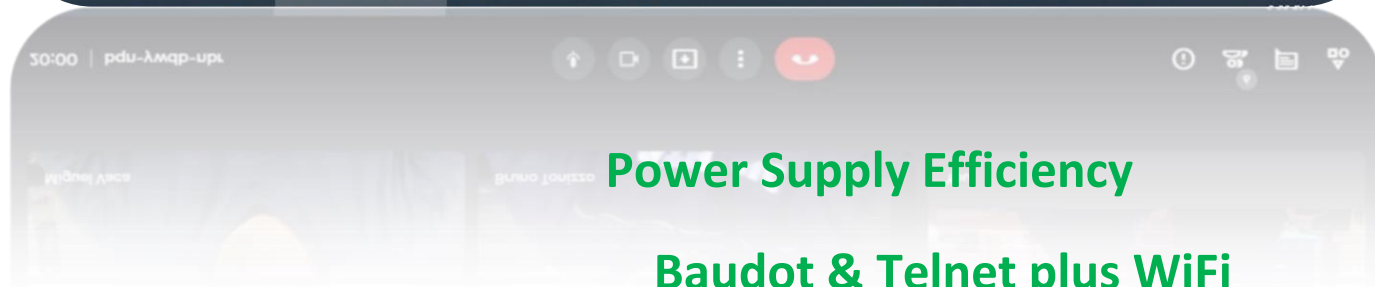




GATEWAY

**The Official Magazine of the Gippsland
Gate Radio & Electronics Club Inc.**

August 2021



Power Supply Efficiency

Baudot & Telnet plus WiFi

Ghidra

And More

18/08/2021



Cover photo, We are NOT back – from lockdown, (I spoke too soon) so roll up - your sleeves.

Contents.

- 3 – President's message
- 4 – From the Editor, Ghidra
- 5 – Baudot & Telnet plus WiFi
- 9 – Power Supply Efficiency
- 11 – Interesting YouTube Videos
- 12 – Meetings on Google
- 14 – Club Information

Note: - club meeting minutes are now via a link in club emails sent out by the secretary.

Event Queue

August:

20th General meeting, [see club emails](#)
28-29th ALARA contest – Courtesy WIA

September:

3rd Prac/Natter night, [see club emails](#)
17th General meeting, [see club emails](#)

October:

2-3rd Oceana contest – Courtesy WIA

November:

13-14th Antennapalooza

**Club run events are only possible with the involvement of ALL members.
Without volunteers to coordinate and participate in club events the club will fail to prosper**

GGREC President's Message

President's Message August 2021

All aspects of our lives are being impacted by the pandemic. We are very fortunate that we can still keep meeting in numbers on the air via Amateur radio. The Sunday WIA broadcast keeps us in touch with the latest amateur radio news and Paul's excellent monthly magazine provides lots of reading. Our on-line meetings provide another avenue to meet and see each other on the computer screen.

HF propagation is sporadic but those of you that have ventured into the world of digital radio can still make worldwide contacts. There are many ways we can enjoy our hobby and many opportunities to try new modes of transmission to try. I warmed up my soldering iron and built up a kit that I had waiting to be assembled. That exercised my brain as I needed to understand how the circuit worked to make sure it was all working as it should.

Ian Jackson, VK3BUF is planning to hold another Antennapalooza this year on the 13th/14th November. This year's event will focus on all aspects of Mobile communications. If you are planning to stay overnight get your tent or caravan ready for the event. Let's hope the lockdown is over by then.

Kind regards,
Bruno Tonizzo VK3BFT

Gateway is the official journal of the Gippsland Gate radio & Electronics Club.
It is published in good faith for its members and other interested parties. The articles here in do not necessarily represent the views of the committee or the membership.
Any material that may be copyright has been included with permission where available.
If there is an issue with copyright please contact the editor

From The Editor



While doing my regular YouTube scans for interesting content – as free-to-air TV seems to have lost the plot, I came across this one



<https://youtu.be/Nyqo-XOuHK4>

I was in two minds as to inserting it into the regular YouTube section, however I thought it deserved a few words. Apart from the most in-depth PC failure investigation I have ever seen – fairly normal for this channel, And well worth adding to your 'Subscribed' Youtube channel lists.

<https://www.youtube.com/channel/UC3bosUr3WlKYm4sBaLs-Adw>

However the part that most caught my attention was the software they were using to dive into the inner workings of this sick computer. It was written, and then released by the US 'National Security Agency', or NSA, Usually these folk are super secretive and don't divulge anything, even if the wheels have fallen off this ideal a few times – leading to some REALLY nasty viruses / ransomware attacks. So to see them release this really caught my attention.

It basically pulls apart software so you can see just what it is doing, and not just PC X86 code, also a few 8 bit micro's and microcontrollers. Unfortunately not the National SC/MP (scamp) that currently has my attention – however I have found a disassembler for that, so I won't have to do it the hard way – manually looking up every instruction etc.

So it looks like I've been pulled away from radio fiddling again, however with the weather not been the best at times, probably good, as most of my radio jobs are outdoors. My 2M vertical is really sick, and my 70cm activity had been dealt a bit of a blow ever since the clubs repeater was moved to a 7MHz split. I never would have believed an in-band move could change things so much. I now get frequent multi-path issues into the repeater.

As for future articles, I am hoping to take that Telnet thing a tad further, as I look for an easy-ish way to get my creations hooked up via WiFi etc. Yes, I could get a cheap Raspberry Pi Zero and pop one inside things, however putting an all but full flung Linux computer into everything seems a bit like using a nuclear bomb to crack walnuts to me. (and a potential security nightmare). I have a few other idea's, however I really am after other people's input, particularly in the radio field.

Paul VK3TGX

Baudot & Telnet plus WiFi...



In my project to connect this old Baudot Time and date generator to the 'net (because everything is going to 'the cloud') I needed to convert its output to ASCII.

