade

An indication of requirements:
16 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 90.0 MB of archives.
After this operation, 4,517 kB of additional disk space will be used.

Do not make this step. We all normally do, but on reboot the WSPR application will not work. This issue will likely get resolved quickly.
Works fine with a stock Raspbian 2017-04-10

The project is hosted on github and we 'clone' the files from this as follows:

```
cd ~
git clone https://github.com/JamesP6000/WsprryPi.git
```

Now make the program:

```
cd WsprryPi
make
```

Try it out!

Hopefully the files compile cleanly and we can now try it out. Assemble hardware and connect a long wire & ground to the antenna connection.

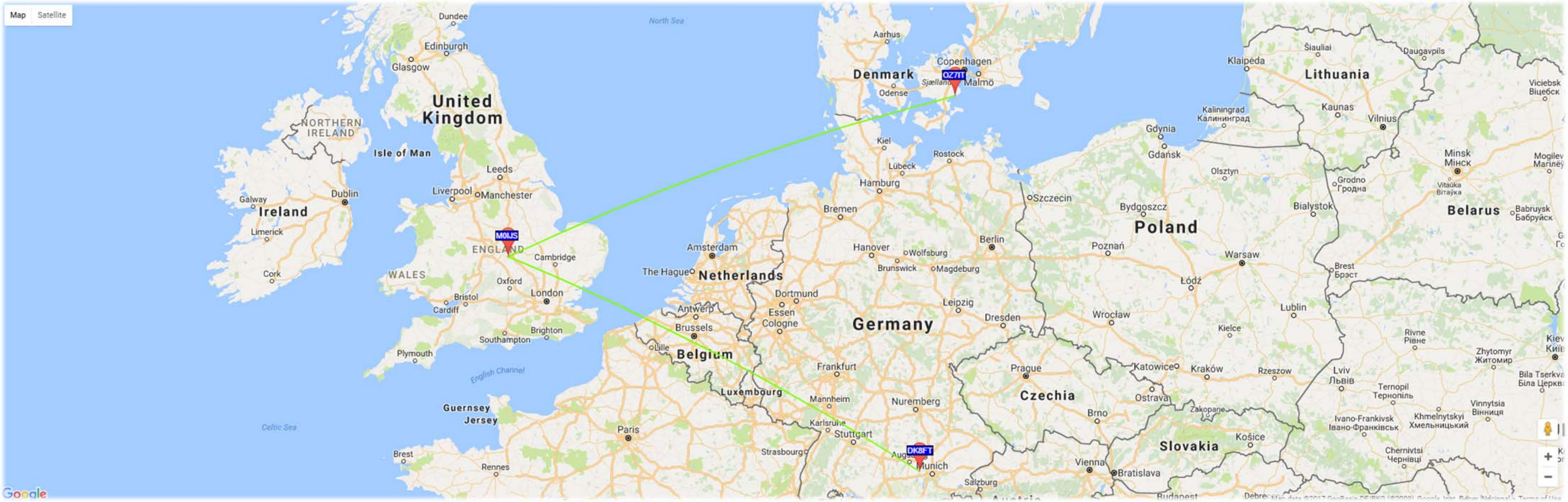
mycall = callsign
myloc = locator
Using these parameters, 10dBm (10mW) 20Mtr band:

pi@raspberrypi:~/WsprryPi $ ***sudo ./wspr -r -o -s mycall myloc 10 20m***
Detected Raspberry Pi version 2/3
WSPR packet contents:
Callsign: M0IJS
Locator: IO92II
Power: 10 dBm
Requested TX frequencies:
14.097100 MHz
Extra options:
NTP will be used to periodically calibrate the transmission frequency
Transmissions will continue forever until stopped with CTRL-C
A small random frequency offset will be added to all transmissions

