How-To Guide: Configuring TCP to Virtual Serial Port(s) to use NMEA Message Based Time Tools

This is a how to guide on how to get started using low-cost and/or free tools to enable the GPS receiver of a project/product like Themis – that can be configured with an optional radio interface to broadcast UDP and/or TCP packets over a LAN containing NMEA messages – so time tools like NMEATime and similar tools can be easily enabled.

http://www.visualgps.net/#nmeatime2-content

Themis can use any radio interface that uses the ConnectBlue (now u- blox) form factor and having a classic UART interface. Themis takes the UART output from the LEA-6T module and passes that to the UART on the ConnectBlue module.

This document will cover the WiFi radio option that is available, some basic configuration discussion of that; and how to enable the GPS messages from Themis to set a PC's clock (as an example) using tools like NMEATime.

The WiFi config in this case is set up with the ConnectBlue tools to send both UDP packets to port 123 and TCP packets to a port like 3434. Note there are several profiles that can be set on the WiFi module – for the remainder of this document we will discuss UDP packets on port 123. The WiFi module is set for DHCP and picks up an address of 192.168.2.196 as an example server IP address on the local LAN. Port 123 is used even though NTP isn't enabled (yet) on Themis and this is just for demonstrating how NMEA messages can be used over a local LAN. Some may wonder why I chose port 123; I did because I wanted to make sure it would work since I am planning on enabling a very light weight SNTP server on Themis. Ports like 3434 work as well to use a port above the well-known ports below 1000.

Eltima has a lot of great tools – and that route is certainly good; but most Hams have a rather limited budget for this sort of thing and I'll show a way to do this that's free (although it costs some time and isn't as elegant as the Eltima tools). Eltima does have an evaluation time that gives you plenty of time to evaluate and I would recommend that be done as a first step just because it's a little easier (some would say a lot easier) to set up and maintain than what I will show here:

https://www.eltima.com/products/serial-over-ethernet/

There's a tool that's free and you get the source code as well – com0com and it's available for download on Sourceforge:

https://sourceforge.net/projects/com0com/

Take a look at that page – and don't be put off when you see the telnet protocol being called out explicitly – that's an option and obviously when the tool was first developed it was very handy to be able to run telnet over TCP and through serial interfaces.

So, install the com0com software. Follow the instructions to install and then use the command line tool for the following:

com0com->Set Up Command Prompt

I like using the set up command prompt because it seems to set up the virtual com ports more consistently than the GUI tool – but maybe that is just my experience.

On the command like just type:

Install PortName=CNCA0 PortName=CNCB0

It will create the virtual com port connection and you may notice it will automatically assign (or should) CNCA0 to COM2. You may need to make sure COM2 is available by checking your com assignments and make sure you don't have any 'zombie' com ports that could conflict. There's plenty of discussion on this that can be found with your preferred search engine so if you have trouble with this step – please make sure you don't have a bunch of zombie ports making your life difficult.

After this is done (and your system could ask to reboot) – once you do that you can open the GUI tool and you will see the new port assignments and/or type list in the command line tool:

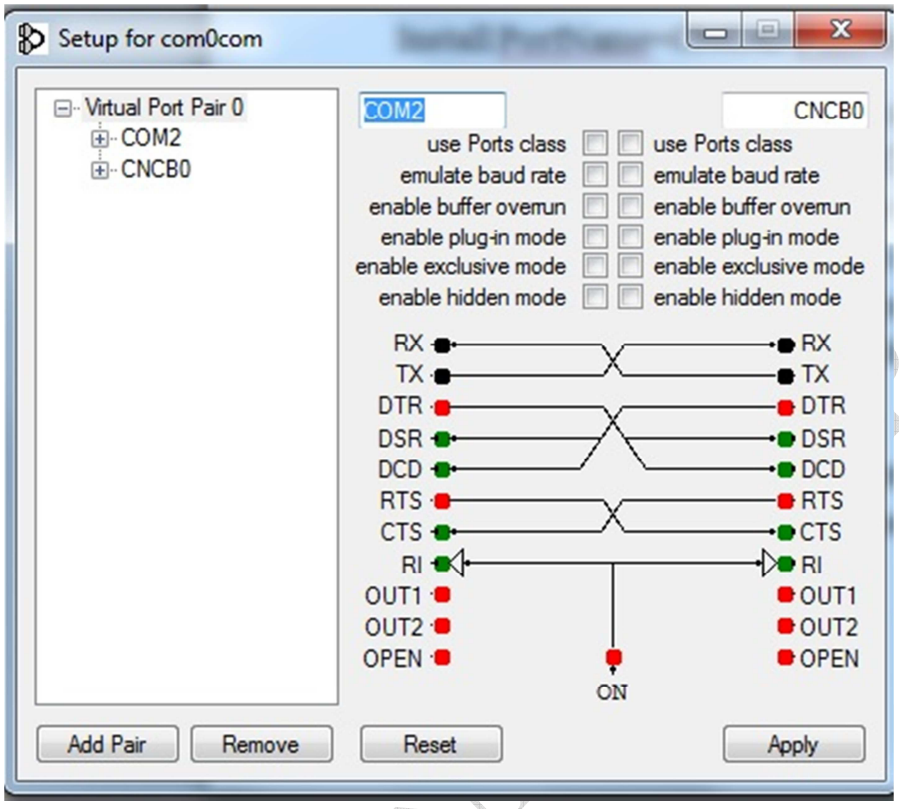

Figure 1: com0com GUI showing COM2-CNCB0 pair.