Well that's not entirely true, I could just pump the 5 unit code through something like Telnet, However 99.99% of anything

that talks Telnet would have no idea how to process that data. So to make life easier I wanted to convert it, so if nothing else I could look at it with 'TerraTerm' or 'Putty' etc. I mentioned making a hardware converter (with lots of flashing lights), however if it's to end up on the 'net then it's going to have to go via a computer of some sorts, so why not just get that computer to do the converting. I already have an industrial controller computer running my security cams, and it has (because of its industrial roots) 6 serial, as in RS232 ports, just waiting to be used.

```
*Baudot-ASCII,2.py - Y:\Python project\Baudot-ASCII,2.py (3.8.10)*
File Edit Format Run Options Window Help

# Baudot (ITA2) to ASCII converter

import serial
import time
ser = serial.Serial('com6', baudrate=50, timeout=5, bytesize=5, stopbits=1.5, rtscts=False, dsrdtr=False)
ser2 = serial.Serial('com10', baudrate=1200, timeout=1, bytesize=7, stopbits=1, rtscts=False, dsrdtr=False)
ser2.writeTimeout=0

cr=chr(0x0D) # carriage return
lf=chr(0x0A) # line feed
dcl=chr(0x11) # figs shift
dc2=chr(0x12) # letters shift
nul=chr(0x00) # null char
bel=chr(0x07) # bell char

lettrs = [nul, 'E', lf, 'A', ' ', 'S', 'I', 'U', cr, 'D', 'R', 'J',
          'N', 'F', 'C', 'K', 'T', 'Z', 'L', 'W', 'H', 'Y', 'P', 'Q',
          'O', 'B', 'G', dcl, 'M', 'X', 'V', dcl]

figs = [nul, '3', lf, '-', ' ', '"', '8', '7', cr, 'b', '4', bel,
        '!', '!', ':', '(', '5', '+', ')', '2', '#', '6', '0', '1',
        '9', '?', '%', dcl, '.', '/', ';', dc2]

shift=0 # shift state (figs/letters) of Baudot%
loop=0 # 0-50 character output loop
delay=0.15 # default Baudot character spacing

while 1:
    char=ser.read(size=1)
    if(len(char)!=0):
        chars=ord(char)
        char=chr(chars).encode()

        if(chars==0x1F): shift=0 # letters shift
        if(chars==0x1B): shift=1 # figures shift
        if(shift==0): aschar=lettrs[chars]
        if(shift==1): aschar=figs[chars]

        ser2.write(aschar.encode())

    x=ser.in_waiting
    if (x<=1): delay=delay+.001
    if (x>=15)&(delay>=.05):delay=delay-.001
    loop=loop+1
    if loop>=50:
        loop=0
        print(x, delay)
    time.sleep(delay) # 50Baud characters take 0.15 sec per character

Ln: 39 Col: 0
```


So enter Python, a programming language that is proving extremely popular lately. It's one of the 'promoted' languages on the Raspberry Pi, and it also runs on a lot of cloud servers. So why not get into something that is very current. (Kind of justify this 'trip' into obscurity)

Now I am no Python guru by any shot of my imagination, this is all but my first bit of code. (The actual first printed 'Test The quick brown fox.... out the same serial port)

It starts by opening two serial ports, 'Com6' being the input connected to the Time & Date box.

As Baudot has two shift states, Figures & Letters, there are two arrays, called, surprise surprise, 'lettrs' and 'figs', there is also a variable called 'shift' that remembers what state we are in.

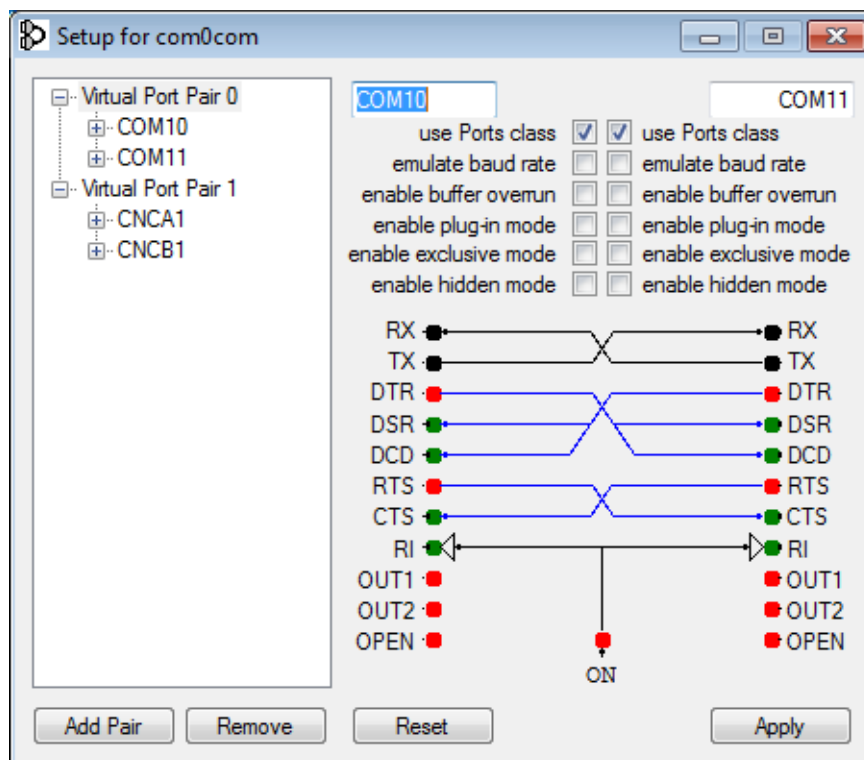
Now don't get confused by a modern keyboard, as the shift key on it operates in a somewhat completely different way. Modern keyboards basically generate ASCII, a 7 bit code that has both upper and lower case letters, and a separate code for every symbol (and more) that is on your keyboard. (and 'yes' to the smarty-pants, there is no ASCII for 'Ctrl', 'shift' etc.) In Baudot the character codes have two alternative characters, and the only way to know which one to use is by remembering what last shift code was sent, that could have been many many characters earlier. (Traditionally, one re-sends the shift code after a 'carriage return' & 'line feed' pair, very important in the early mechanical days as it provided extra time for the carriage to actually return all the way) In current times, with RTTY, a letters shift state is assumed after any space characters, this is to allow for noisy radio links where a 'letters' could get lost, or another character mangled into looking like a 'figures shift', working on the assumption that a plain text QSO is probably not figures, but rather letters. However if you want to monitor a 'traditional' transmission, you'd be advised to disable that function in your software.