WSPR Transmitting…

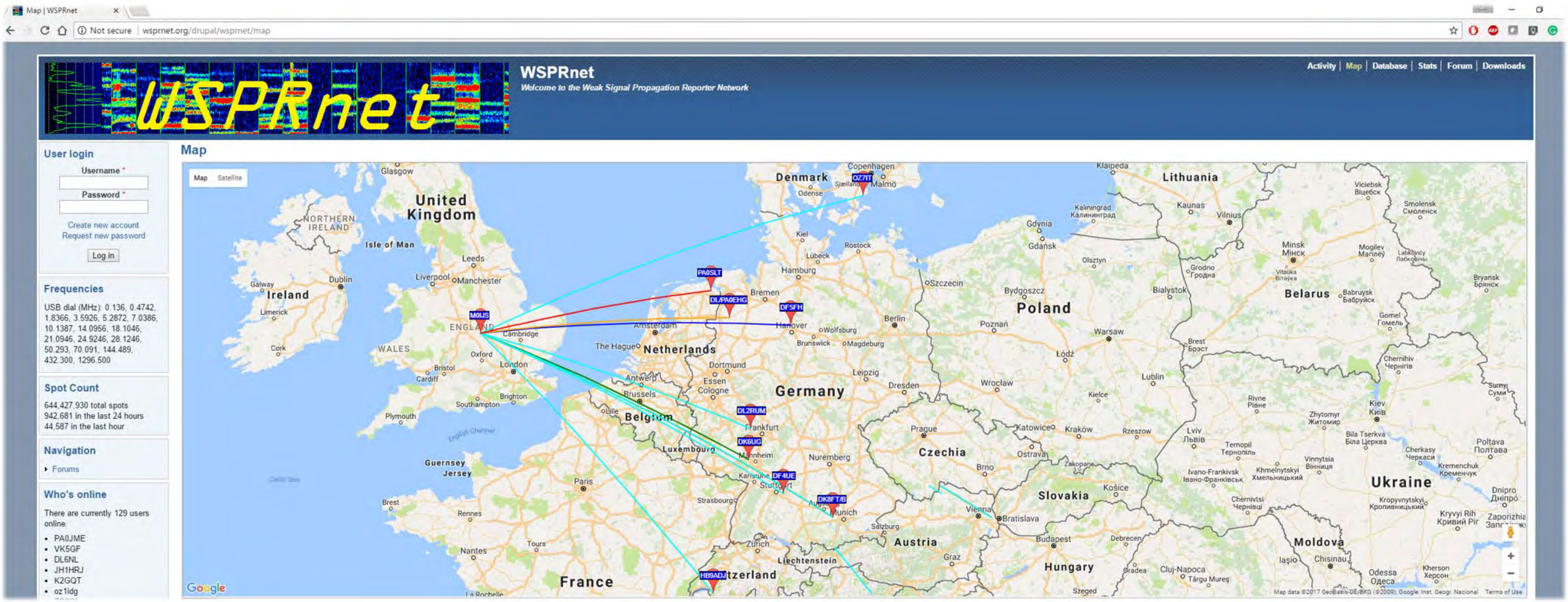
```
Ready to transmit (setup complete)...
Desired center frequency for WSPR transmission: 14.097146 MHz
Waiting for next WSPR transmission window...
Obtained new ppm value: -5.82245
TX started at: UTC 2017-05-03 13:08:01.002
TX ended at: UTC 2017-05-03 13:09:51.685 (110.683 s)
Desired center frequency for WSPR transmission: 14.097042 MHz
Waiting for next WSPR transmission window...
Obtained new ppm value: -6.50426
TX started at: UTC 2017-05-03 13:10:01.003
TX ended at: UTC 2017-05-03 13:11:51.685 (110.682 s)
Desired center frequency for WSPR transmission: 14.097056 MHz
Waiting for next WSPR transmission window...
TX started at: UTC 2017-05-03 13:12:01.001
^CExiting with error; caught signal: 2
pi@raspberrypi:~/WsprryPi $
```