You should see something like this from the command line tool:

command> list

CNCA0 PortName=COM2

CNCB0 PortName=CNCB0

From the Sourceforge page – download the com2tcp tool:

https://sourceforge.net/projects/com0com/files/com2tcp/1.3.0.0/

Install that – it's good to take a look at the docs of course but to configure - to follow the example that has been discussed here:

C:\Users\John\Downloads\com2tcp-1.3.0.0-386\com2tcp-1.3.0.0-386>com2tcp –baud 115200 --ignore-dsr \\.\CNCB0 192.168.2.196 123

Note in this case the WiFi module is using the IP address 192.168.2.196 and the port address is 123 as previously discussed.

The program will display this if a connection is made correctly:

OpenC0C("\\.\CNCB0", baud=115200, data=8, parity=no, stop=1) - OK

Connect("192.168.2.196", "123") - OK

InOut() START

DSR is ON

Note if it can't connect to the server it will time out and stop.

You can use the hub4com tool as well if needed to distribute the NMEA messages – take a look:

https://sourceforge.net/projects/com0com/files/hub4com/2.1.0.0/

The NMEATime tool is configured to use COM2 and 115200 BAUD.

Once the packets are being received – you will see a screen like the following:

(of course this is an example – just as a reference – you can use the tool of your choice)