Back to my Python, now depending on which shift state we are in, either the 'figs' or 'lettrs' table is used to convert back to ASCII. There is a bit of a twist in my code, normally the Baudot 'letters' and 'figures' code would produce no ASCII output at all, just set the 'shift' variable in my code, however I have it generate an ASCII 'DC1' or 'DC2' to represent where those Baudot characters were, these control codes are usually invisible on normal Telnet screens, however if I want to delve deeper into things I can find out exactly what was originally sent, I can, and optionally 'bit perfect' back into Baudot, where I can (for example) see the 8 'letters' shifts used to correct a paper tape. (Letters shift is all holes on paper tape, overwriting any errant text)

The last bit of code is a work-around to overcome the bursty nature of the buffered characters Windows was delivering to me, quite often in 6 character blocks. 50 baud TTY usually has the characters in a steady 150 milliseconds per character stream. At first thought a 0.15 second delay after the last should work, but no, you have to allow time for this Python code, so in reality you need a delay a tad less than 150ms. So rather than guessing, this code monitors the buffer and continually tweaks the delay in one millisecond increments to keep the buffer between 1 and 15 characters. The output is now steady and much more TTY in appearance.

In reality, in this usage, I only need figures, I could simplify the code somewhat, however having a complete converter will save me possible future confusion if I use it elsewhere.

After conversion, you can probably see it just goes to another PC comm port, however this is actually a virtual port, not a physical device on this computer. I am using a program called 'com0com' (com 'zero' com) that generates pairs of virtual ports that are linked together. It's like I connected two spare ports on this computer together with a crossover cable, but without actually wasting any real ports. My Python code thinks 'Com10' is real, it acts that way, however anything sent to it just comes back on Com11



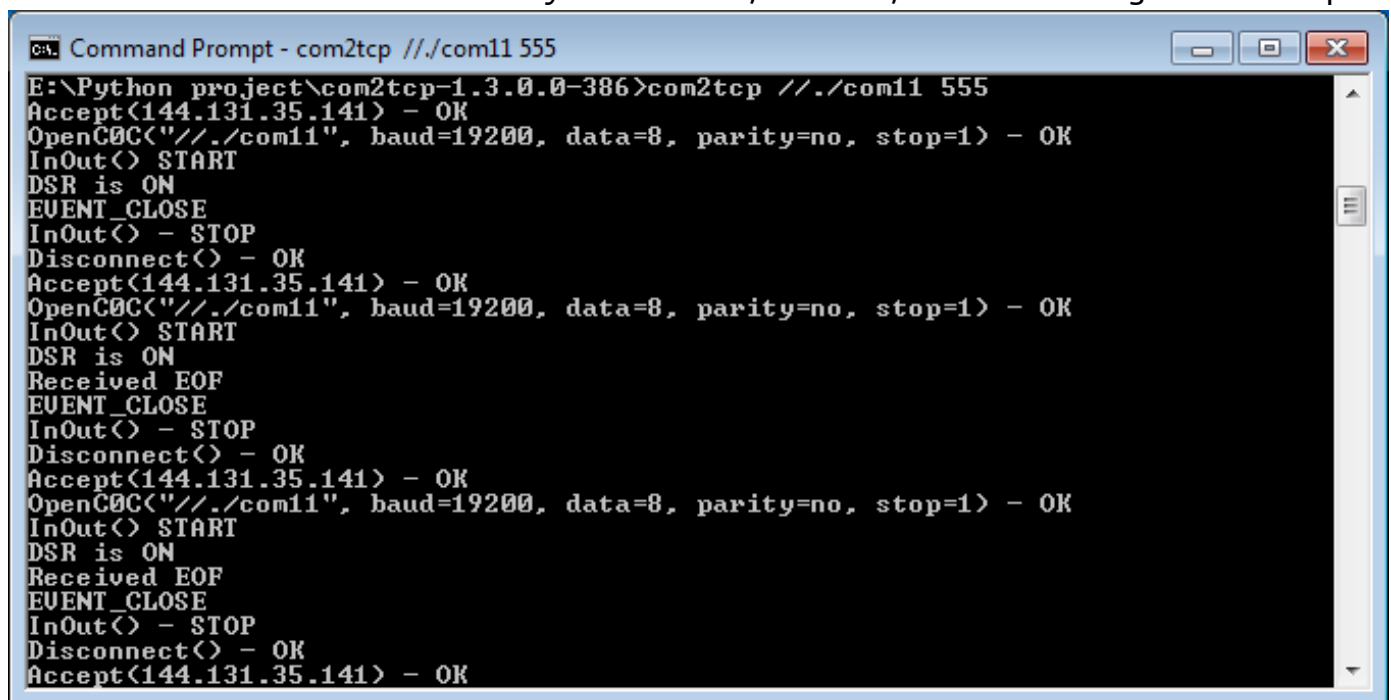
As setup, I have two com port pairs, COM10-Com11 & CNCA1-CNCB1. I can create many more as needed.

This can be used with virtually any software that talks to serial ports. Say HamRadioDelux, and your code to translate its output to talk to a radio it normally does not support. I.E., HRD talking to a Philips PRM8030, thinking it's a Yaesu 'XXX' that it does support.