Resultant Map – May 2017
First transmission



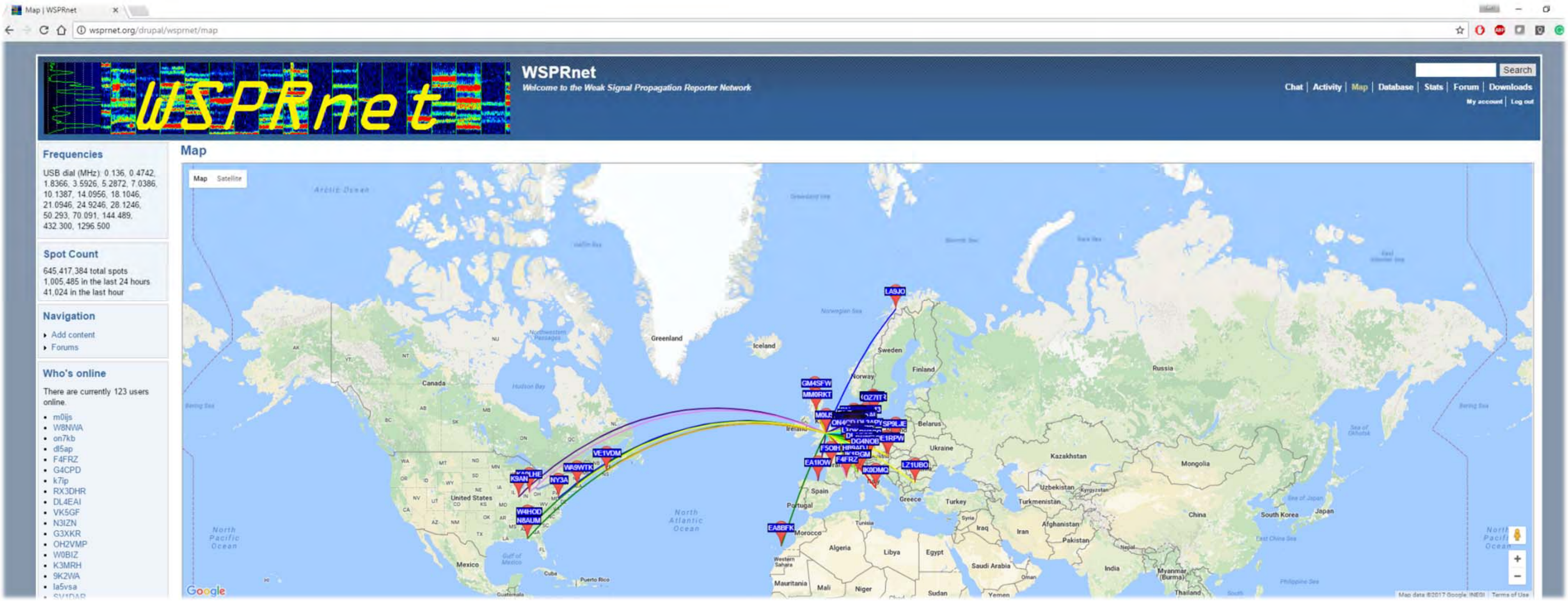
http://wsprnet.org/drupal/wsprnet/map

Resultant Map – May 2017
After 10 minutes, 100mW Tx



http://wsprnet.org/drupal/wsprnet/map

Resultant Map – May 2017
After 24hrs, 100mW Tx



http://wsprnet.org/drupal/wsprnet/map

Dataset – May 2017

- 500 Spots recorded
- 66 Unique spots
- Best DX: N8AUM, 7005KM

WSPRnet

Welcome to the Weak Signal Propagation Reporter Network

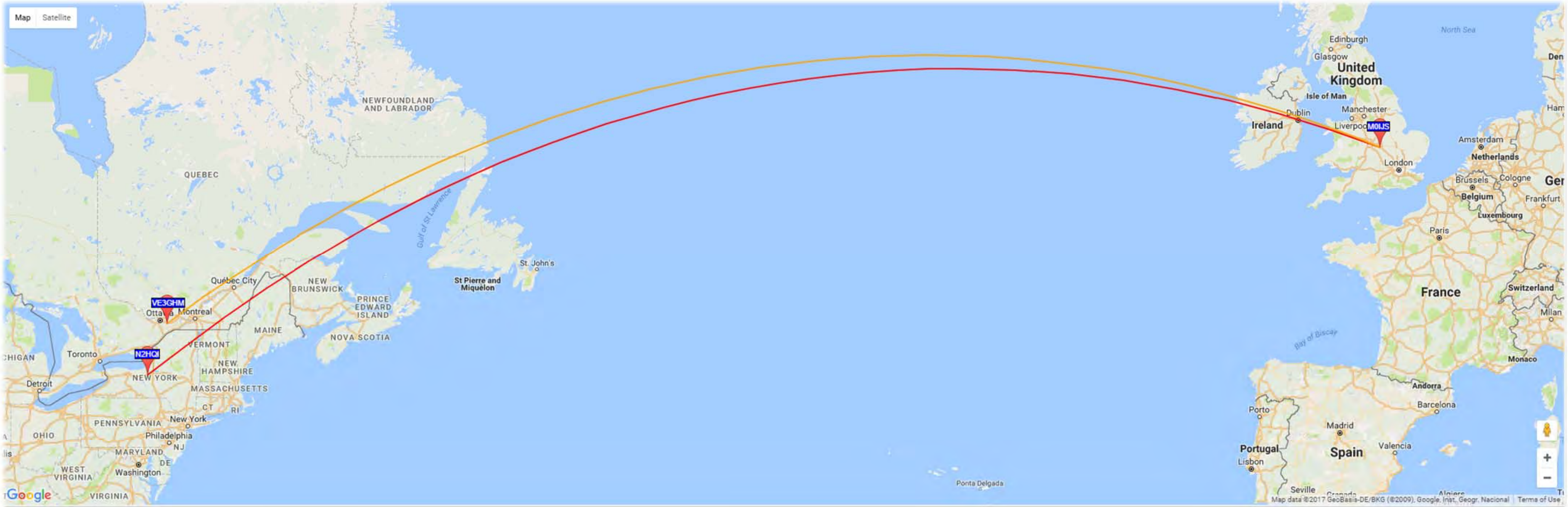
Database

Specify query parameters

66 spots:

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az
2017-05-03 21:42	M0IJS	14.097057	-22	0	IO92ii	0.1	N8AUM	EM70em	7005	286
2017-05-04 11:14	M0IJS	14.097073	-28	0	IO92ii	0.1	W4HOD	EM72eo	6836	287
2017-05-03 18:52	M0IJS	14.097157	-27	0	IO92ii	0.1	K9AN	EN50wc	6405	295
2017-05-03 22:28	M0IJS	14.097156	-25	0	IO92ii	0.1	KA9LHE	EN70lx	6148	294
2017-05-03 18:06	M0IJS	14.097030	-32	0	IO92ii	0.1	NY3A	FM19	5751	288
2017-05-03 21:10	M0IJS	14.097117	-24	0	IO92ii	0.1	WA9WTK	FN42fk	5186	287
2017-05-03 21:42	M0IJS	14.097075	-20	0	IO92ii	0.1	VE1VDM	FN85ij	4457	286
2017-05-04 06:26	M0IJS	14.097158	-23	0	IO92ii	0.1	EA8BFK	IL38bo	2838	206
2017-05-04 10:06	M0IJS	14.097057	-27	0	IO92ii	0.1	LA9JO	JP99gb	2126	22
2017-05-04 07:10	M0IJS	14.097088	-17	-1	IO92ii	0.1	LZ1UBO	KN12	2114	112

Example Spots - 20170517



1Hr dataset

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az
2017-05-26 23:12	M0IJS	14.097004	-15	0	IO92ii	0.1	N2HQI	FN13sa	5461	291
2017-05-26 23:10	M0IJS	14.097000	-24	0	IO92ii	0.1	N2HQI	FN13sa	5461	291
2017-05-26 23:10	M0IJS	14.097061	-24	0	IO92ii	0.1	VE3GHM	FN25ig	5235	293
2017-05-26 22:56	M0IJS	14.097045	-21	0	IO92ii	0.1	N2HQI	FN13sa	5461	291
2017-05-26 22:52	M0IJS	14.097097	-24	-1	IO92ii	0.1	N2HQI	FN13sa	5461	291
2017-05-26 22:50	M0IJS	14.097032	-1	0	IO92ii	0.1	TF4M	HP85fp	1930	328
2017-05-26 22:38	M0IJS	14.097116	-26	0	IO92ii	0.1	N2HQI	FN13sa	5461	291