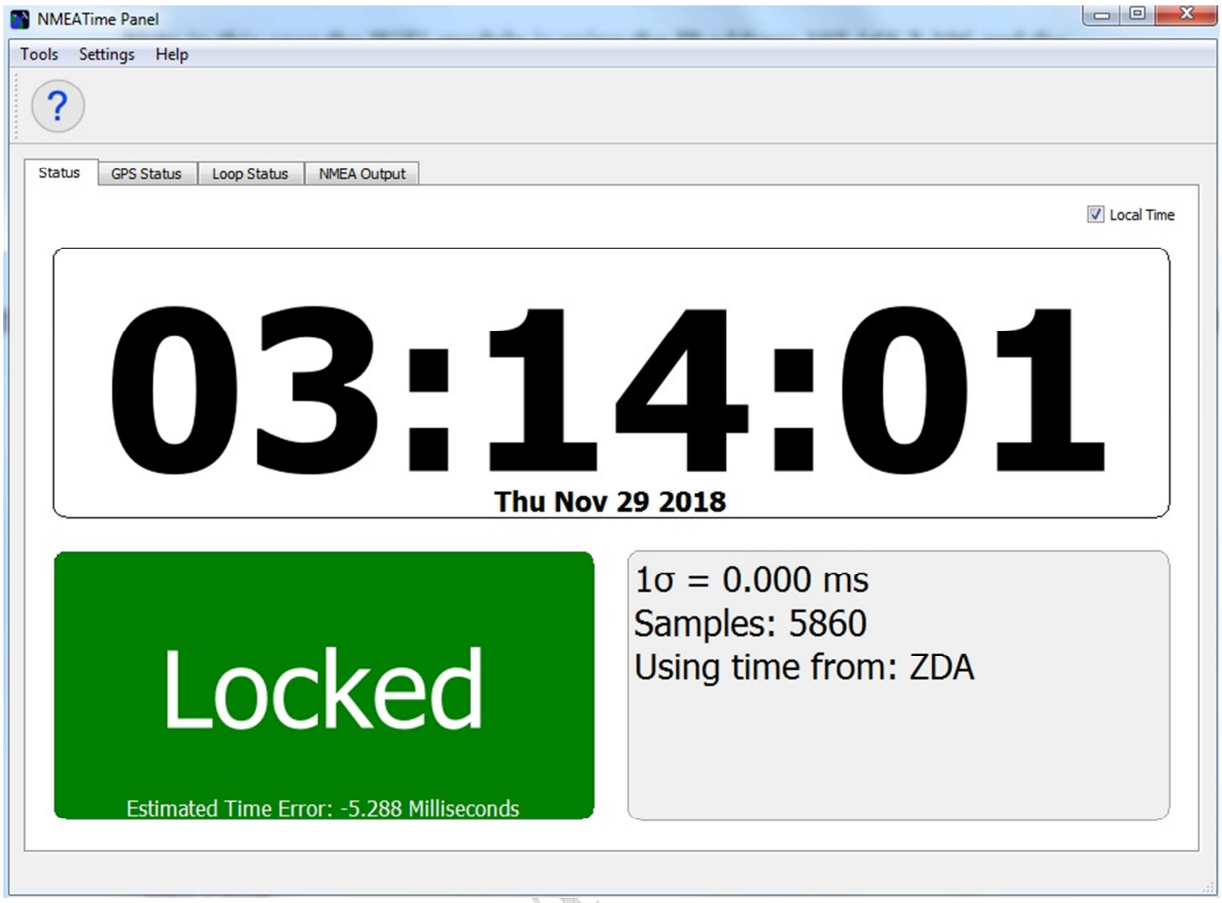

Fig. 2 – NMEA Time – showing a typical screen when lock achieved with GPS 'server'.

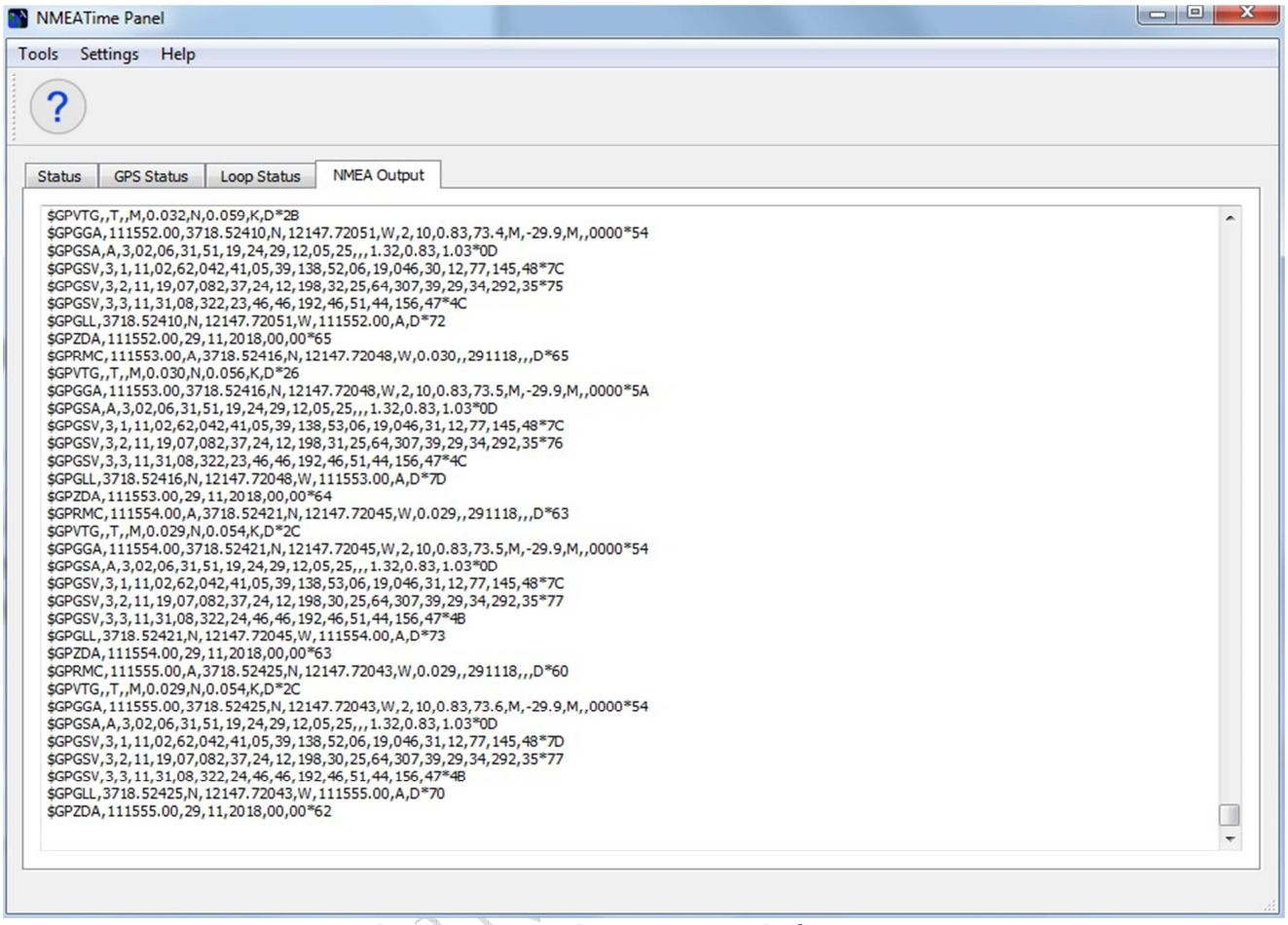

Fig. 3 – NMEA messages being received over network from GPS ‘server’.

I hope this rather brief ‘how-to’ guide has been helpful.