In this case, COM10 is connected to my Python code, and COM11 is connected to another utility called 'com2tcp' that makes this port appear on my network as a Telnet host.

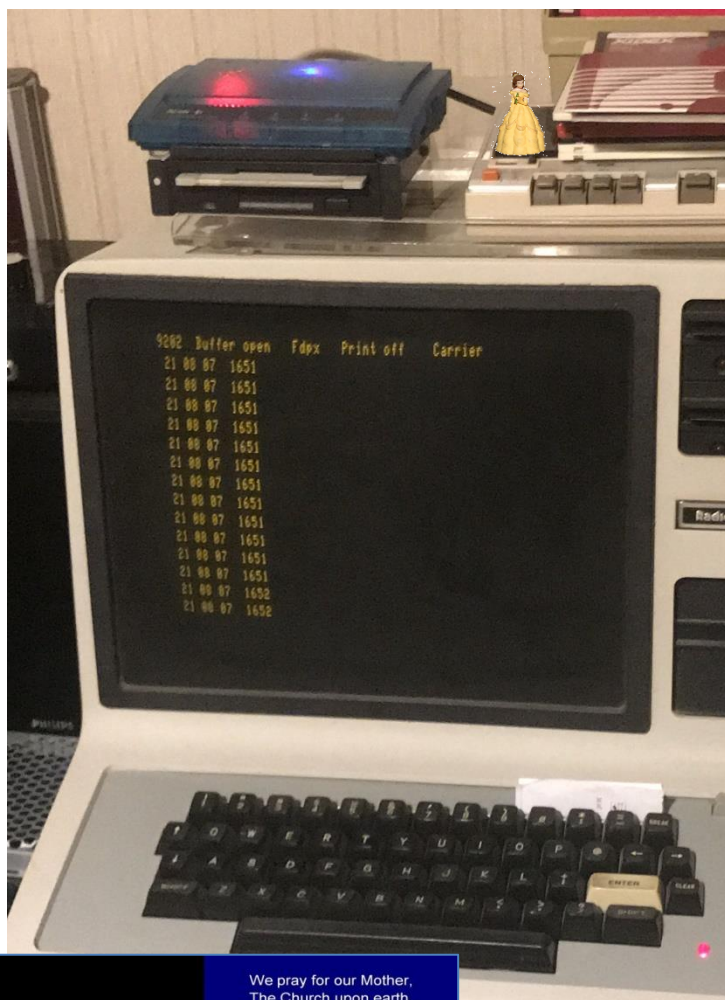
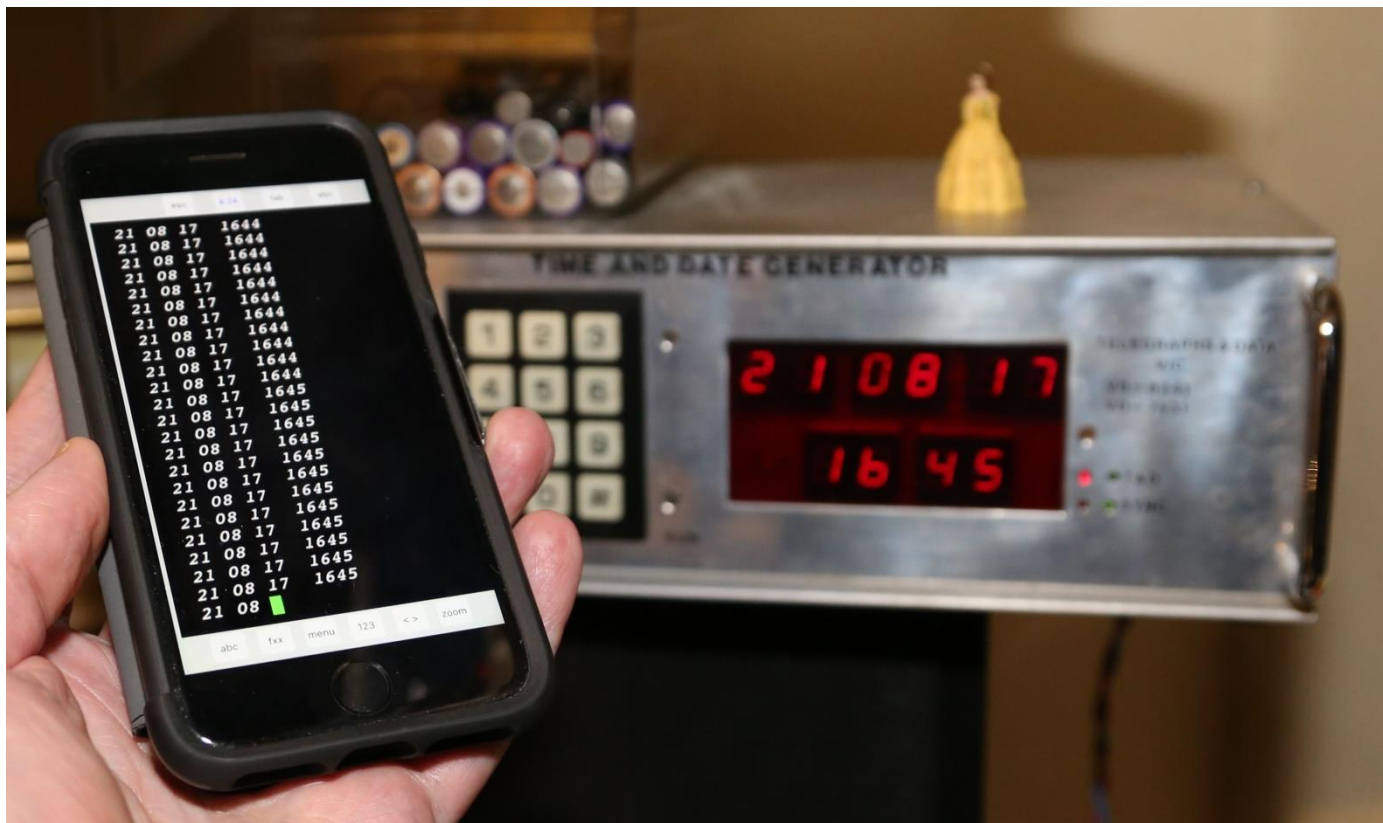
In this case, I open a command prompt (DOS window), and type 'com2tcp //./com11 555' now com11 appears at 192.168.0.8, port 555, usually expressed as '192.168.0.8:555'