Revisiting WSPR using Raspberry Pi – July 2019

- App no longer compiles

  Solution: Build on an older version of Raspbian

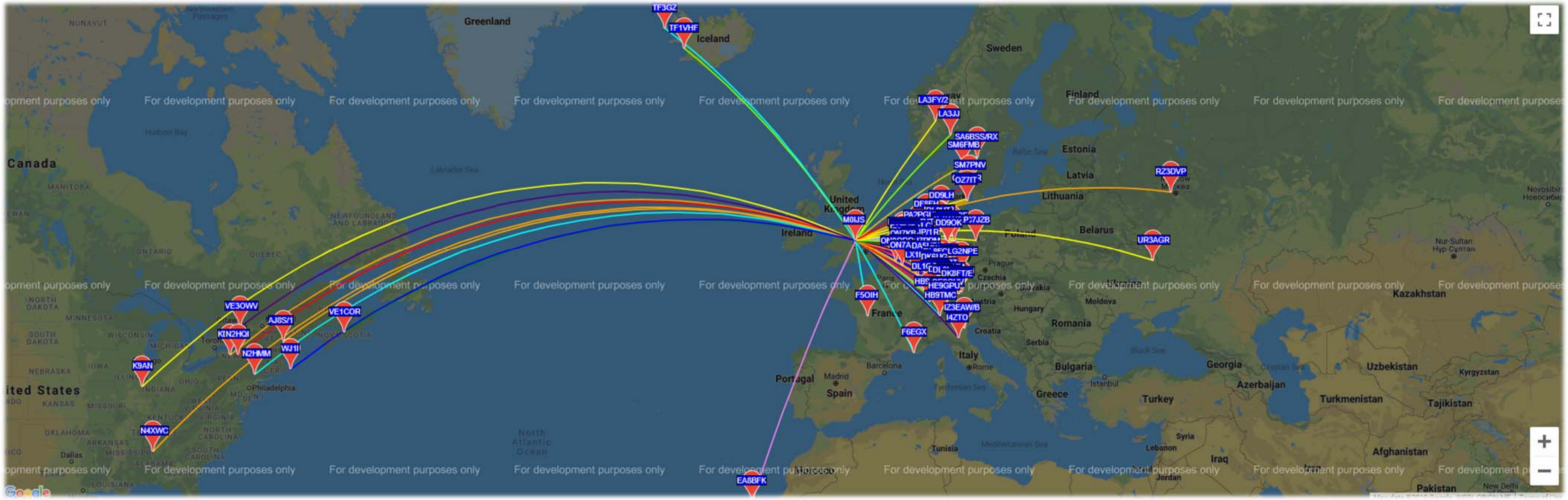
- App crashes

  Solution: Run the app within a loop

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ cat wspr_loop.sh
#!/bin/bash
loop_ctr=0
while true
do
 ((loop_ctr++))
 echo "Script loop: "$loop_ctr
 sudo ./wspr -s -r -o mycall myloc 20 20m
done
pi@raspberrypi:~ $
```

Example Spots - 20190720

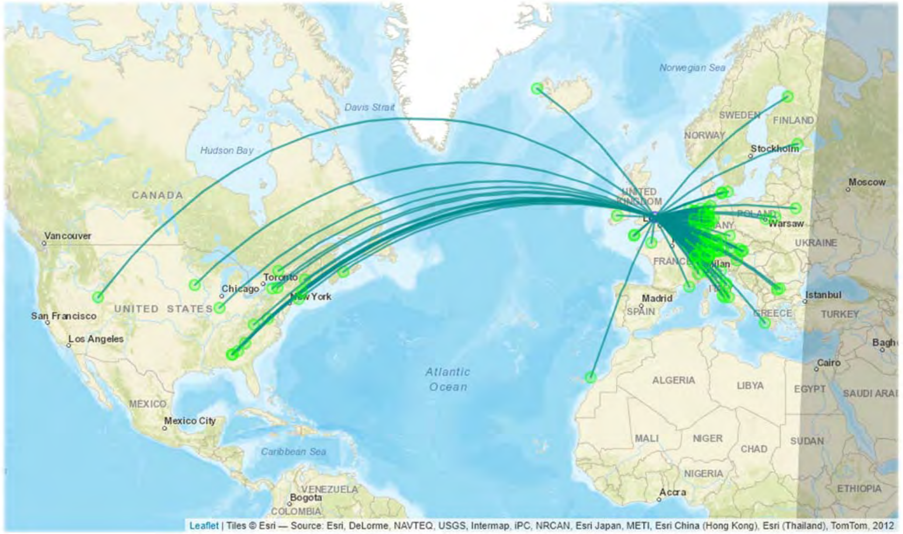


Best dx

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az
2019-07-19 23:36	M0IJS	14.097036	-28	0	IO92ii	0.1	N4XWC	EM63nt	6820	289
2019-07-20 00:12	M0IJS	14.097079	-26	0	IO92ii	0.1	K9AN	EN50wc	6405	295
2019-07-19 23:58	M0IJS	14.097174	-28	0	IO92ii	0.1	KD2OM	FN12gx	5528	291
2019-07-19 22:46	M0IJS	14.097158	-25	0	IO92ii	0.1	KD2OM	FN12gx	5528	291
2019-07-19 22:58	M0IJS	14.097131	-23	0	IO92ii	0.1	KD2OM	FN12gx	5528	291
2019-07-19 22:44	M0IJS	14.097064	-25	0	IO92ii	0.1	KD2OM	FN12gx	5528	291
2019-07-19 23:56	M0IJS	14.097040	-25	0	IO92ii	0.1	KD2OM	FN12gx	5528	291

Example Spots - 20200325

~28hrs running into long wire antenna with 9:1 un-un

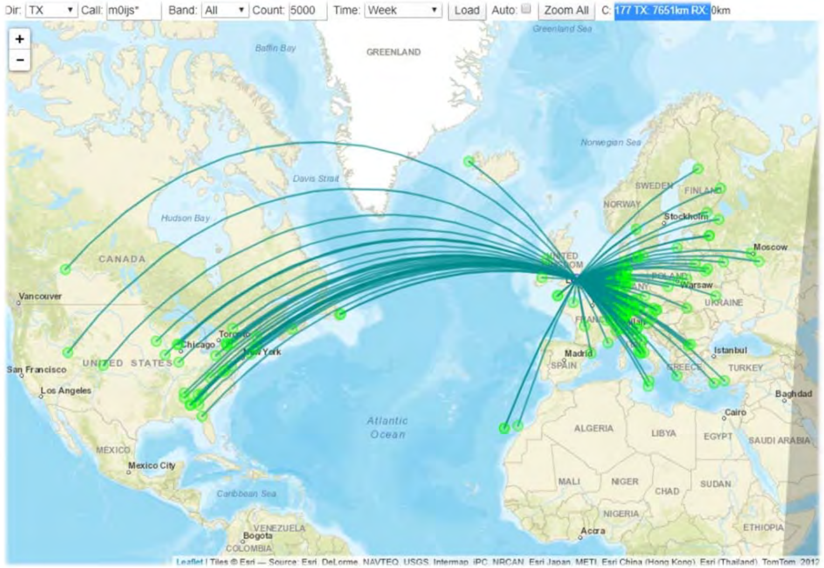


Best dx

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az	# Spots
2020-03-24 17:16	M0IJS	14.097061	-25	0	IO92	0.1	KA7OEI-1	DN31uo	7651	312	10
2020-03-24 17:16	M0IJS	14.097062	-24	0	IO92	0.1	W4HOD	EM72eo	6848	288	63
2020-03-25 13:02	M0IJS	14.097048	-27	0	IO92	0.1	K4IQJ	EM72gn	6841	287	6
2020-03-25 11:26	M0IJS	14.097143	-25	0	IO92	0.1	K4COD	EM73sc	6733	287	3
2020-03-24 18:24	M0IJS	14.097137	-25	0	IO92	0.1	N4GYN	EM84	6532	287	1
2020-03-25 15:48	M0IJS	14.097141	-24	0	IO92	0.1	N0UR	EN33	6446	301	3
2020-03-25 12:08	M0IJS	14.097078	-29	0	IO92	0.1	K9AN	EN50wc	6414	295	14
2020-03-24 18:36	M0IJS	14.097047	-27	0	IO92	0.1	KC8COM	EM97kj	6182	288	7
2020-03-25 11:12	M0IJS	14.097031	-26	0	IO92	0.1	K4RCG	FM08si	5938	288	14
2020-03-24 16:20	M0IJS	14.097045	-24	0	IO92	0.1	KD2OM	FN12gx	5538	291	73

Example Spots - 20200327

~76hrs running into long wire antenna with 9:1 un-un

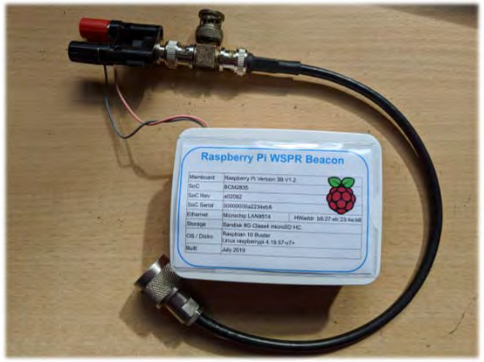


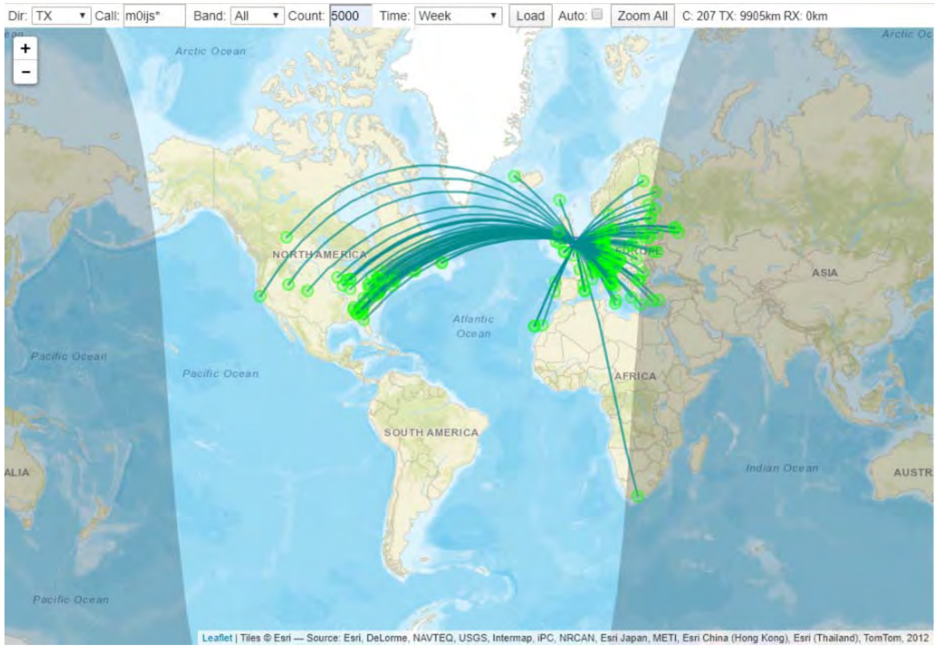
Best dx

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az	# Spots
2020-03-24 17:16	M0IJS	14.097061	-26	0	IO92	0.1	KA7OEI-1	DN31uo	7651	312	16
2020-03-27 15:00	M0IJS	14.097116	-27	0	IO92	0.1	WD0E	DM79qm	7425	305	1
2020-03-26 16:10	M0IJS	14.097029	-25	0	IO92	0.1	W4VS	EM80	6852	284	1
2020-03-24 12:16	M0IJS	14.097189	-24	0	IO92	0.1	W4HOD	EM72eo	6848	288	226
2020-03-24 13:38	M0IJS	14.097070	-26	0	IO92	0.1	K4IQJ	EM72gn	6841	287	36
2020-03-26 20:06	M0IJS	14.097069	-24	0	IO92	0.1	N4XWC	EM63nt	6831	289	5
2020-03-26 11:48	M0IJS	14.097121	-27	0	IO92	0.1	W3PM	EM64or	6753	290	1
2020-03-24 13:42	M0IJS	14.097064	-25	0	IO92	0.1	K4COD	EM73sc	6733	287	17
2020-03-25 21:06	M0IJS	14.097066	-26	0	IO92	0.1	WM4B	EM82em	6725	286	6
2020-03-26 15:48	M0IJS	14.097141	-22	-1	IO92	0.1	VE6JY	DO33or	6628	320	19

Example Spots - 20200329

~124hrs running into long wire antenna with 9:1 un-un



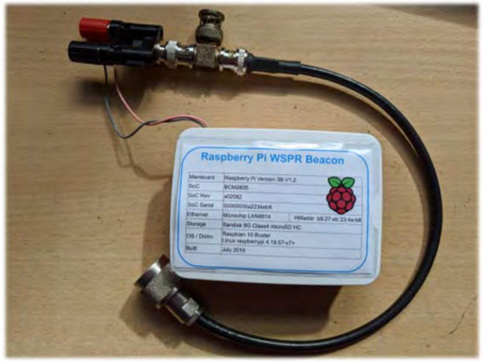


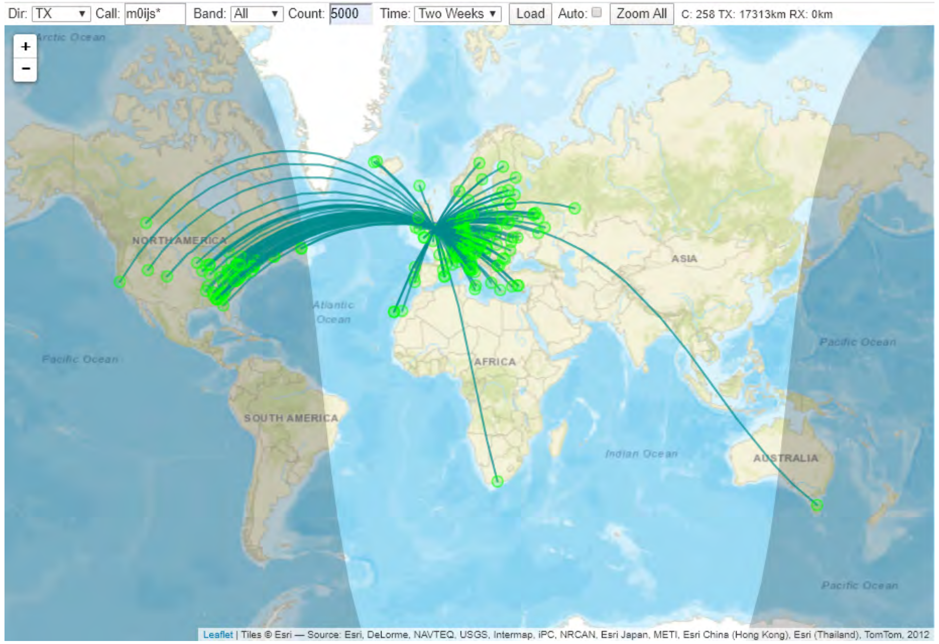
Best dx

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az	# Spots	
2020-03-29 13:20	M0IJS	14.097181	-26	0	IO92	0.1	ZS1LS	KF15nw	9905	160	1	**South Africa!**
2020-03-27 15:34	M0IJS	14.097174	-27	0	IO92	0.1	KPH	CM88mc	8489	317	1	**KPH, San Francisco**
2020-03-24 17:16	M0IJS	14.097061	-26	0	IO92	0.1	KA7OEI-1	DN31uo	7651	312	27	**Utah**
2020-03-27 15:00	M0IJS	14.097116	-27	0	IO92	0.1	WD0E	DM79qm	7425	305	1	
2020-03-26 16:10	M0IJS	14.097029	-25	0	IO92	0.1	W4VS	EM80	6852	284	1	**Florida**
2020-03-24 12:16	M0IJS	14.097189	-24	0	IO92	0.1	W4HOD	EM72eo	6848	288	333	
2020-03-24 13:38	M0IJS	14.097070	-25	0	IO92	0.1	K4IQJ	EM72gn	6841	287	53	
2020-03-26 20:06	M0IJS	14.097069	-27	0	IO92	0.1	N4XWC	EM63nt	6831	289	14	
2020-03-26 11:48	M0IJS	14.097121	-27	0	IO92	0.1	W3PM	EM64or	6753	290	1	
2020-03-24 13:42	M0IJS	14.097064	-25	0	IO92	0.1	K4COD	EM73sc	6733	287	27	

Example Spots - 20200404

11days running into long wire antenna with 9:1 un-un



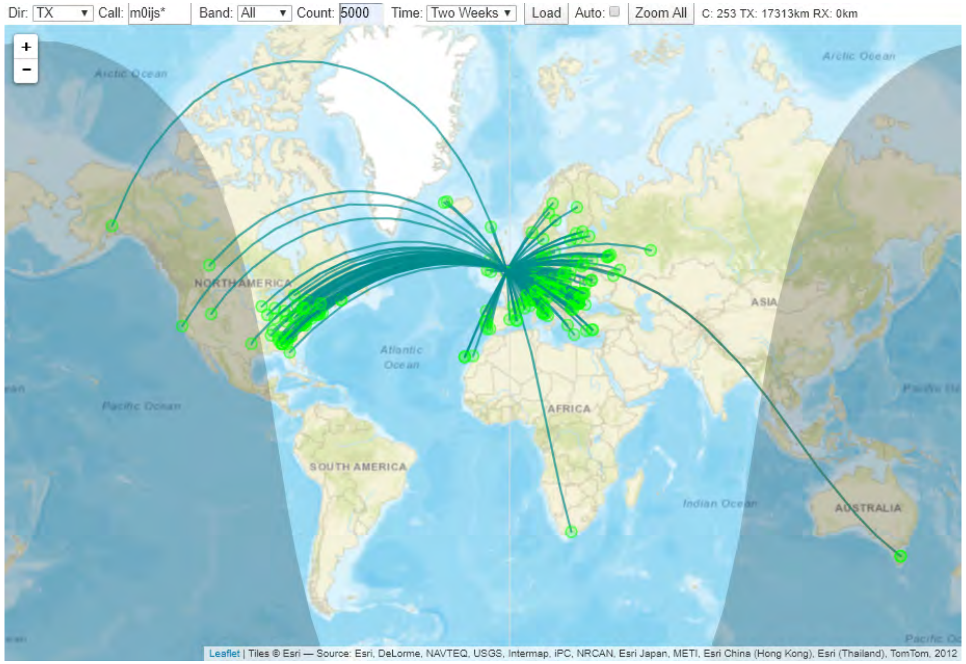


Best dx

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az	# Spots	