'192.168.0.8' is the local network address of the computer, normally the default Telnet port is 23, however I'm using 555 instead – I intend setting up several more, so unique port numbers are needed. Ports below 256 are fairly well defined, however, above that things are more open.



This is both the starting of 'com2tcp' & some of its dialog as I connect, and disconnect.

If you would like to connect outside of your home network, you will need to access your router and enable port forwarding, however NOTE, Telnet is NOT SECURE, all the transactions are in plain text, so don't send important passwords, bank details, etc. etc. there are much better protocols for this, like SSH, (Secure Shell).



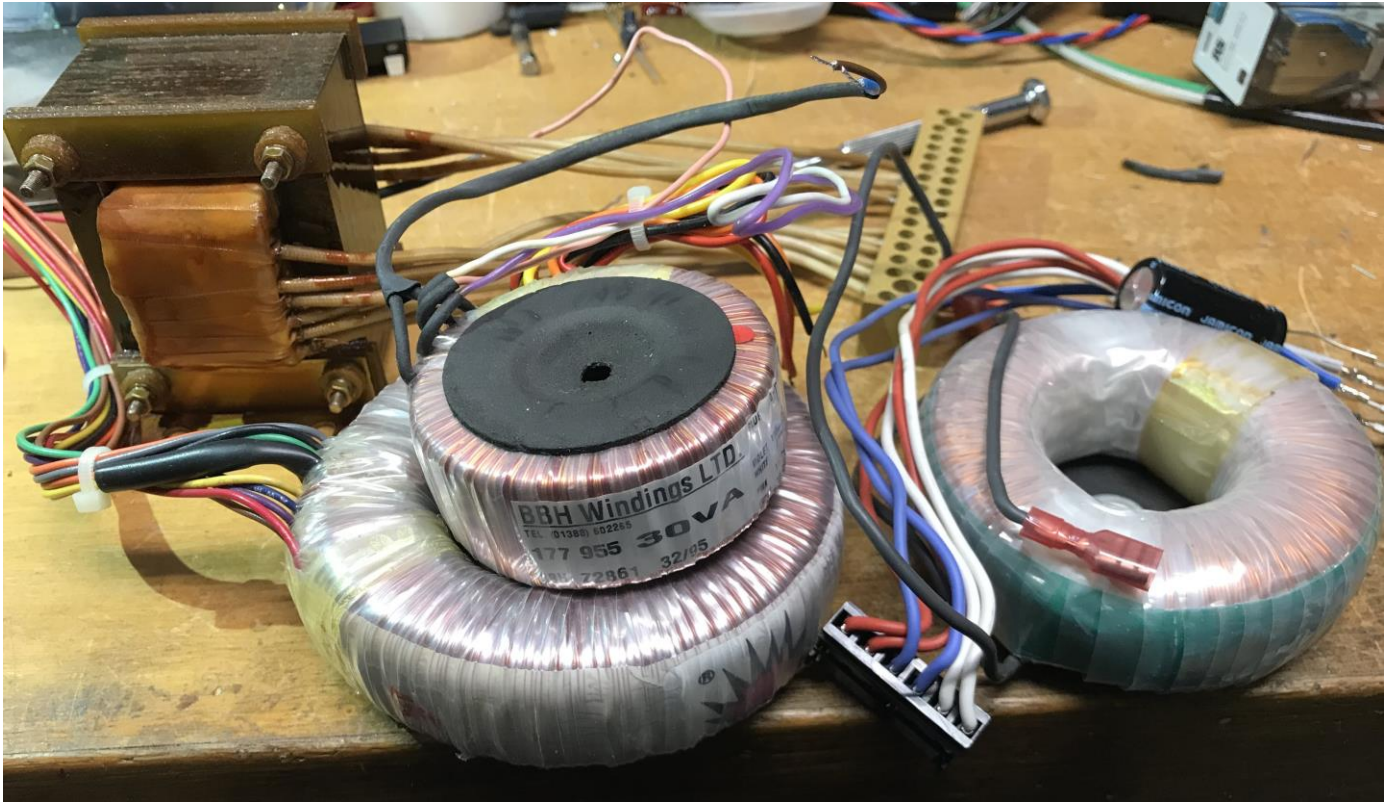
Two examples of accessing the T&D box via WiFi, the second being on my old TRS-80 model 1 via the WiFi64 modem I featured a few years back. This is one reason I'm playing with Telnet, rather than SSH, it's a darn sight easier to access with minimalist hardware, unlike encrypted format's that require security tokens etc. etc. a whole lot of 'black magic' that is kind of well beyond the scope of what I am doing.

Being used to amateur radio where everything is in the open, no encryption, this Telnet thing seems normal to me, just like blasting a few hundred watts of RTTY up the stick for anyone to receive.

As an extra – just because I can, I accessed it from church, and added the T&D to the bottom of a mass live stream, so who knows how far this T&D has gone?, around the world?, not that anyone will have noticed, or cared. Super impractical....



Power Supply Efficiency



Everyone knows a linear power supply is inefficient, and a switchmode is way better? Well are they? 'The other day' I was after a source of DC, about 40 to 50V – so no regulation needed, the device being powered has that all sorted internally.