2020-04-03 20:48	M0IJS	14.097007	-30	0	IO92	0.1	VK7JJ/2	QE38lr	17313	75	2	Tasmania! (Exeter)
2020-03-29 13:20	M0IJS	14.097181	-26	0	IO92	0.1	ZS1LS	KF15nw	9905	160	1	South Africa (Cape Town)
2020-03-27 15:34	M0IJS	14.097174	-29	0	IO92	0.1	KPH	CM88mc	8489	317	2	KPH, San Francisco
2020-03-24 17:16	M0IJS	14.097061	-26	0	IO92	0.1	KA7OEI-1	DN31uo	7651	312	27	Utah
2020-03-27 15:00	M0IJS	14.097116	-27	0	IO92	0.1	WD0E	DM79qm	7425	305	1	Colorado
2020-04-01 14:52	M0IJS	14.097152	-23	0	IO92	0.1	W5KUB/HAB	EM55db	6910	292	8	Tennessee
2020-03-26 16:10	M0IJS	14.097029	-25	0	IO92	0.1	W4VS	EM80	6852	284	1	Georgia
2020-03-24 12:16	M0IJS	14.097189	-24	0	IO92	0.1	W4HOD	EM72eo	6848	288	453	Alabama
2020-03-24 13:38	M0IJS	14.097070	-26	0	IO92	0.1	K4IQJ	EM72gn	6841	287	69	Alabama
2020-04-03 21:44	M0IJS	14.097205	-26	0	IO92	0.1	K4JOP	EM72go	6838	287	1	Alabama

Example Spots - 20200411

Past 14 days running into long wire antenna with 9:1 un-un



Best dx

Timestamp	Call	MHz	SNR	Drift	Grid	Pwr	Reporter	RGrid	km	az	# Spots	
2020-04-03 20:48	M0IJS	14.097007	-30	0	IO92	0.1	VK7JJ/2	QE38lr	17313	75	2	Tasmania! (Exeter)
2020-04-08 16:32	M0IJS	7.040135	-28	0	IO92	0.1	VK7JJ	QE38lr	17313	75	5	
2020-03-29 13:20	M0IJS	14.097181	-26	0	IO92	0.1	ZS1LS	KF15nw	9905	160	1	South Africa (Cape Town)
2020-03-29 18:46	M0IJS	14.097201	-30	0	IO92	0.1	KPH	CM88mc	8489	317	1	KPH, San Francisco
2020-03-28 15:52	M0IJS	14.097075	-27	0	IO92	0.1	KA7OEI-1	DN31uo	7651	312	22	Utah
2020-04-05 22:14	M0IJS	14.097111	-24	0	IO92	0.1	K5XL	EM12kp	7569	295	2	Texas
2020-04-11 07:22	M0IJS	14.097168	-20	0	IO92	0.1	KL7L	BP51ip	7019	344	2	Alaska
2020-04-01 14:52	M0IJS	14.097152	-24	0	IO92	0.1	W5KUB/HAB	EM55db	6910	292	30	Tennessee
2020-04-05 22:26	M0IJS	14.097021	-29	0	IO92	0.1	WD4AH	EL89rt	6873	283	1	Florida
2020-03-28 12:02	M0IJS	14.097092	-24	0	IO92	0.1	W4HOD	EM72eo	6848	288	447	Alabama
2020-03-31 05:48	M0IJS	14.097129	+8	0	IO92	0.1	G8DLX	IO92ii	22	231	141	Mike G8DLX, Rugby.

Write Here!

Your *PSR* editor is working on the next issue of *PSR* and hopes to find a few good writers, particularly ham radio operators working on the digital side of our hobby, who would like to write about their activities and have them published here in *PSR*.

You don't have to be Hiram Percy Maxim to contribute to *PSR* and you don't have to use *Microsoft Word* to compose your thoughts.

Your *PSR* editor can handle just about any text and graphic format, so don't be afraid to submit whatever you have to wa1lou@tapr.org --- she can handle it!

The deadline for the next issue of *PSR* is October 1, so write early and write often.

###

On the Net

By Mark Thompson, WB9QZB

Facebook

As you may know, TAPR has a Facebook page, www.facebook.com/TAPRDigitalHam.

However, I also created a TAPR Facebook Group, www.facebook.com/groups/TAPRDigital/.

If you have a Facebook account, "Like" the TAPR Facebook page and join the TAPR Facebook Group.

If you join the group click on the Events link and indicate you're Going to the events.

On Twitter, Too

Access the TAPR Twitter account at

www.twitter.com/taprdigital.

Also on YouTube

TAPR now has its own channel on YouTube: the TAPR Digital Videos Channel: www.youtube.com/user/TAPRDigitalVideo.

At this time, there are a slew of videos on our channel including many from the TAPR-ARRL Digital Communications Conference (DCC) that you may view at no cost, so have at it!

###

PSR

#145 Augist 2020, ISSN: 1052–3626

Published by TAPR

Phone +1 972 413 8277

E-mail contact@tapr.org

URL www.tapr.org

Facebook www.facebook.com/TAPRDigitalHam

Twitter www.twitter.com/taprdigital

TAPR Office Hours: Monday to Friday, 9 AM to 5 PM Eastern Time

Submission Guidelines

TAPR is always interested in receiving information and articles for publication. If you have an idea for an article you would like to see, or you or someone you know is doing something that would interest TAPR, please contact the editor (wa1lou@tapr.org) so that your work can be shared with the Amateur Radio community. If you feel uncomfortable or otherwise unable to write an article yourself, please contact the editor for assistance. Preferred format for articles is plain ASCII text (OpenOffice or *Microsoft Word* is acceptable). Preferred graphic formats are PS/EPS/TIFF (diagrams, black and white photographs), or TIFF/JPEG/GIF (color photographs). Please submit graphics at a minimum of 300 DPI.