The contenders for the job were a Meanwell 48V Switchmode, and a few old school iron core transformers, and the odd toroidal for good measure..

The load was to be about 150mA, or roughly 7W, so not much, but with 24/7 service probably ensuing, reasonable efficiency seemed a good ideal to strive for.

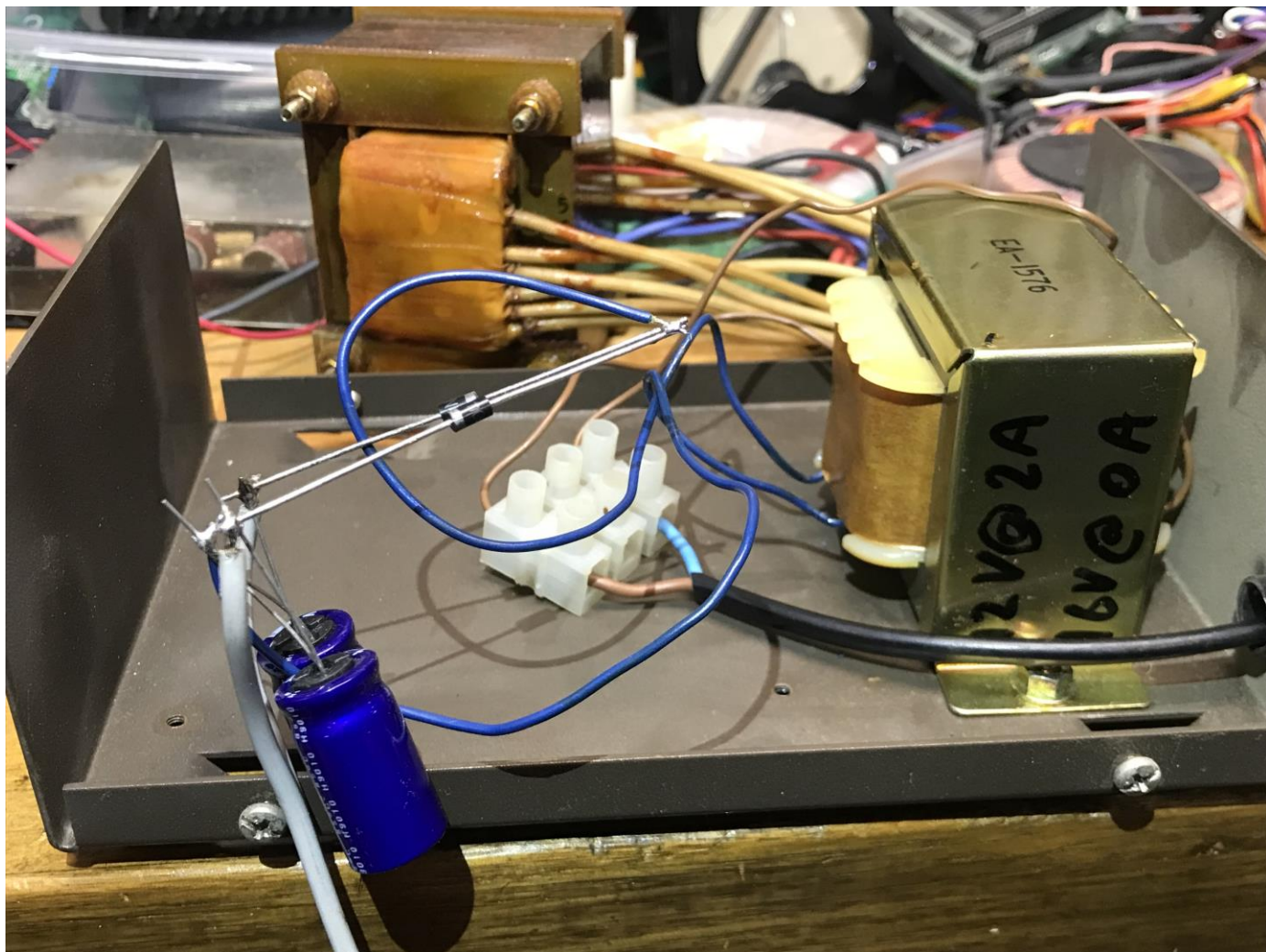
For a test load I ended up using a 240V 150W par flood lamp, at 40V it was drawing about 130mA, in the ballpark for these tests.

The first was that 'E I' transformer at the back, with no load it was drawing 2W, with a just perceptible heating of the laminations after being left on for a few hours. With the 5W load, that pushed us up to 7.5W, not the best but ok. So now let's compare it to the Switchmode.



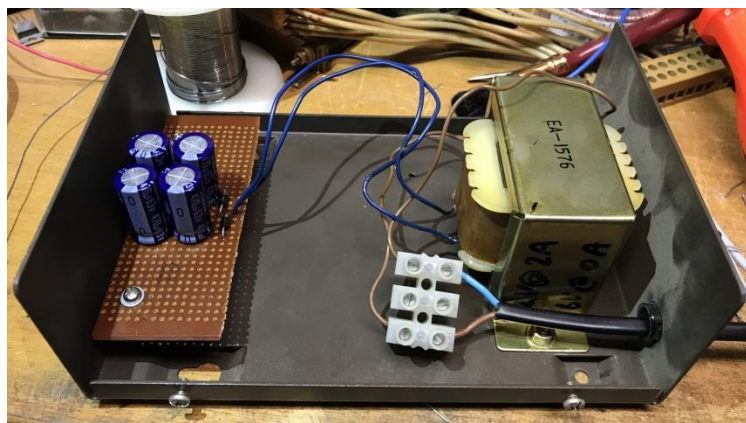
As it stands, with a fan (running as slow as possible) it was drawing 4.4W – Yikes, and with the load of 140mA that rose up to 12W, not good

So I pulled the fan, now we were down to 9.8W with my, now 7.5W load (due to the increased voltage) – mmmm reject! Admittedly this thing is rated for 1A, so it's being very lightly loaded here, I'll have to do some more tests. It's got to be better at near 1A, surely....



I then tried this old Telstra transformer, these were never intended to be used as a power transformer, normally they were run the other way around producing ring current for small office phone systems. However they seem fine with 240V run up them. I have another one serving as a battery charger, it's been running 24/7 for years with no issues. They also pass the Mega test at 1KV, again, all ok. The only problem being insufficient output volts, so a doubler was quickly hacked together to bring it up to requirements. With my 5W load, it was pulling 5.5W from the mains, so winner winner.....

Unfortunately none of the toroid's had suitable windings, so back into the junk box for another day. Yes I could easily add a winding, however with this Telstra ringer doing an acceptable job, why bother, especially as the former came in a sturdy steel case, complete with tapped holes to hold my doubler board.



I prefer metal boxes for mains power supplies, yes the smoke can get out, but hopefully not the flames.

So there you go, don't assume anything, get a meter out and test it.

Yes the mains wiring sucks, it's far from complete

Paul VK3TGX

Interesting YouTube Videos



The Coolest Radio You've Probably Never Heard of

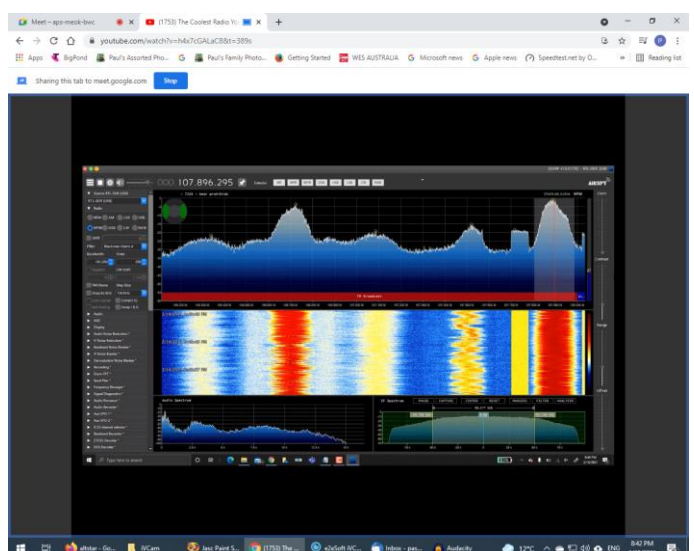
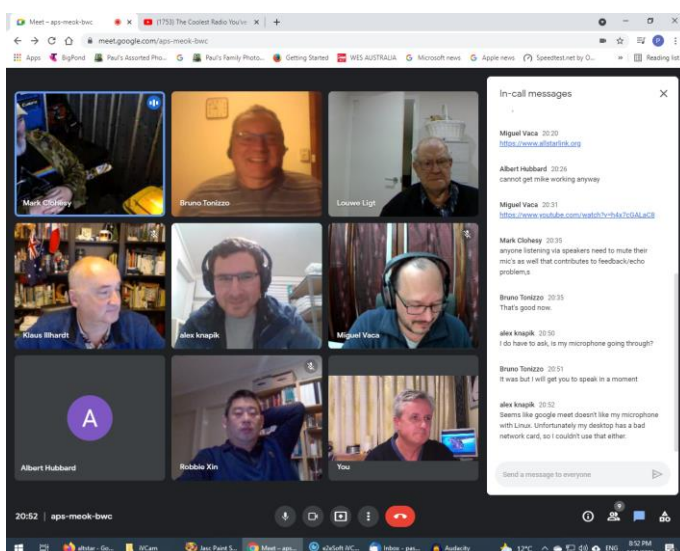
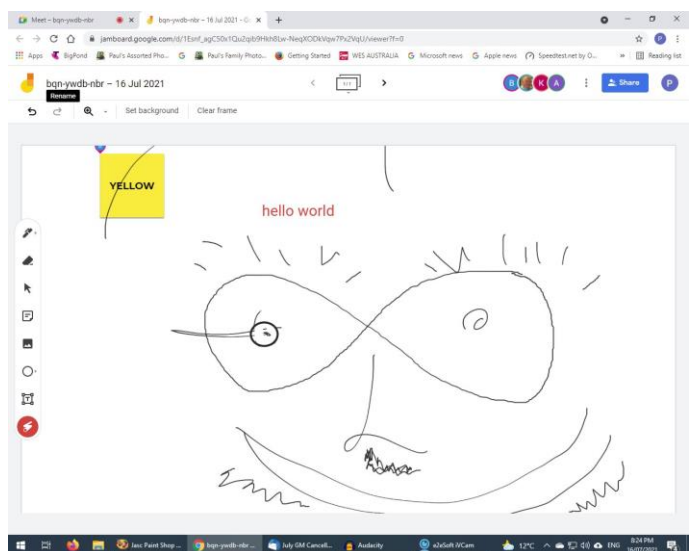
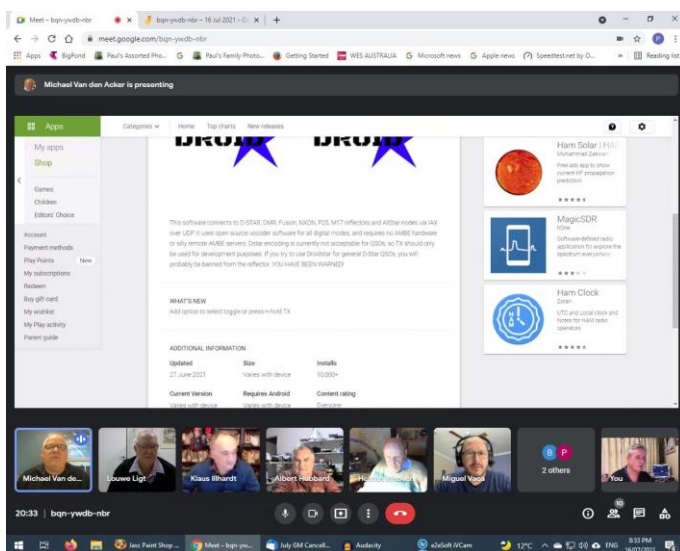
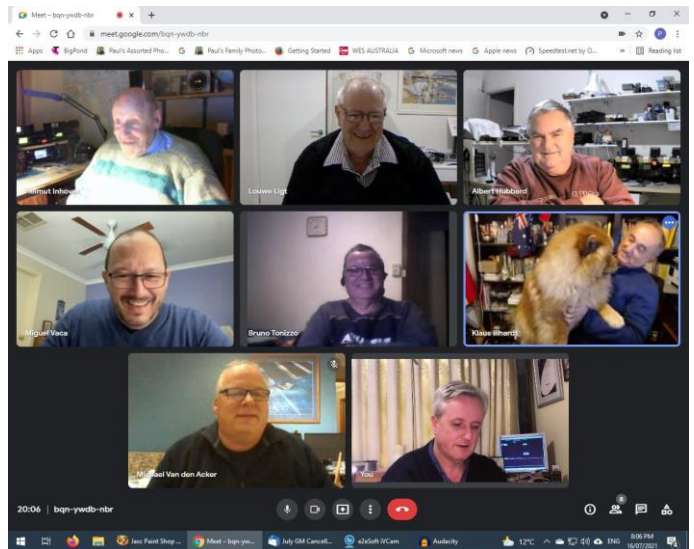
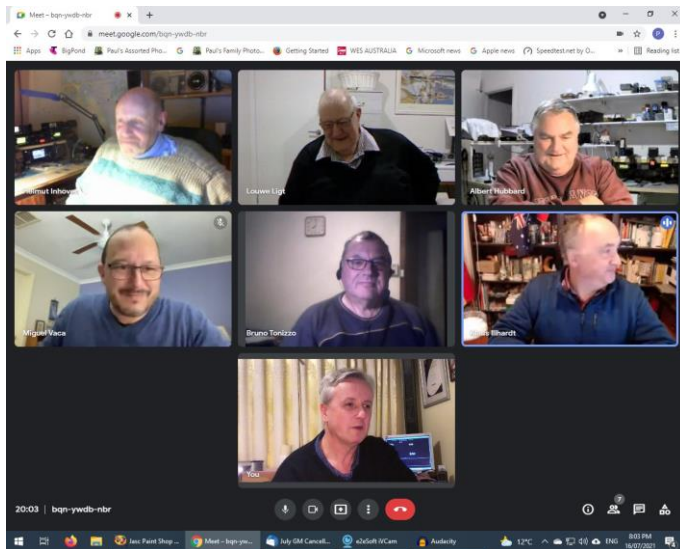
<https://youtu.be/h4x7cGALaC8>