Production / Distribution

PSR is exported as Adobe Acrobat and distributed electronically at www.tapr.org

PSR Editor:

Stana Horzepa, WA1LOU

E-mail wa1lou@tapr.org

TAPR is a not–for–profit scientific research and development corporation [Section 501(c)(3) of the US tax code]. Contributions are deductible to the extent allowed by US tax laws. TAPR is chartered in the State of Arizona for the purpose of designing and developing new systems for digital radio communication in the Amateur Radio Service, and for disseminating information required, during, and obtained from such research.

Membership Application

TAPR

1 Glen Ave., Wolcott, CT 06716-1442

Phone +1 972 413 8277, Monday–Friday, 9AM–5PM Eastern Time

E–mail contact@tapr.org   URL www.tapr.org

Join or renew online at https://tapr.org/logout/

Benefits of a TAPR Membership:

- *Subscription to the quarterly PSR*
- *10% off most TAPR kits and publications*
- *Access to the TAPR digital library*
- *Latest information on TAPR R&D projects*
- *Co-sponsor of the annual TAPR-ARRL Digital Communications Conference (DCC)*

Name____ Call Sign_______________________

Address____

City_____________________________ State/Province_________ Postal Code_____________

Country_________________________ Daytime Phone No._____________________________

E-mail Address____

New ☐ Renewal ☐ $30 X ____ number of years = $_______ total

Payment Method: Check ☐ Credit Card ☐ Money Order ☐ Cash ☐ Don't send cash through the mail !

STOP! Provide the following information only if paying **by mail** with a credit card:

AMEX ☐ Discover ☐ Mastercard ☐ VISA ☐

Credit Card No. _________________________ Expiration Date_______________ Security Code_______

Card Holder's Name____

TAPR is a community that provides leadership and resources to radio amateurs for the purpose of advancing the radio art.