DIY CHEAP WiFi Modem For Vintage/Retro Computers

<https://youtu.be/cfLNiBDwIL8>

Meetings on Google





alltrophies.com.au

Specialising in Corporate Awards Clubs and Schools

03 9762 4844

sales@alltrophies.com.au

ABN 69 089 590 590

**71 Intrepid Street
Berwick VIC 3806**

- Honour Boards
- Acrylic Awards
- Silverware
- Plaques
- Clocks & Barometers
- Medals & Badges
- Club Awards
- Engraving
- Corporate Awards
- Sport Awards
- Customised Awards
- Brass/Bronze
- Photo Plaques
- Commercial Engraving





The GGREC is an affiliated club of the WIA

WIA Affiliated Club

We also give Thanks to



For their generous support over the years



Club Information



Meetings 20:00hrs on third Friday of the month at the
Cranbourne Guide hall, Grant Street Cranbourne
Prac/Natter nights first Friday in the Peter Pavey Clubrooms Cranbourne 19:30hrs
Visitors are always welcome.

Office bearers

President	Bruno Tonizzo	VK3GHM	Web Master	Mark Clohesy	VK3PKT
Admin Sec	Miguel Vaca	VK3CPU	Magazine Editor	Paul Stubbs	VK3TGX
Treasurer	Klaus Illhardt	VK3IU	Property Officer	'committee'	
General 1	Bruce Williams	VK3BRW	Assoc. Secretary	Miguel Vaca	VK3CPU
General 2	Leigh Findlay	VK3FACB			

Call in Frequencies, Beacons and Repeaters

The Club Station VK3BJA operates from the Cranbourne Clubrooms.
6m Repeater Cranbourne VK3RDD, In 52.575 Out 53.575 CTCSS none
70cm Repeater Cranbourne VK3RGW, In 431.425MHz Out 438.425MHz CTCSS 91.5Hz
VK3RGW Repeater supports Remote Internet access (IRLP), Node 6794 **offline**.
70cm Repeater Seaview VK3RWD, In 433.575MHz Out 438.575MHz CTCSS 91.5Hz **offline**
Simplex VHF - 145.450MHz FM, Simplex UHF - TBA
VK3RLP Beacons 1296.532MHz & 2403.532MHz (**currently offline**)

Membership Fee Schedule

- Pensioner member rate \$40.00, Extra family member \$20.00
Standard member rate \$50.00, Junior member rate \$25.00
Fees can be paid by EFT to BSB 633000 - Account 146016746
• Always identify your EFT payments
• Membership fees are due by each April Annual General Meeting (AGM)

Magazine Articles to editor@ggrec.org.au Cut off, 10th of the month
All other Club correspondence to: secretary@ggrec.org.au
or via post : GGREC, 408 Old Sale Rd, Drouin West 3818
GGREC Web Site & Archive may be viewed at: www.ggrec.org.au
Website errors, contact web master: webmaster@ggrec.org.au
Facebook Page www.facebook.com/GippslandGate